

IGAZSÁGOS TORTAOSZTÁSI ALGORITMUSOK EGYENLŐ ÉS NEMEYENLŐ FELOSZTÁSI ARÁNYOKON

FLEINER TAMÁS, ROMSICS ERZSÉBET

A dolgozatban bemutatjuk az arányos tortaosztzkodás elméletéhez tartozó matematikai apparátus egy részét, majd új eredményeket ismertetünk: az arányos osztzkodást végző Boldogság az Egyenjogúságban Algoritmust, amely nem túl erős feltételek mellett erősebb, szigorúan arányos elosztás ad, és a Részvénytársaság Felosztás Algoritmust, amely sikeresen végzi el a k -személyes nem egyenlő részesedések melletti arányos osztzkodást.

1. Bevezetés

Az igazságos osztzkodás számos történelmi forrás szerint már az ókorban is problémákat okozott az akkori társadalomnak. Több írásos emléket találunk, amelyben a hatalom vagy az egyház rögzíti a különböző javak felosztásának szabályait. Attól függően, hogy a felosztandó jószág milyen tulajdonságokkal rendelkezik, többféle, gyakorlati problémára vonatkozó igazságos osztzkodási eljárás ismert. A pénzzel reprezentálható osztzkodások viszonylag egyszerűek, mert a pénz homogén és bárhogy felosztható. Ha például a válófélben lévő házastársak a közös vagynonon osztzkodnak, és a vagyonnak része valamilyen feloszthatatlan jószág (mondjuk egy autó), akkor az osztzkodás közben arra is kell figyelnünk, hogy ezt az autót ne vágjuk ketté. Természetesen sok esetben egyáltalán nem létezik a feladatnak tökéletes megoldása. Az arányos tortaosztás során egy olyan jószágot osztunk fel, amely heterogén állagú és korlátlanul felosztható. Ez azt jelenti, hogy a játékosok értékelhetnek egy adott szeletet (tortarészt) különbözőképpen, továbbá bárhogy is vágjuk a torta valamely szeletét két tetszőleges szeletre, e két szelet összértéke megegyezik az eredeti szelet értékével. A jelen munka célja két konkrét tortaosztó algoritmus bemutatása.

A Boldogság az Egyenjogúságban Algoritmus arányos osztzkodást végez, azaz egyenlő részesedések esetén arányosan osztja szét a tortát. Itt az ún. együttesen abszolút folytonosság tulajdonság megléte esetén könnyen elérhető, hogy az osztzkodásunk szigorúan arányos legyen, azaz mindenki szigorúan többet kapjon a részesedésénél.

A Részvénytársaság Felosztás Algoritmus sikeresen végzi el k személyre a nem egyenlő részesedések melletti arányos osztozkodást. Az algoritmus alapjaként a Részvénytársaság Szétválasztás szubrutin szolgál, amely kettévág egy részvénytársaságot (ami itt a felosztandó jószágot jelenti), majd a játékosok a számukra jövedelmezőbb oldalra próbálnak helyezkedni a részvényeik cserélgetésével. A hatékonyságot alátámasztandó, igazolunk a vágásszámra egy felső becslését. Ez általában nem éles, így többnyire az algoritmus még ennél is jobban teljesít.

A Boldogság az Egyenjogúságban Algoritmus az általunk Oszd Meg és Uralkodj Algoritmusnak nevezett Even–Paz-eljárás alapján készült [2, 3, 4]. Ez utóbbi módszer lényege, hogy egy vágással úgy vágja szét a tortát, és szeparálja ezáltal kisebb csoportokra a játékosokat, hogy mindenki pontosan egy szeleten osztozkodjon tovább. Egy adott szelet szétदारabolását pedig addig folytatjuk, amíg minden szeletnek legfeljebb 2 tulajdonosa van. E két játékos pedig már egyszerűen elosztja az adott szeletet: az egyik vág, a másik választ. Ebben az eljárásban szerepel egy passzív játékos is, aki a többiek egyidejű cselekvése után, azokat mérlegelve hozza meg saját döntését a torta szeleteléséről. Ezzel lehetőséget kap, hogy akár jóval értékesebb szeletet kapjon a tortából, mint a részesedése. A Boldogság az Egyenjogúságban Algoritmus elkerüli ezt a játékosok közti aszimetriát, mégis legtöbbször eléri, hogy mindenki szigorúan többet kapjon a jussánál. Másik eredményünk, a Részvénytársaság Felosztás Algoritmus általánosítja az ismert kétszemélyes Közel Felező Algoritmust. Ezek az algoritmusok arra szolgálnak, hogy olyan osztozkodásokat hajtsanak végre, melyekben a játékosok nem egyenlő arányban részesednek a tortából [3]. A Közel Felező Algoritmusban a vágó játékos közel felezi a tortát, míg a másik játékos szeletet választ, és ezzel lecsökken a részesedése a maradékból. A megmaradt szeleten addig alkalmazzák újra a közel felezés módszerét, amíg egyenlő részesedésekhez jutnak, és egy vágással befejezhetik az osztozkodást. Robertson és Webb könyve megemlíti egy lehetséges általánosítást [3]. Ha feltesszük, hogy $k - 1$ személy már arányosan elosztotta a tortát, akkor a k -személyes felosztást meghatározhatjuk úgy, hogy a k . személy (akinek még nincs tortája) a kétszemélyes Közel Felező Algoritmust használva külön-külön osztozkodik a többi játékoskal a saját szeleteiken. Ennek az algoritmusnak egy hátránya, hogy igen sok vágást igényel, így a sok újraosztás miatt a játékosok tipikusan igen sok különálló darabot kapnak. Akit a probléma részletesebben érdekel, Tasnádi Attila Igazságos elosztások című könyvének ezt a fejezetét olvassa el [4].

Jelen munkánk az alábbiak szerint épül fel. A második fejezetben a tortaosztással kapcsolatos, számunkra szükséges alapvető fogalmakat tisztázzuk. A harmadik a Boldogság az Egyenjogúságban Algoritmust és annak tulajdonságait írja le, majd a negyedik a Részvénytársaság Felosztás Algoritmust mutatja be.

