BUDAPESTI CORVINUS EGYETEM

Közgazdasági és Gazdaságinformatikai Doktori Iskola

TÉZISGYŰJTEMÉNY

Illés Ferenc

A Shapley-érték komplexitása és becslése

című PhD értekezéséhez

Témavezető:

Dr. Csóka Péter

Budapesti Corvinus Egyetem, Pénzügy Intézet

Budapest, 2023

1. Témaválasztás

A kooperatív játékelmélet számos társadalmi dilemma illetve pénzügyi és gazdasági probléma modellje. Általában akkor alkalmazzuk, ha egy közösség által elérhető eredmény meghaladja az egyénenként elérhető eredmények összegét (szinergia), vagy egy tevékenység teljes költsége alacsonyabb, mint a részfeladatok költségeinek összege (megtakarítás) illetve teljes kockázata kisebb, mint a részfeladatok aggregált kockázata (diverzifikáció). A fő kérdés pedig természetesen az, hogy hogyan osszuk el a többletet a szereplők között valamilyen „igazságos" módon.

Gyakorlati alkalmazások

A közösségi döntési problémákat általában szavazással oldjuk meg. Az Európai Gazdasági Közösség (vagy más néven Közös Piac) nevű nemzetközi megállapodást 1957-ben hat ország írta alá, melyek nem egyenlő súlyozással, többségi szavazással hoztak döntéseket. Egy pozitív döntéshez legalább 12 igen szavazat kellett a 17-ből, melyet a tagországok a 1. táblázatban megadott súlyok alapján kaptak (Taylor és Pacelli, 2008). A szavazásnál a „többlet" abból

Ország	Szavazatok száma	Népesség	
Franciaország	4	44	millió fő
NSZK	4	51	millió fő
Olaszország	4	49	millió fő
Belgium	2	9	millió fő
Hollandia	2	11	millió fő
Luxemburg	1	308	ezer fő

1. táblázat. EGK országok szavazóereje és népessége

adódik, hogy nem szükséges a résztvevők teljes egyetértése a döntés meghozatalához, ennek „szétosztása" pedig azt jelenti, hogy rendeljünk minden játékoshoz egy számot, mely reálisan méri az ő érdekérvényesítő képességét. A fenti példa jól illusztrálja, hogy a játékosok súlyai erre nem megfelelőek. Úgy tünik, ugyanis, hogy Luxemburg erősen túlreprezentált, mert negyedakkora a szavazóereje, mint a százötvenszer akkora Nyugat-Németországé, ha azonban kicsit jobban megnézzük a számokat, láthatjuk, hogy a fenti rendszerben nem képzelhető el egyetlen olyan döntési szituáció sem, ahol Luxemburg szavazata számítana. Ez ugyanis csak akkor fordulhatna elő, ha a másik öt ország úgy szakadna két pártra, hogy az adott kérdést támogatók szavazatainak összértéke 11, az ellenzőké pedig 5. Ezt azonban lehetetlen 5 darab páros számból kirakni.

Egy hétköznapibb alkalmazás a következő. Egy kereskedelmi bank treasury osztályán harminc bróker dolgozik, akik a piacon elérhető többezer lehetséges eszközből állíthatnak össze tetszőleges portfóliót. Az osztály teljes költségkerete harminc milliárd forint, a bank aggregált portfóliójának tőkekövetelménye ezt az értéket egyetlen időpillanatban sem haladhatja meg. A tőkekövetelmény a 99,9%-os CVaR értéke[1], melyet a portfóliót alkotó eszközök hozamának historikus adataiból becsült várható értékek és kovarianciamátrixuk alapján számolnak ki, az eszközök hozamának többdimenziós normális eloszlását feltételezve. Az osztályvezető az egyenlőség elvét szem előtt tartva, mindegyik brókernek 1 milliárd forintos keretet hagy jóvá, amit a kollégák külön-külön be is tartanak, azonban, amikor a bank aggregált portfóliójának tőkekövetelményét kiszámolják, az mindössze 13 milliárd forint. A különbözet oka, hogy a tőkekövetelmény, a hozamok közti diverzifikációs hatás miatt nem additív. Ráadásul, az, hogy egy bróker által tartott eszközök hogyan változtatják meg az aggragált portfólió kockázatát, az összes többi eszközzel való korrelációtól is függ. Lehetséges, hogy valaki önmagában látszólag nagyon-nagy kockázatot vállal, de ez a teljes portfólióra nézve valójában fedezetet képez.

[1] *Conditional Value at Risk*, azaz feltételes kockáztatott érték egy gyakran használt kockázati mérték.

Hogyan osszuk szét a portfólió teljes kockázatát az egyes eszközök között? A probléma a szakirodalomban tőkeallokáció vagy kockázatfelosztás (*risk capital allocation*) néven terjedt el (ld. például Colini-Baldeschi et al. (2018), Csóka et al. (2009) illetve Csóka és Pintér (2016)).

A kooperatív játékok számtalan egyéb helyen fellépnek a közgazdaságtan, a pénzügyek, sőt a statisztika és a jog területén is. Az általános megoldást Lloyd Stowell Shapley adta meg 1953-ban *A value for n-person games* című cikkében. Bizonyítható, hogy Shapley megoldása, melyet azóta Shapley-értéknek neveznek, az egyetlen általános, azaz minden játékra működő megoldási koncepció, mely bizonyos igazságossági feltételeknek eleget tesz. Szépséghibája, hogy – az egyszerűen felírható formula és intuitív tartalma ellenére – konkrét játékokra nem feltétlenül tudjuk az értékét pontosan meghatározni, ha a játékosok száma túl nagy, ugyanis a képlet a játékosok számában nem polinomiális számú tag összegéből áll. A disszertáció tartalma, ha egy mondatban kéne összefoglalni: Hogyan lehet (illetve nem lehet) egy kooperatív játék Shapley-értékét hatékonyan kiszámolni vagy megbecsülni?

2. Kutatási előzmények

A kooperatív játékok ma megszokott matematikai formalizmusának alapjai Neumann János és Oskar Morgenstern *Theory of Games and Economic Behaviour* című, először 1944-ben megjelent, könyvéből erednek.

2.1. Definíció. *Átruházható hasznosságú (transferable utility), vagy röviden TU-játékon* (N, v) *rendezett párt értünk, ahol* N *egy nemüres, véges halmaz,* v *pedig* $v : 2^N \to \mathbb{Z}$ *halmazfüggvény, melyre* $v(\emptyset) = 0$*, ahol* 2^N *az* N *halmaz hatványhalmaza, azaz összes részhalmazainak halmaza.*

Az N halmaz elemeit játékosoknak, részhalmazait koalícióknak, a v függvényt pedig karakterisztikus függvénynek nevezzük, a $v(S)$ számot pedig az $S \subseteq N$ koalíció értékének[2]. Jelölje $\mathcal{G}_N$, az N-hez, mint játékosok halmazához tartozó összes lehetséges TU-játékok (vagy röviden, az N fölötti TU-játékok) halmazát. Ezek a koalíciónkkénti műveletekre egy $\mathbb{Z}$ fölötti modulust alkotnak.

2.2. Definíció. *Legyen* $N \neq \emptyset$ *halmaz. Érték alatt* $\varphi : \mathcal{G}_N \to \mathbb{Q}^N$ *függvényt értünk.*

A játék értéke a játék egy lehetséges megoldási koncepcióját jelenti, abban az értelemben, hogy minden játékban, minden játékoshoz hozzárendel egy számot, melyet úgy lehet interpretálni, hogy az adott játékos ennyit kapjon az osztozkodásnál, ha a játék (a karakterisztikus függvény) megszerezhető pénzt ír le, illetve ennyit fizessen, ha költséget ír le. Ha $\varphi : \mathcal{G}_N \to \mathbb{Q}^N$ egy érték, akkor definíció szerint minden $g \in \mathcal{G}_N$-re $\varphi(g)$ egy $N \to \mathbb{Q}$ függvény, melybe be lehet helyettesíteni egy $i \in N$ játékost, megkapva így az i játékos $\varphi(g)(i) \in \mathbb{Q}$ részesedését a g játékban. A jelölések egyszerűsítése érdekében a továbbiakban ezt $\varphi_i(g)$-vel, vagy ha nem okoz félreértést, elhagyva a játékra vonatkozó jelölést, φ_i-vel jelöljük.

2.3. Definíció. *Legyen a* $g \in \mathcal{G}_N$ *játékban* $S \subseteq N$ *egy koalíció,* $i \in N$ *egy játékos, melyre* $i \notin S$*. Az* i *játékos* S *koalíció értékéhez való határ-hozzájárulása (marginal contribution) alatt a*

$$v_i'(S) = v(S \cup \{i\}) - v(S)$$

számot értjük.

2.4. Definíció. *A* $\varphi : \mathcal{G}_N \to \mathbb{Q}^N$ *érték*

- *egyenlően kezelő (ETP - Equal Treatment Property), ha minden* $g \in \mathcal{G}_N$ *játékban, bármely két olyan* i *és* j *játékosra, melyekre* $v_i'(S) = v_j'(S)\ \forall\ i, j \notin S \subseteq N$*-re,* $\varphi_i = \varphi_j$*,*
- *hatékony (EFF - Efficiency), ha* $\sum \varphi_i = v(N)$ *mindegyik* $g \in \mathcal{G}_N$ *játékban,*

[2] Mint azt a bevezetőben lévő példáknál láttuk, a karakterisztikus függvény költséget is leírhat, de általánosságban a „koalíció értéke” kifejezést fogjuk használni.

