Miskolci Egyetem

Gépészmérnöki és Informatikai Kar

Sztochasztikus módszerek alkalmazása optimalizálási és adattisztítási eljárásokban

című PhD értekezés

Készítette:
Gurka Dezsőné Csizmás Edit
okleveles matematika-fizika-számítástechnika szakos középiskolai tanár

Hatvany József Informatikai Tudományok Doktori Iskola

Doktori iskola vezető:
Prof. Dr. Szigeti Jenő
egyetemi tanár

Témavezető:
Prof. Dr. Kovács László
egyetemi tanár

Társtémavezető:
Dr. Kovács Edith Alice
egyetemi docens

Miskolc, 2023.

Nyilatkozat

Alulírott Gurka Dezsőné Csizmás Edit kijelentem, hogy ezt a doktori értekezést magam készítettem, és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Miskolc, 2023. március 3.

Gurka Dezsőné Csizmás Edit

Témavezetői ajánlás
Gurka Dezsőné Csizmás Edit
doktorandusz részére

Gurka Dezsőné Csizmás Edit doktorjelölt hallgató dolgozatában egy hosszabb és több kérdésre is kiterjedő kutatói munka eredményeit foglalta össze. Mélyebben megismerte és elemezte a sztochasztikus modellezés és optimalizálás módszereit és azokat több gyakorlati feladatban is sikerrel alkalmazta. Ezen területek közé tartozik a kopula függvényekkel való többdimenziós eloszlások modellezése, ennek alapján szcenárió generálások, az összefüggések hatásának vizsgálata a kockázati mutatókra, valamint sztochasztikus optimalizálási feladatok hangolása, továbbá az hiányos adatok pótlása forrásadatok esetében.

A PhD képzés alatti tapasztalatok alapján Gurka Dezsőné Csizmás Edit munkája jól mutatta, hogy
- igényes és pontos a munkavégzés során,
- rendkívüli megbízhatóság jellemzi,
- külön dicsérendő, hogy doktoranduszi képzését az oktatói munkájával párhuzamosan végezte,
- több kutató csoport témájában is részt vett,
- több önálló ötlettel gazdagította az induló munkaterveket,
- kutató munkáját kitartás és céltudatosság jellemzi.

Gurka Dezsőné Csizmás Edit jól teljesítette az előírt tanulmányi kötelezettségeket és terv szerint, tudatosan fejlődött a publikációs teljesítmény területén is. Összesen 56, a dolgozathoz kapcsolódóan 21 publikációban szerepel szerzőként, melyek között vannak Q1-es, Q2-es és IF jelzésű folyóirat cikkek is. Emellett mindenképpen kiemelendő, hogy igen nagy számú konferencia publikációt is teljesített.
Az elkészített dolgozatában egyensúlyt teremtett az elmélet és a gyakorlat között, az elméleti elemzések eredményeit mintarendszerekben implementálta és tesztelte.

A jelölt eddigi eredményei és habitusa alapján támogatom a nyilvános védés megindításának az engedélyezését és a nyilvános vita megszervezését.

Miskolc, 2023.02.20.

Prof. Kovács László
témavezető

Témavezetői ajánlás
Gurka Dezsőné Csizmás Edit
doktorandusz részére

Gurka Dezsőné Csizmás Edit PhD dolgozatában több témán keresztül is a bizonytalanság által okozott problémák kezelésével foglalkozik.
A valószínűséggel korlátozott optimalizálási feladatok megoldásához kapcsolódó hozzájárulásainak fontos szerepét említem meg először. Egy kiinduló megengedett megoldás keresésére és a Genz-kód pontosságának dinamikus szabályozására dolgozott ki eljárást.
Egy másik sztochasztikus optimalizálási probléma a portfolió optimális megválasztásához kötődik. A probléma megfogalmazásához a másodrendű sztochasztikus dominancia szerepel feltételként. Mintagenerálással támogatta a portfolió megválasztásának robusztusságát, kétfajta megoldási algoritmus esetében is. Az is bemutatásra került kopulákkal kapcsolatosan, hogy milyen modellezési lehetőséggel bírnak és milyen szerepet játszanak a portfolió kockázatának számszerűsítésében.
A hiányzó adatok szinte minden adatokra épülő eljárásban problémát jelentenek, így különösen fontos ezek pótlása. Erre a problémára releváns változók alapján imputálási módszereket dolgozott ki, amelyek jobbnak bizonyultak az R programcsomagban implementált módszereknél. Egy másik eredménye pedig a szimulációk alapján való prefix-fában tárolás költségének becslése.

Az együttműködésünk során, kiemelendőnek tartom a kutatói kíváncsiságát, a jó és új szempontokat elővezető kérdéseit, kitartását, valamint azt, hogy nem egy kutatási iránynak kötelezte el magát, hanem új kutatási kérdések, kihívások megoldásait is kereste. Nagyon előnyös tulajdonsága, hogy több kutatócsoportba is be tudott kapcsolódni a kutatómunka elősegítésével.

A jelölt összesen 56 publikációval rendelkezik, a dolgozathoz kapcsolódóan 21 publikációban szerepel szerzőként, melyek között vannak Q1-es, Q2-es és IF jelzésű folyóirat cikkek is. Megjegyzendő, hogy további kutatási eredményei is vannak a disszertációban foglaltakon kívül is. Vannak olyanok is, amelyek még publikálásra várnak.

A jelöltet a disszertációja alapján, a publikációk alapján és a személyes közös kutatási tapasztalatok alapján támogatom a doktori cím megszerzésében.

Budapest, 2023. 03. 03.

Dr. Kovács Edith Alice
társtémavezető

Köszönetnyilvánítás

Köszönöm munkahelyi témavezetőmnek, Vajnai Tibor professzor úrnak támogatását, és hogy megismertette velem a kopulákkal való modellezés lehetőségeit.

Köszönöm témavezetőmnek, Kovács László professzor úrnak a sok szakmai segítséget és támogatást.

Köszönöm Kovács Edith Alice docens asszonynak, hogy Vajnai Tibor halála után elvállalta a témavezetésemet, és bátorításával, lelkesítésével új lendületet adott a dolgozat elkészítéséhez és a szakmai segítségével új területeket ismerhettem meg.

Külön köszönöm Fábián Csaba professzor úrnak, hogy részt vehettem az általa vezetett kutatócsoport munkájában és a sztochasztikus optimalizálási fejezet elkészítésében szakmai tanácsaival segítette munkámat. Köszönöm türelmét és optimizmusát, amivel átlendített a kritikus időszakokon.

Köszönöm mindazon kollégáimnak és barátaimnak, akik támogatásukkal, lelkesítésükkel vagy segítségükkel hozzájárultak az értekezés elkészítéséhez.

Hálás vagyok családomnak, különösen gyerekeimnek, akik motiváltak és szeretettel, türelemmel mellettem álltak a kutatómunkám és az értekezés elkészítése során.

Tartalomjegyzék

Ábrák jegyzéke

Táblázatok jegyzéke

1. fejezet

Bevezetés

Egy vagy több esemény bekövetkezésének környezetéről rendelkezésünkre álló információk gyakran hiányosak. Ez történhet például a hiányos adathalmazok miatt vagy például a jövőben történő események bizonytalansága miatt. A bizonytalanság kezelése fontos feladat, célunk olyan modellek kidolgozása, amelyek segítségével szimulálni tudjuk a lehetséges eseményeket, megbecsülhetjük különböző események valószínűségét. Ilyen módon olyan információkhoz juthatunk, amelyek elősegíthetik a "jó" döntések meghozatalát.

Dolgozatomban egyrészt bizonytalanság melletti optimalizálási feladatok megoldását vizsgáltam, különböző módszereket dolgoztam ki és teszteltem. Ezek érintik a kiinduló megengedett megoldások meghatározását és a pontosság szabályozását a valószínűségmaximalizálási feladatok megoldása során. Ezen kívül kidolgoztam többdimenziós eloszlások modellezését kopulákhoz kapcsolódóan és ezek felhasználásából származtatott szimulációkat és szcenáriók generálását. Legfrissebb kutatásaim a hiányzó adatok pótlására irányultak. Ezen a területen kifejlesztettem egy valószínűségi gépi tanulási eljárást, amely a releváns változók alapján becsüli meg a hiányzó értékeket. Egy másik problémakör, ahol szimulációs eljárást terveztem és alkalmaztam a prefix-fa alapú tárolás hatékonyságának költségelemzése volt a fa méretére vonatkozóan.

A dolgozatom a bevezető után négy fejezetet tartalmaz. Mindegyik fejezet tartalmazza a fejezet témájához szigorúan kapcsolódó fogalmakat és a kapcsolatos munkákat, továbbá kiemelésre kerülnek az eredmények és a tézisek.

Az első fejezet a sztochasztikus optimalizáláshoz kapcsolódik, nevezetesen valószínűség maximalizálással foglalkozom.

A második fejezetben az adatgenerálás és szimuláció játszik fontos szerepet. Ezek kidolgozásával portfóliók optimalizálásával, illetve portfóliók kockázatának elemzésével foglalkozom.

A harmadik részben a hiányzó adatok pótlására kidolgozott speciális valószínűségekre épülő gépi tanulási eljárást mutatom be.

A negyedik részben a prefix-fa alapú tárolás költségelemzésével foglalkozom a fa méretére vonatkozóan.

2. fejezet

Sztochasztikus optimalizálás

Az alkalmazásokban gyakran fordul elő olyan feladat, ahol úgy kell optimalizálnunk, hogy bizonyos feltételek, megkötések egy meghatározott nagy valószínűséggel teljesüljenek. Ezeket a feladatokat nevezzük valószínűségi korlátos feladatoknak. Egy korai klasszikus alkalmazás szavasmarha takarmányozásra Van de Panne és Popp [117] munkája. A vízügyi feladatokban is gyakran alkalmazzák, például víztározók méretezésére és működtetésére úttörő projektekről számol be Prékopa és Szántai a [82, 84] cikkekben. További vízügyi alkalmazások Morgan és munkatársai [68] és Van Ackooij és munkatársai [116] publikációiban található. Vegyipari alkalmazásról számol be Henrion és Moller a [42] cikkben. Egy korai klasszikus energetikai alkalmazás a STABIL modell, amelyet Prékopa és munkatársai dolgoztak ki [81]. Egy újabb energetikai alkalmazás Van Ackooij [111] cikkében található.

Az infokommunikációs technológiák fejlődésével új alkalmazási területek jelentek meg, például az intelligens hálózatok és a közlekedési rendszerek témakörében. Ezen területek gyors fejlődése szükségessé teszi olyan módszerek kidolgozását, amelyekkel valószínűségi korlátokat tudunk kezelni.

A valószínűségi korlátok elméletének jó áttekintését és az algoritmikus eljárások bemutatását Prékopa [78, 79] és Dentcheva [20] publikációiban találjuk. A valószínűségi korlátokkal (részben) foglalkozó további monográfiák Kall és Mayer [48], Mayer [64], Dentcheva [19] és Mayer [63], ahol az utóbbi inkább az algoritmusokra koncentrál.

A következőkben bemutatásra kerülő véletlenített eljárás során alkalmazott módszerek az [S4, S5, S8, S11] cikkekben jelentek meg és ott részletesen kifejtésre kerültek.

Először bemutatom, milyen típusú problémákkal foglalkoztunk, majd ezen valószínűséggel megfogalmazott modellek és megoldó eljárások rövid története és a p-efficiens pont megközelítések kerülnek bemutatásra. Ezután rátérek Fábián Csaba vezette kutatócsoport által kidolgozott és vizsgált valószínűség maximalizálási feladat véletlenített eljárással történő megoldási módszerének ismertetésére. Ezen belül részletesen bemutatom a saját eredményeimet. Egy szemléletes bemutatással kezdem, majd részletesen kifejtésre kerülnek az elvégzett kísérletek és ezek elméleti alátámasztása. A dolgozat ezen része a módszer fejlesztésének különböző és egyre hatékonyabb állomásait mutatja be. Végül a pontosságszabályozásra végzett kísérleteimet ismertetem.

Tekintsük a következő típusú valószínűséggel korlátozott feladatot:

$$\mathrm{P}(g(x,\boldsymbol{\xi}) \leq 0) \geq p, \tag{2.1}$$

ahol $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ egy leképezés, $\boldsymbol{\xi} \in \mathbb{R}^m$ egy véletlen vektor a hozzá tartozó P valószínűségi mértékkel és $p \in [0,1]$ a felhasználó által meghatározott valószínűség, aminél nagyobb vagy egyenlő valószínűséggel teljesülnek az egyenlőtlenségek. Ha $k > 1$, akkor gyakran használjuk az *együttes valószínűségi korlát* terminológiát is, mivel azt szeretnénk, hogy a $g(x,\boldsymbol{\xi}) \leq 0$ véletlen egyenlőtlenségrendszer elég nagy valószínűséggel teljesüljön.

Két általános optimalizálási problémával fogunk a továbbiakban foglalkozni a (2.1) egyenlőtlenséghez kapcsolódóan, nevezetesen a valószínűségi függvény maximalizálással és egy klasszikus optimalizálási problémával a (2.1) korlátozás alapján. Ezek a következő formában írhatók fel:

$$\begin{gathered} \max\ \mathrm{P}(g(x,\boldsymbol{\xi}) \leq 0) \\ x \in X \end{gathered} \tag{2.2}$$

és

$$\begin{gathered} \min\ c^T x \\ \mathrm{P}(g(x,\boldsymbol{\xi}) \leq 0) \geq p \\ x \in X \end{gathered} \tag{2.3}$$

ahol X egy konvex kompakt halmaz.

Számos alkalmazásban $X = \{x \in \mathbb{R}^n\ : Ax \leq b\}$ poliéder.

Feltételezzük, hogy $\boldsymbol{\xi}$ folytonos eloszlású és sűrűségfüggvénye logkonkáv, valamint hogy g rendelkezik általánosított konkávitási tulajdonságokkal.

Ezen feltételek mellett:

$$x \mapsto \phi(x) := \mathrm{P}(g(x,\boldsymbol{\xi}) \leq 0) \tag{2.4}$$

leképezés is rendelkezik általános konkávitási tulajdonságokkal (lásd. Prékopa [78] könyvének 10.2 fejezete). Ezen konkávitási tulajdonságok mellett a (2.2) és a (2.3) problémák konvex optimalizálási problémák lesznek.

Jelen dolgozatban azzal a speciális esettel foglalkozunk, amikor

$$g(x,\boldsymbol{\xi}) = \boldsymbol{\xi} - Tx, \tag{2.5}$$

ahol T mátrix. Ekkor a (2.2) és (2.3) problémák a következők lesznek:

$$\begin{gathered} \max\ \mathrm{P}(Tx \geq \boldsymbol{\xi}) \\ Ax \leq b \end{gathered} \tag{2.6}$$

és a valószínűséggel korlátozott probléma

$$\min\ c^T x$$

$$\mathrm{P}(Tx \geq \boldsymbol{\xi}) \geq p \tag{2.7}$$

$$Ax \leq b$$

ahol a döntési vektor: x. Adottak az A, T mátrixok és a b, c megfelelő méretű vektorok. A két sztochasztikus programozási feladat különböző módon tartalmazza a véletlenszerűséget. Az első típusú feladatnál a cél a valószínűség maximalizálása, míg a második típusú feladatnál megkötjük, hogy a feltételek adott, egyhez közeli p valószínűséggel teljesüljenek.

Mindkét esetben a $\boldsymbol{\xi}$ véletlenvektor eloszlása ismert és az alábbi feltételeket teljesíti:

1. a megengedett tartományok nem üresek és korlátosak,
2. a $\boldsymbol{\xi}$ folytonos, létezik sűrűségfüggvénye és az logkonkáv.

A második feltételből, felhasználva Prékopa András eredményeit, következik, hogy *a kumulatív eloszlásfüggvény* $F(\boldsymbol{z}) = P(\boldsymbol{z} \geq \boldsymbol{\xi})$ *logkonkáv.* Ez egy fontos tulajdonság, amit fel fogunk használni.

2.1. A valószínűséggel megfogalmazott modellek és megoldó eljárások rövid története

A valószínűséggel megfogalmazott modell Prékopa András korai eredménye a [73] cikkben, ahol olyan programozási feladatokkal foglalkozik, amelyekben bizonyos paraméterek véletlenek, azonban ismerjük a várhatóértéküket, szórásukat és kovarianciájukat. Ezek alapján az optimális célfüggvényérték eloszlását jellemzi, lehetővé téve konfidenciaintervallum szerkesztését.

A valószínűségi korlátok melletti programozást, mint bizonytalanság melletti döntési modellt Charnes, Cooper és Symonds a [13] cikkben vezette be. Ebben a cikkben a szerzők erre a modellre és annak változataira, valamint a [12] cikkben bemutatott kiterjesztésekre a *valószínűséggel korlátozott programozás* (chance constrained programming) kifejezést használják.

Ezek a korai valószínűséggel korlátozott modellek azonban *egyedi valószínűségi korlátokon* (individual chance constraints) alapultak, azaz a (2.3) probléma típusú megszorítás helyett a következő típusú megszorításokat használták:

$$P(g_i(x, \boldsymbol{\xi}) \leq 0) \geq p_i, \quad i = 1, \ldots, k. \tag{2.8}$$

Egy $\boldsymbol{\xi}$ véletlen jobboldali vektor, amely sztochasztikusan független komponensekkel rendelkezik, valószínűségi korlátok mellett való programozásával (ahogyan az a (2.7) feladatban áll), először Miller és Wagner [65] foglalkozott.

Az általánosabb (2.3) problémát, ahol a $\boldsymbol{\xi}$ sztochasztikusan függő komponensekből áll, Prékopa András [74, 76] vezette be, majd ő és követői tovább vizsgálták.

A valószínűségi korlátok numerikus kezelésében jelentős lépés volt, amikor a logkonkáv mértékek elméletén alapuló konvexitási állításokat Prékopa [75, 77] kidolgozta.

Henrion és munkatársai újabb kutatási iránya (eventual convexity) szerint több olyan eloszlás is van, amelyre elég nagy valószínűségek esetén az eloszlásfüggvény gráfja konkáv. A valószínűségi korlátok konvexitására vonatkozó legújabb eredmények megtalálhatók a [43, 44, 112] cikkekben.

Prékopa és társai a [81] cikkben kidolgoztak egy (2.7) alakú modellt, a magyar villamosenergia-szektor tervezési problémájára (STABIL). Az így kapott sztochasztikus programozási problémát Zoutendijk [126] megengedett irányok módszerével oldották meg. Meg kell azonban jegyezni, hogy Zoutendijk módszere nem konvergál minden körülmények között, amint azt Topkis és Veinott [105] cikke mutatja.

A valószínűségi korlátos problémára kidolgoztak vágósíkos módszereket is közelítve az

$$M(p) := \{x \in \mathbb{R}^n \ : \mathrm{P}(g(x, \boldsymbol{\xi}) \leq 0) \geq p\} \tag{2.9}$$

szinthalmazt. Prékopa és Szántai módszere [82] Slater-pontot alkalmaz annak meghatározásához, hogy hol kell elvégezni a következő vágást. (Nevezetesen az $M(p)$ határának és a Slater-pontot az aktuális próba ponttal összekötő intervallum metszéspontját keresi meg.) A módszer a Veinott-féle [119] cikkhez kapcsolódik.

A STABIL problémához épített megoldójában Szántai [104] cikkében óvatos intervallumfelező algoritmust fejlesztett ki az $M(p)$ határán lévő metszéspont biztonságos kiszámítására, amikor a valószínűségi korlátokat meghatározó valószínűségi értékek nem számíthatók ki tetszőleges nagy pontossággal.

Szántai egy eljárást dolgozott ki úgy, hogy a Slater-pontot mozgatja a megoldási folyamat során, ami gyorsabb konvergenciát eredményez. Erről később kiderült, hogy Veinottnak egy korábban leírt technikájával megegyezik.

Mayer [63] cikkében központosított vágósíkos módszert javasolt, Elzinga és Moore [28] adaptációját. A vágósíkos módszerek kevesebb iterációval konvergálnak, mint a megengedett irányok módszerei, mivel a korábbi gradiens információ megmarad. Ezek a módszerek nyilvánvalóan megkövetelik, hogy a

$$\phi(x) := \mathrm{P}(g(x, \boldsymbol{\xi}) \leq 0) \tag{2.10}$$

gradiensét hatékonyan ki tudjuk számítani.

A ϕ differenciálhatósági feltételeinek azonosítása két fő kutatási irány kidolgozásához vezetett. Az első irány nem használja ki a $\boldsymbol{\xi}$-ről vagy annak mögöttes eloszlásáról szóló specifikus ismereteket, hanem csak a sűrűségfüggvényének differenciálhatósági tulajdonságait és a g differenciálhatóságát. A kutatási irányról bővebb információk Uryasev [107, 108] cikkeiben és az azokban lévő hivatkozásokban találhatók. A második kutatási irány a $\boldsymbol{\xi}$ mögöttes (underlying) eloszlásának specifikus ismereteit aknázza ki, és kapcsolatot teremt a ϕ gradiensének bármely komponense és a ϕ-hez hasonló mennyiség kiértékelése között. Ezt az irányt Prékopa és Szántai a [74, 83, 103] cikkekben, Van Ackooij és Henrion a [115] cikkben tárták fel. Ha olyan kifinomult szoftverekkel kombináljuk, mint például a Genz-kód [38, 37] többváltozós normális eloszlásokra, akkor a nagy dimenziós problémák jelentős hatékonysággal megoldhatók (például egy $k = 168$ esetet vizsgált Van Ackooij és Oliveira a [114] cikkben).

A támaszsík módszernél a ϕ kiértékelésének pontatlanságát figyelembe kell venni az $M(p)$ határán lévő metszéspont kiszámításakor.

A továbbiakban a p-efficiens pont megközelítésekkel fogunk foglalkozni.

2.2. A p-efficiens pont megközelítésekről

Legyen a g leképezés

$$g(x, \boldsymbol{\xi}) := \boldsymbol{\xi} - h(x) \tag{2.11}$$

alakú valamely $h(x)$ vektorértékű függvénnyel. Akkor a $\mathrm{P}(g(x, \boldsymbol{\xi}) \leq 0)$ valószínűségre előírt korlátot szétválaszthatónak mondjuk és

$$\phi(x) = \mathrm{P}(g(x, \boldsymbol{\xi}) \leq 0) = F_{\boldsymbol{\xi}}(h(x)) \tag{2.12}$$

tulajdonságai közvetlenül kapcsolódnak az $F_{\boldsymbol{\xi}}$ többváltozós eloszlásfüggvényéhez.

Ebben a környezetben Prékopa új megoldó eljárást dolgozott ki ezekre a feladatokra, amely a p-efficiens (pLEP) pontokon alapszik. A p-efficiens fogalmát a [86] cikkben vezette be és elméletüket tárgyalta. Az alábbiakban a fogalmat és a hozzá kapcsolódó elméletet tekintjük át és ezen a területen elért további eredményeket.

megközelítést dolgozott ki az úgynevezett p-efficiens pontok fogalmának bevezetésével [86], amely fogalmat az alábbiakban definiálunk:

Egy z pont akkor és csak akkor p-efficiens, ha $F_{\boldsymbol{\xi}}(z) \geq p$ és nem létezik olyan z', hogy $z' \leq z$, $z' \neq z$, $F_{\boldsymbol{\xi}}(z') \geq p$. (Folytonos esetben ezek a pontok a szintvonalon vannak.)

Prékopa, Vizvári és Badics [85] ezt a koncepciót alkalmazza a (2.7) típusú feladatok megoldásában, ahol a véletlen paraméterek diszkrét véges eloszlásúak. Először felsorolják az összes p-efficiens pontot és ezek alapján a probléma konvex enyhítését javasolják. Az enyhített valószínűségi korlát egy z pont létezését írja elő a p-efficiens pontok konvex burkában úgy, hogy $h(x) \geq z$ teljesül. Az enyhített problémát ezután lineáris programozási problémaként írják fel és oldják meg.

Dentcheva, Prékopa és Ruszczyński [23] olyan (2.7) típusú problémákat vizsgál, ahol a véletlen paraméterek egészértékűek. Bebizonyítják, hogy a valószínűségi megszorítás konvex, feltéve, hogy a véletlen paraméterek bizonyos általánosított konvexitási tulajdonsággal bírnak. A valószínűségi megszorítást a következő formában fogalmazzák meg:

$$h(x) \geq z, \tag{2.13}$$

ahol z az $M(p)$ szinthalmazhoz tartozik.

A szerzők megkonstruálnak egy Lagrange-duált a $h(x) \geq z$ kényszer lazításával, és megfigyelik, hogy a duális függvény két függvény összegére bomlik. Az összeadandó függvények két egyszerűbb probléma megfelelő optimális célérték függvényei:

- az első segédprobléma egy lineáris programozási probléma;
- a második segédprobléma pedig egy lineáris függvény minimalizálása az $M(p)$ szinthalmazon.

A duál probléma megoldása után megalkotható a primál optimális megoldás. Megjegyezzük, hogy gyakran előfordulhatnak technikai problémák, amelyeket le kell küzdeni. A szerzők egy új speciális módszert is kidolgoznak, amely elválasztja a p-efficiens pontok

generálását és a közelítő probléma megoldását az ismert p-efficiens pontok alapján. Az új módszer az úgynevezett *kúpgenerálás*, az oszlopgenerálás régi koncepcióját alkalmazza. Az egészprogramozással való lényeges kapcsolatot Vizvári a [121] cikkben mutatja meg, kimutatva, hogy a Lagrange-szorzók egyenlőtlenségekkel korlátozott optimalizálási feladatokban való alkalmazása Everett [30] dolgozatával kezdődött, valamint kapcsolatba hozva a Brooks és Geoffrion [10] cikkével.

Dentcheva, Lai és Ruszczyński [21] kiterjeszti ezeket az eredményeket általános konvex problémákra. A Lagrange-duált $h(x) \geq z$ korlát lazításával állítják elő. A duális függvény két függvény összegére bomlik, mint az előző [23] cikkben tárgyalt speciális esetben. Az első segédprobléma azonban itt egy jól strukturált konvex programozási probléma, a [23] lineáris programozási problémája helyett. A nehéz rész továbbra is a második segédprobléma, amely az $M(p)$ feletti lineáris függvény minimalizálása. A szerzők duális típusú módszert dolgoznak ki, és javaslatot tesznek a primál megoldás visszanyerésének módjára. Sőt kiterjesztik a kúpgenerálási módszert egy általános primál-duál módszerre.

Prékopa [80] cikkében a (2.7) feladattal egyenértékű problémát vesz figyelembe, ahol a véletlen vektornak folytonos logkonkáv eloszlása van. Prékopa, Vizvári és Badics [85] módszerét ötvözi Szántai [104] támaszsík módszerével. Az így kapott hibrid módszer egyidejűleg alkotja meg az $M(p)$ szinthalmaz belső és a külső közelítését. A támaszsík módszer a megoldási folyamat során p-efficiens pontok generálására szolgál.

Dentcheva és Martinez [22] cikkben kidolgozta a fent tárgyalt [21] duális módszer regularizált változatát.

Van Ackooij, Berge, de Oliveira és Sagastizábal [113] olyan megoldási keretrendszert dolgozott ki, amely magában foglalja és kiterjeszti a különféle meglévő formulákat.

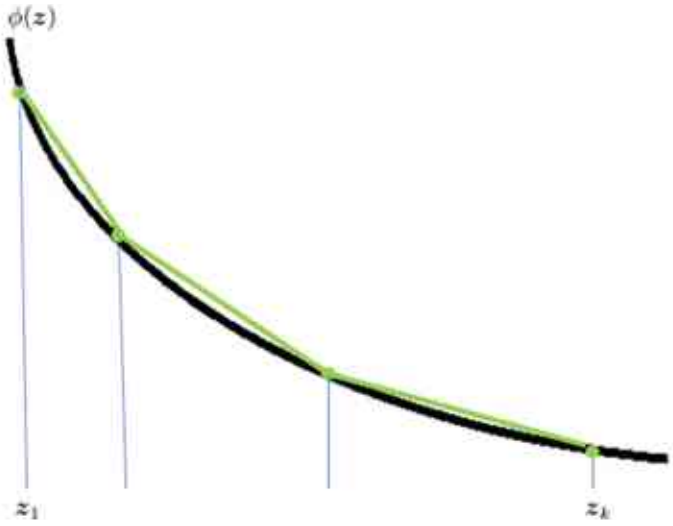
2.3. Az általunk kidolgozott epigráf-közelítő módszer bemutatása és szemléltetése

A valószínűség maximalizálási feladatot két részfeladatra bontjuk. A mester (master) egy lineáris programozási feladat, a javító oszlopok keresése pedig korlátozás nélküli konvex minimalizálási feladat. Az utóbbi feladatot megoldó eljárást kútfőnek (oracle) nevezzük. Az eljárás iteratív, mindig olyan újabb próbapontot keresünk, amelynek a felvételével leginkább javul a célfüggvény értéke. Az alkalmazott eljárás a Prékopa-féle duális megközelítés [86] egy módosított változata, amelyben a függvény epigráfját közelítjük, nem pedig a szinthalmazt. A függvény, amelyet közelíteni akarunk, folytonosan differenciálható.

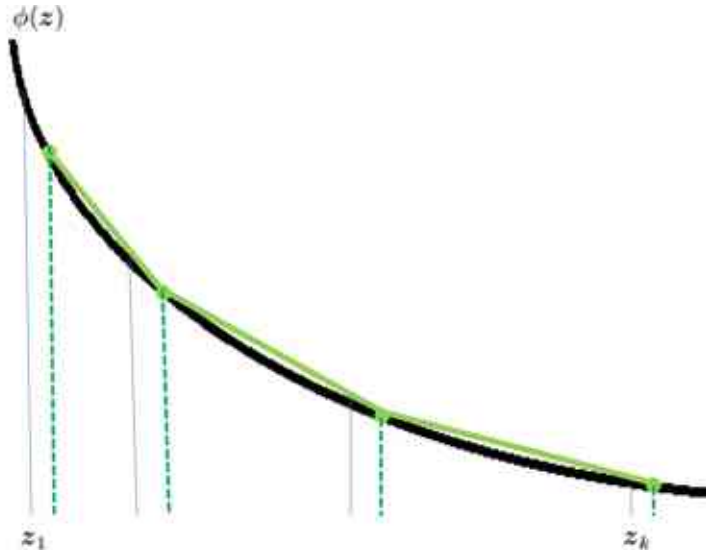
A [S5] cikkben belső közelítést alkalmaztunk a valószínűségi függvényre, amit a 2.1. ábra szemléltet. Ez a módszer pontos, de nagyon munkaigényes. Először kicsi feladatokon pontosan számoltunk, ekkor a futási idő nagyon hosszú volt. A módszer részletes bemutatása a 2.4. fejezetben található.

Később úgy javítottunk az eljáráson, hogy csak közelítőleg számítottuk ki az új próbapontokat, viszont ezek függvényértékét pontosan számítottuk ki (2.2. ábra). Ekkor már nagyobb feladatokat tudtunk megoldani, de még mindig hosszú volt a futási idő. A módszer részletes bemutatása a 2.5. fejezetben található.

Az [S5] cikkben alkalmazott belső közelítés egy véletlenített változatát dolgoztuk ki és teszteltük az [S4] cikkben. A módszer azért jó, mert ha pontatlanul számoljuk ki a



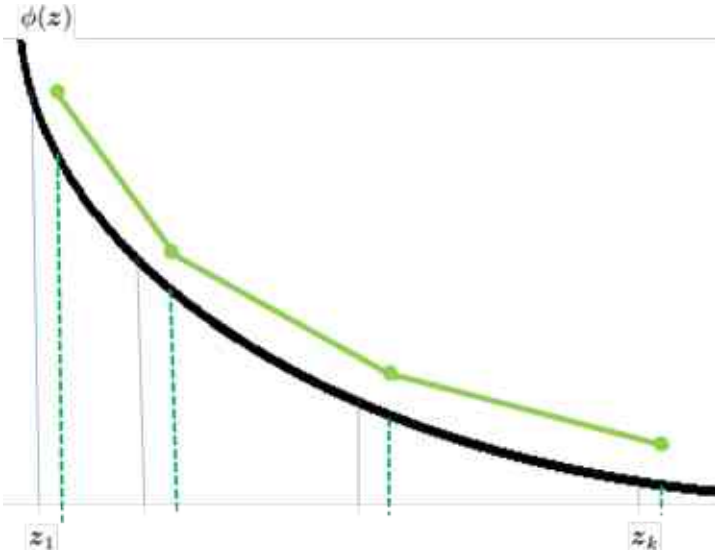
2.1. ábra. A valószínűségi függvény belső közelítése



2.2. ábra. A valószínűségi függvény belső közelítése egy vagy két lépés elvégzésével, majd pontosan számoljuk ki a függvényértéket

gradienseket, akkor is felső becslést adunk a függvényre. A próbapontok nem pontosan vannak megállapítva, de ezekben az elcsúsztatott pontokban a függvényértékeket elég pontosan számítjuk ki (2.3. ábra). A függvényt olyan pontossággal értékeljük ki, hogy a közelítő függvény nagy valószínűséggel fölötte legyen az eredeti függvénynek. A módszer részletes bemutatása a 2.6. fejezetben található.

A továbbiakban ezeket a fent megfogalmazott megoldási módokat és az elvégzett kísérleteket mutatjuk be.



2.3. ábra. A valószínűségi függvény véletlenített módszerrel kapott közelítése

2.4. Oszlopgenerálási eljárás

Először a legegyszerűbb esettel foglalkozunk, amely 2.1. ábrához kapcsolódik. Elsőként a poliéder feletti minimalizálást vesszük figyelembe, mely a (2.6) feladat átfogalmazása:

$$\begin{aligned} &\min\ \phi(T\boldsymbol{x}) \\ &A\boldsymbol{x} \leq \boldsymbol{b}, \end{aligned} \tag{2.14}$$

ahol $\phi : \mathbb{R}^n \to \mathbb{R}$ egy konvex függvény, amelynek gradiens számítása időigényes. Az $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{b} \in \mathbb{R}^r$ vektorok és a T és A mátrixok $n \times m$ és $r \times m$ méretűek.

Ebben a részben idealizált környezetben dolgozunk, a következő feltételezés szerint:

1. Feltevés. *Adott $\boldsymbol{z} \in \mathbb{R}^n$, a $\phi(\boldsymbol{z})$ függvényérték és a $\nabla\phi(\boldsymbol{z})$ gradiensvektor pontosan kiszámítható.*

Először megfogalmazzuk a duális problémát, és megszerkesztjük a primál és a duál problémák poliéderes modelljét.

Megjegyezzük, hogy az [S5]-ben megfogalmazott konstrukciót követjük, bár ott kihasználtuk a valószínűségi cél monotonitását és a változók felosztása a $\boldsymbol{z} \leq T\boldsymbol{x}$ alapján történt.

A jelenlegi idealizált környezetben a változó hasítás (variable split) hagyományos formáját alkalmazzuk. A (2.14) probléma így írható fel:

$$\begin{aligned} &\min\ \phi(\boldsymbol{z}) \\ &A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0} \\ &\boldsymbol{z} - T\boldsymbol{x} = \boldsymbol{0}. \end{aligned} \tag{2.15}$$

A (2.15) problémának van optimális megoldása, mivel a (2.14) megengedett tartománya korlátos és nem üres.

A $-\boldsymbol{y} \in \mathbb{R}^r, -\boldsymbol{y} \geq \boldsymbol{0}$ szorzóvektort bevezetve az $A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0}$ feltételhez és a $-\boldsymbol{u} \in \mathbb{R}^n$ szorzóvektort bevezetve a $\boldsymbol{z} - T\boldsymbol{x} = \boldsymbol{0}$ feltételhez, a (2.15) Lagrange-duálisa így írható fel:

$$\begin{gathered} \max \{\boldsymbol{y}^T\boldsymbol{b} - \phi^\star(\boldsymbol{u})\} \\ (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D}, \end{gathered} \tag{2.16}$$

ahol $\phi^\star(\boldsymbol{u})$ a $\phi(\boldsymbol{z})$ függvény konvex konjugáltja és

$$\mathcal{D} := \left\{ (\boldsymbol{y}, \boldsymbol{u}) \in \mathbb{R}^{r+n} \mid \boldsymbol{y} \leq \boldsymbol{0},\ T^T\boldsymbol{u} = A^T\boldsymbol{y} \right\}.$$

A konvex dualitás elmélet szerint ennek a problémának van optimális megoldása.

Poliedrikus modell

Tegyük fel, hogy kiértékeltük a $\phi(\boldsymbol{z})$ függvényt a $\boldsymbol{z}_i$ pontokban $(i = 0, 1, \ldots, k)$; bevezetjük a $\phi_i = \phi(\boldsymbol{z}_i)$ jelölést a megfelelő függvényértékekre. A $\phi(.)$ belső közelítése a

$$\phi_k(\boldsymbol{z}) = \min_{\lambda_0,\ldots,\lambda_k} \sum_{i=0}^{k} \lambda_i \phi_i, \tag{2.17}$$

ahol

$$\lambda_i \geq 0 \ (i = 0, \ldots, k)$$

$$\sum_{i=0}^{k} \lambda_i = 1$$

$$\sum_{i=0}^{k} \lambda_i \boldsymbol{z}_i = \boldsymbol{z}.$$

Ha $\boldsymbol{z}$ pont nincs benne a $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k$ pontok konvex burkában, akkor legyen $\phi_k(\boldsymbol{z}) := +\infty$.

A (2.15) probléma poliéderes modellje az alábbi:

$$\begin{gathered} \min\ \phi_k(\boldsymbol{z}) \\ A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0} \\ \boldsymbol{z} - T\boldsymbol{x} = \boldsymbol{0}. \end{gathered} \tag{2.18}$$

Feltételezzük, hogy a (2.18) megoldható, azaz optimuma véges. Ez a kezdeti $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k$ pontok megfelelő kiválasztásával biztosítható. A $\phi_k(\boldsymbol{z})$ konvex konjugáltja

$$\phi_k^\star(\boldsymbol{u}) = \max_{0 \leq i \leq k} \{\boldsymbol{u}^T\boldsymbol{z}_i - \phi_i\}. \tag{2.19}$$

Mivel a $\phi_k^\star(.)$ a $\phi^\star(.)$ vágósíkos modellje, a (2.16) probléma poliéderes modellje a következő probléma:

$$\begin{gathered} \max \{\boldsymbol{y}^T\boldsymbol{b} - \phi_k^\star(\boldsymbol{u})\} \\ (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D}. \end{gathered} \tag{2.20}$$

Lineáris programozási formulák

A (2.17)-(2.18) primál modell probléma a következőképpen lesz megfogalmazva

$$\begin{aligned} &\min \sum_{i=0}^{k} \phi_i \lambda_i \\ &\lambda_i \geq 0 \quad (i = 0, \ldots, k) \\ &\sum_{i=0}^{k} \lambda_i = 1 \\ &\sum_{i=0}^{k} \lambda_i \boldsymbol{z}_i - T\boldsymbol{x} = \boldsymbol{0} \\ &A\boldsymbol{x} \leq \boldsymbol{b}. \end{aligned} \tag{2.21}$$

A (2.19)-(2.20) duál modellprobléma, lineáris programozási feladatként megfogalmazva, éppen a (2.21) LP duálja:

$$\begin{aligned} &\max \vartheta + \boldsymbol{b}^T \boldsymbol{y} \\ &\boldsymbol{y} \leq \boldsymbol{0} \\ &\vartheta + \ \boldsymbol{z}_i^T \boldsymbol{u} \leq \ \phi_i \quad (i = 0, \ldots, k) \\ &-T^T \boldsymbol{u} + A^T \boldsymbol{y} = \boldsymbol{0}. \end{aligned} \tag{2.22}$$

Jelölje ($\overline{\lambda}_0, \ldots, \overline{\lambda}_k$, $\overline{\boldsymbol{x}}$) és ($\overline{\vartheta}$, $\overline{\boldsymbol{u}}$, $\overline{\boldsymbol{y}}$) a (2.21) és (2.22) problémák megfelelő optimális megoldásait – mindkettő a (2.18) és így a (2.21) megvalósíthatóságára vonatkozó feltételezésünk miatt létezik. Legyen továbbá

$$\overline{\boldsymbol{z}} = \sum_{i=0}^{k} \overline{\lambda}_i \boldsymbol{z}_i. \tag{2.23}$$

2. Tétel.

(a) $\phi_k(\overline{\boldsymbol{z}}) \ = \ \sum_{i=0}^{k} \phi_i \overline{\lambda}_i \ = \ \overline{\vartheta} + \overline{\boldsymbol{u}}^T \overline{\boldsymbol{z}}$,

(b) $\overline{\vartheta} = -\phi_k^\star(\overline{\boldsymbol{u}})$,

(c) $\phi_k(\overline{\boldsymbol{z}}) + \phi_k^\star(\overline{\boldsymbol{u}}) = \overline{\boldsymbol{u}}^T \overline{\boldsymbol{z}}$ *és így* $\overline{\boldsymbol{u}}$ *szubgradiense a* $\phi_k(.)$ *függvénynek a* $\overline{\boldsymbol{z}}$ *pontban, azaz minden* $\boldsymbol{z} \in \mathbb{R}^n$ *vektorra teljesül:* $\phi_k(\boldsymbol{z}) \geq \phi_k(\overline{\boldsymbol{z}}) + \overline{\boldsymbol{u}}^T (\boldsymbol{z} - \overline{\boldsymbol{z}})$.

A fenti tételt az [S4] cikkben bizonyítottuk.

Az eljárás motivációja

A szimplex módszerrel összefüggésben a Markowitz-szabály az oszlopkiválasztás jól ismert és gyakran használt szabálya. A Markowitz-szabály a legnagyobb redukált költségű vektort választja ki.

Az aktuális modellprobléma optimális duális megoldása (azaz árnyékár-vektora) a $(\overline{\vartheta}, \overline{\boldsymbol{u}}, \overline{\boldsymbol{y}})$ vektor. Adott $\boldsymbol{z} \in \mathbb{R}^n$ vektor esetén a $\boldsymbol{z}_{k+1} = \boldsymbol{z}$ új oszlopot adjuk hozzá a (2.21) feladathoz. Ez javító oszlop, ha a lineáris programozási feladathoz új oszlopként hozzávéve a célfüggvényérték javul, azaz az új oszlop redukált költsége pozitív.

Egy $\boldsymbol{z}$ vektorból szerkesztett új oszlop redukált költsége definíció szerint az alábbi formulából számolható

$$\overline{\rho}(\boldsymbol{z}) \;:=\; \overline{\vartheta} \;+\; \overline{\boldsymbol{u}}^T\boldsymbol{z} - \phi(\boldsymbol{z}). \tag{2.24}$$

Könnyen belátható, hogy a $\overline{\boldsymbol{z}}$ redukált költsége nemnegatív.

Valóban,

$$\overline{\rho}(\overline{\boldsymbol{z}}) \;\geq\; \overline{\vartheta} \;+\; \overline{\boldsymbol{u}}^T\overline{\boldsymbol{z}} - \phi_k(\overline{\boldsymbol{z}}) \;= 0 \tag{2.25}$$

következik a 2. tétel *(a)* pontjából a $\phi_k(.) \geq \phi(.)$ egyenlőtlenséget alkalmazva. Könnyű belátni, hogy ebben az esetben a redukált költséget (2.24) szerint kell számolni.

2.4.1. A megoldó algoritmusa és egy működőképes változat megtervezése

Az alábbiakban ismertetésre kerülő megoldó mesterfeladat részében egy megengedett kezdőmegoldás keresésére, valamint egy olyan mesterfeladat megkonstruálására dolgoztam ki eljárást, amely a megoldó eljárást a megfelelő irányba tereli. Valamint a mesterfeladatot és a kísérletekhez a futtató keretrendszert implementáltam.

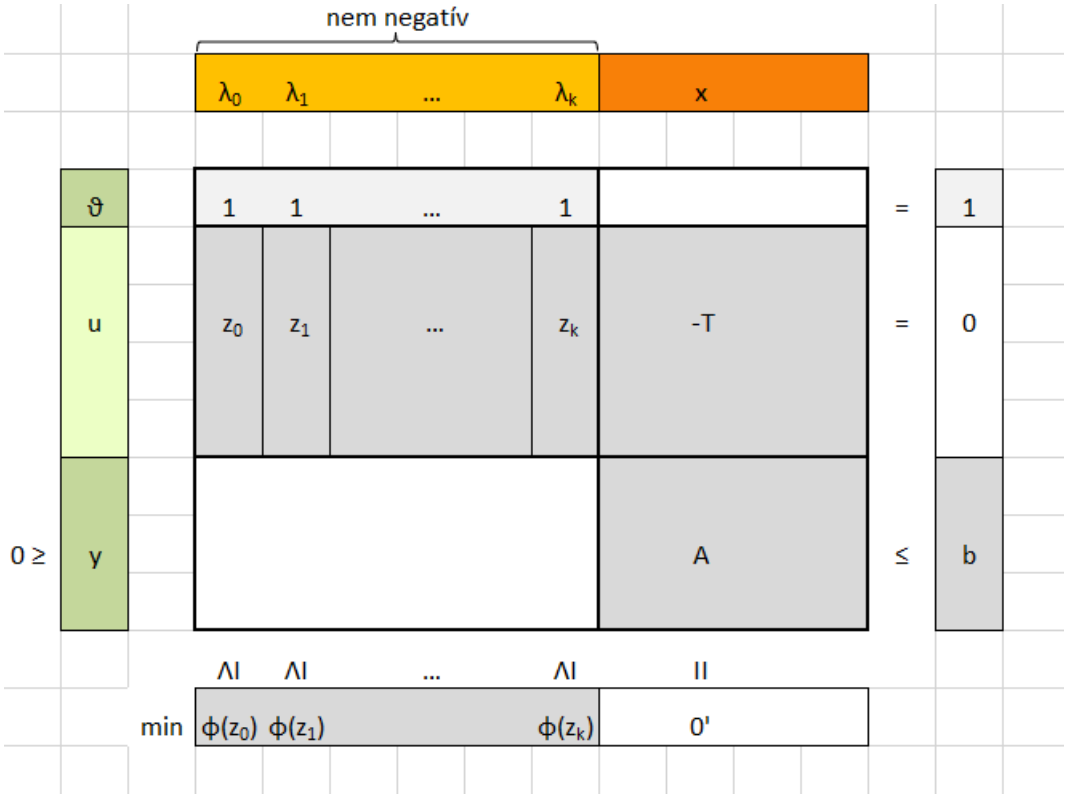
A megoldó két részből áll, az egyik a mester (master) feladatot, a másik a valószínűségi függvény adatainak meghatározására szolgáló kútfő (oracle) feladatot oldja meg. A mesterfeladat a (2.21) lineáris programozási feladat. Ennek rajzát a 2.4. ábrán láthatjuk. Az iterációk során a z_k oszlopokkal bővítjük a feladatot, a kútfőnek a zöld árnyékár-vektor kerül átadásra. A célfüggvény sora feletti relációs jelek az optimalitás feltételei. A rajzról leolvasható, hogy melyik feltételhez melyik duális változó tartozik. A 2.5. ábrán a (2.22) duális feladat rajza látható. Ha a folyamatot a primál szempontból közelítjük, akkor oszlopgenerálási eljárás, ha duál szempontból nézzük, akkor vágósíkos eljárás.

Az implementálásra Matlabot használtunk az IBM ILOG CPLEX (12.6.3. verzió) optimalizálási eszköztárral.

Megoldható mesterfeladat előállítása

A mesterfeladatban megfelelően sok oszlopvektor kell ahhoz, hogy a (2.21) feladatnak legyen megengedett megoldása. Egy megoldható mesterfeladat megszerkesztésére az alábbi eljárást dolgoztam ki.

A feladat megoldása során standard normális eloszlást feltételezek. Legyen r a véletlen vektor komponenseinek száma, ez egyenlő a T mátrix sorainak számával. Először keresek egy megfelelő $z_0 \in \mathbb{R}^r$ vektort, amelyik a primál feladat egy megengedett megoldása. Ez a következő feladat megoldásával történik:



2.4. ábra. A mesterfeladat rajza

$$\max\ t$$

$$1t - Tx \leq 0 \tag{2.26}$$

$$Ax \leq b,$$

ahol $t \in \mathbb{R}$ és $1 \in \mathbb{R}^r$ csupa egyesekből álló vektor.

Abban az esetben, ha (2.26) nem megoldható feladat, akkor az eredeti feladatnak sincs megoldása. Másrészt, ha a célérték nem korlátos, akkor az eredeti feladatban 1 valószínűség érhető el.