2. Az arányos tortaosztás leírása

Legyen az $X = [0, 1]$ halmaz egy heterogén, felosztható jószág, melyet ezentúl tortának hívunk (a leírtak az $\mathbb{R}^n$ minden mérhető, korlátos halmazára általánosíthatóak). Legyen $\{P_1, P_2, \dots, P_k\}$ a játékosok csoportja, akik a tortát egymás közt felosztják. Legyen az indexhalmazunk $I := \{1, 2, \dots, k\}$. Minden P_i játékos ezentúl az $X_i \subset X$ halmazban részesül, valamint egy μ_i mértékkel rendelkezik. Ez a μ_i egy végesen szubadditív mérték:

$$\forall k < \infty : \quad \mu\left(\bigcup_{i=1}^k X_i\right) \leq \sum_{i=1}^k \mu(X_i), \quad (1)$$

abszolút folytonos a Lebesgue-mértékre:

$$\forall X_j \subset X : \quad \lambda(X_i) = 0 \Rightarrow \mu_i(X_j) = 0, \quad (2)$$

és normált, azaz

$$\mu_i(X) = 1. \quad (3)$$

Jegyezzük meg, hogy μ és ν mérték együttesen abszolút folytonos, ha egymásra abszolút folytonosak, azaz

$$\forall X_j \subset X : \quad \mu(X_j) = 0 \Leftrightarrow \nu(X_j) = 0. \quad (4)$$

Az X egy *felosztása* az $\{X_i\}_{i \in I}$ rendszer, ha

$$\bigcup_{i \in I} X_i = X, \text{ és } \forall i \neq j : X_i \cap X_j = \emptyset.$$

A torta, mint egy szétosztandó jószág, részhalmazainak lehet különböző tulajdonsága, vagyis két azonos Lebesgue-mértékű részhalmaz képviselhet más értéket az egyes játékosoknak, és mindenki annál boldogabb, minél nagyobb szeletet kap. Minden játékos egy nemnegatív valós számot rendel a tortaszeletekhez, vagyis értékeli őket a saját mértéke szerint. Fontos megjegyeznünk, hogy a szeletek értékének megadásakor az egyes játékosok döntései egymástól függetlenek. Ez azt jelenti, hogy nem ismerik a többiek mértékét, nem tudják, hogy milyen értékűnek tartják az egyes szeleteket.

A P_i játékos tortából való *részesedését* reprezentálja $d_i \in \mathbb{N}^+$, azaz a P_i játékos $\frac{d_i}{\sum_{j \in I} d_j} \in]0, 1[$ részét szeretné a tortának. Feltehető, hogy a d_i részesedések relatív prímek. Egy felosztást *arányosnak* nevezünk, ha minden játékos úgy gondolja, hogy legalább a részesedésével arányos halmazzal kapott a tortából, azaz

$$\mu_i(X_i) \geq \frac{d_i}{\sum_{j \in I} d_j} \quad \forall i \in I. \quad (5)$$

Arányos osztozkodási feladatnak nevezzük a problémát, ha:

$$d_i = 1 \quad \forall i \in I, \quad (6)$$

vagyis mindenki egyenlő részben részesül a tortából. Ha minden P_i játékos szigorúan nagyobb részét kapja a tortának, mint az arányos részesedése, akkor azt mondhatjuk, hogy az osztozkodás *szigorúan arányos*, azaz ha:

$$\mu_i(X_i) > \frac{d_i}{\sum_{j \in I} d_j} \quad \forall i \in I. \quad (7)$$

Az osztozkodáshoz maximálisan szükséges vágásokról hozott döntések számát a tortaosztó algoritmus *vágásszámának* nevezzük. Ez azt jelenti, hogy egyéb paraméterek kiszámolása és a vágások utáni szeletválasztások nem számítanak vágásnak, hiszen azok nem a vágások helyéről döntenek. Jelölje $D(k, n)$ a vizsgált k -személyes tortaosztó algoritmus maximális vágásszámát, ahol $n = \sum_{i \in I} d_i$. Egyenlő részesedések (azaz arányos osztozkodás) esetén ezt a vágásszámot jelölje $D(k)$.

3. Boldogság az Egyenjogúságban Algoritmus

A *Boldogság az Egyenjogúságban Algoritmus* arányos osztozkodást eredményez, amely bizonyos feltételek mellett szigorúan arányos. A módszer a már ismertetett Oszd Meg és Uralkodj Algoritmus módosítása, mely elhagyja a kitüntetett passzív játékos szerepét. Ezenkívül megköveteli, hogy minden játékos mértéke együttesen abszolút folytonos legyen a Lebesgue-mértékre nézve. Ez abban egészíti ki az eddigi mértékre tett megköveteléseinket, hogy ha egy halmaznak Lebesgue-mérték szerinti pozitív mértéke van, akkor minden játékos úgy gondolja, hogy a saját mértéke szerint is pozitív mértékű. A fejezet végén bemutatjuk, hogy ezek a módosítások nagyságrendben nem változtatnak az Oszd Meg és Uralkodj Algoritmus vágásszámán, vagyis a Boldogság az Egyenjogúságban Algoritmus nagyságrendben optimális marad, és egy nem túl szigorú feltétel teljesülése esetén szigorúan arányosan oszt.

Boldogság az Egyenjogúságban Algoritmus

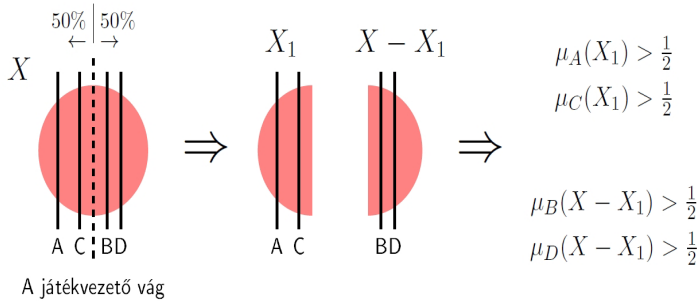
Tegyük fel, hogy minden játékos mértéke együttesen abszolút folytonos a Lebesgue-mértékre nézve.

Ha $k = 1$, akkor az egyetlen játékos megkapja a teljes tortát.

Ha $k \geq 2$, akkor minden játékos jelölje be a tortán egy függőleges, a többiekéhez képest párhuzamos vonallal, hogy hol vágná el a tortát $\lfloor \frac{k}{2} \rfloor : \lceil \frac{k}{2} \rceil$ arányban. A játékvezető balról a $\lfloor \frac{k}{2} \rfloor$ -dik és a $(\lfloor \frac{k}{2} \rfloor + 1)$ -dik jelölés között elvágja a tortát úgy, hogy e két jelölés közti szeletet felezze Lebesgue-mérték szerint.

A vágás bal oldalán lévő szeletet balról számolva az első $\lfloor \frac{k}{2} \rfloor$ játékosnak, a jobb oldalán lévő szeletet pedig a maradék $\lceil \frac{k}{2} \rceil$ játékosnak adja további osztozkodásra. Ha valamelyik vágásra több játékos jelölése esik, akkor a játékvezető tetszés szerint ossza őket az egyes oldalakra úgy, hogy a továbbosztozkodó játékosok száma megfelelő legyen. A bal oldali szeleten a megfelelő játékosok folytatják az osztozkodást az $\ell = \lfloor \frac{k}{2} \rfloor$ -személyes algoritmussal, a jobb oldali szeleten pedig egy $m = \lceil \frac{k}{2} \rceil$ -személyes változatot játszanak majd le.