- *nulla játékos tulajdonságú (NPP - Null Player Property): ha mindegyik $g \in \mathcal{G}_N$ játékban, ha egy i játékosra $v'_i(S) = 0 \quad \forall i \notin S \subseteq N$-re, akkor $\varphi_i = 0$,*
- *additív (ADD), ha $\varphi(g_1 + g_2) = \varphi(g_1) + \varphi(g_2) \quad \forall g_1, g_2 \in \mathcal{G}_N$-re.*

Vegyük észre, hogy az ETP, EFF, és NPP tulajdonság játékonként értendő, azaz minden rögzített $g \in \mathcal{G}_N$ játék esetén a $\varphi(g) : N \to \mathbb{Q}$ függvényre szab ki feltételeket, míg az ADD tulajdonság az egyes játékok értékeit köti össze. Az EFF tulajdonság azt jelenti, hogy a nagykoalíció értékét/költségét osztjuk szét a játékosok között, míg a másik három valamiféle igazságossági feltétel:

- ha két játékos egy játék szempontjából „egyforma", akkor abban a játékban ugyanazt a kifizetést kapják,
- ha egy játékos egy játékban „haszontalan", akkor a kifizetése (abban a játékban) nulla legyen,
- ha egy játékot két másik összegére bontunk, és külön-külön játszuk le őket, ezzel senki se járhasson jobban vagy rosszabbul.

A TU-játékok legismertebb megoldási koncepciójának legfőbb eredménye a következő:

2.1. Tétel (Shapley, 1953). *Minden $N \neq \emptyset$ esetén $\exists!$ az ADD, ETP, NPP és EFF axiómáknak eleget tevő φ érték.*

A 2.1. tétel nem csak egzisztenciatétel, a benne szereplő Shapley-érték explicite is megadható a következőképpen.

2.5. Definíció. *Legyen $g = (N, v)$ egy TU-játék $n = |N|$ játékossal. A játékosok egy érkezési sorrendjén, vagy röviden sorrendjén vagy permutációján egy $o : N \leftrightarrow \{1, 2, \ldots, n\}$ kölcsönösen egyértelmű függvényt értünk. Ha $i \in N$ egy játékos, o egy sorrend és $o(i) = j \in \mathbb{N}$ ez azt jelenti, hogy az i játékos j-ediknek érkezik. A játékosok összes lehetséges sorrendjeiből álló $\{o : N \leftrightarrow \{1, 2, \ldots, n\}$ kölcsönösen egyértelmű függvény$\}$ halmazt $\mathcal{O}(N)$-el jelöljük. Nyilván $|\mathcal{O}| = n!$. Ha $o \in \mathcal{O}(N)$ egy sorrend, $i \in N$ egy játékos, akkor $S_o(i) \subseteq N$ jelöli az i előtt érkező játékosokat: $S_o(i) = \{i' \in N \mid o(i') < o(i)\}$. A $g = (N, v)$ TU-játék Shapley-értéke*

$$Sh_i = \sum_{o \in \mathcal{O}(N)} \frac{v'_i(S_o(i))}{n!} \qquad \forall i \in N\text{-re.} \tag{1}$$

Az i játékos Shapley-értékének kiszámolásakor figyelembevesszük a játékosok összes lehetséges sorrendjét. A $v'_i(S_o(i))$ érték azonban nem változik attól, hogy az i játékost megelőző és követő játékosok egymás közti sorrendje megváltozik, de senki, aki korábban előtte volt, nem kerül mögé és vice versa azaz, ha az i játékos ugyanahhoz a koalícióhoz csatlakozik. Ha ezeket, a nagyon sokszor előforduló, ekvivalens tagokat, melyek száma $|S_o(i)|! \cdot (n - |S_o(i)| - 1)!$ összevonjuk, akkor az

$$Sh_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (n - |S| - 1)!}{n!} \cdot v'_i(S) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \frac{1}{\binom{n-1}{|S|}} \cdot v'_i(S) \tag{2}$$

formulához jutunk. Shapley eredeti cikkében ez a nehezebben értelmezhető képlet szerepel.

Egy TU-játék definíció szerint egy $2^n - 1$ elemű halmazon értelmezett függvény, azaz praktikusan egy $2^n - 1$ elemű sorozat, melyhez képest a (1) formula, mint $n!$ tagszámú összeg nem értékelhető ki polinom-időben, azonban a (2) formula, mely már csak 2^{n-1} tag összege, elvileg hatékonyan kiszámolható. Egy TU-játékot azonban nem praktikus és nem is szokás az összes koalíció értékének felsorolásával megadni, általában valami tömörebb formában adjuk meg, a játékosok számában polinom-sok paramétert tartalmazó formulával, ahogyan a következő példák mutatják:

2.1. Példa (TU-játékok).

Szavazási játék: Legyen $n \in \mathbb{N}$, $N = \{1, 2, \ldots, n\}$, $w = (w_1, w_2, \ldots, w_n) \in \mathbb{N}^n$ és $q \in \mathbb{Z}$. Minden $S \subseteq N$-re legyen

$$v(S) = \begin{cases} 1, & \text{ha} \quad \sum\limits_{i \in S} w_i \geq q, \\ 0, & \text{ha} \quad \sum\limits_{i \in S} w_i < q. \end{cases}$$

Repülőtér-játék: Legyen $n \in \mathbb{N}$, N egy n elemű halmaz, minden $i \in N$-re $c_i \in \mathbb{N}$, és legyen a karakterisztikus függvény $v(S) = \max\{c_i | i \in S\}$.

Csődjáték: Legyen $A \in \mathbb{Z}^+, n \in \mathbb{N}, \quad l_1, l_2, \ldots l_n \in \mathbb{N}$ és legyen (N, v) az a TU-játék, melynek karakterisztikus függvénye $v(S) = \max\{0, A - \sum_{j \notin S} l_j\}$.

Ez azt jelenti, hogy TU-játékokat sok (akár több tízezer) játékosra is meg lehet adni egy kezelhető méretű adathalmazként, azonban a Shapley-érték ilyenkor már sem az (1) sem a (2) formula segítségével nem számolható ki a gyakorlatban. Ennélfogva releváns és értelmes kérdéssé válik egy adott játékosztály esetén a Shapley-érték kiszámítása, mint algoritmikus probléma.

Az egyik legkorábbi ezzel kapcsolatos nemtriviális eredmény Megiddo 1978-as *Computational Complexity of the Game Theory Approach to Cost Allocation for a Tree* című cikkében jelent meg. Legyen a játékosok halmaza $N = \{1, 2, \ldots, n\}$ és legyen $G(N \cup \{0\}, E, d)$ irányított súlyozott élű gyökeres fa, ahol a 0 csúcs a gyökér, és jelölje $d(i, j) \geq 0$ a súlyozást az éleken. Legyen minden $S \subseteq N$-re $v(S)$ az összes olyan él súlyainak összege, melyeket valamelyik S-beli csúcshoz vezető, gyökérből induló út tartalmaz. Ez a játék megadható $n-1$ darab súllyal, így a Shapley-értékének kiszámítása értelmes és érdekes kérdés. Belátható, hogy ez megoldható egy egyszerű gráf-algoritmussal lineáris időben és tárban.

A fent említett szavazási játékok a Shapley-érték bonyolultsága szempontjából a másik végletet képviseli. Könnyen látható, hogy már annak eldöntése, hogy egy adott játékos Shapley-értéke nulla-e egy ilyen játékban a köztudottan NP-teljes részletösszeg-probléma általánosítása (illetve maga a részletösszeg-probléma, egy 1 súlyú játékosra). Mi több, Deng és Papadimitriou (1994) szerint ez a probléma `#P`-teljes.

A komplexitásra vonatkozó eredmények eléggé behatárolják a Shapley-érték kiszámításának lehetőségeit, van azonban két lehetőség, ami a gyakorlatban jól működhet, eredetileg mindkettőt maga Shapley javasolta még a '60-as években.

- Noha ezt a terminológiát sohasem használták, Mann és Shapley (1962) egy pszeudopolinomiális algoritmust ad szavazási játékok esetére, Cantor ötlete alapján, generátorfüggvény rekurzív kiszámításával. Az algoritmus a (szokásos értelemben) polinomiális olyan részjátékosztályokon, ahol a súlyok a játékosok számának egy polinomjával korlátozhatóak.
- Mann és Shapley (1960) javasolja a Shapley-érték Monte-Carlo szimulációval történő becslését.

3. Az értekezés eredményei

Az értekezés három ponton kapcsolódik a téma irodalmához.

- Bizonyítunk egy a Shapley-érték komplexitására vonatkozó új eredményt, egy eddig nem vizsgált játékosztályra. Bemutatunk néhány kisebb eredményt a Shapley-érték közelíthetőségére és játékok reprezentációjára vonatkozóan.
- Bemutatunk egy új Monte-Carlo módszert a Shapley-érték becslésére.
- Bevezetjük a lineárisan reprezentálható játékok fogalmát, és bemutatunk egy a Mann és Shapley (1962) által szavazási játékokra javasolt módszerrel komplexitását tekintve egyenértékű, de általánosabb algoritmust és ennek repülőtér-játékokra való alkalmazását és következményét.

3.1. TU-játékok reprezentációja és komplexitása

Mivel nem praktikus egy TU-játékot a koalíciók értékeinek felsorolásával megadni, első lépésként, tisztázzuk, hogy mit is értünk egy játék megadásán, ha azt egy algoritmus inputjaként szeretnénk értelmezni. Ehhez jelölje $\forall n \in \mathbb{N}$-re $\bar{n} = \{1, 2, 3, \ldots, n\}$ az első n természetes szám halmazát, melyek a játékosokat fogják reprezentálni (ez természetesen csak technikai feltevés, nem jelenti az általánosság valódi megszorítását).

