Rangsorolás csoportos összehasonlítással

Készítette: Hámori Ádám
Nappali tagozat
Programtervező matematikus szak

Témavezető: Dr. Iványi Antal Miklós
Egyetemi tanár

Budapest, 2011. március

Tartalomjegyzék

1. Bevezetés **3**

1.1. A problémakör leírása 3

1.2. Algoritmusokkal kapcsolatos definíciók 4

1.3. Versenyekkel kapcsolatos alapfogalmak 5

1.4. Előzmények 6

2. Korábbi eredmények versenyekkel kapcsolatban **9**

2.1. Teljes 1-versenyek 9

2.2. Teljes a-versenyek 10

2.3. Teljes (a, b)-versenyek 11

2.3.1. Az ellenőrző algoritmus 14

2.3.2. Az algoritmus műveletigénye 15

2.3.3. A probléma megfogalmazása másképpen 15

2.3.4. Módszer f és g meghatározására 17

2.4. Versenyek helyreállítása 21

2.4.1. A helyreállítás alapötlete 21

2.4.2. A helyreállító algoritmus 24

2.4.3. A helyreállító algoritmus magyarázata 26

3. Hiperversenyek pontsorozatai **28**

3.1. Hiperversenyekről általánosságban 28

3.2. Teljes k-hiperversenyek 29

1. Bevezetés

1.1. A problémakör leírása

Gyakran előforduló gyakorlati feladat (iparban, biológiában, sportban, számítógépeknél), hogy objektumokat kell rangsorolni. Egy népszerű módszer az objektumok páronkénti összehasonlítása, és az összehasonlítások eredményének pontokkal való kifejezése. Végül pedig az objektumoknak az elért összpontszám szerinti rangsorolása.

Ábrázolási lehetőségként természetes módon adódnak a mátrixok: n objektum esetén egy $n \times n$-es mátrix, ahol az i. sor j. eleme az i. és j. objektumok összehasonlításának az eredménye. Egy másik lehetséges ábrázolási mód egy irányított gráf: a csúcsok jelentik az összehasonlítandó objektumokat, az élek pedig az összehasonlítások eredményeit.

A vizsgált feladat nagyon jól megfogalmazható a sport nyelvén – épp ezért elterjedt a sportbéli kifejezések hasznalata a témakörben – hiszen a páronkénti összehasonlítások halmaza nem más, mint egy körmérkőzéses bajnokság (verseny) mérkőzései, ahol az objektumok a csapatok (játékosok), a rangsorolás pedig az elért összpontszám alapján történik. Így a továbbiakban gyakran sportbéli kifejezéseket fogok használni.

Egy körmérkőzéses versenyen n csapat összesen $\frac{n(n-1)}{2}$ mérkőzést játszik. Ha egy mérkőzésen k eredmény várható, akkor lehetséges végeredmények száma: $k^{\frac{n(n-1)}{2}}$. A versenyek számos informatikai feladat forrásául szolgálnak. Ilyen feladat például pontvektorok előállítása, ellenőrzése, számuk meghatározása, eredménymátrixok helyreállítása, optimalizálási problémák, feladatok bonyolultságának meghatározása.

Dolgozatomban elsősorban pontsorozatok ellenőrzésével, illetve adott pontsorozatból egy lehetséges eredménymátrix helyreállításásával fogok foglalkozni, az egyszerűbb, speciális esetektől elindulva az egyre általánosabbak felé.

Fontosnak tartom megjegyezni, hogy a pontok mindenhol nemnegatív egész számok, a csapatok száma pedig adott, így a problémakör diszkrét. Ez azt jeleti, hogy minden kérdés megválaszolható véges időben például brute-force algoritmussal, vagy visszalépéses kereséssel. Ezekkel az algoritmusokkal bármilyen pontsorozatnak eldönthető a helyessége, illetve az összes lehetséges eredménymátrix megkapható, de ezen algoritmusok exponenciális futási idővel rendelkeznek. Így számunkra azok az algoritmusok maradnak érdekesek csak, melyek ennél lényegesen kisebb (lineáris vagy alacsony kitevőjű polinomiális) futási idővel rendelkeznek.

1.2. Algoritmusokkal kapcsolatos definíciók

Az algoritmusok elemzése a végrehajtáshoz szükséges erőforrások (ez számítógépen futó algoritmus esetén processzoridő, memóriakapacítás) meghatározását is jelenti. Gyakran nem tudjuk, vagy nem akarjuk az igényelt erőforrás mennyiségét pontosan megadni, ilyenkor megelégszünk az igény nagyságrendjének jellemzésével.

Ennek a jellemzésnek jól bevált eszközei az $\Omega, O, \Theta, \omega, o$ jelölések. Mivel az igények általában nemnegatívak, ezért az alábbi meghatározásokban mindenütt feltesszük, hogy az f és g függvények a pozitív egészek halmazán vannak értelmezve, az $f(n)$ és $g(n)$ függvényértékek pedig nemnegatív valós számok. Az O jelöléssel aszimptotikus felső, az Ω jelöléssel pedig aszimptotikus alsó korlátot tudunk adni, míg a Θ jelöléssel pontosan meg tudjuk adni a vizsgált függvény aszimptotikus viselkedéset.

- $O(g(n))$ azon $f(n)$ függvények halmazát jelenti, amelyekhez léteznek olyan $c \in \mathbb{R}^+$ és $n_0 \in \mathbb{Z}^+$ állandók, hogy minden $n \geq n_0$-ra:

$$f(n) \leq cg(n).$$

- $\Omega(g(n))$ azon $f(n)$ függvények halmazát jelenti, amelyekhez léteznek olyan $c \in \mathbb{R}^+$ és $n_0 \in \mathbb{Z}^+$ állandók, hogy minden $n \geq n_0$-ra:

$$f(n) \geq cg(n).$$

- $\Theta(g(n))$ azon $f(n)$ függvények halmazát jelenti, amelyekhez léteznek olyan $c_1, c_2 \in \mathbb{R}^+$ és $n_0 \in \mathbb{Z}^+$ állandók, hogy minden $n \geq n_0$-ra:

$$c_1 g(n) \leq f(n) \leq c_2 g(n).$$

- *Elemi művelet*: olyan művelet, amely futási ideje konstans. Pl: összeadás, kivonás, szorzás, osztás, értékadás, feltétel vizsgálat, konstans futási idejű függvény vagy eljárás meghívása.

- *Algoritmus futási ideje*: egy soros algoritmus futási ideje egy adott bemenetre a végrehajtott elemi műveletek száma. Valójában a különböző műveletek végrehajtásához nem ugyanaz az idő szükséges, de feltehetjük, hogy a végrehajtási idők megegyeznek, mert nincsenek lényeges különbségek.

- *Algoritmus legrosszabb futási ideje*: n bemenő adat esetén az összes lehetséges futási idő közül a legrosszabb.

1.3. Versenyekkel kapcsolatos alapfogalmak

- *Verseny*: olyan körmérkőzéses bajnokság, ahol mindenki mindenkivel pontosan egy mérkőzést játszik. Ez $2 \leq n \in \mathbb{N}$ csapat estén csapatonként $n-1$, azaz összesen $\frac{n(n-1)}{2}$ mérkőzést jelent (mindenki csak egyszer jatszik a többiekkel, ezért szükséges a 2-vel való osztás). Legyen $B_k := \binom{k}{2}$, $(k = 1, \ldots, n)$, ekkor bármelyik k csapat pontosan B_k mérkőzést játszik egymással, a mérkőzések száma pedig B_n.

- *Játékos*: a verseny résztvevőit nevezzük játékosoknak, jelölése: $\mathcal{P}_i$ $(i = 1, \ldots, n)$.

- *Eredmény*: a verseny összes mérkőzését lejátszva kialakuló végeredmény. Ennek legegyszerűbb ábrázolása mátrixszal történhet. $W \in \mathbb{N}^{n \times n}$, ahol $i \neq j$, $1 \leq i, j \leq n$ esetén $w_{i,j}$ fogja jelenteni a $\mathcal{P}_i$ játékos által a $\mathcal{P}_j$ játékos elleni mérkőzésen szerzett pontokat. Fontos megjegyezni, hogy általában $w_{i,j} \neq w_{j,i}$, hiszen ha mondjuk egy mérkőzés során $\mathcal{P}_i$ 1, $\mathcal{P}_j$ pedig 3 pontot kap, akkor $w_{i,j} = 1$, $w_{j,i} = 3$. Mivel a csapatok önmagukkal nem játszanak, legyen $w_{i,i} := 0$.

- *Egy játékos elért pontja*: A $\mathcal{P}_i$ játékos elért pontja az összes mérkőzésén szerzett pontjainak összege, azaz mátrixos ábrázolás esetén $v_i = \sum_{j=1}^{n} w_{i,j}$ $(1 \leq i \leq n)$.

- *Pontvektor*: a játékosok elért pontjaiból alkotott vektor, azaz $\mathbf{v} = (v_1, v_2, \ldots, v_n)$. A $\mathcal{P}_i$ játékos pontszáma: v_i.

- *Pontsorozat*: olyan pontvektor, melynek elemei nemcsökkenő sorrendbe rendezettek, tehát $\mathbf{s} = (s_1, s_2, \ldots, s_n)$, ahol $s_1 \leq s_2 \leq \ldots \leq s_n$. Az általánosság megszorítása nélkul feltehető, hogy a játékosok indexelése az elért pontszám szerinti nemcsökkenő sorrendben történik, így a kapott pontvektor mindig monoton növekvő lesz. A továbbiakban mindig sorba rendezett pontvektorokról lesz szó, és a pontsorozat elnevezést fogjuk használni.