3.1. Példa. Alice, Bob, Charlie és Dave osztozkodnak a tortán. Minden játékos megjelöli a tortát ott, ahol elfelezné. Tegyük fel, hogy a következő sorrendben látjuk a játékosok jelöléseit balról jobbra: Alice, Charlie, Bob és Dave. Ekkor a játékvezető Charlie és Bob jelölése között elvágja a tortát vigyázva arra, hogy e két jelölése közötti részt Lebesgue-mérték szerint felezzé. A bal oldali szeletet Alice-nek és Charlie-nak adja, a jobb oldalit pedig Bobnak és Dave-nek. Ezzel a négy személyes problémát kettő kétszemélyesre szűkítettük. Az 1. ábra illusztrálja a példában elvégzett osztozkodást.



1. ábra. A BEA futása négy személy esetén.

A Boldogság az Egyenjogúságban Algoritmusról könnyen belátható, hogy arányos osztozkodást valósít meg. Az algoritmus párhuzamos vágásokkal szeleteli fel a tortát, ezért a játékosok jelölései balról jobbra megszámozhatóak az I indexhalmazzal. Ekkor egy $\pi : I \rightarrow I$ permutáció határozza meg, hogy a játékosok milyen sorrendben jelölték meg a tortát:

$$\pi(i) = j \quad \Leftrightarrow \quad \text{az } i. \text{ jelölést } P_j \text{ tette le.}$$

3.1. TÉTEL. *Ha egy játékos más helyen jelöli meg a tortát bármelyik jelölés-kor, mint a többi $k-1$ játékos, akkor a Boldogság az Egyenjogúságban Algoritmus garantálja számára a szigorúan nagyobb, mint $\frac{1}{k}$ részt. Ha minden játékosra igaz, hogy legalább egyszer nem az ő jelölése mentén vágta el egy szeletet, akkor a Boldogság az Egyenjogúságban Algoritmus szigorúan arányos osztozkodást produkál.*

Bizonyítás. Legyen $\ell := \lfloor \frac{k}{2} \rfloor$. Tegyük fel, hogy a játékvezető az első vágást az ℓ . vágás mentén ejtette. Legyen a vágás bal oldalán lévő szelet X_B . Ekkor pontosan tudjuk, hogy $P_{\pi(\ell)}$, vagyis az ℓ . jelölést tevő játékos mekkorának tartja az X_B szeletet:

$$\mu_{\pi(\ell)}(X_B) = \frac{\ell}{k}. \quad (8)$$

Megjegyezhetjük, hogy az ℓ . jelölés menti vágás azt jelenti, hogy balról az ℓ . és $(\ell+1)$. jelölések egybeesnek, vagyis $P_{\pi(\ell+1)}$ játékos is ugyanazt gondolja a bal oldali szeletről, mint $P_{\pi(\ell)}$. Sőt, előfordulhat, hogy több mint két játékos jelölése esett egybe a vágással, ekkor a játékvezető megfelelő számú játékost sorsol a bal és a jobb oldalra.

Az X_B szeleten a vágás után a $P_{\pi(1)}, P_{\pi(2)}, \dots, P_{\pi(\ell)}$ játékosok osztozkodnak tovább. Ekkor a $P_{\pi(\ell)}$ játékos részesedése az X_B szeletből $\frac{1}{\ell}$, az X_B szeletet pedig $\frac{\ell}{k}$ nagyságúnak tartja. Ha megkapja az X_B szeletből a részesedését, akkor számára a saját szelete legalább $\frac{1}{\ell} \cdot \frac{\ell}{k} = \frac{1}{k}$ értékű, vagyis megkapta legalább az arányos részét a teljes tortából. Ha a vágásra ezek közül a játékosok közül többek jelölése esett, azokra ugyanez levezethető.

De mi történik a többiekkel? Ha $P_{\pi(i)}$ egy bal oldalra osztott játékos ($i \in \{1, 2, \dots, \ell-1\}$), és az adott körben tett jelölése nem esik az akkori vágásra, akkor a jelölése és a vágás között van egy pozitív Lebesgue-mértékű halmaz. Mivel a saját mértéke együttesen abszolút folytonos a Lebesgue-mértékre nézve, így ő is pozitív mértékűnek látja ezt a köztes halmazt, vagyis $\exists \varepsilon > 0$, hogy:

$$\mu_{\pi(i)}(X_B) = \frac{\ell}{k} + \varepsilon. \quad (9)$$

Hasonló érvényes a $P_{\pi(\ell+1)}, P_{\pi(\ell+2)}, \dots, P_{\pi(k)}$ játékosokra és a jobb oldali szeleten történetekre. Tehát mindenki, akinek a jelölése nem esik első körben a vágásra, szigorúan többet kap a tortából, mint az arányos része.

Az előzőek általánosíthatók, ha minden körben az új szeletek értékét normáljuk. Ezzel az általánosítással megállapíthatjuk, hogy ha egy játékos mindenki mástól különbözőképpen jelöli meg a tortát bármelyik körben, akkor biztosan nem a jelölése mentén vágnak, vagyis az (9) egyenletben leírtak alapján szigorúan többet fog kapni az arányos részénél.

Ezek alapján pedig ha mindenkire igaz, hogy legalább egyszer más helyen jelölte meg a tortát, mint többiek, akkor az osztozkodás szigorúan arányos. $\square$

A Boldogság az Egyenjogúságban Algoritmus leírása alapján a vágásszámra rekurziós formulát definiálhatunk. Ha k személy osztozkodik a tortán, akkor a legelső körben mindenki megjelöli, hogy hol vágná el. Mivel ezek mind vágásokkal kapcsolatos döntések, amik befolyásolhatják a vágás fizikai helyét, beleszámítanak a vágásszámba. A játékevezető lépései, a fizikai vágás és a játékosok szétosztása nem számítanak bele. A vágás után pedig két, kevesebb résztvevős tortaosztást kell majd végigjátszaniuk a megfelelő játékosoknak, tehát a rekurziós formula a következőképpen írható fel az alsó és felső egészrész függvények segítségével:

$$D(k) = k + D\left(\left\lfloor \frac{k}{2} \right\rfloor\right) + D\left(\left\lceil \frac{k}{2} \right\rceil\right). \quad (10)$$

3.2. TÉTEL. *A k -személyes Boldogság az Egyenjogúságban Algoritmus vágásszáma rekurzív formulájának zárt alakja*

$$D(k) = k \cdot \lceil \log_2 k \rceil - 2^{\lceil \log_2 k \rceil} + k. \quad (11)$$

