Bevezetés

A stabil párosítások elmélete 1962-ben, Gale és Shapley [7] cikke nyomán vált ismertté, és indított el egy széleskörű matematikai kutatást ebben a témakörben. A probléma felvetésén túl a nevezetes cikk tartalmazott egy algoritmust az alapfeladat megoldására, és kijelölt két lehetséges általánosítást. A fenti cikk bevezető példája a következő volt:
"Lányok és fiúk keresnek házaspárt maguknak. Mindenki felállít egy preferenciát az ellenkező neműek között, tehát felsorolja, hogy ki tetszik neki legjobban, ... illetve legkevésbé. Egy lány-fiú párokból álló párosítást akkor nevezünk *stabilnak*, ha nem lesz olyan lány és fiú, akik nincsenek párban, pedig kölcsönösen jobban tetszenek egymásnak, mint a jelenlegi házastársuk."
A páros gráfokon történő vizsgálódásnak az egyik lehetséges általánosítása a *b-párosítás*, ahol nem csak egy, hanem adott számú párt engedünk meg minden pontnak. Ennek egy fajtája az *egy-a-sokhoz* feladat, amelyet például hazánkban is használatnak a felvételi pontszámok kiszámolása. A probléma egy lehetséges megoldását adja az eredeti cikkben Gale és Shapley által javasolt *lánykérő* algoritmus.
A másik fő általánosítási lehetőség, ha nem csak páros gráfokon keresünk stabil párosítást. Ennek, az egyszerűbben *szobatárs* problémának elnevezett feladatnak már nem feltétlenül létezik megoldása. Először Irvingnek sikerült '85-ben polinomiális algoritmussal teljes stabil párosítást találnia [8], ha a gráfban létezett ilyen, majd Tan adott pontos karakterizációt [12], és Hsuehval közösen [13] egy másik algoritmust is a feladatra. Végül Scarf '67-ben publikált [11] algoritmusát felhasználva jutottak el nemrégiben egészen más módon ugyanezen eredményekhez.

Dolgozatom célja a két fenti általánosítás ötvözetének, a **stabil b-párosítás gráfokon** problémakör leírása és algoritmikus megoldása. A probléma karakterizációját a Scarf-lemma segítségével írom le, míg az algoritmikus implementációra példaként Scarf algoritmusán kívül Cechlárová és Fleiner [3] konstrukción alapuló visszavezetését, és a Tan-Hsueh algoritmus kiterjesztését fogom bemutatni. Ez utóbbi új eredménynek mondható. Scarf és Tan-Hsueh algoritmusainak működését a mellékletben minden problémakör esetén egy-egy példán szemléltetem. A példák forrásait, az algoritmusok MAPLE-ben beprogramozott implementációit a honlapomon (www.cs.bme.hu/~pbiro) találhatja meg, és próbálhatja ki az Olvasó.

Ezúton is szeretném megköszönni konzulensemnek, Fleiner Tamásnak a rengeteg segítséget.

Tartalomjegyzék

1 Stabil párosítás és b-párosítás páros gráfon

A fejezet során röviden szeretném bemutatni az elmélet kiindulópontját, a páros gráfon vett stabil párosítás keresésének problémáját, majd ennek természetes általánosítási lehetőségeit Gale és Shapley [7] klasszikus cikke alapján. Magasabb szintű leíró modelleket, kapcsolatokat más tudományterületekhez, és gyakorlati alkalmazási lehetőségeket Fleiner [6] átfogó tanulmányában találhat az Olvasó. Én ebből csupán egy általánosabb leíró modellt és egy alkalmazási lehetőséget mutatok be ebben a problémakörben.

Gale és Shapley [7] cikkükben egy egyetemi felvételi eljárást ismertetnek, amely igazságos abból a szempontból, hogy végeredményeként nem lesz olyan a jelentkező és A egyetem, hogy a-t csak B egyetemre vették fel, pedig listájában A előlrébb van rangsorolva, mint B és az A egyetem is kénytelen volt felvenni a-nál gyengébb jelentkezőt (tehát a kiosztás *stabil*). Sőt az algoritmus azt is biztosítja, hogy a stabil kiosztások közül minden jelentkező a lehető legjobb egyetemre nyer felvételt. Ennek belátására definiálták a feladatot először csak lányok és fiúk esetére, amikor mindenki csak egy párt választhat magának.

Rögtön az elején szeretném felhívni a figyelmet a definíciók egyfajta kettősségére, ami talán nem szerencsés, de a cikkek eredeti nyelvezetén nem kívántam túlzott mértékben változtatni. A kezdeti cikkek, és például Irving, Tan és Hsueh cikke is a pontok szerint definiálja a fogalmakat, míg az újabb irodalmak inkább az élek szerint értelmezik a problémát. Ennek oka, hogy egyrészt a párhuzamos élek így könyebben értelmezhetők, másrészt sok olyan újfajta leírásmód létezik (Scarf algoritmusa, vagy a matroid-kernelek), amelyben természetesebb alaphalmaznak az éleket tekinteni.

Az alapfeladatban tehát egy $G(A, B)$ páros gráfon értelmezzük a két ponthalmaz egymáshoz fűzött preferenciáit, melyet gyakran preferencia-listákban adunk meg. Ha például egy $a \in A$ pont listájában 3 elem szerepel: $[b_1, b_2, b_3]$, akkor ez jelentse azt, hogy a leginkább b_1-et kedveli, másodsorban b_2-t, és utolsóként b_3-at. (Lehet, hogy több pont is van a B halmazban, akikkel a nincs összekötve, vagyis jobban szeret egyedül maradni, mint bármelyikkel is kapcsolatba lépni.) Ha a megfelelő éleket sorrendben e_1, e_2 és e_3 betűkkel jelöljük, akkor a preferenciákat kétféleképpen is leírhatjuk: $b_1 <_a b_2 <_a b_3$ a pontok szerint és $e_1 <_a e_2 <_a e_3$ az élek szerint. Egy $M \subseteq V(G)^2$ párosítás a pontok nyelvén a pontpárok egy halmazát, míg $S \subseteq E(G)$ az élek egy halmazát jelöli. Ha például az $< a, b_2 >$ pár eleme M-nek vagyis $e_2 \in S$, akkor a pontok szerint a párjánál kedvezőbb éleket superior élnek, a rosszabb éleket pedig inferior élnek nevezzük, így például a fenti példában a pont esetében (a, b_1) él superior él, és (a, b_3) él inferior él. Élek esetében azt mondjuk, hogy e_2 él *dominálja* e_3 élet, illetve nem dominálja e_1 élet az a pontban. Ha egy párosítás esetében létezik olyan él, amelyik mintkét végpontjából nézve superior él, vagy másképpen mondva egyik végpontjában sincs dominálva párosítás-beli éllel, akkor ez az él *blokkolja* a párosítást. Ha nincs blokkoló él, akkor a párosítás *stabil.*

1.1 Stabil párosítás páros gráfon

A lányok és fiúk példájában tehát egy egyszerű páros gráfban – amely a pontok preferenciáival van megadva – keresünk stabil párosítást. A feladatot Gale és Shapley vetette fel és bizonyította elsőként a *lánykérő* algoritmus segítségével.

Tétel 1.1. *Páros gráf esetén, ahol a pontok preferenciái adottak, mindig létezik stabil párosítás.*

Párosítás azonban több is létezhet egy adott gráfra. Egy M stabil párosítás b pont számára *optimális*, ha nem létezik egy másik stabil párosítás, amelyben jobb párt kapott volna, mint M-ben. Ha ez minden $b \in B$ pontra, vagyis minden fiúra teljesül, akkor az adott stabil párosítást *fiú-optimális* párosításnak nevezzük.

Tétel 1.2. *Páros gráf esetén mindig létezik fiú-optimális (vagy lány-optimális) stabil párosítás.*

Bizonyítás. A két fenti tétel konstruktív módon egyszerre belátható, ugyanis létezik egy algoritmus, amely bármely páros gráf esetén talál fiú-optimális (vagy lány optimális) stabil párosítást. Ez az algoritmus az ún. *lánykérő* algoritmus a következőképpen működik:
Minden fiú ajánlatot tesz az általa legkedveltebb lánynak. A lányok, ha több kérő közül is választhatnak, akkor csak a legjobb kérőt tartják meg, a többit visszautasítják. A második körben a visszautasított fiúk megkérik a listájukon szereplő második lány kezét. Az algoritmus addig folytatódik, amíg az egyik körben már nem lesz új lánykérés. Ez $O(n^2)$ időn belül bekövetkezik, hiszen kétszer egyik fiú sem tesz ajánlatot ugyanannak a lánynak. A végső állapotról (nevezzük M élhalmaznak) azt fogjuk belátni, hogy egyrészt stabil párosítás, másrészt optimális a fiúk számára.
Mivel minden lány csak egy ajánlatot tart meg mindig, és a fiúk is egyszerre csak egy lánynak tesznek ajánlatot, ezért a maradék élhalmaz párosítás. Amennyiben létezne egy a lány és egy b fiú, akik kölcsönösen jobban tetszenek egymásnak, mint az M-beli párjuk, akkor ez azt is jelentené, hogy az algoritmus során az a lány egyszer már visszautasította a b fiút, tehát volt már jobb kérője, mint b. E miatt M-ben is kedvezőbb párja lesz b-nél, ezért a párosítás stabil.
Az optimalitást indukcióval bizonyítjuk indirekt módon. Legyen b_1 az első olyan fiú, akit visszautasított a_1 lány, pedig létezik olyan stabil párosítás, ahol ők párok lehetnének. Ekkor az algoritmus szerint kell legyen olyan b_2 fiú, aki jobban tetszik a_1-nek, és ezért utasította vissza b_1-et. Megmutatjuk, hogy ebben az esetben a_1 elérhetetlen b_1 számára. Az indukciós feltevés szerint eddig b_2-t csak olyan lányok utasították vissza, akik elérhetetlenek számára. Ha viszont egy M stabil párosításban a_1 mégis b_1-el kerülhetne párba, ez azt jelentené, hogy b_2 rosszabb párt kap, mint a_1, és a_1 is rosszabb párt kap, hiszen $b_2 <_{a_1} b_1$, tehát (a_1, b_2) él blokkolná a párosítást. Ez ellentmondás, tehát a lánykérő algoritmus fiú-optimális megoldást ad abban az esetben, ha a fiúk teszik az ajánlatot és lány optimálisat, ha a lányok teszik. □

Példa 1. Lássunk egy példát olyan páros gráfra, ahol nem csak fiú-optimális és lány-optimális stabil párosítás létezik:

Lányok	Preferencia lista	Fiúk	Preferencia lista
a_1	$[b_1, b_2, b_3]$	b_1	$[a_2, a_3, a_1]$
a_2	$[b_2, b_3, b_1]$	b_2	$[a_3, a_1, a_2]$
a_3	$[b_3, b_1, b_2]$	b_3	$[a_1, a_2, a_3]$

Ebben a példában könnyen ellenőrizhető, hogy stabil párosítást kapunk akkor, ha minden fiúnak a listáján szereplő első lány lesz a felesége (ekkor minden lány a legrosszabb fiúval van párban), ha minden minden lánynak a listáján szereplő első fiú lesz a férje (ekkor minden fiú a legrosszabb lánnyal van párban), és akkor is, ha mind a fiúk, mind a lányok a listájukban második helyen levőkkel alkotnak párt.

Megjegyzés 1.3. Minden stabil párosításban pontosan ugyanazok a fiúk és lányok találnak párt maguknak. Ez speciális esete a *vidéki kórházak* problémájának, amelyet Roth [10] dolgozott ki részletesen. Belátta, hogy amennyiben egy kórház nem tudja kitölteni jelentkezőkkel az összes kvótáját, akkor minden stabil kiosztásban nem csak, hogy ugyanannyi kvótát tölt ki, hanem a felvett személyek is ugyanazok lesznek.

1.2 Stabil b-párosítás páros gráfon

Természetes általánosítás, ha a páros gráfon a pontoknak megengedjük, hogy több párban is szerepeljenek, vagyis a pontokhoz kapacitásokat rendelünk. Abban az esetben, ha csak az egyik oldalon adunk 1-nél nagyobb kapacitásokat, akkor *egy-a-sokhoz* feladatról, ha mind a két oldalon kaphatnak 1-nél nagyobb kapacitást, akkor *sok-a-sokhoz* feladatról beszélünk. Egy e él jelen esetben akkor lesz dominálva (akkor lesz inferior) egy v pontban, melynek kapacitása $b(v)$, ha egy M b-párosításban létezik $b(v)$ darab párosítás-beli él, amely fedi v pontot, és mindegyik előlrébb van a v pont preferencia-listájában, mint az e él. Egy b-párosítás akkor lesz tehát stabil, ha mindegyik nem párosítás-beli él dominálva van az egyik végpontjában.

Az egy-a-sokhoz feladatra Gale és Shapley motivációja az egyetemi felvételi probléma volt. Itt ugyanis minden egyetemnek van egy bizonyos kvótája, és valamilyen szempont szerint rangsorolja a felvételizőket. A jelentkezőknek szintén adva van a listájuk, hogy milyen sorrendben szeretnének felvételt nyerni az egyes egyetemekre, és mindenki csak egy egyetemre nyerhet felvételt. A feladat megoldása teljesen hasonló, mint a fiúk és a lányok esetében. A lánykérő algoritmus úgy módosul, hogy az egyetemek most nem csak az első, hanem a kiszabott kvóta szerint tartják meg a jelentkezőket, és azokat utasítják vissza minden körben, akik kívül esnek a felvehető létszámon. Az algoritmus ebben az esetben is stabil b-párosításhoz vezet, és szintén optimális lesz a jelentkezők szempontjából. [1]

A sok-a-sokhoz feladat megoldására ugyanúgy használható a lánykérő algoritmus. Itt az ajánlattevők az első körben nem egy, hanem annyi partnernek tesznek ajánlatot, amennyi a kapacitásuk. Ezután minden körben annyi új ajánlatot tesznek, ahány visszautasítást kaptak az előzőben. Az algoritmus lépésszáma és bizonyítása megegyezik az alapesettel.

A stabil párosítások és a b-párosítások elmélete páros gráfon esetén igen terjedelmes. Dolgozatomban nem tekintem célnak ennek ismertetését, az érdeklődők Fleiner [6] munkájában találnak további részleteket. Itt megtalálhatóak a problémát leíró mélyebb modellek (monoton kiválasztási függvények, antiláncok az algebrai hálókban, matroid kernelek, lineáris programozás megoldásaként kapott politópok), kapcsolat más matematikai problémákkal (például a listás él-színezés páros gráfokon), és az ismert alkalmazási lehetőségek (játékelmélet, közgazdaságtan). Ízelítőként rámutatok a stabil párosítás létezésének egy általánosabb magyarázatára páros gráfokon, és kiemelek egy közismert alkalmazási lehetőséget.

Tekintsük a páros gráf élgráfját. Kőnig Dénes (1916-ban) megmutatta, hogy $\chi_e(G) = \Delta(G)$, vagyis a páros gráf élgráfja mindig kiszínezhető a gráfban lévő legnagyobb fokszám szerinti színnel, vagyis a páros gráf élgráfja *perfekt.* Maffray (1992-ben) belátta, hogy egy gráf élgráfja akkor és csakis akkor perfekt, ha az élek minden *normális irányítására* az élgráfban van *kernel.* Páros gráf esetén normális irányítást eredményez, ha az élgráf minden pontja között, az élek eredeti lineáris rendezése szerint irányítódnak meg az élek. Tehát, ha $e <_v f$ két él a gráfban, akkor a nekik megfelelő E és F pontokra $F \to E$ legyen az élgráf irányítása. A kernel létezése pedig annyit tesz, hogy létezik olyan független ponthalmaz az élgráfban, amelyre minden, kernelben nem szereplő külső pontból mutat él. Az élgráf kernelje páros gráf esetén egyértelműen megfeleltethető egy stabil párosításnak a gráfban. A fenti gondolatmenet tehát bizonyítását adja a stabil párosítás létezésének, vagyis az 1.1 Tételnek.