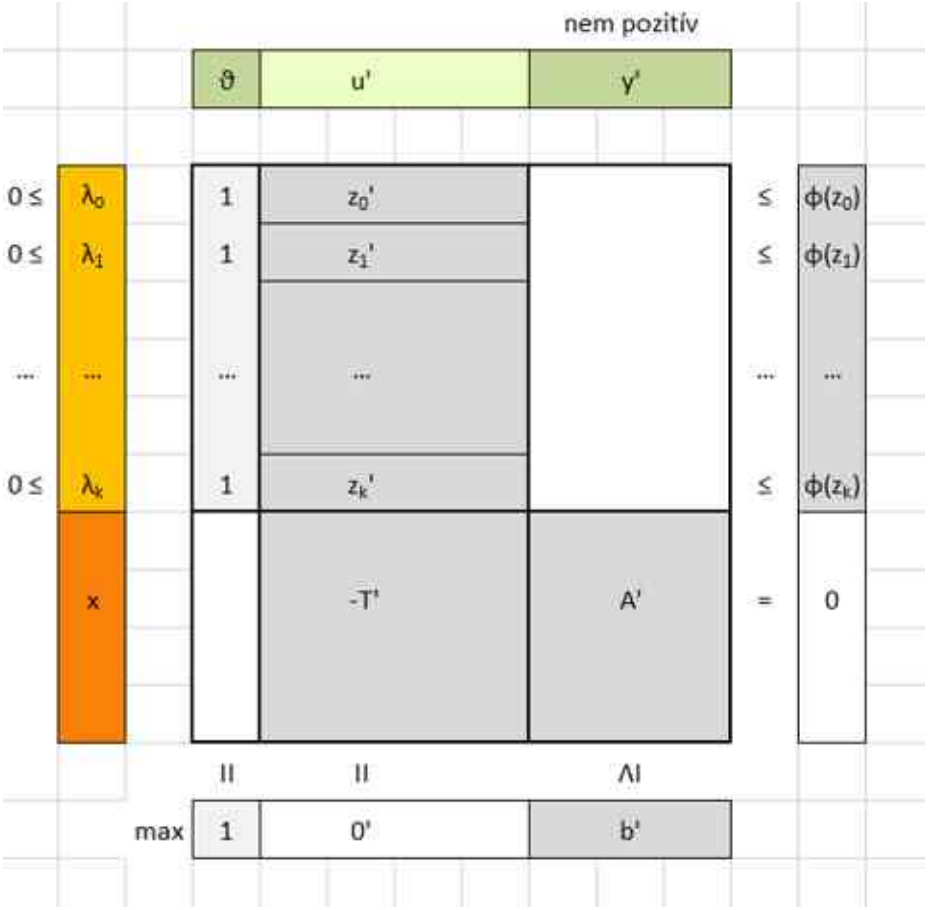
Legyen t^* és x^* a (2.26) optimális megoldása és legyen $z_0 = Tx^*$. Ezzel az egy z_0 oszloppal már megoldható lesz a mesterfeladat. A tapasztalat azt mutatja, hogy az alább leírt oszlopok a megfelelő irányba terelik a megoldó eljárást.

Legyen $Z \subset \mathbb{R}^r$ egy olyan kocka, amelyen kívüli valószínűségi súly elhanyagolható.

Mivel standard normális eloszlással dolgoztam, az r dimenziós Z kockát az origóra szimmetrikusnak tekintem. Kísérleteimben olyan kockával dolgoztam, amelyre $P(Z) \approx 0,99$.

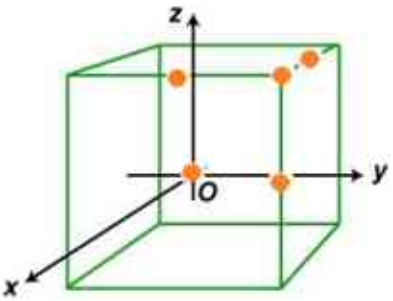
Legyen $z^{max} = (z_1^{max}, \ldots, z_r^{max})$ a Z kocka maximális csúcsa. A megoldás megkönnyítése érdekében a z_0 mellett a $z_l (l = 1, \ldots, r)$, z_{r+1}, z_{r+2} vektorok hozzáadásával inicializálom a primál feladatot, ahol

$$\begin{aligned} z_l &= \left(z_1^{max}, \ldots, z_{l-1}^{max}, 0, z_{l+1}^{max}, \ldots, z_r^{max}\right) \qquad (l = 1, \ldots, r), \\ z_{r+1} &= 0, \\ z_{r+2} &= z^{max}. \end{aligned} \tag{2.27}$$



2.5. ábra. A mesterfeladat duáljának rajza

Szemléletesen a próbapontok a következő pontokból tevődnek össze: az r darab z_l próbapont a Z kocka z^{max} csúcsából induló élek felezőpontjai lesznek, az origó és a z^{max} pont. Így összesen $r+2$ darab próbapontunk lesz.



2.6. ábra. A primál feladat inicializálásához felhasznált z_1, z_2, z_3, z_4, z_5 próbapontok

A 2.6. ábrán a 3 dimenziós esetben 3 darab z_l próbapont látható, a z_4 az origó és a z_5 pedig a z^{max} próbapont.

A mesterfeladat iteratív bővítésének algoritmusa

A megoldó első lépésként egy kezdőmegoldást keres az imént ismertetett módon, ez lesz a mesterfeladat. Ezután következik az oszlopgenerálási iterációs eljárás, amelyben mindig újabb javító oszlopot keresünk. A javító oszlopot úgy keressük, hogy a mesterfeladat megoldásának árnyékár-vektorát kapja meg a kútfő, ebben a pontban kiértékeli a függvényértéket, kiszámolja a gradienst és ennek függvényében lép. Majd a leginkább javító oszlopot adja vissza a mesterfeladatnak, ezt a megoldást javító oszlopként a feladathoz adom és folytatódik a következő iterációval a megoldás. Ha a kútfőtől visszakapott megoldás (nagy valószínűséggel) nem javít lényegesen, akkor készen vagyunk.

Legyen

$$\overline{\mathcal{R}} := \max_{\boldsymbol{z}} \overline{\rho}(\boldsymbol{z}) \tag{2.28}$$

a redukált költség optimális célértéke. A legnagyobb redukált költségű oszlop elméletileg megtalálható a $-\overline{\rho}(\boldsymbol{z})$ függvényre alkalmazott legmeredekebb ereszkedési módszerrel. Ezt keressük meg az alább ismertetett kútfő modullal.

A mesterfeladatot a CPLEX szimplex megoldójával, 10^{-4} tolerancia alkalmazásával implementáltam.

A valószínűségi függvény adatainak meghatározására szolgáló kútfő (oracle)

A 2.4. szakasznak megfelelően célunk a redukált költség (2.24) maximalizálása.

Mivel a $\overline{\vartheta}$ állandó egy adott iterációban, a kútfő közelítő megoldást keres a

$$\max_{\boldsymbol{z}} \{\overline{\boldsymbol{u}}^T \boldsymbol{z} - \phi(\boldsymbol{z})\} \tag{2.29}$$

problémára. Ez a probléma újrafogalmazható a

$$\phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}^T \boldsymbol{z} \tag{2.30}$$

függvény minimalizálásaként.

Itt legyen $\phi(\boldsymbol{z}) := -\log F(\boldsymbol{z})$, ahol $F(\boldsymbol{z})$ a többdimenziós normális eloszlásfüggvény. A $\phi(\boldsymbol{z})$ konvex függvény, az $F(\boldsymbol{z})$ logkonkáv volta miatt.

A 2.4. szakaszban tárgyalt legmeredekebb ereszkedési módszer hozzávetőleges formáját implementáltuk. Iránymenti keresést hajtunk végre ($j = 1$), és még ebben az iránymenti keresésben is megállunk egy hozzávetőleges minimummal. Ugyanis az aranymetszéseljárást alkalmazzuk, amelyik például Luenberger és Ye [60] könyvében van leírva. Csak 1 vagy 2 aranymetszés lépést végzünk.

A legmeredekebb süllyedés irányát a függvény gradiensvektorának kiszámításával találhatjuk meg:

$$\nabla \left(\phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}^T \boldsymbol{z}\right) = \nabla\phi(\boldsymbol{z}) - \overline{\boldsymbol{u}} = -\nabla \log \left(F(\boldsymbol{z})\right) - \overline{\boldsymbol{u}} = -\frac{\nabla F(\boldsymbol{z})}{F(\boldsymbol{z})} - \overline{\boldsymbol{u}}. \tag{2.31}$$

Következésképpen ki kell számítanunk az $F(\boldsymbol{z})$ többdimenziós normális eloszlásfüggvény függvényértékét és gradiensvektorát. Ehhez a számításhoz Prékopa [78] című könyvének 6.6.4. fejezetében található képleteket használjuk. Ezekkel a képletekkel egy többdimenziós valószínűségi eloszlásfüggvény gradiensének kiszámítása redukálható feltételes eloszlásfüggvény értékek kiszámítására. A normális eloszlások esetén a feltételes eloszlások is

normálisak. Véletlenszerű, kvázi véletlenszerű módszert használ a többváltozós normális valószínűség becslésére. Bemeneti paraméterek a pozitív határozott kovarianciamátrix r és az alsó és felső integrációs határ. Az algoritmus m pontot használ a p valószínűség becslésére, és kiszámít egy e hibabecslést is. A gradiensvektor megtalálása után egyetlen, pontatlan iránymenti keresést alkalmazunk.

A többváltozós normális eloszlási értékek numerikus kiszámítását a Genz [38] által megvalósított QSIMVNV Matlab függvény segítségével végeztük. A Genz-módszert a 2.6. alfejezetben fogjuk részletesebben bemutatni.

A 2.7. ábrán a mesterfeladat és a kútfő feladat közötti egy iteráció során történő adatátadás látható.

2.7. ábra. A mesterfeladat és a kútfő közötti adatáramlás

2.4.2. Kísérleti tapasztalatok

Először mesterségesen kreált kisméretű feladaton próbáltuk az eljárást nagy pontossággal megoldani. Ez nagyon hosszú futási időhöz vezetett. Ezt a megoldási módot szemlélteti a 2.1. ábra.

A (2.28) maximalizáló pont megkeresése szoros tűréssel történt, így nagyon időigényes volt.

2.5. A Markowitz-szabály közelítő formájának alkalmazása

Gyakorlati megközelítésben csak korlátozott számú iránymenti keresési lépés hajtható végre, $\overline{\boldsymbol{z}}$-tól kezdve. Ezért a 2.4. fejezet *Az eljárás motivációja* részében ismertetett Markowitz-szabály közelítő formáját alkalmaztuk.

Úgy gyorsítunk az oszlopgenerálási eljáráson, hogy csak egy vagy két iránymenti keresési lépést teszünk, nem csináljuk végig az egész konvex optimalizálási számítást (2.2. ábra).

Azt tapasztaltuk, hogy a teljes futási idő 90 %-át még mindig a Genz-kódban töltötte. Próbáltuk kiegyensúlyozni a különböző irányú erőfeszítéseket. Itt már gyorsabban (kb. 1 óra alatt) meg tudtunk oldani kicsi feladatokat.

Teszt problémák

Az epigráf-közelítő módszer kipróbálására, tesztelésére, javítására az alábbiakban bemutatott 3, 5 és 15 dimenziós feladatokkal kísérleteztünk.

Először nyolc tesztfeladatot vettünk figyelembe, amelyeket Szántai Tamás publikált a [104] cikkben. Ezek a problémák egy kávégyártó cégnél jelentkeznek. A cég három különböző kávékeveréket forgalmaz. Mindegyik keverékre szigorú követelmények vonatkoznak a savasságuk, koffeintartalmuk, lúgosságuk, keménységük és aromájuk szerint. Egy adott hónap első napján a vállalat úgy találta, hogy a rendelkezésre álló zöldkávé-kínálata 8 különböző fajtára korlátozódott. Ezek a zöldkávék ártól, elérhető mennyiségtől és a fent említett öt ízjellemzőtől függően változnak. A következő hónapban a vállalat 3 keverékére vonatkozó igények valószínűségi változók, adott várhatóértékekkel, szórással és korrelációs együtthatókkal. A vállalat azzal a problémával szembesül, hogy meg kell határoznia a rendelkezésre álló zöldkávék optimális kombinációját a következő havi pörkölési művelethez. Tehát egy sztochasztikus programozási problémát kell megfogalmazniuk, hogy az összes véletlenszerű igényt egy előírt (nagy) valószínűséggel kielégítsék, és a lehető legkisebb árat fizessék a zöldkávékért. A 0,9-es valószínűségi szint szerinti összes adat és számszerű eredmény megtalálható Szántai a [104] cikkében. Jelen dolgozatban ezeket a problémákat 'Coffee1', ..., 'Coffee8'-nak fogjuk nevezni.

Másodsorban a kávékeverési probléma kiterjesztett változatát vettük figyelembe. Ebben a kiterjesztésben a vállalat öt különböző kávékeveréket forgalmaz, így a többváltozós normális valószínűségi eloszlás ötdimenziós. Ezt a problémát jelen dolgozatban „Coffee9"-nek nevezzük.

Végül egy készpénz-illesztési problémát vizsgáltunk meg tizenöt dimenziós normális valószínűségi eloszlással. Egy vállalat nyugdíjpénztárának a következő 15 évben bizonyos kifizetéseket kell teljesítenie, amelyeket háromféle kötvény vásárlásával kell finanszírozni. A kezdőtőke adott. A figyelembe vett időhorizont végén rendelkezésre álló készpénz mennyiségét a fizetési korlátozások minden évben való teljesítésének feltétele mellett maximalizálni kell. Ez a probléma a [21] és a [41] forrásokból származik. A készpénz-illesztési tesztproblémát eredetileg költségminimalizálásként fogalmazták meg valószínűségi korlát mellett. A problémát a valószínűség maximalizálására transzformáltuk költségkorlátozás mellett. Ezt a problémát jelen dolgozatban „CashMatching"-nek nevezzük. A CashMatching feladat sajátossága az időt leíró szerkezet, melyben a véletlen paraméterek egymásra következő években felmerülő fizetési kötelezettségeket reprezentálnak. Távolodva a kiindulástól a bizonytalanság nagyobb. Bármely két véletlen paraméter korrelációja pozitív. A korrelációs mátrixban a főátlótól távolodva a komponensek csökkennek.

2.5.1. Kísérleti tapasztalatok

Minden tesztfeladatot a költségmegszorítás különböző jobb oldalával oldottunk meg. Számítási eredményeink a 2.8 - 2.10. ábrákban találhatók.

A tesztproblémákat eredetileg költségminimalizálásként fogalmazták meg valószínűségi korlát mellett, azaz (2.7) alakúak. A problémákat valószínűségmaximalizálásra konvertáltuk, azaz (2.6) alakúra. A költségmegszorítások jobb oldalát úgy állítottuk be, hogy a megfelelő optimális valószínűségi szintek azok legyenek, amelyek táblázataink „előírt

valószínűségi szint" (prescribed probability level) oszlopában szerepelnek. Ezekhez a számításokhoz Szántai [104] számítógépes kódját használtuk.

		1 GSR steps per iter			2 GSR steps per iter		
Problem	prescribed probability level	Genz	itNum	p	Genz	itNum	p
Coffee 1	0.8	103	7	0.7998	105	6	0.7994
	0.85	78	5	0.8501	90	5	0.8504
	0.9	93	6	0.9002	186	11	0.9005
	0.95	70	5	0.9499	116	7	0.9504
	0.98	80	6	0.9798	144	11	0.9803
	0.99	70	6	0.9896	102	8	0.9900
Coffee 2	0.8	132	9	0.7998	208	12	0.8000
	0.85	107	7	0.8499	158	9	0.8499
	0.9	134	9	0.9000	166	9	0.9000
	0.95	120	8	0.9500	119	7	0.9500
	0.98	93	7	0.9800	126	8	0.9800
	0.99	84	8	0.9897	69	6	0.9897
Coffee 3	0.8	167	10	0.8000	148	8	0.8000
	0.85	129	8	0.8500	198	11	0.8500
	0.9	120	8	0.9000	152	8	0.9000
	0.95	167	11	0.9500	159	9	0.9500
	0.98	105	8	0.9800	149	9	0.9800
	0.99	71	7	0.9897	57	5	0.9897
Coffee 4	0.8	158	9	0.8000	207	11	0.8000
	0.85	172	10	0.8500	174	9	0.8500
	0.9	150	9	0.9000	155	8	0.9000
	0.95	139	9	0.9500	153	8	0.9500
	0.98	117	9	0.9800	115	7	0.9800
	0.99	69	7	0.9897	55	5	0.9897

2.8. ábra. A 'Coffee1', ..., 'Coffee4' feladatok számítási eredményei

Minden feladatot a kútfő két beállításával oldottunk meg, minden sorkeresés során 1 vagy 2 aranymetszés arány (GSR) lépést végrehajtva. – A megfelelő adatok az „1 GSR lépés iterációnként" (1 GSR steps per iter) iterációnként és „2 GSR lépés iterációnként" (2 GSR steps per iter) fejléc alatt jelennek meg. Minden esetben felsoroljuk a Genz-szubrutin hívásainak számát (a „Genz" fejléc alatt), az oracle hívások számát (az „itNum" fejléc alatt) és a talált optimumot (a „p" fejléc alatt).

A számítási idő nagy részét minden esetben a Genz-szubrutinokban töltöttük. A 'Coffee' problémák esetén iterációnként 1 GSR lépés végrehajtása valamivel kevesebb hívást eredményezett a Genz-szubrutin felé, mint 2 GSR lépés. Érdekes módon a „CashMatching" probléma lényegesen gyorsabban megoldódott, ha iterációnként egyetlen GSR-lépést hajtottunk végre két lépés helyett. Mindezek az eredmények azt mutatják, hogy az oszlopgenerálási problémák közelítő megoldása elegendő.

A kútfő által visszaadott $\widehat{\boldsymbol{z}}$ vektorok mindig egy viszonylag kis dobozba estek, így a biztonságos tartományokban maradtak, ahol a megfelelő célfüggvények jól kondicionáltak.

A szimplex módszerben használatos szokásos megállási feltétellel állítottuk le az iterációkat, azaz, ha az új oszlopnak, amelyik a lehetséges oszlopok közül a legnagyobb redukált költségű, egy küszöbérték alá csökken a redukált költsége. Vagyis a legnagyobb redukált

		1 GSR steps per iter			2 GSR steps per iter		
Problem	prescribed probability level	Genz	itNum	p	Genz	itNum	p
Coffee 5	0.8	112	7	0.7999	131	7	0.8000
	0.85	114	7	0.8500	125	7	0.8500
	0.9	110	7	0.9000	135	7	0.9000
	0.95	112	7	0.9500	135	8	0.9499
	0.98	109	8	0.9800	113	7	0.9800
	0.99	71	6	0.9897	68	5	0.9897
Coffee 6	0.8	75	5	0.8000	104	6	0.7999
	0.85	77	5	0.8502	119	7	0.8497
	0.9	74	5	0.9004	109	6	0.9003
	0.95	81	6	0.9504	99	6	0.9506
	0.98	145	11	0.9806	84	5	0.9797
	0.99	82	6	0.9900	39	2	0.9894
Coffee 7	0.8	110	7	0.7999	127	7	0.7999
	0.85	97	6	0.8499	145	8	0.8500
	0.9	65	4	0.8997	110	6	0.8999
	0.95	70	4	0.9500	75	4	0.9500
	0.98	89	6	0.9799	108	6	0.9800
	0.99	53	4	0.9899	48	3	0.9900
Coffee 8	0.8	147	8	0.8001	105	5	0.8000
	0.85	71	4	0.8493	106	5	0.8501
	0.9	75	4	0.8999	106	5	0.8999
	0.95	80	4	0.9501	108	5	0.9500
	0.98	114	7	0.9801	114	5	0.9801
	0.99	48	3	0.9901	28	1	0.9894

2.9. ábra. A 'Coffee5', ..., 'Coffee8' feladatok számítási eredményei

		1 GSR steps per iter			2 GSR steps per iter		
Problem	prescribed probability level	Genz	itNum	p	Genz	itNum	p
Coffee 9	0.8	104	5	0.7997	136	6	0.7998
	0.85	98	5	0.8502	110	5	0.8502
	0.9	80	4	0.9000	109	5	0.9001
	0.95	115	6	0.9506	155	7	0.9506
	0.96	94	5	0.9604	132	6	0.9604
	0.97	115	7	0.9705	101	5	0.9706
CashMatching	0.8	634	24	0.7957	783	29	0.7982
	0.85	873	35	0.8483	1078	40	0.8480
	0.9	581	24	0.8981	725	28	0.8982
	0.95	330	13	0.9462	441	17	0.9470
	0.98	159	6	0.9755	324	13	0.9767
	0.99	213	8	0.9863	353	14	0.9865

2.10. ábra. A 'Coffee9' és 'CashMatching' feladatok számítási eredményei

költségű oszlopnak is ez alatt a küszöb alatt van a redukált költsége.

Ebben az alfejezetben a valószínűség maximalizálási megközelítés a valószínűségi függvény epigráfjának poliéderes közelítésén alapul. Új közelítő pontot találni a jelen sémában könnyebb, mint egy p efficiens pontot találni Dentcheva, Prékopa és Ruszczyński [23] klasszikus sémájában. A jelen alfejezetben bemutatott eljárásban egy közelítő pontot

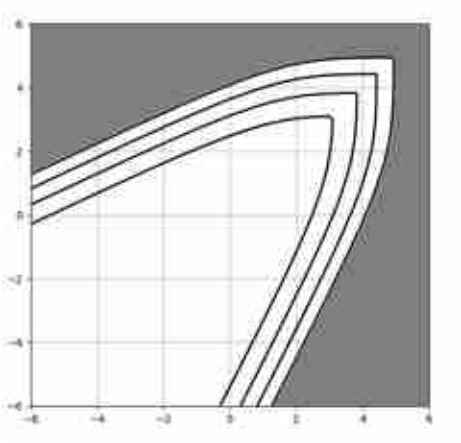
korlátozás nélküli optimalizálással találunk. LP kifejezéssel ez egy oszlopgenerálási séma, ahol a redukált költség maximalizálásával új oszlopokat találunk.

Az epigráf belső közelítő modellje a gradiens számítás során immunis a zajra, a következő értelemben. Tegyük fel, hogy a k iterációnál a következő iteráció, $\boldsymbol{z}_{k+1}$ csak egy hozzávetőleges megoldása a releváns részproblémának (redukált költség maximalizálása). Amíg a $\phi(\boldsymbol{z}_{k+1})$ ésszerű pontossággal kerül kiszámításra, a modell valódi belső közelítés marad.

Számítási kísérleteink azt mutatják, hogy a részproblémák durva közelítő megoldása elegendő a konvergenciához. Ennek a megfigyelésnek elméleti magyarázatát is adjuk, vagyis hogy miért lesz elegendő, hogy a kútfő csak közelítőleg optimális megoldásokat ad vissza.

2.5.2. Az alkalmazott algoritmus elméleti alátámasztása

A valószínűségi függvény jól kondicionált jellegét szemléltetjük kétdimenziós standard normális eloszlás esetén, mérsékelten függő margókkal (kovariancia $0,5$).



2.11. ábra. A $\nabla^2\phi(\boldsymbol{z})$ Hesse-mátrix kisebb sajátértéke
$(-6 \leq z_1, z_2 \leq +6)$

A $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ függvény Hesse-mátrixának sajátértékeit ábrázoljuk, ahol $F(\boldsymbol{z})$ az eloszlásfüggvény. Kiszámoltuk a Hesse-mátrix két sajátértéke közül a kisebbet és a nagyobbat, míg a $\boldsymbol{z}$ mindkét komponense a $[-6, +6]$ intervallumba esik.

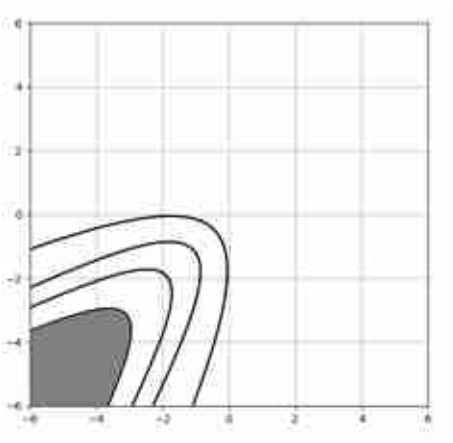
A 2.11. ábra a kisebb sajátértéket ábrázolja. A kontúrvonalak jobbról fent: $1e-5, 1e-4, 1e-3, 1e-2$. A szürkével nem töltött területen a kisebb sajátérték $1e-5$ felett van.

A 2.12. ábra a nagyobb sajátértéket ábrázolja. A kontúrvonalak jobbról fent: 1, 1,2, 1,4, 1,6 dollár. A szürkével nem töltött területen a nagyobb sajátérték 1,6 USD alatt van.

Ezek a kísérletek azt mutatják, hogy van egy meglehetősen nagy biztonságos tartomány, amelyen a $\phi(\boldsymbol{z})$ jól kondicionált.

3. Feltevés. *A $\phi(\boldsymbol{z})$ függvény kétszer folytonosan differenciálható, és ismertek α, ω $(0 < \alpha \leq \omega)$ valós számok, hogy*

$$\alpha I \preceq \nabla^2\phi(\boldsymbol{z}) \preceq \omega I \qquad (\boldsymbol{z} \in \mathbb{R}^n).$$



2.12. ábra. A $\nabla^2\phi(\boldsymbol{z})$ Hesse-mátrix nagyobb sajátértéke
$(-6 \leq z_1, z_2 \leq +6)$

Itt $\nabla^2\phi(\boldsymbol{z})$ a Hesse-mátrix, I az azonosságmátrix, és a $U \preceq V$ reláció a mátrixok között azt jelenti, hogy a $V - U$ pozitív szemidefinit.

Ez szemléletesen azt jelenti, hogy ez a függvény nem torzítja el nagyon a teret.

Ennek a gyakorlati megközelítésnek a hatékonysága a következő jól ismert tétel alapján támasztható alá:

4. Tétel. *Legyen a 3. feltevés érvényes az $f : \mathbb{R}^n \to \mathbb{R}$ függvényre. Minimalizáljuk az $f(\boldsymbol{z})$ értékét $\mathbb{R}^n$ felett a legmeredekebb ereszkedés módszerével, egy $\boldsymbol{z}^0$ pontból kiindulva. Jelölje $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^j, \ldots$ azokat az iterációkat, amelyeket minden lépésben pontos iránykeresés alkalmazásával kapunk. Akkor*

$$f\left(\boldsymbol{z}^j\right) - \mathcal{F} \;\leq\; \left(1 - \frac{\alpha}{\omega}\right)^j \left[\, f\left(\boldsymbol{z}^0\right) - \mathcal{F} \,\right], \tag{2.32}$$

ahol $\mathcal{F} = \min_{\boldsymbol{z}} f(\boldsymbol{z})$.

Ez a tétel megtalálható például Luenberger és Ye [60] cikkének 8.6. fejezetében. Az alábbi következményt bizonyítottuk az [S5] cikkünkben:

5. Következmény. *Legyen β $(0 < \beta \ll 1)$ adott. $O(-\log\beta)$ lépésben a legmeredekebb ereszkedés módszerével találunk egy $\widehat{\boldsymbol{z}}$ vektort úgy, hogy*

$$\overline{\rho}\,(\widehat{\boldsymbol{z}}) \;\geq\; (1-\beta)\,\overline{\mathcal{R}}, \tag{2.33}$$

ahol $\overline{\mathcal{R}}$ a (2.28)-ban definiált legnagyobb elérhető költség.

Tekintettel a korábban említett Markowitz-szabályra, az 5. következményben szereplő $\widehat{\boldsymbol{z}}$ vektor meglehetősen jóljavító vektor az oszlopgenerálási sémában.

Az aktuális megoldás közel optimálisságának ellenőrzésére a szokásos LP leállítási szabályt használjuk: bármely jelölt vektor redukált költségének egy rögzített optimalitástűrés alatt kell lennie. Természetesen nem ismerjük az $\overline{\mathcal{R}}$-t, de legyen

$$\overline{\mathcal{B}} := \frac{1}{1-\beta}\overline{\rho}\,(\widehat{\boldsymbol{z}}), \tag{2.34}$$

az 5. következmény β és $\widehat{\boldsymbol{z}}$ értékével. Leállítjuk az oszlopgenerálási eljárást, ha $\overline{\mathcal{B}}$ az optimalitástűrés alá esik. Ennek a korlátnak az alkalmazásakor a folyamat során egy fix β értékkel dolgozunk, például legyen $\beta = 0,5$. A jelenlegi (2.21) speciális lineáris programozási probléma esetében ez nem csak egy heurisztikus szabály:

6. Megfigyelés. *$\overline{\mathcal{R}}$ (és ebből következően $\overline{\mathcal{B}}$) a (2.21) modellprobléma megfelelő optimuma és a (2.15) eredeti konvex probléma közötti gap felső korlátja.*

7. Megjegyzés. *A $\overline{\mathcal{B}}$ laza optimalitástűrés előírása az oszlopgenerálási folyamat korai leállítását eredményezi. – Az LP problémákkal kapcsolatos általános tapasztalat az, hogy a számítási erőfeszítés jelentősen csökken a leállási tűrés lazításával.*

8. Megjegyzés. *Az oszlopgenerálási megközelítést duális szemszögből nézve vágósíkos módszert láthatunk. Ez a kapcsolat a primál és a duál megközelítés között jól ismert, mely megtalálható például Frangioni [33] cikkében.*

A duális nézőpont igazolja az $\overline{\boldsymbol{u}}$ optimális duális vektorok sorozatának konvergenciáját.

2.6. Valószínűség maximalizálás véletlenített eljárással

Az utóbbi időben a bonyolultabb feladatok megoldásában gyakran alkalmaznak véletlenített eljárásokat.

Mivel eddig a futási idő nagy része még mindig a Genz-kódban volt, lazítottunk a pontosságon, vagyis sem a függvényértékeket, sem a gradiens értékeket nem számoltuk ki nagy pontossággal. A gradienst komponensenként tudjuk számolni, ha növelni akarjuk a pontosságot, az egyre nagyobb számítási erőforrást igényel. A Genz-kód is véletlenített eljárás, ezért csak azt lehet várni, hogy nagy valószínűséggel jó közelítést ad. A függvényérték kiszámítása sokkal kisebb erőfeszítés, mint a gradiens kiszámítása. Ezért jogos, hogy a gradienst nem túl pontosan, a függvényértéket pontosan számoljuk.

A Genz-kódban a pontosságot a későbbiekben ismertetett kísérleteim eredményei alapján a korábbi iterációból kapott eredményből számolt értékkel szabályoztam.

A megoldó újabb módosítása

A Matlabot az IBM ILOG CPLEX (12.6.3-as verzió) optimalizáló eszköztárával használtuk, a többváltozós normális eloszlási értékek numerikus kiszámítását a [38] által megvalósított QSIMVNV Matlab függvény segítségével végeztük.

Megoldónk a 2.4.1. alfejezetben használt megvalósításon alapul. A kezdő megoldást és a mesterfeladat inicializálását az ott ismertetett eljárással állítottam be. A kezdő megoldás beállításához szükséges idő elhanyagolható volt az oszlopgenerálási séma egyetlen iterációjához szükséges időhöz képest.

A véletlenített oszlopgenerálási séma során minden oszlopgenerálási részproblémában egyetlen vonalmenti keresést végzünk. Ez a keresés az aktuális $\overline{\boldsymbol{z}}$ vektorról indul. A $\nabla\phi(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{u}}$ gradienseket meg kell becsülni. (Ezt majd a 11. megjegyzésben fogjuk indokolni.) Ez visszavezet az eloszlásfüggvény $\nabla F(\overline{\boldsymbol{z}})$ gradiensének becsléséhez. A $\nabla F(\overline{\boldsymbol{z}})$ egy komponensét viszont az alábbi (2.35) képlet szerint kapjuk.

A többváltozós normális valószínűségi eloszlásfüggvény értékeinek és gradienseinek becslése

Ha egy többváltozós valószínűség-eloszlásfüggvény mindenhol differenciálható, akkor a parciális deriváltjai az alábbi általános képlettel rendelkeznek

$$\frac{\partial F(z_1,\ldots,z_n)}{\partial z_i} = F(z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n|\ z_i)f_i(z_i) \tag{2.35}$$

ahol $F(z_1,\ldots,z_n)$ a $\xi_1,\ldots,\xi_n$ véletlen változók valószínűségi eloszlásfüggvénye, és $f_i(z)$ a ξ_i véletlen változó valószínűségi sűrűségfüggvénye. $F(z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n|\ z_i)$ a $\xi_1,\ldots,\xi_{i-1},\ \xi_{i+1},\ldots,\xi_n$ véletlen változók feltételes valószínűségi eloszlásfüggvénye, feltéve, hogy $\xi_i = z_i$. Lásd a (6.6.22) képletet Prékopa [78] könyvének 203. oldalán.

Ismert, hogy a többváltozós normális valószínűségeloszlás bármely feltételes valószínűségeloszlása szintén normális. Ezért a (2.35) képletből következik, hogy a többváltozós normális valószínűségeloszlás függvényértékeit és azok parciális deriváltjait ugyanezzel az eljárással kiszámíthatjuk. A következő részben a többváltozós valószínűségeloszlásfüggvény értékek becslésére szolgáló Genz-módszer leírását adjuk meg.

Genz-módszer

A módszert Genz 1992-ben a [38] cikkben publikálta. Ebben a cikkben Genz egy téglalap többváltozós normális valószínűségi tartalmának becslésével foglalkozott, ami általánosabb probléma, mint a többváltozós valószínűség-eloszlásfüggvény értékének kiszámítása.

Az alapötlet az integrációs területnek a $[0,1]^n$ egységkockára való transzformálása elemi transzformációk sorozatával. Ez egy kicsit bonyolultabb integrandus költséggel jár.

A sorozat a Cholesky-transzformációval kezdődik, amely a többváltozós normális eloszlású véletlen vektor komponenseit független véletlen változókká alakítja, azonban az integrálási határértékek bonyolultabbá válnak. Ezután az integrálandó változókat tovább transzformáljuk az egydimenziós standard normális valószínűségeloszlású eloszlásfüggvény inverz függvényével. Ennek a transzformációnak az a hatása, hogy minden integrálandó egyenlő lesz eggyel, de az integrációs határok még bonyolultabbá válnak. Végül egy egyszerű lineáris transzformációval az integrációs tartomány a $[0,1]^n$ egységkockára változik, és az integránsfüggvények a korábbi bonyolult integrációs határok differenciái lesznek.

Megjegyezzük, hogy az i-edik integránsfüggvény mindig független az i-edik integránsváltozótól, és kihúzható egy integrálból, ami lehetővé teszi a legbelső integrál explicit integrálását. Így a numerikus integrálás elvégezhető a $[0,1]^{n-1}$ egységkockán is.

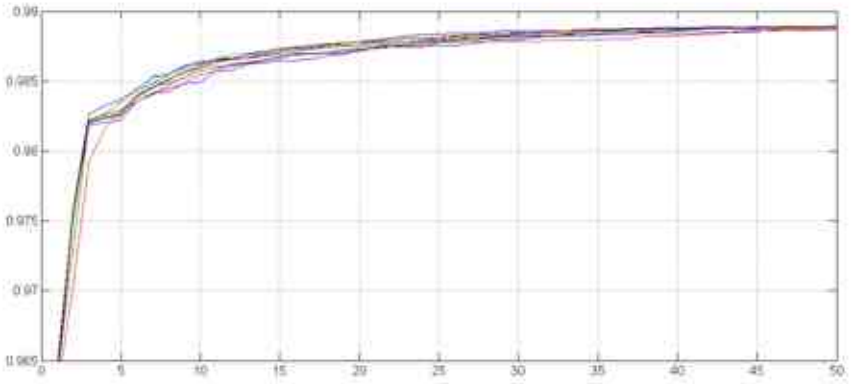
Ez a transzformációsorozat kikényszerít egy prioritási sorrendet az $\mathbf{x}$ komponenseire, ami lehetővé teszi a hatékony integrálást.

Genz három különböző módszert ír le ennek az átalakított integrálnak a megoldására. Az első módszer az integráns polinomiális közelítésén alapul. A jobb teljesítmény érdekében az egységkockát részterületekre osztják, amelyeket később tovább osztanak, ha a közelítés nem eléggé pontos. A második módszer kvázi-véletlenszerű integrálási pontokat használ. Végül a harmadik módszer pszeudo-véletlen integrációs pontokat használ, amelyeknél a hibabecslések statisztikai jellegűek.

2.6.1. Kísérleti tapasztalatok

Az alábbiakban a véletlenített eljárás tesztelését mutatom be. Jó megállási feltételeket kerestünk, ezért 50 iterációt futtattam. (Egy iteráció egy mesterfeladat megoldását és egy oszloppal való kibővítését jelenti.) Az volt a tapasztalatunk, hogy még így is gyorsabban futott le, mint a 2.5.1. alfejezetbeli kísérletekben.

A Genz-kód pontosságának beállítására kísérleteteket végeztem, melyeket később, a 2.7. alfejezetben ismertetek. Kidolgoztam egy eljárást, amellyel a Genz-szubrutinok pontosságát szabályozom a megoldás folyamán. Jelölje $\nabla\phi(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{u}}$ az aktuális gradienst a kútfő feladatában és $\nabla\phi(\overline{\boldsymbol{z}}_{-}) - \overline{\boldsymbol{u}}_{-}$ az előző kútfő feladatban a kezdőpont gradiensét. Jelen kísérletünkben úgy szabályozom a pontosságot, hogy az aktuális gradiens hibájának normája kisebb legyen, mint az előző gradiens normájának egytizede. A Cash-matching probléma megoldása során a $\phi(\boldsymbol{z}_i)$ függvényértékek kiszámítása mindig nagy pontossággal történik, a minta méretét 10000-re állítva.



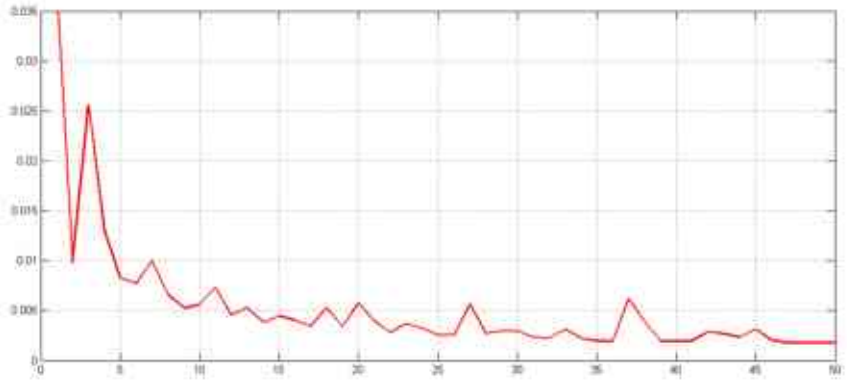
2.13. ábra. A kapott valószínűségi szintek az iterációs számok függvényében. A különböző futásokat különböző szálak képviselik. $p = 0,99$

A véletlenített eljárás 10 futtatását végeztem el, mindegyiket 50 iterációval. A kapott valószínűségi szintek, azaz a $F(\overline{\boldsymbol{z}})$ értékek sorozatait a 2.13. ábra mutatja.

Az eljárás során a $\nabla\phi(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{u}}$ gradienst minden iterációnál $\overline{\boldsymbol{G}} - \overline{\boldsymbol{u}}$ becsüli. Ennek a becslésnek a normája az eljárás előrehaladtával csökken. Egyetlen tipikus futtatás esetén ez a csökkenés a 2.14. ábrán látható.

Az iterációszámon kívül nem volt leállási feltétel. 50 iteráció után a különböző futásokban kapott optimális valószínűségi szintek már nagyon közel voltak egymáshoz (a legmagasabb és legalacsonyabb különbség kisebb, mint 0,0003). Másrészt a 2.7.3. részben (2.42) képlettel definiált $\overline{\mathcal{B}}$ felső becslés a hibára, értéke 0,025 és 0,03 között volt a futtatás végén. Arra a következtetésre jutottunk, hogy bár a határoló eljárás működőképes, további technikai fejlesztésekre van szükség ahhoz, hogy lépést tartson a sztochasztikus közelítési sémával.

Nem írtunk elő korlátozást az új $\boldsymbol{z}_i$ oszlopokra, mégis mindig a $\mathcal{Z}$ kockában maradtak. Ráadásul a valószínűségi szint az oszlopgenerálási folyamat során hozzáadott oszlopokkal minden iterációban magas volt, $F(\boldsymbol{z}_i) \geq 0,9$. Ez lehetővé tette az összes valószínűségi függvényérték nagy pontosságú kiszámítását. A $\boldsymbol{z}' \in \mathcal{Z}$ korlátozása soha nem volt aktív



2.14. ábra. A gradiens norma csökkenése az iterációs számok függvényében, egyetlen futásban.

a mester probléma $\boldsymbol{z}' = \overline{\boldsymbol{z}}$ optimális megoldásában. A parciális deriváltak (2.35) szerinti számításánál előforduló sűrűségfüggvény értékek mindig is szignifikánsak voltak. Az egyetlen gradiens számítás során előforduló 15 sűrűségfüggvény érték közül kettő mindig 10^{-2} nagyságrend körül volt, egy másik $5 * 10^{-3}$, a többi pedig 10^{-3} körül volt. Más problémák esetén (2.35)-ben sok parciális derivált esetén nullához közeli sűrűségfüggvény értékek fordulhatnak elő. Az ilyen komponensek esetében a megfelelő feltételes eloszlásfüggvényt nem kell kiszámítani. Mindegyik futtatásban 50 iterációt hajtottam végre, egy-egy futtatás ideje körülbelül 2 perc volt. Bár hosszúnak tűnik, arra számítunk, hogy a technikai fejlesztések jelentősen lerövidítik a megoldási időt. (Tapasztalataink szerint a technikai fejlesztések egy-két nagyságrendű gyorsulást eredményezhetnek.)

A valószínűségi függvényértékek kiértékeléséről

Számítási kísérleteinkben a valószínűségi függvényértékeket nagy pontossággal számítottuk ki. Ez azért lehetséges, mert az oszlopgenerálási sémában minden egyes $\boldsymbol{z}_i$ oszlopnál $F(\boldsymbol{z}_i) \geq 0,9$ volt.

Előfordulhat azonban, hogy van egy $\boldsymbol{z}_i$ oszlopunk alacsony megfelelő valószínűséggel. Ilyen esetben a $\phi(\boldsymbol{z}_i) = -\log F(\boldsymbol{z}_i)$ függvényérték hibája nagy lehet, mert a logaritmus felerősítené a valószínűségszámítás hibáját.

Ezt a problémát az algoritmikus keretrendszerünkben tudjuk kezelni. Adott egy új $\boldsymbol{z}_i$ oszlop, legyen $\underline{p}_i > 0$ olyan, hogy a $\underline{p}_i \leq F(\boldsymbol{z}_i)$ szinte biztosan érvényes legyen. (A szimulációs eljárások általában egy konfidenciaintervallumot adnak meg becsléssel együtt.) A (2.21) modellproblémában szerepeltetjük a $\boldsymbol{z}_i$ oszlopot a $\phi_i = -\log \underline{p}_i$ költségegyütthatóval. Ilyen elrendezés mellett a modellünk konzisztens marad, azaz a $\phi_k(\boldsymbol{z})$ függvény szinte biztosan a valószínűségi függvény belső közelítése.

Hasonlóan az iránymenti keresési (line search) eljárásokban is mindig biztonságosan alacsonyabb becslésekkel dolgozunk a releváns valószínűségekre. Mivel minden oszlopgenerálási részproblémában iránymenti keresést végzünk, minden iránymenti keresés kiindulópontját az aktuális modellfeladat optimális megoldásaiból kapjuk. Ugyanis a kiindulópont a $\overline{\boldsymbol{z}}$, a (2.23) formula szerint. Az [S4] cikk 25. feltevése szerint ismerünk

egy megvalósítható $\check{\boldsymbol{z}}$ vektort úgy, hogy $F(\check{\boldsymbol{z}}) \geq 0,5$. Ha a kezdeti oszlopok közé a $\check{\boldsymbol{z}}$ értéket is hozzáadjuk, akkor mindig $F(\overline{\boldsymbol{z}}) \geq 0,5$ lesz. Ezért a $\phi(\overline{\boldsymbol{z}})$ nagy pontossággal kerül kiszámításra. Ezenkívül az iránymenti keresés nagy pontossággal történik a $\mathcal{L}(F,\, 0.5) = \{\, \boldsymbol{z} \,|\, F(\boldsymbol{z}) \geq 0,5 \,\}$ régióban, amely nyilvánvalóan tartalmazza az optimális $\boldsymbol{z}$-t is.

2.6.2. A véletlenített kísérletek elméleti alátámasztása

Az alábbiakban kiterjesztjük a módszert a gradiens becslésekre, az 1. feltevés helyett a következőre:

9. Feltevés. *Adott $\boldsymbol{z}, \boldsymbol{u} \in \mathbb{R}^n$, a $\phi(\boldsymbol{z})$ függvényérték nagy pontossággal kiszámítható, és a $\|\nabla\phi(\boldsymbol{z}) - \boldsymbol{u}\|$ normája előre meghatározott pontossággal megbecsülhető. Ezen túlmenően a $\nabla\phi(\boldsymbol{z})$ gradiensvektor torzítatlan becslése előállítható, ezt jelöljük $\boldsymbol{G}$-vel. A $\boldsymbol{G}$ véletlen vektor előállítható úgy, hogy $E(\|\boldsymbol{G} - \nabla\phi(\boldsymbol{z})\|^2)$ egy előre meghatározott tolerancia alatt marad. (Nagyobb pontosság normabecslés esetén, és szigorúbb tolerancia a variancia tekintetében nagyobb számítási erőfeszítést igényel.)*

Az [S4] cikkben kidolgoztuk az oszlopgenerálási módszer véletlenített változatát és bemutattuk megbízhatósági szempontokat a 3. feltevés alapján.

Először kiterjesztjük a 4. tételt. Legyen $f : \mathbb{R}^n \to \mathbb{R}$ úgy, hogy a 3. és a 9. feltevések érvényesek legyenek. A $f(\boldsymbol{z})$ $\mathbb{R}^n$ feletti értékét sztochasztikus ereszkedés módszerrel szeretnénk minimalizálni. Jelöljön $\boldsymbol{z}^\circ \in \mathbb{R}^n$ egy iterációt, és $\boldsymbol{g}^\circ = \nabla f(\boldsymbol{z}^\circ)$ a megfelelő gradiens.

Legyen $\sigma^2 > 0$ adott. A 9. feltevés szerint egy $\boldsymbol{G}^\circ$ véletlen vektor realizációi konstruálhatók, amelyek kielégítik az alábbiakat:

$$\mathrm{E}\,(\boldsymbol{G}^\circ) = \boldsymbol{g}^\circ \qquad \text{és} \qquad \mathrm{E}\left(\|\boldsymbol{G}^\circ - \boldsymbol{g}^\circ\|^2\right) \;\leq\; \sigma^2\,\|\boldsymbol{g}^\circ\|^2\,. \tag{2.36}$$

A (2.36)-ból következik:

$$\mathrm{E}\left(\|\boldsymbol{G}^\circ\|^2\right) \;=\; \mathrm{E}\left(\|\boldsymbol{G}^\circ - \boldsymbol{g}^\circ\|^2\right) + \|\boldsymbol{g}^\circ\|^2 \;\leq\; (\sigma^2+1)\,\|\boldsymbol{g}^\circ\|^2\,. \tag{2.37}$$

A Genz-kódban nem akarunk pontosan számolni. Hanem, hogy definíció szerint a $\boldsymbol{G}^\circ$ szórásnégyzete valamilyen korlát alatt maradjon.

A 4. tétel következő véletlenített alakját vesszük figyelembe:

10. Tétel. *A fenti feltételezések szerint a legmeredekebb süllyedési módszert hajtjuk végre gradiens becslések segítségével: az aktuális $\boldsymbol{z}^\circ$ iterációnál egy $\boldsymbol{G}^\circ$ gradiens becslés jön létre, és iránymenti keresést hajtunk végre abban az irányban. Feltételezzük, hogy a gradiens becslések a megfelelő iterációkhoz függetlenül jönnek létre, és (2.36) - (2.37) mindegyikre érvényes.*

A $\boldsymbol{z}^0$ pontból indulva és j sorkeresést végrehajtva jelölje $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^j$ a megfelelő iterációt. Akkor van

$$E\left[f\left(\boldsymbol{z}^j\right)\right] - \mathcal{F} \;\leq\; \left(1 - \frac{\alpha}{\omega(\sigma^2+1)}\right)^j \left(f\left(\boldsymbol{z}^0\right) - \mathcal{F}\right), \tag{2.38}$$

ahol $\mathcal{F} = \min_{\boldsymbol{z}} f(\boldsymbol{z})$.

A tétel bizonyítása az [S4] cikkben megtalálható.

Ezzel a módszerrel is jó pár iterációval meglehetősen jó javító pontot tudunk találni.

11. Megjegyzés. *A $-\overline{\rho}(\boldsymbol{z})$ függvény gradiensei $\nabla\phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}$ alakúak. Minél tovább halad az oszlopgenerálási eljárás, annál kisebb lesz a $\|\nabla\phi(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{u}}\|$ norma (lásd: 2. tétel (c) pontja).*

A variancia követelményének (2.36) teljesítéséhez egyre jobb becslésekre van szükség. A pontosságot a 9. feltevés szerint szabályozzuk.

2.7. Kísérletek a Genz-kód számítási pontosságának beállítására

A valószínűség maximalizálási feladat 2.4.1. részben megfogalmazott megoldási folyamata során szükségünk van a függvényérték és a gradiens kiszámítására, melyet Genz [38] által kidolgozott módszer Matlabban megvalósított függvényével számoltunk ki. Mivel a Genz-kód véletlenített eljárás, a szimuláció mintanagysága befolyásolja a számítás pontosságát. Mivel az volt a tapasztalatunk hogy a Genz-kódban töltött futási idő lényegesen nagyobb a többi számítás futási idejéhez képest, ezért a költséghatékonyság szempontjából fontos, hogy a Genz-kód a szükséges pontosságot ne meghaladó, de kellő pontossággal számoljon. Az alábbiakban a Genz-kód költséghatékonyságának vizsgálatára végzett kísérleteimet mutatom be.