Bizonyítás. A Boldogság az Egyenjogúságban Algoritmus az Oszd meg és Uralkodj Algoritmus egy egyszerű módosításával született, és előbbi vágásszámát egyszerűen megkapjuk utóbbi vágásszámából (a levezetés Robertson és Webb könyvében megtalálható [2, 3]). A passzív játékos elkerülése a Boldogság az Egyenjogúságban Algoritmusban azt eredményezi, hogy minden egyes körben (jelöléskor) plusz egy vágást ejtünk. Az Oszd Meg és Uralkodj Algoritmus vágásszáma tehát

$$D(k) = k \cdot \lceil \log_2 k \rceil - 2^{\lceil \log_2 k \rceil} + 1.$$

A k résztvevő esetén összesen k darab szeletre vágjuk a tortát, vagyis $k - 1$ darab fizikai vágást ejtünk. Ezt pedig pontosan $k - 1$ kör alatt érhetjük el, ami összesen $k - 1$ plusz vágást jelent az Oszd meg és Uralkodj Algoritmus vágásszámahoz képest. $\square$

Even és Paz [2] Oszd Meg és Uralkodj Algoritmus $\mathcal{O}(k \cdot \log k)$ vágással végzi el az arányos tortaosztást. Edmonds és Pruhs [1] belátta, hogy ez a vágásszám optimális a determinisztikus algoritmusok körében.

3.1. KÖVETKEZMÉNY. *A determinisztikus Boldogság az Egyenjogúságban Algoritmus vágásszáma $\mathcal{O}(k \cdot \log k)$ nagyságrendű, vagyis optimális.*

4. Részvénytársaság Felosztás Algoritmus

Ebben a fejezetben áttérünk a nem egyenlő részesedések melletti arányos osztozkodási feladatra, azaz amikor a játékosok részesedése a tortából egymáshoz képest különböző. Az itt definiált algoritmus természete miatt célszerű újrafogalmazni az eddigi tortaosztással kapcsolatos fogalmakat. Ezentúl a C részvénytársaság reprezentálja a tortát, P_i játékosok pedig a vállalat részvényesei lesznek

d_i részesedéssel ($i \in I = \{1, 2, \dots, k\}$). A részesedések összege $n = \sum_{i \in I} d_i$, jelölje ezentúl a C vállalat részvényeinek számát, valamint a teljes C részvénytársaság érjen minden játékosnak 1-et. Így a részvényesek $\frac{1}{n}$ értékűnek tartják a vállalat minden részvényét.

Az eljárás során a C részvénytársaságot $C^1, C^2, \dots, C^m$ részvénytársaságokra vágjuk (rendre $n^1, n^2, \dots, n^m$ részvénnel), ahol $\sum_{j=1}^m n^j = n$. Egy P_i részvényes akkor tartja *jövedelmezőnek* a C^j vállalatot, ha egy részvény értéke C^j -ben legalább annyit ér számára, mint egy részvény C -ben, azaz ha

$$\frac{\alpha}{n^j} \geq \frac{1}{n}, \text{ ahol } \alpha = \mu_i(C^j) \in]0, 1[. \quad (12)$$

Ezek alapján P_i *jövedelmezőbbnek* tartja a C^j vállalatot bármelyik másikhoz képest, ha számára többet ér egy C^j -beli részvény, mint egy részvény a másik vállalatból. Célunk, hogy minden C^j vállalat pontosan egy részvényes tulajdonában legyen, és minden P_i részvényes pontosan d_i részvénnel rendelkezzen az osztzkodás végére, vagyis

$$\forall i \in I : d_i = \sum_{J_i \subset J} n^j, \text{ ahol } \bigcup_{i \in I} J_i = J \quad (13)$$

Bemutatjuk az ismert Közel Felező Algoritmus egy k -személyes általánosítását, a Részvénytársaság Felosztás Algoritmust, amellyel a fent említett $\{C^j\}_{j \in J}$ felosztás elkészíthető. Az algoritmus alapjaként a következőkben megfogalmazott Részvénytársaság Szétválasztás szubrutin szolgál, amely kettévág egy részvénytársaságot, majd a játékosok a számukra jövedelmezőbb oldalra próbálnak helyezkedni a részvényeik cserélgetésével. A szubrutint újra alkalmazzuk azokra a kisebb vállalatokra, melyeknek még legalább két tulajdonosa van.

A Részvénytársaság Felosztás Algoritmus bemenete maga a $\{C\}$ vállalat és egy $(d_1, d_2, \dots, d_k)^T$ részesedési lista, amely az osztzkodó $P_1, P_2, \dots, P_k$ játékosok részesedését reprezentálja a C vállalatban. Az algoritmus kimenete egy $\mathcal{C} = \{C^1, C^2, \dots, C^m\}$ felosztás és egy $\mathcal{D} \in \mathbb{N}^{k \times m}$ mátrix, melynek d_i^j (i . sor, j . oszlopbeli) eleme megmutatja, hogy a P_i játékos mekkora arányban részesül a C^j vállalatból, ahol a C^j vállalati részvények száma $n^j = \sum_{i \in I} d_i^j$.

$$\mathcal{D} = \begin{bmatrix} d_1^1 & d_1^2 & \dots & d_1^m \\ d_2^1 & d_2^2 & \dots & d_2^m \\ \vdots & \vdots & \ddots & \vdots \\ d_k^1 & d_k^2 & \dots & d_k^m \end{bmatrix} \in \mathbb{N}^{k \times m}. \quad (14)$$

4.1. KÖVETKEZMÉNY. Mivel az Részvénytársaság Felosztás Algoritmus végén minden C^j vállalatnak egy tulajdonosa van, így tudjuk, hogy $\mathcal{D}$ minden oszlopában pontosan egy elem nem 0. Ez nem mondható el a sorokról, hiszen egy játékosnak lehet több vállalata is: az i . sort összegezve d_i -t kapunk.

A Részvénytársaság Szétválasztás szubrutin bemenete egy $\{C^j\}$ tortaszelet és egy $(d_1^j, d_2^j, \dots, d_k^j)^T$ lista, amely az osztozkodó $P_1, P_2, \dots, P_k$ játékosok részesedését reprezentálja a C^j vállalatban (a mátrix j . oszlopa). Az szubrutin kimenete a C^j vállalat egy felosztása: egy C^* és egy C^{**} vállalat, rendre $(d_1^*, d_2^*, \dots, d_k^*)$ és $(d_1^{**}, d_2^{**}, \dots, d_k^{**})$ részesedési listákkal, melyekre teljesül, hogy minden $i \in I$: $d_i^* + d_i^{**} = d_i^j$. Ha a vállalatnak (amit bemenetként adtunk) csak egy tulajdonosa van, akkor a szubrutin visszaadja a bemenetet (vagyis C^{**} nem feltétlenül létezik). A szubrutin kimenetét a Részvénytársaság Felosztás Algoritmus beilleszti a saját C halmazába és D mátrixába, és ekkor a csillagok helyére a megfelelő indexek kerülnek.