3.1. Definíció. *Azt mondjuk, hogy játékok egy $G \subseteq \bigcup_{n\in\mathbb{N}} \mathcal{G}_{\bar{n}}$ osztálya polinomiálisan reprezentálható, ha létezik egy véges Σ ABC, mely tartalmaz egy $*$-al jelölt speciális üres szimbólumot valamint a 0 és 1 szimbólumokat*[3] *és egy Σ feletti legalább három szalagos T_G Turing-gép, valamint $\forall g = (\bar{n_g}, v_g) \in G$-re olyan $x_g \in \Sigma_0^* = \bigcup_{i=0}^{\infty} (\Sigma \setminus \{*\})^i$ szó, melynek hossza n_g-ben polinomiális, és T_G első szalagjára egy n_g hosszú u 0-1-sorozatot, második szalagjára pedig x_g-t írva, polinom-időben kiszámolja a g játékban annak a koalíciónak az értékét, melynek indikátora u. Röviden: $T_G(u, x_g) = v_g(S)$.*

A polinomiális reprezentáció azt jelenti, hogy egy játékosztály megfelelő módon „tömöríthető" ahhoz, hogy Shapley-értékének komplexitása értelmes kérdés legyen. A 2.1. példa minden játéka egy ilyen tömör reprezentációval van adva, például a szavazási játékok esetén ez $n + 1$ darab paraméter, a játékosok súlyai és a kvóta.

3.2. Definíció. *Legyen $G \subseteq \bigcup_{n\in\mathbb{N}} \mathcal{G}_{\bar{n}}$ egy polinomiálisan reprezentálható játékosztály. Ekkor a definícióban szereplő x_g szavak egy $L_G = \{x_g \in \Sigma_0^* \mid g \in G\}$ nyelvet alkotnak. Ezt az L_G nyelvet a definícióban szereplő T_G Turing-géppel együtt, az (L_G, T_G) rendezett párt a G játékosztály egy polinomiális reprezentációjának nevezzük.*

Mivel egy játéknak több polinomiális reprezentációja is lehet, ez felvet egy új problémát, nevezetesen, hogy két formálisan különböző reprezentáció ugyanazt a játékot definiálja-e. Az alábbi állítás szerint bizonyos esetekben már ez is „megoldhatatlan":

3.1. Tétel. *co-NP-teljes annak eldöntése, hogy a 2.1 példa szerint megadott két lineáris reprezentáció ugyanazt a szavazási játékok reprezentálja-e.*

Bizonyítás. Részletesen kiírva, azt kell eldönteni, hogy $2n + 2$ darab $w_1, w_2, \ldots, w_n, q$ és $w'_1, w'_2, \ldots, w'_n, q'$ pozitív egész szám esetén igaz-e hogy minden $S \subseteq \{1, 2, \ldots, n\}$-re

$$\sum_{i\in S} w_i \geq q \Leftrightarrow \sum_{i\in S} w'_i \geq q'.$$

Ez pedig már a $w_i = w'_i$ és $q' = q-1$ speciális esetben is a $\nexists S \subseteq \{1, 2, \ldots, n\}$ hogy $\sum_{i\in S} w_i = q$, azaz a részletösszeg-problémával ekvivalens, tehát már ez is NP-nehéz. A probléma továbbá co-NP-ben van, mert ha két játék nem egyezik meg, akkor erre a bizonyíték egy olyan koalíció, melyek értéke a két játékban különböző. □

A polinomiális reprezentáció illetve reprezentálhatóság lehetővé teszi, hogy a Shapley-érték kiszámításának problémáját általánosan definiáljuk:

3.3. Definíció. *Azt mondjuk, hogy a $G \subseteq \bigcup_{n\in\mathbb{N}} \mathcal{G}_{\bar{n}}$ játékosztályra a Shapley-érték polinom-időben kiszámolható, ha létezik G-nek (L_G, T_G) polinomiális reprezentációja, melyhez létezik olyan T legalább három szalagos Turing-gép, melynek első szalagjára inputként egy L_G-beli szót, mint egy $g \in G$ játék $x_g \in \Sigma_0^*$ reprezentációját, a második szalagjára pedig T_G leírását írva, (T_G-t szubrutinként használva), polinom-időben kiszámolja a g játék Shapley-értékét.*

[3] Így tudunk számokat ábrázolni;

Megjegyzés. A definíció csak azt követeli meg egy adott játékosztályhoz, hogy létezzen (legalább egy) olyan polinomiális reprezentációja, ami alapján a Shapley-érték gyorsan kiszámolható. Elvileg előfordulhat, hogy egy játékosztály többféleképpen is reprezentálható, de az egyikből ki lehet számolni a Shapley-értéket, a másikból viszont nem. Ez nem jelent ellentmondást, csak ekkor nyilván egyik reprezentációból nem lehet kiszámolni a másikat.

A Shapley-érték komplexitásával kapcsolatos eredmények két részre bonthatóak: bizonyos játékokra polinomiális (Megiddo, 1978), más esetekben valamely NP-nehéz eldöntési probléma általánosítása (Deng és Papadimitriou, 1994), azonban a dichotómia egyelőre nyitott kérdés[4]. A következőkben az utóbbi kategóriába tartozó új eredményt ismertetünk.

A *tartozásos játékok* a 2.1 példa csődjátékához nagyon hasonló (polinomiálisan reprezentálható) játékosztály, melyet Csóka és Herings (2019) vezetett be. A játék karakterisztikus függvénye

$$v(S) = \begin{cases} \min\{A, \sum\limits_{j \in S \setminus \{0\}} l_j\} & \text{ha} \quad 0 \in S, \\ \max\{0, A - \sum\limits_{j \notin S \cup \{0\}} l_j\} & \text{ha} \quad 0 \notin S. \end{cases}$$

ahol $N = \{0, 1, \ldots, n-1\}$, A és $l_1, l_2, \ldots l_{n-1}$ pozitív paraméterek, melyekre $A < \sum l_i$. A játék szereplői egy 0-val jelölt játékos, például egy cég, és véges sok hitelező, l_i jelöli a kötelezettséget az i-edik hitelező felé. Az A paraméter a 0-s játékos eszközeinek értéke, melynek értéke kevesebb, mint a kötelezettségek összege, így a cég csődben van. A karakterisztikus függvény azt írja le, hogy az egyes koalíciók mennyit tudnak behajtani az összes követelésükből. A csődbemenő cég is szereplője a játéknak és azok a koalíciók, melyeknek ő is egyik tagja, elsőbbséget élveznek a kötelezettségek kielégítésében. A 0 játékost nem tartalmazó koalíciók csak a maradékon osztozkodhatnak. A játékkal kapcsolatos további részletekért és példákért ld. Csóka (2018). Egy 2022-ben megjelent cikkben (Csóka et al., 2022) belátjuk az alábbi tételt:

3.2. Tétel. *NP-nehéz annak eldöntése, hogy két tartozásos játékban a csődbemenő cég Shapley-értéke megegyezik-e, azaz az* $\{(\mathcal{L}_1, \mathcal{L}_2) \mid Sh_1 = Sh_2\}$ *nyelv NP-nehéz, ahol* $\mathcal{L}_i$ *tartozásos játékok,* Sh_i *pedig a* 0*-s játékos Shapley-értéke az egyes játékokban.*

Bizonyítás. Felhasználjuk, hogy a

$$\text{SUBSET-SUM} = \left\{ (a_1, a_2, \ldots, a_k, M) \in \mathbb{Z}^{k+1} \mid \exists H \subseteq \{1, 2, \ldots, k\} \mid \sum_{i \in H} a_i = M \right\}$$

nyelv NP-teljes. Könnyű belátni, hogy ennek a

$$\text{HALFSUM} = \left\{ (a_1, a_2, \ldots, a_k) \in \mathbb{Z}^k \mid \exists H \subseteq \{1, 2, \ldots, k\} \mid \sum_{i \in H} a_i = \frac{\sum\limits_{i=1}^{k} a_i}{2} \right\}$$

speciális esete is az. Ezt fogjuk visszavezetni a tartozásos játékok Shapley-értékére.

Legyen $(a_1, a_2, \ldots, a_k) \in \mathbb{Z}^k$. Legyenek a kötelezettségek $l_i = a_i$ $A = \frac{\sum a_i}{2}$ és legyen $\mathcal{L}_1$ egy tartozásos játék az l_i és A paraméterekkel. Az $\mathcal{L}_2$ játékban ugyanezek lesznek a kötelezettségek, de az eszközök értékét 1-el csökkentjük. Azt fogjuk megmutatni, hogy az $\mathcal{L}_1$ és $\mathcal{L}_2$ játékban a 0 játékos Shapley-értéke pontosan akkor különböző, ha $(a_1, a_2, \ldots, a_k) \in$ HALFSUM.

Hitelezők egy $S \subseteq \{1, 2, \ldots, n-1\}$ koalíciójának, hitelezők halmazára vonatkozó komplementerét jelölje $S^c = \{1, 2, \ldots, n-1\} \setminus S$. A könnyebb érthetőség kedvéért nevezzük hitelezők egy S koalícióját erősnek, ha a követeléseik összege meghaladja az eszközök értékét: $\sum_{i \in S} l_i > A$,

[4] Azaz sem olyan játékot nem ismerünk, ahol egyik eset sem áll fent, sem olyan eredmény nincs, hogy ilyen játék nem létezik.

közepesnek, ha éppen egyenlő: $\sum_{i\in S} l_i = A$ és gyengének, ha kevesebb: $\sum_{i\in S} l_i < A$. Mivel az eszközök értéke éppen fele az összes követelésnek, minden S esetén S és S^c közül vagy pontosan az egyik erős, a másik gyenge, vagy mindkettő közepes. Egy gyenge koalíció értéke önmagában nulla, mert a követelések hátrasoroltak, és a komplementer koalíciót nem lehet kifizetni. Amikor a 0 játékos egy ilyenhez csatlakozik, akkor viszont az összes követelés kifizethető, ezért a 0 játékos határ-hozzájárulása megegyezik a követelések összértékével. Egy erős koalíció összes követelése akkor sem elégíthető ki, miután a 0 játékos csatlakozott a koalícióhoz, ezért a komplementer koalíció teljes követelését, mely ilyenkor kevesebb az eszközök értékénél áthozza onnan, tehát ilyenkor ez lesz a határ-hozzájárulás. Ezt tömören megfogalmazva:

$$v_0'(S) = \begin{cases} \sum\limits_{i\in S} l_i, & \text{ha } S \text{ gyenge,} \\ \sum\limits_{i\in S^c} l_i, & \text{ha } S \text{ erős.} \end{cases}$$