- *a-verseny*: olyan n résztvevős verseny, ahol egy mérkőzésen pontosan $1 \leq a \in \mathbb{N}$ pontot osztanak ki. A kiosztott pontok bármilyen egész partícionálása megengedett a két fél között. Jelölése n játékos esetén: $\mathcal{T}_n(a)$. $a = 1$ esetén kapjuk a verseny klasszikus definícióját, ezt röviden $\mathcal{T}_n$-el is jelölhetjük. A versenyeket T, a versenyek halmazát $\mathcal{T}$ jelölje.

- (a,b)*-verseny*: olyan n résztvevős verseny, ahol minden mérkőzésen legalább a, de legfeljebb b pontot osztanak szét, ahol $0 \leq a \leq b$, $a,b \in \mathbb{N}$. Jelölése n játékos esetén: $\mathcal{T}_n(a,b)$. Az (a,b)-versenyeket két osztályba szokás sorolni, ezek a következőek:

- *teljes* (a,b)*-verseny*: olyan (a,b)-verseny, ahol a pontok bármilyen lehetséges módon történő kiosztása megengedett. $a = b$ esetén a korábbi definícióban megfogalmazott a-versenyeket kapjuk.

- *hiányos* (a,b)*-verseny*: nem teljes (a,b)-verseny, másképpen fogalmazva a pontoknak csak bizonyos partícionálása megengedett. Ezeket legegyszerűbben a megengedett eredmények S halmazával definiálhatjuk. Legegyszerűbb példa egy futball bajnokság, ami $(2,3)$-verseny, hiszen minden meccsen legalább 2, legfeljebb 3 pont kerül kiosztásra, ugyanakkor nem minden felosztás lehetséges, csak az $S = ((0,3),(1,1),(3,0))$ felosztások megengedettek.

- *verseny helyreállítása*: adott $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ pontsorozathoz, adott a és b értékekhez egy lehetséges eredmény mátrix konstruálása, azaz olyan M mátrix megadása, melyre: $\sum_{j=1}^{n} m_{i,j} = s_i$, $m_{i,i} = 0$, $(i = 1, \ldots n)$, valamint $a \leq m_{i,j} + m_{j,i} \leq b$, $(i,j = 1, \ldots, n;\ i \neq j)$.

A bevezetőben már esett róla szó, hogy a versenyek nagyon jól ábrázolhatóak gráfokkal is. Az ábrázolás a következőképpen történhet:

$\mathcal{T}_n(a,b)$ esetén gráfunk $G = (V,E)$ egy n csúcsú irányított hurokmentes multigráf (többszörös élek megengedettek, viszont a hurokéleket nem engedjük meg), azaz $|V| = n$. A csúcsok jelentik a játékosokat, és bármelyik két csúcs legalább a, de legfeljebb b éllel van összekötve, ezek az élek jelentik a mérkőzések eredményeit. A $\mathcal{P}_i$ játékos $\mathcal{P}_j$ játékos ellen szerzett pontjait a V_i-ből V_j-be mutató élek száma jelenti, ez megegyezik a mátrixos ábrázolás esetén definiált $w_{i,j}$-vel. Ez azt jelenti, hogy egy játékos elért pontja az őt reprezentáló csúcsból kifelé mutató élek száma (röviden a csúcs kifoka). Ebben az esetben a pontvektor a gráf csúcsai kifokának sorozata, azaz $\mathbf{s} = (\deg^-(V_1), \deg^-(V_2), \ldots, \deg^-(V_n))$. Természetesen továbbra is feltesszük, hogy az indexelés olyan sorrendben történik, hogy a D pontvektor nemcsökkenőleg rendezett, azaz pontsorozatról beszélhetünk.

1.4. Előzmények

A versenyek elemzése hosszú múltra tekint vissza. A hazai kutatók közül például Fekete Mihály (1923), Rédei László (1934), Pólya György (1937), Erdős Pál (1942), Radó Richárd (1943), Szele Tibor (1943), Rényi Alfréd (1959), Gallai Tibor (1960), Ádám András (1965), Szekeres Eszter és Szekeres György (1975), Bollobás Béla (1978), Babai László (1979), Lovász László (1979), T. Sós Vera (1981), Kőnig Dénes (1986), Gyárfás András (1988), Tuza Zsolt (1992), Ruszinkó Miklós (1997), Sali Attila és Simonyi Gábor (1999), Iványi Antal (2001, 2009, 2010) [5, 6], Frank András és Király Tamás (2001), Király Zoltán (2002), Pataki Norbert és Kovács Gábor Zsolt (2002) [13] publikáltak eredményeket. Ezen összefoglalás elsősorban Pataki Norbert és Kovács Gábor Zsolt TDK dolgozata [13] alapján készült.

A két legismertebb, rendszerező mű J. W. Moon könyve [10] és K. B. Reid összefoglaló cikke [16].

Az egyik klasszikus tétel a témával kapcsolatban, hogy minden verseny gráfja tartalmaz Hamilton-utat – ez a tétel része a gráfokról szóló egyetemi tananyagnak.

Diplomamunkám központi témája sorozatok ellenőrzése és helyreállítása. Az első elméleti eredmény Landau tétele [8], amely szükséges és elégséges feltételt ad arra, hogy egy monoton nemcsökkenő sorozat 1-verseny pontsorozata legyen. Ezt a tételt Ford és Johnson 1959-ben a-versenyekre [3], majd 1963-ban J. W. Moon [10] valós 1-versenyekre terjesztette ki. Iványi Antaltól származik az (a, b)-versenyekre vonatkozó szükséges és elégséges feltétel. Ezek a feltételek lineáris időben ellenőrizhetőek. Az viszont még nyitott kérdés, hogy eldönthető-e polinomiális idő alatt tetszőleges sorozatról, hogy lehet-e adott hiányos verseny pontsorozata.

Az 1- valamint a-sorozatokra vonatkozó helyreállítási algoritmusok [8, 4] futási ideje $\Theta(n^2)$, ugyanennyi idő alatt a teljes (a, b)-versenyek pontsorozatai is helyreállíthatóak [5, 6].

A vizsgált helyreállítási feladatok a gráfelméletben szokásos helyreállítási problémával szoros kapcsolatban vannak. Például Erdős és Gallai cikkükben [1] arra adnak feltételeket, hogy egy számsorozat fokszámsorozatként egyértelműen meghatározzon egy egyszerű irányítatlan gráfot. Az 1-versenyekre vonatkozó hasonló kérdésre P. Tetali [19] adta meg a választ.

Ugyancsak rokon feladatok a kombinatorikai kutatásokban népszerű élirányítási kérdések: irányíthatóak-e adott gráf élei úgy, hogy előre megadott feltételeknek eleget tegyenek.

Jelentős erőfeszítések történtek a pontvektorok és pontsorozatok leszámlálásával kapcsolatban. Az 1-versenyek pontsorozatainak számára vonatkozó első eredmények Erdős és Moser 1964-es kéziratában találhatók [10]. Később Moser [11], majd Kleitman és Winston [20] értek el aszimptotikus eredményeket. Narayana és társai [12] rekurzív képlettel $O(n^4)$ idő alatt meghatározták az 1-versenyek pontsorozatainak számát. az a-versenyek jó pontsorozatainak számára vonatkozó aszimptotikus eredményeket tartalmaz Kemnitz és Dolff cikke [7].

A nyolcvanas években kezdődött a témakör párhuzamos algoritmusainak kutatása. Az egyik érdekes eredmény D. Soroker [18] nevéhez fűződik, aki $O(n^2 \log n)$ processzoron $O(\log n)$ idő alatt oldotta meg az 1-versenyek helyreállítását. A hazai eredmények közül Pécsy Gábor és Szűcs László cikke és diplomamunkája érdemel említést.

A csapatoknak az összpontszám alapján történő rangsorolása csak a sok lehetőség egyike. Ismert nehéz probléma a "rossz párok" (egy mérkőzés vesztese megelőzi a mérkőzés győztesét) számának minimalizálása. Itt az első eredmény Erdős és Moon [2] nevéhez fűződik.

2. Korábbi eredmények versenyekkel kapcsolatban

2.1. Teljes 1-versenyek

A $\mathcal{T}_n(1)$ teljes 1-verseny a legegyszerűbb verseny. Itt minden mérkőzésen pontosan 1 pontot osztanak ki, a győztes kap 1 pontot, a vesztes nem kap pontot. A következő, a versenyek egyik alapvető tétele szükséges és elégséges feltételt ad arra vonatkozóan, hogy egy n számból álló sorozat lehet-e $\mathcal{T}_n(1)$ verseny pontsorozata. A tétel Landau nevéhez fűződik 1953-ból [8], és nagyon sok további tétel alapjául szolgál. Azóta nagyon sok másik bizonyítás is született, amelyből körülbelül 10 megtalálható K. B. Reid összefoglaló cikkében [16]. Ezeken kívül találhatóak újabbak J. Griggs és K. B. Reid közös cikkében [4], illetve K. B. Reid és C. Q. Zhang közös cikkében [17].

2.1. Tétel. *Egy nemnegatív egészekből álló* $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ *nemcsökkenő sorozat akkor és csak akkor lehet egy* $\mathcal{T}_n(1)$ *verseny pontsorozata, ha*

$$\sum_{i=1}^{k} s_i \geq \binom{k}{2}, \quad (k = 1, \ldots, n) \tag{1}$$

$k = n$ *esetén egyenlőséggel.*

Bizonyítás:

(i) Szükségesség: ez az egyszerűbb irány.