Részvénytársaság Felosztás (RTF) Algoritmus

BEMENET: $C = \{C\}$ és $D = (d_1, d_2, \dots, d_k)^T$.

Meghívjuk a Részvénytársaság Szétválasztás szubrutint: $RTSzV(C, D)$.

A kimenetet feldolgozzuk: $C = \{C^1, C^2\}$, és a $(d_1^1, d_2^1, \dots, d_k^1)^T$ és $(d_1^2, d_2^2, \dots, d_k^2)^T$ részesedési listák lesznek a D mátrix oszlopai. Az $RTSzV(C^1, (d_1^1, d_2^1, \dots, d_k^1)^T)$ és $RTSzV(C^2, (d_1^2, d_2^2, \dots, d_k^2)^T)$ szubrutinok közül meghívjuk azokat, melyeknek a bemenetként megadott részesedési listájában legalább két 0-tól különböző szám van.

Ha egy részesedési listában pontosan egy 0-tól különböző szám van, akkor az ahhoz tartozó vállalat egy ember tulajdonában van, és nem osztozkodnak tovább rajta.

REKURZÍÓ: A kimeneteket feldolgozzuk: a C halmazban minden C^j helyére a C^* és a C^{**} kimenetként kapott vállalatok kerülnek (megfelelő indexszel), a D mátrixból töröljük a $(d_1^j, d_2^j, \dots, d_k^j)^T$ listát, helyette pedig beillesztjük a $(d_1^*, d_2^*, \dots, d_k^*)^T$ és $(d_1^{**}, d_2^{**}, \dots, d_k^{**})^T$ listákat (megfelelő indexszel). A feldolgozás után meghívjuk azokat az $RTSzV(C^j, (d_1^j, d_2^j, \dots, d_k^j)^T)$ szubrutinokat (ahol $C^j \in C$), melyeknek a bemenetként megadott részesedési listájában legalább két 0-tól különböző szám van. Ha egy részesedési listában pontosan egy 0-tól különböző szám van, az ahhoz tartozó vállalat nem osztozkodnak tovább.

KIMENET: $\mathcal{C} = \{C^1, C^2, \dots, C^m\}$ felosztás és egy $\mathcal{D} \in \mathbb{N}^{k \times m}$.

A következőkben bemutatjuk a Részvénytársaság Szétválasztás szubrutint, ami az RTF algoritmus alapja.

Részvénytársaság Szétválasztás (RTSzV) szubrutin

BEMENET: C vállalat és $(d_1, d_2, \dots, d_k)^T$ részesedési lista.

Ha csak egy tulajdonosa van a C vállalatnak, akkor készen vagyunk. Egyébként tegyük fel, hogy P_ℓ a legnagyobb részesedéssel rendelkező játékos, vagy azok egyike. Ekkor $n = d_1 + d_2 + \dots + d_k$.

1.eset: Ha $d_\ell \geq \lceil \frac{n}{2} \rceil$, akkor a P_ℓ játékos *közel felezi* a C vállalatot C^1 és C^2 vállalatokra, rendre $\lceil \frac{n}{2} \rceil$ és $\lfloor \frac{n}{2} \rfloor$ számú részvénnel, azaz $\frac{\mu_\ell(C^1)}{\lceil \frac{n}{2} \rceil} = \frac{\mu_\ell(C^2)}{\lfloor \frac{n}{2} \rfloor}$ teljesül (minden részvény egyenlő értékű P_ℓ -nek). A C^1 vállalatból minden P_i ($i \in I$) játékos $d_i \cdot \lceil \frac{n}{2} \rceil \cdot \frac{1}{n}$ részvényt kap, a C^2 vállalatból pedig $d_i \cdot \lfloor \frac{n}{2} \rfloor \cdot \frac{1}{n}$ részvényt. A C^1 és C^2 vállalat közül a P_ℓ játékoson kívül minden P_i játékos kiválasztja, hogy számára melyik *jövedelmezőbb*. (Ha minden részvény ugyanannyit ér P_i -nek, akkor tetszőlegesen választhat vállalatot.) Ezek után a játékosok elkezdene a P_ℓ játékosal *részvényeket cserélni*. Ha egy játékos jövedelmezőnek tart egy vállalatot, akkor a másik vállalatbeli részvényeit (akár törtszámú részvényt is) örömmel cserél jövedelmezőbbekre.

A részvéncsere addig tart, amíg minden játékosnak egész számú részvénye lesz mindkét vállalatban, és P_ℓ játékoson kívül mindenkinek már csak vagy C^1 -ben, vagy C^2 -ben vannak részvényei. (Ha a P_ℓ játékosnak túl nagy részesedése van a C vállalatból, akkor előfordulhat az az eset, hogy nem lehet csak egy oldalon részvénye.)

Ebben az esetben vagy redukáltuk a k -személyes problémát egy $\lceil \frac{n}{2} \rceil$ és egy $\lfloor \frac{n}{2} \rfloor$ részvénytartalmú, kevesebb résztvevős problémára, vagy csökkentettük P_ℓ részesedését.

2.eset: Ha $d_\ell < \lceil \frac{n}{2} \rceil$, akkor P_ℓ *pontosan felezi* a C vállalatot C^1 és C^2 vállalatokra, vagyis $\mu_\ell(C^1) = \mu_\ell(C^2) = \frac{1}{2}$ teljesül, a részvények pedig *inflálódnak*: mindkét vállalat n részvényt kap, és minden P_i játékos d_i részvényt kap mindkét vállalatból. Ekkor, ahogy az 1. esetben, minden játékos kiválasztja a jövedelmezőbb vállalatot, majd részvényeket cserélnek mindaddig, amíg valamely játékosnak lehetősége van, hogy egy részvényt jövedelmezőbb részvényre cseréljen.

Ebben az esetben redukáltuk a k -személyes problémát két, legfeljebb $(k - 1)$ -személyes problémára, hiszen minden esetben van egy P_i játékos (legrosszabb esetben P_ℓ), akinek elég részvénye van ahhoz, hogy a legkisebb részesedéssű játékosal addig cseréljen, amíg annak csak jövedelmező részvénye marad. Ezenkívül a P_i játékos is tud addig cserélni, amíg csak jövedelmezőbb részvénye van, különben nem ebben az esetben lennének.

Megjegyezzük, hogy $k = 2$ személyre az RTF algoritmus 2. esete nem fordulhat elő, vagyis a kétszemélyes algoritmus ekvivalens a Közel Felező Algoritmussal.

4.1. ÁLLÍTÁS. Az RTSzV szubrutin 1. esetében a vágó P_ℓ játékosnak mindig van elég részvénye, hogy addig cseréljen, amíg minden játékosnak csak az egyik vállalatban van egész számú részvénye.