Ha S közepes, akkor a két eset egybeesik és $v_0'(S) = A$ természetesen. Ezt felhasználva a 0 játékos Shapley-értéke az $\mathcal{L}_1$ játékban a nevezőben lévő $n!$-al felszorozva:

$$n! \cdot Sh_0 = \sum_S z_S \cdot v_0'(S) = \sum_{S \text{ erős}} z_S \cdot v_0'(S) + \sum_{S \text{ közepes}} z_S \cdot v_0'(S) + \sum_{S \text{ gyenge}} z_S \cdot v_0'(S) =$$
$$= \sum_{S \text{ erős}} z_S \cdot \left(\sum_{i\in S^c} l_i\right) + \sum_{S \text{ közepes}} z_S \cdot A + \sum_{S \text{ gyenge}} z_S \cdot \left(\sum_{i\in S} l_i\right)$$

Ahol $z_S = |S|! \cdot |S^c|!$ pozitív súlyok. Nézzük, mi változik, ha a Shapley-értéket az $\mathcal{L}_2$ játékban akarjuk kiszámolni. Ebben a kötelezettségek ugyanazok, de 1-el csökkentjük az eszközök A értékét. Egy eredetileg gyenge koalíció továbbra is nullát ér, mert a komplementer követelések kevesebb eszközzel még kevésbé fizethetőek ki. Amikor a 0 játékos a koalícióhoz csatlakozik, továbbra is kifizethető az összes követelés, mert az A-nál szigorúan kevesebb volt, tehát legfeljebb $A-1$. A 0 játékos határ-hozzájárulása tehát, és így az összeg harmadik tagja nem változik. Hasonlóan könnyű meggondolni, hogy ugyanez a helyzet az erős koalíciókkal, tehát az összeg első tagja sem változik. Az összeg második tagja pontosan akkor nemüres, ha az eredeti HALFSUM probléma megoldható. Egy közepes koalíció mindkét játékban nullát ér, mert a komplementer követelések elérik ($\mathcal{L}_2$-ben meghaladják) az eszközök értékét, így nem marad semmi. Amikor a 0 játékos egy ilyen koalícióhoz csatlakozik, annak értéke $\mathcal{L}_1$-ben éppen A lesz, $\mathcal{L}_2$-ben pedig éppen $A-1$. Az összeg második tagjának tehát minden tagja, s így a tejles összeg is csökken. Ha tehát van közepes koalíció, akkor a 0 játékos Shapley-értéke az $\mathcal{L}_2$ játékban kisebb lesz az $\mathcal{L}_1$-hez képest, és pont ezt akartuk belátni. □

3.2. A Shapley-érték becslése Monte-Carlo szimulációval

A Shapley-érték, mint a játékosok átlagos határ-hozzájárulása, egy diszkrét valószínűségi változó várható értéke, s így – ha kiszámolni nem is tudjuk – megbecsülhető Monte-Carlo szimulációval. Az ötletet először maga Shapley vetette fel már 1960-ban (Mann és Shapley, 1960) és több lehetséges módszert is vizsgált a becslési hiba (variancia) csökkentésére a természetes benchmark-nak tekinthető, egyszerű véletlen mintavételezéses eljáráshoz képest. Az értekezés negyedik fejezetében azt a kérdést vizsgáljuk, hogy hogyan lehet a Monte-Carlo szimuláció becslési hibáját csökkenteni, illetve arra felső korlátot adni. Ismertetjük az Illés és Kerényi (2019) által bemutatott algoritmust, mely onnan kapta a nevét, hogy a generált minta, nem független, de továbbra is igaz rá a nagy számok törvénye.

3.2.1. Elméleti háttér

3.4. Definíció. *Legyen $X_1, X_2, \ldots, X_i, \ldots$ azonos eloszlású, véges $\mu = E(X_j)$ várható értékű valószínűségi változók sorozata. Azt mondjuk, hogy a sorozat ergodikus, ha igaz rá a nagy számok*

erős törvénye, azaz

$$\mathbb{P}\left(\lim_{n\to\infty}\frac{\sum\limits_{i=1}^{n}X_i}{n}=\mu\right)=1.$$

Az ergodicitás tehát egy a függetlenségnél gyengébb feltétel, mely biztosítja hogy a mintaátlagot használhassuk a várható érték becslésére. Nem független, de ergodikus sorozatot a következő általános konstrukcióval kaphatunk:

Legyen $(\Omega_1, \mathcal{A}_1, \mathbb{P}_1)$ klasszikus valószínűségi mező, azaz, $|\Omega_1|$ véges halmaz, $\mathcal{A}_1 = 2^{\Omega_1}$ a teljes hatványhalmaz, $\mathbb{P}_1(E) = \frac{|E|}{|\Omega_1|}$ $\forall E \subseteq \Omega_1$-re, és legyen $X : \Omega_1 \to \mathbb{R}$ tetszőleges valószínűségi változó μ várható értékkel. Legyen $(\Omega, \mathcal{A}, \mathbb{P})$ az Ω_1 megszámlálhatóan végtelen hatványa (a mértékterek szorzatának szokásos kontrukciója szerint), és legyen minden $j \in \mathbb{N}$ és $\omega = (\omega_1, \omega_2, \dots) \in \Omega$-re $X_j(\omega) = X(\omega_j)$. Vegyük észre, hogy az X_j-k független, az eredeti X-el azonos eloszlású valószínűségi változók. Legyen továbbá $t : \Omega_1 \leftrightarrow \Omega_1$ kölcsönösen egyértelmű transzformáció Ω_1-en és legyen $T : \Omega \leftrightarrow \Omega$ a t pontonkénti kiterjesztése Ω-ra, azaz $T(\omega) = T(\omega_1, \omega_2, \dots) = (t(\omega_1), t(\omega_2), \dots)$. Végül legyen $K \in \mathbb{N}$ egy pozitív egész paraméter, és tekintsük a következő sorozatot:

- $\forall\ l \geq 1$-re legyen $Y_{1,l} = X_l$
- minden $2 \leq k \leq K$-ra és $l \geq 1$-re legyen $Y_{k,l} = Y_{k-1,l} \circ T = Y_{1,l} \circ \underbrace{T \circ T \cdots \circ T}_{k-1} = Y_{1,l} \circ T^{k-1}$
- Álljon az Y_j sorozat az $Y_{k,l}$ változókból „soronként”,

  $$Y_{1,1}, Y_{2,1}, \dots Y_{K,1}, Y_{1,2}, Y_{2,2}, \dots, Y_{K,2}, Y_{1,3}, Y_{2,3}, \dots$$

  sorrendben felsorolva őket, azaz $Y_{(l-1)K+k} = Y_{k,l}$.

Az Y_j változókra a továbbiakban szimpla és dupla indexekkel ellátva is fogunk hivatkozni, de ezek mindig ugyanazt a sorozatot jelentik.

3.3. Tétel. *A fent definiált Y_j sorozat minden $K \geq 2$ és $t : \Omega_1 \longleftrightarrow \Omega_1$ transzformáció esetén ergodikus.*

3.2.2. Az algoritmus leírása

Noha a 3.3 tétel megfogalmazása igen absztrakt és bonyolultnak tűnik, valójában nagyon egyszerű a tételben szereplő mintát generálni. Legyen $\Omega_1 = \mathcal{O}(N)$ a játékosok N halmazának összes lehetséges sorrendjeiből álló halmaz, míg X valamely $i \in N$ játékos határ-hozzájárulása. Az algoritmus két lépcsőben működik. A 3.3 tétel szerint szükségünk van egy $K \in N$ számra, és egy olyan $t : \Omega_1 \leftrightarrow \Omega_1$ transzformációra, mely kölcsönösen egyértelmű az Ω_1 halmazon, emellett lehetőleg gyorsan kiszámolható. Az algoritmus első lépésben egy ilyen transzformációt keres, majd második lépésben ezt felhasználja a Shapley-érték becslésére. Az egyszerűség kedvéért először ezt a második lépést tárgyaljuk. Ha a 3.3 tételben szereplő t transzformációt már valahogy kiszámoltuk vagy megsejtettük, akkor a következőképpen használhatjuk fel:

- Legyen $m_2 \in N$ pozitív egész paraméter és generáljunk m_2 darab független $\omega \in \Omega_1$ sorrendet.
- Minden, az előző pontban kapott ω sorrendhez számítsuk ki a további $K-1$ darab $t^j(\omega)$: $1 \leq j \leq K-1$ sorrendet (ahol a t^j kitevő a t transzformáció j-szeri egymás utáni alkalmazását jeleni.
- Az összes így kapott Km_2 darab sorrend esetén számítsuk ki az i játékos határ-hozzájárulását, majd ezeket átlagoljuk ki, így kapjuk a Shapley-érték becslését.

Most rátérünk arra a kérdésre, hogy hogyan találhatunk egy megfelelő transzformációt az algoritmushoz. Jegyezzük meg, hogy a 3.3 tétel a t transzformációtól semmit nem követel meg, azon kívül, hogy legyen kölcsönösen egyértelmű, és ebből kifolyólag mértéktartó, azonban, ha a transzformáció nem negatívan korreláló mintákat eredményez, akkor nem lesz hasznos, mert akkor az algoritmus nem csökkeni, hanem növeli a Shapley-érték becslési hibáját. A cél az, hogy az algoritmus valahogy saját maga konstruáljon meg egy működőképes transzformációt. Ez teljes általánosságban reménytelenül nehéznek illetve számításigényesnek tűnik, egyelőre csak a $K = 2$ speciális esetre tudunk ilyen módszert bemutatni.