Mivel egy mérkőzésen 1 pontot osztanak ki, és mivel k játékos pontosan $\frac{k(k-1)}{2} = \binom{k}{2}$ mérkőzést játszik egymással, így bármely k játékos az egymás elleni meccsek során pontosan $\binom{k}{2}$ pontot kell, hogy szerezzen összesen, így a k legkisebb pontszámú is. És mivel legyőzhettek k-nál nagyobb indexű játékost is, így elképzelhető, hogy az összpontszámuk nagyobb $\binom{k}{2}$-nél. Az pedig, hogy $k = n$ esetén egyenlőség áll fent, azért nyilvánvaló, mert n csapat összesen $\binom{n}{2}$ mérkőzést játszik, így azokon pontosan $\binom{n}{2}$ pont kerül kiosztásra.

(ii) Elégségesség: a bizonyítás n szerinti teljes indukcióval történik.

$n = 2$ esetén az állítás nyilvánvaló, hiszen 1 mérkőzés, 1 pont, ami éppen $\binom{2}{2}$. Tegyük fel, hogy az állítás igaz n-re, és legyen adott egy $\mathbf{s} = (s_1, s_2, \ldots, s_{n+1})$ számsorozat, amely kielégíti a tétel feltételeit. Ez a következőket jelenti:

$$\sum_{i=1}^{k} s_i \geq \binom{k}{2}, \quad (k = 1, \ldots, n) \tag{2}$$

valamint

$$\sum_{i=1}^{n+1} s_i = \binom{n+1}{2} \tag{3}$$

(2)-ben k helyére n-et helyettesítve, és ezt (3)-ból kivonva kapjuk:

$$s_{n+1} = \sum_{i=1}^{n+1} s_i - \sum_{i=1}^{n} s_i \leq \binom{n+1}{2} - \binom{n}{2} = n$$

Először vizsgáljuk azt az esetet, amikor $s_{n+1} = n$. Ez azt jelenti, hogy (2)-ben $k = n$-re egyenlőség van, ami az indukciós feltevés szerint azt jelenti, hogy az $(s_1, s_2, \ldots s_n)$ számsorozat egy $\mathcal{T}_n(1)$ verseny pontsorozata, amit egy $W \in \mathbb{N}^{n\times n}$ mátrixszal ábrázolhatunk. Vegyünk a mátrixhoz hozzá egy $(n+1)$-dik sort és oszlopot, és legyen $w_{n+1,j} = 1,\ j = (1, \ldots, n)$, illetve $w_{i,n+1} = 0,\ i = (1, \ldots, n+1)$. Ekkor a kapott mátrix egy $\mathcal{T}_{n+1}$ verseny lesz, melynek $(s_1, s_2, \ldots s_n)$ a pontsorozata.

A másik esetben, amikor $s_{n+1} \leq n$ legyen $d := n - s_{n+1}$. Ekkor létezik legalább d darab s_i úgy, hogy $s_i \geq i - 1$ (itt nem bizonyítom). Csökkentsük a d legnagyobb ilyen tulajdonságú pontot eggyel, ekkor az így kapott $(t_1, t_2, \ldots, t_n)$ számsorozat szintén eleget tesz a tétel feltételeinek (itt nem bizonyítom). Így az indukciós feltevés szerint ez egy $\mathcal{T}_n(1)$ verseny pontsorozata, amit szintén egy $W \in \mathbb{N}^{n\times n}$ mátrixszal ábrázolhatunk. Vegyünk a mátrixhoz hozzá egy $(n+1)$-dik sort és oszlopot, és legyen $w_{n+1,i} = 0, w_{i,n+1} = 1$ azokra az $i = (1, 2, \ldots, n)$ indexekre, ahol eggyel csökkentettünk (így visszakapva az egy pontot), és legyen $w_{n+1,i} = 1, w_{i,n+1} = 0$ azokra az i indexekre, ahol nem csökkentettünk, $w_{n+1,n+1}$ pedig legyen nulla. Ekkor az új mátrix egy $\mathcal{T}_{n+1}$ verseny lesz, melynek pontsorozata $(s_1, s_2, \ldots s_n)$.

■ (Kérdés, hogy leírjak-e ezzel kapcsolatban egy helyreállító algoritmust, illetve a bizonyítás hiányzó részeit leírjam-e, vagy nem érdemes ennyire belemenni ebbe a részbe.) Esetleg ellenőrző algoritmus, futási idők ezen kívül

2.2. Teljes a-versenyek

A $\mathcal{T}_n(a)$ teljes a-verseny az 1-vereny általánosítása. Itt minden mérkőzésen pontosan a pontot osztanak szét a két játékos között, és mivel teljes versenyről van szó, így a a pont bármilyen felosztása megengedett. Legegyszerűbb példa egy labdarúgó bajnokság még a régi 2 pontos rendszerben, amikor a győzelemért 2 pont járt. Itt a pontok tetszőleges partícionálása megengedett, $S = ((0,2),(1,1),(2,0))$ a lehetőségek, így ez egy $\mathcal{T}_n(2)$ verseny.

1963-ban J. W. Moon a következő általánosítását bizonyította Landau tételének [9].

2.2. Tétel. *Egy nemnegatív egészekből álló* $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ *nemcsökkenő sorozat akkor és csak akkor lehet egy* $\mathcal{T}_n(a)$ *verseny pontsorozata, ha*

$$\sum_{i=1}^{k} s_k \geq a\binom{k}{2} = aB_k, \quad (k = 1, \ldots, n) \tag{4}$$

és $k = n$ *esetén egyenlőség áll fent.*

Bizonyítás:

(i) Szükségesség: hasonlóan megy a bizonyítás, mint az előző tételnél.

Mivel minden mérkőzésen pontosan a pont kerül kiosztásra, és bármelyik k játékos pontosan $\frac{k(k-1)}{2} = \binom{k}{2} = B_k$ mérkőzést játszik egymással, így bármelyik k játékos az egymás elleni mérkőzések során pontosan aB_k pontot kell hogy gyűjtsön együttesen, így a k legkisebb pontszámú is. De mivel legyőzhettek k-nél nagyobb indexű játékosokat is, így könnyen elképzelhető, hogy ennél több pontot is szereztek közösen, ezért nem lehet egyenlőség. Ugyanakkor az összpontszámnak mindenképpen meg kell egyeznie a-szor a mérkőzések számával, azaz aB_n-el, így $k = n$-re muszáj egyenlőségnek fennállnia.

(ii) Elégségesség: kérdés, hogy ide mi kerüljön majd, esetleg egy konstruktív bizonyítás helyreállító algoritmussal.

■

2.3. Teljes (a, b)-versenyek

A $\mathcal{T}_n(a, b)$ teljes (a, b)-verseny az a-verseny további általánosítása. Ebben az esetben nem tudjuk pontosan, hány pontot osztunk ki mérkőzésenként, de biztosan egy a és b közötti pontszámot. A pontszámok bármilyen partícionálása megengedett. (a, b)-versenyek pontsorozataira Iványi Antal adott szükséges és elégséges feltételt [5].

Először alsó korlátot adunk a pontsorozatra:

2.1. lemma. *Egy nemnegatív egészekből álló* $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ *nemcsökkentő sorozat akkor és csak akkor lehet egy* $\mathcal{T}_n(a, b)$ *verseny pontsorozata, ha*

$$\sum_{i=1}^{k} s_i \geq aB_k \quad (k = 1, \ldots, n) \tag{5}$$

Bizonyítás: Ez a feltétel nyilvánvalóan szükséges. Hasonlóan az a-versenyeknél leírt szükséges feltételhez, itt is bármelyik k csapat mérkőzésen legalább mérkőzésszám szorozva minimum pontszámmal, azaz aB_k pont kerül szétosztásra. ■

Most próbáljunk meg felső korlátot adni a pontsorozatra:

$$\sum_{i=1}^{n} s_i \leq bB_n$$

Ezen feltétel szükségessége szintén nyilvánvaló, hiszen összesen nem kerülhetett több pont kiosztásra, mint az összes mérkőzés száma szorozva a maximálisan kapható pontszámmal, azaz bB_n.

Egy egyszerű példán látható, hogy ez utóbbi feltétel nem elegséges: legyen $a = 2$, és $b = 5$, a sorozat pedig $\mathbf{s} = (1, 1, 11)$. Itt az jelenti a problémát, hogy $\mathcal{P}_1$ és $\mathcal{P}_2$ mindössze 2 pontot osztott szét a lehetséges 5 helyett, így 3 pont elveszett. Ezért a pontok összege legfeljebb 12 lehet a $bB_3 = 15$ helyett. Ez adhatja az ötletet a következőkhöz.

Vezessünk be egy L_k úgynevezett veszteség függvényt (loss function) ($k = 0, 1, \ldots, n$), amit a következő rekurzióval definiálunk: legyen $L_0 = 0$, és

$$L_k = \max\left(L_{k-1}, bB_k - \sum_{i=1}^{k} s_i\right), \quad k = (1, \ldots, n) \tag{6}$$

Ezzel L_k egy alsó korlátot ad az elvesztett pontokra a $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$ játékosok mérkőzésein. Azt jelenti, hogy az első k játékos elvesztett pontszáma a lehetséges maximum és az általuk megszerzett pontok közötti különbség, de legalább annyi, mint az első $k-1$ játékos elvesztett pontjai. Ez nem feltétlenül a pontos érték, mivel a $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$ játékosok szerezhettek pontokat a $\mathcal{P}_{n+1}, \ldots, \mathcal{P}_n$ játékosok ellen. Ez alapján pontosíthatjuk a pontsorozatra adott felső korlátot:

2.2. lemma. *Ha az $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ pontsorozat egy $\mathcal{T}_n(a, b)$ teljes (a, b)-verseny pontsorozata, akkor*

$$\sum_{i=1}^{k} s_i + (n-k)s_k \leq bB_n - L_k \quad (k = 1, \ldots, n) \tag{7}$$