2.7.1. A Genz-kód számítási pontossága és a mintanagyság közötti összefüggés vizsgálata

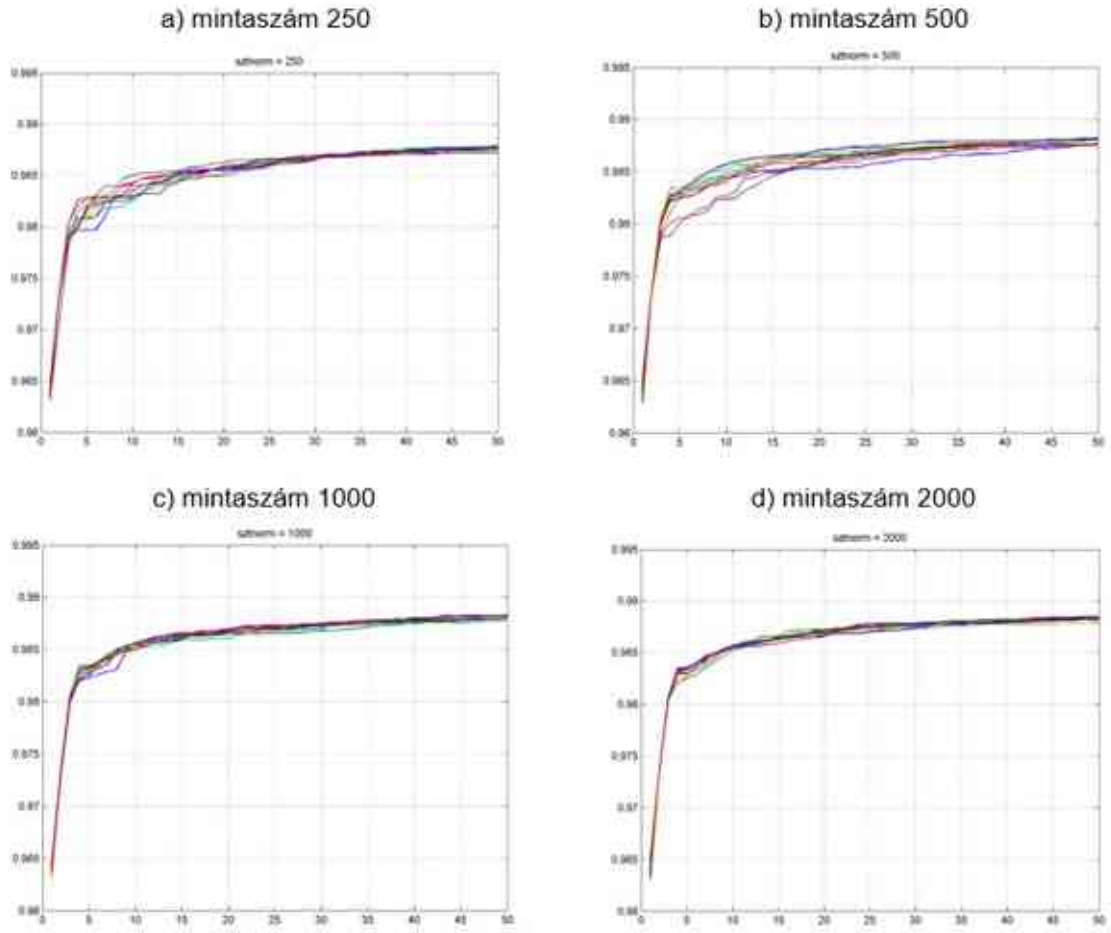
A Genz-kód számítási pontossága és a mintanagyság összefüggésére végzett kísérleteket az [S8] cikkben mutattam be. A véletlenített módszer teszteléséhez a 2.5. részben ismertetett feladatok közül a „cash matching" feladatot használtam fel. Az eredetileg valószínűségi korlát melletti költségbecsléssel kialakított tesztfeladatot valószínűség maximalizálására fordítottuk át. A költségkorlátok jobb oldalát úgy állítottuk be, hogy a megfelelő optimális valószínűségi szint $p = 0,99$ legyen. Ezekre a számításokra Szántai számítógépes kódját használtuk [104]. Az előzetes tesztekben a futási időket vizsgáltam különböző pontosságú becslések esetén. A Genz-féle kódban a gradiens kiszámítása szimulációs eljárással történik, a mintaszám befolyásolja a pontosságot. Különböző $(250, 500, 1000, 2000)$ mintaszámokkal végeztem futtatásokat (10-et minden mintaszám esetén), és ezeket átlagoltam. Azt vizsgáltam, hogy milyen összefüggés van a mintaszám és a futási idő között, illetve, hogy a mintaszám mennyiben befolyásolja azt, hogy a megoldás milyen gyorsan konvergál az optimumhoz. Azért, hogy az optimumhoz való konvergálást összehasonlíthassam, képeztem az egyes mintaszámok esetén a 10 futtatáskor kapott valószínűségek átlagait, majd ezen átlagok és az elvárt p valószínűségi szint RMSE (root mean squared error) értékekeit számítottam ki a következő képlet alapján:

$$RMSE_m = \sqrt{\frac{\sum_{i=1}^{n} (p_{i,m} - p)^2}{n}}, \tag{2.39}$$

ahol m a mintaszám, n az iterációk száma, $p_{i,m}$ az m mintaszám esetén, az i-edik iteráció során kapott valószínűségek átlaga.

2.7.2. Kísérleti tapasztalatok

A 2.15. ábrán jól látható, hogy a kisebb mintaszámok esetén a valószínűség lassabban konvergál az optimumhoz, s hogy a különböző futási eredmények között is nagyobb a különbség. A nagyobb mintaszám esetén az optimumhoz való konvergencia sokkal gyorsabb.



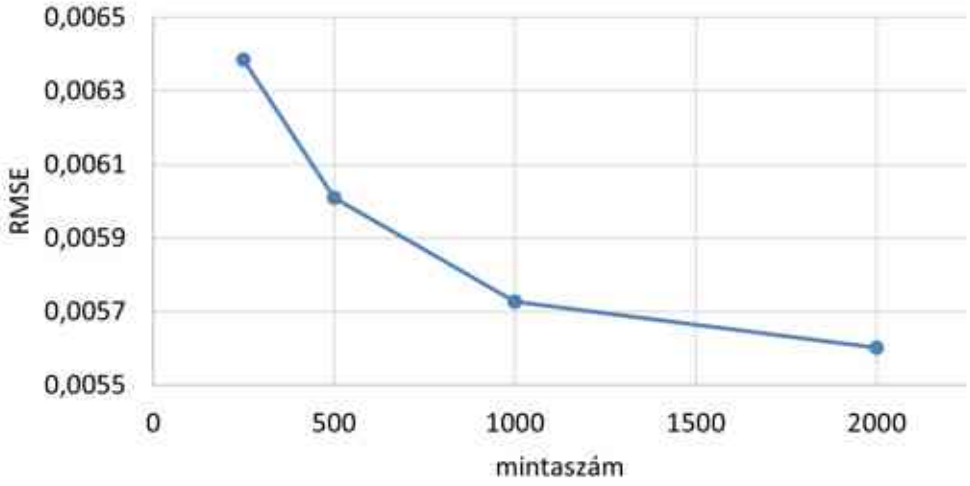
2.15. ábra. Az iterációk során kapott valószínűségek különböző mintaszámok esetén (vízszintes tengely: iterációk száma; függőleges tengely: a kapott valószínűség)

Az RMSE (2.39) alapján kiszámított értékekei a 2.16. ábrán jól láthatóan mutatják, hogy a mintaszám növekedésével csökken a hiba, tehát egyre pontosabb eredményeket kaptam. A hiba csökkenés ütemére az volt a sejtésem, hogy az a mintaméret reciprokával vagy reciprokának négyzetével arányos. Ezért a következő kísérleteket ennek ellenőrzésére végeztem, melyet a következő részben fogok ismertetni.

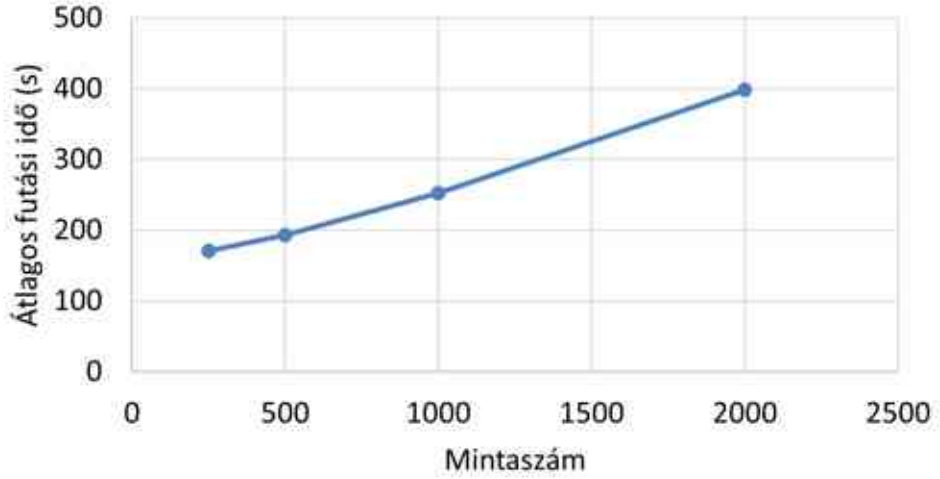
Egy-egy futtatás átlagos futási idejét a 2.1. táblázat tartalmazza. A táblázatban és a 2.17. ábrán is jól látszik, hogy a nagyobb mintaszámok esetén a futási idő megnövekszik.

2.1. táblázat. Az átlagos futási idők

Mintaszám	250	500	1000	2000
Futási idő (s)	170,95	193,37	252,42	398,82



2.16. ábra. Az RMSE értéke a mintaszám függvényében



2.17. ábra. Az átlagos futási idők a mintaszám függvényében

Az eredmények alapján elmondható, hogy a pontosabb eredmény kiszámítása nagyobb futási időt igényel.

Ebben a kísérletben az [S4] cikkben kidolgozott véletlenített eljárás implementációját és tesztelését mutattam be. Ebben a módszerben a gradiens számítása nem annyira pontos, mint a belső közelítéssel való megoldás, viszont a függvényértékeket megfelelő pontossággal számítjuk ki, az epigráfot így is felülről közelítjük. Az eredmények azt mutatják, hogy érdemes ezzel a módszerrel dolgozni, mert rövidebb futási időket kapunk.

2.7.3. A mintanagyság dinamikus változtatásának hatásaira végzett vizsgálatok

A véletlenített módszer tesztelésére végzett másik kísérleteim arra vonatkoztak, hogy a Genz-kódnak megadott mintanagyság különböző módokon való dinamikus változtatása hogyan hat a szükséges iterációszámra és a futási időre. Ebben a kísérletben is a "cash matching" problémát használtam a tesztelésre. Az eredményeket a TEAM 2018 konferencián [S11] mutattam be.

A költségmegszorítások jobb oldalát úgy állítom be, hogy a megfelelő optimális valószínűségi szint $p = 0,9$ legyen. A számításokhoz Szántai [104] számítógépes kódját használtam.

A pontos $\overline{g}$ gradiens nem ismert, de tudunk adni erre egy véletlen becslést a Genz-kód segítségével. Jelölje a $\overline{G}$ a Genz-kód segítségével kapott véletlen vektort. Akkor a Genz-kód eljárása biztosítja, hogy a $\overline{G}$ véletlen vektor torzítatlan becslése a gradiensnek, ami azt jelenti, hogy

$$E(\overline{G}) = \overline{g}. \tag{2.40}$$

A Genz-kódban a pontosságot lehet szabályozni, ha adva van egy $\Delta \in \mathbb{R}$, $\Delta > 0$ hibatűrés és egy kicsi $p(0 < p \ll 1)$ valószínűség, akkor létezik egy olyan kicsi $\overline{\mathcal{I}}$ konfidencia-intervallum, hogy

$$P(\overline{G} \in \overline{\mathcal{I}}) \geq 1 - p \quad \text{és} \quad \text{diag}(\overline{\mathcal{I}}) \leq \Delta\ . \tag{2.41}$$

Ezt szemléletesen úgy lehet elképzelni, hogy $\overline{\mathcal{I}}$ egy kis kocka, amelynek középpontja a $\overline{g}$ gradiens és $\text{diag}(\overline{\mathcal{I}})$ a kis kocka átmérőjét jelenti.

Kísérleteket végeztem a pontosság szabályozására és a leállási feltétel meghatározására. Az [S4] cikkünkben Fábián Csaba megmutatta, hogy az alábbi $\overline{\mathcal{B}}$ mennyiség nagy $(1-p)$ valószínűséggel felső becslést ad a hibára.

$$\overline{\mathcal{B}} = (\phi_k(\overline{z}) - \phi(\overline{z})) + \max_{z \in Z}(\overline{u} - \overline{G})^T(z - \overline{z}) + \Delta \text{diag}(Z), \tag{2.42}$$

ahol a $\Delta > 0$ adott hibatolerancia, $\overline{\boldsymbol{z}} \in \mathcal{Z}$ az aktuális iterátor, $\overline{G}$ egy gradiens becslés, $\text{diag}(Z)$ a Z kocka átmérője. Itt a különböző összeadandó tagok különböző fajta számítási hibákat jelentenek a függvényértékekben és a gradiensekben. A $\overline{\mathcal{B}}$ felső becslés a hibára, ha egy érték alá csökken, leállítom az eljárást. A Genz [38] kódja szimulációt alkalmaz, ezért a mintaméretek befolyásolják a pontosságot. Kísérleteim során összehasonlítottam az iteráció számokat és a futási időket, amelyek ahhoz szükségesek, hogy a $\bar{\mathcal{B}}$ értéket $0,03$ alá csökkentsem.

Különböző stratégiákat teszteltem a mintanagyság növelésére az optimalizálási folyamat során. Egyrészt változtattam a kezdeti mintanagyságot, másrészt dinamikusan változtattam az oszlopgenerálási részprobléma aktuális becsült gradiens normájának reciproka alapján. A mintanagyságot vagy a norma reciproka arányában, vagy a norma reciprokának négyzetével arányosan növeltem.

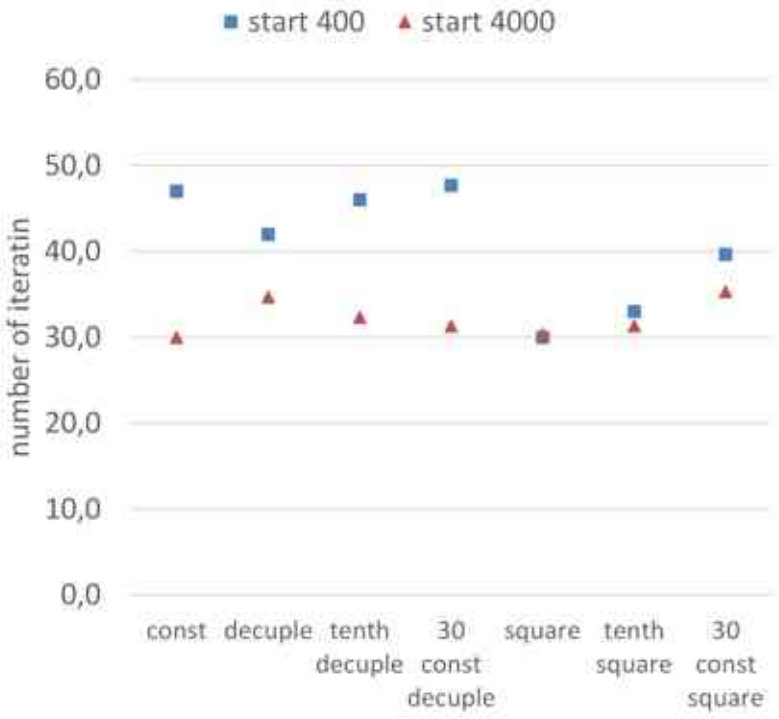
A stratégiák a következők voltak:

- **const**: Mindig állandó
  $mm(i) = mm0$
- **decuple**: A norma reciprokának tízszeresével növelve minden iterációban
  $mm(i) = round(1/gradnormobjective(i) * 10)$

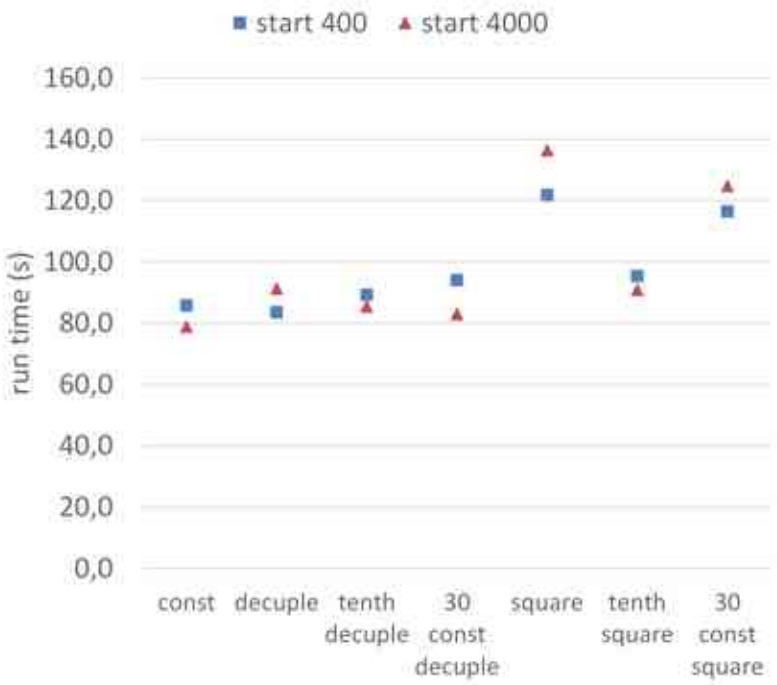
– **tenth decuple**: A norma reciprokának tízszeresével növelve minden tizedik iterációban. Állandó, majd minden 10. $mm(i) = round(1/gradnormobjective(i) * 10)$
– **30 const decuple**: Az első harminc állandó, utána a norma reciprokának tízszeresével növelve minden iterációban $mm(i) = round(1/gradnormobjective(i) * 10)$
– **square**: A norma reciprokának négyzetével növelve minden iterációban $mm(i) = round(1/(gradnormobjective(i)^2))$
– **tenth square**: A norma reciproka négyzetével növelve minden tizedik iterációban állandó, majd minden 10. $mm(i) = round(1/gradnormobjective(i)^2)$
– **30 const square**: Az első harminc állandó, utána a norma reciprokának négyzetével növelve minden iterációban $mm(i) = round(1/gradnormobjective(i)^2))$

2.7.4. Kísérleti tapasztalatok

A 2.18. ábra az iterációk számát mutatja különböző kezdő mintamérettel és különböző növelési stratégiákkal. Természetesen kevesebb iterációra van szükség, ha a kezdő mintaméretek nagyobbak. A futási idők nem mutatnak szignifikáns különbséget a különböző kezdő mintaméreteknél, de a norma reciproka négyzetén alapuló (utolsó 3) növelési stratégiák hosszabb futási időt eredményeznek (2.19. ábra). Ezért a későbbiekben a véletlenített eljárással kapcsolatos vizsgálatainkban a gradiens számításnál a norma reciprokának tízszeresével való növeléssel dolgoztam (lásd. 2.6.1. alfejezet), mivel a futási idő is és a iterációk száma is ennél a módszernél bizonyult legkedvezőbbnek.



2.18. ábra. Az iterációk száma különböző mintanövelési stratégiák esetén ($\overline{\overline{\mathcal{B}}} = 0,03$)



2.19. ábra. Futási idők különböző mintanövelési stratégiák esetén ($\overline{\overline{\mathcal{B}}} = 0,03$)

2.8. Összegzés

A 2. fejezetben a Fábián Csaba vezette kutatócsoport által, valószínűség maximalizálására kidolgozott epigráf-közelítő módszert mutattam be. Először a valószínűséggel megfogalmazott modellek és megoldó eljárások rövid történetét tekintettem át, majd a módszer alapjául szolgáló p-efficiens pont megközelítéseket.

Az általunk kidolgozott epigráf-közelítő módszer szemléletes bemutatását tartalmazó fejezet után részletes kifejtésre került a módszer, mégpedig a fejlesztés időbeli sorrendjében, ahogy egyre hatékonyabb és gyorsabb lett a megoldás. Ezen belül először bemutattam az oszlopgenerálási eljárást és a megoldó felépítését, amelyen az első kísérleteket végeztük. Mivel nagyon pontosan akartunk számolni, nagyon lassú volt a megoldás. A kutatás ezen részében a mesterfeladattal kapcsolatban a kezdőmegoldás keresést és a feladat inicializálását dolgoztam ki, valamint a megoldó (solver) futtató keretrendszerét és mesterfeladatot megoldó részt Matlabban implementáltam.

A következő fejezetben bemutatásra került az a javítás, hogy az új próbapontokat csak közelítőleg számítottuk ki, viszont a függvényértéket pontosan. A megoldóban elvégeztem a módosításokat. Az általam végzett kísérletek tapasztalatai azt mutatták, hogy még mindig hosszú volt a futási idő. A fejezet végén ismertettem az elméleti alátámasztását annak, hogy miért elég, ha a próbapontokat csak közelítőleg számoljuk ki.

A következő fejezetben az újabb javítás arra vonatkozott, hogy a véletlenített eljárás, azaz a gradiens becslések, nagy valószínűséggel kellő pontosságúak legyenek, ugyanakkor a függvényértékeket se számoljuk ki pontosan, hanem csak kellő pontossággal. Ami jelen esetben azt jelentette, hogy a pontosságot annak függvényében adjuk meg menet közben, hogy mennyire közelítettük meg a megoldást. A pontosság dinamikus szabályozására

eljárást dolgoztam ki és kísérleteket végeztem a módszer tesztelésére. A kísérleti tapasztalatokat és a véletlenített módszer működésének elméleti alátámasztását is ismertettem.

Az utolsó fejezetben a véletlenített eljárásban alkalmazott pontosság beállítására végzett kísérleteimet részletesen mutattam be. A kísérletek eredményeképpen egy gyakorlatban jól használható megoldó eljárást sikerült kidolgozni.

1. Tézis:

A valószínűség maximalizálási feladatok megoldásához alkalmazott oszlopgenerálás első lépéseként kidolgoztam egy eljárást megoldható mesterfeladat előállítására.

Ezen felül a mesterfeladatot úgy inicializáltam megfelelő oszlopok hozzávételével, hogy a megoldó eljárás megfelelően javító irányba induljon el.

A megoldónak a mesterfeladat részét és a kísérletekhez a futtató keretrendszert implementáltam, valamint kísérleteket végeztem különböző feladatokon.

Kidolgoztam egy eljárást a Genz-kód számítási pontosságának dinamikus szabályozására. Olyan megoldást sikerült kidolgozni, amellyel megoldás közben a gradiens becslés nagy valószínűséggel kellő pontosságú lett, és a függvényérték a szükséges pontosságot nem meghaladóan, de kellően pontos lett. Az eljárás tesztelésére és a megfelelő paraméterek megtalálására kísérleteket végeztem.

A kísérletek eredményeképpen egy gyakorlatban jól használható megoldó eljárást sikerült kidolgozni.

Kapcsolódó publikációk: [S4], [S5], [S8], [S11], [S10], [S12]

3. fejezet

Kockázat modellezése kopulák felhasználásával

A 2008-as pénzügyi válság után a matematikai modellek jósága erősen megkérdőjelezhetővé vált, ezért az új alapokra való helyezést célozták meg. Kiderült hogy a ritka, nagy veszteséggel járó események, amelyek tipikusan egyszerre következnek be, pénzügyi válsághoz vezetnek. Ebből következik, hogy fontos a ritka események és az összefüggés struktúrájának modellezése. Ezt a célt tartom szem előtt ebben a fejezetben bevezetett eljárásokkal.

A fejezet a következőképpen épül fel. Az első alfejezet a felhasznált fogalmakat tartalmazza, szigorúan leszűkítve azokra, amelyeket a szimulációs kutatásban használtam. Ebben a részben a kockázati mutatókra tett hatást is vizsgálom szimulált majd valós portfóliókra is. A második alfejezetben bemutatom a VaR és CVaR modellezési lehetőségeit rávilágítva az összefüggési struktúra szerepére. Bemutatom a Gauss-kopula előnyeit, melyről egyébként kevés szó esik és azt, hogy hogyan hatnak a peremek az aszimmetrikus összefüggésre, noha a kopula szimmetrikus. Három-dimenziós eloszlásokat generálok, bemutatva, hogyan hat az összefüggési struktúra, ugyanazon normális peremeloszlások, és korrelációk mellett, majd rámutatok, hogyan változnak az együttes eloszlások a struktúra és a peremek megváltoztatása által, adott korreláció mellett.

A harmadik fejezet az SSD portfólió optimalizálási problémával foglalkozik. A fejezet első részében beszélek a optimalizálási eljárásról, majd a második fejezetben ismertetem, az általam végzett szcenárió generálásokat és tárgyalom az eredményeket. A kopulákkal való szcenáriógenerálással portfólió optimalizálásban viszonylag kevesen foglalkoztak. Az SSD optimalizálás esetére pedig ez az első ilyen megközelítés.

A fejezet a következő cikkekre épít: [S1, S2, S6, S9].

3.1. Kapcsolódó fogalmak

Ebben az alfejezetben ismertetem azokat a fogalmakat, amelyeket az itt bemutatott kutatásban felhasználtam.

3.1.1. A kopula-függvény

A kopulák a mértékelmélethez vezethetők vissza, a fogalmat Sklar 1959-ben vezette be [99] cikkében. A kopulák azért lettek nagyon népszerűek, mert lehetővé teszik a peremeloszlások és a köztük lévő függőség külön modellezését. Így a többváltozós valószínűségi eloszlások sokkal rugalmasabb módon modellezhetők. Kopulákról és kopula családokról áttekintő művek jelentek meg, többek között Nelsen [70] és Joe [46] könyvei. Korai magyar nyelvű áttekintő cikkeket Varga József [118] és Kovács Edith Alice [51, 50, 52, 53] jelentettek meg a kopulák alkalmazásával kapcsolatban.

A kopulaelmélet alapjainak első összefoglaló áttekintése Nelsen 1999-es *Introduction to Copulas* című munkája [70]. Ebben a műben a d-dimenziós kopula általános definíciója a következő:

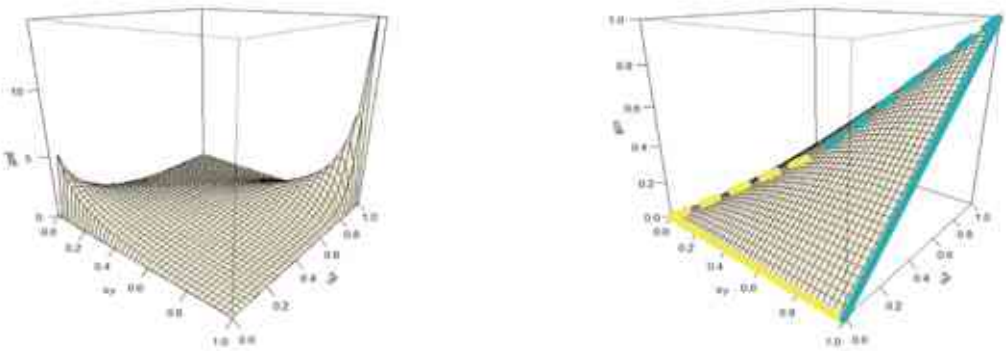
12. Definíció. *A* $C : [0,1]^d \to [0,1]$ *függvény d-kopula (d-dimenziós kopula), ha rendelkezik a következő tulajdonságokkal:*

(i) $\forall u_i \in [0,1]$ *esetén, ha van legalább egy* $u_i = 0$*, akkor* $C(u_1, \ldots, u_d) = 0$*,*

(ii) $\forall u \in [0,1]$ *esetén* $C(1, \ldots, 1, u, 1, \ldots, 1) = u$*,*

(iii) $\forall\ u_i \in [0,1]$ *esetén* $C(u_1, \ldots, u_d) \geq 0$ *d-növekvő.*

A kétdimenziós kopula valószínűségi sűrűségfüggvényét (PDF) és az eloszlásfüggvényét (CDF) a 3.1. ábra szemlélteti, 3.1. táblázatban ismertetett θ paraméterrel.



3.1. ábra. A kétdimenziós Gumbel-kopula PDF és CDF grafikonja (u_1, u_2 egyenletes eloszlású peremekkel és $\theta = 2$ paraméterrel)

Szemléletesen a kopula valójában egy többváltozós véletlen vektor eloszlásfüggvénye, amelynek az 1-dimenziós peremeloszlásai egyenletesek. A kétdimenziós esetben a 12. definícióban szereplő tulajdonságokat a 3.1. ábra jobb oldalán a színes vonalak szemléltetik. A sárga vonal az (i) tulajdonságot, a kék vonal az (ii) tulajdonságot szemlélteti. A (iii) tulajdonság azt fejezi ki, hogy a kopula alatti térfogat pozitív.

A következő, 1959-ben publikált tétel [98] a kopulaelmélet központi tételének tekinthető.

13. Tétel (Sklar tétel)**.** *Legyen F egy d-dimenziós eloszlásfüggvény egydimenziós $F_1, \ldots, F_d$ peremekkel. Ekkor létezik egy C kopulafüggvény*

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)) \tag{3.1}$$

minden valós $(x_1, \ldots, x_d)$ vektorra.
Ha a peremek folytonosak, akkor C egyértelmű.

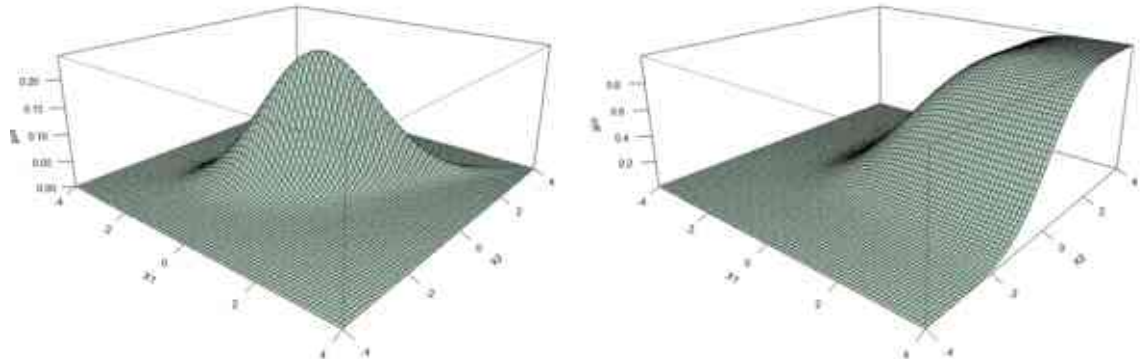
Ha feltesszük, hogy a (3.1) képletbeli C és F_i-k differenciálhatók, akkor az $f(x_1, .., x_d)$ valószínűségi sűrűségfüggvény a következőképpen fejezhető ki:

$$f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \cdot f(x_1) \cdot \ldots \cdot f(x_d). \tag{3.2}$$

Az $F_i(x_i) = u_i$ transzformáció segítségével megkapjuk a kopula $c(u_1, \ldots, u_d)$ sűrűségfüggvényét:

$$c(u_1, \ldots, u_d) = \frac{\partial^d C(u_1, \ldots, u_d)}{\partial u_1 \ldots \partial u_d}. \tag{3.3}$$

A kétváltozós normális peremekkel rendelkező Gumbel-kopula sűrűségfüggvénye (PDF) és eloszlásfüggvénye (CDF) látható a 3.2. ábrán.



3.2. ábra. A kétváltozós normális peremekkel rendelkező Gumbel-kopula PDF és CDF grafikonja ($\mu = 0,\ \sigma = 1$ normális eloszlású peremekkel és $\theta = 2$ paraméterrel)

A pénzügyi alkalmazásokban a kopula családok közül leginkább használt két kopula család az elliptikus kopulák és az archimédeszi kopulák családja [70, 16]. Az elliptikus kopulák többváltozós elliptikus eloszlásokból származnak, ezért ezek a kopulák sugárszimmetrikusak. A két legfontosabb kopula ebben a kopula családban a Gauss-kopula (vagy más néven normális kopula) és a Student-féle t-kopula. Szerkezetileg ez a két kopula közel van egymáshoz a középső részeiken és a széleiken (in the tails) pedig akkor vannak közelebb, amikor a t-kopula szabadsági fokainak száma növekszik.

Ezek a kopulák a következő képlettel írhatók le [61]:

$$C(u_1, \ldots, u_d) = F(G^{-1}(u_1), \ldots, G^{-1}(u_d)), \qquad (u_1, \ \ldots \ , u_d) \in [0,1]^d, \tag{3.4}$$

ahol Gauss-kopula esetén az $F = \Phi_R$ a többváltozós Gauss-eloszlás, amely megfelel egy R korrelációs mátrixnak, és $G^{-1} = \phi^{-1}$ az egyváltozós standard normális eloszlás inverz

függvénye; Student-kopula esetén $F = t_{R,\nu}$ a többváltozós Student-eloszlás, a $G^{-1} = {t_\nu}^{-1}$ pedig az egyváltozós standard t-eloszlás inverz függvénye ν szabadságfokkal.

Az elliptikus kopulák mellett az archimédeszi kopulák is nagyon népszerűek a többváltozós valószínűségi eloszlás modellezésében. A kétdimenziós archimédeszi kopulákat Genest és MacKay vezették be [35] és [36] cikkekben. Meghatározásukat ezután könnyen kiterjesztették a többváltozós esetre. Először az úgynevezett generátorfüggvényt kell definiálni.

14. Definíció. *A $\varphi : [0;1] \longrightarrow [0;\infty]$ függvényt generátorfüggvénynek nevezzük, ha rendelkezik a következő tulajdonságokkal:*

- *φ folytonos,*
- *φ szigorúan monoton csökkenő és $\varphi(1) = 0$,*
- *φ konvex a $(0;1)$ intervallumon.*

A φ generátorfüggvény általános definíciója birtokában a többváltozós archimédeszi kopula a következő módon definiálható:

15. Definíció. *A φ generátorfüggvénnyel definiált kopulákat*

$$C(u_1, \ldots, u_d) = \varphi^{-1}(\varphi(u_1), \ldots, \varphi(u_d))$$

archimédeszi kopuláknak nevezzük.

16. Megjegyzés. *Abban az esetben, ha a 14. definícióban a φ nem értelmezhető 0-ban, de fennáll, hogy $\lim_{t\to 0} \varphi(0) = \infty$, a φ függvény is generátorfüggvény szerepet játszhat a 15. definícióban, továbbá igaz, hogy a kopula szigorúan növekvő, kivéve azokban az esetekben, ha valamelyik változója egyenlő 0-val.*

Az archimédeszi kopula általában egy vagy két paramétertől függ, amelyeket a Kendall-τ alapján lehet kiszámítani. Nelsen [70] könyvében 22 archimédeszi kopula generátorfüggvényét ismerteti. A 3.1. táblázatban található a kutatásomban használt Clayton-, Gumbel- és Frank-kopulák generátorfüggvénye és tulajdonságaik.

3.1. táblázat. Az archimédeszi kopulák generátorai [70]

	$\varphi_\theta(t)$	θ	Tulajdonságok
Clayton	$\frac{1}{\theta}(t^{-\theta} - 1)$	$[-0, \infty) / \{0\}$	szig, ha $\theta \geq 0$
Gumbel	$(-\ln t)^\theta$	$[1, \infty)$	szig
Frank	$-\ln \frac{e^{-\theta t}-1}{e^{-\theta}-1}$	$(-\infty, \infty) / \{0\}$	szig, radszim

szig: szigorú; radszim: radiálisan szimmetrikus

A kopula generátora szigorú, ha $\varphi(0) = \infty$ [70]. Ezek a kopulák különböző típusú függőségeket írhatnak le. A Clayton-kopula az erősebb alsó farokfüggést, a Gumbel-kopula az erősebb felső farokfüggést, a Frank-kopula pedig az előbbi kettővel ellentétben szimmetrikus kopula, farokfüggés nélkül.

A 3.2. háromdimenziós ábrán szemléltettem a kétváltozós normális peremekkel rendelkező Gumbel-kopula sűrűségfüggvényét (PDF) és eloszlásfüggvényét (CDF).

Egy sokkal rugalmasabb kopula struktúra a vine-struktúra, melyet Joe [46] vezetett be. Fa struktúrával való megjelenítésük Bedford és Cook nevéhez fűződik [7]. Egy nagyon jó bevezetést és szemléltetést adnak Aas és munkatársai [1]-ben. A vine-kopulák bevezetésük óta nagyon elterjedtek, sok kutatás foglalkozik velük, mivel többfajta összefüggést tudnak egyidejűleg modellezni. Hátrányuk viszont az, hogy a magas dimenzióval nagyon megnő a paraméterek száma. Ily módon túltanuláshoz és rossz általánosításhoz vezethet. Ebben a részben csak röviden ismertetem az eredményeim megértéséhez szükséges fogalmakat.

A vine-kopulákat egy speciális fa szekvenciával írják le, amely éleihez hozzárendelik a párkopulákat, illetve a feltételes párkopulákat. Az alábbi 3.3. és 3.4. ábrán két speciális fára épülő vine-kopulára látunk 5 dimenziós példát. A C-vine-kopula esetén a fa csillag és a D-vine-kopula esetén pedig a fa egy lánc.

A vine-kopulát egy gráf struktúra határoz meg, amelyhez hozzá vannak rendelve párkopulák (két dimenziós kopulák) és feltételes pár-kopulák (feltétel előtt csak két változó van).

Kovács és Szántai [49] bevezettek egy alternatív definícót, amely cseresznyefákra épül, ezt fogom itt megadni, mivel sokkal egyszerűbb az eredeti [55] leírásnál.

Ehhez szükséges ismeret a cseresznyefa struktúra. Ennek megértéséhez szükségesek az alábbi fogalmak:

Klaszter gráf egy olyan gráfstruktúra (hipergráfként is fel lehet fogni), amely a merevkörű gráfok maximális klikkjeihez hozzá rendel egy-egy klasztert (élt). Két klaszter között akkor fut él, ha a metszetük nem üres halmaz.

Junction tree–t úgy határozzuk meg, hogy a klasztergráfban, amelyben az éleket az élekben lévő elemek számosságával súlyozzuk, megkeressük a maximális feszítőfát.

A **cseresznyefa** egy speciális junction tree, amelynek klaszterei k eleműek, élei $k-1$ eleműek.

A **vine-struktúra** cseresznyefák sorozataként áll elő.

Truncated vine-t úgy kapjuk meg, hogy a felső fákat elhagyjuk, feltételes függetlenséget feltéve.

A cseresznyefa reprezenációjára láthatunk egy példát a 3.5. ábrán.

17. Definíció. *A vine-kopula egy gráfstruktúrából áll, és az ehhez hozárendelt párkopulákból és feltételes párkopulákból. A vine-kopula gráfstruktúrája egy $T_1, T_2, \ldots, T_{d-1}$ cseresznyefa szekvenciából áll, amelyet a következő módon definiálunk.*

- *T_1 egy szokásos fa a $V = \{1, \ldots, d\}$, csúcsok halmazán*
  $E_1 = \{e_i^1 = (l_i, m_i)\ ,\ i = 1, \ldots, d-1\ l_i, m_i \in V\}$ az élhalmaz.
- *T_2 a másodrendű cseresznyefa a $V = \{1, \ldots, d\}$ halmazon*
  $E_2 = \{e_i^2, i = 1, \ldots, d-1 | e_i^2 = e_i^1\}$, $|e_i^1| = 2$, (a klaszterek 2 csúcsot tartalmaznak)
- *T_k egy k rendű cseresznyefa $V = \{1, \ldots, d\}$-n, amely klaszterhalmaza*
  $E_k = \left\{e_i^k, i = 1, \ldots, d-k+1\right\}$, ahol $\left|e_i^k\right| = k$ és úgy kaptuk meg, hogy vettük az előző T_{k-1} cseresznyefa összekötött klasztereinek az únióját $(k-1)$.

A fenti gráfstruktúrához hozzárendeljük a párkopulákat a következő módon:

A $c_{l_i,m_i}\left(F_{l_i}\left(x_{l_i}\right),F_{m_i}\left(x_{m_i}\right)\right)$ *kopula sűrűségfüggvényeket az első fa* T_1 *éleihez rendeltük hozzá.*

A

$$c_{a_{ij}^l,b_{ij}^l|S_{ij}^l}\left(F_{a_{ij}^l|S_{ij}^l}\left(x_{a_{ij}^l}|\mathbf{x}_{S_{ij}^l}\right),F_{b_{ij}^l|S_{ij}^l}\left(x_{b_{ij}^l}|\mathbf{x}_{S_{ij}^l}\right)\Big|\,\mathbf{x}_{S_{ij}^l}\right)$$

kopula sürűségfüggvények hozzárendelendők az e_i^l *és az* e_j^l*, a* T_l *cseresznyefában összekötött klaszterekhez, ahol*

$$\begin{aligned} S_{ij}^l &= e_i^l \cap e_j^l, \\ a_{ij}^l &= e_i^l - S_{ij}^l, \\ b_{ij}^l &= e_i^l - S_{ij}^l, \quad l=2,\ldots,d-1. \end{aligned} \tag{3.5}$$

Egyszerű belátni, hogy a_{ij}^l és b_{ij}^l, $l=2,\ldots,d-1$ egyetlen elemű halmazok.

A 17. definícióban szereplő jelölésekkel, most kijelenthető a tétel, amely megadja, hogyan írható le egy általános sűrűségfüggvény vine-kopula segítségével. A tétel eredeti [8] cikkben lévő változata most a cseresznyefás megfogalmazásnak megfelelően lett átírva.

18. Tétel. *Egy cseresznyefa szekvenciához hozzárendelt vine-kopula alapján az együttes sürüségfüggvény a következőképpen írható le:*

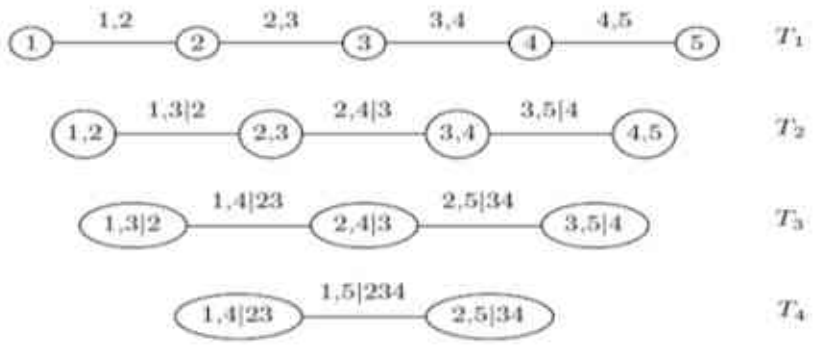
$$\begin{aligned} f\left(x_1,\ldots,x_d\right) = &\left[\prod_{i=1}^{d} f_i\left(x_i\right)\right]\left[\prod_{(l_im_i)\in E_1} c_{l_im_i}\left(F_{l_i}\left(x_{l_i}\right),F_{m_i}\left(x_{m_i}\right)\right)\right]\cdot \\ \cdot &\prod_{l=2}^{d-1}\prod_{e_i^l,e_j^l\in N(T_l)} c_{a_{i,j}^l,b_{i,j}^l|S_{ij}^l}\left(F_{a_{i,j}^l|S_{ij}^l}\left(x_{a_{i,j}^l}|\mathbf{x}_{S_{ij}^l}\right),F_{b_{i,j}^l|S_{ij}^l}\left(x_{b_{i,j}^l}|\mathbf{x}_{S_{ij}^l}\right)|\mathbf{x}_{S_{ij}^l}\right), \end{aligned} \tag{3.6}$$

ahol $e_i^l,e_j^l\in N\left(T_l\right)$ *azt jelöli, hogy* e_i^l,e_j^l *össze vannak kötve a* T_l *fában, és* $S_{ij}^l,a_{i,j}^l,b_{i,j}^l$*, a (3.5) képletek által vannak megadva és* $F_{a_{i,j}^l|S_{ij}^l}$ *a következőképpen számolható ki*

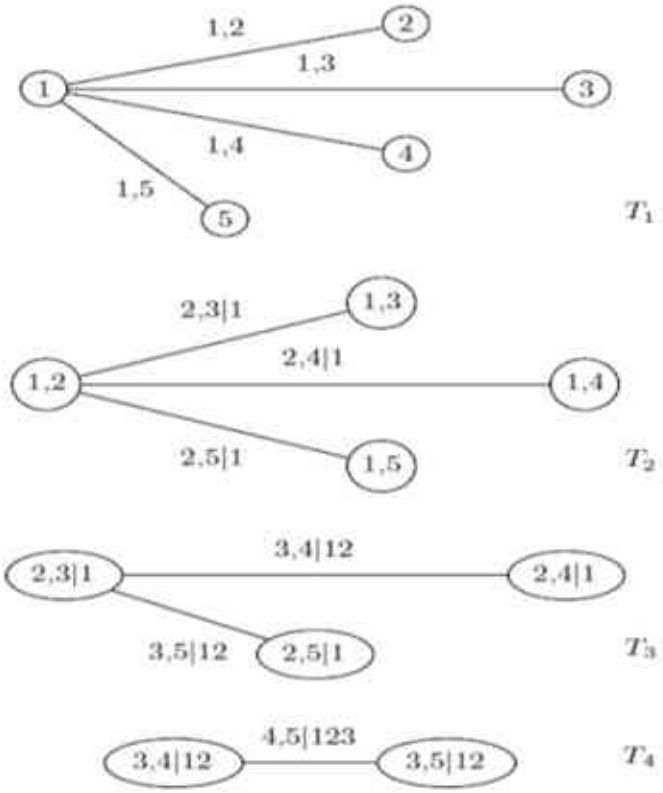
$$F_{j(e)|S_{ij}^l)}\left(x_{j(e)}|\mathbf{x}_{S_{ij}^l}\right)=\frac{\partial C_{i,j(e)|S_{ij}^l\setminus\{i\}}\left(u_i,u_j\right)}{\partial u_i}\Bigg|_{\substack{u_i=F_{i|S_{ij}^l\setminus\{i\}}\left(x_i|\mathbf{x}_{S_{ij}^l\setminus\{i\}}\right)\\ u_j=F_{j(e)|S_{ij}^l\setminus\{i\}}\left(x_{j(e)}|\mathbf{x}_{S_{ij}^l\setminus\{i\}}\right)}},$$

ahol $i\in S_{ij}^l$ *és* $j(e)=a_{i,j}^l$.

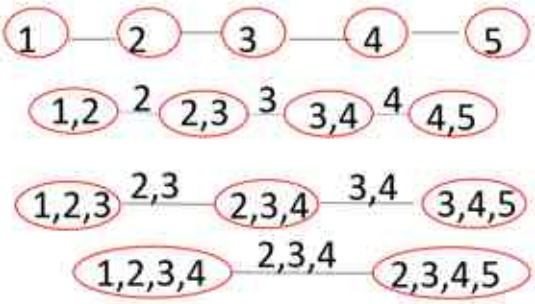
Fontos megjegyezni, hogy ez a felírás teszi lehetővé, hogy a gráf struktúrában, illetve a peremekben megjelenő párkopulák különböző összefüggésfajtákat is modellezhetnek. Ez adja meg a vine-kopulák rugalmasságát a modellezésben.



3.3. ábra. 5 dimenziós D-vine-kopula



3.4. ábra. 5 dimenziós C-vine-kopula



3.5. ábra. 5 dimenziós D-vine-kopula (Kovács Edith és Szántai Tamás cseresznyefa reprezentációjával)

3.1.2. Kockázati mutatók

A bizonytalanság minden jövőbeli megvalósult eseményben jelen van, ezért mivel biztosan nem tudjuk, csak becsülhetjük a jövőt. A becslések pedig kockázatokat rejtenek magukban. Ha a bizonytalanság valószínűségi eloszláson keresztül modellezhető, akkor a kockázat valószínűségek segítségével számszerűsíthető. Az emberek többnyire egyetértenek abban, hogy a várható hozamot tekintik egy portfólióteljesítmény mérőszámának, azonban nincs konszenzus abban, hogy melyik kockázati mérőszám ragadja meg jobban a kockázatot.

Markowitz [62] alapművében a variancia szerepel kockázati mérőszámként. Ez jó választás lehet, ha a valószínűségi eloszlás normális vagy legalább szimmetrikus.

Az alábbi fejezetekben a kockázat különféle fogalmait mutatom be, amelyek jelen dolgozatomban szerepelni fognak.

VaR és a CVaR

A kockáztatott érték (Value at Risk - VaR) nagyon népszerű kockázati mérőeszközzé vált, amióta a Bank of International Settlements és az USA szabályozó ügynökségei 1988-ban bevezették, és a Bázeli Bizottság is szabványként javasolta.

A kockáztatott érték (VaR) és a feltételes kockáztatott érték (CVaR) népszerű kockázati mérőszámokká váltak a portfólió kockázatkezelésében. A VaR-ral kapcsolatos fontos eredmények megtalálhatók Wipplinger és Jorion [122] közleményében, a CVaR-ral kapcsolatos eredményekről pedig Pflug [72] és Rockafellar és Uryasev [89, 90] munkáiban olvashatunk.

Először megadjuk a koherens kockázati mérték definícióját. Jelölje $\mathcal{R}$ a hozamok terét és feltesszük, hogy $\mathcal{R} = \mathcal{L}^2(\Omega, \mathcal{M}, P)$.

19. Definíció. *A koherens kockázati mérték egy* $\rho : \mathcal{R} \to [-\infty, +\infty]$ *leképezés, ami a következő tulajdonságoknak tesz eleget [4], [18]:*

Szubadditivitás: $\rho(R + R') \leq \rho(R) + \rho(R')$ *minden* $R, R' \in \mathcal{R}$.

Pozitív homogenitás: $\rho(\lambda R) = \lambda\rho(R)$ *minden* $R \in \mathcal{R}$ *és* $\lambda \geq 0$.

Monotonitás: $\rho(R) \leq \rho(R')$ *minden* $R, R' \in \mathcal{R},\ R \geq R'$.

Sallangmentesség [102] (translation equivariance): $\rho(R + \varrho) = \rho(R) + \varrho$ *minden* $R \in \mathcal{R},\ \varrho \in \mathbb{R}$.