A gyakorlati alkalmazások közül talán a legismertebb a két-oldalú piacok elmélete. Itt két különböző érdekeltségi csoport, például vállalatok és munkások alkotják a két oldal szereplőit. Minden munkás egyéni preferenciával rendelkezik a munkahelyekkel szemben, amelyet több szempont motiválhat. Hasonlóképpen minden vállalat megfelelő munkaerőt szeretne adott lehetőségeihez mérten. Az egyensúlyi állapot elérése a párosítás stabilitását jelenti.

[1] A magyar felvételi pontok kiszámítása is ugyanezzel az algoritmussal működik. Gondot okozhat viszont, hogy egyrészt az egyetemek preferenciáiban lehetnek egyenlőségek, abban az esetben, ha két jelentkező azonos pontszámot ér el. Másrészt vannak olyan egyetemek, ahol lehet egyszerre két szakot választani, és olyan egyetemek is, ahol ez kötelező. Végül az állam is megszabhat bizonyos szakági kvótákat. Mindezek okán az algoritmus lefuttatása után ma is csak széleskörű egyeztetések után alakulnak ki a végső ponthatárok hazánkban.

2 Stabil párosítás gráfokon

A probléma felvetése már Gale és Shapley [7] klasszikus cikkében is szerepelt. Adtak is egy egyszerű példát arra nézve, hogy stabil párosítás nem páros gráfon már nem feltétlenül létezik. A problémát, ahogy azt később részleteiben megismerhetjük, bizonyos páratlan hosszú ciklusok okozzák.

Példa 2. A példában szereplő négy személy preferenciája olyan, hogy az első három közülük "körbe szereti egymást". E miatt semelyik kettő nem léphet stabil kapcsolatba egymással, mert a harmadik ezt nem engedi.

Személy	Preferencia lista
1	$[2,3,4]$
2	$[3,1,4]$
3	$[1,2,4]$
4	tetszőleges

A fejezet során először azt fogom bemutatni, hogy a modell szintjén egyszerű, gráfon dolgozó algoritmussal miként sikerült Irvingnek [8] polinomiális időben megoldást tatánia, majd Tan [12] hogyan módosította ezt az algoritmust, és adott pontos karakterizációt a feladatra. Bemutatom Tan és Hsueh [13] algoritmusát, amely dinamikusan kezeli a feladatot, és amelynek a kiterjesztésével a későbbiekben még foglalkozok. Végül ismertetem Scarf lemmáját, és az azon alapuló algoritmust, amely alapján nemrégiben merőben más úton is eljutottak a fenti karakterizációhoz, és amelynek a továbbgondolásával a stabil b-párosítás és a stabil allokáció problémája is kezelhetővé válik.

2.1 Gráf alapú algoritmusok

A következőkben ismertetésre kerülő két algoritmusban közös, hogy az általuk használt modell maga az adott gráf, és a pontok preferencia-listái. Az eljárások, és főként helyességüknek igazolása komplikált, de cserébe a megvalósuló algoritmusok gyorsak és hatékonyak.
Irving volt az első, akinek '85-ben sikerült olyan algoritmust alkotnia, amely polinom időben talál egy stabil párosítást, amennyiben a gráfban létezik ilyen, és *nemleges* választ ad, ha nem létezik.
Tan '90-ben bevezette a *stabil partíció* fogalmát, amellyel sikerült megadni a probléma pontos leírását, és Irving algoritmusából kiindulva nem csak egy lehetséges stabil párosítást talált meg az adott gráfon, hanem egy okot is, amely a nem-létezést igazolja.
Az alfejezet végén bemutatom Tan és Hsueh algoritmusát, amely némileg más módon jut el a fenti stabil partícióhoz. Ennek az algoritmusnak a kiterjesztését fogjuk a későbbiekben használni a stabil b-párosítás problémájának gráf alapú megoldásához.

2.1.1 Irving algoritmusa

Ebben az alfejezetben Irving [8] cikkét ismertetem.

A két évtizedig fennálló kérdést, – hogy vajon létezik-e polinomiális algoritmus, mely eldönti, hogy van-e stabil párosítás egy tetszőleges gráfban – '85-ben válaszolta meg Irving [8]. Az alábbiakban ismertetem algoritmusát, amely $O(n^2)$ időben talál egy teljes stabil párosítást a gráfban, vagy igazolja ha ilyen nem található benne.
Input: Egy gráf, adott preferencia-listákkal
Output: Vagy egy teljes stabil párosítás, vagy annak megállapítása, hogy nem létezik benne.
Működés: Az algoritmus a preferencia-listák redukcióján, – vagyis gráfbeli élek törlésén – alapul, és két fázisban jut el a megoldáshoz. Működését az alábbi példán szemléltetjük:

Példa 3. A példában szereplő hat személy preferenciája a következő:

Személy	Preferencia lista
1	$[4, 6, 2, 5, 3]$
2	$[6, 3, 5, 1, 4]$
3	$[4, 5, 1, 6, 2]$
4	$[2, 6, 5, 1, 3]$
5	$[4, 2, 3, 6, 1]$
6	$[5, 1, 4, 2, 3]$

1. fázis:
Az első redukciós fázis során a személyek sorban tesznek ajánlatokat (pl: első lépésben **1** tesz ajánlatot az általa legkedveltebbnek: **4**-nek), és ezután kitöröljük azokat a személyeket az ajánlat fogadójának listájáról, akiket kevésbé kedvel, mint az ajánlat tevőjét. (Tehát példánkban az első lépésben **4** listájáról töröljük az **1** után levő **3**-at.) Az algoritmus helyességének belátásához azt kell igazolnunk, hogy ekkor nem törölhetünk ki olyan élet, amely szerepelhet stabil párosításban.

Lemma 2.1. *Egy 1. fázisban kitörölt él nem lehet része semmilyen stabil párosításnak.*

Bizonyítás. Indirekt módon bizonyítunk. Legyen (b, x) az első olyan él, amit töröltünk az első fázisban, annak ellenére, hogy része lehetett volna egy M stabil párosításnak. Egy új történhetett meg, hogy b ajánlatot kapott egy a ponttól, amely előrébb volt a preferencia-listáján, mint x.

Személy	Preferencia lista
$.$	$[.,.,.,.,.]$
a	$[.,b,.,.,.]$
x	$[.,.,\underline{b},.,.]$
$.$	$[.,.,.,.,.]$
b	$[.,.,.,a,\underline{x}]$
$.$	$[.,.,.,.,.]$

Ekkor viszont (a,b) él blokkoló él volna M stabil párosításban, ugyanis a-nak nem lehet jobb partnere M-ben b-nél, mert ekkor (b,x) nem az első kitörlött ellenpélda lenne; és b is jobban kedveli a-t, mint x-et.[2]

□

Megjegyzés 2.2. Az élkitörlés során új stabil párosítás sem jöhet létre. Ez ugyanis azt jelentené, hogy az eredeti gráfban a kitörlött élek blokkoltak, ami viszont nem lehetséges, mert stabil párosításban b csak a-val, vagy a-nál jobb személyekkel alkothat párt a 2.1 Lemma szerint.

Ha az eredeti gráfot nézzük, akkor a listáról való törlés megfelel egy él elhagyásának, az ajánlattételt pedig jelezhetjük az él megirányításával az ajánlat fogadója felé. Az ajánlattétel sorrendje lényegében tetszőleges. Az algoritmus addig folytatódik míg vagy ki nem ürül valamely pont listája, – ekkor nincs teljes stabil párosítás a gráfban – vagy minden pontnak pontosan egy kimenő és egy bemenő éle nem lesz. (Ezek az élek adott pont redukált listájának elején illetve végén találhatóak, így az eredeti gráfból, irányított diszjunk irányított körök és oda-vissza irányított élek, illetve néhány irányítatlan él marad.) Mivel az algoritmusban a fenti lemma miatt nem ismétlődhet meg egy ajánlattétel sem, ezért az algoritmus futásának idejét az élek száma felülről korlátozza.

Következmény 2.3. Az 1. fázis futásideje $O(n^2)$.

A fenti példának a következő lefutása lesz:

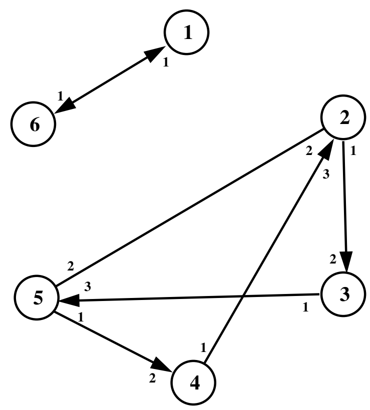
1 ajánlatot tesz 4-nek;	4 törli 3-at
2 ajánlatot tesz 6-nak;	6 törli 3-at
3 ajánlatot tesz 5-nek;	5 törli 6-ot és 1-et
4 ajánlatot tesz 2-nek;	-
5 ajánlatot tesz 4-nek;	4 törli 1-et
6 ajánlatot tesz 1-nek;	6 törli 2-t és 3-at
1 ajánlatot tesz 6-nak;	1 törli 4-et és 2-t
2 ajánlatot tesz 3-nak;	-
3 ajánlatot tesz 5-nek;	-

A redukált preferencia-listák a következők lesznek:

[2] Az eredeti bizonyításban csak akkor volt törlés az ajánlattevő listáján, ha a fogadó visszautasította az ajánlatot, mert már volt neki kedvezőbb. A listák korrekciója, szimmetrikussá tétele csak ezután következett. A bizonyítás menete lényegében megegyezik az eredetivel.

Személy	Preferencia lista
1	[6]
2	[3, 5, 4]
3	[5, 2]
4	[2, 5]
5	[4, 2, 3]
6	[1]

A redukált gráf pedig:



Ábra 1: Redukált gráf

Egy stabil párosításban szereplő élek, tehát szerepelnek a redukált listában minden pont esetén. Ha van olyan pontpár, akik kölcsönösen egymaguk szerepelnek a másik listáján, akkor ők biztosan párok lesznek, amennyiben létezik stabil párosítás. Ha ez minden pontpárra teljesülne, tehát minden lista 1 hosszúra redukálódna az 1. fázis során, akkor az egy, és egyben egyetlen teljes stabil párosítását eredményezné a gráfnak. Ha viszont nem ürült ki egy pontunk listája sem, vagy nem jutottunk el az előbbi állapotba, akkor folytatnunk kell az algoritmust.

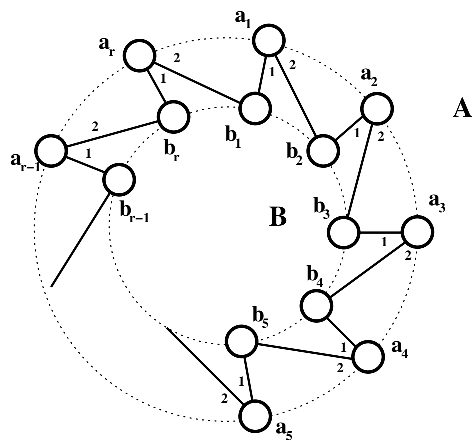
2. fázis:
A második fázis során olyan éleket hagyunk el, melyek szerepelnek stabil párosításban. Megmutatjuk, hogy ha így elveszítünk stabil párosítást, marad legalább egy másik stabil párosítás a gráfban. A törlést úgynevezett *rotációk* mentén végezzük.

Definíció 2.4. Egy $R = (a_1, a_2, ..., a_r)|(b_1, b_2, ..., b_r) = R(A|B)$ ciklus-párt *rotációnak* nevezünk, ha a ciklusokon belül minden elem különböző, és teljesül rájuk a következő feltétel: a_i redukált listáján az első elem b_i, a második pedig

b_{i+1} (mod r).

Személy	Preferencia lista
a_1	b_1 , b_2 , ...
a_2	b_2 , b_3 , ...
...	...
a_i	b_i , b_{i+1} , ...
...	...
a_r	b_r , b_1 , ...

A rotáció a következőképpen nézhet ki:



Ábra 2: Rotáció

Rotáció keresése a következőképpen történik: veszünk egy p_1 pontot, melynek listájában legalább két elem van. Rotációt keresni pontosan akkor lehet, ha létezik ilyen pont. Legyen p_1 listáján szereplő második elem q_2. Majd q_2 listáján szereplő utolsó elem legyen p_2, és így tovább. Ekkor igaz, hogy p_i listáján szereplő első elem q_i és második elem q_{i+1} minden i-re. Ezt az eljárást addig folytatjuk, míg $p_i = p_j$ valamely $i \neq j$-re. Végül

$$R = (A|B) = (a_1, a_2, ..., a_r)|(b_1, b_2, ..., b_r) := (p_i, p_{i+1}, ..., p_{j-1})|(q_i, q_{i+1}, ..., b_{j-1})$$

Fontos megemlíteni, hogy találhatunk olyan rotációt, melyet alkotó két ciklusnak van közös eleme, de ekkor a rotáció különleges tulajdonsággokkal rendelkezik, melyet később részletezünk.

Lemma 2.5. *Legyen $R = (A|B)$ egy rotáció, ekkor*

1. *tetszőleges stabil párosításra vagy az összes (a_i, b_i) élt tartalmazza vagy egyiket sem.*

2. ha (a_i, b_i) éle egy stabil párosításnak, akkor van olyan stabil párosítás is, melynek nem éle.

Bizonyítás.

1. Ha a_i párja b_i-nek egy stabil párosításban, akkor a_{i-1} is párja kell legyen b_{i-1}-nek. Ugyanis egyrészt (b_i, a_{i-1}) inferior él, mert b_i listáján a_i utolsó, másrészt, ahhoz hogy b_{i-1} részéről ez az él ne legyen blokkoló, b_{i-1}-nek jobbat kell választania a_i-nél, aki egyedül csak a listáján első, a_{i-1} lehet.

2. Először is meg kell jegyeznünk, hogy ebben az esetben A és B ciklusnak nem lehet közös eleme. Ha ugyanis (a_i, b_i) élek részei egy stabil párosításnak, és feltesszük, hogy $a_j = b_k$ akkor a_j mind a listáján szereplő első, mind pedig utolsó ponttal párosítva lenne.
   Vegyük tehát a két ciklust és lássuk be, hogy ha (a_1, b_1), (a_2, b_2), ... ,(a_r, b_r) élek részei egy stabil párosításnak, és ezeket kicseréljük (a_1, b_2), (a_2, b_3), ... ,(a_r, b_1) élekre, akkor szintén egy stabil párosítást kapunk, amit jelöljünk M'-vel. ekkor ugyanis M-hez képest csak a_i-k kerültek rosszabb partnerhez M'-ben, mégpedig a listáikon első helyen szereplők helyett a másodikok lettek a párjaik. Az (a_i, b_i) élek tehát superior élekké váltak az a_i-k részéről, viszont b_i-k részéről ezek inferior élek, mert a_i-k a redukált listáik végén találhatóak. Más superior él pedig nem keletkezett a gráfban, tehát M' párosítás is stabil marad. □

Következmény 2.6. Ha volt a gráfban stabil párosítás, akkor az $R(A|B)$ rotáció *küllöinek*, vagyis (a_i, b_i) éleinek törlésével kapott redukált gráfban is marad legalább egy stabil párosítás.

A rotáció küllőinek törlése után vagy újra lefuttatjuk az első fázist, vagy automatikusan kitöröljük b_i listákon az a_{i-1}-nél rosszabb éleket.[3]

Megjegyzés 2.7. Új stabil párosítás nem jöhet létre a 2. fázis törlései során. Ha ugyanis b_i-k lehetséges legrosszabb partnere a_{i-1} (tehát nálánál jobb elemmel nem lehet párban egy stabil párosításban sem), akkor az ennél rosszabb élek hasonlóképpen nem blokkolhatnak, mint az 1. fázis során.

A 2. fázis szerinti redukciót addig folytatjuk, míg ki nem ürül valamelyik pont listája, – ekkor nincs teljes stabil párosítás – avagy nem lesz minden lista egy hosszú, amely egyben egy lehetséges megoldását jelenti a feladatnak.

Megjegyzés 2.8. A második fázis lépésszáma szintén $O(n^2)$, ezért az első fázis lépésszámára vonatkozó megjegyzéssel együtt az egész algoritmusra fennáll ez a korlát.

[3] Az első változat Irving-é [8], míg a második Tan [12] módosítása.