Bizonyítás: A fenti egyenlőtlenség a következőket jelenti: bal oldalon az első k játékos összpontszáma plussz a maradék $n-k$ játékos pontszáma (játékosonként legalább s_k a sorozat monotonítása miatt) áll. Ennyi pontot mimimálisan szerezniük kell a csapatoknak összesen. Ugyanakkor jobb oldalon a maximálisan szerezhető összpontszám és az elveszett pontok különbsége áll, ami azt jelenti, hogy L_k ismeretében mi lehet az elméletileg maximálisan szerezhető összpontszám. Azaz a fenti feltétel azt jelenti, hogy a minimálisan szerzendő összpontszám nem lehet nagyobb a maximálisan szerezhetőnél, ami nyilvánvaló. ■

Ebből adódik a következő lemma:

2.3. lemma. *Ha az $(s_1, s_2, \ldots, s_n)$ pontsorozat egy $\mathcal{T}_n(a, b)$ teljes (a, b)-verseny pontsorozata, akkor:*

$$aB_k \leq \sum_{i=1}^{k} s_i \leq bB_n - L_k - (n-k)s_k \quad (k = 1, \ldots, n) \tag{8}$$

Bizonyítás: A lemma (5) és (7) algebrai következménye. ■

Kicsit jobban megvizsgálva (7)-et érdekes megfigyelést tehetünk. Nézzük meg mi történik k növelésével: L_k a definíciója miatt (6) monoton növő, azaz az egyenlőtlenség jobb oldala monoton csökkenő. Ugyanakkor $s_k \leq s_{k+1}$ az $\mathbf{s}$ sorozat monotonítása miatt, ami alapján:

$$\sum_{i=1}^{k} s_i + (n-k)s_k \leq \sum_{i=1}^{k} s_i + (n-k)s_{k+1} = \sum_{i=1}^{k+1} s_i + (n-(k+1))s_{k+1}$$

Tehát az egyenlőtlenség bal oldala viszont monoton növekszik k növelésével. Ez azt jelenti, hogy ha egy bizonyos k_0-ra nem áll fenn az egyenlőtlenség, akkor már semmilyen $k \geq k_0$-ra sem fog fennállni, speciálisan $k = n$ esetén sem.

Ebből az következik, hogy elég $k = n$ esetén ellenőrizni a felső korlátot. Ekkor (7)-ben $k = n$ helyettesítéssel kapjuk:

$$\sum_{i=1}^{n} s_i \leq bB_n - L_n$$

Másrészt L_k definíciójából (6) $k = n$ esetén kapjuk, hogy:

$$L_n \geq bB_n - \sum_{i=1}^{n} s_i$$

Azaz a fenti két egyenlőtlenségből azt kapjuk, hogy:

$$\sum_{i=1}^{n} s_i = bB_n - L_n$$

Ezek alapján a 2.3 lemma a következő féleképpen módosítható:

2.4. lemma. *Ha az $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ pontsorozat egy $\mathcal{T}_n(a, b)$ teljes (a, b)-verseny pontsorozata, akkor:*

$$aB_k \leq \sum_{i=1}^{k} s_i \quad (k = 1, \ldots, n)$$

és

$$\sum_{i=1}^{n} s_i = bB_n - L_n \tag{9}$$

2.1. Megjegyzés. *$b = a$ esetén L_k definíciója miatt (6) $L_k = 0,\ k = (1, \ldots, n)$, (9) miatt pedig $k = n$ esetére megkapjuk az egyenlőséget, így lényegében megkapjuk a teljes a-versenyekre vonatkozó 2.2 tételt.*

2.3.1. Az ellenőrző algoritmus

A fenti lemmák alapján megadhatunk egy pontsorozat ellenőrző algoritmust.

Bemenet:
a és b: a mérkőzésenként minimálisan és maximálisan szétosztandó pontszám ($0 \leq a \leq b$);
n: a játékosok száma ($n \geq 2$);
$\mathbf{s} = (s_1, s_2, \ldots, s_n)$: a pontsorozat, azaz egész számok nemcsökkenő sorozata.
Kimenet:
W logikai változó ($\mathsf{W} =$ IGAZ jelenti, hogy $\mathbf{s}$ egy $\mathcal{T}_n(a, b)$ verseny pontsorozata).
$\mathbf{B} = (B_0, B_1, \ldots, B_n)$: a binomiális együtthatók ($B_i = \binom{i}{2}$);
$\mathbf{L} = (L_0, L_1, \ldots, L_n)$: a veszteség függvény értékei (6);
$\mathbf{S} = (S_0, S_1, \ldots, S_n)$: az k legkisebb pontszám összege ($S_k = \sum_{i=1}^{k} s_i$).

PONTELLENŐRZÉS($n, a, b, \mathbf{s}, \mathbf{B}, \mathbf{L}, \mathbf{S}, \mathsf{W}$)

$L_0 \leftarrow S_0 \leftarrow B_0 \leftarrow 0$
$\mathsf{W} \leftarrow$ IGAZ
for $i \leftarrow 1$ **to** n **do**
    $S_i \leftarrow S_{i-1} + s_i$
    $B_i \leftarrow B_{i-1} + i - 1$
    $L_i \leftarrow \max(L_{i-1},\ bB_i - S_i)$
    if $S_i < aB_i$ **then**
        $\mathsf{W} \leftarrow$ HAMIS
        return W
    end if
    if $S_i > bB_n - L_i - (n - i)s_i$ **then**
        $\mathsf{W} \leftarrow$ HAMIS
        return W
    end if

```
15: end for
16: return W
```

Tekintsük a következő pontsorozatot: $\mathbf{s} = (4, 5, 10, 13, 20)$, azt akarjuk eldönteni, hogy lehet-e egy $\mathcal{T}_5(2, 8)$-verseny pontsorozata. Ebben az esetben $L_0 = 0$, $L_1 = 0$, $L_2 = 0$, $L_3 = 5$, $L_4 = 16$, $L_5 = 28$. (8) feltételei fennállnak: $0 \leq S_1 = 4 \leq 64$, $2 \leq S_2 = 9 \leq 65$, $6 \leq S_3 = 19 \leq 55$, $12 \leq S_4 = 32 \leq 51$, $20 \leq S_5 = 52 \leq 52$. Az algoritmus 7. és 11. sorában található elágazások feltételei sosem teljesülnek, így a függvény igaz értékkel tér vissza.

2.3.2. Az algoritmus műveletigénye

Az algoritmus futási ideje a legrosszabb esetben $\Theta(n)$. Ez például akkor fordulhat elő, ha a pontsorozatunk megfelelő. Ugyanakkor rossz sorozatokra a futási idő rövid is lehet. Például ha $a > 0$ és $s_1 = s_2 = 0$, akkor a futási idő $O(1)$. A algoritmus memória igénye $\Theta(n)$.

2.3.3. A probléma megfogalmazása másképpen

Felmerül a kérdés, hogy mi történik akkor, ha egy megadott $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ pontsorozat nem lehet egy $\mathcal{T}_n(a, b)$ verseny pontsorozata. Vajon létezik-e olyan a és b, amelyekre a feltételek már teljesülnek. A problémát megfogalmazhatjuk gráfokkal is. Létezik-e olyan $G = (V, E)$ irányított hurokmentes multigráf, melynek a kifoksorozata pont $\mathbf{s}$, és amennyiben létezik, akkor mi a lehetséges minimális és maximális értéke a két csúcsot összekötő élek számának. A kérdéskörrel Iványi Antal foglalkozott [6], ismertetném a fontosabb eredményeket.

Ha az egy mérkőzésen kapható pontok számát nem korlátozzuk, akkor nagyon egyszerűen konstruálható $\mathcal{T}_n(a, b)$-verseny bármilyen $\mathbf{s}$ pontsorozathoz.

2.5. lemma. *Legyen $n \geq 2$, ekkor bármilyen nemnegatív egészekből álló $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ nemcsökkenő számsorozathoz létezik olyan $\mathcal{T}_n(0, s_n)$ verseny, melynek $\mathbf{s}$ a pontsorozata.*

Bizonyítás: Tekintsük a verseny M eredménymátrixát, legyen $m_{n,1} = s_n$, legyen $m_{i,i+1} = s_i$, $i = (1, 2, \ldots, n-1)$ esetén, és legyen az összes többi $m_{i,j}$ értéke nulla. ■

Azt most már láthatjuk, hogy bármely $\mathbf{s}$ nemnegatív egészekből álló pontsorozathoz létezik (a, b)-verseny, ugyanakkor kérdés, hogy a és b értéke milyen határok között mozoghat, azaz létezik-e a-ra maximum, b-re minimum, és ezek együttesen elérhetőek-e. A kérdés megválaszolásához vezessük be a következő definíciókat:

2.1. Definíció. *Adott egy* T *(a,b)-verseny (a és b értéke nem érdekes), jelölje M az eredménymátrixát, ekkor legyen:*

$$\mathsf{E}(\mathsf{T}) := \max_{1\le i,j\le n} m_{i,j}$$
$$\mathsf{F}(\mathsf{T}) := \max_{1\le i<j\le n} (m_{i,j} + m_{j,i})$$
$$\mathsf{G}(\mathsf{T}) := \min_{1\le i<j\le n} (m_{i,j} + m_{j,i})$$

jelölje $\Delta(\mathbf{s})$ az összes verseny halmazát, melyeknek $\mathbf{s}$ a pontsorozata, ekkor:

$$\mathsf{e}(\mathbf{s}) := \{\min \mathsf{E}(\mathsf{T}) | \mathsf{T} \in \Delta(\mathbf{s})\}$$
$$\mathsf{f}(\mathbf{s}) := \{\min \mathsf{F}(\mathsf{T}) | \mathsf{T} \in \Delta(\mathbf{s})\}$$
$$\mathsf{g}(\mathbf{s}) := \{\max \mathsf{G}(\mathsf{T}) | \mathsf{T} \in \Delta(\mathbf{s})\}$$

A definícióval kapcsolatban érdemes megjegyezni, hogy a $\Delta(\mathbf{s})$ halmaz véges, így $\mathsf{e}, \mathsf{f}, \mathsf{g}$ definícióiban a minimumok és maximumok léteznek. Észrevehetjük, hogy f jelentése nem más, mint legkisebb olyan b, melyhez létezik $\mathcal{T}_n(a,b)$ verseny, melynek a pontsorozata $\mathbf{s}$. Hasonlóan g a legnagyobb ilyen a-t jelenti. A következő lemma felső korlátot ad f-re.