Jelölje Π_n az $\{1, 2, \ldots, n\}$ halmazon ható szimmetrikus csoportot. Minden $\pi \in \Pi_n$ természetes módon meghatároz egy sorrendeken ható $t_\pi : \mathcal{O}(N) \to \mathcal{O}(N)$ transzformációt. Legyen ugyanis minden $o \in \mathcal{O}(N)$-re és $\pi \in \Pi_n$-re $\forall i \in N$-re $t_\pi(o)(i) = (\pi \circ o)(i) = \pi(o(i))$. Ez teljesen értelmes jelölés, mert a 2.5. Definíció szerint egy sorrend értékkészlete pont az a halmaz, ahol a szimmetrikus csoport hat, és kölcsönösen egyértelmű függvények kompozíciója is kölcsönösen egyértelmű. Ha például egy $\pi \in \Pi_n$-re $\pi(j) = k$ (ahol $1 \leq j, k \leq n$ egész számok), ez azt jelenti, hogy a játékosok minden $o \in \mathcal{O}(N)$ sorrendjéhez egy olyan $t_\pi(o)$ sorrendet rendelünk, ahol az eredetileg j-ediknek értkező játékos most k-adiknak fog érkezni. Az összes $t : \mathcal{O}(N) \to \mathcal{O}(N)$ transzformációk száma $|\mathcal{O}(N)|! = n!!$, míg $|\Pi_n| = n!$, így a legtöbb transzformáció ilyen alakban nem kapható meg. Hogy ezzel mennyit veszítünk, azt nem tudjuk, de az így megkapható transzformációk száma is túl nagy ahhoz, hogy mindet végignézzük, csak olyanok között fogunk keresgélni, melyekre $\pi^2 = \mathrm{id}$ a Π_n csoport egységeleme, azaz melyek előállnak diszjunkt transzpozíciók szorzataként. Ez szemléletesen azt jelenti, hogy párba állítjuk az $\{1, 2, \ldots, n\}$ halmaz elemeit (lehetnek elemek, amelyek saját magukkal állnak párban) és a párokat felcseréljük. Azt, hogy hogyan érdemes a párokat kialakítani, statisztikai alapon döntjük el. Legeneráljuk a játékosok „néhány" lehetséges sorrendjét, minden játékos párt felcserélünk, kiszámoljuk ezen sorrendekhez tartozó határ-hozzájárulásokat, megnézzük, mely két játékos felcserélése esetén fognak ezek minél inkább negatívan korrelálni, majd ezeket a párokat összekombináljuk. Legyen tehát m_1 egy tetszőleges természetes szám $i \in N$ egy tetszőleges (de rögzített) játékos, és tekintsük a következő procedúrát:

- Generáljunk m_1 darab lehetséges sorrendet a játékosok N halmazán, legyenek ezek: $s_1, s_2, \ldots, s_{m_1}$, és jelölje $\mathcal{S}$ az i játékos határ-hozzájárulásaiból álló vektort, ez egy m_1 hosszú adatsor.

- Minden $a, b \in \{1, 2, \ldots, n\}$ pár esetén legyen $s_1^{(a,b)}, s_2^{(a,b)}, \ldots, s_{m_1}^{(a,b)}$ a játékosok azon sorrendje, melyet úgy kapunk, hogy a megfelelő s_j-ben az a-adiknak és b-ediknek érkező játékost felcseréljük. Itt az a-adik és b-edik játékos felcserélése egy (speciális) transzformáció: $s_j^{(a,b)} = \langle a, b \rangle \circ s_j$ ahol $\langle a, b \rangle \in \Pi_n$ az a és b elemek felcserélése.

- Jelölje a $S_{(a,b)}$ az i játékos határ-hozzájárulásaiból álló vektort az $s_1^{(a,b)}, s_2^{(a,b)}, \ldots, s_{m_1}^{(a,b)}$ sorrendek mellett. Ez tehát minden (a, b) pár esetén egy m_1 elemű vektor.

- Legyen minden (a, b)-re $C_{(a,b)}$ az *eredeti* S és az $S_{(a,b)}$ adatsorok kovarianciája.[5]

- A $C_{(a,b)}$ számok az $\{1, 2, \ldots, n\}$ alaphalmazon, mint csúcshalmazon értelmezett teljes gráf éleinek egy súlyozásaként is tekinthetőek. Keressünk mohó algoritmussal ebben a gráfban egy minimális súlyú teljes párosítást, és feleltessük meg egymásnak azokat a pontokat, melyeket a párosítás összeköt. Ez részletesebben kiírva a következőt jelenti: Legyen $E \subseteq \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$ kezdetben az üres halmaz.

  1. Tekintsük az $(a, b) \subseteq \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$ párok $C_{(a,b)}$ szerinti növekvő sorrendbe rendezését.

  2. Adjuk hozzá E-hez a sorozat első elemét.

[5] Ha m_1 elég nagy, akkor számolhatnánk a korrelációs együtthatóval is, de így egyszerűbb, mert biztosan elkerüljük a 0-val való osztást.

3. Húzzuk ki a sorozatból az összes olyan (a, b) elemet, melyre akár a akár b szerepel valamelyik E-beli párban (bármilyen sorrendben).
4. Ha nem marad több elem a sorozatban, kész vagyunk, egyébként ugorjunk a 2.) pontra.

Most legyen π az a permutáció, mely minden $a \in \{1, 2, \ldots, n\}$ számot felcserél azzal az egyértelműen létező b számmal, melyre (a, b) vagy (b, a) E-ben van. Ezzel kész vagyunk.

Megjegyzés. A mohó algoritmus természetesen nem alkalmas a minimális súlyú teljes párosítás megkeresésére, de annak egy elfogadható közelítését adja. Valódi teljes párosítást is kereshetnénk a magyar módszerrel, de annak nagyobb a műveletigénye, és a véges mintából becsült kovarianciákban lévő zaj miatt az algoritmust érdemben valószínűleg nem javítaná.

3.2.3. Komplexitás

Legyen a játék komplexitása (a karakterisztikus függvény kiértékelésének műveletigénye) G_n és vizsgáljuk meg az algoritmus lépésszámát! Ha élünk azzal a nem túl erős megszorítással, hogy $G_n = \Omega(n)$ akkor láthatjuk, hogy itt messze a legnagyobb műveletigényű lépés az $m_1 n^2$ darab határ-hozzájárulás kiszámítása, ami összesen $O\left(m_1 \binom{n}{2} G_n\right)$ művelet. Amikor két játékost, mondjuk az a-adikat és b-ediket felcseréljük, akkor az i játékos határ-hozzájárulása nem változik, ha ő eredetileg nem az a-adik után és a b-edik előtt érkezett (vagy fordítva). Mivel az i játékos egyenletes eloszlással, bármikor érkezhet, a és b pedig végigfut 1-től n-ig, a karakterisztikus függvényt átlagban csak kb. az esetek 1/3-ában kell újra kiértékelni, az esetek kb. 2/3-ában a csere nem változtatja meg azt a koalíciót, amihez az i játékos csatlakozik. Ebből kifolyólag azt mondhatjuk, hogy az algoritmus első lépésének futásideje kb. $\frac{m_1 n^2 G_n}{6}$. A második lépés sokkal egyszerűbb, itt m_2 darab véletlen sorrend generálása történik, ami $O(m_2 n)$ művelet, majd ezek transzformálása, ami ugyanennyi, végül mind a $2m_2$ sorrend mellett a határ-hozzájárulások kiszámítása, az egész összesen $O(m_2 G_n)$ művelet. Egy m elemű független mintával az egyszerű mintavételezés komplexitása nyilvan mG_n. Az algoritmus paramétereit tehát úgy a legigazságosabb beállítani, hogy

$$\frac{m_1 n^2 G_n}{6} + 2m_2 G_n = m_2 G_n \qquad \text{azaz} \qquad \frac{m_1 n^2}{6} + 2m_2 = m$$

teljesüljön, ez esetben az átlagos futásideje egy m elemű független mintás Monte-Carlo szimulációval megegyező (vagy legalábbis, lehető legjobban közelítő).

3.2.4. Eredmények

Az algoritmus tesztelésére a következő nyolc játékot használtuk:

1. Az amerikai elnökválasztás 51 játékosból álló aszimmetrikus szavazási játékát, amelyben a játékosok súlyai:

$$w = (45, 41, 27, 26, 26, 25, 21, 17, 17, 14, 13, 13, 12, 12, 12,$$
$$11, \underbrace{10, \ldots, 10}_{4}, \underbrace{9, \ldots, 9}_{4}, 8, 8, \underbrace{7, \ldots, 7}_{4}, \underbrace{6, \ldots, 6}_{4}, 5, \underbrace{4, \ldots, 4}_{9}, \underbrace{3, \ldots, 3}_{7}),$$

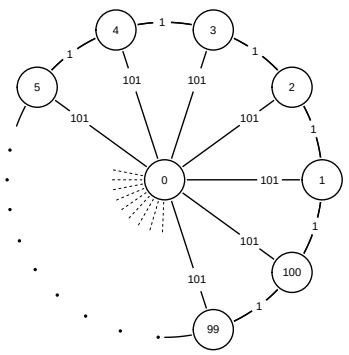
   a kvóta pedig $q = \frac{\sum w_i}{2} + 1 = 270$.

2. Szimmetrikus szavazási játék 100 játékossal (minden játékos súlya azonos, a kvóta 50).

3. Legyen $N_1 = \{1, 2, \ldots, 50\}$, $N_2 = \{51, 52, \ldots, 100\}$, $N = N_1 \cup N_2$, és legyen minden $S \subseteq N$-re $v(S) = \min(|S \cap N1|, |S \cap N_2|)$. Ez a játék a szakirodalomban eredetileg *shoes game*, magyarul *kesztyűjáték* néven ismert, minden koalíció értéke az egymást kiegészítő tárgyakból képezhető párok száma.