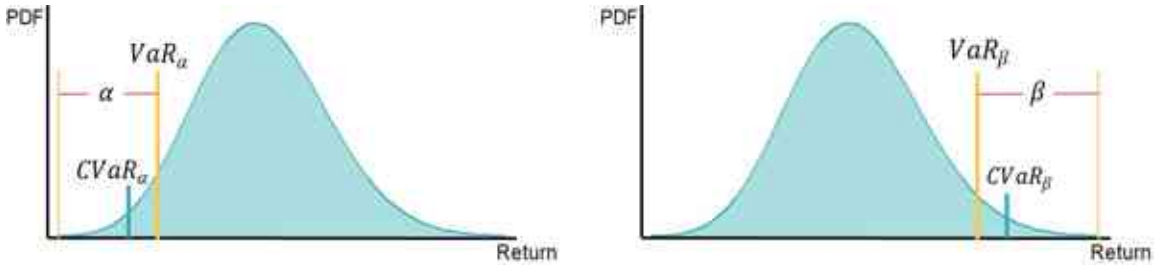
A VaR-t és a CVaR-t általában a veszteségeloszlással kapcsolják össze, azonban jelen vizsgálatban a veszteség vagy nyereség valószínűségi eloszlásának mindkét végén megvizsgáljuk értékeiket.

Valószínűségelméleti szempontból a VaR_α^L egy kvantilis, amely egy α^L valószínűségnek felel meg, míg a $CVaR_\alpha^L$ a valószínűségi eloszlás farkában számított feltételes várhatóérték.

Mivel nem csak a kockázat érdekel minket, hanem a profit is, ezért mindkét oldalon figyelembe vesszük a szélsőértékeket (3.6. ábra és 3.2. táblázat). Továbbá nem csak az alacsony hozam lehet kockázatos, hanem a magas hozam is, például a folyók vízmennyisége is terméshozamnak tekinthető, ekkor két folyó találkozásakor a túl magas hozamok árvizet okozhatnak. Ezért a kopulákat gyakran használják a hidrológiában is [69].

3.2. táblázat. A kockázati érték és a feltételes kockázat képletei

	Kockázat az alsó farokban	Kockázat a farok felső részén
VaR	$VaR_\alpha^L = F^{-1}(\alpha)$	$VaR_\beta^U = F^{-1}(1-\beta)$
CVaR	$CVaR_\alpha^L = \frac{1}{\alpha}\int_{-\infty}^{VaR_\alpha^L} tf\,(t)\,dt$	$CVaR_\beta^U = \frac{1}{\beta}\int_{VaR_\beta^U}^{\infty} tf\,(t)\,dt$



3.6. ábra. A kockáztatott érték (VaR) és a feltételes kockázati érték (CVaR)

A VaR-t két okból kritizálták. Először is, mivel a VaR nem egy koherens kockázati mérőszám, ahogyan azt Artzner és munkatársai [5] definiálták, nem konvex kockázati mérőszám, így sok lokális szélsőértéke lehet, ami technikai problémákat okozhat a portfólió optimalizálásakor.

Másodszor, egy százalékos aránynak megfelelő érték, és nem ad képet a lehetséges veszteségekről [101].

Egy másik intézkedés, amit javasoltak, a feltételes kockáztatott érték (Conditional Value at Risk - CVaR, a másik elnevezéssel Expected Shortfall - ES).

A CVaR az eloszlás farkának „szélességétől" függ. A CVaR áttekintése a Zhu és Fukushima [125] cikkében található. Rockafellar és Urysaev [109], [90] olyan minimalizálási formulát javasoltak, amely általában konvex vagy lineáris problémát eredményez.

A gyakorlatban azonban függetlenül attól, hogy milyen típusú kockázati mérőszámot alkalmaznak, a becslésének érvényessége az eszközhozamok együttes valószínűségi eloszlás becslésének jóságától függ.

Másodrendű sztochasztikus dominancia

Intuitívan akkor mondjuk, hogy egy X valószínűségi változó dominál másodrendű sztochasztikus dominanciával egy Y valószínűségi változót, ha X jobban megjósolható, mint Y. Jelöljön R, illetve R' két valószínűségi változót, amelyek várhatóértékei végesek. Először megadjuk a másodrendű sztochasztikus dominancia definícióját, ezt követően még 2 ekvivalens meghatározását is megadjuk.

20. Definíció. *Azt mondjuk, hogy R másodrendű sztochasztikus módon dominálja R' -t ($R \succ_{SSD} R'$), ha minden U nem csökkenő, (nem szigorúan) konkáv hasznossági függvényre*

$$E(U(R)) \geq E(U(R')) \tag{3.7}$$

teljesül.

A definícióban szereplő sztochasztikus dominancia fogalom másik két ekvivalens módon is megadható:

$$\mathrm{Tail}_\alpha(R) \ \geq \ \mathrm{Tail}_\alpha(R'), \tag{3.8}$$

ahol a $\mathrm{Tail}_\alpha(R)$ megfelel a $CVaR^L_\alpha(R)$-nek. Ez más szóval azt jelenti, hogy

$$CVaR^L_\alpha(R) \geq \ CVaR^L_\alpha(R'),$$

vagyis, ha az ábrát nézzük, akkor láthatjuk, hogy minél inkább jobbra toljuk a $CVaR^L_\alpha$ értékét, annál gyorsabban cseng le baloldalt az eloszlás, vagyis annál kevésbé kockázatos.

A fenti definíció a következő egyenlőtlenséggel is ekvivalens:

$$\mathrm{E}([t-R]_+) \ \leq \ \mathrm{E}([t-R']_+), \tag{3.9}$$

amely fenn áll minden $t \in \mathbb{R}$-re.

A hasznossági függvény konkávitása a kockázatkerülő magatartást jellemzi.

3.2. Az összefüggési struktúra hatása a VaR és a CVaR értékeire

Ebben az alfejezetben a függőségi struktúra hatását fogom vizsgálni két kockázati mérőszámra. Az itt következő eredményeim a [S6, S2] cikkekben jelentek meg, illetve [S18] konferencián hangzott el.

Megmutatom, hogy a VaR és a CVaR kockázati mutató értékeit hogyan befolyásolják a különböző típusú függőségi struktúrák. Ezeket az összefüggési struktúrákat különböző kopulákkal való modellezéssel mutatom be.

A függőségi struktúra VaR-ra és CVaR-ra gyakorolt hatását szimulált és valós adatokon is szemléltetem. A vizsgálatokat szimulált és valós portfólió kockázati adatokon végzem.

A szimulációkat R nyelven implementáltam, a peremek illesztéséhez a *fitdistrplus*, a kopulák illesztéséhez a *copula* és *rvinecopulib* R csomagokat használtam.

A Gauss-eloszlást általánosan alkalmazták a többváltozós valószínűségi eloszlás modellezésében, a kalibrálásban és szimulációban használt hatékony algoritmusok miatt. Az egyik nyilvánvaló hátránya a szimmetriája, ami azt jelenti, hogy a veszteségek valószínűsége megegyezik a nyereség valószínűségével. A tanulmányok azt sugallják, hogy a

mérőszámok erősebb együttes mozgást mutatnak válság idején [3], [45]. Az együttmozgás modellezésének egyik módja a keverteloszlások alkalmazása. A kopulákat Zhu és Fukushima [125] javasolta a robusztus "worst case" szcenárió modellezésére.

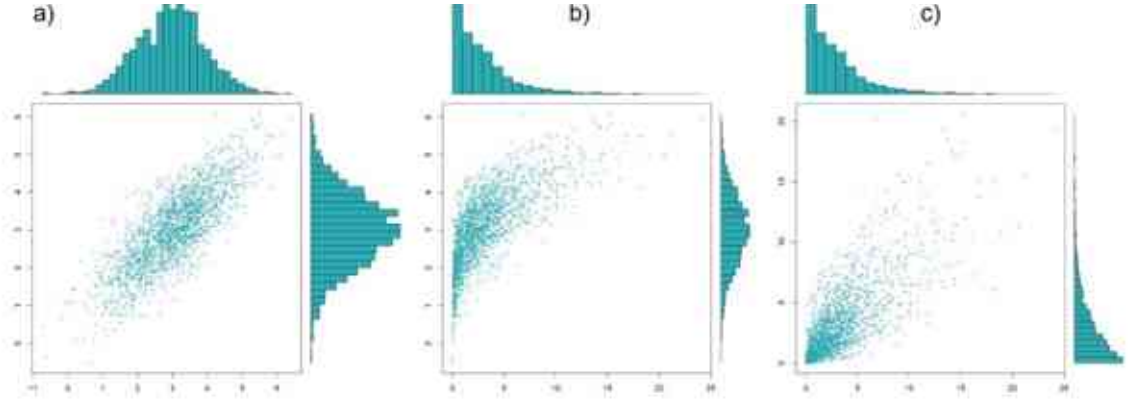
A kockázat számítása egy S véletlen változó egyváltozós valószínűségi eloszlása alapján történik, amely az összetevők hozamainak megfelelő, súlyozott véletlen változók összege. A kockázat kezelése nem csak a portfóliók esetében fontos, ugyanez a probléma megjelenik a két vagy több folyó által okozott árvizek esetében is. Számos tanulmány született a kapcsolódó témákban, a legtöbbjük kopulákat használ a jövőbeli adatok szimulációjában, és azzal foglalkozik, hogy pontos előrejelzést adjon [19]. A legtöbb írás idősorok esetében az együttes reziduum eloszlás modellezésére használ kopulákat [25, 2].

3.2.1. A Gauss-kopula előnyös tulajdonságai speciális eloszlások modellezésében

A 2008-as válságot követően, komoly támadások érték a matematikai modelleket. A támadások egyik fő célpontja a Gauss-kopula alkalmazása volt. Jelen rész célja egyrészt a Gauss-kopula néhány jó tulajdonságának bemutatása – amelyeket sokszor figyelmen kívül hagynak – elméleti szempontból, illetve szimulációk segítségével. Ezt a részt a Gauss-kopula egy rehabilitációjának szánom a [94] replikájaként.

Elsődlegesen néhány fontos elméleti előnyre mutatok rá. Egy együttes eloszlásban, például egy portfólióban, az értékpapírok különböző módon korellálnak egymással. Ha egy együttes normális eloszlást tételeznénk fel, abból azonnal következne, hogy a peremek is normális eloszlásúak kell legyenek, ami mint tudjuk elég szigorú feltétel. Ha azonban Gauss-kopulát használunk, a peremek bármilyen eloszlásúak lehetnek. Ezzel rávilágítottam, miért hasznosabb a Gauss-kopulával való modellezés az együttes Gauss-eloszlásnál. Kritikaként fölmerült, hogy nem modellezi jól a farkokban lévő összefüggéseket. Például az archimédeszi kopulák ezeket jól modellezik, mégsem lehetnek a Gauss-kopulának alternativái a magasabb dimenziókban. Az ok pedig a következő: az értékpapír párok különböző módon függnek egymástól, ezeket az összefüggéseket egy többdimenziós archimédeszi kopula paramétere nem tudja leírni, a Gauss-kopula (és a Student-kopula) a kovariancia mátrixon keresztül pedig igen.

Ebben az alfejezetben kétdimenziós együttes valószínűségi eloszlások három esetét mutatjuk be. Mindhármat Gauss-kopulák generálják, $\varrho = 0,8$ korrelációs együtthatóval. Az első esetben (3.7a. ábra) mindkét marginális valószínűségeloszlás Gauss, vagyis egy együttes normális eloszlásról van szó, a második esetben (3.7b. ábra) az egyik marginális valószínűségeloszlás exponenciális, a másik normális, a harmadik esetben pedig mindkettő exponenciális (3.7c. ábra).



3.7. ábra. Gauss-kopula ($\varrho = 0,8$) a) két normális ($\mu = 3,\ \sigma = 1$), b) normális ($\mu = 3,\ \sigma = 1$) és exponenciális ($\lambda = \frac{1}{3}$), c) két exponenciális ($\lambda = \frac{1}{3}$) peremekkel ($N = 2000$)

A 3.7b. ábrán megfigyelhető az együttes valószínűségi eloszlás aszimmetriája és az erős alsó farokfüggés. A 3.7c. ábrán a ferdeség és az erős alsó függőség is megfigyelhető. A 3.7. ábra a peremeloszlások szerepét mutatja be a Gauss-kopula esetében. Ezek a példák azt mutatják, hogy erős farokfüggés és aszimmetria is modellezhető a Gauss-kopulával, kihasználva azt a vonzó tulajdonságot, hogy tetszőleges peremeloszlások választhatók. Az alfejezet konklúziójaként kiemelem, hogy ebből az is látszik, hogy a peremeloszlások kombinálásának milyen fontos szerepe van a speciális farokösszefüggések modellezésében.

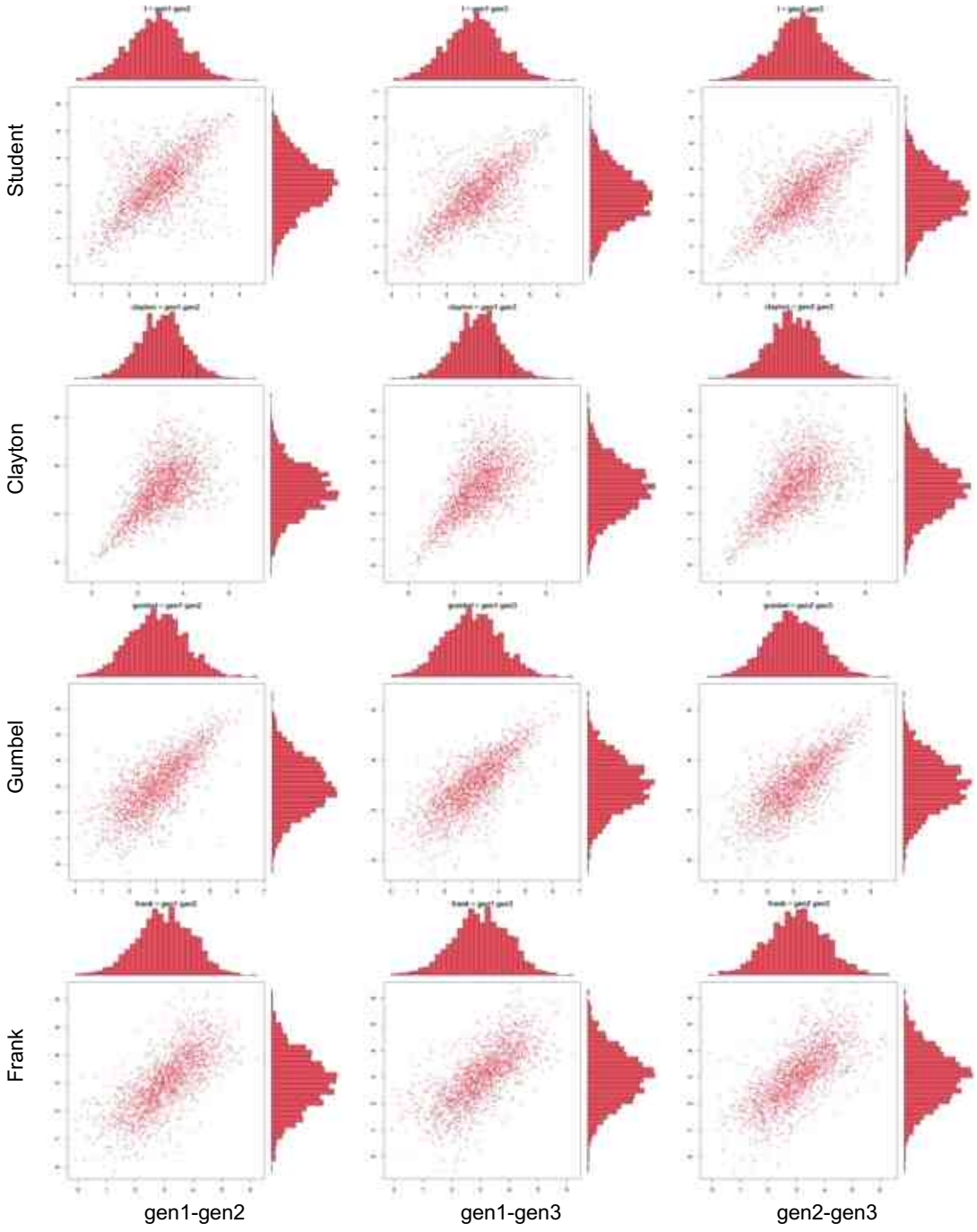
3.2.2. A kopula szerepe az együttes valószínűségi eloszlások modellezésében

Ebben a kísérletben a referencia adathalmazt egy 3-dimenziós normális kopulával generáltuk úgy, hogy először minden marginális értéket normális eloszlásúnak vettünk, majd egy exponenciális és a másik kettő normális eloszlásúval dolgoztunk. Ebben a kísérleti esetben ugyanazt a $\varrho = 0,8$ korrelációs együtthatót vettük az összes perem között.

Ezután először különböző kopulákat illesztettünk a referencia adatokra, megőrizve a normális peremeket. A különböző függőségi struktúrák modellezésére Student-, Clayton-, Gumbel-, Frank-kopulákat használtunk (3.8. ábra). Az egyes sorokban egy adott harmadrendű kopula (Student az első sorban, Clayton a második sorban, Gumbel a harmadik sorban és Frank az utolsó sorban) normális peremekkel szimulált pontjainak kétdimenziós vetületei láthatók. Minden sorban más-más kopula hatását figyelhetjük meg. Az első sorban a pontok a t-kopula miatt ellipszis alakúak, a második sorban a Clayton-kopula miatt alsó farokfüggés látható, a harmadik sorban a Gumbel-kopula miatt a felső farokfüggést látni, az utolsó sorban pedig a Frank-kopula által generált pontok, farokfüggés nélkül.

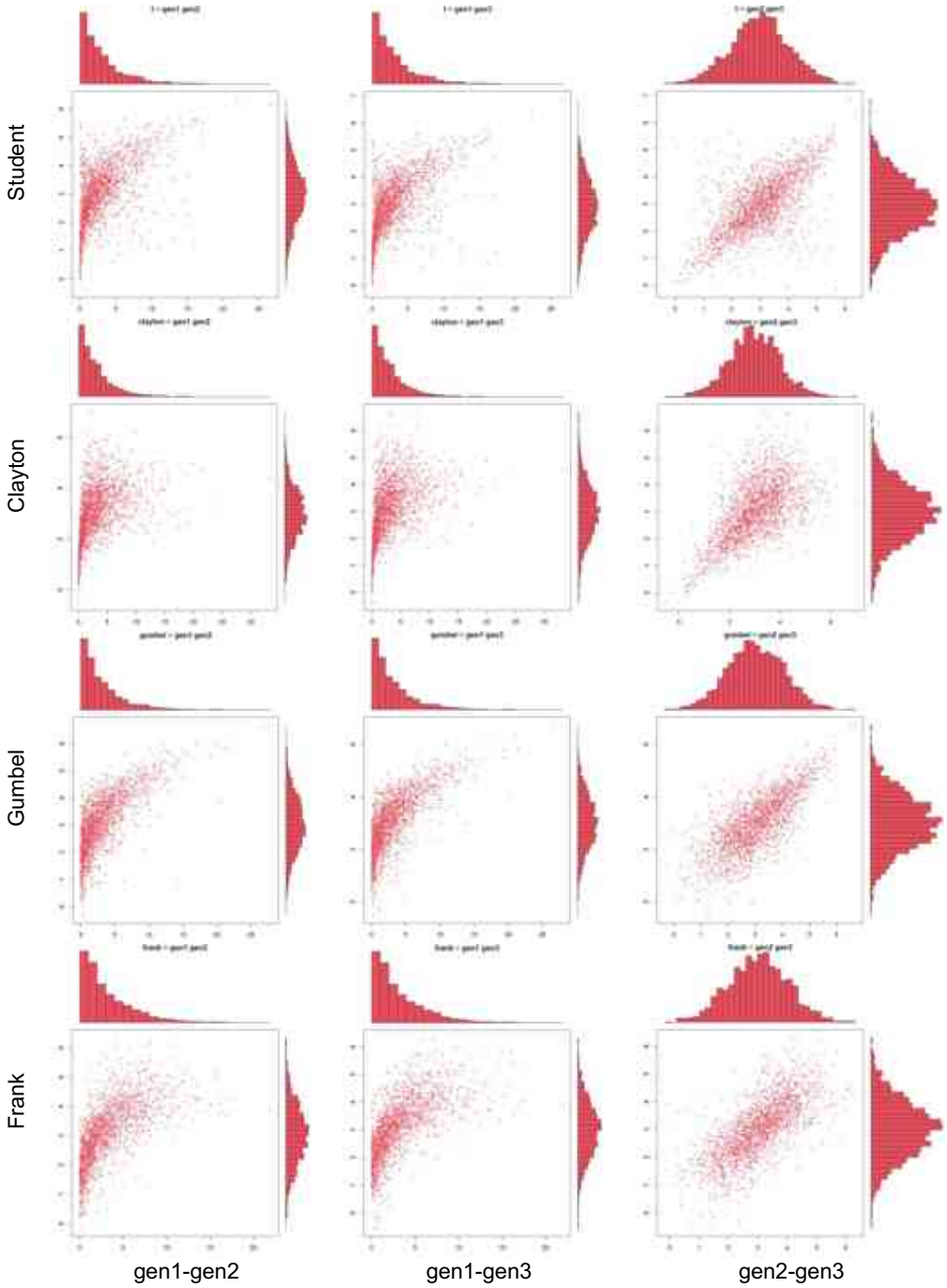
Másodjára egy exponenciális és két normális peremet választunk. A kétdimenziós peremeloszlásokat a 3.9. ábrán mutatjuk be, a kopula típusát és a peremeloszlások típusát is változtatjuk. Egy adott kopula esetében a kétváltozós vetületeket (peremeloszlások) ugyanabban a sorban szemléltetjük.

Általában a többváltozós normális vagy a t-kopulát meghatározó kovariancia- vagy korrelációs mátrix különböző értékeket tartalmazhat. Ez teszi a normális és a t-kopulát magasabb dimenziókban sokkal rugalmasabbá, a többváltozós archimédeszi kopuláknál.



3.8. ábra. A különböző függőségi struktúrák modellezése Student-, Clayton-, Gumbel-, Frank-kopulákkal, Gauss-kopulából ($\varrho = 0,8$) három normális ($\mu = 3, \quad \sigma = 1$) peremmel generált értékekre ($N = 2000$).

Ezért ez a normális és a t-kopula nagyon vonzó tulajdonsága. Ezt a tulajdonságot kihasználva egyszerre modellezhetünk olyan többváltozós eloszlást, amelynek összetevői között különböző típusú függőségek vannak. A korrelációs mátrixra vonatkozó egyetlen megkötés, hogy pozitív szemidefinit legyen.



3.9. ábra. A különböző függőségi struktúrák modellezése Student-, Clayton-, Gumbel-, Frank-kopulákkal, Gauss-kopulából ($\varrho = 0,8$) egy exponenciális ($\lambda = \frac{1}{3}$) és két normális ($\mu = 3,\ \sigma = 1$) peremmel generált értékekre (N=2000)

3.2.3. A függőségi struktúra hatása a kockázati mutatókra

Az előző két 3-dimenziós szimuláció adatait felhasználva kiszámítom a három perem összegéhez, mint portfólióhoz tartozó alsó és felső VaR és CVaR értékeket, $\alpha = 0,025$ és $\beta = 0,025$ értékekkel dolgoztunk. Az ott használt $\varrho = 0,8$ korrelációjú peremekkel generált referencia adathalmaz mellett még két $\varrho = 0,95$ és $0,1$ korrelációs együttható értékkel is elvégeztem a fenti szimulációt.

A 3.10. a. ábra oszlopait tekintve megfigyelhetjük a különböző kopulatípusok hatását a portfólióra, azonos marginálisok mellett. Megfigyelhető, hogy a Clayton-kopula esetében alacsonyabb VaR (UVaR) és CVaR (UCVaR), a Gumbel-kopula esetében magasabb VaR (LVar) és alacsonyabb CVaR (LCVaR) értéket kapunk. A 3.10. b. ábra azt is mutatja, hogy milyen különbséget jelent, ha különböző típusú peremekkel rendelkezik az adathalmaz. A Clayton-kopula és a Gumbel-kopula jellegzetes hatása sokkal jobban megfigyelhető. Gumbel erős felső függés a felső extremitásban, Clayton erős függés az alsó extremitásban.

Még az is jól látszik, ha a bal és jobb oldalt összehasonlítjuk, hogy a normális peremek sokkal kevésbé térnek el felfelé a várhatóértéktől, mint a nem normális peremek.

Egy másik érdekes kérdés, hogy ϱ értéke hogyan befolyásolja a kopulák hatásai közötti különbségeket. Ha ϱ értéke közel 1, akkor a véletlen változók között nagyon szoros függőség áll fenn, ez jobban kidomborítja a kopulák specifikumait (3.10. c. és d. ábra). Abban az esetben, ha ϱ értéke nagyon közel van a 0-hoz, minden kopula közelebb kerül a függetlenségi kopulához, így a szimulációs eredmények is egyre közelebb kerülnek egymáshoz. Az is érdekes, hogy a függetlenség (alacsony függőség) esetében az UVaR és az LVaR közötti intervallum szűkebb, ami teljesen összhangban van a portfóliódiverzifikáció céljával (3.10. e. és f. ábra), a portfólió kockázatának csökkentése érdekében.

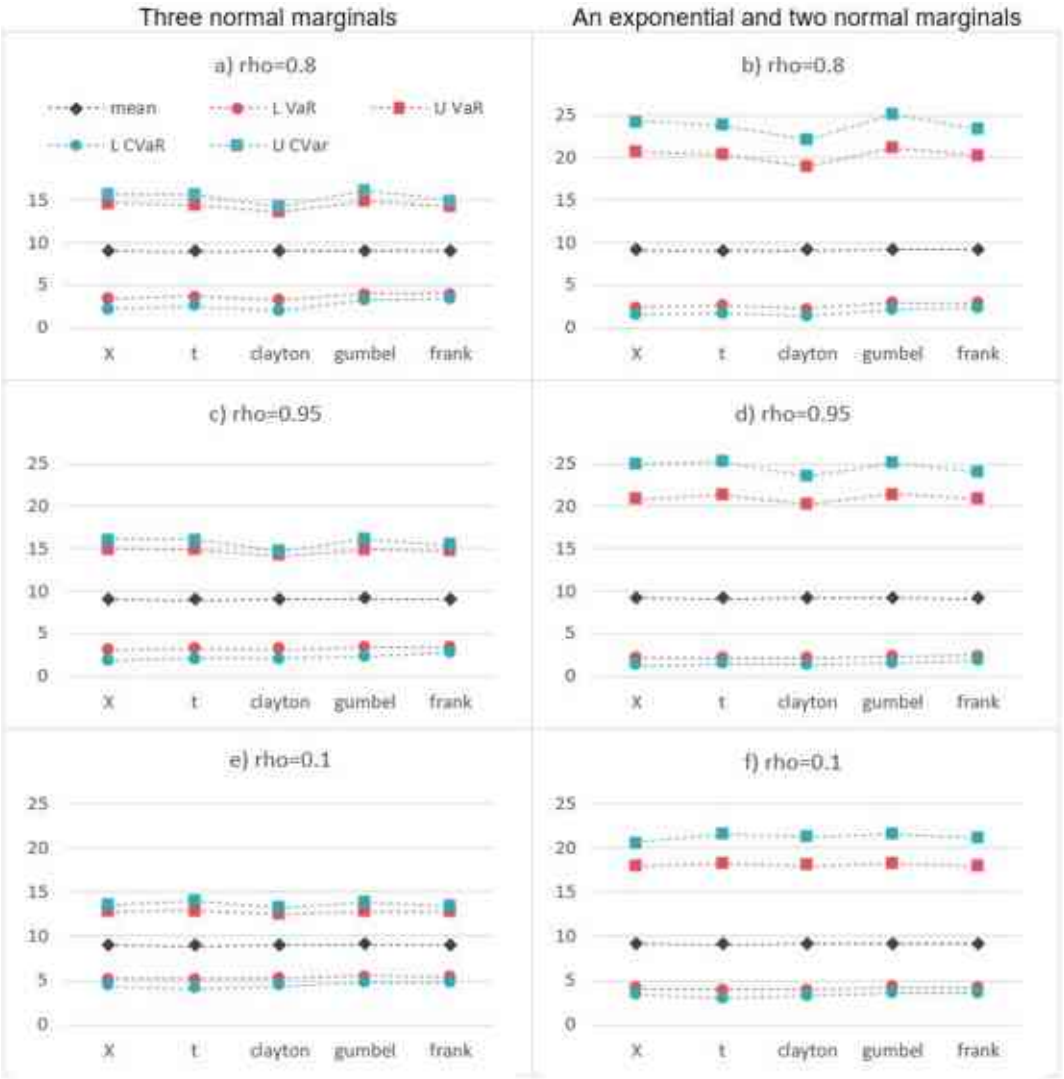
3.2.4. A összefüggési struktúra hatásának bemutatása valós pénzügyi adatokon

Az értékpapírok és a portfólió idősorkénti előrejezéséhez, több cikk is alkalmazta a kopulákat. Legtöbb esetben az idősor modellek illesztését követően, a reziduumok együttes ingadozásának modellezésében használták a kopula függvényeket [2, 25]. Azonban nagyon kevés cikk van, amely a loghozamok együttes eloszlását modellezi kopulákkal szcenáriógenerálási célból [67].

Ebben a részben a kopula által kódolt függőség kockázati mutatókra gyakorolt hatását tárgyalom és szemléltetem.

Az adatokat a Yahoo Finance-ból töltöttem le (https://finance.yahoo.com/). Két portfóliót vettem górcső alá. Először egy olyan portfóliót vettem figyelembe, amely három részvény 10 éven keresztül (2011.01.04-től 2020.12.31-ig) naponta regisztrált árfolyamaiból áll. A három részvény (Tesco PLC, Vodafone Group Plc, Rolls-Royce Holdings plc) az FTSE100-ból való. Az időhorizont $T = 2521$. A három idősor és összegük grafikonjait a 3.11. ábra szemlélteti.

A második portfólió 6 értékpapírt tartalmaz, minden részvényből egyet. Ezt az alfejezet második részében elemzem.



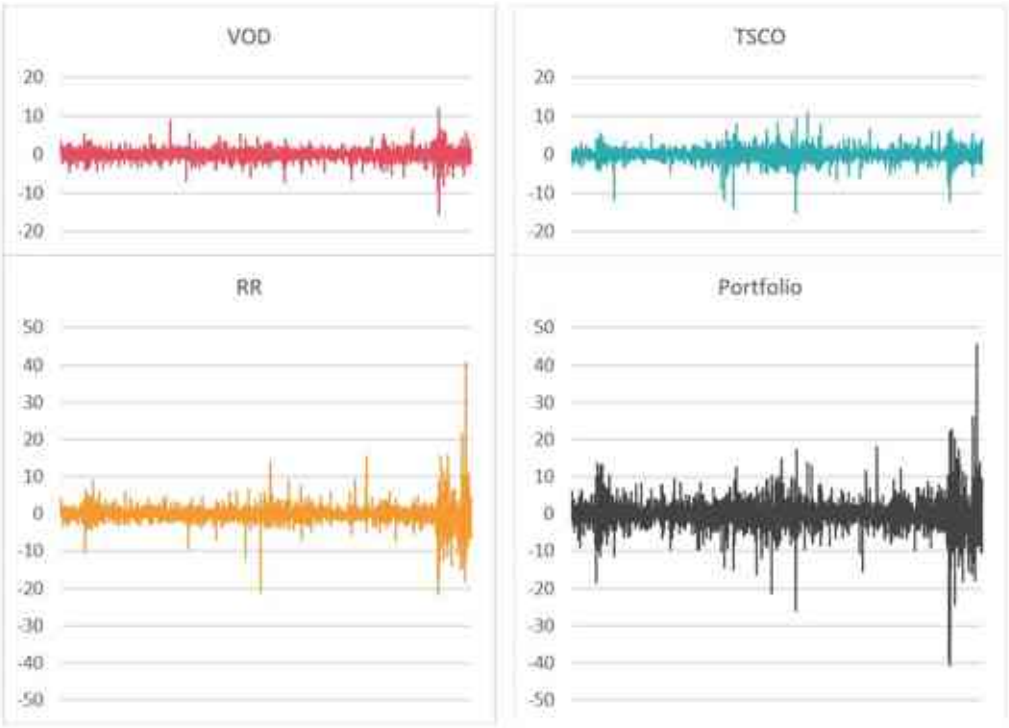
3.10. ábra. A portfólió VaR és CVaR értékei a különböző kopulák esetén

3.11. ábra. A részvények idősorai

Könnyen megfigyelhető, hogy ezek az idősorok először is azért nem stacionáriusak, mert tartalmaznak egy trendet. Ahhoz, hogy trend nélküli idősorokká alakítsuk át őket, a következő, logaritmikus hozam nevű transzformációt vezetjük be:

$$r_t = ln\frac{p_t}{p_{t-1}}.$$

A szakirodalomban gyakorlati szempontból sokszor a súlyokat $\frac{1}{m}$-mel helyettesítik, m részvény esetén. Ennek egyik oka, hogy a súlyoknak időben állandónak kell maradniuk, hogy képet kapjunk a portfólió kockázatosságáról. A következőkben az érintett részvények logaritmikus hozamainak összegével foglalkozunk (ld. Miskolci Panna [66]). Most a véletlen változók a három részvény logaritmikus hozamai, százalékban kifejezve. Ezeket az idősorokat a 3.12. ábra mutatja. A kockázatot felső és alsó VaR és felső és alsó CVaR értékekkel kívánjuk jellemezni, a különböző kopulacsaládok legjobban illeszkedő kopulái esetén.



3.12. ábra. A részvények és portfóliójuk hozamai

Az eljárásom a következő lépésekből áll.

21. Algoritmus. *Algoritmus a VaR és CVaR értékeinek kiszámítására*

1. *lépés: A napi árakat loghozamokra transzformálom.*

2. *lépés: Jó peremeloszlások illesztése. Az empirikus adatok egyváltozós peremeloszlásait illesztjük (loglikelihood maximalizálásával). Ezeket a valószínűségi eloszlásokat használjuk a kísérlet során.*

3. *lépés: A legjobb kopula illesztés egy adott kopulacsaládból. Az empirikus peremeloszlásokat egyenletes peremeloszlásokká alakítjuk. A d-dimenziós empirikus kopulát, amely egyenletes peremekkel rendelkezik, arra használjuk, hogy egy adott kopulacsaládból (normális, t, Clayton, Gumbel, Frank) a legjobb kopulát (loglikelihood maximalizálásával) illesszük rá. Ennek eredményeként megkapjuk azokat a paramétereket, amelyek az egyes családok legjobb kopuláját jellemzik.*

4. *lépés: Az egyenletes peremeket az 1. lépésben kapott folytonos peremekkel helyettesítjük. Így megkapjuk az együttes eloszlást.*

5. *lépés: Minden kopula-típusra (a 2. lépésben kiválasztott paraméterekkel) a 4. lépésben kapott együttes eloszlás alapján 10 000 d-dimenziós adatpontot szimulálunk.*

6. *lépés: Minden egyes adatpont egy loghozamokból álló vektort határoz meg, ami alapján kiszámítható a neki megfelelő loghozamösszeg. Így az egyenlő súlyozású loghozam portfóliót számoljuk ki.*

7. *lépés: Kiértékeljük a felső és alsó VaR-t és a CVaR-t.*

Az eredmények bemutatása előtt tegyünk néhány észrevételt. Ebben az esetben a normális és a Student-féle t-kopulának nem csak egy korrelációs együtthatója van, mint az előző részben, hanem ezek különböznek a mintának megfelelően. Mindegyik párnak más-más korrelációs együtthatója van. Ez ismét megmutatja a többváltozós Student- vagy Gauss-kopula rugalmasságát a magasabb dimenzióban is egyetlen paraméterrel (vagy kettővel, bizonyos esetekben hárommal 3.3. táblázat) rendelkező archimédeszi-kopulával szemben. Az összes egyváltozós peremhez a t-eloszlás adja a legjobb illeszkedést 3,40, 2,66 és 1,81 szabadsági fokkal.

Most bevezetek egy új közelítést, amit a feltételes függetlenség alapján határozok meg. Bemutatom ennek elméleti hátterét három változóra. Természetesen ezt lehet több változóra is általánosítani.

Tegyük fel, hogy három véletlen változóval rendelkezünk: X, Y, Z. A láncformula alkalmazásával a következőket kapjuk

$$f_{XYZ}(x,y,z) = f_X(x)\, f_{Y|X}(y|x)\, f_{Z|XY}(z|x,y).$$

Most tegyük fel, hogy $f_{XYZ}(x,y,z)$-t olyan valószínűségi eloszlással akarjuk közelíteni, amely azt kódolja, hogy X és Z függetlenek adott Y esetén. Ebben az esetben az utolsó tényező

$$f_{Z|XY}(z|x,y) = f_{Z|Y}(z|y)$$

tehát

$$f_{XYZ}(x,y,z) = f_X(x)\, f_{Y|X}(y|x)\, f_{Z|Y}(z|y),$$

ami a feltételes valószínűségi sűrűségek képleteinek alkalmazásával a következőképpen fejezhető ki:

$$f^{app}_{XYZ}(x,y,z) = \frac{f_{XY}(x,y)\cdot f_{YZ}(y,z)}{f_Y(y)}.$$

Most a (3.2) képlet segítségével az összes sűrűséget ki tudjuk fejezni kopulák segítségével:

$$c^{app}_{XYZ}(u,v,w) = c_{XY}(u,v)\cdot c_{YZ}(v,w),$$

ahol $u = F_X(x)$, $v = F_Y(y)$, $v = F_Z(z)$.

3.3. táblázat. Az illesztett kopulák paraméterei

kopula típus	paraméterek
CI	t(0,29;0,31)-t(6,56;5,22)
normális	rho(0,29;0,3;0,25)
t	rho(0,23;0,24;0,21) df(2)
clayton	alfa(0,33)
gumbel	alfa(1,2)
frank	alfa(1,65)

A három peremeloszláspárból csak kettőt fogunk használni. Azokat használjuk, amelyeknél a legmagasabb a Kendall érték, Chow és Liu ötlete nyomán [17] ez adja a legjobb illeszkedést az ilyen típusú közelítéseknél.

Ez a fajta kopula nagyobb rugalmassággal rendelkezik, mint a többváltozós archimédeszi kopulák. A feltételes kopula esetében az érintett kopulák egyszerre különböző típusú nemlineáris függőségekkel rendelkezhetnek.

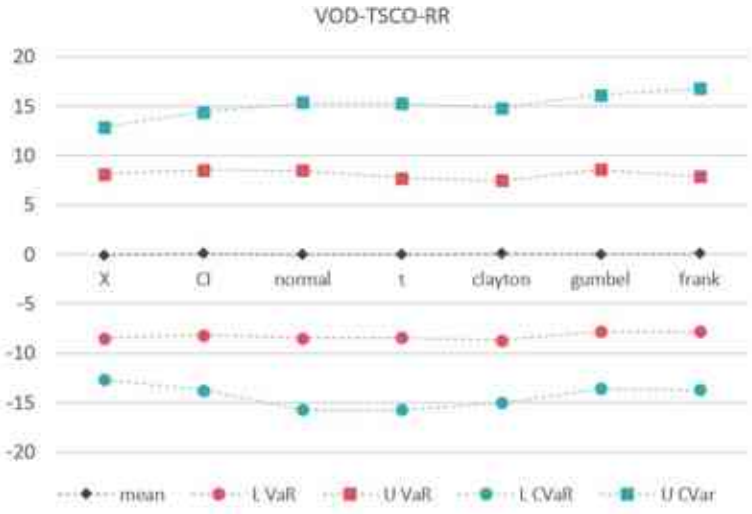
A többváltozós Gauss-kopula és a t-kopula is rugalmas, mert egyszerre különböző típusú függőségeket tesz lehetővé. Ezek a függőségek azonban korrelációra épülő függőségek. A feltételes függetlenséget kihasználva a modellezésben akár két különböző családot is használhatunk (háromváltozós esetben), aminek köszönhetően aszimmetrikus összefüggések is megjelenhetnek, illetve széles farkú összefüggések is.

Most áttérek az adott három változó modellezésére. A három kétdimenziós peremeloszlás közül csak kettőt akarunk használni. Azokat használjuk, amelyeknek a Kendall értéke a legmagasabb (3.4. táblázat). Ennek megfelelően a modellben csak két kétdimenziós kopulát fogunk használni. Ez egy olyan közelítő eloszlás lesz, amelybe belekódoltunk egy feltételes függetlenséget azok között a változók között, amelyek nem szerepelnek egy párban, feltéve arra, amely mindkét párban szerepel. Ez a típusú kopula nagyobb rugalmassággal rendelkezik, mint a többváltozós archimédeszi kopulák. A feltételes függetlenséget tartalmazó kopula esetében az érintett kopulák egyszerre különböző típusú nem okvetlenül szimmetrikus függőségekkel rendelkezhetnek. Ha csak párkopulákat használunk, akkor be lehet bizonyítani, hogy a legjobb közelítést a maximális feszítőfa alapján kapjuk meg. (A teljes gráf éleit, amiben keressük a feszítőfát, az összefüggés erőssége súlyozza.)

3.4. táblázat. A részvények Kendall-τ matrixa

	VOD	TSCO	RR
VOD	1	0,188	0,201
TSCO	0,188	1	0,168
RR	0,201	0,168	1

A különböző típusú kopulákkal modellezett portfóliók kockázati mérőszámai közötti különbségeket a 3.13. ábra mutatja be. A feltételes függetlenséget tartalmazó kopulát CI-kopulaként jelöljük.



	X	CI	normal	t	clayton	gumbel	frank
mean	-0.07	0.07	-0.03	-0.04	0.08	0.04	0.10
L VaR	-8.46	-8.20	-8.46	-8.45	-8.66	-7.83	-7.79
U VaR	8.12	8.55	8.50	7.75	7.47	8.61	7.86
L CVaR	-12.70	-13.75	-15.73	-15.72	-15.03	-13.60	-13.71
U CVaR	12.90	14.41	15.37	15.26	14.83	16.13	16.81

3.13. ábra. A portfólió VaR és CVaR értékei (X az eredeti adathalmaz)

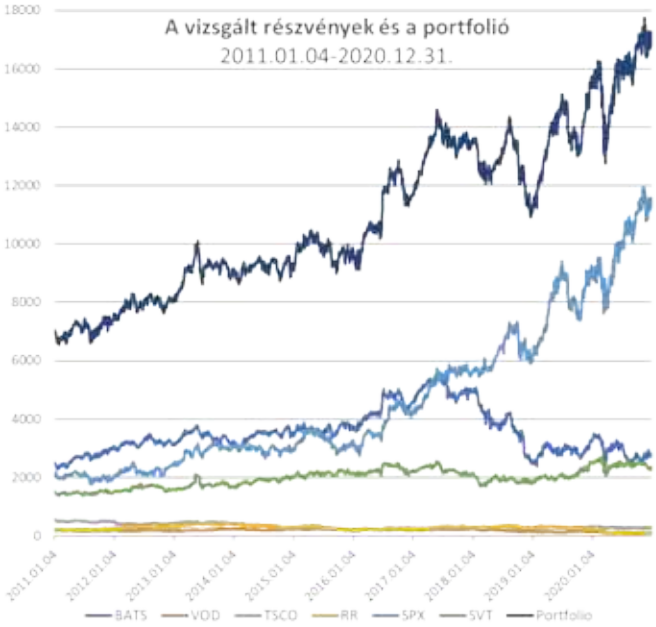
Mindegyik családon belül kiválasztottuk a legjobban illeszkedő kopulát a loglikelihood maximalizálása alapján. Ezek a loglikelihood értékek láthatók a 3.5. táblázatban.

Ebből a táblázatból leolvasható, hogy a legjobban illeszkedő kopula az, amelyik feltételes függetlenséget tartalmaz. Ennek magyarázata az, hogy a különböző érintett párok különböző típusú függőséggel rendelkeznek, t(0,29;0,31)-t(6,56;5,22), ami nem modellezhető korrelációval.

3.5. táblázat. A kopulaillesztés log-likelihoodjai a legjobb kopula paramétereivel kiértékelve

	CI	normal	t	clayton	gumbel	frank
loglik	290,83	268,63	90,29	223,33	257,88	242,44

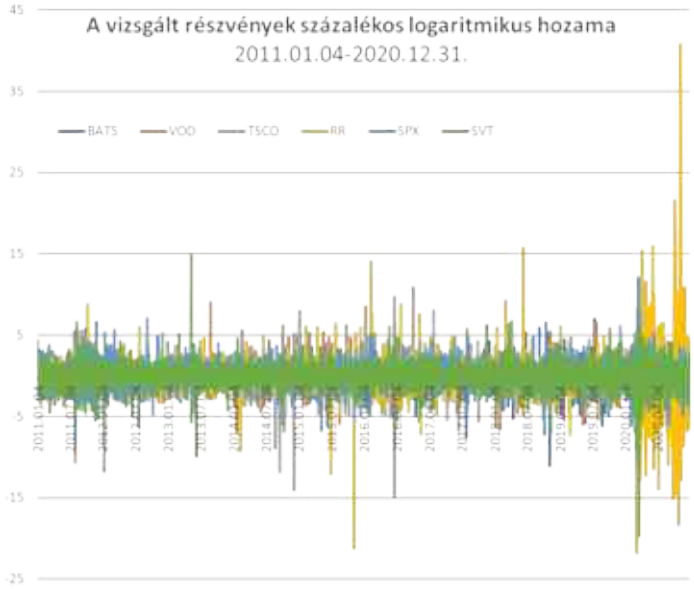
Most áttérünk egy nagyobb portfólió kopulákkal való modellezésére. A portfólió ezúttal 6 értékpapírt tartalmaz. Az értékpapírok idősorai és az egyenlő súlyokkal súlyozott portfólió a 3.14. ábrán látható. Az alsó 3 részvény kinagyítva a 3.15. ábrán látható.



3.14. ábra. Részvények és a portfólió idősorai

3.15. ábra. A 3 legkisebb részvény idősorai

Napi megfigyelésekről van ismét szó, amelyeknek a loghozamát nézzük. A loghozamokat egyben a 3.16. ábrán és részvényenként a 3.17. ábrán láthatjuk.

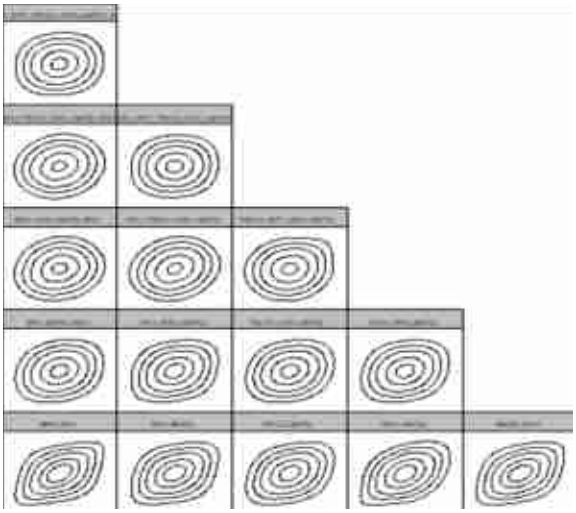


3.16. ábra. Az értékpapírok loghozamai egyben

3.17. ábra. Az értékpapírok loghozamai

Ebben a kísérletben is ugyanazokat a kockázati mutatókat vizsgálom, mint a 3-dimenziós esetben. Most is a 21. algoritmussal dolgozok, azzal a különbséggel, hogy előzőkben használt kopulákhoz hozzáveszem a vine-kopula struktúrát is. Az eredményeket ismertettem a MOK Konferencián [S18].

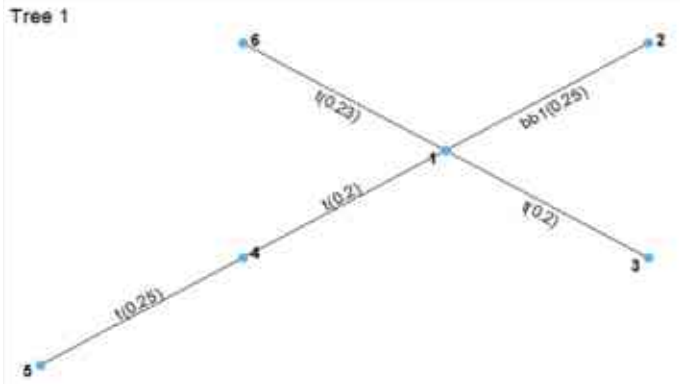
A párkopulák szintvonalai a 3.18. ábrán láthatók.