Bizonyítás. Mindkét új vállalatához egész számú részvény tartozik, és ebből következik, hogy kaphatunk kimenetként egészelemű részesedési listákat.

A legrosszabb eset, ha a P_ℓ vágó játékoson kívül mindenki ugyanazt a vállalatot tartja jövedelmezőbbnek, és ennek a vállalatnak $\lfloor \frac{n}{2} \rfloor$ részvénye van, ugyanis ekkor van szüksége a P_ℓ játékosnak a legtöbb részvényre a másik vállalatból. Ezek alapján teljesülnie kell a következő egyenlőtlenségnek, hogy mindig elég részvénye legyen a P_ℓ játékosnak:

$$\begin{aligned} d_\ell \cdot \frac{\lfloor \frac{n}{2} \rfloor}{n} &\geq (n - d_\ell) \cdot \frac{\lceil \frac{n}{2} \rceil}{n} \\ d_\ell \cdot \left\lfloor \frac{n}{2} \right\rfloor &\geq (n - d_\ell) \cdot \left\lceil \frac{n}{2} \right\rceil = n \left\lceil \frac{n}{2} \right\rceil - d_\ell \cdot \left\lceil \frac{n}{2} \right\rceil \\ d_\ell \cdot \left\lfloor \frac{n}{2} \right\rfloor + d_\ell \cdot \left\lceil \frac{n}{2} \right\rceil &\geq n \cdot \left\lceil \frac{n}{2} \right\rceil \\ d_\ell \cdot n &\geq n \cdot \left\lceil \frac{n}{2} \right\rceil \\ d_\ell &\geq \left\lceil \frac{n}{2} \right\rceil \end{aligned}$$

Vagyis megkaptuk azt a feltételt, ami alapján a szubrutin 1. esetébe kerültünk. $\square$

4.2. ÁLLÍTÁS. Az RTSzV szubrutin 2. esetében mindegyik vállalatnak legalább eggyel kevesebb tulajdonosa lesz, mint az eredetiben volt.

Bizonyítás. Legyen a legkevesebb részesedéssel rendelkező játékos P_m . Mivel nem egyenlő részesedésekkel dolgozunk, így lesz legalább egy játékos (aki akár a P_ℓ vágó játékos is lehet), akinek a részesedése több, mint a P_m játékosé. A nagyobb részesedéssel rendelkező tud annyi részvényt adni P_m -nek, hogy csak egy vállalatnál legyen részvénye.

A legrosszabb eset, amikor a P_ℓ játékoson kívül minden játékos ugyanazt a vállalatot tartja jövedelmezőbbnek. Ellenkező esetben a P_ℓ játékoson kívüli P_i játékosok ($i \neq \ell$) elkezdik a részvéncserét, és a P_ℓ játékos részvényeire csak részben van szükség ahhoz, hogy teljesüljön az állítás. Megmutatjuk, hogy a P_ℓ játékos a legrosszabb esetben el tudja cserélni a mindenki által jövedelmezőbbnek tartott vállalati részvényeit, és ezzel a P_m és P_ℓ játékosok teljesítik a feltételt. Az esetszétválasztás szerint akkor vagyunk a 2. esetben, ha $d_\ell < \lceil \frac{n}{2} \rceil$. Ekkor $n - d_\ell \geq \lceil \frac{n}{2} \rceil$, és pontosan $n - d_\ell$ részvény kellene a P_ℓ vágó játékosnak, hogy részvéncserélés során mindenki másnak csak a jövedelmezőbb oldalon legyen részvénye, de a feltevés miatt ennyi nincs neki. Tehát meg tudjuk oldani, hogy a P_ℓ játékos részvényei elfogyjanak a jövedelmezőbb vállalatban. $\square$

4.3. ÁLLÍTÁS. *Az RTSzV szubrutin során a játékosok arányosan osztozkodnak. Következésképpen az RTF algoritmus kimenete arányos felosztást szolgáltat.*

Bizonyítás. Igazoljuk, hogy az RTSzV szubrutin végrehajtása során egyetlen játékos vagyona sem csökken, azaz a $\sum_j \mu_i(C^j) \cdot d_i^j$ mennyiség legalább akkora a szubrutin végrehajtása után, mint amekkora az előtt volt minden i esetén. Az RTSzV szubrutin két lépésből áll: először egy C^j vállalatból egy C^* és egy C^{**} vállalat keletkezik. Ennek során az i . játékos vagyonának változása $\mu_i(C^*) \cdot d_i^j + \mu_i(C^{**}) \cdot d_i^j - \mu_i(C^j) \cdot d_i^j = d_i^j \cdot (\mu_i(C^*) + \mu_i(C^{**}) - \mu_i(C^j)) = d_i^j \cdot (\mu_i(C^j) - \mu_i(C^j)) = 0$.

Ha a szubrutin második részében végrehajtott részvéncsere során mondjuk az i . játékos a C^* vállalatban ε mennyiségű részvényét elcseréli ugyancsak ε mennyiségű részvényre a C^{**} vállalatban, akkor teljesül, hogy $\mu_i(C^*) \leq \mu_i(C^{**})$. Az i . játékos vagyonának változása tehát $\varepsilon \cdot (\mu_i(C^{**}) - \mu_i(C^*)) \geq 0$, így a részvéncsere során sem csökkenhet egyetlen játékos vagyona sem.

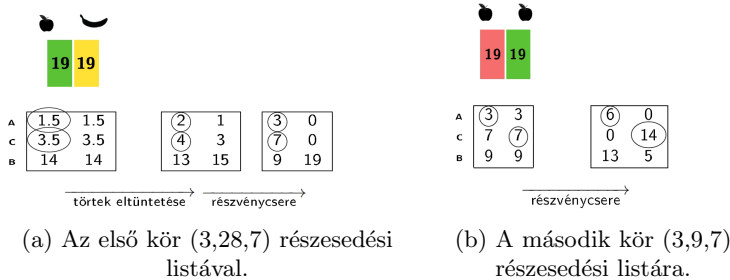
Tekintettel arra, hogy a játékosok vagyona kizárólag az RTSzV szubrutin során változhat az RTF algoritmus futása során, ezért az algoritmus kimeneteként kapott felosztásban az i . játékos vagyona legalább akkora lesz, mint amekkora az eredeti részesedése volt, azaz legalább d_i . Nekünk pedig éppen ezt kellett igazolnunk. $\square$