Lássuk végül a példánkon, hogyan működik a redukció 2. fázisa:

Személy	Preferencia lista
3	$[5, 2]$
4	$[2, 5]$

Az első megtalált rotáció szerint töröltjük a $(3, 5)$ és $(4, 2)$ küllőket, ezután automatikusan elhagyjuk **2** listájából a $(2, 3)$-nál rosszabb $(2, 5)$ élet. Ezzel máris megkapjuk a megoldást: az $(1, 6), (2, 3), (4, 5)$ élek példánkban teljes stabil párosítást adnak.

Megjegyzés 2.9. Irving algoritmusa eredetileg $2n$ pontú teljes gráfon keresett teljes stabil párosítást. Ezt két módon is általtánosíthatjuk: egyrészt lehet a gráfunk ritkább, vagy páratlan pontú, másrészt a keresett stabil párosítás sem kell, hogy teljes legyen.
Ha az adott listák nem is szimmetrikusak (tehát $(a, b) \in E(G)$, de $(b, a) \notin E(G)$), akkor az algoritmust ezen elemek törlésével kell kezdenünk. A kapott nem-teljes gráfra az algoritmus már az eredeti formájában működik.
Nem teljes stabil párosítás esetén már van értelme páratlan pontszámú gráfon is keresnünk. Itt egy lista kiürülése nem jelent azonnal negatív választ, (kivéve teljes páros pontszámú gráfok esetén, ahol nem maradhat két izolált pont). A megállási kritérium pontos megadása a következő rész ismeretében egyszerűen meghatározhatóvá fog válni.

2.1.2 Tan leírása: a stabil partíció

Tan első ezirányú cikkében [12] Irving algoritmusából kiindulva talált egy pontos karakterizációt a feladatra, és kiterjesztette Irving algoritmusát úgy, hogy ezt meg is találja. Majd nem sokkal később Hsueh-val közösen [13] megalkottak egy új algoritmust is, amely polinom időben, de más úton vezet el a pontos megoldáshoz. Az alfejezetben Tan [12] cikke alapján bemutatom a karakterizációt, és Irving algoritmusának módosítását, amely megoldja az új problémát.

Definíció 2.10. A leírás alapja a *stabil partíció* fogalma. A gráf pontjait ennek során részekre bontjuk: egy elemű *izolált pontok*, két elemű *párok*, és legalább három pontból álló *ciklusok* halmazára. Egy $A =< a_1, a_2, ..., a_r >$ *ciklikusra* annak kell teljesülnie, hogy minden pontja jobban kedveli a körben előtte álló szomszédját, mint az utána következőt, tehát $a_{i-1} <_{a_i} a_{i+1}$ $\forall$ i-re modulo r. A párokat és a ciklusokat alkotó éleket *rész-éleknek* nevezzük. Egy Π partíció akkor lesz *stabil*, ha a rész-élek dominálják a gráfban szereplő összes élt.

Tan ezt úgynevezett superior és inferior élek segítségével fogalmazta meg. Az inferior élek egy pont szemszögéből lényegében a pontban rész-él által dominált éleket jelölik, míg az superior élek azokat, akik nincsenek dominálva. Pontosabban:

1. izolált pont esetén minden belőle kiinduló él superior él.
2. párban lévő pontra a párjánál jobb pontokhoz superior, a párjánál rosszabbakhoz pedig inferior él megy.
3. Egy ciklus esetén a_i szemszögéből superior élek futnak az a_{i-1} pontnál kedveltebb pontokhoz, inferior élek az a_{i-1}-nél rosszabb élekhez.

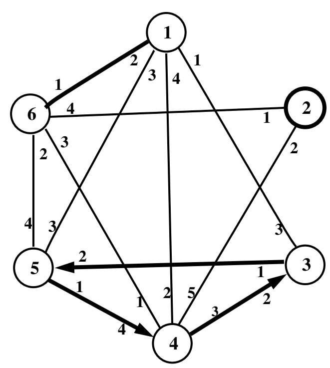
Ebben a megfogalmazásban a stabilitás feltétele az, hogy ha egy él superior él az egyik végpontja szerint, akkor a másik végpontja szerint inferior él kell, hogy legyen.

Lássunk egy példát stabil partícióra gráf esetén:

Példa 4. A példában szereplő hat személy preferenciája a következő:

Személy	Preferencia lista
1	$[3, 6, 5, 4]$
2	$[6, 4]$
3	$[5, 4, 1]$
4	$[6, 1, 3, 5, 2]$
5	$[4, 3, 1, 6]$
6	$[1, 5, 4, 2]$

Feladatunk a stabil partíció megtalálása:

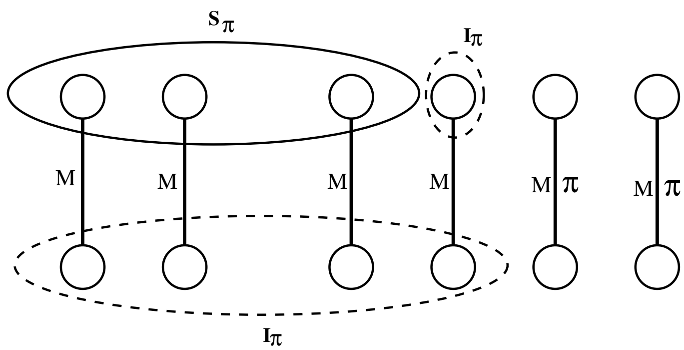


Ábra 3: Stabil partíció: $< 2 >, < 1, 6 >, < 3, 4, 5 >$

Azonnal adódik, hogy ha a stabil partíció csak párokból áll, akkor ezek az élek a gráf egy teljes stabil párosítását adják, és ez megfordítva is igaz. Továbbá, ha a stabil partíció nem tartalmaz páratlan részeket – tehát kizárólag párokat és páros hosszú ciklusokat – akkor a páros hosszú ciklusok minden második élét kivéve, a maradék párok szintén egy teljes stabil párosítását adják a gráfnak. Gondot tehát csak a páratlan részek jelenthetnek.

Tétel 2.11. *Ha egy adott gráfra és preferencia-listákra létezik olyan stabil partíció, amely tartalmaz páratlan részeket, akkor nem létezik teljes stabil párosítás a gráfban.*

Bizonyítás. Tegyük fel, hogy mégis létezik egy M teljes stabil párosítás, annak ellenére, hogy Π stabil partíció tartalmaz páratlan részt. Nevezzük el S_Π-nek a pontok azon halmazát, akiknek az M-beli párjukhoz Π-ben superior él vezet. Hasonlóképpen definiáljuk az I_Π ponthalmazt is.



Ábra 4: S_Π és I_Π ponthalmazok M szerint nézve

Ha az M-beli párokat nézzük, akkor nyilvánvaló, hogy egy párt alkotó két pont közül legfeljebb csak az egyik lehet S_Π-beli, mert különben sérülne a partíció stabilitása. Így az M-beli párokra összegezve: $|S_\Pi| \leq |I_\Pi|$
Másrészt, ha a Π partíció szerint számolunk, akkor a részeire igazak a következő állítások:

1. Egy v izolált pont csak S_Π-beli lehet. Belőle ugyanis csak superior élek indultak ki.

2. Egy $< u, v >$ pár mindkét eleme nem lehet I_Π-beli. Mert, ha a Π-beli pár mindkét vége kölcsönösen jobban tetszene egymásnak, mint az M-beli párja, akkor ezzel sérülne M stabilitása.

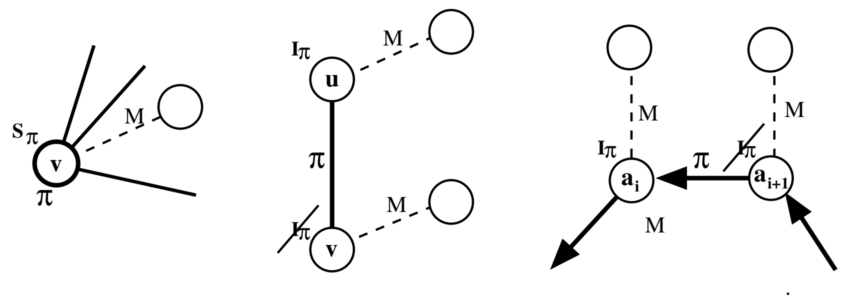
3. Egy ciklus két egymás követő eleme, a_i és a_{i+1} közül nem lehet mindkettő I_Π-beli. Ez hasonlóan az előzőekhez ellentmondana M stabilitásával, mert az (a_i, a_{i+1}) rész-él mindkét pont részéről dominálná az M-beli párjukat.

Ebből követezik, hogy ha A egy páratlan rész a Π partícióban, akkor

$$|A \cap S_\Pi| > |A \cap I_\Pi|$$

ha páros, akkor

$$|A \cap S_\Pi| \geq |A \cap I_\Pi|$$



Ábra 5: S_Π és I_Π ponthalmazok Π szerint nézve

Tehát összegezve minden A_i részre:

$$|S_\Pi| = \sum |A_i \cap S_\Pi| > \sum |A_i \cap I_\Pi| = |I_\Pi|$$

Ellentmondásra jutottunk. □

A következő tétel szerint igaz a fenti tétel megfordítása is. Ha nem létezik teljes stabil párosítás, akkor a stabil partícióban egyértelműen megtaláljuk a nemkívánatos páratlan részeket.

Tétel 2.12. *Minden adott gráfra és preferencia-listákra létezik legalább egy stabil partíció, és minden stabil partíció pontosan ugyanazokat a páratlan részeket tartalmazza.*

Megjegyzés 2.13. Ez az egyértelműségi tétel általánosítja a páros gráfok esetében tett 1.3 Megjegyzést. Nevezetesen, nem csak a páratlan ciklusok, hanem az izolált pontok is egyértelműek, tehát minden stabil partícióban ugyanazok a pontok találnak párt maguknak.

Tan bizonyításában Irving algoritmusának módosításával, konstruktív úton jut el a keresett stabil partícióhoz. Hasonlóképpen két fázisban törli ki az éleket, és egyrészt ügyel arra, hogy legalább egy stabil partíció megmaradjon (új ne keletkezzen), és a páratlan részek el legyenek különítve. A bizonyításnak csak néhány alapgondolatát írom le az alábbiakban.

Bizonyítandó, hogy pontosan akkor találunk rá egy páratlan hosszú ciklusra, ha a 2. fázisban egy rotáció törlése során az egyik pont két elemű listája üressé válik; ekkor a törölt $R(A|B)$ rotációban $A = B$. Ha ez nem következik be, akkor két eset lehetséges. Lehet, hogy a rotáció küllő közül senki sem szerepelt stabil partícióban, ekkor nyugodtan törölhetjük őket. Ha viszont egy él is része egy stabil partíciónak, akkor az összes küllő rész-él lesz, és $A \cap B = \emptyset$. Ebben az esetben vagy ciklikusan helyettesíthető párokat hagyunk el, ahogy Irving tette, vagy egy páros hosszú ciklus minden második élét.
Tan algoritmusa akkor fejeződik be, ha az úgynevezett *aktív* listák elfogynak.

Egy lista *inaktívvá* válik, ha egy rotáció törlése során az előbb leírtak szerint része volt egy páratlan ciklusnak, vagy ha egyéb éltörlések során kiürült a listája – ekkor izolált lesz a pont –, illetve ha egy elem marad a listájában, ekkor egy pár eleme lesz a stabil partícióban.

A bizonyítás azért nem ismertetem részleteiben, mert egyrészt nagyon hasonló Irving algoritmusának bizonyításához, – hiszen a gondolatmenet az ő algoritmusán alapszik – másrészt pedig Tan által megadott pontos karakterizációhoz a következőkben el fogunk jutni a Tan-Hsueh algoritmussal, és Scarf lemmáján keresztül is. Annak viszont, hogy mégis megemlítettem létét az az oka, hogy ennek az algoritmusnak a kibővítésével talán épp úgy el lehet jutni a stabil b-partíció megoldásához, mint ahogy az Irving algoritmus kiterjesztésével nemrégiben eljutottak a b-párosítás gráf alapú megoldásához (lásd Cechlárová és Fleiner [3]). Ez pedig hatékonyságában megelőzheti az általam javasolt,– a későbbiekben részletezésre kerülő – Tan-Hsueh algoritmus kiterjesztésével kapott változatot.

2.1.3 Tan-Hsueh algoritmus

Az alfejezetben Tan-Hsueh [13] cikkét foglalom össze. A bizonyítás gondolatmenetén nem változtattam, csak kis tömörítést végeztem el rajta. A 2.12 Tétel bizonyításában egy saját ötlet is szerepel.

Legyen adva egy n pontú G gráf és a pontok preferencia-listái. Célunk egy stabil partíció megtalálása a gráfban. A Tan-Hsueh algoritmus dinamikusan dolgozik, tehát a pontokat egyenként hozzáadva n lépésben keresi meg a stabil partíciót. Az algoritmus bizonyítása során pontok "letakarásával" és "hozzávételével", vagyis a pontból kiinduló élek és a listákban szereplő elemek törlésével illetve hozzáadásával átmenetileg stabil partíciókon keresztül jutunk el a megoldáshoz. Ezt a műveletet jelöljük G gráf és v pont esetén $G-v$ illetve $G+v$ módon. Ha Π egy stabil partíció volt a G gráfban, akkor jelöljük Π_v-vel a stabil partíciót $G-v$ részgráfon.

Tegyünk fel tehát, hogy k pont esetén G^k részgráfra már találtunk egy stabil partíciót, Π^k-t, és a következőkben azt fogjuk megmutatni, hogy egy pontot hozzávéve a k+1 pontú $G^{k+1} = G^k + v_{k+1}$ gráfra az algoritmus talál egy másik Π^{k+1} stabil partíciót. Az algoritmus $k+1$. lépése tehát a következő eredményt hozza:
Input: G^k gráf Π^k stabil partícióval – ahol minden ciklus páratlan – és v_{k+1} pont adott preferenciákkal.
Output: Π^{k+1} stabil partíció G^{k+1} gráfon, melyben minden ciklus páratlan.

Az algoritmus $k+1$-edik lépése úgy kezdődik, hogy az új pont, – aki aktuálisan kimarad a stabil partícióból, ezért nevezzük el $act := v_{k+1}$-nek – ajánlatot tesz mindenkinek sorban a preferencia-listája szerint. Milyen esetek lehetségesek?

1. Ha mindenki visszautasítja, akkor *act* izolált pontként hozzávehető az eredeti partícióhoz, vagyis $\Pi^{k+1} = \Pi^k \cup \{< act >\}$ stabil partíció lesz. Az *act* pontból kiinduló superior élekkel szemben csupa inferior él áll.

2. Ha van olyan pont, aki elfogadja *act* ajánlatát, akkor a legelső közülük legyen **x**. Aszerint, hogy x pont milyen részben volt a Π^k partícióban a következő esetek lehetségesek:

   (a) izolált pont volt: $< x > \in \Pi^k$. Ebben az esetben *act* az x-el párt alkotva lesz része az új stabil partíciónak:

   $$\Pi^{k+1} = \Pi^k \setminus \{< x >\} \cup \{< act, x >\}$$

   Mivel csak x pont néhány superior éle változott inferiorrá, ezért a stabilitás igazolt.

   (b) része volt egy $A =< a_1, a_2, \ldots, a_{2m}, x > \in \Pi$ páratlan ciklusnak. Ekkor *act* az x-el ismét párt alkot, és a ciklus maradék része értelemszerűen párokba rendeződik, éppen úgy, ahogy azt a páros ciklus párokra osztásakor láttuk, vagyis:

   $$\Pi^{k+1} = \Pi^k \setminus A \cup \{< act, x >, < a_1, a_2 >, \ldots, < a_{2m-1}, a_{2m} >\}$$

   Az eredeti gráfban most sem keletkezik új superior él.