2.6. lemma. *Legyen $n \ge 2$, és legyen $\mathsf{h} = \left\lceil \frac{s_n}{n-1} \right\rceil$ ekkor bármilyen $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ pontsorozathoz létezik olyan $\mathsf{T} \in \mathcal{T}_n(0,b)$ verseny, melyre:*

$$\mathsf{E}(\mathsf{T}) \le \mathsf{h} \quad \text{és} \quad b \le 2\mathsf{h},$$

és h a legkisebb felső korlát e-re, és $2\mathsf{h}$ a legkisebb felső korlát b-re

Bizonyítás: Ha minden játékos a lehető legegyenletesebben szerzi a pontjait, azaz:

$$\max_{1\le j\le n,\, i\ne j} m_{i,j} - \min_{1\le j\le n,\, i\ne j} m_{i,j} \le 1 \quad (i = 1, 2, \ldots, n),$$

akkor

$$\max_{1\le i,j\le n} m_{i,j} \le \max_{1\le j\le n} m_{n,j} = \left\lceil \frac{s_n}{n-1} \right\rceil = \mathsf{h},$$

azaz $\mathsf{E}(\mathsf{T}) \le \mathsf{h}$. Ugyanakkor a $\mathcal{P}_n$ játékosnak $(n-1)$ mérkőzésen s_n pontot kellett gyűjtenie, így biztosan kell lennie olyan mérkőzésnek, amelyen $\mathcal{P}_n$ legalább $\left\lceil \frac{s_n}{n-1} \right\rceil$ pontot szerez, azaz $\mathsf{E}(\mathsf{T}) \ge \mathsf{h}$, így a korlát éles. Mivel $\mathsf{E}(\mathsf{T}) \le \mathsf{h}$, így $\max_{1\le i<j\le n} m_{i,j} + m_{j,i} \le 2\mathsf{h}$, azaz $b \le 2\mathsf{h}$. Legyen a pontsorozatunk $\mathbf{s} = (s_1, s_2, \ldots, s_n) = (c(n-1), c(n-1), \ldots, c(n-1))$, ekkor $m_{i,j} = c = \mathsf{h}$, így $b = 2\mathsf{h}$, tehát ez a korlát is éles. ■

2.1. Következmény. *$n \geq 2$ esetén bármilyen $\mathsf{s} = (s_1, s_2, \ldots, s_n)$ pontsorozatra $\mathsf{e}(\mathsf{s}) = \lceil s_n/(n-1) \rceil$.*

Bizonyítás: Az előző lemma miatt $\mathsf{h} = \left\lceil \frac{s_n}{n-1} \right\rceil$ a létező legkisebb felső korlát. ■

Legyen $S_i = \sum_{j=1}^{i} s_j$, azaz s első i elemének összege, $B_i = \binom{i}{2}$, $i = (1, 2, \ldots n)$. Az n játékos összesen csak úgy szerezhetett S_n pontot, ha $\mathsf{f}B_n \geq S_n$. Ezen észrevétel és az előző lemma alapján kaphatjuk a következő lemmát:

2.7. lemma. *Ha $n \geq 2$ esetén az $\mathsf{s} = (s_1, s_2, \ldots, s_n)$ pontsorozathoz létezik $\mathcal{T}_n(g, f)$ verseny, akkor f-re és g-re teljesülnek a következő korlátok:*

$$\max\left(\left\lceil \frac{S_n}{B_n} \right\rceil, \left\lceil \frac{s_n}{n-1} \right\rceil \right) \leq \mathsf{f} \leq 2 \left\lceil \frac{s_n}{n-1} \right\rceil,$$

$$0 \leq \mathsf{g} \leq \mathsf{f}$$

Bizonyítás: Az állítások következnek a fenti észrevételből, az előző lemmából, valamint f és g definíciójából. ■

Az előző 3 lemma megadta arra a kérdesre a választ, hogy ha egy adott s pontsorozathoz akarunk megfelelő a és b értékeket találni, melyekre s egy $\mathcal{T}_n(a, b)$ verseny pontsorozata lehet, akkor milyen intervallumban érdemes keresnünk. Azt is láttuk, hogy $a = 0$ és $b = 2 \left\lceil \frac{s_n}{n-1} \right\rceil$ esetén létezik s-hez (a, b)-verseny, kérdés, hogy a-t meddig lehet növelni, illetve b-t meddig lehet csökkenteni, másképpen fogalmazva mi f és g pontos értéke. A 2.4 lemma megadta a szükséges feltételt arra vonatkozóan, hogy s a $\mathcal{T}_n(a, b)$ verseny pontsorozata legyen. Ez adhatja az ötletet, hogy végezzük el ezt a pontellenőrzést az összes szóba jöhető f és g értékekre, és így megkaphatjuk a pontos értékeket.

2.3.4. Módszer f és g meghatározására

A 2.4 lemmát jobban megfigyelve észrevehető, hogy a két szükséges feltétel egyike csak a értékétől, míg a másik feltétel csak b értékétől függ, ez azt jelenti, hogy f és g értéke egymástól teljesen függetlenül meghatározható. Ezen értékek meghatározásához használhatjuk a PontEllenőrzés algoritmust, de érdemesebb egy módosított változatot használni. Egyrészt a pontszámok összege (S_i), és a binomiális együtthatók (B_i) nem változnak a és b változtatásával, így ezeket elegendő egyszer kiszámolni, és utána a függvény megkaphatja paraméternek, vagy tekinthetőek globális változóknak. Másrészt az alsó és felső korlátokat érdemes külön ellenőrizni, így az algoritmust érdemes két részre bontani.

Az első algoritmus a PontEllenőrzés egy módisított változata, amelyet f értékének meghatározásához fogunk használni. Egy bemenetként kapott b értékről eldönti, hogy megfelel-e a szükséges feltételnek.

Bemenet:
b: a mérkőzésenként maximálisan szétosztható pontszám $(0 \leq b)$;
Kimenet:
W logikai változó (W = Igaz jelenti, hogy b megfelel a feltétlnek);
Globális változók:
n: a játékosok száma $(n \geq 2)$;
$\mathbf{B} = (B_0, B_1, \ldots, B_n)$: a binomiális együtthatók $(B_i = \binom{i}{2})$;
$\mathbf{S} = (S_0, S_1, \ldots, S_n)$: az k legkisebb pontszám összege $(S_k = \sum_{i=1}^{k} s_i)$.

B-Ellenőrzés(b, W)

$L_0 \leftarrow 0$
W $\leftarrow$ Igaz
for $i \leftarrow 1$ **to** n **do**
    $L_i \leftarrow \max(L_{i-1}, bB_i - S_i)$
    if $S_i > bB_n - L_i - (n-i)s_i$ **then**
        W $\leftarrow$ Hamis
        return W
    end if
end for
return W

A következő algoritmus a PontEllenőrzés másik ágának módosított változata, melyet g meghatározásához fogunk majd használni. Egy bemenetként kapott a értékről dönti el, hogy megfelel-e a szükséges feltételnek.

Bemenet:
a: a mérkőzésenként minimálisan szétosztandó pontszám $(0 \leq a)$;
Kimenet:
W logikai változó (W = Igaz jelenti, hogy b megfelel a feltétlnek);
Globális változók:
n: a játékosok száma $(n \geq 2)$;
$\mathbf{B} = (B_0, B_1, \ldots, B_n)$: a binomiális együtthatók $(B_i = \binom{i}{2})$;
$\mathbf{S} = (S_0, S_1, \ldots, S_n)$: az k legkisebb pontszám összege $(S_k = \sum_{i=1}^{k} s_i)$.

A-Ellenőrzés(a, W)

```
W ← IGAZ
for i ← 1 to n do
    if S_i < aB_i then
        W ← HAMIS
        return W
    end if
end for
return W
```

Hasonlóan az eredeti algoritmushoz, ezen algoritmusok műveletigénye is $\Theta(n)$ legrosszabb esetben, ugyanakkor rossz sorozatokra (illetve jelen esetben rossz a és b értékekre) $O(1)$. Ez az oka annak, hogy b ellenőrzésénél ugyan elég lenne S_n vizsgálata, de mivel az L_i értékek meghatározása mindenképpen szükséges, így érdemesebb S_i-t is minden lépésben ellenőrizni.

Már csak az a kérdés, hogy milyen módszerrel érdemes végignézni az összes szóba jöhető a és b értéket. A legegyszerűbb módszer a lineáris keresés a megadott intervallumon, viszont ez nagy értékekre elég nagy futási idővel rendelkezhet, így sokkal célszerűbb egy logaritmikus keresés alkalmazása.

Az algoritmus ismertetése előtt két fontos megjegyzést érdemes tenni. Egyrészt mi nem egy megadott tulajdoságú elemet keresünk az intervallumban, hanem a legkisebb (legnagyobb) értéket, amely rendelkezik a szükséges tulajdonsággal. Másrészt olyan nem képzelhető el, hogy egy érték nem rendelkezik a megkívánt tulajdonsággal, ugyanakkor található nála kisebb és nagyobb érték is az intervallumon belül, melyek rendelkeznek, azaz egy határ van a két halmaz között. Ez utóbbi könnyedén belátható, ezt most nem bizonyítom, ugyanakkor a logaritmikus kereséshez szükséges ez a feltevés, hogy ne fordulhasson elő, hogy átugrunk jó értékeket.