4.1. Példa. Alice, Bob és Charlie részesedése a Gyümölcs vállalatból rendre (3, 28, 7). A felosztáskor Bob vág először, mert övé a legnagyobb részesedés a vállalati részvényekből. Mivel $n = 38$, és $d_B \geq \lceil \frac{n}{2} \rceil = 19$, Bob közel felezi a Gyümölcs vállalatot két, 19 részvénnel rendelkező részre, az Alma és Banán vállalatokra. A vágás után Alice és Charlie jövedelmezőbb oldalt, mondjuk az Alma vállalatot választják, lásd a 3. ábra (A) részén. A törtek eltüntetése után Alice 2 Alma és 1 Banán, Bob 13 Alma és 15 Banán, Charlie pedig 4 Alma és 3 Banán részvénnel rendelkezik. A részvéncsere után Bob megkapja a teljes Banán vállalatot, hiszen már csak ő a tulajdonosa. A részesedése ezzel csökken, és folytatják az osztozkodást (3,9,7) részesedési listával. A vágó játékos ismét Bob lesz. Mivel $n = 19$, és

$d_B < \lceil \frac{n}{2} \rceil = 10$, Bobnak nincs elég részvénye a közel felező esethez, ezért pontosan felez, és a részvények inflálódnak, lásd a 3. ábra (B) részén. A részvénytársaság után redukáltuk a problémát két darab kétszemélyes problémára, és innentől a Közel Felező Algoritmust használhatjuk külön-külön a két vállalatra.

részeseések		(3, 8)		(3, 2)		(1, 2)		(1, 1)	
vállalatok		5 6		2 3		1 2		1 1	
törtek eltüntetése	A	15/11	18/11	6/5	9/5	1/3	2/3	0.5	0.5
	B	40/11	48/11	4/5	6/5	2/3	4/3	0.5	0.5
	A	2	1	2	1	0	1	1	0
	B	3	5	0	2	1	1	0	1
részvéncsere		3							
		2							
kiosztott részek		{6}		{6}		{6}, {1}		{2}, {1}	
		{6}		{6}		{6}, {1}		{6}, {1}, {1}	

2. ábra. A Gyümölcs vállalat egy kétszemélyes felosztása (3,8) részesedési listával.



3. ábra. Redukciós lépések a háromszemélyes RTF algoritmusban (3,28,7) részesedési listával.

4.1. TÉTEL. Ha $k \geq 2$, $n \geq 3$, és $n > k$, akkor a k -személyes Részvénytársaság Felosztás Algoritmus, melyben a résztvevők részesedéseinek összege n , felülről becsülhető a következőképpen:

$$D(k, n) \leq (2^{k-1} - 1) \cdot \lceil \log_2 n \rceil + 1. \quad (15)$$

Speciálisan, ha $k = 2$, akkor

$$D(2, n) = \lceil \log_2 n \rceil.$$

Bizonyítás. Ha $k = 2$, akkor az algoritmusunk megegyezik a Közel Felező Algoritmussal, vagyis a vágásszámuk is megegyezik [3].

Hajtsunk végre indukciót a lexikografikus rendezett (k, n) párokon. Az algoritmus leírását követve az alábbi felső becslést adhatjuk $D(k, n)$ -ra:

$$D(k, n) \leq 1 + \max \left\{ 2 \cdot D(k-1, n); D(k, \lceil \frac{n}{2} \rceil); \right. \\ \left. \left\{ D(k-i+1, \lceil \frac{n}{2} \rceil) + D(i, \lfloor \frac{n}{2} \rfloor) \quad \forall i = 2, 3, \dots, \lceil \frac{k+1}{2} \rceil \right\} \right\}.$$

Ahhoz, hogy bebizonyítsuk a tételt, be kell látnunk a következőt:

$$1 + \max \left\{ 2 \cdot D(k-1, n); D(k, \lceil \frac{n}{2} \rceil); \right. \\ \left. \left\{ D(k-i+1, \lceil \frac{n}{2} \rceil) + D(i, \lfloor \frac{n}{2} \rfloor) \quad \forall i = 2, 3, \dots, \lceil \frac{k+1}{2} \rceil \right\} \right\} \leq \\ \leq (2^{k-1} - 1) \cdot \lceil \log_2 n \rceil + 1.$$

Használjuk a tételbeli (15) felső becslést a maximumbeli elemekre. Egyszerűen belátható, hogy a legtöbb elhagyható közülük, mert $D(k, \lceil \frac{n}{2} \rceil)$ felső becslésével őket is felülről becsüljük. Írjuk fel a tételbeli (15) felső becslést $D(k, \lceil \frac{n}{2} \rceil)$ -ra:

$$D(k, \lceil \frac{n}{2} \rceil) \leq (2^{k-1} - 1) \cdot \lceil \log_2 \lceil \frac{n}{2} \rceil \rceil + 1. \quad (16)$$

Minden $i \in \{2, 3, \dots, \lceil \frac{k+1}{2} \rceil\}$ indexre felírható a következő:

$$D(k-i+1, \lceil \frac{n}{2} \rceil) + D(i, \lfloor \frac{n}{2} \rfloor) \leq \\ \leq (2^{k-i} - 1) \cdot \lceil \log_2 \lceil \frac{n}{2} \rceil \rceil + 1 + (2^{i-1} - 1) \cdot \lceil \log_2 \lfloor \frac{n}{2} \rfloor \rceil + 1 \\ = (2^{k-i} + 2^{i-1} - 2) \cdot \lceil \log_2 \lceil \frac{n}{2} \rceil \rceil + 2,$$

és felülről becsülhetjük az egyenlőtlenség jobb oldalát az előzőekben felírt (16) felső becsléssel, majd rendezhetjük:

$$(2^{k-i} + 2^{i-1} - 2) \cdot \lceil \log_2 \lceil \frac{n}{2} \rceil \rceil + 2 \leq (2^{k-1} - 1) \cdot \lceil \log_2 \lceil \frac{n}{2} \rceil \rceil + 1 \\ \frac{1}{\lceil \log_2 \lceil \frac{n}{2} \rceil \rceil} \leq (2^{k-1} - 2^{k-i} - 2^{i-1} + 1) = (2^{k-i} - 1) \cdot (2^{i-1} - 1) \quad (17)$$

Mivel $n \geq 3$ esetén az előző (17) egyenlőtlenség bal oldalán mindig 1-nél nem nagyobb, a jobb oldalán pedig 1-nél nem kisebb tényezőket szorzunk, a tételbeli (15) felső becsléssel a következőt kapjuk:

$$D(k, n) \leq 1 + \max \left\{ 2 \cdot D(k-1, n); D(k, \lceil \frac{n}{2} \rceil) \right\}.$$

Ahhoz, hogy a tételbeli (15) felső becslés mindig igaz legyen, a következő két egyenlőtlenséget kell még belátnunk:

$$1 + 2 \cdot D(k-1, n) \leq (2^{k-1} - 1) \cdot \lceil \log_2 n \rceil + 1, \quad (18)$$

$$1 + D(k, \lceil \frac{n}{2} \rceil) \leq (2^{k-1} - 1) \cdot \lceil \log_2 n \rceil + 1. \quad (19)$$