   (c) párban volt: $< x, y > \in \Pi^k$. Ekkor *act* elcsábítja x-et, és **y** egyedül marad. Ebben az esetben

   $$\Pi^k_y := \Pi^k \setminus \{< x, y >\} \cup \{< act, x >\}$$

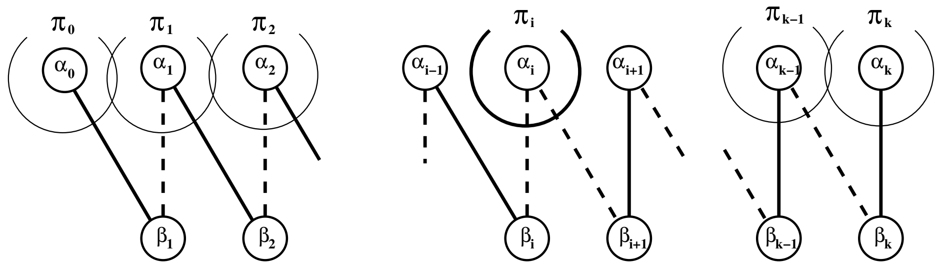
   stabil partíció a $G^{k+1} - y$ gráfon.

Az algoritmus a fentiekből adódik. Az algoritmus egy lépése csak abban az esetben nem ér véget rögtön, ha az új pont ajánlatát egy párban lévő pont fogadja el elsőként. Ekkor viszont az egyedül maradt pont, esetünkben y lesz a párkereső, vagyis $act := y$. Ezen az úton folytatva az algoritmust, ismételten stabil partícióhoz jutunk akkor, ha vagy senki sem fogadja el az *act* pont ajánlatát, vagy az elfogadó pont egy páratlan részbe tartozott.

Az egyetlen eset, amit végig kell nézni az, ha ismétlődés folytán mindig párban lévő pontok fogadják el az ajánlatot. A következőkben azt fogjuk belátni, hogy ekkor az ismétlődő pontokból egy páratlan ciklust tudunk elkülöníteni. A továbbiakban legyen $G := G^{k+1}$, és a keresett partíció, $\Pi := \Pi^{k+1}$.

Definíció 2.14. Adott egy $\Pi = \Pi_0$ stabil partíció $G - \alpha_0$ gráfon. Ekkor a pontok egy $\alpha_0, \beta_1, \alpha_1, \beta_2, \alpha_2, \ldots, \beta_k, \alpha_k$ sorozatát *alternáló sorozatnak* nevezzük, amennyiben $G - \alpha_i$-hez tartozó stabil partíciók Π_i sorozatára $i = 1, 2, \ldots, k$ teljesül, hogy

$$\Pi_{i+1} = \Pi_i - \{< \beta_{i+1}, \alpha_{i+1} >\} \cup \{< \alpha_i, \beta_{i+1} >\}$$



Ábra 6: Alternáló sorozat

Alternáló sorozat pontosan úgy jön létre az algoritmus során, hogy az i-edik lépésben az ajánlatot elfogadó első pont, β_i egy pár része volt Π_{i-1} stabil partícióban, és ezért az elhagyott párja, α_i lesz a következő ajánlattevő pont. Az alternáló sorozat egy triviálisan belátható tulajdonságát rögtön szögezzünk le:

Állítás 2.15. *Minden $\alpha_0, \beta_1, \alpha_1, \ldots, \beta_k, \alpha_k$ alternáló sorozatra és a hozzá tartozó $\Pi_0, \Pi_1, \ldots, \Pi_k$ stabil partíciókra igaz, hogy*

1. *α_i-nek rosszabb partnere lesz Π_{i+1}-ben, mint Π_{i-1}-ben volt, vagyis ekkor $\beta_i <_{\alpha_i} \beta_{i+1}$ $\forall$ i-re.*
2. *β_i-nek jobb partnere lesz Π_i-ben, mint Π_{i-1}-ben volt, vagyis $\alpha_{i-1} <_{\beta_i} \alpha_i$ $\forall$ i-re.*

Egy pont helyzete többször javulhat, illetve romolhat, tehát létezhet $i \neq j$, hogy $\alpha_i = \alpha_j$ vagy $\beta_i = \beta_j$. Azonban a listák, és a pontok száma véges, ezért nem fordulhat elő, hogy minden pont helyzete vagy csak javuljon, vagy csak romoljon, tehát a véget nem érő alternáló sorozatban léteznie kell $i \neq j$ sorszámnak, melyre $\alpha_i = \beta_j$. Az ilyen alternáló sorozatot nevezzük *visszatérő sorozatnak.*

Alternáló sorozat esetén a továbbiakban mindig az első visszatérésnél $\alpha_i = \beta_j$-nél kezdjük el a vizsgálódást, ahol feltesszük, hogy $i < j$. A sorozat elejétől eltekintünk, tehát $\alpha_0 := \alpha_i$ lesz a visszatérő sorozat első eleme, és egyben ezen visszatérő elemek közt az utolsó előfordulás, vagyis $\alpha_k \neq \beta_j \; \forall \; k = i+1, \ldots, j-1$. Másik fontos megjegyzés, hogy az $i < j$ feltételezéssel azért élhettünk, mert ha létezik $i > j$, hogy $\alpha_i = \beta_j$ legelső visszatérés, akkor $\alpha_{j-1} = \beta_i$ is teljesül. Ugyanis β_j a j. lépésben α_{j-1}-nek lett párja, így amikor az $i+1$. lépésben visszatér az α-k között, akkor ez azt jelenti, hogy őt az addigi párja, β_i hagyta el. Mindezek következménye az alábbi állítás:

Állítás 2.16. *Legyen $\alpha_0, \beta_1, \alpha_1, \ldots, \beta_k, \alpha_k, \beta_{k+1}$ az első visszatérő sorozat, ahol $\alpha_0 := \beta_{k+1}$ és $\alpha_i \neq \beta_j \; \forall \; i < j$-re. Ekkor $\alpha_i \neq \beta_j$ minden i-re és j-re, tehát a visszatérő sorozat minden eleme különböző.*

A következő tétel már tartalmazza mindazt, amire az algoritmus helyes működésének igazolásához szükségünk van. Bizonyításához viszont még több állítást be kell látnunk.

Tétel 2.17. *Legyen $\alpha_0, \beta_1, \alpha_1, \ldots, \beta_k, \alpha_k$ egy alternáló sorozat, ahol a β_{k+1}-nél visszatérés van. Ekkor az alternáló sorozat kiterjeszthető úgy, hogy β_{k+m+1}-nél a sorozat visszatér α_k-hoz és ezen második visszatérő sorozatra a két alábbi állítás is igaz:*

1. *$\alpha_k, \beta_{k+1}, \alpha_{k+1}, \ldots, \beta_{k+m}, \alpha_{k+m}$ $2m+1$ darab különböző pont.*

2. *$A =< \alpha_{k+m}, \beta_{k+m}, \alpha_{k+m-1}, \beta_{k+m-1}, \ldots, \alpha_{k+1}, \beta_{k+1}, \alpha_k >$ egy páratlan ciklust alkot, vagyis $\Pi = \Pi_k \setminus \{< \beta_i, \alpha_i > | i = k+1 \ldots k+m\} \cup \{A\}$ stabil partíció az egész G gráfra.*

Vizsgáljuk meg, hogyan jön létre az alternáló sorozat, és mi történik az első visszatérés után! Míg Π_0 stabil partíciótól eljutunk Π_{k+1} stabil partícióig, addig az α_i pontok helyzete folyamatosan romlik, és a β_j pontok helyzete pedig mindig javul. Ezt úgy is megfogalmazhatjuk, hogy α_i pontok helyzete Π_0-ban, β_j pontok helyzete pedig Π_{k+1}-ban volt a legjobb az első visszatérésig.

Állítás 2.18. *Legyen $\alpha_0, \beta_1, \alpha_1, \ldots, \beta_k, \alpha_k, \beta_{k+1}$ visszatérő sorozat ($\alpha_0 = \beta_{k+1}$) és $\Pi_0, \Pi_1, \ldots, \Pi_k, \Pi_{k+1}$ a hozzá tartozó stabil partíciók. Ekkor*

1. *$\alpha_{k+1} = \beta_1$ továbbá*

2. *$\beta_{k+2} = \alpha_i$ valamely $1 \le i \le k$-ra.*

Bizonyítás.

1. Mivel egy visszatérő sorozatban α_0 egyik oldalon sem jelenik meg újra, csak β_{k+1}-ben, ez azt jelenti, hogy az 1. lépésben szerzett párja (α_1) mindvégig a párja is marad, ezért a $k+1$. lépésben $\alpha_0 = \beta_{k+1}$ őt hagyja el, így $\alpha_{k+1} = \beta_1$.

2. Az 1. lépésben β_1 elhagyta α_1-et, akinek a továbbiakban a helyzete csak romolhatott, ezért amikor $\beta_1 (= \alpha_{k+1})$ ismét párt keres magának a $k+2$. lépésben, akkor α_1 biztosan azok között lesz, akik szívesen párba lépnének vele. De vajon találhat-e jobb párt magának $\beta_1 (= \alpha_{k+1})$? Ha igen, akkor csak azon pontok közül, akiknek helyzete Π_0-hoz képest romlott, tehát az α_i-k közül valamely $1 \le i \le k$-ra, mert más esetben sérülne Π_0 partíció stabilitása. □

A fenti állításból két dolgot is kiolvashatunk. Egyrészt megállapíthatjuk, hogy $\beta_1 (= \alpha_{k+1})$ helyzete nem lett rosszabb, mint amilyen eredetileg Π_0-ban volt, hiszen legrosszabb esetben akkori párját, α_1-et kapta újra meg párnak; és persze $\beta_{k+1} = \alpha_i$ jól járt a cserével. Másrészt konstatálhatjuk, hogy a $k+2$. lépésben $\alpha_i = \beta_{k+1}$ miatt egy újabb, az előzőnél biztosan nem hosszabb visszatérő sorozathoz jutottunk. Ezen gondolatok alapján kiterjesztéssel igazolhatjuk az alábbiakat:

Következmény 2.19. Minden alternáló sorozatra teljesül, hogy az első visszatérés után

1. minden lépésben visszatér a sorozat, ezért nem fejeződik be soha. Új elemek nem kerülhetnek be, és a visszatérések távolsága sem nő.

2. Egyik pontnak sem lesz már annál rosszabb partnere, mint amilyen az első ciklusban volt a számára legrosszabb pár. (α_i-k esetén β_{i+1}-ek, míg β_j-k esetében α_i-k közül kerül ki ez a legrosszabb pár)

Állítás 2.20. *Semelyik β_i és β_j, ahol $1 \leq i, j \leq k$, nem lehet párban egymással az algoritmus során.*

Bizonyítás. Az első visszatérésig semmiképpen nem lehettek párok, továbbá az is igaz, hogy a visszatérés után nem kaphattak rosszabb partnert, mint a Π_0-beli párjukat: α_i-t és α_j-t. Tehát ha később mégis párba kerülnének, akkor ez azt jelentené, hogy kölcsönösen jobban kedvelik egymást, mint Π_0-beli párjukat, ami viszont ellentmond Π_0 partíció stabilitásának. $\square$

Bizonyítás. (Tétel 2.17)

1. Lássuk először azt be, hogy az első visszatérés után α_k-tól kezdődően is indul visszatérő sorozat. Tegyük fel az ellenkezőjét, és legyenek $s < k < t$ azon k-hoz legközelebb álló indexek, melyekre az α_s-ből induló sorozat visszatér β_q-ba, és az α_t-ből induló pedig β_{q+1}-be, míg α_k-nak nincs visszatérése. Ekkor, a 2.18 Állítás első részéből következik, hogy $\alpha_q = \beta_{s+1}$ továbbá az is, hogy α_t, – mivel β_t már biztosan visszatérés volt, – megegyezik egy β_j-vel, ahol $1 \leq j \leq k$. Tehát mind α_q mind pedig β_{q+1} szerepelt már az első visszatérést megelőzően a β-k közt. Az, hogy ennek ellenére párok a Π_{q+1} stabil partícióban, ellentmond az előző állításnak.

   Az pedig, hogy az első visszatérést követően minden pontból indul és minden pontba érkezik visszatérő sorozat, és ezek hossza nem nő, pontosan azt jelenti, hogy az α_k-tól induló visszatérő sorozat már ciklikusan ismétlődik a folytatásban és az egy ciklusban lévő elemek különbözőek.

2. Ahhoz, hogy belássuk, hogy $\Pi = \Pi_k \setminus \{<\beta_i, \alpha_i> | i = k+1 \ldots k+m\} \cup \{A\}$ stabil partíció, $A = <\alpha_{k+m}, \beta_{k+m}, \alpha_{k+m-1}, \beta_{k+m-1}, \ldots, \alpha_{k+1}, \beta_{k+1}, \alpha_k>$ páratlan ciklus, azt kell igazolnunk, hogy ebben a partícióban nem áll két superior él egymással szemben.

   Tegyük fel az ellenkezőjét: létezik olyan x pont, amely a ciklus egyik pontjával, például α_k-val egymás irányában kölcsönösen superior élek. Először azt látjuk be, hogy x nem lehet a cikluson kívüli elem. (α_k, x) él superior volta Π-ben pontosan azt jelenti, hogy $x <_{\alpha_k} \beta_{k+1}$. Ez viszont fennáll például Π_{k+1} partícióra is, ahol α_k a β_{k+1} párja, vagyis Π_{k+1} partíció sem stabil. Ezek után már csak azt a két esetet kell megnézni, ha

$x = \alpha_i$ vagy $x = \beta_j$ valamely ciklusbeli elemre.

Ha $x = \beta_j$ volna, akkor tekintsük Π_{k+1} partíciót. Ebben (α_k, β_{k+1}), illetve (α_i, β_i) pár minden $k+2 \leq i \leq k+m$-re. Ekkor (α_k, β_{k+1}) superior él Π-ben pontosan akkor, ha superior Π_1-ben is. Ez viszont az jelenti, hogy (β_{k+1}, α_k) inferior él Π_1-ben, ami pontosan akkor áll fent, ha inferior Π-ben is. Ez ellentmondás.

Ha $x = \alpha_i$ volna, akkor tekintsük Π_{k+m+1} partíciót. Ebben (α_i, β_{i+1}) pár minden $k \leq i \leq k+m$-re. Ekkor (α_k, α_i) superior él Π-ben pontosan akkor, ha superior Π_{k+m+1}-ben is. Ez viszont az jelenti, hogy (α_i, α_k) inferior él Π_{k+m+1}-ben, ami pontosan akkor áll fent, ha inferior Π-ben is. Ez ismét ellentmondáshoz vezet. □

Az algoritmus működése ezek után már érthető: a kérdéses esetben az alternáló sorozat első visszatérése utáni ciklus beilleszthető az addigi stabil partícióba.

Az algoritmus futásideje egy lépésre $O(n^2)$, ezért a teljes lépésszám $O(n^3)$.

Állítás 2.21. *Egy gráf, amelynek stabil partíciója k darab páratlan részt tartalmaz, pontosan k darab pont elvételével, vagy hozzáadásával olyan gráffá alakítható, amelynek létezik teljes stabil párosítása.*

Bizonyítás. Az elv egyszerű: töröljük, vagy semlegesítsük minden páratlan rész egy pontját. Ez utóbbi esetben a hozzáadott pontok, a semlegesítendő pontokkal kölcsönösen egymás preferencia-listái első helyén szerepeljenek. □

A 2.21 Állítás és a 2.11 Tétel segítségével két lépésben igazolhatjuk Tan egyértelműségi tételét, a 2.12 Tételt. Az első lépés Tan bizonyításának módosítása, míg a második lépés bizonyítása saját gondolat.

Állítás 2.22. *Egy gráf stabil partícióiban a páratlan részek száma megegyezik.*

Bizonyítás. Tegyük fel, hogy létezik két olyan partíció, Π_1 és Π_2, ahol a páratlan részek száma különböző $k_1 < k_2$. Adjunk hozzá a gráfhoz k_1 darab pontot úgy, hogy ezek Π_1 mindegyik páratlan részéből egyenként pontosan egy ponttal legyenek összekötve és az új pontok a páratlan ciklusbeli pontok preferencia-listái élén legyenek. Ezekkel a ponthozzávételekkel a Tan-Hsueh algoritmus szerint az első esetben mindegyik páratlan rész felbomlik és a kapott Π'_1 stabil partíció már nem fog páratlan részt tartalmazni a kibővített gráfon. Másrészről viszont a Π_2 stabil partícióból kapott Π'_2 tartalmaz legalább $k_2 - k_1$ darab páratlan részt, ugyanis egy pont hozzáadásával legfeljebb eggyel változtozik meg a páratlan részek száma. Ez ellentmond a 2.11 Tételnek. □

Bizonyítás. (Tétel 2.12) Tegyük fel, hogy egy gráfnak két olyan stabil partíciója létezik Π_1 és Π_2, ahol a páratlan részek száma megegyezik, de nem esnek egybe az őket alkotó izolált pontok és ciklusok. A lehetséges eseteket ponthozzáadással fogjuk visszavezetni az előző állításra.