Bemenet:
n: a játékosok száma ($n \geq 2$);
$\mathbf{s} = (s_1, \ldots, s_n)$: a pontsorozat.
Kimenet:
a: g, azaz a legnagyobb G;
b: f, azaz a legkisebb F.
Globális változók:
n: a játékosok száma ($n \geq 2$);
$\mathbf{B} = (B_0, B_1, \ldots, B_n)$: a binomiális együtthatók ($B_i = \binom{i}{2}$);

$\mathbf{S} = (S_0, S_1, \ldots, S_n)$: az k legkisebb pontszám összege ($S_k = \sum_{i=1}^{k} s_i$);

W: logikai változó (értékét az A- és B-Ellenőrzés függvények állítják be).

```
MinF-MaxG(n, s, a, b)
B_0 ← S_0 ← 0
for i ← 1 to n do
    B_i ← B_{i−1} + i − 1
    S_i ← S_{i−1} + s_i
end for
l ← max(⌈S_n/B_n⌉, ⌈s_n/(n − 1)⌉)
u ← 2⌈s_n/(n − 1)⌉
B-Ellenőrzés(l, W)
if W = Hamis then
    while u ≠ l + 1 do
        B-Ellenőrzés(⌊(u + l)/2⌋, W)
        if W = Igaz then
            u ← ⌊(u + l)/2⌋
        else
            l ← ⌊(u + l)/2⌋
        end if
    end while
end if
b ← u
l ← 0
A-Ellenőrzés(u, W)
if W = Hamis then
    while u ≠ l + 1 do
        A-Ellenőrzés(⌊(u + l)/2⌋, W)
        if W = Igaz then
            l ← ⌊(u + l)/2⌋
        else
            u ← ⌊(u + l)/2⌋
        end if
    end while
end if
a ← l
return a, b
```

2.4. Versenyek helyreállítása

2.4.1. A helyreállítás alapötlete

Az előző fejezetekben szükséges feltételeket tárgyaltunk arra vonatkozóan, hogy egy adott $\mathbf{s} = (s_1, \ldots, s_n)$ pontsorozat egy verseny pontsorozata lehessen, ugyanakkor a feltételek elégségességét nem bizonyítottuk. Ennek egy lehetséges módja egy verseny konstruálása, ami azt jelenti, hogy a verseny $M \in \mathbb{N}^{n\times n}$ mátrixát kitöltjük a feltételeknek megfelelően, azaz úgy, hogy a $\mathcal{P}_i$ játékos pontszáma pontosan s_i legyen, illetve $\mathcal{P}_i$ és $\mathcal{P}_j$ játékosok a és b közötti pontszámon osztozzanak. Ha minden - a szükséges feltételt kielégítő - pontsorozathoz tudunk versenyt készíteni, akkor a feltétel elégséges is.

Az algoritmus alapötlete az, hogy vegyük a legtöbb pontot szerzett játékost $\mathcal{P}_n$-t, és határozzuk meg a mérkőzéseinek eredményét a következő feltételek szerint.

- A $\mathcal{P}_n$ játékos pontszáma megfelelő legyen, azaz $\sum\limits_{j=1}^{n-1} m_{n,j} = s_n$.
- A $\mathcal{P}_n$ játékos a többi játékos elleni mérkőzéseken megfelelő számú ponton osztozkodjon, azaz $a \leq m_{n,j} + m_{j,n} \leq b$, $(j = 1, \ldots, n-1)$.
- A többi játékos pontszámából a $\mathcal{P}_n$ ellen szerzett pontokat levonva, az így kapott $(n-1)$ elemű vektorra, azaz $\mathbf{p}^{(n-1)} = (s_1 - m_{1,n}, s_2 - m_{2,n}, \ldots, s_{n-1} - m_{n-1,n})$-re is teljesüljön, hogy az elemek nemcsökkenő sorrendben követik egymást.
- $\mathbf{p}^{(n-1)} = (p_1^{(n-1)}, \ldots, p_{n-1}^{(n-1)})$-re teljesüljön az $(a$-$b)$-versenyekre vonatkozó szükséges feltétel, azaz $aB_k \leq \sum\limits_{i=1}^{k} p_i^{(n-1)} \leq bB_{n-1} - L_k - (n-1-k)s_k$, $k = (1, \ldots, n-1)$.

Ezzel a módszerrel egy n elemű pontsorozatból $(n-1)$ eleműt csinálhatunk, azaz lépésenként eggyel csökkenthető a dimenzió. Ezt megtehetjük egészen addig, míg egy kettő hosszúságú $\mathbf{p}^{(2)} = (p_1^{(2)}, p_2^{(2)})$ sorozatot nem kapunk, amikor pedig legyen $m_{1,2} = p_1^{(2)}$ és $m_{2,1} = p_2^{(2)}$.

Tekintsük a következő példát, legyen a verseny mátrixa az 1. ábrán látható:

Ez a rész még nagyon hianyos, a következő példa a [6] 57.-60. oldalán található Mini-Max és Score-Slicing2 algoritmusok alapján készült, a sorok számara való hivatkozások is ezen algoritmusok alapján történnek.

Az 01. sorban MinF-MaxG algoritmus tehát megállapította, hogy $a = 2$ és $b = 3$. Következő lépés a mátrix feltöltése a kezdeti értékekkel, illetve az ideiglenes

Játékos/Játékos	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	Pontszám
$\mathcal{P}_1$	-	1	1	1	1	0	0	4
$\mathcal{P}_2$	1	-	1	1	1	0	0	4
$\mathcal{P}_3$	1	1	-	1	0	1	0	4
$\mathcal{P}_4$	1	1	1	-	0	1	0	4
$\mathcal{P}_5$	1	1	2	2	-	1	0	7
$\mathcal{P}_6$	2	2	1	1	1	-	0	7
$\mathcal{P}_7$	3	3	3	3	3	3	-	18

1. ábra. Egy T $(2,3)$-verseny eredmény táblázata

Játékos/Játékos	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	Pontszám
$\mathcal{P}_1$	-	0	0	0	0	0	**0**	4
$\mathcal{P}_2$	3	-	0	0	0	0	**0**	4
$\mathcal{P}_3$	3	3	-	0	0	0	**0**	4
$\mathcal{P}_4$	3	3	3	-	0	0	**0**	4
$\mathcal{P}_5$	3	3	3	3	-	0	**0**	7
$\mathcal{P}_6$	3	3	3	3	3	-	**0**	7
$\mathcal{P}_7$	**3**	**3**	**3**	**3**	**3**	**3**	-	**18**

2. ábra. A T verseny helyreállított táblázata a Score-Slicing algoritmus első futása ($k = 7$) után. A vastaggal szedett értékek már véglegesek.

pontszámvektorba az eredeti pontsorozat értékeinek másolása (02-08). Ezután meghívjuk a Score-Slicing2 függvényt $k = 7$ értékkel, és a $p = (4, 4, 4, 4, 7, 7, 18)$ vektorral. A 02-04 sorokban kiszámoljuk P és A vektorokat: $P = (4, 8, 12, 16, 23, 30, 48)$, $A = (4, 6, 6, 4, 3, 0, 6)$. A 05. sorban megállapítjuk, hogy $M = 0$, így az $M > 0$ feltételek sehol sem teljesülnek, az algoritmus futása az utolsó sorával folytatódik, ahol visszatér a kapott ideiglenes számsorozattal.

A következő lépésben $k = 6$, $p = (4, 4, 4, 4, 7, 7)$ paraméterekkel hívjuk a Score-Slicing függvényt. P, A és M értékei a következőek lesznek: $P = (4, 4, 12, 16, 23, 30)$, $A = (4, 6, 6, 4, 3, 0)$, $M = 5 * 3 - 7 = 8$. Mivel $M > 0$ és $A_5 > 0$, a 06-22. sorokban található ciklussal folytatjuk. A 07-12. sorokban megkapjuk, hogy $x = 5$ és $f = 1$. A 13-14. sorokból kapjuk, hogy $d = 3$ és $m = 3$, majd a 15. sorban kezdődő ciklust 1-szer végrehajtva kapjuk, hogy $y = 3$, $m_{5,6} = 0{+}3 = 3$, $p_5 = 7{-}3 = 4$, $m_{6,5} = 3{-}3 = 0$, $M = 8 - 3 = 5$. A 21. sorból jön, hogy $A_5 = 3 - 3 = 0$, ami miatt a 06. sorban

Játékos/Játékos	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	Pontszám
$\mathcal{P}_1$	-	0	0	0	0	**0**	**0**	4
$\mathcal{P}_2$	3	-	0	0	0	**0**	**0**	4
$\mathcal{P}_3$	3	3	-	0	0	**0**	**0**	4
$\mathcal{P}_4$	3	3	3	-	0	**0**	**0**	4
$\mathcal{P}_5$	3	3	3	3	-	**3**	**0**	7
$\mathcal{P}_6$	**2**	**2**	**2**	**2**	**0**	-	**0**	7
$\mathcal{P}_7$	**3**	**3**	**3**	**3**	**3**	**3**	-	**18**

3. ábra. A T verseny helyreállított táblázata a Score-Slicing algoritmus második futása után ($k = 6$). A vastaggal szedett értékek már véglegesek.

kezdődő ciklus többször nem hajtódik végre, így a vezérlés a 23. sorban folytatódik. A 24-27. sorban lévő ciklus lefutása után az M mátrixban a következő változások történnek: $m_{6,4} = 3 - 1 = 2$, $m_{6,3} = 3 - 1 = 2$, $m_{6,2} = 3 - 1 = 2$, $m_{6,1} = 3 - 1 = 2$, valamint M értéke 1-re csökken. Tovább már nem tudunk csökkenteni, a Score-Slicing futása a $k = 6$ paraméterrel véget ér.