4. Repülőtér-játék. Legyen $N = \{1, 2, \ldots, 100\}$ és $c = (\underbrace{1, \ldots, 1}_{8}, \underbrace{2, \ldots, 2}_{12}, \underbrace{3, \ldots, 3}_{6}, \underbrace{4, \ldots, 4}_{14}, \underbrace{5, \ldots, 5}_{8}, \underbrace{6, \ldots, 6}_{9}, \underbrace{7, \ldots, 7}_{13}, \underbrace{8, \ldots, 8}_{10}, \underbrace{9, \ldots, 9}_{10}, \underbrace{10, \ldots, 10}_{10})$, és legyen a karakterisztikus függvény $v(S) = \max\{c_i | i \in S\}$. A c vektor azt írja le, mekkora költség az egyes repülőgépekhez kifutópályát építeni. Egy repülőgép tehát csak akkor növeli egy koalíció költségét, ha nagyobb, mint a koalícióban a legnagyobb gép, és csak miatta kell nagyobbra építeni a kifutópályát.

5. Legyen a játékosok halmaza $N = \{0, 1, 2, \ldots, 100\}$ és legyen tetszőleges S koalíció értéke a 1. ábrán lévő gráf S-beli csúcsai és a 0 csúcs által feszített részgráf minimális költségű feszítőfájának költsége.



1. ábra. Minimális költségű feszítőfa (MCST) játék gráfja

6. A 2.1 példa csődjátéka $n = 100$ játékosra. Az eszközök értéke 200, a követelések megegyeznek repülőtér-játékban lévő költségekkel.

7. A 2.1 példában bemutatott tartozásos játék, a paraméterei megegyeznek a csődjáték paramétereivel.

8. Legyen $N_1 = \{1, 2, \ldots, 50\}$, $N_2 = \{51, 52, \ldots, 100\}$, $N = N_1 \cup N_2$, és tegyük fel, hogy minden játékos az N_1 halmazban pontosan egy N_2-belivel van párba állítva. Legyen minden S koalíció $v(S)$ értéke az S-beli elemekből kirakható párok száma. Ez tehát hasonlít a *shoes game* játékra, de itt nem képezhető pár bármely jobb és bármely bal lábas cipőből hanem mindenkinek csak egy párja van, például képzelhetjük úgy, hogy minden pár cipő különböző színű.

A numerikus szimuláció eredményeit a 2. táblázat tartalmazza. Mindegyik játékra öt különböző paraméterbeállítással futtattunk egy 1000 elemű szimulációt, hogy megbecsüljük az algoritmus által becsült Shapley-érték szórását. Ezt tartalmazza a σ_E oszlop. Összehasonlításként feltüntettük az egyszerű mintavétel szórását, és az utolsó oszlopban a kettő hányadosát. A 4. és 6. játék esetében a legnagyobb súlyú utolsó játékos Shapley-értékét számoltuk, a többinél az elsőét. Az eredményeket röviden három pontban foglalhatjuk össze.

- Ha az algoritmust túl kis elemszámú mintával futtatjuk, akkor a számítási kapacitás túl nagy részét köti le a megfelelő t transzformáció megtalálása, így rosszabb eredményt produkál, mint az egyszerű mintavételezés. A táblázat ezen soraiban 1-nél nagyobb szám van az utolsó oszlopban.

Játék	m	m_1	m_2	σ_R	σ_E	σ_E/σ_R
	100 000	100	28 325	0,000897	0,001201	1,34
	600 000	100	278 325	0,000366	0,000362	0,99
1.	4000 000	100	1 978 325	0,000142	0,000136	0,96
	600 000	1000	83 250	0,000366	0,000714	1,95
	4000 000	1000	1 783 250	0,000142	0,000146	1,03
	1000 000	500	83 333	0,000099	0,000237	2,39
	2000 000	50	958 334	0,000070	0,000100	1,43
2.	10 000 000	50	4 958 334	0,000031	0,000030	0,98
	2000 000	1000	166 667	0,000070	0,000170	2,43
	10 000 000	1000	4 166 667	0,000031	0,000035	1,12
	400 000	100	116 667	0,000791	0,000848	1,07
	3000 000	100	1 416 667	0,000289	0,000229	0,79
3.	10 000 000	100	4 916 667	0,000158	0,000127	0,81
	3000 000	1500	250 000	0,000289	0,000415	1,43
	10 000 000	1500	3 750 000	0,000158	0,000101	0,64
	200 000	50	58 333	0,002825	0,003602	1,28
	2000 000	50	958 333	0,000893	0,000911	1,02
4.	7000 000	50	3 458 333	0,000478	0,000465	0,97
	2000 000	500	583 333	0,000893	0,001159	1,30
	7000 000	500	3 083 333	0,000478	0,000498	1,04
	250 000	100	41 667	0,162058	0,204571	1,26
	1700 000	100	766 667	0,062146	0,052712	0,85
5.	4000 000	100	1 916 667	0,040514	0,033662	0,83
	1700 000	1000	16 667	0,062146	0,198159	3,19
	4000 000	1000	1 166 667	0,040514	0,023188	0,57
	200 000	100	16 667	0,010639	0,018708	1,76
	1000 000	100	416 667	0,004758	0,003526	0,74
6.	10 000 000	100	4 916 667	0,001505	0,001047	0,70
	1000 000	500	83 333	0,004758	0,007708	1,62
	10 000 000	500	4 583 333	0,001505	0,001051	0,70
	200 000	100	166 667	0,180138	0,244463	1,36
	1000 000	100	416 667	0,080560	0,056791	0,70
7.	10 000 000	100	4 916 667	0,025475	0,015614	0,61
	1000 000	500	83 333	0,080560	0,045475	0,56
	10 000 000	500	4 583 334	0,025475	0,006730	0,26
	250 000	100	41 667	0,001000	0,001520	1,52
	2000 000	100	916 667	0,000354	0,000278	0,79
8.	5000 000	100	2 416 667	0,000224	0,000198	0,88
	2000 000	1000	166 667	0,000351	0,000442	1,26
	5000 000	1000	1 666 667	0,000224	0,000150	0,67

2. táblázat. Ergodikus és független minta összehasonlítása 1000 elemű szimulácó alapján

- Ha az algoritmus „elég nagy” elemszámú mintát generál, akkor a nyolc esetből háromnál (1., 2. és 4. játék) nem képes érdemi javulást elérni, és nagyon hasonló eredményt produkál, mint az egyszerű mintavételezés. A táblázat ezen sorainak utolsó oszlopában 1-hez nagyon közeli számok láthatóak.
- A nyolc esetből 5-nél az algoritmus szignifikánsan csökkenti a becslés hibáját az egyszerű mintavételezéshez képest, a legdrasztikusabban a 7. játékban (a táblázat 35. sora).

3.3. Egzakt, pszeudopolinomiális algoritmusok

Egy algoritmust akkor nevezünk pszeudopolinomiálisnak, ha a futási ideje polinomiális függvénye az input numerikus értékének (nem feltétlenül az input méretének). A fogalomnak csak olyan problémák esetén van értelme, ahol az input számokból vagy számsorozatokból áll. A TU-játékok tipikusan ilyenek, ezért van értelme a Shapley-érték kiszámítására olyan algoritmust keresni, mely ugyan nem polinomiális az input méretének függvényében, de a futási ideje,

a brute force módszerrel ellentétben, a játékosok számával nem növekszik exponenciálisan. Az értekezés utolsó fejezetében egy ilyen eredményt mutatunk be.

3.3.1. Multilineáris kiterjesztés

Jelölje pozitív $a, b \in \mathbb{N}$ egészekre $\beta(a, b)$ az Euler-féle béta függvényt:

$$\beta(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}\ dx = \frac{(a-1)! \cdot (b-1)!}{(a+b-1)!}.$$

A Shapley-érték (2) képlete szerint:

$$Sh_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (n - |S| - 1)!}{n!} \cdot v_i'(S)$$

ami $a = |S| + 1$, $b = n - |S|$ jelöléssel:

$$Sh_i = \sum_{S \subseteq N \setminus \{i\}} \frac{(a-1)! \cdot (b-1)!}{(a+b-1)!} \cdot v_i'(S) = \sum_{S \subseteq N \setminus \{i\}} \int_0^1 x^{a-1}(1-x)^{b-1} \cdot v_i'(S)\ dx =$$

$$\sum_{S \subseteq N \setminus \{i\}} \int_0^1 x^{|S|}(1-x)^{n-|S|-1} \cdot v_i'(S)\ dx = \int_0^1 \sum_{S \subseteq N \setminus \{i\}} x^{|S|}(1-x)^{n-|S|-1} \cdot v_i'(S)\ dx.$$

Az $x^{|S|}(1-x)^{n-|S|-1}$ kifejezés annak a valószínűsége, hogy ha minden $N \setminus \{i\}$ -beli játékos egymástól függetlenül x valószínűséggel dönt a kooperáció mellett, akkor a kooperáló játékosok halmaza éppen S. Mivel ezt minden lehetséges $S \subseteq N \setminus \{i\}$ koalícióra kiszámoljuk és összeadjuk, az integrál mögötti súlyozott összeg is egy várható értékként értelmezhető. Emellett egy függvény integrálja 0-tól 1-ig (1-el osztva) is a függvény átlagértékének tekinthető. Ezzel a Shapley-érték egy új interpretációját nyerjük, mint a határ-hozzájárulás várható értékét.

$$Sh(i) = \int_0^1 \mathbb{E}_x v_i'(S)\ dx \tag{3}$$

ahol

$$\mathbb{E}_x v_i'(S)\ dx = \sum_{S \subseteq N \setminus \{i\}} x^{|S|}(1-x)^{n-|S|-1} \cdot v_i'(S).$$

3.3.2. Lineárisan reprezentálható játékok

A Shapley-érték pszeudopolinomiális algoritmussal az alábbi esetben számolható ki:

3.5. Definíció. *Egy (N, v) TU-játékra azt mondjuk, hogy lineárisan reprezentálható, ha a játékosokhoz rendelhetőek $a_j : j \in N$ súlyok és létezik $f : \mathbb{N} \to \mathbb{N}$ függvény, hogy minden $S \subseteq N$ koalíció értéke előáll $v(S) = f\left(\sum_{i \in S} a_i\right)$ alakban. Ezesetben a fenti f függvényt és az a_j számokat a játék egy lineáris reprezentációjának nevezzük. Ha egy játéknak adott egy lineáris reprezentációja, akkor azt mondjuk, hogy a játék ez által lineárisan reprezentált.*

Megjegyzés. A lineárisan reprezentálhatóság egy elég általános fogalom, a szavazási játékok, a csődjátékok a tartozásos játékok, a repülőtér játékok mind egy lineáris reprezentációval vannak adva, a költségallokációs játékok azonban nem.