Becsüljük felülről az előzőekben felírt (18) egyenlőtlenség bal oldalát, hogy eljussunk a jobb oldaláig:

$$\begin{aligned} 1 + 2 \cdot D(k-1, n) &\leq 1 + 2 \cdot [(2^{k-1-1} - 1) \cdot \lceil \log_2 n \rceil + 1] = \\ &= 3 + (2^{k-1} - 2) \cdot \lceil \log_2 n \rceil \\ &= (2^{k-1} - 1) \cdot \lceil \log_2 n \rceil + 1 - \underbrace{\lceil \log_2 n \rceil + 2}_{\leq 0 \text{ } (\forall n \geq 3)} \\ &\leq 0 \text{ } (\forall n \geq 3) \end{aligned}$$

Valamint becsüljük felülről az előzőekben felírt (19) egyenlőtlenség bal oldalát, hogy eljussunk a jobb oldaláig, és közben használjuk ki, hogy $\forall x \in \mathbb{R} : \lceil \log_2 \lceil x \rceil \rceil = \lceil \log_2 x \rceil$:

$$\begin{aligned} 1 + D(k, \lceil \frac{n}{2} \rceil) &\leq 1 + (2^{k-1} - 1) \cdot \lceil \log_2 \lceil \frac{n}{2} \rceil \rceil + 1 = \\ &= 1 + (2^{k-1} - 1) \cdot \lceil \log_2 \frac{n}{2} \rceil + 1 \\ &= 1 + (2^{k-1} - 1) \cdot \lceil \log_2 n - \log_2 2 \rceil + 1 \\ &= 1 + (2^{k-1} - 1) \cdot (\lceil \log_2 n \rceil - 1) + 1 \\ &= (2^{k-1} - 1) \cdot \lceil \log_2 n \rceil + 1 - \underbrace{2^{k-1} + 2}_{\leq 0 \text{ } (\forall k \geq 2)} \\ &\leq 0 \text{ } (\forall k \geq 2) \end{aligned}$$

Ezzel beláttuk, hogy a tételbeli (15) felső becslés mindig elegendő vágás lesz.

□

Köszönetnyilvánítás

Jelen munka az OTKA K108383 és az MTA KEP-6/2017 sz. kutatási projekt támogatásával készült. Továbbá köszönettel tartozunk a lektornak, az MTA Mechanizmus tervezés kutatócsoportjának és Tasnádi Attilának.

Hivatkozások

- [1] JEFF EDMONDS AND KIRK PRUHS: *Cake cutting really is not a piece of cake*, ACM Transactions on Algorithms (TALG), Vol. **7** No. **4**, p. 51 (2011).
- [2] SHIMON EVEN AND AZARIA PAZ: *A note on cake cutting*, Discrete Applied Mathematics, Vol. **7** No. **3**, pp. 285-296 (1984).
- [3] JACK ROBERTSON AND WILLIAM WEBB: *Cake-cutting algorithms: Be fair if you can*, p. 177 (1998).
- [4] ATTILA TASNÁDI: *Igazságos elosztások*, Budapesti Corvinus Egyetem, Typotex, p. 147 (2014).



Fleiner Tamás 1971-ben született Budapesten. A budapesti Fazekas Gimnázium speciális matematika tagozatán tett érettségit követően szerzett okleveles matematikus diplomát az ELTÉ-n, majd az amszterdami CWI-ben töltött 4 év után "Stable and crossing structures" című disszertációját az eindhoveni TU/e egyetemen 2000-ben védte meg. Egy évig az MTA RAM-KI fiatal kutatója, majd Magyary Zoltán poszt-doktori ösztöndíjas az ELTE Operációkutatási Tanszékén. 2003-tól Bolyai-ösztöndíjasként a BME Számítástudományi és Információelméleti

Tanszékén adjunktus, 2006-tól docens, 2014-ben pedig habilitál. Kutatási területe a diszkrét matematika és közgazdaságtan határterülete, 65 tudományos közlemény szerzője, több mint 300 független hivatkozással. Több OTKA-pályázat témavezetője, nemzetközi kutatások résztvevője, a Kürschák József Matematikai Tanulóverseny szervezőbizottságának elnöke. Jelenleg a BME-n és az AIT-n oktat, társasházi közös képviselő, műkedvelő kerékpár- és varrógépműszerész, valamint gerillakertész.

FLEINER TAMÁS

Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudományi és Információelméleti Tanszék
1117 Budapest, Magyar tudósok körútja 2., I épület, IB132
fleiner@cs.bme.hu



Romsics Erzsébet Kalocsán született 1994. július 10-én, jelenleg Budapesten él. A gimnáziumi tanulmányait a kalocsai Szent István Gimnáziumban végezte, majd alapszakos diplomát szerzett a Budapesti Műszaki és Gazdaságtudományi Egyetem matematika alapszakán 2016-ban. Jelenleg is az egyetem matematikus mesterszakán tanul. Az alapszakos témavezetője Fleiner Tamás volt, jelenleg pedig Biró Péter, az MTA Közgazdaságtudományi Intézetének tagja. Az egyetem alatt több tanszéken töltött be demonstrátori pozíciót, javában mérnöki matematika tárgyakat oktatott. 2016-ban az igazságos tortaosztásról írt dolgozata első helyezést ért el az egyetemi tudományos konferencián. Kisebb feladatokban részt vett az MTA Mechanizmustervezés Kutatócsoport munkájában

2017-ben, majd 2018-tól az MTA Tudánypolitikai és Tudományelemzési Osztályával dolgozik együtt egy hálózatelméleti projekten. 2017-ben a BME Természettudományi Kara a 'Kar Kiváló Hallgatója' díjat adományozta neki.

ROMSICS ERZSÉBET

Budapesti Műszaki és Gazdaságtudományi Egyetem
Sztocasztika Tanszék
1111 Budapest, Eötvös József u. 1.
romsics.csore@gmail.com

ALGORITHMS FOR CAKE CUTTING WITH EQUAL AND UNEQUAL SHARES

TAMÁS FLEINER, ERZSÉBET ROMSICS

An unceasing problem of our prevailing society is the fair division of goods. The problem of fair cake cutting is dividing a heterogeneous and divisible resource, the cake, among k players who value pieces according to their own measure function. The goal is to assign each player a not necessarily connected part of the cake that the player evaluates at least as much as her equal fair share.

In this paper, we present two new algorithms. Our first algorithm guarantees a piece strictly larger than $\frac{1}{k}$ of the whole cake for certain measure functions. In the second setting, we investigate the problem of unequal shares, where each player needs to be assigned a piece that she evaluates at least as much as her predefined fair share. We present an algorithm that delivers such a solution and in some cases, runs faster than all known algorithms. In both problems, we establish upper bounds on the number of cuts our algorithms execute.