A 3. ábra mutatja az aktuális állapotot. Ha alaposan megfigyeljük, akkor észrevehetjük, hogy baj van, ugyanis a $\mathcal{P}_6$ játékos pontszáma 8 lett, pedig csak 7 kellene, hogy legyen. A probléma forrása az, hogy $\mathcal{P}_6$ 3 pontot adott $\mathcal{P}_5$-nek, azaz $m_{5,6} = 3$, miközben $P_6 = 30$, $B_6 = 15$, így az első 6 játékos egymás elleni mérkőzésein pontosan 2 pontot lehetne csak szétosztani mérkőzésenként. Ez azért fordulhatott elő, mert a pontok szeletelésénél nem vettük figyelembe azt a tényt, hogy $A_6 = 0$. Ez ugyanis azt jelenti, hogy az első 6 játékosnak együttesen semennyi további pontja sincs a minmálisan szerzendőhöz képest (ami mérkőzésszám és a minimális pontszám szorzata, jelen esetben $B_6 \cdot a = 15 \cdot 2 = 30$), így egyik mérkőzésen sem lehetne a a-nál több pontot szétosztani. Ez általános esetben azt jelenti, hogy amikor a $\mathcal{P}_k$ játékos pontjait szeleteljük, akkor egyik játékos sem kaphat belőle többet $A_k + a$-nál. Ezen kívül A_k értékét az a feletti értékkel csökkenteni kell, ugyanis ez jelenti azt, hogy mennyivel oszthatunk több pontot szét a minimumnál.

Egy másik fontos észrevétel a következő: A_i jelentése az, hogy az első i játékos összpontszámát legfeljebb mennyivel növelhetjük. Ha a $\mathcal{P}_i$ játékosnak pontokat adunk a $\mathcal{P}_k$ játékostól, akkor az első i játékos összpontszáma nő, ezáltal A_i értéke csökken, ugyanakkor bármilyen ($k > j > i$)-re az első j játékos összpontszáma is nő, ami miatt A_j értéke szintén csökken. Ezt a csökkentést egészen A_{k-1}-ig végig kell vinni. A_j viszont sehol sem lehet negatív, mert az azt jelentené, hogy $\mathcal{P}_j$ pontszáma már túl sok. Ezekből viszont az következik, hogy nem létezhet olyan, hogy $A_i > A_j$ miközben $i < j$, hiszen például nem növelhetjük az első két játékos összpontszámát 5-el, miközben az első négy játékosét csak 1-el lehetne. Ez azt jelenti, hogy az A sorozat tagjai monoton növekvőek kell, hogy legyenek. Ezt a tulajdonságot minden lépés után helyre kell állítani a túl nagy értékek csökkentésével. Az előző példán szemléltetve: $A = (4, 6, 6, 4, 3, 0)$ helyett $A = (3, 3, 3, 3, 3, 0)$ lenne a helyes. Fontos még megjegyezni, hogy a k. taggal nem kell foglalkozni ebből a szempontból, hiszen ott csak csökkentünk, nem növelünk.

Ezen megfontolások alapján kaphatjuk a következő algoritmust:

2.4.2. A helyreállító algoritmus

Pont-Szeletelés$(k, \mathbf{p}, R)$

$P_0 \leftarrow 0$
for $i \leftarrow 1$ **to** k **do**
    $P_i \leftarrow P_{i-1} + p_i$
    $A_i \leftarrow P_i - aB_i$
end for
for $i \leftarrow k - 2$ **downto** 1 **do**
    if $A_i > A_{i+1}$ **then**
        $A_i \leftarrow A_{i+1}$
    end if
end for
$M \leftarrow (k - 1)b - p_k$
$\mathsf{C} \leftarrow$ Igaz
while $M > 0$ **and** $\mathsf{C} =$ Igaz **do**
    $\mathsf{C} \leftarrow$ Hamis
    $x \leftarrow k - 1$
    while $r_{x,k} = b$ **or** $r_{x,k} - A_k \geq a$ **do**
        $x \leftarrow x - 1$
    end while
    $f \leftarrow 1$
    while $p_{x-f+1} = p_{x-f}$ **and** $x - f \geq 1$ **do**
        $f \leftarrow f + 1$
    end while
    $d \leftarrow p_{x-f+1} - p_{x-f}$

$m \leftarrow \min(b, d, \lceil A_x/f \rceil, \lceil M/f \rceil)$
for $i \leftarrow f$ **downto** 1 **do**
$n \leftarrow A_k + \max(0, a - r_{x+1-i,k})$
$y \leftarrow \min(b - r_{x+1-i,k}, m, M, A_{x+1-i}, p_{x+1-i}, n)$
if $y \neq 0$ **then**
$\mathsf{C} \leftarrow \textsc{Igaz}$
$r_{x+1-i,k} \leftarrow r_{x+1-i,k} + y$
$p_{x+1-i} \leftarrow p_{x+1-i} - y$
$r_{k,x+1-i} \leftarrow r_{k,x+1-i} - y$
$M \leftarrow M - y$
if $r_{x+1-i,k} > a$ **then**
$A_k \leftarrow A_k - \min(y, r_{x+1-i,k} - a)$
end if
for $j \leftarrow i$ **downto** $x - k + 2$ **do**
$A_{x+1-j} \leftarrow A_{x+1-j} - y$
end for
for $j \leftarrow k - 2$ **downto** 1 **do**
if $A_j > A_{j+1}$ **then**
$A_j \leftarrow A_{j+1}$
end if
end for
end if
end for
end while
if $M > 0$ **then**
for $i \leftarrow k - 1$ **downto** 1 **do**
$y \leftarrow \min(r_{k,i}, M, r_{k,i} + r_{i,k} - a)$
$r_{k,i} \leftarrow r_{k,i} - y$
$M \leftarrow M - y$
end for
end if
return $\mathbf{p}, R$

2.4.3. A helyreállító algoritmus magyarázata

Az algoritmus paraméterként megkap egy k értéket, ami az aktuális lépésben a játékosok számát jelenti, egy $\mathbf{p}^{(k)} = (p_1, p_2, \ldots, p_k)$ vektort a pontszámokkal (a továbbiakban és a fenti pszeudokódban egyszerűsítve $\mathbf{p}$), valamint az R mátrixot, melybe a mérkőzések eredményei kerülnek.

Az első 5 sorban $\mathbf{P} = (P_1, P_2, \ldots, P_k)$ és $\mathbf{A} = (A_1, A_2, \ldots, A_k)$ kiszámítása történik, hasonlóan Score-Slicing2 algoritmushoz. P_i jelentése az i legkisebb pontszám összege, csak $\mathbf{A}$ kiszámításához van szükségünk rá. $\mathbf{A}$ pedig az úgynevezett további pontokat jelenti (additional points), A_i jelentése, hogy a $\mathcal{P}_i$ játékos pontszámát legfeljebb mennyivel csökkenthetjük (másképpen fogalmazva a $\mathcal{P}_i$ játékos legfejebb hány pontot nyerhetett el a $\mathcal{P}_k$ játékostól) úgy, hogy csökkentett sorozatra is teljesüljön a 2.4 lemma első feltétele, azaz $aB_i \leq \sum_{j=1}^{i} p_j$, $(i = 1, 2, \ldots, k)$. Korábban már említettem, de érdemes újra megjegyezni, hogy p_i csökkentésével P_j is csökken minden $j \geq i$ esetén, így A_j-nek is csökkennie kell $j \geq i$ esetén. Mivel az $\mathbf{A}$ vektor elemei nem lehetnek negatívak, hiszen ez azt jelentené, hogy már túl sokat csökkentettünk, és nem teljesülne a szükséges feltétel, így az sem fordulhat elő, hogy $A_j < A_i$ miközben $j > i$. A 6-10. sorig tartó ciklus azt a célt szolgálja, hogy csökkenti $\mathbf{A}$ elemeit ott, ahol szükséges, hogy azok monoton növekvő sorban kövessék egymást.

A 11. sorban M kiszámítása történik, ami az úgynevezett hiányzó pontokat (missing points) jelenti. Ez a elnevezés a $\mathcal{P}_k$ játékos hiányzó pontjaira utal, ami maximálisan megszerezhető és a megszerzett közti különbség. Ezek a pontok valahol elvesztek, ami az jelenti, hogy egyrészt kisebb indexű játékos elnyerte őket $\mathcal{P}_k$-tól, másrészt b-nél kevesebb pont került kiosztásra a mérkőzéseken. A feladat a hiányzó pontok helyének megtalálása, ezáltal M nullára csökkentése. Ha ezt elértük, az azt jelenti, hogy $\mathcal{P}_k$ eredményeit meghatároztuk, és versenyt $(k-1)$ résztvevősre redukálhatjuk. Mindezt természetesen úgy, hogy a maradék pontsorozatra a fejezet elején ismertetett feltételek fennálljanak.

A 12. sorban található C logikai változó jelentése, hogy 13-47. sorig tartó ciklus egyszeri lefutása során történt-e változás az eredménymátrixban. Ha nincs akkor kilépünk a ciklusból, ugyanis akkor a többszöri lefutás sem hozna változást. A 13. sorban kezdődő ciklusban hiányzó pontok azon részét keressük meg, melyeket más csapatok nyertek el. x jelenti, hogy melyik legnagyobb indexű játékos, amelynek meg lehet növelni a pontjait, azaz $\mathcal{P}_k$ pontokat adhat át neki. f jelenti, hogy hány játékos rendelkezik azonos pontszámmal, ezeknél párhuzamosan növelünk.

m jelenti, hogy $\mathcal{P}_k$ előreláthatólag mennyi pontot adhatunk át annak a játékosnak, amelynek épp növelni szerertnénk a pontjait.

itt hianyzik meg par dolog

Ha kiléptünk a 13.-47. sorig tartó ciklusból, de M még nem nulla, akkor ez azt jelenti, hogy a maradék hiányzó pontok nem kerültek kiosztásra a mérkőzéseken. A 49.-53. sorokban $\mathcal{P}_j$ más játékosok ellen szerzett pontszámait csökkentjük a lehetséges mértékben. Azaz nem lehet negatív, nem lehet egy mérkőzésen a-nál kevesebb pont, és M-nél többet sem csökkenthetünk.