3.18. ábra. Párkopulák szintvonalai

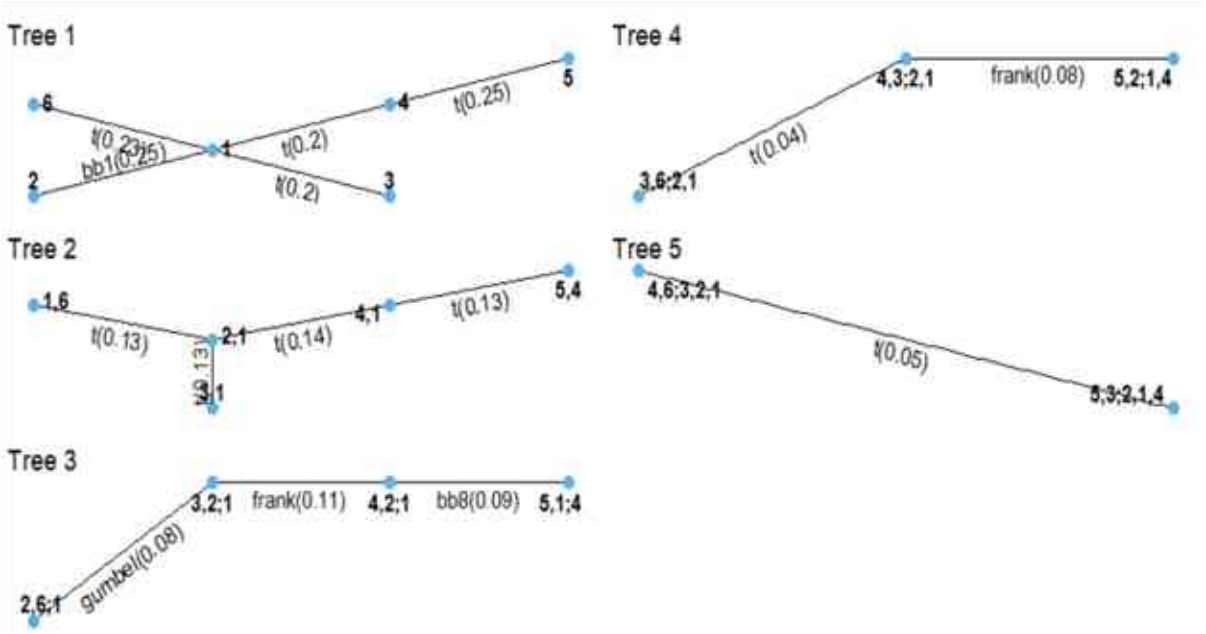
Az egyik közelítést, az első fa struktúra adja meg. Az alapötlete, hogy olyan kopulával közelítünk, mely a legösszefüggőbb két-dimenziós peremek szorzatát tartalmazza, amelyek egy feszítőfát határoznak meg.

A fa, amely élein láthatók a megfelelő párra illesztett kopulák, a 3.19. ábrán látható.



3.19. ábra. A vine-kopula első fája, ami alapján keletkezett a truncVineL1

Illesztettem az adatokra teljes vine-kopulát is, amely az összes fát tartalmazza. Ennek a kopulának viszonylag magas, 15 a paramétereinek száma, mivel most minden élen egy paramétertől függő kopula van. A teljes fastruktúra az illesztett párkopulák adataival a 3.20. ábrán látható (a bb1 és bb8 is egyfajta kopula).



3.20. ábra. A teljes vine-kopula az összes fával és az összes éllel

A loglikelihoodokat tartalmazó táblázatból 3.21. látható, hogy az összes közül a VineAll (ami a teljes vine-kopulát jelöli) illeszkedik a legjobban. Ez érthető, ha azt is figyelembe vesszük, mennyi paraméter meghatározásától függ. Második legjobban illeszkedő a normális kopula, ebből is jól látszik a hatékonysága, ami annak köszönhető, hogy a kovariancia mátrix sok paramétert tartalmaz. A harmadik legjobban illeszkedő a VineCopulaTruncL1, ami az egy fa alapján kapott kopula szorzatot jelenti. A paraméterek száma ebben az esetben egyenlő a változók száma mínusz egy, ami sokkal kevesebb paramétert jelent, mint a teljes vine-kopulánál. Ebben az esetben 5. Habár a teljes vine-kopula illeszkedik a legjobban én mégis a normális kopulát vagy a fa alapú kopulát javaslom modellezésre.

truncVineL1	vineAll	normal	t	clayton	gumbel	frank
906,362	**1294,158**	1080,705	191,182	839,350	838,545	810,514

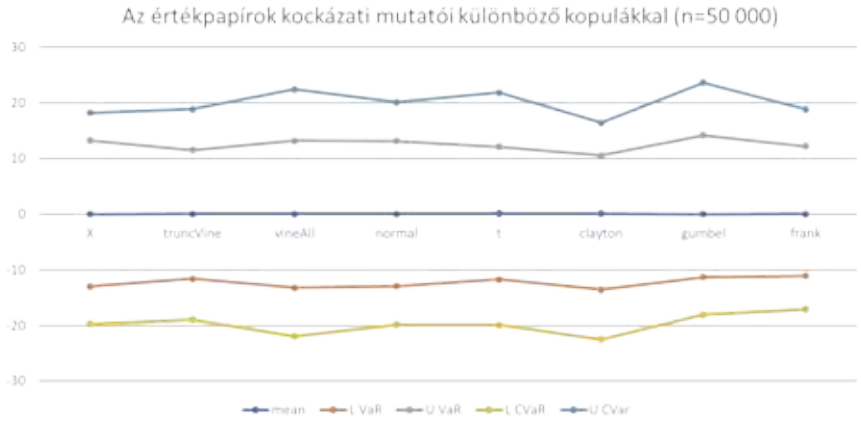
3.21. ábra. Az illesztett kopulák loglikelihood értékei

Továbbiakban azzal foglalkozom, hogyan hatnak ezek az összefüggési struktúrák a kockázati mutatókra.

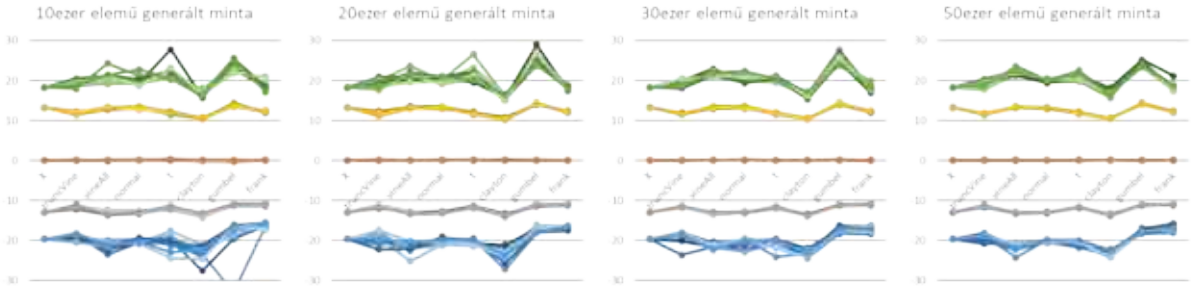
A 3.22. táblázatban láthatók a kockázati mértékek számszerűen, illetve a 3.23. ábrán illusztrálva $n = 50000$ mintanagyság esetén. A kísérleteket elvégeztem különböző mintanagyságra ($n = 10; 20; 30; 40$ és 50 ezer) és minden mintanagyság esetén 10 futtatást végeztem. A generált minták VaR és CVaR értékei különböző mintanagyságok esetén a 3.24. ábrán láthatók, valamint a 10 futtatás átlagai a 3.25. ábrán. Jól látszik, hogy a mintanagyság a modell robusztusságához vezet egyrészt, másrészt pedig ezekből az ábrákból a kopulák specifikumai is leolvashatók: Clayton- erős alsó összefüggés, Gumbel- erős felső összefüggés.

	X	truncVine	vineAll	normal	t	clayton	gumbel	frank
mean	0.02106698	0.06849757	**0.06536403**	0.09293175	0.12486112	0.11186184	0.01845675	0.08589353
L VaR	-12.967275	-11.603458	**-13.174997**	-12.916322	-11.702098	-13.521835	-11.274115	-11.047528
U VaR	13.2832615	11.5682318	**13.1918598**	13.1721855	12.1253014	10.5545869	14.2221863	12.2114352
L CVaR	-19.699669	-18.912081	**-21.876136**	-19.823838	-19.909325	-22.474796	-18.019131	-17.042969
U CVar	18.2376401	18.8835466	**22.4650981**	20.1518874	21.8738729	16.4593402	23.6771339	18.856271

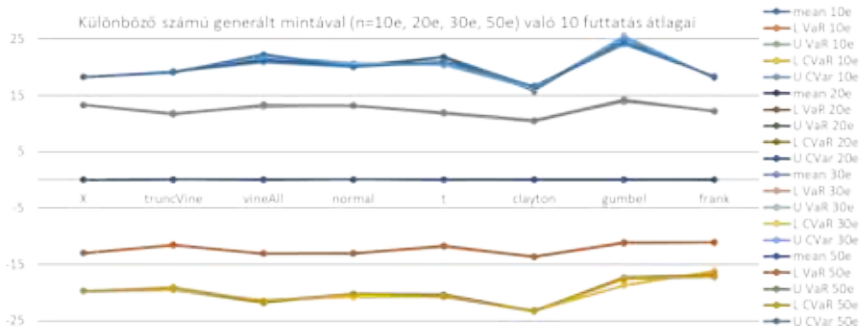
3.22. ábra. A kockázati mutatók értékei ($n = 50000$)



3.23. ábra. A kockázati mutatók különböző kopulák esetén ($n = 50000$)



3.24. ábra. Kockázati mutatók különböző mintanagyságok esetén (10 kísérlet)



3.25. ábra. Kockázati mutatók különböző mintanagyságok esetén (10 kísérlet átlagai)

3.3. Portfólió optimalizálás másodrendű sztochasztikus dominancia mellett

Ebben alfejezetben egyrészt egy sztochasztikus optimalizálási feladat megoldása lesz a portfólió súlyozása, másrészt pedig a portfólió optimális kiválasztását adatgenerálással támogattam meg. A harmadik érdekesség, hogy ebben az esetben egy 68 értékpapírból álló portfóliót kell összeállítani. A feladatom volt a minta alapján szcenáriók generálásával bemeneti adatot adni a portfólió optimalizálási feladatnak.

Először felvezetem a portfólió optimalizáláshoz tartozó sztochasztikus optimalizálási problémát, annak feltételrendszerét, és röviden vázolom az optimalizálási eljárásokat, amelyeket felhasználunk. A feltételrendszer a másodrendű sztochasztikus dominancia fogalommal lesz megadva. Ezen keresztül vizsgáljuk a portfólió kockázatát. A fejezet második része mutatja be az általam végzett eljárásokat. Azt vizsgáltam, hogy különböző szcenárió generálások hogyan befolyásolják az optimalizálás megoldását, vagyis a portfólió kialakítását és annak tulajdonságait. A fejezet az [S1] könyvfejezetre és az [S13, S14, S15, S16] konferencia előadásokra támaszkodik.

3.3.1. Előzmények, optimalizálási eljárások

A portfólió optimalizálást erősen befolyásolja az értékpapírok együttes eloszlásának a hiányos ismerete. Tegyük fel, hogy van n darab értékpapírunk, amelyekbe be szeretnénk fektetni egy egységnyi összeget egy adott fix időtartamra. A portfóliót egy n dimenziós vektor határozza meg,

$$\boldsymbol{x} = (x_1, \dots x_n) \in \mathbb{R}^n,$$

ahol az i. pozícióban lévő elem megmutatja, hogy az összeg hanyad részét fektettük az i. értékpapírba. Feltesszük, hogy ezeknek összege 1. A megengedett portfóliók halmaza X. A portfólió hozama egy valószínűségi változó, amely a következő módon függ az értékpapírok véletlen hozamától:

$$R_{\boldsymbol{x}} := x_1 R_1 + \ldots + x_n R_n.$$

Az alábbi fogalmak a másodrendű sztochasztikus dominancia fogalmát használják, amelyet a 20. definícióban adtunk meg.

22. Definíció. *Azt mondjuk, hogy egy $\boldsymbol{x}^\star$ portfólió SSD-efficiens, ha nem létezik másik megengedett portfólió, amely ezt dominálná SSD módon, vagyis nem létezik $\boldsymbol{x} \in X$, amelyre a következő áll fenn:*

$$R_{\boldsymbol{x}} \succ_{SSD} R_{\boldsymbol{x}^\star}.$$

A következőben áttekintjük az ehhez kapcsolódó optimalizálási feladatokat. Tegyük fel, hogy rendelkezésünkre áll egy $\widehat{R}$, hozam, amely ismert diszkrét eloszlással rendelkezik; $\widehat{R}$ lehet például egy értékpapír hozama, vagy egy benchmark portfólió hozama.

Dentcheva és Ruszczyński [24]-ben megfogalmazták az SSD feltétel alapú portfólió

optimalizálási modellt:

$$\max\ f(\boldsymbol{x})$$

$$\text{feltéve hogy} \quad \boldsymbol{x} \in X, \tag{3.10}$$

$$R_{\boldsymbol{x}} \succeq_{SSD} \widehat{R},$$

ahol f egy a konkáv függvényt jelöl.

Speciálisan legyen $f(\boldsymbol{x}) = \mathrm{E}(R_{\boldsymbol{x}})$. Az optimalizálási feladathoz, a szerzők a 2. ekvivalens felírást (3.9) alakban alkalmazták és bebizonyították, hogy véges diszkrét eloszlások esetén a dominanciát tartalmazó feltétel felírható egy véges egyenletrendszerrel.

Roman, Darby-Dowman és Mitra a [91]-ben a (3.8) képlet másodrendű sztochasztikus dominancia felírását használták az optimalizálási eljárásban. Véges diszkrét eloszlások feltételezéséből indultak ki, amelynek a realizációi egyenlő valószínűséggel következnek be. Ennek a feltételezésnek megfelelnek a folytonos eloszlásból vett független mintaelemek.

Az SSD relációt visszavezették egy véges egyenlőtlenségrendszerre.

$$\mathrm{Tail}_{\frac{i}{S}}\left(R_{\boldsymbol{x}}\right) \ \geq\ \mathrm{Tail}_{\frac{i}{S}}\left(\widehat{R}\right) \quad (i = 1, \ldots, S), \tag{3.11}$$

ahol S a (egyenlő valószínűségű) szcenáriók száma.

A következő megközelítés a (3.8) egyenlőtlenség felírásból következik.

Az alapötlet az, hogy az $\boldsymbol{x}$ portfólió, amelynek a várható hozama $R_{\boldsymbol{x}}$, minél közelebb kerüljön a referencia portfólió hozamához, sőt emulálja azt egyenletes értelemben. Ezt egy többszempontú optimalizálási feladatnak tekinthetjük, amely Pareto-optimális megoldásai SSD-efficiens portfóliók lesznek. Jelöljük a referencia portfólió hasznát $\widehat{R}$-rel. A farkokban lévő különbségek segítségével a következőképpen értendő. Legyen a "legrosszabb farok különbség" az összes szcenárión:

$$\vartheta = \min_{i=1\ldots S}\left(\mathrm{Tail}_{\frac{i}{S}}\left(R_{\boldsymbol{x}}\right) - \mathrm{Tail}_{\frac{i}{S}}\left(\widehat{R}\right)\right)$$

A feladatunk a következő lesz:

$$\max\ \vartheta$$

$$\text{feltéve, hogy} \quad \vartheta \in \mathbb{R}, \quad \boldsymbol{x} \in X, \tag{3.12}$$

$$\mathrm{Tail}_{\frac{i}{S}}\left(R_{\boldsymbol{x}}\right) \ \geq\ \mathrm{Tail}_{\frac{i}{S}}\left(\widehat{R}\right) + \vartheta \qquad (i = 1, \ldots, S).$$

Továbblépve Fábián, Mitra, Roman és Zverovich a [32]-ben a következő optimalizálási feladatot ajánlották:

$$\max\ \vartheta$$

$$\text{feltéve, hogy} \quad \vartheta \in \mathbb{R}, \quad \boldsymbol{x} \in X, \tag{3.13}$$

$$R_{\boldsymbol{x}} \ \succeq_{SSD} \widehat{R} + \vartheta.$$

Az $R_{\boldsymbol{x}} \succeq_{SSD} \widehat{R}+\vartheta$ azt jelenti, hogy az $R_{\boldsymbol{x}}$ nem csak az $\widehat{R}$ referencia hozamát dominálja, hanem emellett még egy kockázatmentes befektetés ϑ hozamával is.

Feltéve, hogy S egyenlő valószínűségű realizációt tekintünk, az

$$R_{\boldsymbol{x}} \succeq_{SSD} \widehat{R} + \vartheta$$

feltétel átfogalmazható a következőképpen:

$$\mathrm{Tail}_{\frac{i}{S}}\big(R_{\boldsymbol{x}}\big) \;\geq\; \mathrm{Tail}_{\frac{i}{S}}\big(\widehat{R} + \vartheta\big) \quad (i = 1, \ldots, S).$$

Mivel

$$\mathrm{Tail}_{\frac{i}{S}}\big(\widehat{R} + \vartheta\big) = \mathrm{Tail}_{\frac{i}{S}}\big(\widehat{R}\big) + \frac{i}{S}\,\vartheta \quad (i = 1, \ldots, S)$$

a (3.13) feladat ekvivalens lesz a következővel:

$$\begin{aligned} &\max\ \vartheta \\ &\text{feltéve, hogy} \quad \vartheta \in \mathbb{R}, \quad \boldsymbol{x} \in X, \\ &\mathrm{Tail}_{\frac{i}{S}}\big(R_{\boldsymbol{x}}\big) \;\geq\; \mathrm{Tail}_{\frac{i}{S}}\big(\widehat{R}\big) + \tfrac{i}{S}\,\vartheta \qquad (i = 1, \ldots, S). \end{aligned} \tag{3.14}$$

Az különbség a (3.12) optimalizálási feladat és a (3.14) modell között, hogy az utóbbiban skálázva vannak a farkak.

A fejezet további részében ezekre a modellekre a következőképpen hivatkozunk majd. A (3.12) optimalizálási feladatot skálázatlan modellnek, míg a (3.14) és a (3.13) feladatokat skálázott modellnek nevezzük.

A következőkben szükség lesz a következő formulára, amelyet Rockafellar és Uryasev [89], [90] vezettett be

$$\mathrm{Tail}_{\frac{i}{S}}\big(R_{\boldsymbol{x}}\big) = \max_{t_i \in \mathbb{R}} \left\{ \frac{i}{S}\,t_i - \frac{1}{S}\sum_{j=1}^{S} \Big[\, t_i - \boldsymbol{r}^{(j)\,T}\boldsymbol{x} \,\Big]_+ \right\},$$

ahol t_i döntési változó, $\boldsymbol{r}^{(j)\,T}\boldsymbol{x}$ pedig az $\boldsymbol{x}$ portfólió hozama, ha a j-edik forgatókönyv következik be. Ezt a képletet használták a [91] cikkben, hogy a nemskálázott (3.12) modellt egy lineáris programozási feladatra transzformálják.

Fábián és munkatársai a [31] cikkben adaptálták a Künzi-Bay és Mayer [54] vágósíkos módszert és a következő formulához jutottak:

$$\begin{aligned} \mathrm{Tail}_{\frac{i}{S}}\big(R_{\boldsymbol{x}}\big) = \quad &\min\ \tfrac{1}{S}\sum_{j \in \mathcal{J}_i} \boldsymbol{r}^{(j)\,T}\boldsymbol{x} \\ &\text{feltéve, hogy} \quad \mathcal{J}_i \subset \{1, \ldots, S\}, \;\; |\mathcal{J}_i| = i. \end{aligned} \tag{3.15}$$

Felhasználva a (3.15) felírást, a nemskálázott (3.12) modellt a következőképpen lehet átírni:

$$\begin{aligned} &\max \qquad \vartheta \\ &\text{feltéve, hogy} \quad \vartheta \in \mathbb{R}, \quad \boldsymbol{x} \in X, \\ &\qquad \vartheta + \widehat{\tau}_i \;\leq\; \tfrac{1}{S}\sum_{j \in \mathcal{J}_i} \boldsymbol{r}^{(j)\,T}\boldsymbol{x} \quad \text{minden} \quad \mathcal{J}_i \subset \{1, \ldots, S\}, \;\; |\mathcal{J}_i| = i, \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{ahol} \quad i = 1, \ldots, S. \end{aligned} \tag{3.16}$$

Hasonló módon a (3.14) skálázott modell a következő formulával adható meg:

$$\begin{array}{ll} \max & \vartheta \\ \text{feltéve, hogy} & \vartheta \in \mathbb{R}, \quad \boldsymbol{x} \in X, \\ & \frac{i}{S}\vartheta + \widehat{\tau}_i \;\leq\; \frac{1}{S}\sum\limits_{j\in\mathcal{J}_i} \boldsymbol{r}^{(j)\,T}\boldsymbol{x} \quad \text{minden} \quad \mathcal{J}_i \subset \{1,\ldots,S\},\ |\mathcal{J}_i| = i, \\ & \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{ahol} \quad i = 1,\ldots,S. \end{array} \tag{3.17}$$

A [31] és a [32] cikkekben Fábián és munkatársai bemutatták, hogy a vágósíkos módszer hatékony a skálázott és a nem skálázott problémákon is.

3.3.2. Valós adatokon való portfólió optimalizálás, generálási kísérletek előkészítése

Ehhez a részhez kötődik a kutatási munkám. A feladatom általános megfogalmazása az volt, hogy szimuláljak szcenáriókat, különböző feltételek mellett. Ezekből a futásokból származó eredményeket a kutatócsoporttal közösen értélkeltük ki.

Adatok és a kísérlet leírása

Az adathalmaz $n = 68$ értékpapír értékét az FTSE 100 kosárból és az FTSE 100 index hozamát tartalmazza heti bontásban, ugyanazon intervallumra. Jelölje $T = 835$ a hetek számát 1993. januártól 2009. januárig. Adott a t-edik $(0 \leq t \leq T)$ héten $n+1$ érték, az egyes komponensek árai és a részvényindex.

Jelöljük s_0^t -vel a részvényindexet, és jelölje s_k^t $(1 \leq k \leq n)$ a k-adik részvény árát. (Megjegyezzük, hogy ezek hétvégi adatok, ha $t \geq 1$, kiinduló hétnek a 0. hetet gondoljuk.)

A hozamokat a következőképpen határozzuk meg:

$$r_k^t := (s_k^t - s_k^{t-1})/s_k^{t-1} \quad (0 \leq k \leq n).$$

Ezeket egy $(n+1)$-dimenziós vektorban helyezzük el.

$$\boldsymbol{r}^t = (r_0^t, r_1^t, \ldots, r_n^t)$$

Tehát a valós adatokból kiszámolt kiinduló adathalmazunk a következő:

$$\mathcal{R} := \left\{ \boldsymbol{r}^t \mid t = 1,\ldots,T \right\}.$$

Azzal a feltételezéssel dolgozunk, hogy a hozam eloszlások ugyanazok mindegyik héten.

A kísérlet kiértékeléséhez felosztjuk az $\mathcal{R}$ adathalmazt két részre, egy $\mathcal{H}$ és egy $\mathcal{T}$ halmazra véletlenszerűen, de betartva $\mathcal{H} \cup \mathcal{T} = \mathcal{R}$, $\mathcal{H} \cap \mathcal{T} = \emptyset$.

A $\mathcal{H}$ halmazt portfólió megválasztásra használtuk, a $\mathcal{T}$ halmazt pedig az out-of-sample portfólió értékelésre.

Az optimális portfólió meghatározásokat a skálázott és nem skálázott modellek optimalizálásával tettük.

Szcenáriók generálása

A következő szcenárió halmazokat használjuk:

- *A ($\mathcal{H}$) tanuló minta* $\mathcal{H}$ elemeit egyenlő valószínűségű vektoroknak tekintjük. Ez képezi a tanuló mintákat, illetve az in-sample adatokat.
- *A ($\mathcal{G}$) generált minta (szcenáriók)*, amit Gauss-kopula és lognormális peremek segítségével generálunk. Ezeket a $\mathcal{H}$ mintára illesztve határozzuk meg.
- *A ($\mathcal{T}$) teszt halmaz* elemeit, mint vektorokat azonos valószínűségű vektoroknak tekintjük.

A fenti halmazok a következő számosságúak: $|\mathcal{H}| = 668$, $|\mathcal{T}| = 167$ (az arány $|\mathcal{H}| : |\mathcal{T}| = 4 : 1$). A generált minta $|\mathcal{G}| = 10\,000$.

A következőkben 12 szcenárió halmazt fogok szimulálni, ami alapján az optimális portfóliót határozzuk meg.

A kísérletek a következő módon történnek. Az $\mathcal{R}$ adatok partícióját 12-szer, egymástól függetlenül végeztük el. $\mathcal{H}^{(\ell)} \cup \mathcal{T}^{(\ell)}$ $(\ell = 1, \ldots, 12)$

Jelöljük $\mathcal{G}^{(\ell)}$-lel az ℓ-edik kísérlethez tartozó szcenáriót.

Kétfajta portfólió optimalizálást alkalmaztunk: a skálázott (scaled) és nem skálázott (unscaled) eljárást.

Az $\mathcal{R}$ adatok szétválasztását két $\mathcal{H} \cup \mathcal{T}$ halmazba véletlen számgenerátorral tettem.

A szcenáriók generálását megvalósító programot Matlabban implementáltam. A szcenárió generáláshoz Gauss-kopulát használtam, a peremeket pedig lognormális eloszlásokkal modelleztem.

A peremek közötti összefüggést Spearman-féle rangkorrelációval számoltam, mert ez jobban kifejezi az adatok között lévő összefüggést és segítségével becsülni tudjuk a Gauss-kopula paramétereit (lásd. Cherubini, Luciano, Vecchiato [16] könyvének 7.5.4. fejezete).

Az egydimenziós peremeloszlásokat a tanuló adatokra való lognormális eloszlás illesztéssel határoztam meg. Az adatokból megbecsültem a korreláció-mátrixot, amely input paramétere a Gauss-kopulának.

Mivel a beépített csomagok nem tudták kezelni a sok inputváltozót, illetve a nagy mintát, ezért az adatgenerálásra a kopula-függvény definíciója alapján a következő algoritmust dolgoztam ki [S9].

23. Algoritmus. ***Szimulációs modell Gauss-kopulával***

Bemenet: *Az adathalmaz $N \times d$ méretű mátrix pl. hozam, N az adathalmaz sorainak száma, d az oszlopok (változók) száma*
Kimenet: *d elemű vektor az illesztett d-dimenziós Gauss-kopulával modellezett együttes eloszlás egy pontja*

Kalibrálás:

1. *Az F_i peremeloszlások meghatározása, beépített lognormális eloszlásfüggvény illesztésével. Az F_i^{-1} inverzek származtatása F_i függvényekből. $(i = 1, \ldots, d)$*

2. *A Gauss-kopula korrelációinak becslése: A kopulához tartozó* $R = [\rho_{i,j}]$ *korrelációs mátrix a Spearman-féle rangkorreláció empirikus becslésével:*

$$\rho_{i,j} = \sin\left(\frac{\pi}{6}\rho_s\left(X_i, X_j\right)\right) \tag{3.18}$$

Véletlen vektorparaméterek generálása:

1. *Cholesky-dekompozíciós eljárással az A alsó háromszög mátrix meghatározása:* $R = AA^T$

2. *Független normális eloszlású véletlen* z_i *változókból* $z = (z_1, \ldots, z_d)^T$ *vektor szimulálása.* $(i = 1, \ldots, d\ )$

3. $y = Az$ *mátrix szorzat kiszámítása*

4. *Legyen* $u_i = \Phi(y_i)$. $(i = 1, \ldots, d)$

5. *Legyen* $X_i = F_i^{-1}(u_i)$, *ahol az* F_i *a peremeloszlás eloszlásfüggvénye, jelen esetben lognormális eloszlás.* $(i = 1, \ldots, d)$

3.3.3. A generált szcenáriók alapján, a két módszerrel meghatározott optimális portfóliók kiértékelése

Ebben a részben összehasonlítjuk a különböző optimalizálási módszerrel meghatározott optimális portfólió jellemzőit. Általánosan a következő két megfigyelés tehető:

– Az index eloszlása relatív hosszú bal farokkal rendelkezik. A skálázatlan optimális portfólió eloszlása baloldalt hamarabb cseng le, miközben a skálázott optimális portfólió eloszlásának a bal farka hasonló az index eloszlásáéhoz.

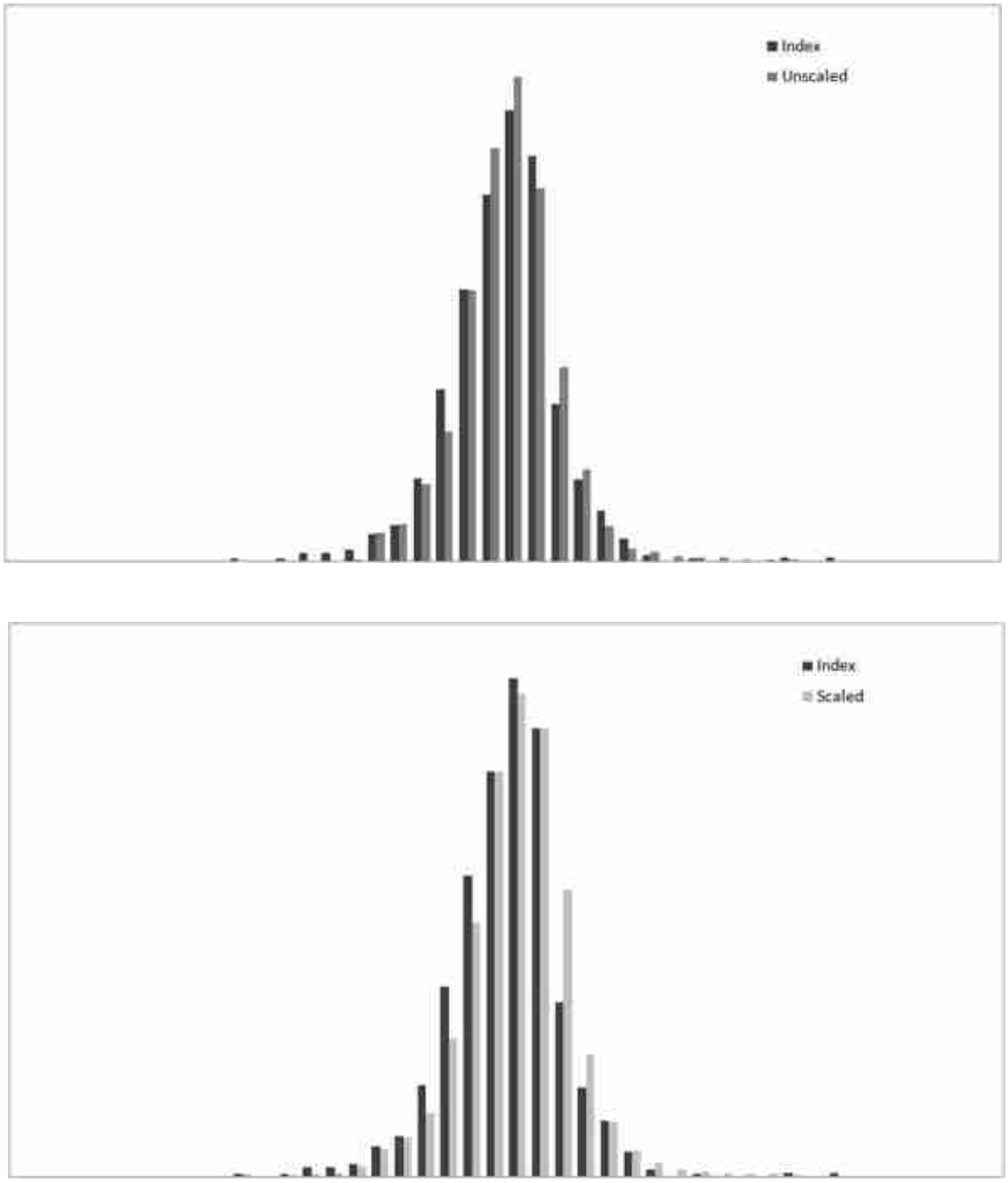
– Mindkét módszerrel kapott portfólió nagyobb várhatóértékkel és kisebb szórással rendelkezik, mint az index portfólió. A skálázott portfóliónak nagyobb a várhatóértéke, mint a skálázatlan portfóliónak, de a szórása is nagyobb a skálázatlanénál.

A tanuló mintán (in sample) optimalizált portfólió a következő várhatóértéket és szórást eredményezték. Látszik, hogy mindkét módszerrel meghatározott portfólió dominálja az indexet (FTSE100), várhatóértékük nagyobb, szórásuk kisebb.

	expect.	st.dev.
index	$0,0006$	$0,0239$
unscaled	$0,0023$	$0,0210$
scaled	$0,0045$	$0,0231$

Részletesebb képet kapunk 3.26. ábra alapján, ahol a két hisztogram, a skálázatlan, illetve skálázott eljárással kapott portfólió eloszlását mutatja az index eloszlással együtt.

A következő táblázatban láthatjuk a tanuló halmazon a két módszerrel meghatározott, két portfólió eredményeit a tesztelő halmazon (out of sample). A skálázatlan eljárásból származott portfólió eloszlás legtöbbször sztochasztikusan dominálta az index eloszlását



3.26. ábra. Tanuló adathalmazon optimalizált portfóliók eloszlása a tanuló mintán (in sample). Fent a skálázatlan módszerrel, lent a skálázott módszerrel. Mindkét ábrán az index eloszlása is látható

(9-ben a 12-ből). A skálázott eloszlás nem dominálta általában az index eloszlást (3-szor dominálta a 12-ből esetből). Megjegyezzük azonban, hogy a 9 nem dominált esetben csak néhány olyan feltétel nem teljesül, ami az extrém bal oldalon van.

	expect.	st.dev.
index	$0,0014$	$0,0235$
unscaled	$0,0023$	$0,0219$
scaled	$0,0032$	$0,0234$

A 3.27. ábrán láthatjuk a két hisztogramot, a skálázatlan, illetve skálázott eljárással kapott portfólió eloszlását, a tesztelő halmazon (out of sample) az index eloszlással együtt.

A következő várhatóértéket és szórást eredményezték a tanuló mintából generált szcenáriókra optimalizált portfóliók, alkalmazva a tanuló mintán:

	expect.	st.dev.
index	$0,0004$	$0,0239$
unscaled	$0,0021$	$0,0184$
scaled	$0,0046$	$0,0237$

A 3.28. ábrán láthatjuk a két hisztogramot, a skálázatlan, illetve skálázott eljárással kapott portfóliót, amit már a tanuló minta alapján generált szcenáriók alapján lettek optimalizálva. Mindegyik hisztogramon a tanuló mintára való alkalmazott portfólió eredményeit mutatja és az ehhez tartozó index eloszlást.

A következő várhatóértéket és szórást eredményezték a tanuló mintából generált szcenáriókra optimalizált portfóliók, alkalmazva a tesztelő mintán:

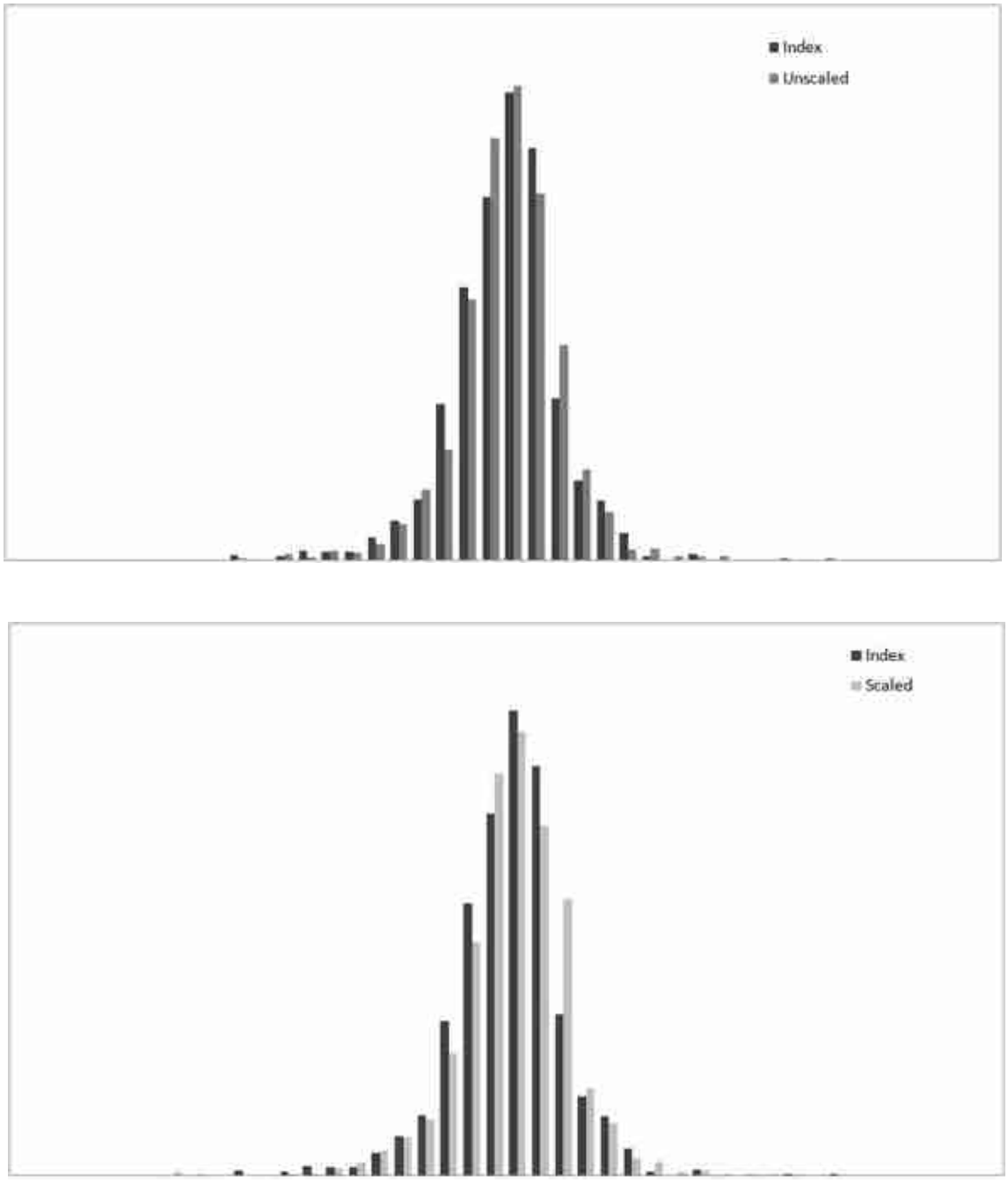
	expect.	st.dev.
index	$0,0014$	$0,0235$
unscaled	$0,0022$	$0,0188$
scaled	$0,0028$	$0,0235$

A 3.29. ábrán, láthatjuk a két hisztogramot, a skálázatlan, illetve skálázott eljárással kapott portfóliót, amelyek a tanuló minta alapján generált szcenáriók alapján lettek optimalizálva. Mindegyik hisztogram a tesztelő mintára alkalmazott portfólió eredményeit mutatja és az ehhez tartozó index eloszlást (out of sample).

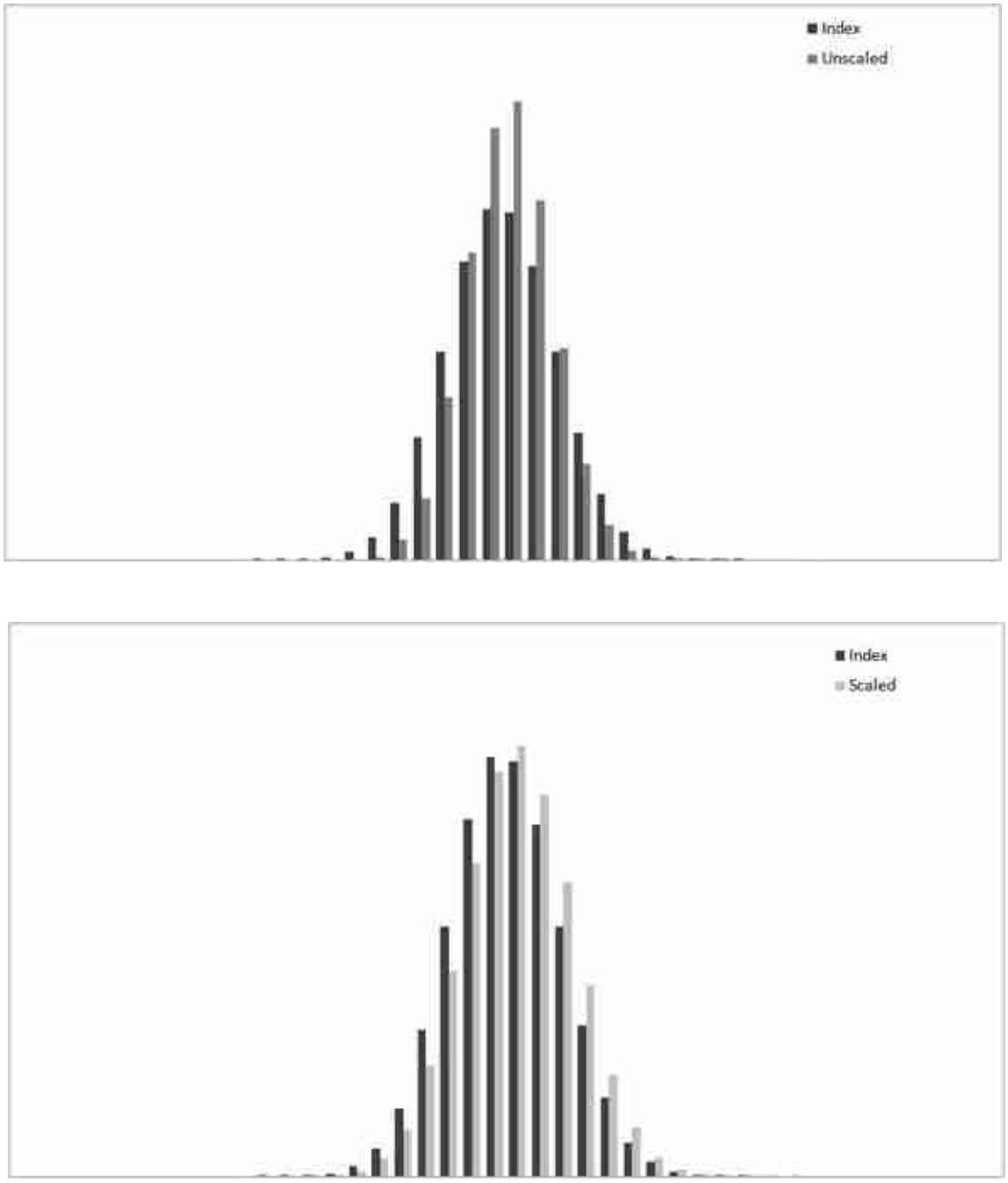
Összehasonlítva a portfólió összetételeket, amiket a generált szcenáriók alapján kaptunk, illetve az eredeti tanuló mintából kaptunk, elég nagy különbségeket tapasztaltunk. A vektorok közt maximális eltérés a 12 szcenárióban $80°$, átlagban $70°$. Minden kísérletben, a 12-ből, azt kaptuk, hogy a szimulált adatok alapján választott portfólió eloszlása kisebb szórású, mint a historikus adatok alapján választott portfólió. Kiemeljük, hogy a fő különbség a bal farok hosszában és szélességében rejlik, amelyek a szimulált adatok alapján választott portfólió esetén a rövidebb. A szcenárió generálás 10-20 százalékkal csökkentette a portfólió szórását. Ábrán illusztráljuk a két hisztogramot.

A 3.30. ábrán láthatjuk a különbséget a skálázatlan módszerrel kapott portfóliók között, amit a sima tanuló mintán optimalizáltunk, illetve, amit az általam generált szcenáriók alapján optimalizáltunk. A hisztogram a tesztelő adatokra készült (out of sample).

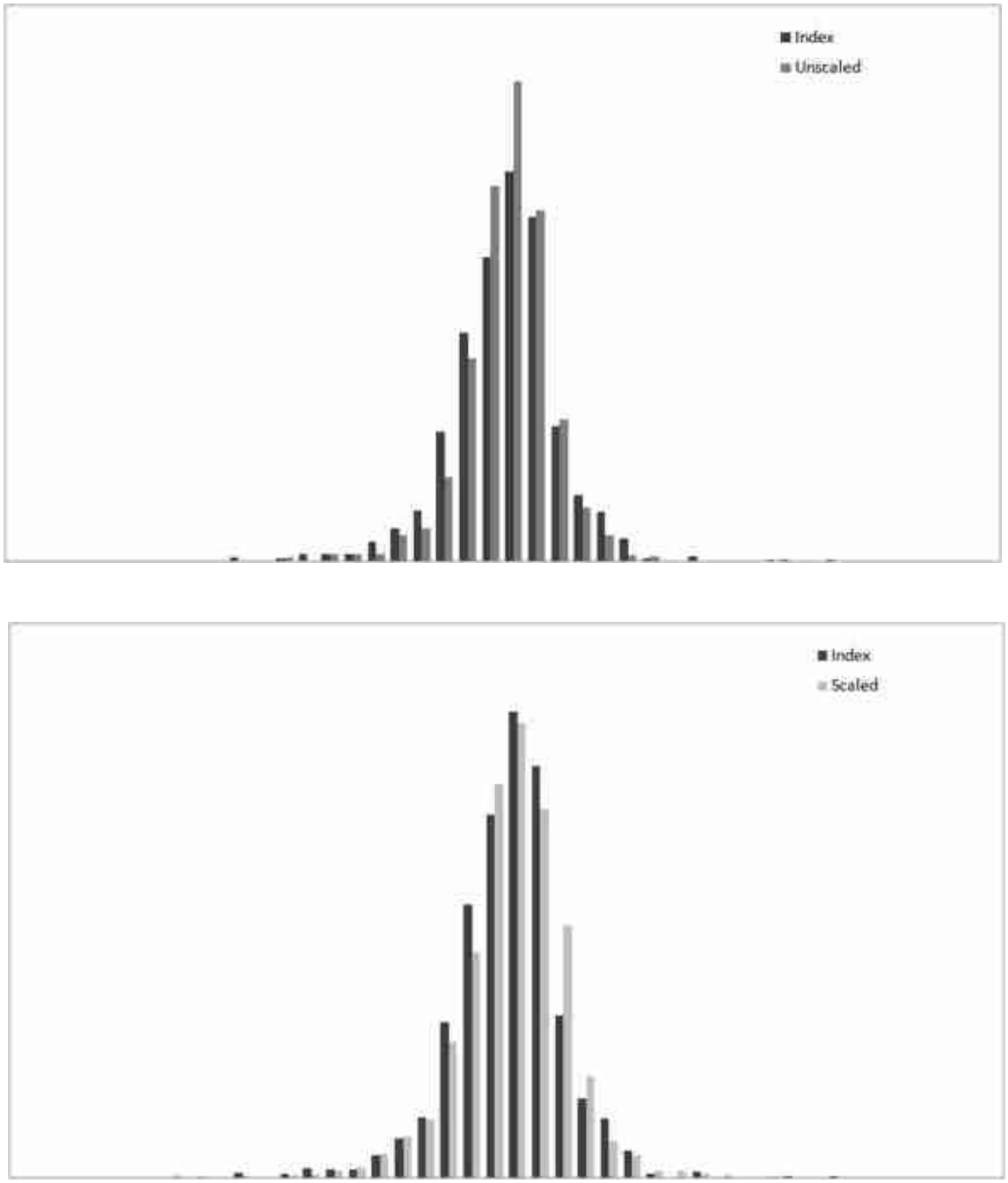
A 3.31. ábrán látható a szcenárió generálás hatása a skálázott módszerrel optimalizált portfólióra, a teszt adaton (out of sample). Az ábrán a tanuló mintán optimalizált, illetve



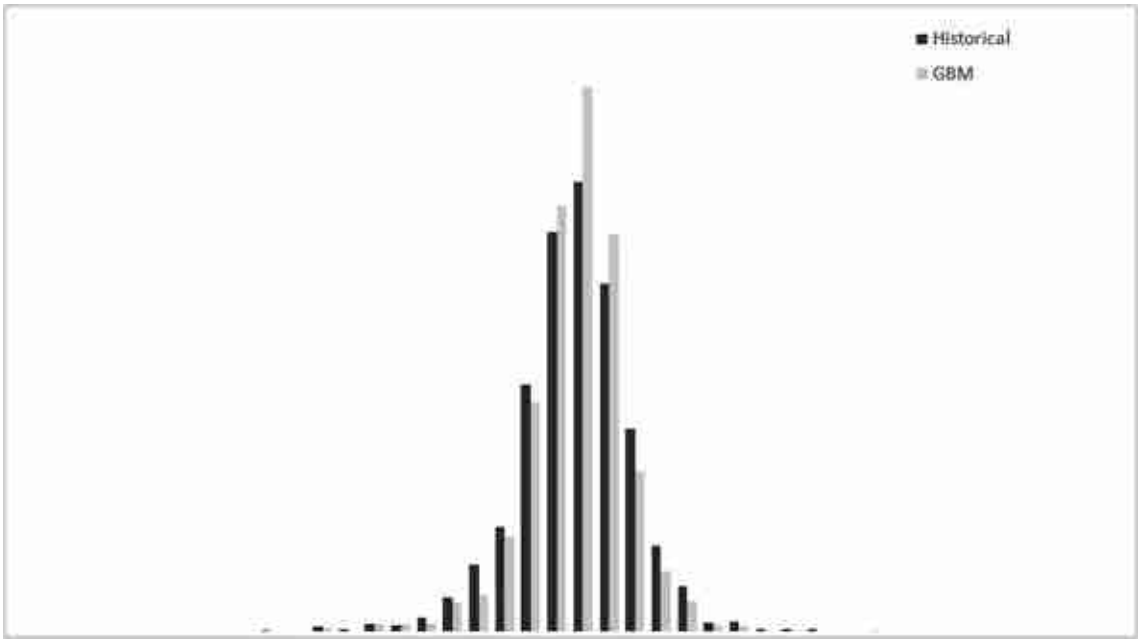
3.27. ábra. Tanuló adathalmazon optimalizált portfóliok eloszlása a tesztelő mintán (out of sample). Fent a skálázatlan módszerrel, lent a skálázott módszerrel. Mindkét ábrán az index eloszlása is látható.



3.28. ábra. Tanuló adathalmazból Gauss-kopulával generált szcenáriók alapján optimalizált portfóliók eloszlása a tanuló mintán (in sample). Fent a skálázatlan módszerrel, lent a skálázott módszerrel. Mindkét ábrán az index eloszlása is látható.



3.29. ábra. A tanuló adathalmazból Gauss-kopulával generált szcenáriók alapján optimalizált portfóliók eloszlása a tesztelő mintán (out of sample sample) Fent a skálázatlan módszerrel, lent a skálázott módszerrel. Mindkét ábrán az index eloszlása is látható.