Az értekezés egyik fő eredménye az alábbi

3.4. Tétel. *Legyen a $G = (N, v)$ játék egy lineáris reprezentációja $\big(f, \{a_j : j \in N\}\big)$. Tegyük fel, hogy minden $a_j \leq 2^j$, és az f függvény lineáris időben és tárban kiszámolható. Ekkor tetszőleges $i \in N$ játékos Shapley-értéke a G játékban kiszámítható $O(n^3K)$ időben és $O(n^2K)$ tárban, ahol $K = \sum\limits_{j \neq i} a_j$.*

Bizonyítás. Az algoritmus a Shapley-értéket az

$$Sh(i) = \int\limits_0^1 \mathbb{E}_x v_i'$$

(3) egyenlet felhasználásával fogja kiszámolni. Legyen $S \subseteq N$ egy tetszőleges, az i játékost nem tartalmazó koalíció. A lineáris reprezentáció szerint $v(S) = f\left(\sum\limits_{j \in S} a_j\right)$, amiből:

$$v_i'(S) = f\left(\sum_{j \in S} a_j + a_i\right) - f\left(\sum_{j \in S} a_j\right)$$

Legyen most $\mathcal{S}_x \subseteq N \setminus \{i\}$ az a véletlen koalíció, melyet az i nélküli $n-1$ játékos alkot, midőn egymástól függetlenül x valószínűséggel a kooperációról döntenek. Ezzel a jelöléssel

$$\mathbb{E}_x v_i' = \sum_{S \subseteq N \setminus \{i\}} \mathbb{P}\left(\mathcal{S}_x = S\right) \cdot \left(f\left(\sum_{j \in S} a_j + a_i\right) - f\left(\sum_{j \in S} a_j\right)\right) =$$

Legyen $K = \sum\limits_{j \in N \setminus \{i\}} a_j$ és csoportosítsuk az összeg tagjait az S-beli súlyok összege szerint:

$$= \sum_{k=0}^{K} \sum_{\substack{S \subseteq N \setminus \{i\} \\ \sum\limits_{j \in S} a_j = k}} \mathbb{P}\left(\mathcal{S}_x = S\right) \cdot \Big(f\left(k + a_i\right) - f\left(k\right)\Big) =$$

$$= \sum_{k=0}^{K} \Big(f\left(k + a_i\right) - f\left(k\right)\Big) \cdot \sum_{\substack{S \subseteq N \setminus \{i\} \\ \sum\limits_{j \in S} a_j = k}} \mathbb{P}\left(\mathcal{S}_x = S\right) = \sum_{k=0}^{K} \Big(f\left(k + a_i\right) - f\left(k\right)\Big) \cdot \mathbb{P}(\mathcal{G}_x = k),$$

ahol $\mathcal{G}_x$ jelöli a $\sum\limits_{j \in \mathcal{S}_x} a_j$ valószínűségi változót, azaz, a játékosok által kialakított véletlen koalíció összsúlyát, midőn egymástól függetlenül x valószínűséggel a kooperációról döntenek. A $\mathbb{P}(\mathcal{G}_x = k)$ valószínűségek az x változó egész együtthatós $(n-1)$-ed fokú polinomjai, melyeket rekurzíve ki tudunk számolni, a következőképpen:

- Tekintsük az a_i-n kívüli súlyok (vagyis az i-től különböző játékosok) egy tetszőleges, de előre rögzített sorrendjét, és legyen $P[j, k]$ annak a valószínűsége, hogy az első j játékos egy k összsúlyú koalíciót alkot ($0 \leq j \leq n-1$ és $0 \leq k \leq K$). Ez minden j-re és k-ra az x változó egy j-edfokú polinomja. Nyilván $j = 0$-ra $P[0, 0] = 1$ és $P[0, k] = 0$ $k \geq 1$-re.

- Minden $1 \leq j \leq n-1$-re, ha $P[j-1, k]$ már ismert minden $0 \leq k \leq K$-ra, akkor a teljes várható érték tétele szerint:

$$P[j, k] = \begin{cases} (1-x) \cdot P[j-1, k] & \text{ha } k < a_j \\ x \cdot P[j-1, k-a_j] + (1-x) \cdot P[j-1, k] & \text{ha } k \geq a_j \end{cases}$$

A $P[n-1,k]$ számok épp a $\mathbb{P}(\mathcal{G}_x = k)$ valószínűségek. Ezek egyszerű súlyozott összegeként kapjuk a $\mathbb{E}_x v_i' = \sum_{k=0}^{K} \Big(f(k+a_i) - f(k) \Big) \cdot \mathbb{P}(\mathcal{G}_x = k)$ kifejezést, mely szintén egy $n-1$-ed fokú polinom, így az integrálját is ki tudjuk számolni analítikusan: egyszerűen kiintegráljuk tagonként (a konstans tagot nullának tekintve), majd a primitív függvény (egy n-ed fokú polinom) együtthatóit összeadjuk. Ez egy racionális szám, a játék pontos Shapley-értéke.

Ami az algoritmus komplexitását illeti, egy $n \times K$-as mátrixot kell kitölteni, melynek minden eleme egy legfeljebb n-ed fokú polinom, azaz praktikusan n darab egész szám, melyekre tudunk exponenciális felső becslést adni. Az első sorban csak olyan (konstans) polinomok vannak, melyek mindegyik együtthatója 0 vagy 1. Minden sor legfeljebb három, előző sorban lévő polinom összege (nem csak kettő, hanem három, mert az $(1-x)$-es tag eleve két polinom összege, és ehhez jön még az x-es tag). így j szerinti indukcióval $P[j,k]$ mindegyik együtthatója legfeljebb 3^j ami legfeljebb 3^n, ami egy legfeljebb $O(n)$ bites szám. Legfeljebb n darab legfeljebb $O(n)$ bites számot tárolni legfeljebb $O(n^2)$ tárhely, és két ilyet összeadni legfeljebb $O(n^2)$ művelet. A $P[j,k]$ mátrix tehát kiszámolható $O(n^3K)$ futási idővel. Az algoritmus során szükségtelen a teljes mátrixot tárolni, elegendő mindig a két aktuális sor, ami csak $O(n^2K)$ memóriát igényel. Könnyen látható, hogy az f-re tett egyszerűsítő feltevések mellett, a $P[n-1,k]$ polinomok előállítása a leginkább számításigényes művelet, az algoritmus többi lépése (az f függvény kiértékelése K pontban, a polinomok összesúlyozása, és integrálása) elvégezhető $O(n^2K)$ lépésben, a teljes algoritmus futási ideje tehát $O(n^3K)$. □

Az algoritmust n-szer függetlenül lefuttatva az összes játékosra nyilván ki tudjuk számolni mindegyik Shapley-értéket $O(n^4K)$ időben, azonban erre nincsen szükség, ugyanis a leginkább számításigényes részt elég egyszer megcsinálni.

3.1. Lemma. *Legyen $j_1, j_2, \ldots, j_{n-1}$ az i-től különböző játékosok egy tetszőleges sorrendje, és tekintsük a 3.4. tétel szerinti $P[j,k]$ mátrixot. Ekkor:*

a) Ha az ℓ-edik és $\ell+1$-edik játékost felcseréljük, akkor a $P[j,k]$ mátrix ℓ-edik során kívül egyis sem változik.

- ** Speciálisan: a mátrix j-edik sora nem függ az első j játékos (sem az utolsó $n-1-j$ játékos) egymás közötti sorrendjétől.*
- *** Még speciálisabban: a mátrix utolsó sora független a játékosok sorrendjétől.*

b) A játékosok előre rögzített sorrendjének ismeretében a mátrix tetszőleges sorából megkapható az előző sor.

A rekurzió megfordításához vegyük észre, hogy a $P[j,k]$ mátrix első oszlopa mindig $(1-x)^j$, sőt általában, ha $k < a_j$ akkor $P[j,k] = P[j-1,k] \cdot (1-x)$, így a j-edik sor első $a_j - 1$ eleme mindig osztható $(1-x)$-el (maradék nélkül) és $P[j-1,k] = P[j,k]/(1-x)$. Innentől balról jobbra haladva a $(j-1)$-edik sor többi eleme is kiszámolható, hiszen ha $k \geq a_j$ és minden $l < k$-ra $P[j-1,l]$ már ismert, akkor

$$P[j,k] = x \cdot P[j-1,k-a_j] + (1-x) \cdot P[j-1,k]$$

miatt $P[j,k] - x \cdot P[j-1,k-a_j]$ osztható $(1-x)$-el, és

$$P[j-1,k] = \frac{P[j,k] - x \cdot P[j-1,k-a_j]}{1-x}$$

Ebből már látszik, hogyan használható a módszer az teljes játék Shapley-értékének kiszámítására. A számításigényes művelet a $P[j,k]$ mátrix utolsó sorának kiszámolása, amelyet elég egyszer megcsinálni.