Ha v izolált pont Π_1-ben, de Π_2-ben nem az, akkor vegyük hozzá egy x pontot a gráfhoz úgy, hogy az egyedüli új él, (v, x) csak Π_1-ben legyen blokkoló, vagyis $u <_v x$ az (u, v) Π_2-beli rész-élre. Ekkor $\Pi'_1 = \Pi_1 \setminus \{< v >\} \cup \{< v, x >\}$ és $\Pi'_2 = \Pi_2 \cup \{< x >\}$ új stabil partíciókra az utóbbiban kettővel több páratlan rész van.

Tegyük fel, hogy a két stabil partícióban létezik egy-egy olyan páratlan hosszú ciklus A és B, amely nem esik egybe, de metszi egymást. Ekkor találhatunk olyan v pontot a gráfban, hogy $v = a_i \in A$ és $v = b_j \in B$, továbbá $a_{i+1} \neq b_{j+1}$. Ha úgy vesszük hozzá az x pontot és (v, x) élet a gráfhoz, hogy $b_{i+1} <_v x <_v a_{j+1}$, akkor (v, x) él megint csak az egyik partícióban lesz blokkoló, tehát csak az egyik ciklus lesz semlegesítve. Így $\Pi'_1 = \Pi_1 \setminus \{A\} \cup \{< a_i, x >, < a_1, a_2 >, \ldots, < a_{i-2}, a_{i-1} >, < a_{i+1}, a_{i+2} >, \ldots, < a_{2m}, a_{2m+1} >\}$ és $\Pi'_2 = \Pi_2 \cup \{< x >\}$ új stabil partíciók ismét különböző számú páratlan részt tartalmaznak.

Az utolsó lehetőségként azt az esetet kell megvizsgálnunk, amikor van olyan $A \in \Pi_1$ ciklus, melynek elemei mind pár-beli elemek Π_2-ben. Mivel az A ciklus páratlan hosszú, ezért van olyan $v = a_i \in A$ pontja, melyre $< u, v > \in \Pi_2$ és (u, v) nem rész-él Π_1-ben. Ha (v, u) superior él Π_1-ben, akkor a hozzávett x pontot ismét beilleszthetjük úgy v preferencia-sorrendjébe, hogy csak a ciklust semlegesítje. Ugyanis $u <_v x <_v a_{i+1}$ esetben $\Pi'_1 = \Pi_1 \setminus \{A\} \cup \{< a_i, x >, < a_1, a_2 >, \ldots, < a_{i-2}, a_{i-1} >, < a_{i+1}, a_{i+2} >, \ldots, < a_{2m}, a_{2m+1} >\}$ és $\Pi'_2 = \Pi_2 \cup \{< x >\}$ áll fent. Amennyiben (v, u) inferior él Π_1-ben, akkor $< a_{i+1}, w > \in \Pi_2$-re nem lehet, hogy $w = a_{i+2}$ és az sem, hogy (a_{i+1}, w) Π_1-ben inferior él legyen, mert mindkét esetben (a_i, a_{i+1}) él blokkolna Π_2-ben. Ha pedig (a_{i+1}, w) superior él Π_1-ben, akkor az előzőek szerint jutunk ellentmondáshoz. □

A következő alfejezetben a Scarf-lemma segítségével, másik úton is eljutunk a 2.12 Tétel igazolásához.

2.2 A Scarf-lemma

A fejezet során először ismertetem a Scarf-lemmát, és jelentését a stabil párosítás területén. Bemutatom, hogy hogyan lehet gráfok esetén a stabil partíció karakterizálását megvalósítani a lemma alapján, végül részletesen leírom a hozzá tartozó algoritmikus implementációt.

A fejezet Scarf [11] eredeti cikkén, továbbá Aharoni és Fleiner [1] és [6] cikkein alapszik.

2.2.1 Karakteritáció a Scarf lemmával

Scarf lemmáját egy játékelméleti probléma motiválta. Adva van n játékos, akik bizonyos összetételekben koalíciót alkothatnak egymással. Ha egy koalíciós kiosztás után már nem létezik olyan új koalíció, amelyben minden szereplő kedvezőbb helyzetben érezné magát, mint a jelenlegi koalíciójában (tehát, ha nincs blokkoló koalíció), akkor a kiosztás stabil. Kérdés, hogy létezik-e ilyen kiosztás.

Scarf feladatában lényegében a stabil tört-párosítás keresésének problémáját vizsgálta hipergráfok körében, amiből egyszerűen lehet következtetni a gráfok esetére is. Scarf lemmájának fontosságát a stabil párosítások elméletében csak az utóbbi időkben ismerték fel (lásd: Aharoni és Fleiner [1]). A válasz pedig az a feltett kérdésre, hogy ha megengedjük, hogy a játékosok több koalícióban is benne legyenek, és a koalíciók nem csak egység súllyal, hanem bármilyen részsúllyal is működhetnek, vagyis ha az emberek az idejüket tetszőleges oszthatják el az egyes koalíciókra, akkor mindig létezik megoldása a feladatnak. Másképpen fogalmazva stabil tört-párosítást mindig tudunk találni a hipergráfban.

Természetesen Scarf lemmájából visszakövetkeztetni lehet a gráfok esetére, ahol – mint már Tan bizonyításában láttuk – nem mindig létezik egész megoldása a feladatnak. *Fél-egész* megoldása azonban mindig van, vagyis stabil partíciót minden gráf esetén lehet találni. A fenti gondolatmenet igazolásához mindenekelőtt szükségünk van arra, hogy a stabil partíciót az élek nyelvén is definiáljuk.

Definíció 2.23. Legyen adva egy $G = (V, E)$ gráf és a pontok preferenciái. Ekkor $S \subseteq E$ élhalmaz stabil partíció a gráfban, ha teljesíti az alábbi három feltételt:

1. S minden komponense kör vagy diszjunkt él.
2. Minden kör komponense S-nek preferencia-ciklus, tehát minden egymást követő élre és a rajta fekvő pontokra teljesül, hogy $v_{k-1} <_{v_k} < v_{k+1}$, ahol $e_{k-1} = (v_{k-1}, v_k)$ és $e_k = (v_k, v_{k+1})$.
3. Minden $e \in E \backslash S$ élre létezzen olyan S élhalmaz által fedett v pont, amelyre $e <_v s$ minden $s \in S$ élre, amely fedi v pontot. (Ez másként megfogalmazva azt jelenti, hogy minden él, amely nem része a stabil partíciónak dominálva van legalább az egyik végpontjában S-beli élekkel.)

Tétel 2.24. *(Scarf-lemma) Legyen $n < m$ pozitív egész, és b egy vektor $\mathbb{R}^n_+$-ben. Továbbá legyen $B = (b_{i,j})$ és $C = (c_{i,j})$ két $n \times m$-es mátrix, amely az alábbi feltételeket teljesíti:*

1. *B mátrix első n oszlopa, legyen az $n \times n$-es egységmátrix.*
2. *Az $\{x \in \mathbb{R}^n_+ : Bx = b\}$ halmaz legyen korlátos és nem üres.*

3. Végül legyen $c_{i,i} < c_{i,k} < c_{i,j}$ minden $i \in [n]$ és $k \in [m] \setminus [n]$-re.

Ekkor létezik egy nemnegatív $x \in \mathbb{R}^n_+$ vektor, amelyre $Bx = b$ és az x tartójához, supp(x)-hez tartozó oszlopok C-ben egy domináló halmazt alkotnak, tehát minden $i \in [m]$ oszlopra létezik egy $k \in [n]$ sor C-ben, amelyre $c_{k,i} \le c_{k,j}$ minden $j \in supp(x)$-re.

Scarf a lemmát úgy bizonyította, hogy mutatott egy elemi lineáris algebrai algoritmust, amely minden esetben – bár nem feltétlenül polinomiális idő alatt – eljut a megoldáshoz. Ezt az algoritmust a későbbiekben ismertetem. Aharoni és Fleiner megmutatta, hogy a lemma egy általánosabb fixponttétellel rokon (lásd Fleiner [6]).

Definíció 2.25. Legyen w egy függvény, amely nem-negatív súlyokat rendel egy H hipergráf éleihez, melynek adva van preferenciája. Ekkor w függvényt *tört-párosításnak* nevezzük, ha

$$\sum\nolimits_{v\in h} w(h) \le 1 \tag{1}$$

minden v pontra, továbbá *stabil tört-párosításnak*, hogy ha minden e él tartalmaz egy olyan v pontot, amelyre

$$\sum\nolimits_{v\in h, h\le_v e} w(h) = 1 \tag{2}$$

tehát minden él vagy telített, vagy dominálva van az egyik csúcspontjában.

Tétel 2.26. *Minden hipergráf esetén, ahol adott a pontok élekre vonatkoztatott preferenciája, létezik stabil tört-párosítás.*

Bizonyítás. Legyen $H = (V, E)$ hipergráf, adott preferencia-listákkal. Legyen B a H illeszkedési mátrixa, kibővítve bal oldalról egy egységmátrixszal. Legyen C' olyan $n \times m$-s mátrix, amelyik kielégíti a következő két feltételt:

1. $c'_{v,e} < c'_{v,f}$ amennyiben $v \in e \cap f$ és $e <_v f$.
2. $c'_{v,f} < c'_{v,e}$ amennyiben $v \in f \setminus e$.

C' mátrixot egészítsük ki balról egy négyzetes mátrixszal, úgy, hogy a diagonál elemei minimálisak, a többi pedig nagyobb legyen C' elemeinél. Vagyis az így kapott C mátrix teljesítse a lemma feltételeit. A b pedig legyen csupa $\mathbf{1}$ vektor. Jelöljük x-el a keresett $n + m$ hosszú vektort, és x'-vel x megszorítását az E élhalmazra. Az, hogy x' tört-párosítás rögtön következik a $Bx = \mathbf{1}$ lineáris feltételből. A stabilitás igazolásához vegyük a hipergráf egy tetszőleges e elemét. A lemma állítása szerint ekkor létezik olyan v pont, amelyre $c_{v,e} \ge c_{v,j}$ minden $j \in supp(x)$-re. Miután $c_{v,v} < c_{v,e}$, ezért biztos, hogy $v \notin supp(x)$. A $Bx = \mathbf{1}$ feltételből viszont következik, hogy vagy maga az e él által, vagy más $supp(x)$-beli él(ek) által van fedve a v pont, akik viszont a Scarf-lemma teljesülése miatt dominálják az e élt. ∎

Gráfok esetén a Tan által megkonstruált stabil partíció triviálisan megoldása a Scarf lemmának. Ekkor a pár-beli élek egy súllyal, a ciklus-beli élek pedig fél súllyal szerepelnek az x megoldásban. Az állítás azonban megfordítva is igaz, ami azt jelenti, hogy Tan karakterizációjának egy altarnatív bizonyítását adja.

Tétel 2.27. *Gráfok esetén a Scarf lemma megoldásának tartója, supp*(x) *stabil partíciót ad a gráfban.*

Bizonyítás. Legyen egy adott G gráf, és a hozzá tartozó preferenciára x egy stabil tört-párosítás, és S ennek tartója: $S = supp(x)$. Azt szeretnénk belátni, hogy S stabil partíció a gráfban.

Az x tört-párosítás stabilitása révén irányítsunk meg minden élet abba a végpontjába, amelyben az él dominálva van, tehát amelyre (2) feltétel teljesül. Legyen D az eredményül kapott irányított gráf. Ha e él tört-él ($0 < w(e) < 1$), a stabil tört-párosításban, és $\vec{e}$ a v pontba mutat, akkor kell, hogy legyen a v pontban egy másik f tört-él (1) miatt, amelynek megfelelő $\vec{f}$ (2) miatt, nem mutathat szintén v-be. Ebből adódik, hogy a D-beli $\vec{e}$, $\vec{f}$, … élek irányított köröket adnak. A körökben két egymást követő él súlyának összege 1, ezért páratlan körök esetén a tört-élek súlya mindenképpen 1/2 (a stabil partícióban ezek alkotják a páratlan ciklusokat). Páros körök esetén pedig választható úgy, hogy mindegyik súlya 1/2 (páros hosszú ciklus), vagy minden második él súlya 1 (páros ciklus felbontása stabil párokra). A stabil partíció 3. feltétele azért teljesül S-re, mert, ha $e \in E \setminus S$ él irányított párja $\vec{e} = uv$, akkor az (1) feltétel miatt $e <_v s$ minden $s \in S$-re. Tehát $S = supp(x)$ egy stabil partíció a gráfban. ☐

Megjegyzés 2.28. Mivel a fél-egész stabil tört-párosítások és a stabil partíciók között természetes megfeleltetés létezik, ezért a 2.12 Tétel miatt a fél-egész stabil tört-párosítások tört-élei által alkotott páratlan körök a gráfban egyértelműen helyezkednek el.

2.2.2 Stabil partíció keresése Scarf-algoritmussal

input: Egy gráf és a pontok preferencia-listái.
output: Egy stabil tört-párosítás (stabil partíció a gráfon).
működés: STABIL és PIVOT lépések váltakozó használata.

Az algoritmus elején beállítjuk a C mátrixot, és a B mátrix, b vektor kezdeti értékeit. Ahhoz, hogy a program futása során elkerüljük az ismétlődést, egyrészt a C mátrix sorában kell minden elemet különbözőre állítanunk, másrészt a csupa-1 b vektort kell megperturbálnunk, hogy a bázistraszformációknál ne jöhessen létre degeneráció. Tehát $b(i) = 1 + e(i)$, $B = (I_n, B')$, vagyis az $n \times n$-es egységmátrix és a gráf illeszkedési mátrixa egymás mellé téve. A C mátrix pedig négyféle elemből áll össze: $\alpha < \beta < \gamma < \delta$, ahol

1. $\alpha = C_{i,i}$, ahol $i = 1, \ldots, n$ (pl: $\alpha = 0$);

2. $\beta = C_{i,j}$, ahol $j \geq n, B_{i,j} = 1$, (pl: $C_{i,j} = pr_i(j-n)$, tehát a $j-n$. pont helye az i. pont preferenciájában.);

3. $\gamma = C_{i,j}$, ahol $j \geq n, B_{i,j} = 0$, (pl: $C_{i,j} = m + 2n - j$);

4. $\delta = C_{i,j}$, ahol $j \leq n, i \neq j$ (pl: $C_{i,j} = m + 2n - j$).

Kezdeti állapotban legyen a megoldás tartója: $supp(x) = \{1, 2, 3, \dots, n\}$. Ez kielégíti a Scarf-lemma stabil tört-párosításra vonatkozó feltételei közül az egyiket: $Bx = \mathbf{1}$ (a továbbiakban *bázis-feltétel*), viszont nem teljesíti a domináló oszlopok feltételét (továbbiakban *stabil-feltétel*) vagyis, hogy a megoldásban szereplő oszlopok mindegyike pontosan egy sorban dominál (i. oszlop esetén jelöljük ezt a sort $q(i)$-nek.)

A kezdeti közös bázist jelöljük kétféleképpen: BP_0 és BS_0. Az algoritmus működésének lényege, hogy a kezdeti helyzetéből kiindulva a megoldás egyszer a stabil-feltételt, egyszer a bázis-feltételt elégíti ki. A két bázis mindig csak egy elemben különbözik, és egyszer ún. PIVOT lépés egyszer pedig az ún. STABIL lépéssel érjük el, hogy teljesüljön az egyik illetve a másik feltétel. Az algoritmusnak akkor lesz vége, ha egyszer a két bázis megegyezik.