3. Hiperversenyek pontsorozatai

3.1. Hiperversenyekről általánosságban

A bevezetőben említettem, hogy a rangsorolásra egy népszerű módszer az objektumok páronkénti összehasonlítása, ami alapján pontszámokat osztunk szét közöttük. De könnyen elképzelhető, hogy a páronkénti összehasonlítás különböző okokból kifolyólag nem lehetséges. Példának okáért képzeljük azt el, hogy a játékosaink az ulti nevű kártyajátékban szeretnék összemérni erejüket. Az ultit közismerten három játékos játsza, így nem lenne lehetséges a játékosok páronkénti összehasonlítása. Továbbra is a kártyánál maradva gondolhatunk a bridzsre vagy épp a tarokkra, ezeket a játékokat pedig négyen játszák.

Ezek alapján kézenfekvő lehet az ötlet, hogy ne páronként hasonlítsunk, hanem ulti esetén hármasával, bridzs vagy tarokk esetén négyesével, általános esetben k-asával. n játékos esetén vegyük az összes lehetséges k elemű részhalmazát a játékosoknak, és az éppen összehasonlított k játékos között osszuk szét a pontokat, ezt nevezzük k-hiperversenynek. Egyszerű meggondolással látható, hogy ez $k = 2$ esetén pont az eddig tárgyalt "hagyományos" versenyeket adja. n elem k elemű ($n \geq k$) részhalmazainak a száma pontosan: $\binom{n}{k}$, tehát ha az összes lehetséges módon hasonlítani akarunk, az ennyi összehasonlítást, vagyis mérkőzést jelent. Egy játékos mérkőzéseinek száma pedig $\binom{n-1}{k-1}$, hiszen a maradék $(n-1)$ játékos közül kell neki $(k-1)$ ellenfelet választani az összes lehetséges módon.

Szintén említésre került korábban, hogy a versenyek szoros kapcsolatban állnak a gráfokkal, egészen pontosan az irányított, hurokmentes teljes gráfokkal. A csúcsok megfeleltethetőek a játékosoknak, míg az élek a mérkőzéseknek. Hiperversenyek esetében a mérkőzések kettőnél több résztvevő között zajlanak, így az élek is egyszerre kettőnél több csúcsot kell, hogy összekössenek, ezen gondolat alapján eljuthatunk a hipergráfokhoz. A hipergráf olyan gráf, melyben az élek nem csúcspárokat jelentenek, hanem csúcshalmazokat (nem szükségszerűen azonos elemszámú halmazokat). Egy olyan élet, amely k csúcsot tartalmaz k-élnek hívunk. Ha egy hipergráf minden éle k-él, akkor a gráfot k-hipergráfnak nevezzük. Egy k-hipergráf teljes, ha élhalmaza tartalmazza az összes lehetséges k-élet. Ezen fogalmak alapján látható, hogy egy k-hiperversenynek pont egy teljes k-hipergráf fog megfelelni. Megjegyezném, hogy a hiperverseny (hypertournament) elnevezés is a hipergráfokkal való szoros kapcsolatból adódik.

Hasonlóan a versenyekhez, a hiperversenyeket is lehet ábrázolni mátrixokkal, bár az ábrázolás módja kevésbe egyértelmű. Egy lehetőség k-hiperverseny ábrázolására

Játékos/Halmaz	$\mathcal{P}_1, \mathcal{P}_2$	$\mathcal{P}_1, \mathcal{P}_3$	$\mathcal{P}_2, \mathcal{P}_3$	$\mathcal{P}_1, \mathcal{P}_4$	$\mathcal{P}_2, \mathcal{P}_4$	$\mathcal{P}_3, \mathcal{P}_4$	Pontszám
$\mathcal{P}_1$	-	-	0	-	0	0	0
$\mathcal{P}_2$	-	1	-	0	-	0	1
$\mathcal{P}_3$	0	-	-	0	1	-	1
$\mathcal{P}_4$	1	1	0	-	-	-	2

4. ábra. Egy 4 résztvevős 3-hiperverseny eredményének táblázata.

egy k dimenziós mátrix ($M \in \mathbb{N}^{n \times n \times, \ldots, \times n}$) alkalmazása, tehát egy olyan mátrixé, amelyben az elemeknek nem 2, hanem k indexük van, viszont ez az ábrázolás több okból is problémás lehet. Egyrészt nem egyértelmű, hogy mi kerüljön a mátrix egy adott $(i_1, i_2, \ldots, i_k,)$, $(i_s \neq i_t$, ha $s \neq t)$ indexű, és ennek $k!$ számú permutációival indexelt helyeire, miközben csak k értékünk van (hagyományos versenyek esetén egyszerű volt a dolog, mivel $2! = 2$). Másrészt nehéz lenne egy ilyen mátrixot írásban áttekinthetően ábrázolni.

Egy másik lehetőség - ezt az ábrázolást fogom alkalmazni - egy hagyományos 2 dimenziós mátrix, melynek sorai fogják jelenteni a játékosokat, ennek következtében n sora lesz. Oszlopai pedig az n játékos összes lehetséges $(k-1)$ játékosból álló részhalmazát, így a mátrix oszlopainak a száma $\binom{n}{k-1}$. Ebben az esetben $m_{i,j}$ jelentése az, hogy a $\mathcal{P}_i$ játékos hány pontot szerzett a j. oszlop által jelentett $\mathcal{P}_{j_1}, \mathcal{P}_{j_2}, \ldots, \mathcal{P}_{j_{k-1}}$ játékosok elleni mérkőzésen. Ebben az esetben a $\mathcal{P}_i$ játékos pontszáma pontosan a mátrix i. sorában található értékek összege. Amennyiben $\mathcal{P}_i \in \{\mathcal{P}_{j_1}, \ldots \mathcal{P}_{j_{k-1}}\}$, akkor természetesen legyen $m_{i,j} = 0$, mivel a játékosok nem játszanak saját magukkal, és így nem romlik el a sorösszeg. A 4. ábrán látható egy példa az említett ábrázolásra (a táblázatban "-"-el jelöltem ezeket az értékeket a jobb áttenkinthetőség kedvéért).

3.2. Teljes k-hiperversenyek

Hivatkozások

[1] P. Erdős, T. Gallai, Graphs with prescribed degrees of vertices (in Hungarian) *Matematikai Lapok* **11** (1960), 264-274.

[2] P. Erdős, J. W. Moon: On sets of consistent arcs in a tournament. *Canadian Mathematical Bulletin*, **8** (1965) 269-271.

[3] L. S. Ford, S. M. Johnson, A tournament problem, *American Mathematical Monthly*, **66** (1959), 387-389.

[4] J. Griggs, K. B. Reid, Landau's theorem revisited, *Australas. J. Comb.*, **20** (1999) 19-24.

[5] A. Iványi, Reconstruction of interval tournaments, *Acta Univ. Sapientiae, Informatica*, **1**, 1 (2009), 71-88.

[6] A. Iványi, Reconstruction of interval tournaments II, *Acta Univ. Sapientiae, Mathematica*, **2**, 1 (2010), 47-71.

[7] A. Kemnitz, S. Dolff, Score sequences of multitournaments, *Congressus Numerantium*, **127** (1997), 85-95.

[8] H. G. Landau, On dominance relations and the structure of animal societies. III. The condition for a score sequence, *Bull. Math. Biophys.*, **15** (1953), 143-148.

[9] J. W. Moon, An extension of Landau's theorem on tournaments, *Pacific J. Math.*, **13** (1963) 1343-1345.

[10] J. W. Moon, *Topics on Tournaments*, Holt, Rinehart and Winston. New York, 1968.

[11] L. Moser, Asymptotics of tournament scores, *Combinatorics XIX. American Mathematical Society*, Providence (1971), 175-176.

[12] T. V. Narayana, D. H. Bent, Computation of the mumber of tournament score sequences in round-robin tournaments, *Canad. Math. Bull.*, **7** (1964), 133-136.

[13] N. Pataki, G. Zs. Kovács, *Rangsorolási sorozatok elemzése.*, TDK dolgozat, Budapest, 2002.

[14] S. Pirzada, G. Zhou, On k-hypertournament losing scores, *Acta Univ. Sapientiae, Informatica*, **2**, 1 (2010), 5-9.

[15] S. Pirzada, G. Zhou, A. Iványi, Score lists in multipartite hypertournaments, *Acta Univ Sapientiae, Informatica*, **2**, 2 (2010), 184-193.

[16] K. B. Reid, Tournaments: Scores, kings, generalizations and special topics, *Congr. Numer.*, **115** (1996), 171-211.

[17] K. B. Reid, C. Q. Zhang, Score sequences of semicomplete digraphs, *Bull. Inst. Combin. Appl.*, **24** (1998) 27-32.

[18] D. Soroker, Fast parallel algorithms for finding Hamiltonian paths and cycles in a tournaments. *Journal of Algorithms*, **9** (1) (1988), 276-286.

[19] P. Tetali, Unique tournaments, *Journal of Combinatorical Theory Series B* **72** (1) (1998), 157-159.

[20] K. J.Winston, D. J. Kleitman, On the asymptotic number of tournament score sequences, *J. Combin. Theory Ser. A*, **35** (1983), 208-230.