3.30. ábra. A szcenário generálás hatása a skálázatlan módszerrel optimalizált portfólióra, a teszt adaton (out of sample). A tanuló mintán optimalizált, illetve a tanuló minta alapján Gauss-kopulával generált minta alapján optimalizált portfólió eloszlása a teszt adatokon.

a tanuló minta alapján Gauss-kopulával generált minta alapján optimalizált portfólió eloszlása látható a teszt adatokon (Out of sample)

Összehasonlítva a portfólió összetételeket, amelyeket a generált szcenáriók alapján kaptunk, illetve az eredeti tanuló mintából kaptunk, különbségeket tapasztalunk, de nem olyan nagyot, mint az előző esetben. A vektorok közt maximális eltérés a 12 szcenárióban 45°, átlagban 35° volt.

Ebben az esetben sokkal inkább hasonló portfólió eloszlásokhoz jutottunk, mint a tanulóhalmaz alapján.

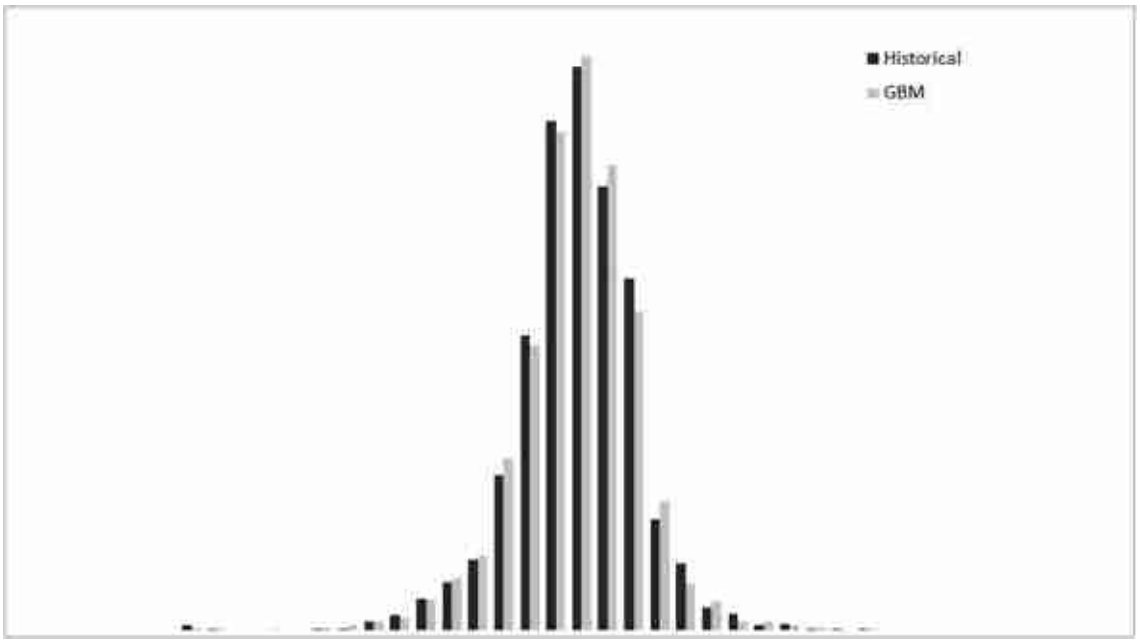
Generált adatnagyságtól függő konvergencia

Egy további tesztben megvizsgáltuk az optimális megoldásvektorok konvergenciáját bővülő forgatókönyvkészletek vonatkozásában.

Különböző mintaméretekkel 10-10 kísérletet végeztünk. Azt vizsgáltuk, hogy a 10 feladat megoldásából kapott optimális portfóliók mekkora sugarú gömbbe férnek bele. Minél kisebb a gömb sugara, annál közelebb vannak egymáshoz a megoldások. A táblázatban a különböző forgatókönyv készletekhez tartozó gömb sugarai láthatók.

	10000	20000	30000	40000	50000	60000
unscaled	0, 216	0, 213	0, 201	0, 199	0, 210	0, 189
scaled	0, 151	0, 101	0, 085	0, 081	0, 070	0, 066

A skálázatlan esetben hiába növeljük a mintaméretet nem vagy alig zsugorodnak a gömbök, a skálázott esetben viszont jóval gyorsabban zsugorodnak. Ezért a skálázott eljárás stabilabban működik.



3.31. ábra. A szcenárió generálás hatása a skálázatlan módszerrel optimalizált portfólióra, a teszt adaton (out of sample). A tanuló mintán optimalizált, illetve a tanuló minta alapján Gauss-kopulával generált minta alapján optimalizált portfólió eloszlása a teszt adatokon.

3.4. Összegzés

A fejezet első részében bemutattam a többváltozós valószínűségi eloszlás modellezésének lehetőségeit kopulák segítségével és ezeknek hatását a kockázatra. Rávilágítottam arra, hogy a széles farokeloszlásokat nemcsak a speciális összefüggéssel rendelkező kopulák használata okozhatja, hanem az alkalmazott peremeloszlások is. Az aszimmetriát is elő lehet állítani, akár Gauss-kopula és különböző peremek segítségével.

Ha a feltételes függetlenségi reláció kihasználásával jó közelítést kapunk, akkor a marginális párokhoz még többféle kopulát használhatunk egyszerre. Valójában ez lesz az úgynevezett vine-kopulák használatának alapja a többváltozós kopulák modellezésében.

A vizsgált kopulákhoz hozzávéve a teljes vine-struktúrát kiderült, hogy az adja meg a legjobb közelítést, ennek ellenére, mivel a paraméterei száma nagyon magas, nem igazán előnyös használni, az adatokra való túltanulás miatt. Helyette a sokkal inkább az első szinten elvágott (Truncated) fa kopulát célszerű használni.

A második részben egy valós adathalmazon, amely egy három egyenlő súlyozású eszközből álló portfólióból áll, bemutattam a függőségi modellezéshez használt különböző kopulák hatásait. Továbbá megmutattam, hogy ebben az esetben a feltételes függetlenségen alapuló approximáció még jobban közelíti a valós adatokat.

Egy másik valós adathalmazon, amely hat egyenlő súlyozású eszközből álló portfólióból áll, bemutattam a függőségi modellezéshez használt különböző kopulák hatásait, a teljes vine-kopulával való szimulációval kiegészítve.

Bemutattam, hogy a kopulákkal modellezett összefüggések különbözőképpen hatnak a

kockázati mértékekre. Különböző kísérleteket végeztem a függőség különböző aspektusainak illusztrálására.

A harmadik részben pedig összehasonlítunk két SSD dominanciára épülő portfólió optimalizálási módszert, azzal hogy szcenáriókat generáltam egy tanuló mintából kiindulva. Az alapcél egy olyan portfóliót találni, aminek hozam eloszlása minél jobban dominálja az index hozam eloszlását.

Két optimalizálási eljárást alkalmaztunk, skálázatlan [91] és skálázott [32], valamint ezek eredményeit tárgyaltuk in sample és out of sample, továbbá összehasonlítottuk a tanulóhalmaz generálás nélküli adatokból számolt optimális portfólióval.

A legfontosabb eredmény, hogy az általam generált minta alapján meghatározott portfóliók robusztusak. A nemskálázott eljárás többnyire megtartja a dominanciáját out of sample is. A skálázott modell esetében sok feltételt teljesít, de van néhány, amit nem, amelyek az extremitásokban vannak. Miközben a skálázatlan modell védekezik az extrém vesztességektől, a skálázott modell jobban leköveti az index hozam eloszlását. A szcenáriógenerálás hozzájárult a dominancia megtartásához az out of sample esetben. A skálázott eljárást kifejezetten támogatta a szcenáriógenerálás.

2. Tézis:

Szimulációt terveztem többdimenziós eloszlások modellezési lehetőségeire, különböző kopulákat felhasználva. Tárgyaltam a Gauss-kopula esetét, és megmutattam, hogy többféle kapcsolatot tud egyszerre modellezni, továbbá alkalmas aszimmetrikus eloszlások modellezésére is, különböző farokösszefüggésekkel. Ezekre a szakirodalomban nem térnek ki. Bemutattam 2- és 3-dimenziós szimulált kopulák felhasználásával ugyanazon normális peremek mellett a Student-, Clayton-, Gumbel-, Frank-kopula hatásait, majd bemutattam, hogy különböző peremek milyen hatást váltanak ki, ami az aszimmetria és a farokösszefüggéségeket illeti.

Kapcsolódó publikációk: [S6], [S2], [S18]

3. Tézis:

Mivel a kockázati mutatók a ritka eseményekhez kötődnek, ezeket farokösszefüggések jellemezik. Szimulációt terveztem és bemutattam a szimulált adatokon, hogyan befolyásolja az összefüggéseket leíró kopula a kockázati mutatókat. Az is igazolódott, hogy a közel-függetlenség csökkenti a portfólió kockázatát. Valós 3, majd 6 értékpapírból álló egyenletes portfólió esetén, kétirányú kutatást végeztem. Az egyik arra vonatkozott, hogy milyen típusú kopula illeszkedik legjobban az adatokra, a másik pedig a kockázati mutatókra való hatására vonatkozott. A kísérleteket elvégezve arra jutottam, hogy a sokparaméterű kopulák illeszkednek jobban, vagyis a vine- és a Gauss-kopula. Ugyanakkor jó eredményeket hozott a feltételes függetlenségeket tartalmazó (CI) kopula közelítés is, amely sokkal kevesebb paraméterrel rendelkezik, mint a vine-kopula. A portfólióból jól látszanak a Gumbel-, illetve a Clayton-kopula specifikus farokösszefüggései. Rámutattam, hogy kis méretű portfóliók esetén, a kockázat modellezésében komoly szerepe van a kopulatípusnak, vagyis az összefüggés modellezésnek, továbbá a generált mintanagyságnak is.

Kapcsolódó publikációk: [S6], [S2], [S18]

4. Tézis:
Portfólió optimalizáláshoz készítettem szcenáriógenerálási eljárást és algoritmust, amely segítségével több adatot tudtam szolgáltatni a két portfólió optimalizálási eljárásnak. A szimulációs eljárást Matlabban implementáltam. Az eredmények biztatóak, mivel a tanuló mintából tanult és szimulált adatok a Gauss-kopula és lognormális peremeloszlások segítségével elősegítettem az optimális portfólió megválasztását. A skálázott és skálázatlan esetben is a portfólió várhatóértéke nőtt és a szórás csökkent. A különbség az, hogy a skálázott esetben mintha a hozamot szemléltető hisztogramot jobbra toltuk volna, azaz minden egyenletesen jobbra tolódott. A skálázatlan esetben pedig mintha levágtuk volna a "bal farkát". Ezért a skálázatlan megoldás olyan befektetőknek jó, akik félnek az extrém kis valószínűséggel előforduló extrém nagy veszteségtől, míg a skálázott azoknak jó, akik inkább az általános befektetői hozzáállásnak felelnek meg.

Kapcsolódó publikációk: [S1], [S9], [S13], [S14], [S19], [S15], [S16], [S17], [S21]

4. fejezet

Adatpótlás

Az adatpótlás az adattudomány egyik kulcsfontosságú problémaköre. A hiányzó adatok megnövelik a döntéshozók bizonytalanságát, illetve csökkentik a modellek megbízhatóságát.

A valószínűségi gépi tanulás és a neurális hálózatok eszközeit használó, újonnan javasolt megközelítéseket az [S3] cikkben dolgoztuk ki. Az alábbiakban a valószínűségi gépi tanuláshoz tartozó új adatpótlási módszereimet mutatom be. A bevezetett módszerek hatékonyságát is vizsgáltam, összehasonlítva az R programcsomagban lévő ismert adatpótlási módszerekkel. A hiányzó adatok pótlására kidolgoztam egy valószínűség alapú gépi tanulási módszert, melynek keretében attribútum csökkentési eljárásokat is kidolgoztam. A módszerek hatékonyságának összehasonlítására kidolgoztam két mutatót, az egyiket a pontosságra, a másikat a végrehajtási időre vonatkozóan. Az algoritmusokat R nyelven implementáltam. A módszerek összehasonlítására kísérleteket végeztem több, különböző tulajdonságú adathalmazon.

A fejezet felépítése a következő. Az első részben rövid áttekintést adok az adatpótláshoz kapcsolódó eredményekről és ezen belül elhelyezem a jelen kutatásomat. A második alfejezetben az adatok előkészítése kerül bemutatásra, ahol a kategorikus és folytonos attribútumokat diszkrét attribútumokká alakítom, viszonylag kis értékkészlettel. A harmadik alfejezetben bemutatom a releváns attribútum-kiválasztási módszereket, amelyek az adatpótlási módszerem alappillére. Az ezeken alapuló adatpótlási algoritmusokat a negyedik részben ismertetem. Az ötödik részben először két általánosan is használható módszert mutatok be, amely alapján össze lehet hasonlítani az adatpótlási módszerek pontossági eredményeit és a végrehajtási időket különböző attribútumok és adatkészletek esetén. Majd ismertetem az adatkészleteket, és illusztrációként bemutatok néhány adatpótlási eredményt, amelyeket az R-ben használt módszerekkel hasonlítok össze. Az utolsó szakasz következtetéseket tartalmaz.

4.1. A probléma elhelyezése

Az adathalmazok különböző okok miatt hiányzó értékeket tartalmazhatnak. A kutatók sok esetben úgy kezelik a hiányzó adatokat, hogy az elemzésben csak a teljes mintavektorokat veszik számításba, azokat a megfigyeléseket, amelyeknél a változók egyikében sincs hiányzó adat. Az ilyen elemzések eredményei azonban torzíthatnak. Továbbá, ha több változóban is hiányzó adatok vannak különböző mintaelemekben, akkor a mintaelemek

kihagyása az eredeti minta egy nagyobb részének a kizárásához is vezethet, ez pedig a modell pontosságának és előrejelző erejének jelentős csökkentését okozhatja.

A legtöbb statisztikai és gépi tanulási algoritmust, beleértve a neurális hálózati architektúrákat is, befolyásolják a hiányzó adatok. Csak néhány olyan mesterséges intelligencia-módszer van, amely képes kezelni a hiányos bemenetet. Itt említhetjük a döntési fán alapuló technikákat [9] vagy a Viharos et al. által javasolt érdekes neurális hálózati kiterjesztést [120]. Általánosságban elmondható, hogy a legtöbb gépi tanulási módszer érzékeny a hiányzó adatokra. Ezért az adattudományi közösség folyamatosan dolgozik a hiányzó értékek kezelésének problémáján, különböző területekről származó módszerekkel [58, 92, 96, 6, 95, 124].

A hiányzó adatok kezelésének legjobb módja a hiányzó adatok pótlása (imputation), amely kicsit lazán fogalmazva, a hiányzó értékek helyettesítését jelenti plauzibilis becsült értékekkel. Little és Rubin [58] három mechanizmust fogalmazott meg, amelyek hiányzó adatokat generálhatnak.

- A teljesen véletlenszerű hiányzás (missing completely at random, MCAR) a legmagasabb szintű véletlenszerűség, ez azt jelenti, hogy a hiányzó értékek mintázata teljesen véletlenszerű, és nem függ semmilyen változótól, amely az elemzésben szerepelhet vagy nem szerepelhet. Az MCAR feltételezése az, hogy a hiányzás valószínűsége nem függ sem más változók megfigyelt értékeitől, sem az adatállomány megfigyeletlen részétől.

- A véletlenszerű hiányzás (missing at random, MAR) azon tulajdonsággal rendelkezik, hogy a hiányzó adatok valószínűsége az adathalmaz megfigyelt változóitól függ. Ez azt jelenti, hogy a hiányzás valószínűsége függ a megfigyelt adatoktól, de nem függ a megfigyeletlen résztől.

- A "nem véletlenszerű hiányzás" (missing not at random, MNAR) esetében a hiányzás a megfigyeletlen változóktól függ, nem pedig a megfigyelt változóktól.

A hiányzó adatok idősorokból is származhatnak. Ebben az esetben a hiányzás korrelálhat az idővel is. Jelen kutatásban olyan hiányzó adatokra összpontosítok, amelyek nem idősorokból származnak. A hiányzó értékeket úgy pótolom ki, hogy ezek az értékek összhangban legyenek a meglévő adatokkal, vagyis MAR típusú hiányzásokat feltételeztem.

Ebben a fejezetben bevezetett új módszereket összehasonlítom az R-ben megvalósított legnépszerűbb adatpótási módszerekkel. Az R nyelv gyakran használt platform az adatelemzésekben, a nyelv különféle csomagokat biztosít a hiányzó értékek pótlására. A beépített R-csomagok teljesítménye különböző adatkészletek esetében eltérő lehet, függhet az adatkészletek méretétől és az adatkészletekben található hiányzó értékek mennyiségétől.

Az egyes adatpótlási módszerek teljesítményének összehasonlíthatósága érdekében különböző adatkészletekre és különböző hiányzó érték arányokra alkalmaztam a módszereket. Az egyes adatpótlási módszerek értékeléséhez bevezettem egy, a relatív pontosságon (accuracy) alapuló pontosságszámítási módszert. A módszerek végrehajtási idejének összehasonlítása érdekében egy relatív végrehajtási időn alapuló mutatót is bevezettem.

Általában két fő esetet kell megkülönböztetnünk. Az első eset az, amikor a hiányzó értékeket tartalmazó változó folytonos, a második eset, amikor a hiányzó értékeket tartalmazó változó diszkrét vagy kategorikus. Ebben a részben diszkrét vagy kategorikus attribútumok pótlására adunk módszereket, amelyek kiterjeszthetők folytonos változókra is. Az itt bemutatott módszerek valószínűségi gépi tanulási háttéren alapulnak.

Az adatpótlási problémát különböző nézőpontból tárgyalják. Az egyik probléma az összes változó hiányzó értékének pótlása, a másik probléma pedig az, amikor csak azokat a változókat pótolják, amelyek hozzájárulnak egy jobb regressziós vagy osztályozási modellhez. A jelen kutatásban a hiányzó értékeket tartalmazó összes jellemző adatpótlásával foglalkozok. Így módszereim hasznosak lehetnek a felügyelet nélküli gépi tanulási feladatok esetére is.

Az adatpótlási módszerek két nagyobb csoportba sorolhatók: egyetlen értékkel való adatpótlási módszerek és több értékkel történő adatpótlási módszerek. Az egyetlen értékkel való pótlás az adathalmazban lévő adott változó minden egyes hiányzó értékének egy értékével való kipótlásán alapul, ami összhangban van a többi értékkel és adattal. A pótlást követően úgy használhatjuk az adatokat, mintha az összes adat eredetileg megfigyelt lenne. Néhány népszerű egyetlen értékes adatpótlási módszer a konstanssal való pótlások közül az átlaggal, mediánnal vagy módusszal való pótlás.

Egy másik módszer, amely egyetlen érték pótlását adja, a lineáris regresszión alapul, ahol a hiányzó értéket más változók nem hiányzó adatain alapuló lineáris regresszióval előrejelzett értékkel pótolnak. Ez a módszer az attribútumok közötti erős lineáris kapcsolat feltételezésén alapul. A módszer előnye az átlaggal való pótlás módszerrel szemben az, hogy a regresszióra épülő pótlás képes megőrizni az eloszlást, ami például az átlaggal való pótlás esetében nem lehetséges. Ha magas Pearson-korrelációk vannak, akkor érdemes ezt a módszert használni más bonyolultabb módszerek helyett.

A többszörös adatpótlás esetében a hiányzó értéket a lehetséges értékek feletti valószínűségi eloszlással imputálják. Úgy imputálnak ebből kiindulva egy értékkel, hogy az adott valószínűségi eloszlásból mintavételeznek egy értéket, vagy az értékek közül a legvalószínűbbet választják ki.

Egy másik népszerű módszer a KNN módszer [87], amely a k-legközelebbi adatpontokat használja a hiányzó érték pótlásához. Ez a módszer olyan távolságot használ, amelyet a magyarázó változók (attribútumok) által meghatározott altérre alkalmaznak. A módszer a k-legközelebbi pontokhoz szavazással hozzárendelt diszkrét értéket/kategóriát használja.

Liao és munkatársai az [57]-ben különböző KNN-alapú módszereket dolgoztak ki. Átfogóan összehasonlították a komplex betegségek modern orvosbiológiai kutatása során gyűjtött nagyszámú demográfiai és klinikai változót tartalmazó adatok meglévő adatpótlási módszereit, beleértve a cikkükben kidolgozott KNN adatpótlási módszerek négy kidolgozott változatát is. Arra a következtetésre jutottak, hogy egyik módszer sem teljesített általánosan minden helyzetben a legjobban.

A MissForest csomag egy nem parametrikus, döntési fákon alapuló adatpótlási módszert használ [100]. A módszer úgy működik, hogy az imputálandó attribútumot célváltozónak tekinti és a hiányzó érték előrejelzéséhez a többi attribútum információinak felhasználásával véletlen erdő kerül kialakításra. A MissForest számítási szempontból vonzó és hatékonyan kezeli a nagydimenziós adatokat is.

A HMISC csomag [40] az imputációhoz additív regressziót és bootstrappinget, valamint prediktív átlagillesztést használ. A kategorikus esetekben jellemzően a Naive Bayes-módszert használták [34]. Yadav és Roychoudhury a [123] tanulmányban R-ben használt népszerű – nevezetesen a VIM, MICE, MissForest és KNN – adatpótlási csomagokat hasonlítanak össze. Összehasonlítják a módszereket a minta mérete, a hiányzó adatok aránya és a futási idő alapján.

Egy másik érdekes, friss munkában Tsai és Hu [106] olyan adatpótlási módszerek összehasonlítását végzi el, mint a KNN, a CART, az MLP és a Naive Bayes. A szerzők megállapították, hogy az MLP, az SVM és a CART a három legjobb adatpótlási módszer az osztályozási pontosság szempontjából, és teljesítményük nem különbözik jelentősen. A szerzők megállapították, hogy a vegyes adathalmazokban a numerikus adatok RMSE-je szempontjából a CART és az SVM a legjobb. Ha az összes értékelési mérést és a számítási időt is figyelembe vesszük, a CART tűnik a legjobb választásnak [106].

A témához kapcsolódó érdekes tanulmány született Emmanuel és munkatársaitól, a hiányzó adatok szerepéről a gépi tanulásban [29]. Ebben a tanulmányban gépi tanuláson alapuló technikákat hasonlítanak össze olyan esetekben, amikor a hiányzó adatok aránya viszonylag alacsony, 5% és 20% között van.

Egy másik, a mi valószínűségi gépi tanulási megközelítésünkhöz közel álló, nemrégiben Ye és munkatársai által megjelentetett munka [124] a Bayes-hálózatokon alapul. Az alapötletük, hogy először egy Bayes-hálózatot kell építeni a "releváns kapcsolatok" feltárására. Az élek hozzáadása a megbízhatósági pontszám alapján történt, azaz az élek iránya egy megbízhatóbb attribútumtól egy kevésbé megbízható felé mutat. A megbízhatósági pontszámhoz a két attribútum közötti Pearson-féle korrelációs együttható abszolút értékét használták. Ily módon ez a struktúra valójában nem oksági kapcsolat alapján épült fel, mint az az általános Bayes-hálózatok esetén szokott. Az adatpótlási feladathoz a Bayes-féle következtetési mechanizmust alkalmazták. Az imputált érték az az érték, amely maximalizálja az attribútum feltételes valószínűségét, figyelembe véve a szülők értékeit.

A jelen dolgozatban bemutatott új megközelítés szintén releváns attribútumokon alapul, de ezeket külön-külön állapítom meg, minden egyes imputálandó attribútumra, és a Pearson-féle korreláció mellett más relevancia-mutatókat is használok.

Az adatadatpótlási módszer egyik fontos szempontja a tanító adatokból meghatározott magyarázó attribútumhalmaz minősége. Bizonyos esetekben az attribútumok egy részhalmaza jobb modellt fog generálni, mint az eredeti teljes attribútumkészlet, mivel egyes attribútumok irrelevánsak, és további zajokat generálnak az adathalmazban. Liu és munkatársai [59] cikkének célja, hogy megvizsgálja az attribútum kiválasztás (Feature Selection) elvégzésének hatását az orvosi adatkészletek hiányzó értékeinek pótlására. A vizsgálat az olyan attribútumcsökkentési megközelítésekre terjedt ki, amelyek az információnyereségen, mint szűrőalapú módszeren, genetikus algoritmuson, mint wrapperalapú módszeren és döntési fán, mint beágyazott alapú módszeren alapulnak. Az adatpótlási módszerek tekintetében a szerzők a k-legközelebbi szomszéd, többrétegű perceptron és támogató vektorgép megközelítéseket (SVM) valósították meg. A szerzők összehasonlító teszteket végeztek öt, különböző attribútum számosságú orvosi adatkészleten. Három különböző típusú attribútum kiválasztási módszert és adatpótlási technikát teszteltek. Megmutatták, hogy az attribútum kiválasztás kombinálása számos orvosi adatkészlet ese-

tében növeli az adatpótlási módszerek hatékonyságát.

4.2. Adatok előkészítése és az adatok diszkretizálása

Jelen kutatásban jellemzően diszkrét és kategorikus változók adatpótlására szolgáló módszereket vezettem be. Ahhoz, hogy ezek a módszerek általánosabb esetekre is alkalmazhatók legyenek, először néhány transzformációt kell elvégezni az adatpótlási probléma típusának megfelelően. Az adatelőkészítés lehetővé teszi számunkra az alábbi adatpótlási problémák megoldását: az attribútumok diszkrétek (azaz véges számú értéket vesznek fel, a minta méreténél jóval kisebb számút) vagy kategorikusak (nem, családi állapot, tandíj stb.), az utóbbi esetben a változókat számszerűvé kell transzformálni. Ha az imputálandó attribútum folytonos, vagy a magyarázó változók folytonosak, akkor fontos lépés a diszkretizálásuk, azaz a változók tárolókba sorolása.

Ha a folytonos attribútumokat diszkrét attribútumokká kell alakítani, a kimenet egy intervallum lesz, amelybe a hiányzó érték tartozik. Ez nagyon hasznos lehet az adott terület szakértői számára. Egy másik lehetőség az adott intervallumhoz tartozó értékek átlagértékét vagy mediánját használni az adatpótlásra.

Az itt bevezetett adatpótlási módszerek középpontjában az a meggondolás áll, hogy viszonylag kevés, de informatív attribútum legyen kiválasztva. Az imputálandó attribútumot célattribútumnak vagy célváltozónak nevezzük. Az itt bemutatott valószínűségi gépi tanulási megközelítés jellemzően kevés, a lehető leginformatívabbnak választott magyarázó változót használ.

Az adatokat a következő algoritmussal diszkretizálom. Jelölje M a kategóriák maximális számát az adathalmazban. Vizsgálataimban az M értéket az [5,20] intervallumból választom ki, az adathalmaz méretétől is függően. Jelen esetben $M = 10$.

Legyen X kategorikus attribútum $Dom(X)$ értékkészlettel, az első lépés a $Dom(X)$ elemeinek $LDom(X)$ listába rendezése. A $Dom(X)$-ben található v' érték esetében $i(v)$ jelölje a v helyét a $LDom(X)$-ben. A v célértékét a következőképpen számítom ki

$$v' = round\left(\frac{i(v)}{|Dom(X)|} min(M, |Dom(X)|)\right).$$

Az X numerikus attribútum esetén a célértéket a $Dom(X)$-ben lévő v' értékre a következő módon számítom ki

$$v' = round\left(\frac{v - min(X)}{max(X) - min(X)} M\right) ,$$

ahol $min(X)$, illetve $max(X)$ a $Dom(X)$ minimális és maximális elemét jelöli.

Vizsgálataimban ezt a diszkretizációs algoritmust a folytonos változókra alkalmaztam.

4.3. Magyarázó változók kiválasztási módszerei

Az itt bevezetett módszer a hiányzó értékek pótlására a releváns attribútumok rendelkezésre álló értékeit használja fel. Ez biztosítja, hogy az imputált értékek összhangban legyenek a meglévő adatokkal. A kísérletek alapján megfigyelhető, hogy a magyarázó változók különböző kiválasztásai nagyon eltérő pontossági eredményeket adhatnak ugyanarra

a célváltozóra. Széles körben elfogadott, hogy az irreleváns attribútumok kiküszöbölése növelheti az adatpótlási folyamat hatékonyságát.

Módszerem egyik fontos gondolata, hogy egy attribútum adatpótlása a leginformatívabb attribútumok által felvett értékek alapján történjen. Fontos, hogy az adatpótlási folyamatban használt magyarázó attribútumok száma a lehető legalacsonyabb legyen. Ha a magyarázó változók sok hiányzó értéket tartalmaznak, akkor a legalább egy magyarázó attribútumban hiányzó értéket tartalmazó sorok törlése nagy mennyiségű adat elvesztéséhez vezethet. Ezért a célváltozó adatpótlásához az első lépés az, hogy meghatározom az adott célváltozóhoz használandó erős magyarázó változók minimális halmazát. Ily módon központi feladat a releváns változók kiválasztása. Ehhez először néhány jól ismert korrelációs együtthatót alkalmazok, amelyek a valószínűségi változók közötti függőség leírására szolgálnak. Az adatpótlási feladathoz egy új módszert vezetek be, amely a releváns attribútumok kiválasztására épül és információelméleti fogalmakat használ fel.

4.3.1. Módszer a legjobb magyarázó jellemzők kiválasztására bináris együtthatók alapján: cor_k, *ken*_k, *sp*_k, *MI*_k

Az alábbiakban bemutatok néhány jól ismert együtthatót, amelyek leírják, hogy mennyire erős a függőség két változó között. Ezután bemutatom az úgynevezett információtartalmat, amely a kölcsönös információ általánosítása. Majd a bemutatott bináris együtthatók alapján a bevezetett attribútum választási algoritmust ismertetem, ezeket is fogom használni az új adatpótlási módszerben.

Pearson-korrelációs együttható

Numerikus változók esetén a korreláció számításának legelterjedtebb módja a Pearson-korreláció.

A Pearson-korreláció meghatározása a következő:

$$corr(X,Y) = \frac{Cov(X,Y)}{\sigma\left(X\right) \cdot \sigma(\,Y)}\,,$$

ahol a $Cov(\cdot,\cdot)$ a kovarianciát, a $\sigma\left(\cdot\right)$ pedig a szórást jelöli.

Kendall-féle konkordancia együttható

A nemlineáris összefüggések jellemzésére a Kendall-féle konkordancia együtthatót szokták használni.

Tekintsünk két X és Y valószínűségi változót és ezek $(x_1, y_1), \ldots, (x_n, y_n)$ realizációit. Azt mondjuk, hogy (x_i, y_i) és (x_j, y_j), $i < j$ konkordánsak, ha ($x_i > x_j$ és $y_i > y_j$) vagy ($x_i < x_j$ és $y_i < y_j$), ellenkező esetben azt mondjuk, hogy diszkordánsak.

A Kendall-τ együttható a következő módokon számítható ki:

$$\tau = \frac{\text{(konkordáns párok száma)-(diskordáns párok száma)}}{\binom{n}{2}}\,.$$

Ez a következőképpen is kifejezhető:

$$\tau = \frac{\sum\limits_{i<j} sgn(x_i - x_j) sgn(y_i - y_j)}{\begin{pmatrix} n \\ 2 \end{pmatrix}} .$$

Spearman-együttható

Diszkrét vagy folytonos attribútumok esetén a köztük lévő kapcsolat vagy függőség leírására alkalmas a Spearman-féle együttható. Az ötlet az, hogy a két attribútum értékeit rangokká alakítjuk át, így az eredeti adatokban két egymást követő érték közötti távolság nem lesz azonos a rangsorolt adatokban lévővel. Jelöljük X rangsorolt adatait $R(X)$-szel, Y rangsorolt adatait pedig $R(Y)$-nal.

A Spearman-rangkorrelációt a következőképpen határozzuk meg:

$$\varrho_{R(X),R(Y)} = \frac{Cov(R(X), R(Y))}{\sigma(R(X)) \cdot \sigma(R(Y))} ,$$

ahol $Cov(\cdot, \cdot)$ a kovariancia, a $\sigma(\cdot)$ a szórás.

Kölcsönös információ (Mutual Information)

Ha csak az attribútumok közötti függőségre, azaz az együttes valószínűségi eloszlásra vagyunk kíváncsiak, használhatunk entrópián alapuló együtthatókat. Ezek nem függnek az attribútumok által felvett értékektől (kategóriáktól).

Az (X, Y) véletlenvektor kölcsönös információját a következő képlet adja meg:

$$I(X, Y) = H(X) + H(Y) - H(X, Y),$$

ahol a $H(X)$ és $H(X, Y)$ az entrópiákat jelöli, amelyeket az alábbi képletek adnak meg:

$$H(X) = -\sum_i p_i \log p_i$$

és

$$H(X, Y) = -\sum_{i,j} p_{ij} \log p_{ij},$$

ahol p_i az X értékein való valószínűségi eloszlást, a p_{ij} az (X, Y) értékvektorokon való valószínűségi eloszlást jelöli. Hasonlóan írható fel az entrópia többváltozós valószínűségi vektor esetében is.

Információ tartalom

Az $(X_1, \ldots X_k)$ véletlen vektor információtartalmát (amely a kölcsönös információ általánosításának egyik módja) a következő módon adjuk meg:

$$I(X_1, \ldots X_k) = \sum_{i=1}^{k} H(X_i) - H(X_1, \ldots X_k).$$

Fontos tulajdonsága, hogy a meglévő változókhoz új változó hozzáadásával nő az információtartalom. Ezt a következő egyenlőtlenség fejezi ki:

$$I\left(Y, X_1, \ldots, X_k\right) \leq I\left(Y, X_1, \ldots, X_k, X_{(k+1)}\right). \tag{4.1}$$

Az előzőekben bemutatott bináris együtthatók alapján a legjobb magyarázó változókat, amelyeket az adatpótlási feladatban fogok használni a következőképpen választom ki. A magyarázó változókat a célváltozótól (az imputálandó attribútumtól) való függésük csökkenő sorrendjében rendezem, és a Pearson (cor_k), Kendall (ken_k), Spearman (sp_k) és a kölcsönös információ (MI_k) segítségével az első k darab változót választom ki.

4.3.2. A leginformatívabb attribútumhalmaz kiválasztásának módszere: MIN_k

A módszer lényege, hogy egyenként kiválasztok egy új változót, amely maximalizálja az Y célváltozót is tartalmazó attribútumhalmaz információtartalmát. Tegyük fel, hogy már m magyarázó változót (attribútumot) választottam a célváltozót is tartalmazó halmazba. Az $m+1$-edik változót a következőképpen választom ki.

$$X_{i_{m+1}} = \underset{i_{m+1} \in V \setminus \{i_1, \ldots i_m\}}{\arg\max} I\left(Y, X_{i_1}, \ldots, X_{i_m}, X_{i_{m+1}}\right).$$

Így minden egyes lépésnél maximalizálom az információtartalmat, kiválasztva a "legjobbat" a megmaradt változók közül. Ezt az algoritmust alkalmazó módszert MIN_k-nak nevezem, ahol k a kiválasztott magyarázó változók számát jelöli. Ez minden esetben egy növekvő függvény lesz a (4.1) képletnek köszönhetően.

4.3.3. Módszer a leginformatívabb jellemzőkészlet kiválasztására redundanciacsökkentéssel: mrmr_k

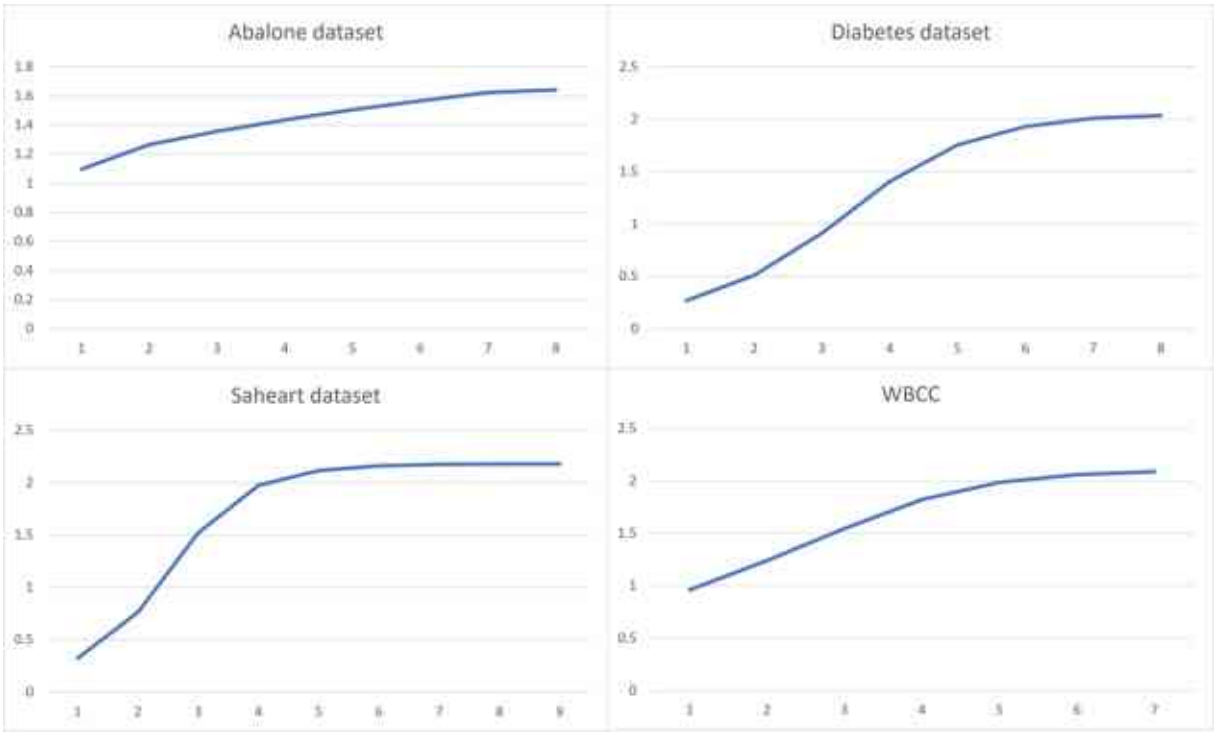
A következőkben leírt módszerhez a kutatásaink során jutottunk el, teljesen függetlenül az alábbi cikktől. A képletet és a bizonyítást a cseresznyefa eloszlásokkal való megközelítésből származó ötletek inspirálták [49]. Miután megalkottuk a mutatót, találtunk rá arra, hogy ezt már 2005-ben bevezették, ennek ellenére kevésbé terjedt el.

A módszer lényege, hogy az új magyarázó jellemzőt az információtartalom maximalizálásával és a redundancia minimalizálásával választjuk ki. A jellemző kiválasztásának ezt az ötletét Peng, Long és Ding a [71] cikkben dolgozta ki, a módszernek a *max Relevancia-min Redundancia* (mRmR) nevet adták. Most ezt használom az attribútum kiválasztáshoz.

Tegyük fel, hogy az $X_{i_1}, \ldots X_{i_m}$ változókat már kiválasztottuk az Y célváltozó magyarázó készletébe és az $X_{i_{m+1}}$ -edik változót kell kiválasztani. Egy új $X_{i_{m+1}}$ változó magyarázó erejét a már meglévő magyarázó változókhoz hozzáadva $J_Y(X_{i_{m+1}})$ -vel jelöljük, és az alább képlet szerint számoljuk:

$$J_Y(X_{i_{m+1}}) = I\left(Y, X_{i_1}, \ldots, X_{i_m}, X_{i_{m+1}}\right) - I\left(X_{i_1}, \ldots, X_{i_m}, X_{i_{m+1}}\right).$$

A fenti képlet az Y célváltozót is tartalmazó $\left(Y, X_{i_1}, \ldots, X_{i_m}, X_{i_{m+1}}\right)$ véletlenvektor információtartalma és a csak magyarázó változókat tartalmazó $\left(X_{i_1}, \ldots, X_{i_m}, X_{i_{m+1}}\right)$ véletlenvektor információtartalmának különbsége. Ezen elképzelés alapján az új magyarázó attribútumot úgy választjuk ki, hogy az maximalizálja a $J_Y(X_{i_{m+1}})$ értéket.



4.1. ábra. Az mRmR információ alapú együttható ($J_Y(X_{i_m})$) függése a magyarázó attribútumkészletben lévő attribútumok számától (m) különböző adatkészleteken

Az $m+1$ magyarázó változó kiválasztására szolgáló algoritmus egy mohó algoritmus, amely minden egyes lépésben a $J_Y(X_{i_{m+1}})$ mohó maximalizálását alkalmazza, amíg a magyarázó változók száma el nem ér egy fix számot, amelyet egy k paraméter ad meg.

Az algoritmus lényege, hogy megtalálja az Y célattribútumra vonatkozó leginformatívabb magyarázó változók halmazát, amelyek "a lehető legfüggetlenebbek" (alacsony információtartalom) egymástól.

Az előzőekben bemutattam azokat az együtthatókat, amelyeket az adatpótlási feladatban fogok használni. A kísérleti részben arra a következtetésre jutottam, hogy az esetek többségében a magyarázó változók viszonylag alacsony száma által adott információtartalom nem javítható jelentősen.

A 4.1. ábrán négy adatkészlet esetében látható, hogy a magyarázóerő a magyarázó attribútumkészletben szereplő attribútumok számával hogyan növekszik. Legtöbb esetben négy magyarázó változóig tapasztalható növekedés az mRmR értékben, nagyobb számú magyarázó változónál azonban már alig növekszik.

A következő alfejezetben a valószínűségi gépi tanulási adatpótlási módszereket mutatom be.

4.4. Adatpótlás valószínűségi gépi tanulási módszerek alapján

Ebben a részben diszkrét és kategorikus adatok pótlására vezetünk be valószínűségi gépi tanulási módszereket. Ezek a módszerek folytonos adatokra is alkalmazhatók, ha előzete-

sen elvégezzük a diszkretizálást.

Az itt bevezetett új módszerek alapötletének rövid megfogalmazása a következő: A valószínűségi gépi tanuláson alapuló adatpótlási módszerek a legrelevánsabb jellemzőket választják ki a legvalószínűbb értékkel való imputáláshoz.

Egy másik valószínűségi gépi tanulási adatpótlási módszer a Naive Bayes, amely azt a hipotézist használja, hogy a magyarázó változók függetlenek az adott célváltozótól. Eredményeimet ehhez a módszerhez is viszonyítom.

Egy döntő probléma minden egyes célváltozóhoz (imputálandó változóhoz) a magyarázó változók leginformatívabb részhalmazának kiválasztása. A magyarázó változók számának a lehető legalacsonyabbnak kell lennie, mivel az összes attribútum tartalmazhat hiányzó értékeket. Diszkrét változók esetében, ha a célváltozóval való korreláció abszolútértékei magasak, valószínűleg nagyon jó eredményeket kapunk lineáris regresszió alkalmazásával, vagy a magyarázó változók korreláció alapján történő kiválasztásával. Ha a célváltozó kategorikus, akkor a Pearson-, Spearman-, Kendall-, korreláció nem megfelelő, mert a jellemző értékei nem számok, még akkor sem, ha diszkrét számokat társítunk hozzájuk. Ebben az esetben csak az információnyereségen alapuló mértékekre támaszkodhatunk, mert ezek csak a valószínűségi eloszlást veszik figyelembe, és nem a felvett "értékeket".

A módszer célja az attribútumok hiányzó értékeinek pótlása úgy, hogy minden adat pótlásához a leginformatívabb attribútumokat használjuk. A módszer két fő lépésen alapul:

- A "legjobb" attribútumok listájának meghatározása a 4.3. szakaszban bemutatott jellemzőválasztási módszerek segítségével (az mRmR, MI, MIN, cor, ken, sp listából).
- A magyarázó attribútumok értékei alapján a legvalószínűbb értéket választjuk ki az adatpótláshoz (lásd a 24. algoritmust).

Jelöljük D-vel az adathalmazt, amelyben a sorok az adatpontokat jelentik (minták). Tegyük fel, hogy $\{X_1, \ldots, X_k\}$ a magyarázó változók halmaza egy Y célváltozóhoz, amely $\{y_1, \ldots, y_m\}$ diszkrét értékeket vesz fel. A jelölés rövidítése céljából használjuk az $(x_1, \ldots, x_k)$ jelölést az $(X_1 = x_1, \ldots, X_k = x_k)$ helyett. Az általános gépi tanuláson alapuló Valószínűségi Imputálási Algoritmus (Probabilistic Imputation Algorithm) megadja az Y imputált értékét. Amennyiben az $(x_1, \ldots, x_k)$ vektort nem találjuk a megfelelő k-adrendű peremben, akkor a kisebbrendű peremekben keressük az egyezést. Az $N(y_s)$ összeg azt jelenti, hogy hányszor jelenik meg az y_s együtt az $(x_1, \ldots, x_k)$ értékekkel. A $Mode(Y)$-t akkor vesszük figyelembe, ha egyik peremértékhez sem találunk megfelelő y-t. Ez a tanuló mintában nem fordulhat elő, viszont a pótlandó adatokban igen. A 24. algoritmust R nyelven implementáltam és az R-ben található adatpótlási csomagokkal hasonlítottam össze.

24. Algoritmus. ***Valószínűségi Imputálási Algoritmus***

Bemenet: Az y hiányzó értékhez tartozó $(x_1, \ldots, x_k)$ *értékek*
Kimenet: Az imputált y_{imp} *érték*

$l = 0$
while $l < k$ *do*
Számítsuk ki az $N(y_s)$ *összeget az összes* $\{i_1, \ldots, i_{k-l}\} \subseteq \{1, \ldots, k\}$ *részhalmazon*

$$N(y_s) = \sum_{(x_{i_1},\ldots,x_{i_{k-l}})\in D} 1_{\{(x_{i_1},\ldots,x_{i_{k-l}},y_s)\in D\}} \ , \quad \textit{ahol } y_s \in \{y_1, \ldots, y_t\}$$

if $N(y_s) \neq 0$ *then*
$y_{imp} = \underset{y_s\in\{y_1,\ldots,y_t\}}{\arg\max}\ N(y_s)$
break
else $l = l + 1$
end if
end while
if $l == k$ *then*
$y_{imp} = Mode(Y)$
end if

4.5. A pontosság mérése és tesztadatbázisok bemutatása

Ez a rész három alfejezetből áll. Az elsőben bevezetek egy módszertant, amely lehetővé teszi több módszer összehasonlítását különböző adathalmazokon. A második alfejezet a felhasznált adatkészletek rövid leírását és a szemléltetéshez használt attribútumok bemutatását tartalmazza. A harmadik alfejezet a bemutatott módszerek adatpótlási eredményeit és más, már ismert módszerekkel való összehasonlítását tartalmazza.

4.5.1. Az adatpótlási módszerek értékelésének módszertana. Összehasonlítás több adatkészleten

A tesztelt módszerek több adathalmazon való összehasonlíthatósága érdekében bevezetek egy együtthatót a módszerek relatív pontosságának jellemzésére, valamint egy másik együtthatót a módszerek relatív végrehajtási idejének jellemzésére.

Először a D_j adatkészlet minden egyes imputálandó x_{i,D_j} attribútumra 5-ször végzem el az imputációt, és az öt imputáció pontosságának (accuracy: az eltalált pótlások számának és az összes pótlás számának hányadosa) átlagát veszem. Ezt az átlagos pontosságot $acc(x_{i,D_j}, M_k)$-val jelöljük, ahol M_k a k-adik adatpótlási módszer.

Ezután minden egyes imputált értékre minden egyes adathalmazban meghatározom a **relatív pontosságot** úgy, hogy az egyes módszerek átlagos pontosságát elosztom a legnagyobb átlagos pontossággal (a legjobb módszer pontosságával):

$$acc_{rel}(x_{i,D_j}, M_k) = \frac{acc(x_{i,D_j}, M_k)}{max_k\{acc(x_{i,D_j}, M_k)\}} \ ,$$

ahol a max függvény argumentuma az összes módszer halmazán fut.

Az M_k módszer **összesített relatív pontossága** a relatív pontosságok átlagaként kerül kiszámításra, az összes adathalmaz összes imputált változójára vonatkozóan. Jelölje N az összes imputált érték számát az összes vizsgált adatkészletben. Az M_k *módszer összesített relatív pontossága* a következő képlettel határozható meg

$$Acc(M_k) = \frac{1}{N} \sum_{i,j} acc_{rel}(x_{i,D_j}, M_k).$$

Ennek a mértéknek a fő célja, hogy normalizálja az egyes módszerek nagyon különböző adathalmazokon mért eltérő pontossági értékeit. Így össze tudom a módszereket hasonlítani.

A végrehajtási idők vizsgálatára bevezetem az **összességében relatív végrehajtási idő**-t. Minden egyes imputált X_{i,D_j} attribútumra a D_j adatállományban 5 imputációt végzek, és az öt imputáció végrehajtási idejének átlagát veszem. Ezt az átlagos végrehajtási időt $t(x_{i,D_j}, M_k)$-val jelöljük, ahol M_k a k-adik adatpótlási módszer. Ezután minden egyes imputált attribútumra minden egyes adatkészletben meghatározzuk a **relatív végrehajtási idő**-t úgy, hogy az egyes módszerek átlagos végrehajtási idejét elosztjuk a leghosszabb átlagos végrehajtási idővel.

$$t_{rel}(x_{i,D_j}, M_k) = \frac{t(x_{i,D_j}, M_k)}{max_k\{t(x_{i,D_j}, M_k)\}} ,$$

ahol a max függvény argumentuma az összes módszer halmazán fut.

Az M_k módszer **összességében relatív végrehajtási idő** a relatív végrehajtási idők átlagaként kerül kiszámításra, az összes adathalmaz összes imputált változójára vonatkozóan.