0. lépés:

A kiinduló helyzetből a kezdeti lépés speciális: $BS_1 := BS_0 \setminus \{1\} \cup t_1$, ahol $t_1 = max_t\{C_{1,t}, t \notin supp(x)\}$. Tehát az 1 pontot kivesszük a stabil bázisból és helyette az első sor azon elemei közül választjuk ki a maximálisat, amely nem szerepel a tartóban. Így a stabil-feltétel már teljesülni fog, a bázis-feltétel viszont nem. A két bázis különbsége az 1. oszlop.

k. lépés:

PIVOT lépés: A t_k. oszlopot bevesszük a pivot-bázisba, és a bázistranszformáció során a bázisból kieső oszlop legyen a s_k. oszlop. Ez egyértelműen meghatároztott. Így az új pivot-bázis, $BP_{k+1} := BP_k \cup t_k \setminus s_k$ a megfelelő x megoldásvektorral már kielégíti a bázis-feltételt. Ha $s_k = 1$, akkor a két bázis megegyezik, az algoritmus véget ér, ha viszont $s_k \neq 1$, akkor a stabil-feltétel nem teljesül, a két bázis különbsége ismét egy oszlop.

STABIL lépés: Az s_k oszlopot kivesszük a domináló oszlopok közül. Ebben az esetben viszont a maradék $n-1$ oszlop között lesz egy olyan j-edik oszlop, amelyik két pontban is dominál. Az algoritmus STABIL lépése szerint ekkor a stabil bázishoz úgy veszünk hozzá egy új m. oszlopot, hogy a j. oszlop megőrizze új dominanciáját az s_k. sorban, viszont a régi sorában a dominanciát az m. oszlop vegye át úgy, hogy a többi oszlop helyzete ne változzon. Tehát m legyen az az oszlop, ahol $C_{q(i),m}$ maximális feltéve, hogy $m \notin BS_k$ és $C_{q(j),m} > C_{q(j),j}$ minden $j \in BS_k$-ra. Ha a sorokban minden elem különbözik, akkor a kiválasztás egyértelmű. Vagyis $q(m) := q(i)$ és $q(i) := s_k$ az új dominanciák, és az új bázis

$t_{k+1} = m$ esetén, $BS_{k+1} := BS_k \setminus s_k \cup t_{k+1}$. Ha $t_{k+1} = 1$, akkor a két bázis megegyezik és az algoritmus befejeződik, ha nem, akkor a bázis-feltétel ismét nem teljesül.

Az algoritmus nem ismétlődik, mivel a perturbáció révén degeneráció nem léphet fel a PIVOT lépés során, és a C mátrix egy sorban lévő elemeinek különbözősége a STABIL lépésben szintén biztosítja az egyértelműséget, ezért egyik lépésben sem juthatunk vissza egy korábbi bázisba. Az algoritmus tehát véges időn belül befejeződik. A futásidő elméletileg nem polinomiális, de a tapasztalat szerint stabil partíció keresése esetén a lépések száma ritkán nagyobb, mint az élek száma.[4]

A kezdeti feltételek beállítása után az algoritmus már egyértelműen jut el az új egyensúlyi pontba. Ennek mélyebb magyarázatát egy fixponttétel adja (lásd Aharoni és Fleiner [1]).

[4]A nagy méretű mátrixok kezelése viszont már nagyjából 20 pont, és átlagos elsűrűség mellett nem teszi lehetővé az algoritmus végigfuttatását egy átlagos számítógépen MAPLE programcsomag alatt.

3 Stabil b-párosítás nem páros gráfon

Az alapprobléma két főbb kiterjesztésének közös része, ha általános gráfon keresünk stabil b-párosítást. A kérdéskör irodalma jelenleg még szegényesnek mondható bár, mint látni fogjuk több egyszerű visszavezethetőségi lehetőség is létezik. A dolgozatban hárommal fogok részletesen foglalkozni. Először a Scarf-lemma megfelelő mátrixokon történő alkalmazásával jutunk el a megoldáshoz, és egyben a feladat karakterizációjához. Ezután a gráf alapú megoldások közül először Cechlárová és Fleiner [3] konstrukción alapuló visszavezetését, majd a Tan-Hsueh algoritmus közvetlen kiterjesztését fogom bemutatni.

3.1 Scarf-lemma b-párosításra

Az alfejezetben először bemutatom, hogy a Scarf-lemma segítségével hogyan oldható meg a b-párosítás problémája gráfok esetén, és miként karakterizálható ennek segítségével a megoldás. Majd megmutatom, hogy hogyan működik az algoritmus implementációja a gyakorlatban. Végül ismertetek egy másik általánosítási lehetőséget, a stabil allokáció problémáját, amelyet szintén megold a Scarf-lemma.

3.1.1 A b-párosítás karakterizációja

A bizonyítás gondolatmenete hasonló lesz a b-párosítás esetében: először definiáljuk magát, a b-párosítást hipergráfokra, majd a Scarf-lemmát alkalmazzuk jól megválasztott, kiterjesztett mátrixokra, ezzel belátjuk a *stabil tört-b-párosítás* létezését. Ezt követően belátjuk, hogy gráfok esetén mindig találhatunk *fél-egész* stabil tört-b-párosítást is, amely megfelel a gráfban egy *stabil b-partíciónak*. Végül ennek karakterizációjára nézve teszünk – a Scarf-lemmából adódóan – megállapításokat. Ebben az alfejezetben főként Fleiner [5] cikkére támaszkodom.

Definíció 3.1. Egy preferencia-listákkal megadott hipergráf esetén definiálhatjuk $b : V(H) \to \mathbb{N}$, vektort, amely a hipergráf minden pontjához hozzárendeli annak kapacitását. Az élek egy M halmazát *b-párosításnak* nevezzük, ha a hipergráf minden v pontjára

$$|\{e_i : e_i \in M, v \in e_i\}| \leq b(v) \tag{3}$$

Egy f él az M b-párosítás által dominálva van egy v pontban, ha

$$|\{e_i : e_i \in M, e_i <_v f\}| = b(v) \tag{4}$$

tehát ha v kapacitása telítve van M-ben f-nél kedvezőbb élekkel. Egy S, b-párosítást *stabilnak* nevezünk, ha minden $f \in E(H) \setminus S$ dominálva van egy csúcspontjában.

Amennyiben megengedjük, hogy az élek nem csak 1 súllyal szerepelhetnek, akkor *tört-b-párosításról* beszélhetünk.

Definíció 3.2. Legyen w egy függvény, amely nem-negatív súlyokat rendel egy H hipergráf éleihez, vagyis $w : E(H) \to \mathbb{R}^+$ Ekkor w függvényt *tört-b-párosításnak* nevezzük, ha $w(e) \leq 1$ minden élre és

$$\sum\nolimits_{v \in h} w(h) \leq b(v) \tag{5}$$

minden v pontra, továbbá *stabil tört-b-párosításnak*, hogy ha minden e él, amelyre $w(e) \neq 1$, tartalmaz egy olyan v pontot, amelyre

$$\sum\nolimits_{v \in h, h \leq_v e} w(h) = b(v) \tag{6}$$

tehát minden él vagy telített, vagy dominálva van az egyik csúcspontjában.

Tétel 3.3. *Minden preferenciákkal adott hipergráfra, és b pont-kapacitásokra létezik stabil tört-b-párosítás.*

Bizonyítás. A Scarf-lemmát fogjuk használni csakúgy, mint a 2.26 Tétel bizonyításában. Legyen adva a B és C $(n{+}m){\times}(n{+}2m)$-es mátrix a következőképpen:

$$B := \begin{bmatrix} I_n & 0 & B' \\ 0 & I_m & I_m \end{bmatrix}$$

ahol B a hipergráf illeszkedési mátrixa, amely kiegészül egység és csupa-0 mátrixokkal. A C mátrix jobb felső $n \times m$-es részmátrixa, C' legyen olyan, mint az alapesetben:

1. $c'_{v,e} < c'_{v,f}$ amennyiben $v \in e \cap f$ és $e <_v f$.
2. $c'_{v,f} < c'_{v,e}$ amennyiben $v \in f \setminus e$.

C' mátrixot egészítsük ki először alulról egy $m \times m$-es mátrixszal, melynek diagonál elemei kisebbek, a többi eleme pedig nagyobb C' mátrix elemeinél. Ezután balról tegyünk mindehhez hozzá egy $(n+m) \times (n+m)$-es négyzetes mátrixot, úgy, hogy a diagonál elemei az egész C-ben minimálisak, a többi elem pedig maximális legyen. Az így kapott C mátrix teljesíti a lemma feltételeit. A b vektor első n eleme legyen $b(v)$ minden $v \in V(G)$-re, vagyis a pont kapacitása, és a maradék m elem pedig csupa $\mathbf{1}$. Láthatjuk, hogy a két mátrix első n oszlopa a pontok maradék-kapacitását, míg az azt követő m oszlop az élek maradék-kapacitását jelenti. A $Bx = b$ egyenlet első n sora az (5) feltételt, vagyis a pont-kapacitás korlát teljesülését, második m sora a $w(e) \leq 1$ feltételt, vagyis az élek súlyának természetes, 1-ben történő korlátozását jelenti. (Ez utóbbi feltételt fogjuk a stabil allokáció probléma esetén általánosítani.)

Jelöljük x-el a Scarf-lemma megoldásaként kapott $n + 2m$ hosszú vektort, $supp(x)$-el ennek tartóját és x'-vel x megszorítását az utolsó m oszlopra, vagyis az E élhalmazra. Az, hogy x' tört-párosítás rögtön következik az imént taglalt $Bx = b$ lineáris feltételből. A stabilitás igazolásához vegyük egy tetszőleges e

elemét a hipergráfnak. Ha e él nincs telítve ($w(e) \neq 1$), akkor a lemma állítása szerint, létezik olyan v pont, amelyre $c_{v,e} \geq c_{v,j}$ minden $j \in supp(x)$-re. Miután $c_{v,v} < c_{v,e}$, ezért biztos, hogy $v \notin supp(x)$. Ez viszont a $Bx = b$ feltételből következően azt jelenti, hogy v teljes $b(v)$ kapacitását $supp(x)$-beli élek elégítik ki, akik a Scarf-lemma feltétele miatt dominálják az e élet. Tehát teljesül a (6) feltétel, az e él a v pontban dominálva van, vagyis a tört-b-párosítás stabil. □

Tétel 3.4. *Gráfok esetén minden preferenciára, és b pont-kapacitásra, létezik fél-egész stabil tört-b-párosítás.*

Bizonyítás. A bizonyítás itt is hasonló, mint $b = \mathbf{1}$ esetén volt. Azt fogjuk belátni, hogy minden tört-él ($0 < w(e) < 1$) egy kör része. Irányítsuk meg a tört éleket úgy, hogy mindegyik él mutasson abba a pontba, amelyben dominálva van. Ha az így kapott D gráfban $\vec{e}$ él v-be mutat, akkor teljesül rá a (6) feltétel, tehát kell legyen egy másik f tört-él is v-ben, ami dominálja e-t. Az f él is dominálva van viszont, de nem v-ben, hanem a másik végpontjában, hiszen egy pontban csak egy tört-él lehet dominálva. Ebből a gondolatmenetből adódik, hogy $\vec{e}$, $\vec{f}$, ..., élek irányított utat adnak, amelynek eredményeként a D gráf élei diszjunkt irányított körök, melyekben a két egymást követő tört-él összege 1, tehát, ha az irányított kör páratlan hosszú, akkor minden tört-él súlya szükségképpen 1/2 (ez jelenti a páratlan hosszú ciklust a stabil partícióban), ha páros hosszú, akkor választhatjuk az összes súlyt szintén 1/2-ednek (páros hosszú ciklus), vagy minden másodikat 1-nek (ez a páros hosszú ciklus felbontása párokra). □

Az egyszerű, $b = \mathbf{1}$ esetben már adva volt a feladat karakterizációja, a stabil partíció Tan által bevezetett fogalmával, és erről láttuk be, hogy megegyezik egy a Scarf-lemma megoldásaként kapott fél-egész stabil tört-párosítással. A b-párosításra viszont a karakterizáció még nem definiált, ezért azt közvetlenül származtathatjuk az előző eredményből.

Definíció 3.5. Legyen adva G gráf, a pontok preferenciái és a pontok kapacitása. Ekkor $S \subseteq E$ élhalmaz *stabil b-partíció* a gráfban, ha $S = P \cup C$, vagyis pontpároknak megfelelő élek és ciklusok éleinek egyesítése, ahol P és C halmaz egymástól diszjunkt az élek tekintetében, és a C-beli ciklusok pontjaikban is diszjunktak egymástól. Továbbá ha $D(v)$ a v pontból kiinduló élek halmaza, akkor a P és C élhalmaz teljesíti az alábbi feltételeket:

1. minden v pontra
$$|D(v) \cap P| + 1/2|D(v) \cap C| \leq b(v) \tag{7}$$
2. a (7) feltétel egyenlőséggel teljesül, azon pontokra, amelyeket C lefed, ekkor C egyik éle a legrosszabb az $S \cap D(v)$ élek közül.
3. minden $e \in E \backslash S$ élre az egyik végpontjában a fenti (7) feltétel egyenlőséggel teljesül, tehát $s <_v e$ minden $s \in D(v) \cap S$ élre.

3.1.2 Stabil b-partíció keresése Scarf-algoritmussal

Az algoritmus működése természetesen megegyezik a korábban elmondottakkal, itt csupán a kezdeti értékek beállításának egy lehetséges, – a mellékelt programokban is használt – módját szeretném részletezni.

Az ismétlődések elkerülése miatt ebben az esetben is fontos, hogy mind a C mátrix soraiban különböző elemek legyenek, mind pedig a b vektor értékei meg legyenek perturbálva. A C mátrix egy lehetséges megadásában az elemek álljanak a következő nagyságbeli sorrendben: $\alpha < \alpha' < \beta < \gamma < \delta$, ahol

1. $\alpha = C_{i,i}$, ahol $i = 1 \ldots (n+m)$ (pl: $\alpha = 0$);
2. $\alpha' = C_{i,i+m}$, ahol $i = (m+1) \ldots (n+m)$ (pl: $\alpha' = 1$);
3. $\beta = C_{i,j+n+m}$, ahol $j \geq n, B'_{i,j} = 1$, (pl: $C_{i,j+n+m} = pr_i(j)$, tehát a j. pont helye az i. pont preferenciájában.);
4. $\gamma = C_{i,j}$, ahol $j \geq n, B'_{i,j} = 0$, (pl: $C_{i,j} = 2m + 2n - j$);
5. $\delta = C_{i,j}$, ahol $j \leq n, i \neq j$ (pl: $C_{i,j} = 2m + 2n - j$).

Az algoritmus lépésszáma elméletileg ezúttal sem polinom rendű, de a tapasztalatom szerint kis méretű gráfoknál még mindig közel van az élek számához. Az algoritmus működésére példát a mellékletben, és kipróbálható változatot a honlapomon találhat az Olvasó. [5]

Mielőtt rátérnék a stabil b-párosítások egy másik – Tan-Hsueh algoritmus kiterjesztésével kapott – változatának bemutatására előbb ismertetem a Scarf-lemma egy újabb alkalmazását, amely a b-párosítás fenti megoldásából közvetlenül adódik.

3.1.3 Stabil allokáció probléma

Abban az esetben, ha a stabil b-párosításnál nem csak a pontok kapacitását adjuk meg, hanem az élekét is, akkor beszélünk *stabil allokáció problémáról.* A probléma felvetése és megoldása páros gráfok esetére Baïou és Balinski [2] cikkében szerepel, de nem-páros gráfok esetén pontos leírást nem találtam az irodalomban. A megoldás a b-párosítás felírása után már természetesen adódik a Scarf-lemmából.

Fontos megjegyezni, hogy egy olyan algoritmus, amely a stabil b-párosítás keresését párhuzamos éleket tartalmazó gráfokon is elvégzi, – ilyen Cechlárová és Fleiner [3] nemrégiben publikált algoritmusa – egyszerű visszavezetéssel megoldja a fenti problémát. Erről részletesebben az alfejezet végén még szót fogok ejteni.

[5] A mátrixok méretéből fakadóan azonban algoritmus már csak 10-15 pontú gráf esetén fut le egy átlagos számítógépen MAPLE programcsomag alatt.