3.1. Következmény. *Legyen a $G = (N, v)$ játék egy lineáris reprezentációja $\big(f, \{a_j : j \in N\}\big)$, ahol $a_j \leq 2^j$, és az f függvény lineáris időben és tárban kiszámolható, azaz teljesülnek a 3.4. tétel feltételei. Ekkor a játék Shapley-értéke kiszámítható $O(n^3K)$ időben és $O(n^2K)$ tárban, ahol $K = \sum a_j$.*

Bizonyítás. Az 3.4. tétel bizonyításában bemutatott algoritmust kell kissé módosítani. Ahelyett, hogy először kiválasztanánk egy i játékost, akinek a Shapley-értékét tudni szeretnénk, először kiszámoljuk a $P[j, k]$ mátrixot az összes játékosra, azaz az összes a_j súly figyelembevételével. A mátrix így $(n + 1) \times K$-as lesz és nyilván még mindig $O(n^3K)$ időben előállítható. Legyen most $i \in N$ egy tetszőleges játékos. A 3.1. lemma szerint a mátrix utolsó sorában előálló $P[n, k]$ polinomok (nevezzük ezeket a továbbiakban az egyszerűség kedvéért alappolinomoknak) nem függnek attól, hogy az a_j súlyok milyen sorrendjéből állítottuk elő a mátrixot továbbá a mátrix utolsó előtti sora

$$P_i[n-1,k] = \begin{cases} \frac{P[n,k]}{1-x}, & \text{ha } k < a_i, \\ \frac{P[n,k]-x\cdot P[n-1,k-a_i]}{1-x}, & \text{ha } k \geq a_i \end{cases}$$

lett volna, ha épp az i játékos lett volna a sorrendben az utolsó. (A képletben feltüntettük az eddig elhanyagolt i-től való függést is). A polinomosztás (és nyilván az összes többi művelet is) elvégezhető $\mathbb{Z}[x]$-ben, $O(n^2)$ időben, azaz a mátrix utolsó sorából az utolsó előtti $O(n^2K)$ lépésben előállítható. A $P_i[n, k]$ polinomok, ugyanazok a polinomok, melyek a 3.4. tétel szerint az i játékos Shapley-értékének kiszámítására felhasználhatóak (és a számítás további lépései sem haladják meg az $O(n^2K)$ komplexitást). Mivel ezek a polinomok egymástól függetlenül is előállíthatóak $O(n^2K)$ időben a teljes mátrix nulláról való felépítése nélkül, a Shapley-értékek egyesével kiszámolhatóak $O(n^2K + n^3K) = O(n^3K)$ időben és $O(n^2K)$ tárban. □

3.3.3. Repülőtér-játékok

A repülőtér-játékok speciális költségallokációs játékok (a gráf egyetlen útból áll), így Shapley-értékük lineáris időben kiszámítható. Ezek a 3.5. definíció értelmében nem lineáris reprezentációval vannak adva, mert a koalíciók értéke nem a súlyok összegének, hanem azok maximumának függvénye, azonban a $P[j, k]$ polinomok ez esetben is felírhatóak rekurzívan: a súlyokat nemcsökkenő sorrendben tekintve, minden sor úgy kapható a felette lévőből, hogy mindenkit megszorzunk $(1 - x)$-el, az adott játékos súlyával megegyező indexű oszlopban pedig még hozzáadunk x-et. Mivel $P[j, k] = 0$ minden j-re, minden olyan k-ra, amely nem egyezik meg egyik játékos költségével sem, a mátrix minden oszlopa, legfeljebb m oszlop kivételével azonosan nulla lesz, ahol m a különböző súlyok száma. Ezen módosítással az algoritmus $O(n^5)$ időben és $O(n^3)$ tárban fut, mert a mátrix K helyett csak $m \leq n$ oszlopból áll, azonban minden sor kiszámítása során (a maximum függvény miatt) minden nemnulla elemmel és nem csak kettővel kell számolni.

Az ötödfokú algoritmus önmagában nem lenne túl hasznos felfedezés, de mindjárt más a helyzet, ha észrevesszük, hogy a repülőtér-játékok esetén a mátrixot nem szükséges rekurzíve kiszámolni, mert kapásból fel tudjuk írni az utolsó sorát:

$$P[n-1,c_k] = \left(1-(1-x)^{\ell_k}\right)(1-x)^{\sum\limits_{c_h>c_k}\ell_h} = (1-x)^{\sum\limits_{c_h>c_k}\ell_h} - (1-x)^{\sum\limits_{c_h\geq c_k}\ell_h}$$

ahol ℓ_i a c_i súlyú játékosok száma. Ezeket a polinomokat nem szükséges monomok összegére kibontani, mert csak az integráljukra van szükségünk 0 és 1 között, ami nem változik, ha a

függvényt tükrözzük az $x = 1/2$ egyenesre, azaz x-et $1 - x$-el helyettesítjük. Ebből az

$$\int_0^1 P[n-1, c_k] = \int_0^1 (1-x)^{\sum\limits_{c_h > c_k} \ell_h} - (1-x)^{\sum\limits_{c_h \geq c_k} \ell_h} =$$

$$= \int_0^1 x^{\sum\limits_{c_h > c_k} \ell_h} - \int_0^1 (1-x)^{\sum\limits_{c_h \geq c_k} \ell_h} = \frac{1}{\sum\limits_{c_h > c_k} \ell_h + 1} - \frac{1}{\sum\limits_{c_h \geq c_k} \ell_h + 1}$$

illetve az

$$\int_0^1 P[n-1, 0] = \int_0^1 (1-x)^n = \frac{1}{n+1}$$

formulák adódnak. A kimaradó i játékos Shapley-értékének számolásakor a $c_k < c_i$ tagokat kell súlyozni $c_i - c_k$ hátárhozzájárulásokkal, ami egy könnyen láthatóan lineáris időben kiszámolható teleszkópikus összeg. Ezzel kaptunk egy új bizonyítást (és egy új algoritmust) a repülőtér-játékok Shapley-értékének komplexitására (és kiszámítására) a költségallokációs játékok felhasználása nélkül.

4. Az értekezés főbb eredményeinek összefoglalása

- Bevezettük a polinomiálisan reprezentálható játékok fogalmát, melyek keretében a Shapley-érték komplexitása értelmes számítástudományi kérdés.
- Megmutattuk, hogy a nemrég bevezetett tartozásos játékok (liability games - Csóka és Herings (2019)) osztályára a Shapley-érték kiszámítása polinom-időben nem megoldható (hacsak nem P = NP).
- Bemutattuk a Shapley-érték Monte-Carlo szimulációval történő becslésének egy új módszerét.
- Bevezettük a lineárisan reprezentálható játékok fogalmát és megmutattuk, hogy ezekre a Shapley-érték kiszámítása pszeudopolinomiális. Ez közös általánosítása több korábbi eredménynek és lefed korábban nem vizsgált eseteket, mert mind a szavazási játékok, mint a csődjátékok, mind a tartozásos játékok lineárisan reprezentálhatóak.
- Megmutattuk, hogy az algoritmus a nem lineárisan reprezentált repülőtér-játékokra is működik, sőt nagymértékű egyszerűsödés után lineáris időben fut, ami egy ismert eredmény új bizonyítása.

A fenti eredmények nagy részét eddig egy cikk és két műhelytanulmány formájában sikerült publikálni:

- Csóka, Péter, Ferenc Illés, and Tamás Solymosi. "On the Shapley value of liability games." *European Journal of Operational Research* 300.1 (2022): 378-386.
- Illés, Ferenc, and Péter Kerényi. "Estimation of the Shapley value by ergodic sampling." *arXiv preprint arXiv:1906.05224* (2019).
- Illés, Ferenc. "Linearly representable games and pseudo-polynomial calculation of the Shapley value." *arXiv preprint arXiv:2205.12324* (2022).

Hivatkozások

Colini-Baldeschi, R., Scarsini, M., és Vaccari, S. Variance allocation and Shapley value. *Methodology and Computing in Applied Probability*, 20(3):919–933, 2018.

Csóka, P. Az adósságelengedés modellezése kooperatív játékelmélettel. *Közgazdasági Szemle*, 65(7-8):768–779, 2018.

Csóka, P. és Herings, P. J.-J. Liability games. *Games and Economic Behavior*, 116:260–268, 2019. URL `https://doi.org/10.1016/j.geb.2019.05.007`.

Csóka, P. és Pintér, M. On the impossibility of fair risk allocation. *The BE Journal of Theoretical Economics*, 16(1):143–158, 2016.

Csóka, P., Herings, P. J.-J., és Kóczy, L. Á. Stable allocations of risk. *Games and Economic Behavior*, 67(1):266–276, 2009.

Csóka, P., Illés, F., és Solymosi, T. On the shapley value of liability games. *European Journal of Operational Research*, 300(1):378–386, July 2022. doi: 10.1016/j.ejor.2021.10.012. URL `https://doi.org/10.1016/j.ejor.2021.10.012`.

Deng, X. és Papadimitriou, C. H. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994. URL `http://www.jstor.org/stable/3690220`.

Illés, F. Linearly representable games and pseudo-polynomial calculation of the shapley value. *arXiv preprint arXiv:2205.12324*, 2022.

Illés, F. és Kerényi, P. Estimation of the Shapley value by ergodic sampling. *arXiv preprint arXiv:1906.05224*, 2019.

Mann, I. és Shapley, L. S. Values of large games, iv: Evaluating the electoral college by montecarlo techniques. 1960.

Mann, I. és Shapley, L. S. Values of large games, vi: Evaluating the electoral college exactly. 1962.

Megiddo, N. Computational complexity of the game theory approach to cost allocation for a tree. *Mathematics of Operations Research*, 3(3):189–196, 1978. doi: 10.1287/moor.3.3.189. URL `https://doi.org/10.1287/moor.3.3.189`.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 2(28): 307–317, 1953.

Taylor, A. D. és Pacelli, A. M. *Mathematics and politics: strategy, voting, power, and proof.* Springer Science & Business Media, 2008.

von Neumann, J. és Morgenstern, O. *Theory of games and economic behavior.* Princeton University Press, 1944.