Jelöljük N-nel az összes imputált érték számát az összes vizsgált adatkészletben. Az M_k *módszer összes relatív végrehajtási ideje* a következő képlettel határozható meg

$$T(M_k) = \frac{1}{N} \sum_{i,j} t_{rel}(x_{i,D_j}, M_k).$$

Ennek a mértéknek a fő célja, hogy normalizálja az egyes módszerek nagyon különböző adathalmazokon mért eltérő végrehajtási időértékeit. Így össze lehet őket hasonlítani.

4.5.2. Az adatkészletek rövid bemutatása

A kísérleti tesztek során a következő hat benchmark adatkészleten végeztem kísérleteket: a BankChurners, Pima Indians Diabetes és Saheart-adatkészletek a Kaggle weboldaláról [47] származnak; a WBCC és az Abalone adatkészletek az UCI ML Repositoryból [27] származnak; a Deals adatkészlete a Rapidminer [88] weboldalról származik. Három kulcsparaméter van: az attribútumok száma, az attribútumok típusa és az adatkészlet rekordjainak száma. Az adatkészlet méretét tekintve a legnagyobb adatkészlet 10127, míg a legrövidebb 462 rekordot tartalmaz. Az attribútumok száma 3 és 30 között van. A kiválasztott benchmark adatkészletek paraméterértékeit a 4.1. táblázat foglalja össze.

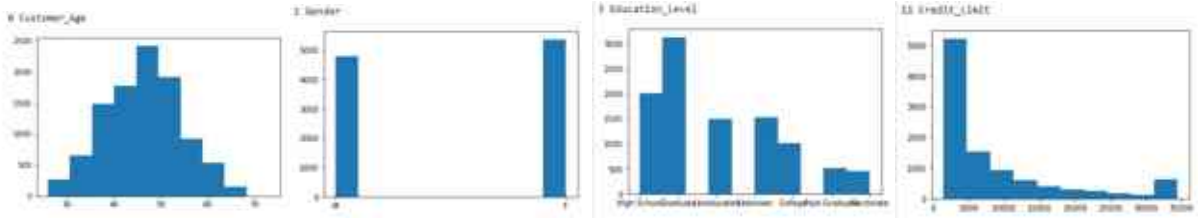
A következőkben néhány szemléltető adatpótlási példát mutatok be a vizsgált adatkészletekhez tartozó néhány attribútumon. A módszerek összehasonlítása nem korlátozódott ezekre a jellemzőkre. Az általános pontosságot az összes változóra nézve e szakasz végén mutatom be.

adatkészlet	n_{lines}	n_{attr}	adattípus
BankChurners	10127	20	kategorikus és folytonos
WBCC	569	30	folytonos
Pima Indians Diabetes	768	8	folytonos
Deals	1000	3	kategorikus és folytonos
Saheart	462	9	folytonos és kategorikus
Abalon	4177	9	folytonos és kategorikus

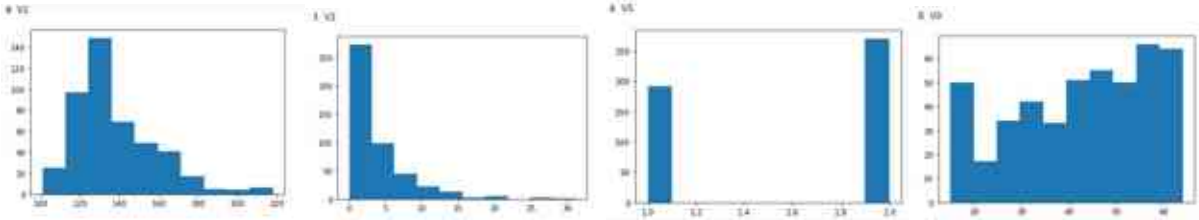
4.1. táblázat. Az adatkészletek paraméterei

A BankChurners-adatkészlet nagyon különböző típusú attribútumokat tartalmaz. Vannak bináris kategóriaértékkel rendelkező oszlopok (mint például az Attrition Flag vagy a Gender) és vannak folytonos attribútumok, mint például az ügyfél életkora. A kiválasztott attribútumok, az ügyfél életkora, a nem, az oktatási szint és a hitelkeret értékeloszlása a 4.2. ábrán látható. A Saheart-adatállomány esetében a bemutatásra kiválasztott változók az sbp, a dohány, a famhist és az életkor. Csak a harmadik elem kategorikus. Az értékhisztogramok a 4.3. ábrán láthatók. Az Abalone adatkészlet esetében a bemutatásra kiválasztott változók a nem (kategorikus), a hossz, a súly és a gyűrűk (egész szám). Az értékhisztogramok a 4.4. ábrán láthatók. A Pima Indians Diabetes adatállománya esetében a bemutatásra kiválasztott változók a terhességek száma (egész szám), a glükóz, a BMI és az életkor (egész szám). Az értékhisztogramok a 4.5. ábrán láthatók.

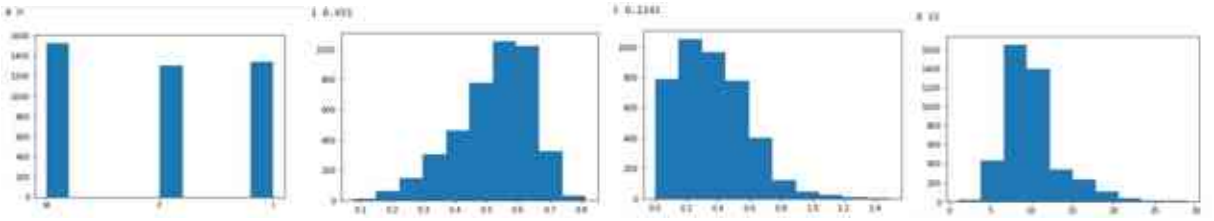
A hiányzó adatelemek generálására a véletlenszerűen hiányzó (MAR) módszert alkalmaztuk, vagyis annak valószínűsége, hogy egy megfigyelés hiányzik, függhet a megfigyelt halmaztól, de nem függ a nem megfigyelt résztől.



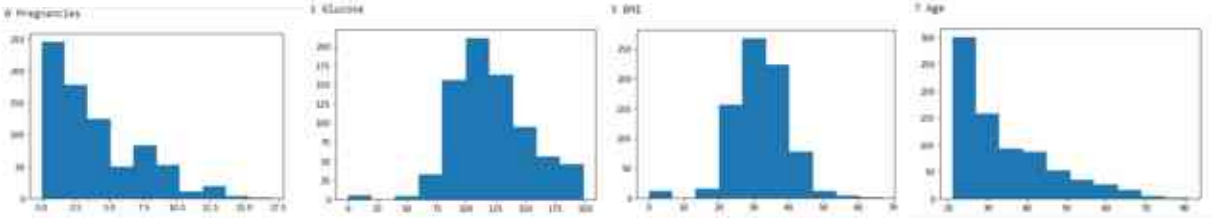
4.2. ábra. Az életkor, nem, iskolai végzettség és hitelkeret hisztogramjai a BankChurners-adatkészletben



4.3. ábra. A spb, dohány, családtörténet, kor hisztogramjai a Saheart-adatállományban



4.4. ábra. A nem, hosszúság, tömeg és gyűrűk hisztogramjai az Abalone-adatkészletben



4.5. ábra. A terhességek, a glükóz, a BMI és az életkor hisztogramjai a Pima Indians Diabetes adatállományban

4.6. A bevezetett adatpótlási módszerek eredményei

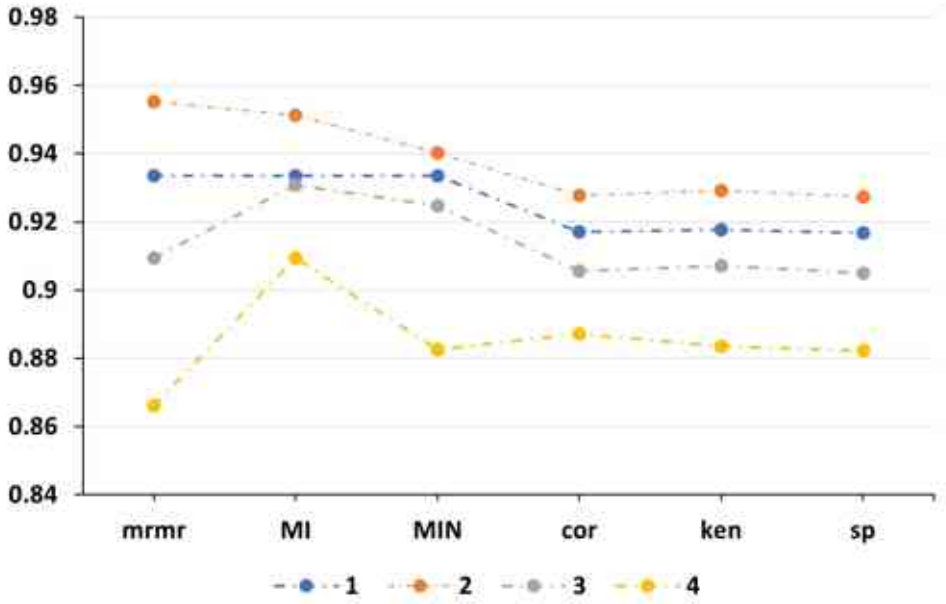
Ez az alfejezet az imputációk eredményeinek értékelését és a kiválasztott, R csomagokban implementált módszerekkel való összehasonlítást tartalmazza. Itt adom meg a végrehajtási idők összehasonlítását is.

A következő táblázatokban összefoglalom a kiválasztott benchmark-adatkészleteken a legfontosabb mérési eredményeket mind a pontosság, mind a végrehajtási idő tekintetében. A kiinduló adatkészletek teljesek voltak, a módszereink tesztelésére a vizsgált attribútum értékeinek 10%-át, illetve 50%-át véletlenszerűen töröltem. Minden kísérlet esetében a hiányzások véletlenszerűen lettek kiválasztva, de a különböző módszereket ugyanazokon a hiányos adathalmazokon teszteltem.

4.6.1. Pontossági (accuracy) tesztek

A 4.6. ábrán látható, hogy az összesített relatív pontosság (y tengely) hogyan függ a módszerektől (x tengely) és a kiválasztott magyarázó változók számától (különböző színek). Jól látható, hogy az attribútum kiválasztó új módszerek mindegyikén a 2 magyarázó változó kiválasztása teljesít a legjobban. Ezért használok az illusztratív példákban $k = 2$ magyarázó változót. Azt is észrevehetjük, hogy az 1, 2 és 3 magyarázó változó esetén az információ tartalmon alapuló módszerek teljesítenek a legjobban és ezeknél még az egy magyarázó változó kiválasztás is jobban teljesít, mint a korreláció alapú kiválasztó módszerek 2 magyarázó változóval.

A pontossági vizsgálatoknál az értékeket %-os értékben adjuk meg. Felhívjuk a figyelmet arra, hogy a knn, NB, mice és mF módszerek különböző R csomagokhoz tartoznak, és bemenetként minden magyarázó változót megkaptak. Az oszlopok a következő jelöléseket tartalmazzák:



4.6. ábra. Az összesített relatív pontosság függése a módszerektől és a magyarázó változók számától (színek)

1. Tcol: a célattribútum azonosítója, a 4.5.2. részben felsorolt 4-4 attribútum.
2. mrmr_2 : a maximális relevancia minimális redundancia módszerén alapuló attribútumcsökkentés (a legjobb 2 attribútum).
3. MI_2 : a kölcsönös információn alapuló attribútumcsökkentő módszer (legjobb 2 attribútum)
4. MIN_2 : MI-alapú attribútumcsökkentés (legjobb 2 attribútum)
5. cor_2 : Pearson-korreláción alapuló attribútumcsökkentés (legjobb 2 attribútum)
6. ken_2 : Kendall-korreláción alapuló attribútumcsökkentő módszer (legjobb 2 attribútum)
7. sp_2 : Spearman-korreláción alapuló attribútumcsökkentő módszer (legjobb 2 attribútum)
8. knn : k-NN osztályozási adatpótlási módszer
9. NB : Naive Bayes osztályozóval adatpótlás
10. mice : Többváltozós adatpótlás láncolt egyenletek segítségével (Multivariate Imputation by Chained Equations)
11. mF : MissForest véletlen erdő adatpótlási módszer

A 4.2., 4.3., 4.4. és 4.5. táblázatban láthatók a korábban bemutatott attribútumokra vonatkozó $acc(x_{i,D_j}, M_k)$ pontossági eredmények a BankChurners, Saheart, Abalone, Pima Indians adathalmazokban. Mindegyik esetben 10%-os, illetve 50%-os hiányossággal. Az oszlopok nevei az adatpótlási módszereket jelölik. A pontossági értékek százalékban vannak megadva. A legmagasabb értékek félkövérrel vannak szedve.

hiányzó értékek százalékos aránya=10%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	89	89	89	87	87	87	92	88	85	**95**
1	**43**	**43**	**43**	39	39	39	33	38	28	40
3	**36**	**36**	**36**	26	27	27	29	32	22	34
11	33	33	33	**34**	**34**	**34**	32	22	27	33

hiányzó értékek százalékos aránya=50%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	90	90	90	87	87	87	92	88	86	**95**
1	**43**	**43**	**43**	39	39	39	33	39	29	39
3	34	**35**	**35**	26	27	27	29	32	22	34
11	32	33	33	33	33	33	33	22	27	**34**

4.2. táblázat. A kiválasztott változók adatpótlásának pontosságai a BankChurners-adatkészletben

hiányzó értékek százalékos aránya=10%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	27	28	28	28	28	28	23	**30**	20	23
1	**49**	48	48	48	48	48	41	46	44	40
4	**59**	**59**	57	57	56	55	26	55	49	24
8	18	17	23	17	17	17	23	17	17	**25**

hiányzó értékek százalékos aránya=50%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	23	22	22	22	22	22	**25**	**25**	20	23
1	**49**	48	48	48	48	48	45	42	48	44
4	55	57	56	58	57	57	24	**61**	57	25
8	18	20	20	**21**	**21**	**21**	20	19	18	20

4.3. táblázat. A kiválasztott változók adatpótlásának pontosságai a Saheart-adatkészletben

Annak érdekében, hogy jó áttekintést és átfogó képet kapjunk a módszerek jóságáról, az *összesített relatív pontosság* értékeit a 4.7. ábra és a 4.6. táblázat mutatja be. Látható, hogy a 2 magyarázó attribútumot kiválasztó új módszerek jobbak, mint az R csomagok vizsgált módszerei. Az új attribútum kiválasztási módszerek közül azok a legjobbak, amelyek az információtartalom kritériumai alapján lettek kiválasztva.

A vizsgált módszerek összesített relatív pontosságának összehasonlítása látható a 4.8. és a 4.9. ábrán. Az első oszlopdiagram a módszerek, a második oszlopdiagram az adatkész-

hiányzó értékek százalékos aránya=10%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	**55**	52	52	**55**	**55**	**55**	44	50	45	46
1	**78**	**78**	**78**	**78**	**78**	**78**	76	66	66	**78**
5	**72**	67	67	71	71	71	57	68	49	57
8	53	**55**	**55**	54	54	54	44	15	39	51

hiányzó értékek százalékos aránya=50%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	**55**	53	53	54	54	54	45	51	46	46
1	77	77	76	77	77	77	76	65	66	**78**
5	**71**	67	67	69	69	69	54	65	47	55
8	**54**	**54**	**54**	**54**	**54**	**54**	48	17	39	51

4.4. táblázat. A kiválasztott változók adatpótlásának pontosságai az Abalone-adatkészletben

hiányzó értékek százalékos aránya=10%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	35	**37**	35	35	36	36	25	26	26	32
1	28	28	29	28	28	28	30	12	26	**32**
5	**46**	44	42	44	44	45	43	27	34	44
7	45	**46**	45	45	**46**	**46**	34	39	38	33

hiányzó értékek százalékos aránya=50%

Tcol	mrmr_2	MI_2	MIN_2	cor_2	ken_2	sp_2	knn	NB	mice	mF
0	33	**34**	33	33	33	33	27	29	26	29
1	23	25	24	25	25	25	30	15	23	**31**
5	**43**	42	**43**	42	42	42	41	23	31	**43**
7	44	**45**	44	**45**	**45**	**45**	37	34	32	34

4.5. táblázat. A kiválasztott változók adatpótlásának pontosságai a Pima Indians Diabetes adatkészletben

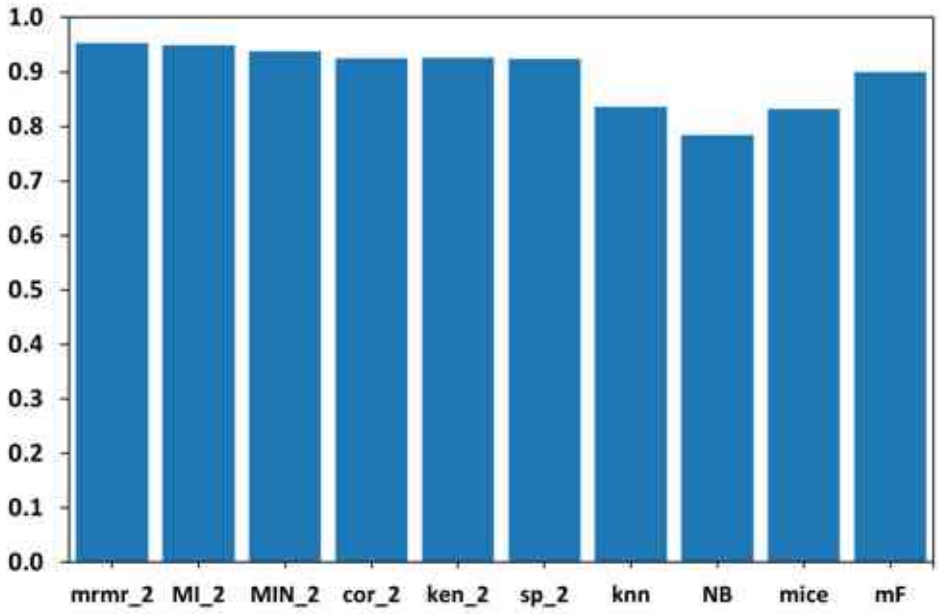
letek szerint csoportosítva mutatja az adott módszer, adott adatkészlet és adott százalékos hiányzás adatpótlási eredményeit.

4.6.2. Végrehajtási idő tesztek

Figyelembe véve a vizsgált módszerek kivitelezési költségeit, nagyon eltérő költségtartományokat láthatunk. Általánosságban elmondható, hogy a knn és missForest lényegesen több időt igényelnek a modellalkotáshoz, mint a többi modell. Egy másik jellemző tapasztalat, hogy a csökkentett attribútumkészletű modellek lényegesen kevesebb végrehajtási költséget igényelnek a modellkészítéshez, mint a teljes adatkészlettel rendelkező modellek. Ez a tény az attribútumcsökkentés másik fontos előnyét mutatja. A 4.10. ábra a módszerek összesített relatív végrehajtási idejét mutatja. A legidőigényesebb módszer a MissForest (mF) volt. Az R csomagokhoz tartozó módszerek közül a Naive Bayes kevésbé időigényes. Az attribútum kiválasztáson alapuló új módszerek kevésbé időigényesek, mint

Módszer	Összesített relatív pontosság
mrmr_2	0,95
MI_2	0,95
MIN_2	0,94
ken_2	0,93
cor_2	0,92
sp_2	0,92
mF	0,90
knn	0,84
mice	0,83
NB	0,78

4.6. táblázat. A vizsgált módszerek összesített relatív pontosság értékei

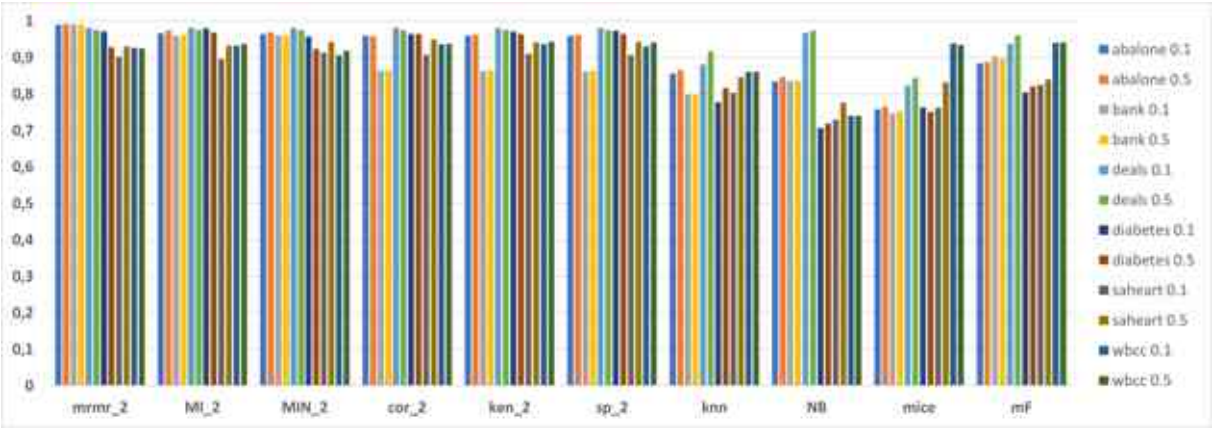


4.7. ábra. A vizsgált módszerek összesített relatív pontosságának összehasonlítása

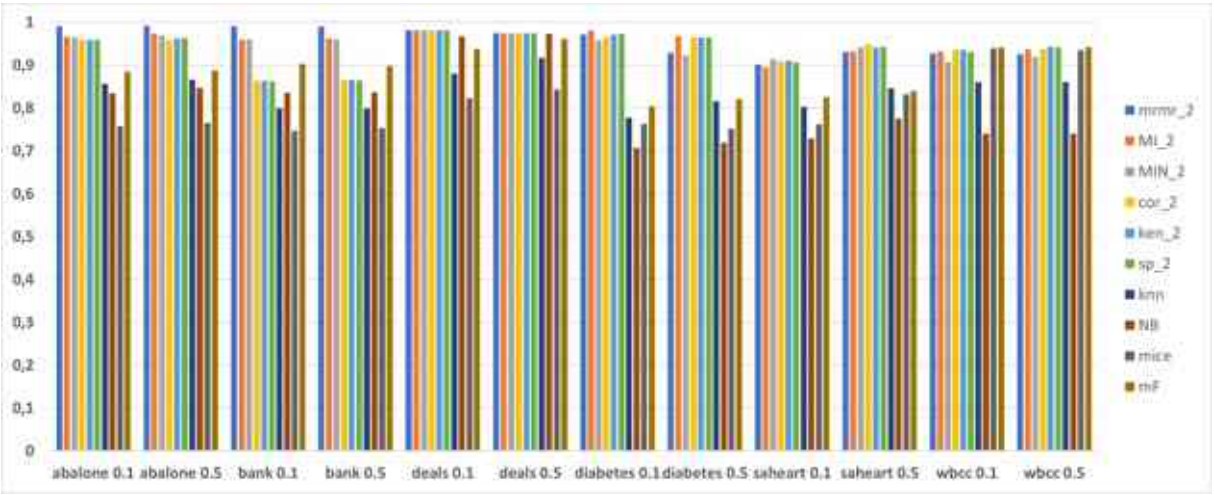
a knn, mice és mF.

4.7. Összegzés

Ebben a fejezetben attribútum kiválasztáson alapuló valószínűségi gépi tanulási adatpótlási módszereket mutattam be. A módszereket diszkrét adatokra fejlesztettem ki, de kategorikus és folytonos adatokra is alkalmazhatók. A folytonos adatokat az előfeldolgozási fázisban diszkretizálom. Ha az imputált változó folytonos, eredményként egy intervallumot kapunk, amelyhez az imputált érték tartozik (egy adott szakterület szakértői számára ez néha nagyon hasznos információ lehet), vagy az imputált érték a kapott intervallumhoz tartozó értékek mediánja vagy átlaga. Az információelméleti attribútum



4.8. ábra. A vizsgált módszerek összesített relatív pontosságának összehasonlítása a módszerek szerint



4.9. ábra. A vizsgált módszerek összesített relatív pontosságának összehasonlítása az adatkészletek szerint

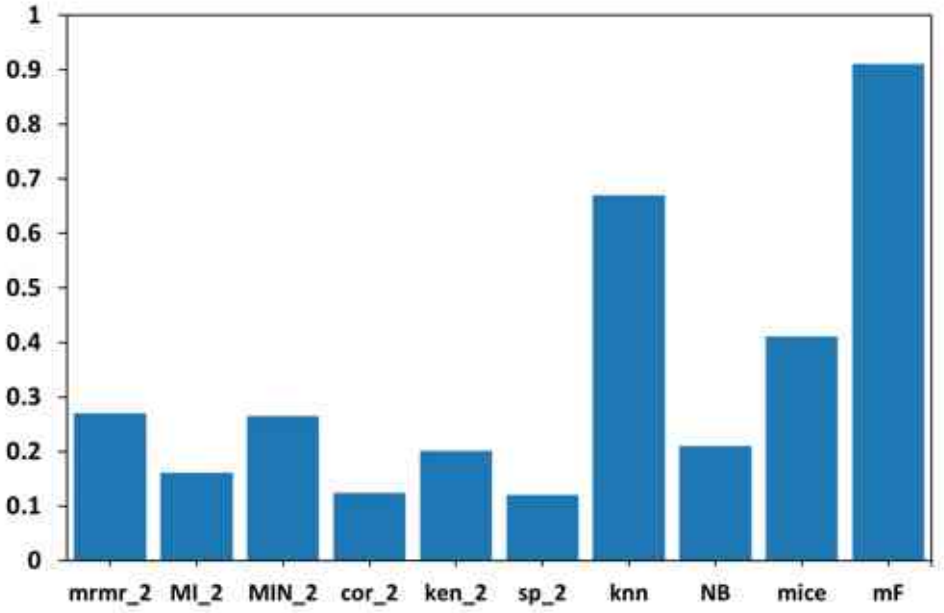
kiválasztáson alapuló módszerek kategorikus változókra is alkalmazhatók, mivel csak a valószínűségi eloszlástól függnek, nem pedig a felvett értékektől.

A fejezetben különböző adatadatpótlási módszerek hatékonysági elemzését mutattam be, az egyszeri adatpótlási megközelítésre összpontosítva. A bevezetett módszereimet összehasonlítottam négy, az R programcsomagokban megvalósított módszerrel.

A különböző módszerek különböző adathalmazokon történő összehasonlítása érdekében bevezettem két mutatót. Az egyik az általános relatív pontosság, amely a hiányzó adatok különböző módszerekkel való pótlásának pontosságát összehasonlíthatóvá teszi különböző adathalmazokon. A másik az általános relatív végrehajtási idő, amely a hiányzó adatok különböző módszerekkel való pótlásának végrehajtási idejét teszi összehasonlíthatóvá különböző adathalmazokon.

Az eljárást implementáltam.

Az elvégzett összehasonlító tesztek fő következtetése, hogy a javasolt módszereim minden kategóriában jobbnak bizonyultak. A valószínűségi gépi tanulás esetében az információtartalom-alapú módszereim bizonyultak a legjobbnak, de fontos megjegyezni,



4.10. ábra. A vizsgált módszerek összesített relatív végrehajtási idejének összehasonlítása

hogy a hagyományos korrelációs együtthatókon alapuló jellemzőválasztás is hatékony volt. Az időköltséget tekintve a valószínűségi gépi tanulási módszereim jellemzően gyorsabbak voltak.

Az elvégzett elemzések és tesztek azt mutatják, hogy a javasolt adatpótlási valószínűségi gépi tanulási technikák egyértelműen előnyösebbek a szokásos gépi tanulási technikáknál.

5. tézis: Attribútum kiválasztáson alapuló valószínűségi gépi tanulási adatpótlási módszereket vezettem be. A módszereket diszkrét adatokra fejlesztettem ki, de kategorikus és folytonos adatokra is alkalmazhatóak.

A fejezetben különböző adatadatpótlási módszerek hatékonysági elemzését mutattam be, az egyszeri adatpótlási megközelítésre összpontosítva. A bevezetett módszereimet összehasonlítottam négy, az R programcsomagokban megvalósított módszerrel.

A különböző módszerek különböző adathalmazokon történő összehasonlítása érdekében bevezettem két mutatót. Az egyik a pontosságot, a másik a végrehajtási időt jellemzi.

A valószínűségi gépi tanulás esetében az információtartalom-alapú attribútumcsökkentési módszereim bizonyultak a legjobbnak, de a hagyományos korrelációs együtthatókon alapuló jellemzőválasztás is hatékony volt. Az időköltséget tekintve a valószínűségi gépi tanulási módszereim jellemzően gyorsabbak voltak.

Az elvégzett elemzések és tesztek azt mutatják, hogy a javasolt adatpótlási valószínűségi gépi tanulási technikák egyértelműen előnyösebbek a vizsgált beépített gépi tanulási technikáknál.

Kapcsolódó publikációk: [**S3**]

5. fejezet

Prefix-fa felépítés költségelemzése

Nagy mennyiségű adat tárolása, elemzése során gyakran alkalmazott struktúrák a listák és a halmazok. A különböző feladatok megoldása során fontos, hogy a tárolást és a keresést hatékony algoritmusokkal oldjuk meg. Többféle tárolási lehetőséget szoktak alkalmazni, mint például a relációs adatbázisok, listák, kereső fák és a prefix-fák. A fastruktúra hatékony tárolást és adatkezelést biztosít a különböző adatlistákhoz vagy adatkészletekhez. A prefix-fa egy speciális fastruktúra az adatelemek rendezett listáinak tárolására, széles körben használt adatstruktúra számos alkalmazási területen. A prefix-fában a közös előtagrésszel rendelkező listák ugyanazon az útvonalon osztoznak. Mivel a prefix-fa számos adatmanipulációs algoritmusban részt vehet, a fa felépítésének költségbecslése fontos összetevője a teljes adatmanipulációs algoritmus költségfüggvényének. Jelen fejezetben a prefix-fa alapú tárolás hatékonyságának azon részével foglalkozom, amely egy véletlenszerű objektumlistákból álló bemeneti halmazra nyújt költségelemzést a fa méretére vonatkozóan. Az elemzés analitikus és szimulációs módszereket egyaránt tartalmaz, és a bemutatott fő eredmény egy gamma-eloszláson alapuló közelítő függvény.

A rendezett lista egy jó megvalósítási struktúra a halmazok ábrázolásához. Az elemek rendezése lehetővé teszi a halmazműveletek hatékonyabb végrehajtását. Például a halmazok metszése egy rendezett egyesítési algoritmussal valósítható meg. Ezt a fajta megvalósítást használják többek között a relációs adatbázisok lekérdezőmotorjában egy belső összekapcsolási művelet végrehajtása során. Így az alábbi teljesen rendezett objektumhalmazzal dolgozunk:

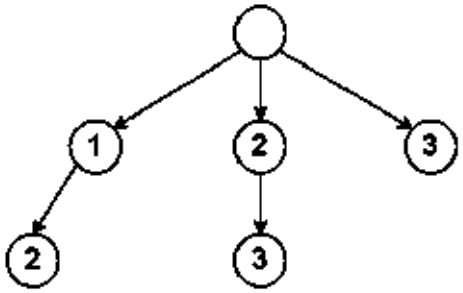
$$\langle Oo_i, \leq \rangle .$$

Az O alaphalmaz méretét M-mel jelöljük. Egy objektumhalmaz az alábbi rendezett listával ábrázolható:

$$l = o_1, ..., o_n , \qquad \text{ahol } o_{i-1} \leq o_i.$$

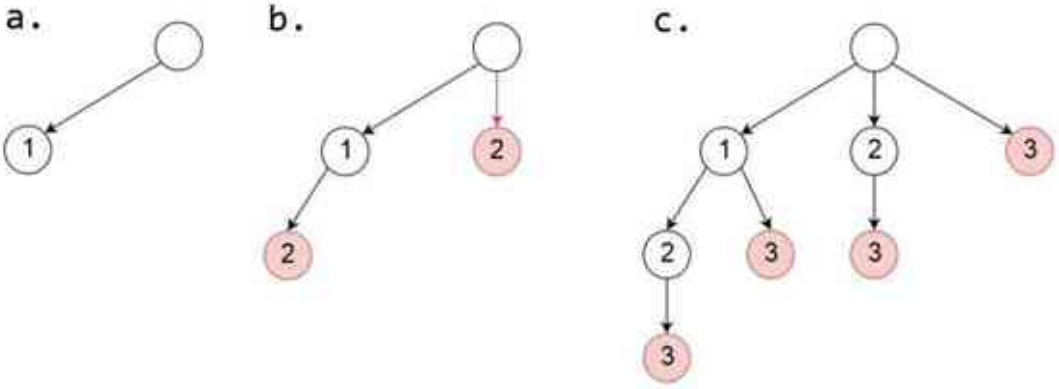
Az $L = \langle l_j \rangle$ listák (szekvenciák) halmaza egy speciális fával, a T prefix-fával ábrázolható [39], ahol minden l listát a gyökérelemtől kiinduló úttal reprezentálunk. Az O alaphalmazon értelmezett listákból felépített prefix-fát T_O-val jelöljük. A hasonló ösvények a fában ugyanazon a szakaszon lesznek. A fában minden csomópontot egy $o \in O$ elemhez rendelünk, kivéve a gyökeret, amely üres csomópont. A fában minden csomópont a gyökértől induló ösvény mentén lévő csomópontokhoz kapcsolódó objektumsorozatot képvisel. Mivel egyes listák allistákként más listákba is belekerülhetnek, a csomópontokban egy speciális jelzőre van szükségünk a listák végszimbólumának jelölésére. Egy

adott $o_1, ..., o_n$ lista akkor szerepel T-ben, ha létezik olyan útvonal, ahol az útvonal i-edik eleméhez o_i tartozik és az o_n-hez kapcsolódó csomópontnál a végcsomópont-jelző 1-re van állítva. Az 5.1. ábrán az (1,2),(2,3),(3) szekvenciák prefix-fa reprezentációja látható.



5.1. ábra. Minta egy prefix-fára

A T prefix-fa méretén a benne lévő pontok számát értjük. A T_O prefix-fa teljes, ha az O alaphalmaz minden lehetséges szekvenciáját reprezentálja. Az M elemű O alaphalmazon a teljes prefix-fa elemeinek száma 2^M. Ez teljes indukcióval könnyen belátható. Ha $M = 1$, a teljes prefix-fa mérete 2^1 (5.2. a. ábra). Ha az O alaphalmazhoz felveszünk egy új elemet, akkor a fa minden csomópontjából kell egy leágazást csatolni az új elemmel, vagyis egy újabb levél keletkezik. Így minden lépésben a fa mérete megduplázódik. Ezeket a lépéseket láthatjuk az 5.2. ábrán. A következő alfejezetben az 5.2. b. ábrához tartozó $M = 2$ esetről lesz szó.



5.2. ábra. A teljes prefix-fa felépítése

Általában az adatmanipulációs műveletek költségei az adatszerkezet méretétől függenek. Így a prefix-fa mérete fontos tényező az adatmanipulációs algoritmusok költségbecslésében. Mivel az adatfa mérete a bemeneti lista halmazától függ, a teljes költség is a bemeneti adatok függvénye. Az adathalmazt figyelembe véve két döntő tényező van az adatgenerálásban:

- a szekvenciák száma,

- tárolandó szekvenciák eloszlása.

A fa struktúrák leszámlálása intenzíven vizsgált téma a szakirodalomban. A fák általános komplexitáselemzésével kapcsolatban számos publikációt találunk, mint például [11], [14], [15]. Ezzel szemben a prefix-fa struktúrára, mint speciális fára vonatkozóan leszámlálással foglalkozó tanulmányok nincsenek az irodalomban. Hozzájárulásom célja, hogy egy kezdeti elemzést adjak a prefix-fa struktúrák méretkomplexitásáról.

Vizsgálatomban első lépésként egy függetlenségi feltételezést alkalmazok, ami azt jelenti, hogy a listában lévő o_i elem létezésének valószínűsége független bármely más o_j elem valószínűségétől. A bemeneti adatok generálásának fő paraméterei a következők:

- M: az O alaphalmaz elemeinek száma,
- K: a bemeneti adathalmaz szekvenciáinak száma,
- p_i: az o_i elem valószínűsége,
- $p_O = (p_1, ..., p_M)$: az O adathalmazra vonatkozó valószínűség vektor.

5.1. A prefix-fa méretének költségelemzése

Munkámban két alapvető megközelítést elemzek. Az első, egzakt valószínűségi képletek alkalmazásán alapul a generált prefix-fák méretének meghatározására. Ez a módszer pontos értékeket szolgáltat, de a képletek túl bonyolulttá válnak, ha $M > 2$. A második módszer a prefix-fák felépítéséhez szimulációt használ a véletlen szekvenciák generálására. A generált prefix-fák méretét statisztikai módszerekkel vizsgálom. A tapasztalatok alapján ez a megközelítés alkalmasabb a fa méretének közelítésére nagyobb adathalmazok esetén is.

Felsorolási képlet

Az $M = 2$ esetre az analitikus, egzakt képletek egyszerű módon konstruálhatók. A fa méretére vonatkozó képlet a következő összetevőkből épül fel.

- $P_0 = (1 - p_a)(1 - p_b)$, annak valószínűsége, hogy a két elem közül egyik sem fordul elő a listában
- $P_a = p_a(1 - p_b)$, annak valószínűsége, hogy csak az a elem fordul elő a listában
- $P_b = (1 - p_a)p_b$, annak valószínűsége, hogy csak a b elem fordul elő a listában
- $P_{ab} = p_a \cdot p_b$, annak valószínűsége, hogy mindkét elem, az a és a b elem is előfordul a listában.

Igazolható, hogy

$$P_0 + P_a + P_b + P_{ab} = (1 - p_a)(1 - p_b) + p_a(1 - p_b) + (1 - p_a)p_b + p_a p_b = 1$$

Az eredményül kapott prefix-fa méretét figyelembe véve kiszámíthatók a különböző méretértékek valószínűségei. Az i méret valószínűségét P_i-vel jelöljük, és értéke a következő módon számítható ki:

$$P_1 = P_0^K$$

$$P_2 = \sum_{i=1}^{K} \binom{K}{i} P_a^i P_0^{K-i} + \sum_{i=1}^{K} \binom{K}{i} P_b^i P_0^{K-i}$$

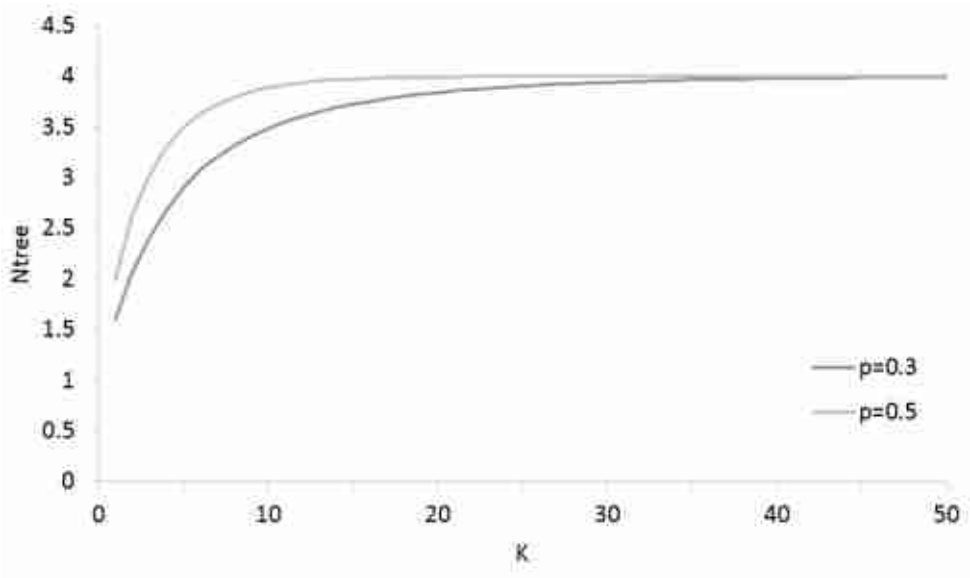
$$P_3 = \sum_{i=1,j=1}^{K} \binom{K}{i}\binom{K-i}{j} P_a^i P_b^j P_0^{K-i-j} + \sum_{i=1,j=0}^{K} \binom{K}{i}\binom{K-i}{j} P_a^j P_{ab}^i P_0^{K-i-j}$$

$$P_4 = \sum_{i=1,j=1,l=0}^{K} \binom{K}{i}\binom{K-i}{j}\binom{K-i-j}{l} P_a^l P_{ab}^i P_b^j P_0^{K-i-j-l}$$

A fa méretének átlagos értékét a következőképpen számíthatjuk ki:

$$E[Ntree] = P_1 + 2 \cdot P_2 + 3 \cdot P_3 + 4 \cdot P_4. \tag{5.1}$$

A pontos képleteket Matlabban számoltam ki különböző valószínűségi értékekre. Az egyszerűség kedvéért mindkét elem valószínűsége azonos volt: $p_a = p_b$. Az így kapott méretfüggvényt az 5.3. ábra mutatja.



5.3. ábra. Az átlagos méret különböző valószínűségek esetén (M=2 és $p_a = p_b = p$)

Mivel tetszőleges M esetén a 5.1. képletben szereplő összeadandó tagok száma 2^M-re nő, ez a megközelítés nem használható nagyobb problémák elemzésére. Így az egzakt felsorolási formulák alkalmazása nagyon korlátozott.

5.2. Költségelemzés szimulációval

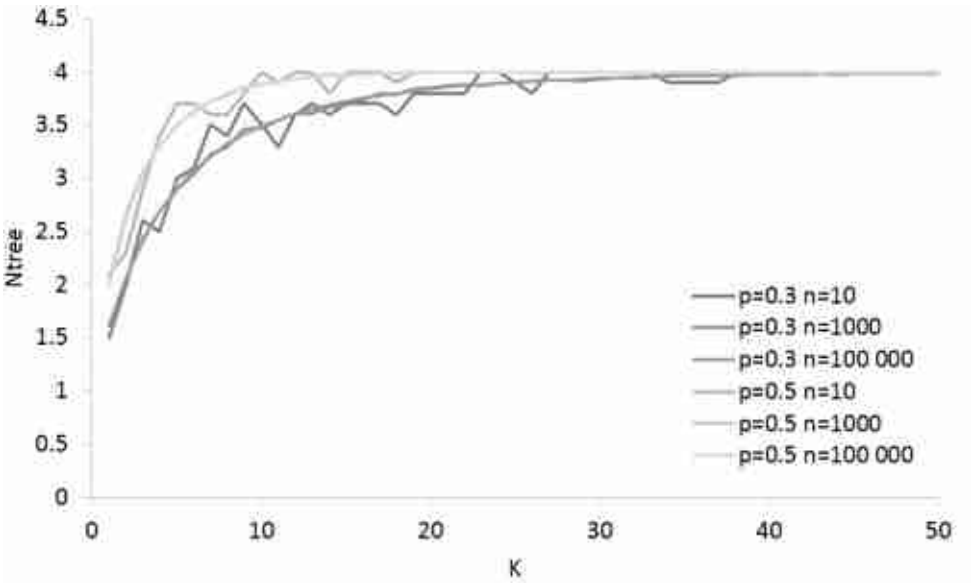
A szimulációban az előtagfákat véletlenszerű listákból építjük fel. A véletlen listák felépítésére szolgáló generáló függvény a Monte Carlo-módszeren alapul, és három paramétertől függ:

- M: az O alaphalmaz elemeinek száma,
- K: A bemeneti adathalmaz szekvenciáinak száma,
- p: az elemek listában való megjelenésének közös valószínűsége.

A szimuláció célja a fa mérete és a bemeneti paraméterek közötti függőség meghatározása. Az implementációban a szekvenciákat bináris mátrix formájában tárolom, ahol az oszlopok az elemeket, a sorok pedig a generált szekvenciákat jelölik. Az $m(i, j)$ mátrixelem értéke 1, ha az i-edik lista tartalmazza a j-edik elemet, ellenkező esetben az érték 0.

A fák mérete a következő algoritmussal számolható ki:

- a bemeneti K darab szekvencia rendezése a szekvenciában lévő elemek száma szerint
- a más szekvenciában szereplő szekvenciák kiszűrése
- faépítés a redukált szekvenciák halmazából, a prefix-fa méretének meghatározása



5.4. ábra. A szimulációval közelített költségfüggvény (N_{tree}) a K függvényében. $M = 2$; $p = 0,3$ és $0,5$; $N = 10; 1000; 100000$

A szimulációban n az elvégzett tesztek számát jelöli. A tesztek eredményeinek átlagát a méretfüggvény megalkotásához használom. A tesztek által $M = 2$ esetén kapott empirikus költségfüggvényt az 5.4. ábra mutatja.

Amint az várható volt, a szimulált költségfüggvény nagyobb n értékek esetén simább, mint a kis n értékek esetén. A vizsgált paramétertartományokban a szimulált költségfüggvények nagyon hasonlóak a pontosan számított költségfüggvényekhez. A szimulált és a számított költségértékek különbségeként számított megfelelő hibaértékeket az 5.1. táblázat tartalmazza.

5.1. táblázat. A szimulált költségfüggvény hibája

	N=10	N=1000	N=100000
p=0,3	0,6981146	0,098185	0,010077
p=0,5	0,549177	0,064796	0,007265

Nagyobb M és K értékek esetén, mivel a szimuláció futási ideje jelentősen megnő, csak kisebb számú teszt készült. A tesztsorozatban vizsgált paramétertartományokat az 5.2. táblázat tartalmazza.

5.2. táblázat. A szimulációkban vizsgált paramétertartományok

	min	max	step
M	5	50	5
K	200	12000	200
p	0,3	0,5	0,2

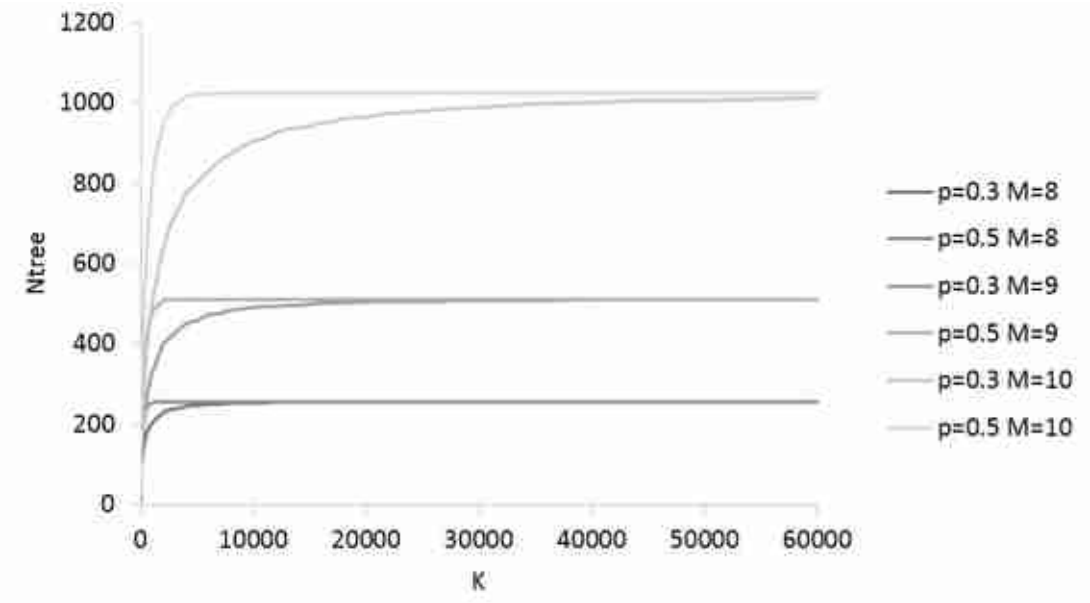
Az 5.5. és az 5.6. ábrán a fa mérete látható a K függvényében különböző M értékek esetén. Az 5.5. ábrán látható, hogy nagyobb K érték esetén a prefix-fa teljes lesz, és a költség állandó marad. A költségfüggvény 5.6. ábrán látható alakja azt mutatja, hogy a vizsgált paramétertartomány még mindig messze van a telítési küszöbtől.

A telítési küszöb értékét, vagyis, ahol $Ntree$ először eléri a 2^M értéket, az M érték függvényében is megvizsgáltam. Az eredményt az 5.7. ábra mutatja. Tapasztalataim szerint a K_t telítési érték az M exponenciális függvénye, és a növekedés nagyobb $p = 0,3$ esetén, mint $p = 0,5$ esetén.

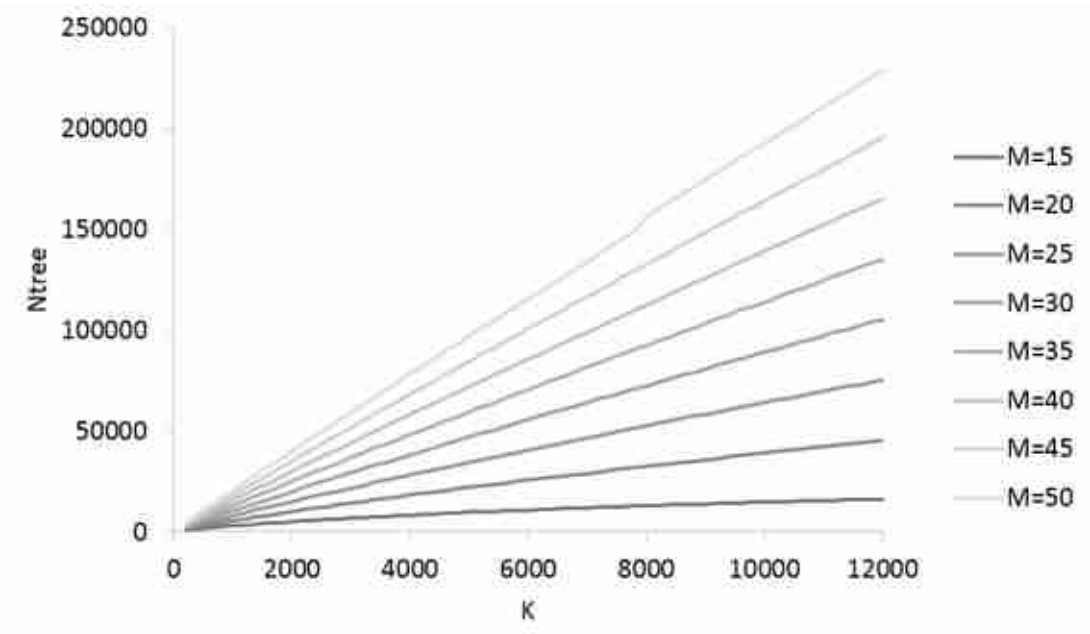
5.3. A költségfüggvény közelítése

A megfelelő analitikus közelítő függvény megtalálása érdekében először normalizáltam a kísérleti függvényeket. Az y tengelyen történő normalizálás azt jelenti, hogy a költségértéket a [0,1] intervallumba transzformáljuk, a 2^M-nel való osztással. Az x tengely mentén is elvégeztem a 2^M-nel való osztást. Így a költségfüggvény alakját az M aktuális értékétől független formára alakítottam. Az kapott grafikonok segítségével a költségfüggvényt minden M értékre azonos módon írhatjuk le.