Definíció 3.6. Legyen tehát adva egy hipergráf a preferenciáival, és a rajta értelmezett $b : V(H) \to \mathbb{N}$ pont-kapacitás, és $b_e : E(H) \to \mathbb{N}$ élkapacitás. Feladat egy $A : E(H) \to \mathbb{N}$ stabil allokáció keresése, amelyre $A(f) \leq b_e(f)$ minden f élre. Ekkor a stabil b-párosítás feltételei a következőképpen módosulnak: Legyen A' azon élek halmaza, amelyek nem 0 értéket vesznek fel és v a hipergráf egy pontja, ekkor

$$\sum\nolimits_{e_i \in A', v \in e_i} A(i) \leq b(v) \tag{8}$$

Egy f él az A' allokáció által dominálva van egy v pontban, ha

$$\sum\nolimits_{e_i \in A', e_i <_v f} A(i) = b(v) \tag{9}$$

tehát ha v kapacitása telítve van A'-beli, f-nél kedvezőbb élekkel. Egy A_s allokációt *stabilnak* nevezünk, ha minden $f \in E(H) \setminus A'_s$ dominálva van egy végpontjában.

Hasonlóképpen definiálhatjuk a tört- és a stabil tört-allokációt.

Definíció 3.7. Legyen w egy függvény, amely nem-negatív súlyokat rendel egy H hipergráf éleihez, vagyis $w : E(H) \to \mathbb{R}^+$, úgy hogy $w(f) \leq b_e(f)$ minden f élre, akkor w függvényt *tört-allokációnak* nevezzük, ha

$$\sum\nolimits_{v \in h} w(h) \leq b(v) \tag{10}$$

minden v pontra, továbbá *stabil tört-allokációnak*, hogy ha minden f él, amelyre $w(f) \neq b_e(f)$, tartalmaz egy olyan v pontot, amelyre

$$\sum\nolimits_{v \in h, h \leq_v f} w(h) = b(v) \tag{11}$$

tehát minden él vagy telített, vagy dominálva van az egyik végpontjában.

Tétel 3.8. *Gráfok esetén minden preferenciára, b pont-kapacitásra, és b_e él-kapacitásra, létezik fél-egész stabil tört-allokáció.*

Bizonyítás. A bizonyítás menete mindössze annyiban tér el a 3.3 Tétel bizonyításától, hogy a $w(f) \leq 1$ feltétel teljesülése helyett $w(f) \leq b_e(f)$ feltétel áll fenn, és ennek megfelelően a Scarf-lemmában mindössze az élekhez tartozó m hosszú csupa-**1** elemeit kell kicserélni az él-kapacitások $b_e(f)$ értékeire minden f élre. Ezután a bizonyítás már teljes mértékben megegyezik at ott elmondottakkal. □

Tétel 3.9. *Gráfok esetén minden preferenciára, és b pont-kapacitásra, létezik fél-egész stabil tört-b-párosítás.*

Bizonyítás. A bizonyítás szintén megegyezik a 3.4 Tétel bizonyításával, mindössze annyi különbséggel, hogy a *tört-élnek* ez esetben azon f éleket nevezzük, amelyekre $0 < w(f) < b_e(f)$. □

A feladat karakterizációjára pedig definiálhatjuk a *stabil allokáció-partíció* fogalmát:

Definíció 3.10. Legyen adva G gráf, a pontok preferenciái és a pontok, illetve az élek kapacitása. Ekkor $A_s \subseteq E$ élhalmaz *stabil allokáció-partíció* a gráfban, ha $A_s = A_p \cup A_c$, vagyis pontpároknak megfelelő élek és ciklusok éleinek egyesítése. A_p, azon éleknek felel meg, amelyek egy megfelelő stabil tört-allokációban pozitív egész súllyal, és A_c azoknak az éleknek, melyek súlyának törtrésze fél. Ekkor A_p és A_c halmaz egymástól diszjunkt az élek tekintetében, és a A_c-beli élek pontjaikban is diszjunkt ciklusokat alkotnak. Továbbá ha $D(v)$ a v pontból kiinduló élek halmaza, és A tetszőleges élhalmazra $\lfloor A \rfloor$, és $\{A\}$ az $f \in A$ élek kihasznált kapacitásának egészrészének, illetve törtrészének összege , akkor a A_p és A_c élhalmaz teljesíti az alábbi feltételeket :

1. minden v pontra
$$\lfloor D(v) \cap A_p \rfloor + \lfloor D(v) \cap A_c \rfloor + 1/2(\{D(v) \cap A_c\}) \leq b(v) \qquad (12)$$
2. a (12) feltétel egyenlőséggel teljesül, azon pontokra, amelyeket A_c lefed, és ekkor A_c egyik éle a legrosszabb az $A_s \cap D(v)$ élek közül.
3. minden $e \in E \backslash A_s$ élre az egyik végpontjában a fenti (12) feltétel egyenlőséggel teljesül, tehát $s <_v e$ minden $s \in D(v) \cap A_s$ élre.

Az algoritmikus implementációra példát a mellékletben illetve a honlapomon talál az Olvasó.

Végül visszatérve a bevezető gondolatokhoz, Cechlárová és Fleiner [3] algoritmusa – amely az Irving algoritmus kiterjesztésén alapul – párhuzamos éleket tartalmazó gráfok esetén is megoldja a b-párosítás problémáját. Az ő motivációjuk erre az volt, hogy egy pár többféle kapcsolatba is kerülhet egymással. Ha például egy klubban vagyunk, akkor másként ítélhetünk meg egy óra teniszt, sakkot, vagy kártyázást ugyanazzal a partnerrel. Hasonló a helyzet a repülőgépvezetőknél, akik másként ítélhetik meg egy társukkal való repülést attól függően, hogy első- vagy másodpilótaként repülnek vele.

Ha pontosan annyi másolatot készítünk egy-egy élből, amennyi az él kapacitása, és ezeket a többi élhez képest hasonlóképpen viszonyítjuk, mint az eredeti élet, a másolatok közti sorrend pedig tetszőleges, akkor a fenti algoritmus megoldja a stabil allokációs problémát. Sőt, ebben az esetben lehetőségünk van arra, hogy a kapacitás kihasználtságának függvényében az élek preferenciája megváltozzon (a harmadik órát is teniszezéssel tölteni A-val már lehet, hogy kevésbé szeretnénk, mint ping-pongozni egyet B-vel).

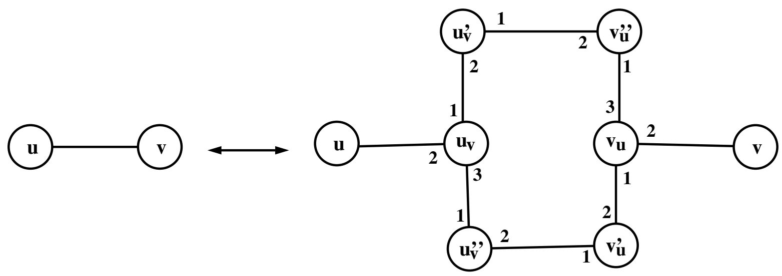
3.2 Gráf alapú algoritmusok a b-párosításra

A gráf alapú algoritmusok keresésére azért van szükség b-párosítás esetén is, mert az előbb ismertetett Scarf algoritmus elméletileg sem polinomiális, és a hatalmas mátrixok miatt a gyakorlatban sem ad hatékony megoldást.

A stabil b-párosítás megoldására először ismertetek egy konstrukciót, amelynek a segítségével a probléma visszavezethető a stabil párosítás problémájára – abban az esetben is, ha voltak a gráfban párhuzamos élek – majd a Tan-Hsueh algoritmus kiterjesztésével a stabil b-partíció problémájára is adunk egy lehetséges megoldást.

3.2.1 Megoldás közvetlen visszavezetéssel

A visszavezetés alapjául szolgáló konstrukció Cechlárová és Fleiner [3] cikkében található. A visszavezetés első lépésében az *éleket húzzuk szét* kis 6-os ciklusokkal, ezzel elérjük, hogy egyrészt eltűnjenek a párhuzamos élek, másrészt minden él legalább egyik végpontjának kapacitása 1 legyen, ezután a második lépésben *csúcstöbbszörözéssel* fejeződik be a visszavezetés.



Ábra 7: Élszéthúzás

Az első lépésben egy adott G gráf minden élét kibővítjük egy 6-os ciklussal a fenti módon. Az élek helye az eredeti pontok preferenciáiban nem változik, az új pontok között pedig az ábrán látható módon rendezzük az éleket. Az így kapott gráfot, az új preferenciákkal együtt nevezzük el G'-nek. A következőkben az fogjuk belátni, hogy minden stabil párosítás G'-n egyértelműen megfeleltethető egy stabil párosításnak G-n.

Tétel 3.11. *Minden stabil párosítás G'-n indukál egy stabil párosítást G-n, és minden stabil párosítás G-n származtatható egy G'-beli stabil párosításból. Ez azt is jelenti, hogy akkor és csak akkor létezik stabil párosítás G-n, ha G'-n is létezik.*

Bizonyítás. Mindkét irány bizonyításának alapja, hogy az (u, u_v) él pontosan akkor szerepel egy stabil párosításban G'-n, ha (v_u, v) is szerepel benne. Ez egyszerűen következik a megadott preferenciákból.
Legyen M' egy stabil párosítás G' gráfon, ekkor $M := \{(u, v) : (u, u_v) \in M'(\iff (v_u, v) \in M')\}$ stabil párosítás G-n, ugyanis minden $e = (u, v) \notin M$

él pontosan akkor lesz dominálva G gráf egy u pontjában, ha $e' = (u, u_v) \notin M'$ él dominálva volt G'-ben.

A másik irányban: ha M stabil partíció volt, akkor M' álljon a következő élekből: $\{(u, u_v), (v_u, v), (u'_v, v''_u), (u''_v, v'_u)\} \in M' \iff (u, v) \in M$ és $\{(u_v, u'_v), (v_u, v''_u), (u''_v, v'_u)\} \in M' \iff (u, v) \notin M$, abban az esetben, ha (u, v) él a v végpontjában volt dominálva G-ben. Ha viszont (u, v) él az u végpontjában volt dominálva, akkor az utóbbi megfeleltetés a következő megadásra módosul: $\{(u_v, u''_v), (v_u, v'_u), (u'_v, v''_u)\} \in M' \iff (u, v) \notin M$. A párosítás stabilitása a 6-os ciklusok éleire az ábra alapján könnyen igazolható, a gráf eredeti éleire pedig hasonlóképpen bizonyítható, mint az előző esetben. □

A következő lépésben a *csúcstöbbszörözés* konstrukcióval foglalkozunk, amelyet már páros gráfok esetén korábban is használtak visszavezetésre. Adott b pont-kapacitásokra definiáljuk G^b gráfot a következőképpen:

$$V(G^b) := \{v^i : v \in V(G) \text{ és } i = 1, 2, \ldots, b(v)\}$$

$$E(G) := \{e^{i,j} = (u^i, v^j) : e = (u, v) \in E(G) \text{ és } u^i, v^j \in V(G^b)\}$$

A preferenciákra pedig

$$v^j <_{u^i} w^k \iff \begin{cases} u, <_u w \text{ ,vagy} \\ v = w \text{ és } j < k. \end{cases}$$

Ebben az esetben igaz a következő tétel.

Tétel 3.12. *Legyen adva egy G gráf a preferenciáival és a b pont-kapacitásokkal. Teljesüljön rá továbbá az, hogy minden élének legalább az egyik végpontja 1 kapacitású legyen (egy-a-sokhoz tulajdonság). Ekkor minden stabil párosítás G^b-n indukál egy stabil b-párosítást G-n és minden stabil b-párosítás G-n származtatható egy G^b-beli stabil párosításból.*

Bizonyítás. Legyen M^b egy stabil párosítás G^b-n. Ekkor belátjuk, hogy

$$M := \{e \in E(G) : e^{i,j} \in M^b \text{ valamely } 1 \le i \le b(u) \text{ és } 1 \le j \le b(u)\} \tag{13}$$

stabil b-párosítás G-re. Mivel egy v pont egyik másolata sem lehet fedve 1-nél több M^b-beli éllel, ezért M valóban b-párosítás. A stabilitás igazolásához azt kell belátnunk, hogy minden $e = (u, v) \notin M$ él b-dominált G-ben. A tétel feltétele szerint e egyik végpontja, mondjuk u legyen 1-kapacitású. Ha $e^{1,j} \notin M^b$ él valamely $j = 1, 2, \ldots, b(v)$-re a G^b-ben az u pontban volt dominálva, akkor G-ben is dominálva lesz u pontban. Ha ez nem teljesül, akkor $e^{1,j} \notin M^b$ él minden $j = 1, 2, \ldots, b(v)$-re v^j-ben volt dominálva valamely m_j^{j,k_j} élek által, amelyeknek a képe a tétel feltétele miatt különböző $m_1, m_2, \ldots, m_{b(v)}$ élei M-nek. Mivel $m_i <_v e$ fennáll minden i-re , ezért e él b-dominált M-ben.

Legyen M egy stabil b-párosítás G-n, ekkor kell lennie egy M^b párosításnak, amelyből (13) szerint származtatható. Ebben az esetben, ha egy $e^{i,j}$ él blokkolna M^b-ben, akkor a neki megfelelő e él nem lehetne b-dominált M párosítással, ezért M^b stabil párosítás. □

A stabil b-párosítás párhuzamos élek esetén probléma visszavezetése a stabil párosítás problémára tehát ezen két konstrukció egymás utáni alkalmazásával valósul meg. A preferenciákkal és b pont-kapacitással megadott G gráfból először az élek széthúzásával G' gráfot, majd a csúcstöbbszörözéssel $(G')^b$ gráfot kapjuk, ahol amennyiben találunk egy stabil párosítást, az indukál egy stabil b-párosítást G-n, és minden b-párosítás G-n származtatható egy $(G')^b$ gráfon vett stabil párosításból. Amennyiben nem találunk stabil párosítást $(G')^b$-n, akkor G gráfra sem létezik stabil b-párosítás. A stabil párosítás megkeresésére használhatjuk például Irving algoritmusát.

Megjegyzés 3.13. Cechlárová és Fleiner [3] a konstrukció alapján létrehozta Irving algoritmusának egy kiterjesztett változatát. Ez hasonlóképpen $O(m^2)$-es lépésszámmal talál stabil b-párosítást a párhuzamos éleket is tartalmazó gráfon, de a konstans szorzó lényegesen kisebbé válik, mint ha csak egyszerűen az Irving algoritmust használnánk a visszavezetéssel kapott gráfra.

Megjegyzés 3.14. Ez a visszavezetés valószínűleg működik stabil b-partíció keresésre is. Az élszéthúzás esete hasonló módon bizonyítható, mint párosítás esetén, viszont a csúcstöbbszörözés konstrukciónál már nem ilyen egyértelmű a visszavezetés. Ha azonban bebizonyosodik ennek helyessége, akkor a Tan vagy a Tan-Hsueh algoritmussal a konstrukción keresztül meg tudjuk oldani a stabil b-partíció problémáját párhuzamos éleket tartalmazó gráfokon is, és ezzel a stabil allokáció-partíció is megoldhatóvá válik gráfalapú algoritmussal. Továbbá, ha Tan algoritmusát hasonlóképpen kiterjesztjük, mint ahogy Cechlárová és Fleiner [3] tette Irving algoritmusával, akkor a Tan-Hsueh algoritmus kiterjesztésénél hatékonyabb algoritmust kaphatunk a stabil b-partíció problémára.

3.2.2 A Tan-Hsueh algoritmus kiterjesztése stabil b-partícióra

A Scarf-lemmából levezetett 3.1 karakterizációt fogalmazzuk meg először új formában a gráf pontjaira koncentrálva, figyelembe véve Tan eredeti 2.10 stabil partícióra vonatkozó definícióját.

Definíció 3.15. Adott egy G gráf a pontok preferenciáival és a b pontkapacitásokkal. A gráf pontjai az élek mentén alkothatnak *párokat* és *ciklusokat*, úgy, hogy mindegyik $v \in V(G)$ pont legfeljebb $b(v)$ darab párban szerepelhet, vagy pontosan $b(v) - 1$ darab párban és egy ciklusban (ennek definíciója megegyezik a 2.10-belivel). Ez utóbbi esetben a kevésbé kedvelt ciklusbeli szomszédjánál mindegyik párját jobban kedveli. A párokhoz tartozó éleket nevezzük *rész-éleknek*. A pontok halmazát ezek után három részre osztjuk: $V(G) = T \cup C \cup P$. T jelölje azon pontok halmazát, amelyek nincsenek telítve. C azon pontok halmaza, melyek szerepelnek ciklusban. P pedig azokat a pontokat jelölje, akiknek a kapacitása párbeli élekkel van telítve, és speciálisan a 0 kapacitású pontokat is ebbe a csoportba soroljuk.
Ezt követően nevezzünk el minden nem rész-élet superior vagy inferior élnek a következőképpen.