A normalizált költségfüggvények elemzése azt mutatja, hogy közös alakjuk van, amely nagyon hasonlít a gammaeloszláshoz. Az 5.8. a normalizált költségfüggvény formáját mutatja a normalizált K érték függvényében. A gammaeloszlás egy folytonos valószínűségi



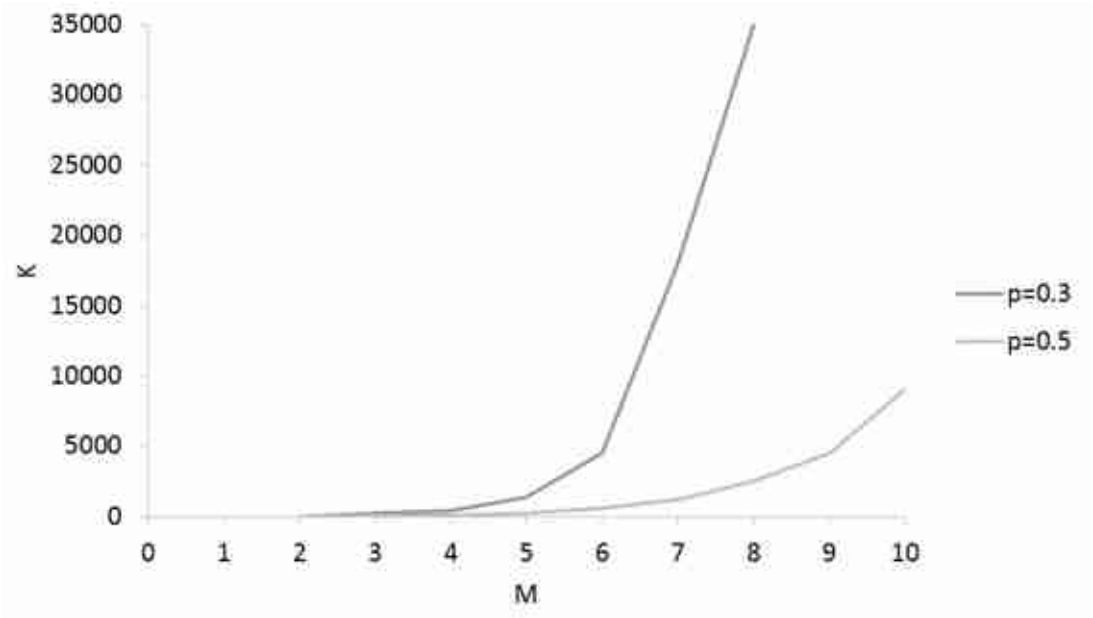
5.5. ábra. Szimulált költségfüggvény a telítési küszöbhöz képest nagy K értékekre



5.6. ábra. Szimulált költségfüggvény a telítési küszöbhöz képest kis K értékekre ($p = 0,5$)

eloszlás két paraméterrel [56]. A két paraméter a (k) alakparaméter és a (θ) skálaparaméter. A gammaeloszlás valószínűségi sűrűségfüggvénye a következő képlettel adható meg:

$$f(x, k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)},$$



5.7. ábra. A telítettségi érték (K_t) az M függvényében ($p = 0,3sp = 0,5$)

ahol

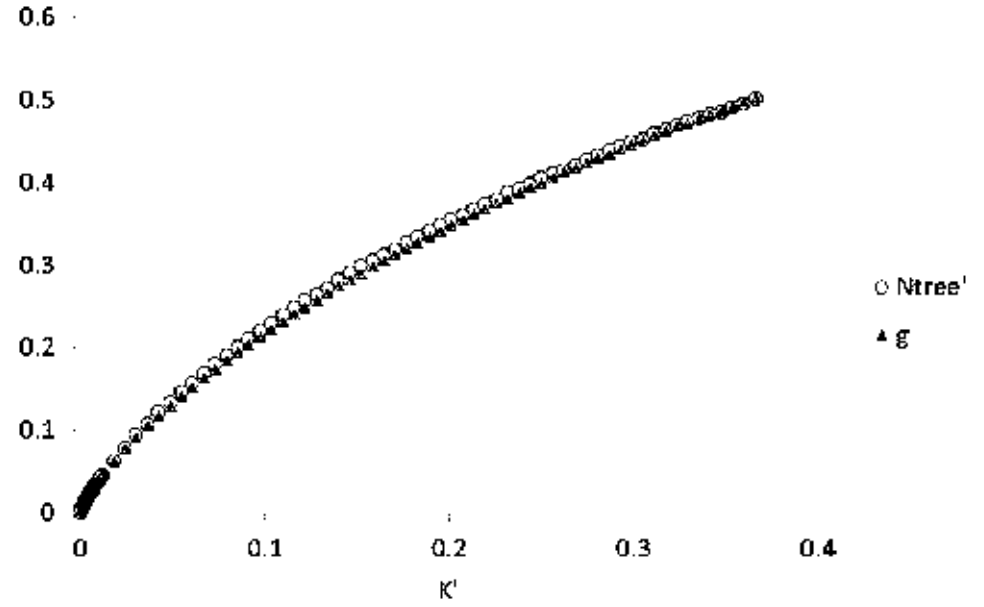
$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx.$$

A megfelelő eloszlásfüggvény a következőképpen adható meg:

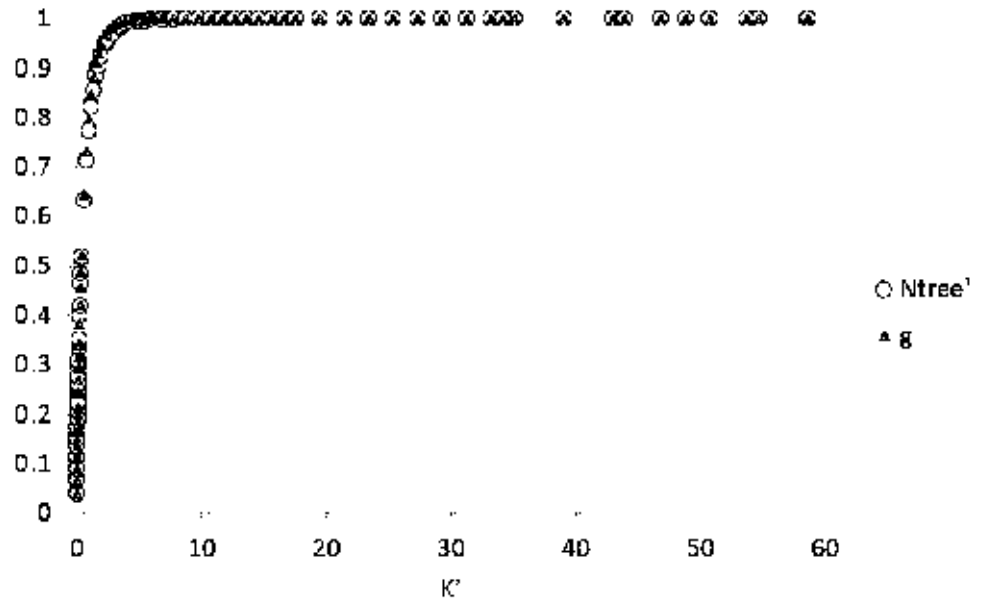
$$P(k, \theta x) = \frac{1}{\Gamma(a)} \int_0^{\theta x} z^{k-1} e^{-z} dz.$$

A közelítő algoritmust Matlabban implementáltam a gamcdf függvény segítségével [26].

A (k, θ) regressziós módszerrel a legjobban illeszkedő $f(x, k, \theta)$ gammaeloszlás meghatározásához a kísérleti költségfüggvények nagyon jó közelítését lehetett elérni. Az 5.8., az 5.9. és az 5.10. ábrán a közelítő gammaeloszlásfüggvények háromszöggel vannak jelölve.



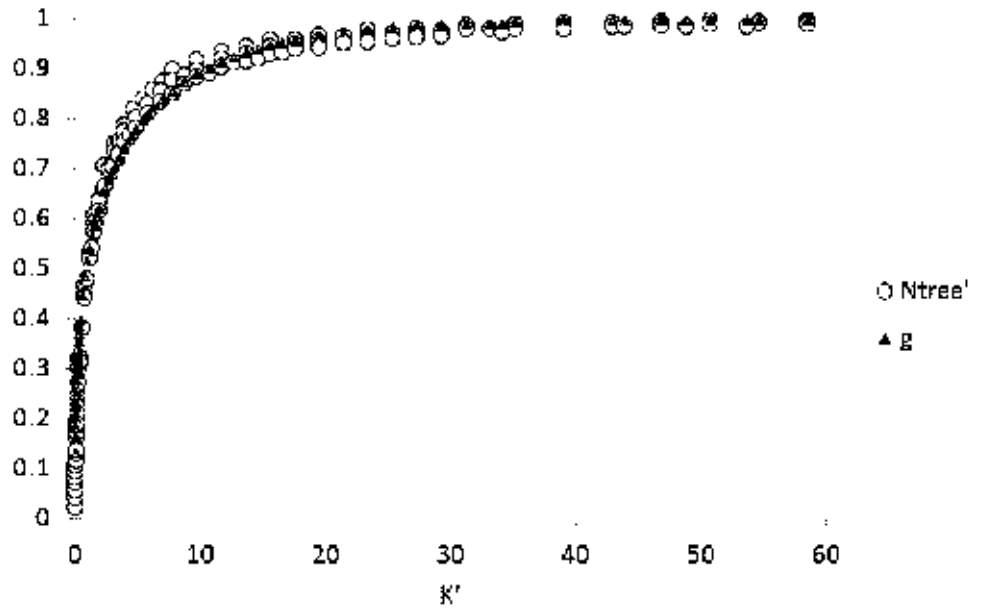
5.8. ábra. A normalizált költségfüggvény a normalizált K értékek függvényében, M=15-50 p=0,5; k=0,75; $\theta = 0,8$



5.9. ábra. A normalizált költségfüggvény és a gamma közelítés a normalizált K értékek függvényében, M=8; 9; 10; p=0,5; k=0,75; $\theta = 0,8$

5.4. Összegzés

Az elvégzett elemzésben analitikus és kísérleti úton vizsgáltam a prefix-fa méretfüggvényét a bemeneti adatok méret- és értékeloszlásától való függésben. A vizsgálati eredmények azt mutatják, hogy a szimuláción alapuló kísérletek jó közelítést adnak az analitikus költség-



5.10. ábra. A normalizált költségfüggvény és a gamma közelítés a normalizált K értékek függvényében, M=8; 9; 10; p=0,3; k=0,3; $\theta = 12$

függvényekhez. A kísérleti költségfüggvények normalizálása után megállapítottam, hogy a normalizált költségfüggvények hatékonyan közelíthetők gammaeloszlással. A kutatási projekt következő fázisában az elemzés célja egy hatékony módszer kidolgozása a legjobban illeszkedő gammaeloszlás skála- és alakparamétereinek kiszámítására.

6. tézis: Adott alaphalmazhoz tartozó prefix-fa költségbecslésére kidolgoztam egy hatékony szimuláción alapuló közelítő eljárást. Az eljárás bemenő adathalmaz alapparamétereihez (elemszám, listák száma, elemek előfordulásának valószínűségei) regressziós paraméterbecslés mellett meghatároztam a prefix-fa várható költségét. Az elvégzett teszt vizsgálatok alapján a módszer igen pontos (4-5 %-os pontosság) eredményt szolgáltat egy alacsony számítási költség mellett.

Kapcsolódó publikációk: [S7], [S20]

6. fejezet

Összefoglalás

Dolgozatomban olyan témákkal foglalkoztam, amelyek valamilyen módon a bizonytalanság kezeléséhez kapcsolódnak. Új eljárásokat dolgoztam ki, szimulációkat terveztem és implementáltam, valamint ezeken kísérleteket futtattam. Négy fejezetben mutattam be a kutatásaimat.

Az első fejezetben ismertetett kutatás során, a Fábián Csaba vezette kutatócsoportban, valószínűség maximalizálási feladat epigráf-közelítő módszerrel való megoldását valósítottuk meg. Ennek egyik részfeladataként a mesterfeladat kezdőmegoldásának megkeresésére és inicializálására dolgoztam ki eljárásokat, valamint a módszer lényegét képező véletlenített eljárás keretében a pontosság szabályozására dolgoztam ki egy módszert. Az eljárásokat Matlabban implementáltam, és kísérleteket végeztem az eljárások tesztelésére.

A második fejezetben portfólió optimalizálással, illetve kockázatelemzéssel foglalkoztam. Adatgenerálás és szimuláció, illetve valós pénzügyi adatok segítségével megmutattam, hogy kopulákkal jól modellezhetők a portfóliók VaR és CVaR kockázati mutatói, de a különböző kopulák különböző módon befolyásolják ezeket a mutatókat. A szimulációkat R nyelven implementáltam, és különböző paraméterekkel kísérleteket végeztem. Egy másik kutatás keretében valós adatokhoz készítettem szcenáriógenerálási eljárást és algoritmust, amely segítségével több adatot tudtam szolgáltatni kétféle − skálázott és skálázatlan − portfólió optimalizálási eljárásnak, ennek segítségével a portfóliók robusztusabbá váltak. A Gauss-kopulával való adatgenerálást megvalósító algoritmust és a szimulációt végző programokat Matlabban implementáltam.

A harmadik részben a hiányzó adatok pótlására kidolgoztam egy speciális valószínűségekre épülő gépi tanulási eljárást. Bemutattam az eljárás részét képező attribútum kiválasztáshoz használt módszereket. Az új módszert összehasonlítottam ismert módszerekkel, különböző arányú hiányzásokkal, különböző adathalmazokon. A módszerek összehasonlíthatósága érdekében bevezettem két mutatót, az egyiket a pontosság, másikat a végrehajtási idő összehasonlítására. Az algoritmust és a szimulációkat megvalósító programot R nyelven implementáltam.

A negyedik részben a prefix-fa alapú tárolás költségelemzésével foglalkoztam a fa méretére vonatkozóan. Kidolgoztam egy szimulációs eljárást a költség becslésére, majd az eredményeket felhasználva gamma függvénnyel közelítettem a költségfüggvényt. A szimulációt végző programot és az illesztést Matlabban implementáltam.

1. Tézis:
A valószínűség maximalizálási feladatok megoldásához alkalmazott oszlopgenerálás első lépéseként kidolgoztam egy eljárást megoldható mesterfeladat előállítására.

Ezen felül a mesterfeladatot úgy inicializáltam megfelelő oszlopok hozzávételével, hogy a megoldó eljárás megfelelően javító irányba induljon el.

A megoldónak a mesterfeladat részét és a kísérletekhez a futtató keretrendszert implementáltam, valamint kísérleteket végeztem különböző feladatokon.

Kidolgoztam egy eljárást a Genz-kód számítási pontosságának dinamikus szabályozására. Olyan megoldást sikerült kidolgozni, amellyel megoldás közben a gradiens becslés nagy valószínűséggel kellő pontosságú lett, és a függvényérték a szükséges pontosságot nem meghaladóan, de kellően pontos lett. Az eljárás tesztelésére és a megfelelő paraméterek megtalálására kísérleteket végeztem.

A kísérletek eredményeképpen egy gyakorlatban jól használható megoldó eljárást sikerült kidolgozni.

Kapcsolódó publikációk: [S4], [S5], [S8], [S11], [S10], [S12]

2. Tézis:
Szimulációt terveztem többdimenziós eloszlások modellezési lehetőségeire, különböző kopulákat felhasználva. Tárgyaltam a Gauss-kopula esetét, és megmutattam, hogy többféle kapcsolatot tud egyszerre modellezni, továbbá alkalmas aszimmetrikus eloszlások modellezésére is, különböző farokösszefüggésekkel. Ezekre a szakirodalomban nem térnek ki. Bemutattam 2- és 3-dimenziós szimulált kopulák felhasználásával ugyanazon normális peremek mellett a Student-, Clayton-, Gumbel-, Frank-kopula hatásait, majd bemutattam, hogy különböző peremek milyen hatást váltanak ki, ami az aszimmetria és a farokösszefüggéségeket illeti.

Kapcsolódó publikációk: [S6], [S2], [S18]

3. Tézis:
Mivel a kockázati mutatók a ritka eseményekhez kötődnek, ezeket farokösszefüggések jellemzik. Szimulációt terveztem és bemutattam a szimulált adatokon, hogyan befolyásolja az összefüggéseket leíró kopula a kockázati mutatókat. Az is igazolódott, hogy a közel-függetlenség csökkenti a portfólió kockázatát. Valós 3, majd 6 értékpapírból álló egyenletes portfólió esetén, kétirányú kutatást végeztem. Az egyik arra vonatkozott, hogy milyen típusú kopula illeszkedik legjobban az adatokra, a másik pedig a kockázati mutatókra való hatására vonatkozott. A kísérleteket elvégezve arra jutottam, hogy a sokparaméterű kopulák illeszkednek jobban, vagyis a vine- és a Gauss-kopula. Ugyanakkor jó eredményeket hozott a feltételes függetlenségeket tartalmazó (CI) kopula közelítés is, amely sokkal kevesebb paraméterrel rendelkezik, mint a vine-kopula. A portfólióból jól látszanak a Gumbel-, illetve a Clayton-kopula specifikus farokösszefüggései. Rámutattam, hogy kis méretű portfóliók esetén, a kockázat modellezésében komoly szerepe van a kopulatípusnak, vagyis az összefüggés modellezésnek, továbbá a generált mintanagyságnak is.

Kapcsolódó publikációk: [S6], [S2], [S18]

4. Tézis:
Portfólió optimalizáláshoz készítettem szcenáriógenerálási eljárást és algoritmust, amely segítségével több adatot tudtam szolgáltatni a két portfólió optimalizálási eljárásnak. A szimulációs eljárást Matlabban implementáltam. Az eredmények biztatóak, mivel a tanuló mintából tanult és szimulált adatok a Gauss-kopula és lognormális peremeloszlások segítségével elősegítettem az optimális portfólió megválasztását. A skálázott és skálázatlan esetben is a portfólió várhatóértéke nőtt és a szórás csökkent. A különbség az, hogy a skálázott esetben mintha a hozamot szemléltető hisztogramot jobbra toltuk volna, azaz minden egyenletesen jobbra tolódott. A skálázatlan esetben pedig mintha levágtuk volna a "bal farkát". Ezért a skálázatlan megoldás olyan befektetőknek jó, akik félnek az extrém kis valószínűséggel előforduló extrém nagy veszteségtől, míg a skálázott azoknak jó, akik inkább az általános befektetői hozzáállásnak felelnek meg.

Kapcsolódó publikációk: [S1], [S9], [S13], [S14], [S19], [S15], [S16], [S17], [S21]

5. Tézis:
Attribútum kiválasztáson alapuló valószínűségi gépi tanulási adatpótlási módszereket vezettem be. A módszereket diszkrét adatokra fejlesztettem ki, de kategorikus és folytonos adatokra is alkalmazhatóak.

A fejezetben különböző adatadatpótlási módszerek hatékonysági elemzését mutattam be, az egyszeri adatpótlási megközelítésre összpontosítva. A bevezetett módszereimet összehasonlítottam négy, az R programcsomagokban megvalósított módszerrel.

A különböző módszerek különböző adathalmazokon történő összehasonlítása érdekében bevezettem két mutatót. Az egyik a pontosságot, a másik a végrehajtási időt jellemzi.

A valószínűségi gépi tanulás esetében az információtartalom-alapú attribútumcsökkentési módszereim bizonyultak a legjobbnak, de a hagyományos korrelációs együtthatókon alapuló jellemzőválasztás is hatékony volt. Az időköltséget tekintve a valószínűségi gépi tanulási módszereim jellemzően gyorsabbak voltak.

Az elvégzett elemzések és tesztek azt mutatják, hogy a javasolt adatpótlási valószínűségi gépi tanulási technikák egyértelműen előnyösebbek a vizsgált beépített gépi tanulási technikáknál.

Kapcsolódó publikációk: [S3]

6. Tézis:
Adott alaphalmazhoz tartozó prefix-fa költségbecslésére kidolgoztam egy hatékony szimuláción alapuló közelítő eljárást. Az eljárás bemenő adathalmaz alapparamétereihez (elemszám, listák száma, elemek előfordulásának valószínűségei) regressziós paraméterbecslés mellett meghatároztam a prefix-fa várható költségét. Az elvégzett teszt vizsgálatok alapján a módszer igen pontos (4-5 %-os pontosság) eredményt szolgáltat egy alacsony számítási költség mellett.

Kapcsolódó publikációk: [S7], [S20]

Summary

The information we have about the environment is often incomplete. This may be due to incomplete data sets or because of the uncertainty of future events. This is why dealing with uncertainty is an important task, and our goal is to develop models that allow us to simulate possible realizations and estimate the probability of different events. This way, we can obtain information that can help us make the "right" decisions. This motivation led us to different fields like probability theory, data science, machine learning, stochastic optimization, and programming. The topics of the present dissertation are related to uncertainty management. I developed new procedures, designed simulations, and ran experiments in order to deal with the problem of uncertainty, I also implemented codes in order to apply them. The research is presented in four chapters.

In the first chapter, I gave an introduction to the probability maximization problem using an epigraph approximation, a method that was developed and implemented by the research team led by Csaba Fábián. As a sub-task of this work, I developed procedures for finding and initializing the starting solution of the master problem and a method for controlling the accuracy in the context of the randomized procedure which is the essence of the method. I implemented the procedures in Matlab and performed experiments to test the procedures.

In the second chapter, I dealt with portfolio optimization and risk analysis. Using data generation and simulation and real financial data, I showed how copulas can be used in modeling and their effects on the VaR and CVaR of portfolios. I showed how different copulas affect these indicators. I implemented the simulation procedures in R and performed experiments with different parameters. As further research, I created a scenario generation procedure and algorithm for real data, which allowed me to provide more input data to two different − scaled and unscaled − portfolio optimization procedures, with the help of which the portfolios became more robust. I implemented the Gaussian copula data generation algorithm and simulation programs in Matlab.

In the third part, I developed a machine learning procedure based on feature selection to impute the missing data with a value with high probability. I presented methods used for attribute selection as part of the procedure. Then I compared the new method with known methods with different missingness rates on different data sets. To make the methods comparable, I introduced two metrics, one to compare the accuracy and the other to compare execution time. I implemented the imputation algorithms and the comparison methods in R.

In the fourth chapter, I dealt with the cost analysis regarding the tree size of prefix-tree-based storage. I developed a simulation procedure to estimate the cost and then used the results to approximate the cost function with a gamma function. I implemented the simulation procedure and the model fitting in Matlab.

Irodalomjegyzék

[1] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

[2] Ruzanna Ab Razak and Noriszura Ismail. Portfolio risks of bivariate financial returns using copula-var approach: A case study on malaysia and us stock markets. *Global Journal of Pure and Applied Mathematics*, 12(3):1947–1964, 2016.

[3] Andrew Ang and Joseph Chen. Asymmetric correlations of equity portfolios. *Journal of Financial Economics*, 63(3):443–494, March 2002.

[4] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[5] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent Measures of Risk. *Mathematical Finance*, 9(3):203–228, July 1999.

[6] S. Awawdeh, H. Faris, and H. Hiary. Evoimputer: An evolutionary approach for missing data imputation and feature selection in the context of supervised learning. *Knowledge-Based Systems*, 236:107734, 2021.

[7] Tim Bedford and Roger M Cooke. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence*, 32:245–268, 2001.

[8] Tim Bedford and Roger M Cooke. Vines–a new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068, 2002.

[9] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[10] R Brooks and Arthur Geoffrion. Finding everett's lagrange multipliers by linear programming. *Operations Research*, 14(6):1149–1153, 1966.

[11] Arthur Cayley. A theorem on trees. *Quart. J. Math.*, 23:376–378, 1878.

[12] A. Charnes and W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11:18–39, 1963.

[13] A. Charnes, W.W. Cooper, and G.H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958.

[14] Cedric Chauve, Serge Dulucq, and Olivier Guibert. Enumeration of some labelled trees. In *Formal Power Series and Algebraic Combinatorics: 12 th International Conference, FPSAC'00, Moscow, Russia, June 2000, Proceedings*, pages 146–157. Springer, 2000.

[15] Christine T Cheng. On computing the distinguishing numbers of trees and forests. *the electronic journal of combinatorics*, pages R11–R11, 2006.

[16] Umberto Cherubini, Elisa Luciano, and Walter Vecchiato. *Copula Methods in Finance*. John Wiley & Sons, 2004.

[17] C.K. Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[18] Freddy Delbaen. Coherent risk measures on general probability spaces. *Advances in finance and stochastics: essays in honour of Dieter Sondermann*, pages 1–37, 2002.

[19] D. Dentcheva. Optimization models with probabilistic constraints. In G. Calafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 49–97. Springer, 1st edition, 2006.

[20] D. Dentcheva. *Optimisation Models with Probabilistic Constraints. Chapter 4 in [97].* MPS-SIAM series on optimization. SIAM and MPS, Philadelphia, 2009.

[21] D. Dentcheva, B. Lai, and A. Ruszczyński. Dual methods for probabilistic optimization problems. *Mathematical Methods of Operations Research*, 60:331–346, 2004.

[22] D. Dentcheva and G. Martinez. Regularization methods for optimization problems with probabilistic constraints. *Mathematical Programming*, 138:223–251, 2013.

[23] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89:55–77, 2000.

[24] Darinka Dentcheva and Andrzej Ruszczyński. Portfolio optimization with stochastic dominance constraints. *Journal of Banking & Finance*, 30(2):433–451, 2006.

[25] Jeffrey Dissmann, Eike C Brechmann, Claudia Czado, and Dorota Kurowicka. Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69, 2013.

[26] MATLAB Documentation. https://www.mathworks.com/help/stats/gamcdf.html.

[27] Dheeru Dua and Casey Graff. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml, 2017.

[28] J. Elzinga and T.G. Moore. A central cutting plane method for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.

[29] Tlamelo Emmanuel, Thabiso Maupong, Dimane Mpoeleng, Thabo Semong, Banyatsang Mphago, and Oteng Tabona. A survey on missing data in machine learning. *Journal of Big Data*, 8(1):1–37, 2021.

[30] Hugh Everett III. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, 11(3):399–417, 1963.

[31] Csaba I Fábián, Gautam Mitra, and Diana Roman. Processing second-order stochastic dominance models using cutting-plane representations. *Mathematical Programming*, 130:33–57, 2011.

[32] Csaba I Fábián, Gautam Mitra, Diana Roman, and Victor Zverovich. An enhanced model for portfolio choice with ssd criteria: a constructive approach. *Quantitative Finance*, 11(10):1525–1534, 2011.

[33] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13:117–156, 2002.

[34] Antonio JT Garcia and Eduardo R Hruschka. Naive bayes as an imputation tool for classification problems. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, pages 3–pp. IEEE, 2005.

[35] Christian Genest and Jock MacKay. The Joy of Copulas: Bivariate Distributions with Uniform Marginals. *The American Statistician*, 40(4):280–283, 1986. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].

[36] Christian Genest and R. Jock MacKay. Copules archimédiennes et familles de lois bidimensionnelles dont les marges sont données. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 14(2):145–159, 1986. Publisher: [Statistical Society of Canada, Wiley].

[37] A. Genz and F. Bretz. *Computation of multivariate normal and t probabilities.* Number 195 in Lecture Notes in Statistics. Springer, Dordrecht, 2009.

[38] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of computational and graphical statistics*, 1(2):141–149, 1992.

[39] Mohadeseh Hamedanian, Mohammad Nadimi, and Mohammad Naderi. An efficient prefix tree for incremental frequent pattern mining. *Int. J. Inf*, 3(2), 2013.

[40] Frank E Harrell Jr and Maintainer Frank E Harrell Jr. Package 'hmisc'. *CRAN2018*, 2019:235–236, 2019.

[41] R. Henrion. Introduction to chance-constrained programming. Technical report, Tutorial paper for the Stochastic Programming Community Home Page, 2004. [Online] http://www.wias-berlin.de/people/henrion/ccp.ps.

[42] R. Henrion and A. Möller. Optimization of a continuous distillation process under random inflow rate. *Computer & Mathematics with Applications*, 45:247–262, 2003.

[43] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41:263–276, 2008.

[44] R. Henrion and C. Strugarek. *Convexity of Chance Constraints with Dependent Random Variables: the use of Copulae. (Chapter 17 in [?]).* Springer New York, 2011.

[45] Ling Hu. Dependence patterns across financial markets: a mixed copula approach. *Applied Financial Economics*, 16(10):717–729, June 2006.

[46] Harry Joe. Multivariate models and dependence concepts. *Monographs on Statistics and Applied Probability*, (73), 1997.

[47] Kaggle. https://www.kaggle.com.

[48] P. Kall and J. Mayer. *Stochastic Linear Programming: Models, Theory, and Computation.* Springer's International Series in Operations Research and Management Science. Kluwer Academic Publishers, 2005.

[49] Edith Kovács and Tamás Szántai. On the connection between cherry-tree copulas and truncated r-vine copulas. *Kybernetika*, 53(3):437–460, 2017.

[50] Edith Alice Kovács. Speciális többváltozós eloszlások modellezése kopulák segítségével. *Tudományos Közlemények (Általános Vállalkozási Főiskola)*, 13:105–114, 2005.

[51] Edith Alice Kovács. Valószínűségi változók együttes eloszlásának, illetve összefüggésének jellemzése kopulák segítségével. *Tudományos Közlemények (Általános Vállalkozási Főiskola)*, 12:137–151, 2005.

[52] Edith Alice Kovács. Kopulafüggvény egyes sztochasztikus programozási feladatok megoldásában. *Tudományos Közlemények (Általános Vállalkozási Főiskola)*, 14-15:177–186, 2006.

[53] Edith Alice Kovács. Kapcsolatok az információelmélet tükrében: sztochasztikus kapcsolatok elemzése entrópiával, kopulák segítségével. *Tudományos Közlemények (Általános Vállalkozási Főiskola)*, 17:143–151, 2007.

[54] Alexandra Künzi-Bay and János Mayer. Computational aspects of minimizing conditional value-at-risk. *Computational Management Science*, 3(1):3–27, 2006.

[55] Dorota Kurowicka and Roger M Cooke. *Uncertainty analysis with high dimensional dependence modelling.* John Wiley & Sons, 2006.

[56] Kenneth Lange, J Chambers, and W Eddy. *Numerical analysis for statisticians*, volume 1. Springer, 2010.

[57] Serena G Liao, Yan Lin, Dongwan D Kang, Divay Chandra, Jessica Bon, Naftali Kaminski, Frank C Sciurba, and George C Tseng. Missing value imputation in high-dimensional phenomic data: imputable or not, and how? *BMC bioinformatics*, 15(1):1–12, 2014.

[58] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[59] C.H. Liu and et al. The feature selection effect on missing value imputation of medical datasets. *Applied Sciences*, 10.7:2344, 2020.

[60] D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research and Management Science. Springer, 2008.

[61] Jan-Frederik Mai and Matthias Scherer. *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*, volume 06 of *Series in Quantitative Finance*. WORLD SCIENTIFIC, 2 edition, August 2017.

[62] Harry Markowitz. Portfolio Selection. *The Journal of Finance*, 7(1):77–91, 1952. Publisher: [American Finance Association, Wiley].

[63] J. Mayer. *Stochastic Linear Programming Algorithms: A Comparison Based on a Model Management System*. Gordon and Breach Science Publishers, 1998.

[64] J. Mayer. *On the Numerical solution of jointly chance constrained problems. Chapter 12 in [110]*. Springer, 1st edition, 2000.

[65] B.L. Miller and H.M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965.

[66] Panna Miskolczi. Note on simple and logarithmic return. *Applied Studies in Agribusiness and Commerce*, 11(1-2):127–136, June 2017.

[67] Panna Miskolczi. Comparison of Risk Calculation Based on Historical Simulation and the Copula Function. *Public Finance Quarterly*, 63(1):80–95, 2018. Publisher: State Audit Office of Hungary.

[68] D.R. Morgan, J.W. Eheart, and A.J. Valocchi. Aquifer remediation design under uncertainty using a new chance constraint programming technique. *Water Resources Research*, 29:551–561, 1993.

[69] Saba Naz, Muhammad Ahsanuddin, Syed Inayatullah, Tanveer Ahmed Siddiqi, and Muhammad Imtiaz. Copula-Based Bivariate Flood Risk Assessment on Tarbela Dam, Pakistan. *Hydrology*, 6(3):79, August 2019.

[70] Roger B. Nelsen. *An Introduction to Copulas*. Springer Series in Statistics. Springer-Verlag, New York, 2 edition, 2006.

[71] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[72] Georg Ch. Pflug. Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk. In Stanislav P. Uryasev, editor, *Probabilistic Constrained Optimization: Methodology and Applications*, Nonconvex Optimization and Its Applications, pages 272–281. Springer US, Boston, MA, 2000.

[73] András Prékopa. On the probability distribution of the optimum of a random linear program. *SIAM Journal on Control*, 4(1):211–222, 1966.

[74] A. Prékopa. On probabilistic constrained programming. In H.W. Kuhn, editor, *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138. Princeton University Press, Princeton, New Jersey, 1970.

[75] A. Prékopa. Logarithmic concave measures with applications to stochastic programming. *Acta Scientiarium Mathematicarum (Szeged)*, 32:301–316, 1971.

[76] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4:202–221, 1973.

[77] A. Prékopa. On logarithmic concave measures and functions. *Acta Scientiarium Mathematicarum (Szeged)*, 34:335–343, 1973.

[78] A. Prékopa. *Stochastic Programming.* Akadémiai Kiadó, Budapest, 1995.

[79] A. Prékopa. *Probabilistic programming. In [93] (Chapter 5).* Elsevier, Amsterdam, 2003.

[80] A. Prékopa. On the relationship between probabilistic constrained, disjunctive and multiobjective programming. Technical Report 7-2007, Rutgers Center for Operations Research, Rutgers University, Piscataway, NJ, 2007. (RUTCOR Research Report).

[81] A. Prékopa, S. Ganczer, I. Deák, and K. Patyi. The STABIL stochastic programming model and its experimental application to the electrical energy sector of the Hungarian economy. In M.A.H. Dempster, editor, *Stochastic Programming*, pages 369–385. Academic Press, London, 1980.

[82] A. Prékopa and T. Szántai. Flood control reservoir system design using stochastic programming. *Math. Programming Study*, 9:138–151, 1978.

[83] A. Prékopa and T. Szántai. A new multivariate gamma distribution and its fitting to empirical streamflow data. *Water Resources Research*, 14:19–24, 1978.

[84] A. Prékopa and T. Szántai. On optimal regulation of a storage level with application to the water level regulation of a lake. *European Journal of Operations Research*, 3:175–189, 1979.

[85] A. Prékopa, B. Vizvári, and T. Badics. Programming under probabilistic constraint with discrete random variable. In F. Giannesi, T. Rapcsák, and S. Komlósi, editors, *New Trends in Mathematical Programming*, pages 235–255. Kluwer, Dordrecht, 1998.

[86] András Prékopa. Dual method for the solution of a one-stage stochastic programming problem with random rhs obeying a discrete probability distribution. *Zeitschrift für Operations Research*, 34(6):441–461, 1990.

[87] U. Pujianto, A. P. Wibawa, and M. I. Akbar. K-nearest neighbor (k-nn) based missing data imputation. *Conference on Science in Information Technology (ICSITech)*, pages 83–88, 2020.

[88] Rapidminer. https://rapidminer.com/.

[89] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000.

[90] R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

[91] Diana Roman, Ken Darby-Dowman, and Gautam Mitra. Portfolio construction based on stochastic dominance and target return distributions. *Mathematical Programming*, 108:541–569, 2006.

[92] Donald B Rubin. The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The sir algorithm. *Journal of the American Statistical Association*, 82(398):543–546, 1987.

[93] A. Ruszczyński and A. Shapiro. *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, 2003.

[94] Felix Salmon. Recipe for disaster: The formula that killed wall street. *Wired Magazine*, 17(3):17–03, 2009.

[95] Manar D. Samad, Sakib Abrar, and Norou Diawara. Missing value estimation using clustering and deep learning within multiple imputation framework. *Knowledge-Based Systems*, 249:108968, 2022.

[96] Joseph L Schafer and John W Graham. Missing data: our view of the state of the art. *Psychological methods*, 7(2):147, 2002.

[97] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming. Modeling and Theory*, volume 9 of *MPS-SIAM series on optimization*. SIAM and MPS, Philadelphia, 2009.

[98] Abe Sklar. Random variables, joint distribution functions, and copulas. *Kybernetika*, 9(6):449–460, 1973. Publisher: Institute of Information Theory and Automation AS CR.

[99] M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

[100] Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[101] Giorgio Szegö. Measures of risk. *European Journal of Operational Research*, 163(1):5–19, May 2005.

[102] Giorgio Szegő. Kockázat és szabályozás. *Hitelintézeti Szemle*, 3(2):1–31, 2004.

[103] T. Szántai. *Numerical evaluation of probabilities concerning multi-dimensional probability distributions*. PhD thesis, Hungarian Academy of Sciences, 1985.

[104] T. Szántai. A computer code for solution of probabilistic-constrained stochastic programming problems. In Y.M. Ermoliev and R.J.-B. Wets, editors, *Numerical Techniques for Stochastic Optimization*, pages 229–235. Springer-Verlag, Berlin, 1988.

[105] D.M. Topkis and A.F. Veinott. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM Journal on Control*, 5(2):268–279, 1967.

[106] Chih-Fong Tsai and Ya-Han Hu. Empirical comparison of supervised learning techniques for missing value imputation. *Knowledge and Information Systems*, 64(4):1047–1075, 2022.

[107] S. Uryas'ev. Derivatives of probability functions and integrals over sets given by inequalities. *Journal of Computational and Applied Mathematics*, 56(1-2):197–223, 1994.

[108] S. Uryas'ev. Derivatives of probability functions and some applications. *Annals of Operations Research*, 56:287–311, 1995.

[109] Stanislav Uryasev and R. Tyrrell Rockafellar. Conditional Value-at-Risk: Optimization Approach. In Panos M. Pardalos, Donald Hearn, Stanislav Uryasev, and Panos M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications*, volume 54, pages 411–435. Springer US, Boston, MA, 2001. Series Title: Applied Optimization.

[110] S. Uryas'ev (ed). *Probabilistic Constrained Optimization: Methodology and Applications*. Kluwer Academic Publishers, 2000.

[111] W. van Ackooij. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. *Mathematical Methods of Operations Research*, 80:227–253, 2014.

[112] W. van Ackooij. Eventual convexity of chance constrained feasible sets. *Optimization (A Journal of Math. Programming and Operations Research)*, 64:1263–1284, 2015.

[113] W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research*, 77:177–193, 2017.

[114] W. van Ackooij and W. de Oliveira. Level bundle methods for constrained convex optimization with various oracles. *Computation Optimization and Applications*, 57(3):555–597, 2014.

[115] W. van Ackooij and R. Henrion. (sub-)gradient formulae for probability functions of random inequality systems under gaussian distribution. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):63–87, 2017.

[116] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15:509–531, 2014.

[117] C. van de Panne and W. Popp. Minimum-cost cattle feed under probabilistic protein constraints. *Managment Science*, 9:405–430, 1963.

[118] József VARGA. Kopulák alkalmazása a pénzügyi kockázat menedzsmentbenmatematikai alapok. *SZIGMA Matematikai-közgazdasági folyóirat*, 35(3-4):91–106, 2004.

[119] A.F. Veinott. The supporting hyperplane method for unimodal programming. *Operations Research*, 15:147–152, 1967.

[120] Zs J Viharos, Laszlo Monostori, and T Vincze. Training and application of artificial neural networks with incomplete data. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 649–659. Springer, 2002.

[121] B. Vizvári. The integer programming background of a stochastic integer programming algorithm of Dentcheva-Prékopa-Ruszczyński. *Optimization Methods and Software*, 17:543–559, 2002.

[122] Evert Wipplinger. Philippe Jorion: Value at Risk – The New Benchmark for Managing Financial Risk: 3rd Edition, ISBN 0-07-146495-6, McGraw–Hill, 2007, 602 pages, approx. 120 CHF (hardcover). *Financial Markets and Portfolio Management*, 21(3):397–398, September 2007.

[123] Madan Lal Yadav and Basav Roychoudhury. Handling missing values: A study of popular imputation packages in r. *Knowledge-Based Systems*, 160:104–118, 2018.

[124] Chen Ye, Hongzhi Wang, Wenbo Lu, and Jianzhong Li. Effective bayesian-network-based missing value imputation enhanced by crowdsourcing. *Knowledge-Based Systems*, 190:105199, 2020.

[125] Shushang Zhu and Masao Fukushima. Worst-Case Conditional Value-at-Risk with Application to Robust Portfolio Management. *Operations Research*, 57(5):1155–1168, 2009. Publisher: INFORMS.

[126] G. Zoutendijk. *Methods of Feasible Directions: A Study in Linear and Non-Linear Programming*. Elsevier Publishing Co., Amsterdam, 1960.

Saját publikációk

[**1. Könyvfejezet**]

[*angolul*]

[S1] C.I. Fábián, G. Mitra, D. Roman, V. Zverovich, T. Vajnai, E. Csizmás, O. Papp. Portfolio choice models based on Second-order Stochastic Dominance measures: An overview and a computational study. In: M. Bertocchi, G.Consigli, M.A. H. Dempster (szerk.). *Stochastic Optimization Methods in Finance and Energy: New Financial Products and Energy Market Strategies*. New York (NY), Amerikai Egyesült Államok: Springer-Verlag London Ltd, pp. 441-469., 2011. DOI: 10.1007/978-1-4419-9586-5_18 **SCOPUS Független idéző: 6**

[*magyarul*]

[S2] Csizmás Edit, Kovács Edith: Kopulák a kockázatszámításban. In: Johanyák Zsolt Csaba, Kovács Lóránt, Pásztor Attila, Ferenczy Tibor, Weltsch Zoltán, Tóth Ákos, Dobjánné Antal Elvira (szerk.). *Kutatás és innováció 2021 : GAMF Közlemények tanulmánykötete*. Kecskemét, Magyarország. pp. 383-388. 2021.

[**2. Folyóirat cikk**]

[*angolul*]

[S3] Edit Csizmás, Edith Alice Kovács, László Kovács. Data Imputation Methods based on feature selection. **Under Review**, in: *Knowledge-Based Systems*, 2023.

[S4] C.I. Fábián, E. Csizmás, R. Drenyovszki, T. Vajnai, L. Kovács and T. Szántai. A randomized method for handling a difficult function in a convex optimization problem, motivated by probabilistic programming. *Annals of Operations Research*, To appear in S.I.: Stochastic Modeling and Optimization, in memory of András Prékopa (editors: E. Boros, M. Katehakis, A. Ruszczyński). pp. 1-32. 2019. DOI: 10.1007/s10479-019-03143-z. **SCOPUS Q1 IF=4.820 Független idéző: 2**

[S5] Csaba I Fábián, Edit Csizmás, Rajmund Drenyovszki, Wim van Ackooij, Tibor Vajnai, Lóránt Kovács and Tamás Szántai. Probability maximization by

inner approximation. *Acta Polytechnica Hungarica*, 15(1):105–125, 2018. DOI: 10.12700/APH.15.1.2018.1.7 **SCOPUS Q2 IF=1,711 Független idéző: 5**

[*angolul hazai*]

[S6] Edit Csizmás, Edith Kovács. The effect of the dependence structure on risk measures. *GRADUS* 8 : 3 pp. 157-171. 2021. DOI: 10.47833/2021.3.CSC.004

[S7] Csizmás Edit, Kovács László. Cost analysis of the prefix tree data structure. *PRODUCTION SYSTEMS AND INFORMATION ENGINEERING*. 8 pp. 39-49. 2019. DOI: 10.32968/psaie.2019.003

[*magyarul*]

[S8] Edit Csizmás, Rajmund Drenyovszki, Tibor Vajnai, Lóránt Kovács and Csaba Fábián. Valószínűség maximalizálás. *GRADUS*, 5(1):128–133, 2018.

[S9] Vajnai Tibor, Csizmás Edit, Fábián Csaba. Véletlen vektorparaméterek szimulációja. *A GAMF KÖZLEMÉNYEI* 23, pp. 131-140. 2009.

[**3. Konferencia**]

[*angolul*]

[S10] Rajmund Drenyovszki, Edit Csizmás, Csaba Fabian, Lorant Kovacs, Tamas Szantai. A probabilistic formulation of a demand-side management problem, and its solution with a randomized scheme. *VOCAL Optimization Conference : Advanced Algorithms*. Budapest, 2022. május 25-27., Előadás

[S11] E. Csizmás, T. Vajnai, C.I. Fábián. Experiments with randomized method for probability maximization. In: Bojan, Lalić (szerk.) *Proceedings of TEAM 2018 : 9th International Scientific and Expert Conference: 10-12th October 2018*, pp. 237-240. 2018.

[S12] Fábián Csaba, Csizmás Edit, Drenyovszki Rajmund, Vajnai Tibor, Kovács Lóránt, Szántai Tamás. A randomized method for probabilistic problems. *AGTEDU 2018*. Kecskemét : 2018. november 15., Előadás

[S13] Csaba Fabian, Edit Csizmas, Tibor Vajnai. Modeling uncertainty for stochastic optimisation. In: Andrea Ádámné Major, Lóránt Kovács, Zsolt Csaba Johanyák, Róbert Pap-Szigeti (szerk.): *Proceedings of TEAM 2014 : 6th International Scientific and Expert Conference of the International TEAM Society*. Kecskemét, pp. 174-178. 2014.

[S14] Tibor Vajnai, Olga Papp, Edit Csizmás, Csaba I. Fábián. Using scenario generation for decision making under uncertainty. In: Dominika Lehocká, Ján Cárach, Lucia Knapčíková, Sergej Hloch (szerk.): *Proceedings of the 5th International Scientific and Expert Conference of the International TEAM Society : (Technique, Education, Agriculture & Management)*. Presov, Szlovákia : International TEAM Society Press, pp. 150-153. 2013.

[S15] O. Papp, E. Csizmás, T. Vajnai, C.I. Fábián, G. Mitra, D. Roman, V. Zverovich. A comparison of copula-based scenario generation methods. In: L. Suhl, G. Mitra, C. Lucas, A. Koberstein, L. Beckmann (szerk.). *Applied Mathematical Optimization and Modelling : Extended Abstracts of the APMOD 2012 Conference*. Paderborn, Németország : Universität-Gesamthochschule Padernborn, pp. 89-94. 2012.

[S16] Olga Papp, Edit Csizmás, Csaba I. Fábián, Tibor Vajnai. A comparison of scenario generation methods with risk-averse decisions. In: Marija Živić, Tomislav Galeta (szerk.). *TEAM 2012 : Proceedings of the 4th International Scientific and Expert Conference : Mechanical Engineering Faculty*. pp. 223-226. 2012.

[S17] Csaba Fabian, Tibor Vajnai, Gautam Mitra, Diana Roman, Olga Papp, Edit Csizmas. A computational study on the utility of scenario generation methods in portfolio optimisation. *Veszprém Optimization Conference : Advanced Algorithms VOCAL 2010*. Veszprém, 2010., Előadás

[*magyarul*]

[S18] Csizmás Edit, Kovács Edith. A hozamok összefüggésének hatása a kockázati mutatókra. In: Sziklai R. Balázs (szerk.) *XXXIV. Magyar Operációkutatási Konferencia: Absztraktok könyve*. Budapest, Magyarország : Gazdaságmodellezési Társaság. pp. 44. 2021., Előadás

[S19] Csizmás Edit, Vajnai Tibor. Kopula függvények használata véletlenek szimulálására. In: Dobjánné Antal Elvira, Nagy Péter (szerk.). *Matematikát, Fizikát és Informatikát Oktatók 42. Országos Konferenciája MAFIOK 2018 : Konferencia absztraktok* Kecskemét, Magyarország. pp. 26. 2018., Előadás

[S20] Csizmás Edit, Kovács László. Prefix-fa felépítés költségelemzése Monte-Carlo szimulációval. In: Kovács László, Piller Imre (szerk.). *Doktoranduszok Fóruma 2016 : Gépészmérnöki és Informatikai Kar szekciókiadványa*. Miskolci Egyetem, pp. 37-42. 2017.

[S21] Papp Olga, Csizmás Edit, Fábián Csaba, Vajnai Tibor. Kísérletek szcenárió generálási eljárásokkal. In: Ferencz, Árpád; Borsné, Pető Judit; Lipócziné, Csabai Sarolta; Kovács, Lóránt (szerk.) AGTEDU 2011 : a Magyar Tudomány Ünnepe alkalmából rendezett 12. tudományos konferencia pp. 369–374. 2011.