1. T-beli pontból csak superior élek indulnak ki.

2. C-beli pont esetén a rosszabbik ciklus-beli élnél jobb élek superior élek, a legalább olyan rossz élek pedig inferior élek.

3. P-beli pontok esetén a legrosszabb rész-élnél jobb élek superior élek, az annál rosszabbak pedig inferior élek.

Egy b-partíciót – amely a pontok $V(G) = T \cup C \cup P$ részekre osztását indukálja – akkor nevezünk *stabilnak*, ha nem létezik olyan él. amelyik blokkol, vagyis amelyik mindkét végpontja szerint superior él.

A definíciókat átgondolva felismerhetjük a kapcsolatot a fenti ponthalmazok, és az eredeti bizonyításban szereplő izolált pontok, párok és ciklusok pontjai között. A Tan-Hsueh algoritmushoz hasonlóan a kiterjesztett algoritmus is dinamikusan fog működni, de most nem a pontok számát, hanem a pontok kapacitását fogjuk egyenként növelni. Egy ilyen kapacitás-növelést egy *ütemnek* nevezünk, és egy pont teljes kapacitásának feltöltését egy lépésnek. Így tehát az algoritmus összesen n lépésből és $\sum_{i=1}^{n} b(v_i)$ ütemből fog állni.

Legyen G adott a preferenciáival, és a $b(v)$ pontkapacitásokkal. Jelölje G_{v^+} azt a gráfot, ahol a v pont kapacitását eggyel megnöveltük, és G_{v^-} azt, ahol lecsökkentettük eggyel. Amennyiben $b\Pi$ stabil b-partíció volt a G gráfon, akkor jelöljük $b\Pi_v$-vel a stabil b-partíciót G_{v^-} gráfon.

A következőkben azt fogjuk belátni, hogy ha ismerjük egy tetszőleges v pontra a $b\Pi_v$ stabil partíciót G_{v^-} gráfon, akkor v pont kapacitását 1-el növelve meg tudunk határozni egy $b\Pi$ stabil b-partíciót a G gráfon. Egy ütemben tehát a következő műveletet végzi el az algoritmus:
Input: $b\Pi_v$ páros ciklus mentes stabil b-partíció G_{v^-} gráfon.
Output: $b\Pi$ páros ciklus mentes stabil b-partíció a G gráfon.

Az algoritmus megalkotásához vegyük végig újra a Tan-Hsueh algoritmus gondolatmenetét. Első lépésben nézzük meg, hogy hogyan változik meg a stabil b-partíció, attól függően, hogy milyen pontnak növeltük a kapacitását 1-gyel.

1. Ha v pont telítetlen volt, tehát $v \in T$, akkor G-ben nem jönnek létre új superior élek, ezért az eredeti stabil b-partíció stabil marad. Vagyis $b\Pi := b\Pi_v$ megoldás.

2. Ha v pont része volt egy páratlan $A =< a_1, a_2, \ldots, v(= a_{2k}), \ldots, a_{2m+1} >$ ciklusnak, tehát $v \in C$, akkor egyszerűen belátható, hogy $b\Pi := b\Pi_v \setminus A \cup \{< a_1, a_2 >, \ldots < a_{k-1}, v >, < v, a_{k+1} >, \ldots, < a_{2m}, a_{2m+1} >\}$ stabil b-párosítás lesz, ugyanis új superior élek nem jelennek meg, az inferior élek közül pedig csak azok tűnnek el, amelyek rész-élek lesznek.

3. Ha $v \in P$ pont, tehát vagy párokkal volt telítve, vagy egy új (0 kapacitású) pont, akkor a kapacitásának növelését követően v ajánlatot fog tenni minden szomszédjának, akik eddig inferior élekkel voltak összekötve vele (a superior élű szomszédjaival nem érdemes próbálkozni).

   Ekkor két eset lehetséges: ha senki sem fogadta el a v pont ajánlatát, akkor ez azt jelenti, hogy minden megkérdezett x pontra (x, v) él is inferior volt $b\Pi_v$-ben, tehát az eredeti stabil b-partíció stabil marad, tehát $b\Pi = b\Pi_v$, és a v pont átkerül a P ponthalmazból a telítetlen pontok T halmazába. A másik lehetőség, hogy létezik x pont, amelyik elfogadja v ajánlatát. Ekkor aszerint, hogy x milyen ponthalmazba tartozott, a következő esetek lehetségesek:

   (a) Ha x telítetlen pont volt, tehát $x \in T$. Ekkor x és v pár lesz, tehát $b\Pi := b\Pi_v \cup \{< v, x >\}$ stabil b-partíció lesz G gráfban, és v pont marad P halmazban, x pont pedig attól függően kerül át P halmazba, hogy a kapacitása az új párral telítődik-e.

   (b) Ha x része volt egy $A =< a_1, a_2, \ldots, a_{2m}, x(= a_{2m+1}) >$ páratlan ciklusnak, tehát $x \in C$ esetében a ciklus hasonlóképpen bomlik fel, mint az eredeti Tan-Hsueh algoritmusban.

   $$b\Pi := b\Pi_v \setminus A \cup \{< a_1, a_2 >, \ldots, < a_{2m-1}, a_{2m} >, < v, x >\}$$

   stabil b-partíció lesz G-ben.

   (c) Az utolsó eset, hogy $x \in P$, vagyis x pont párokkal volt telítve (kapacitással nem rendelkező pontok nem fogadhatnak ajánlatot), és x pont jobb ajánlatot kapott, mint a legrosszabb eddigi párja (jelöljük $r(x)$-vel), tehát x pont párt fog alkotni v-vel és felmondja kapcsolatát $r(x)$-el. Ez után már $r(x)$ pont fog megpróbálni új párt keresni magának.

Tehát, csak abban az esetben nem ér véget rögtön az algoritmus egy üteme, ha mind az a pont, aki az ajánlatot tette, mind pedig az, aki ezt elfogadta P-ben volt. Ebben az esetben viszont, ha $r(x)$ kapacitását 1-el csökkentenénk, akkor $b\Pi_{r(x)} := b\Pi_v \setminus \{< x, r(x) >\} \cup \{< v, x >\}$ stabil b-párosítás lenne $G' := (G_{v^-})_{r(x)^+}$ gráfon.

Az algoritmus a fenti gondolatmenetből adódik. A bizonyításhoz hasonlóképpen definiálhatjuk az alternatív sorozatokat, ahogy a 2.14 Definícióban tettük.

Definíció 3.16. Adott egy $b\Pi_0$ stabil b-partíció a $G_{\alpha_0^-}$ gráfon. Ekkor az $\alpha_0, \beta_1, \alpha_1, \beta_2, \alpha_2, \ldots, \beta_k, \alpha_k$ pont-sorozatot *alternáló b-sorozatnak* nevezzük, ha $G_{\alpha_i^-}$ gráfhoz tartozó stabil b-partíciók $b\Pi_i$ sorozatára $i = 1, 2, \ldots, k$ teljesül, hogy $b\Pi_{i+1} = b\Pi_i - \{< \beta_{i+1}, \alpha_{i+1} >\} \cup \{< \alpha_i, \beta_{i+1} >\}$.

Ezen definíció segítségével teljesen hasonló módon igazolhatjuk az alábbi tételt, mint ahogy Tan és Hsueh tette.

Tétel 3.17. *Legyen* $\alpha_0, \beta_1, \alpha_1, \ldots, \beta_k, \alpha_k$ *egy alternáló b-sorozat, ahol a* β_{k+1}*-nél visszatérés van. Ekkor az alternáló b-sorozat kiterjeszthető úgy, hogy* β_{k+m+1}*-nél a sorozat visszatér* α_k*-hoz és ezen második visszatérő sorozatra a két alábbi állítás is igaz:*

1. $\alpha_k, \beta_{k+1}, \alpha_{k+1}, \ldots, \beta_{k+m}, \alpha_{k+m}$ $2m+1$ *darab különböző pont.*

2. $A =< \alpha_{k+m}, \beta_{k+m}, \alpha_{k+m-1}, \beta_{k+m-1}, \ldots, \alpha_{k+1}, \beta_{k+1}, \alpha_k >$ *egy páratlan ciklust alkot, vagyis* $b\Pi = b\Pi_k \setminus \{< \beta_i, \alpha_i > | i = k+1 \ldots k+m\} \cup \{A\}$ *stabil b-partíció az egész G gráfra.*

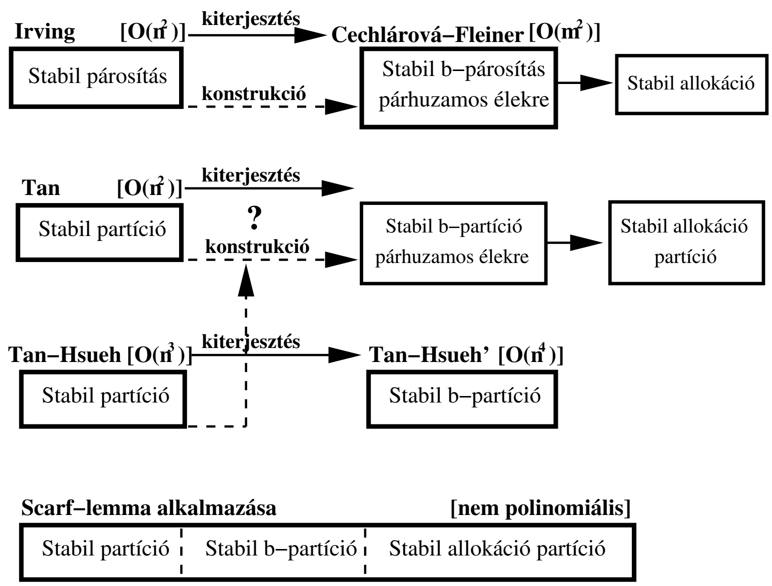
A tétel bizonyításához a Tan-Hsueh algoritmus igazolásához használt állítások ebben az esetben is mind igazak, csupán lépés helyett ütemről beszélünk, az izolált pontok megfelelnek a telítetlen pontoknak, a ciklus-beliek a C-belieknek, a pár-beliek pedig a P-belieknek. Az alternáló sorozat átírandó alternáló b-sorozatra és a pont párja, helyett a pont legrosszabb párja lesz az, akit elhagy a pont. A bizonyítások ismételt leírásától e miatt eltekinthetünk.

Az algoritmus lépésszáma egy ütemben $(O(n^2)$, tehát ha a kapacitások összege k, vagyis $\sum_{i=1}^{n} b(v_i) = k$, akkor az algoritmus lépésszáma $O(n^2 k)$, ami általános esetben tekinthető $O(n^4)$-nek.

Az algoritmus gyakorlati megvalósítására példát a mellékletben találhat az Olvasó. Megjegyzem, hogy az algoritmus általam írt változata a kapacitásokat sorban, pontonként tölti fel. Ez természetesen másképp is megvalósítható. Emellett még annyiban gyorsítottam a lépéseket, hogy amennyiben a kapacitását éppen növelő pont kap ajánlatot olyan ponttól, akihez superior él megy belőle, akkor az ajánlat elfogadása után nem utasítja el legrosszabb párját, hanem növeli kapacitását 1-el.

Konklúzió

Dolgozatomban a gráfok stabil b-párosításának problémáját igyekeztem összefoglalni. Bevezetésként ismertettem a stabil párosítás alapproblémáját páros esetre, majd ennek kiterjesztését b-párosításra. Nem páros gráfok esetén először Irving algoritmusát ismertettem, majd megemlítettem Tan módosítását, amellyel stabil partíciót is találhatunk. Részletesen bemutattam a Tan-Hsueh algoritmust, amely dinamikusan oldja meg az előző problémát. A Scarf-lemma segítségével a stabil partíció mellett a b-partíció, és az allokáció partíció problémáját is kezelni tudjuk. Sőt ennek segítségével karakterizáltuk a stabil b-partíció megoldását. Ezután Cechlárová és Fleiner konstrukción alapuló visszavezetését ismertettem, amellyel a stabil b-párosítás kérdésköre párhuzamos éleket tartalmazó gráfokon is megoldhatóvá válik. Végül bemutattam a Tan-Hsueh algoritmus egy lehetséges kiterjesztését a stabil b-partíció probléma megoldására. A fenti gondolatmenetet összefoglalása található meg a következő ábrán.



Ábra 8: Összefoglalás

Az ábrán található kérdőjel jelenti az egyik konkrét átgondolandó kérdést, amit a 3.14 megjegyzésben már taglaltam. Ha esetleg ennek megoldásával sikerül visszavezetni a stabil b-partíció kérdését a b-partíció kérdésére, akkor talán hasonlóképpen kaphatunk a Tan algoritmus kiterjesztésével egy, a Tan-Hsueh kiterjesztésénél hatékonyabb algoritmust a stabil b-partíció problémájára. A kérdéskör még számos további általánosítási lehetőséget rejt. Az egyik ezek közül a preferencia-egyezések kérdése. Itt – ahogy az egyetemi felvételi rendszer

működésében előfordulhat, hogy két felvételizőnek ugyanannyi a pontszáma – lehet, hogy két él azonos szinten van a pont preferenciájában, tehát a rendezés nem szigorú. Ennek a problémának van olyan értelmezése, amely bizonyítottan NP-teljes. A kérdéskörrel, nem páros gráfok esetén Irving és Manlove [9] foglalkozott, de b-párosítás esetében a problémának még nem ismert leírása.
Végül szintén érdemes utánnajárni, hogy a stabil b-párosítás, és a stabil allokáció problémájának milyen valódi alkalmazási lehetőségei lehetnek a matematikában vagy más tudományterületeken, és a gyakorlatban.

Irodalom

[1] Ron Aharoni and Tamás Fleiner. On a lemma of scarf. *J. Algorithms*, 87(1):72–80, 2003.

[2] Mourad Baïou and Michel Balinski. The stable allocation (or ordinal transportation) problem. *Math. Oper. Res.*, 27(3):485–503, 2002.

[3] Katarína Cechlárová and Tamás Fleiner. On a generalization of the stable roommates problem. Technical report, EGRES report TR-2003-2, ISSN 1587-4451, May 2003. `http://www.cs.elte.hu/egres`.

[4] Tamás Fleiner. On the stable b-matching polytope. Technical report, EGRES report TR-2002-03, June 2002. `http://www.cs.elte.hu/egres`.

[5] Tamás Fleiner. Some results on stable matchings and fixed points. Technical report, EGRES report TR-2002-8, ISSN 1587-4451, December 2002. `http://www.cs.elte.hu/egres`.

[6] Tamás Fleiner. Stable and crossing structures, August, 2000. PhD dissertation, `http://www.renyi.hu/~fleiner`.

[7] D. Gale and L.S. Shapley. College admissions and stability of marriage. *Amer. Math. Monthly*, 69(1):9–15, 1962.

[8] Robert W. Irving. An efficient algorithm for the "stable roommates" problem. *J. Algorithms*, 6(4):577–595, 1985.

[9] Robert W. Irving and David F. Manlove. The stable roommates problem with ties. *J. Algorithms*, 43(1):85–105, 2002.

[10] Alvin E. Roth. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica*, 54(2):425–427, 1986.

[11] Herbert E. Scarf. The core of an N person game. *Econometrica*, 35:50–69, 1967.

[12] Jimmy J. M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *J. Algorithms*, 12(1):154–178, 1991.

[13] Jimmy J. M. Tan and Yuang Cheh Hsueh. A generalization of the stable matching problem. *Discrete Appl. Math.*, 59(1):87–102, 1995.

4 Melléklet

A melléklet összesen 5 algoritmus futásának kivonatát tartalmazza. A honlapomon ennek kétszerese szerepel, ugyanis minden programnak megtalálható a randomizált, és a manuális input-megadási lehetősége. A vizsgált problémák és a megoldáshoz használt algoritmusok szerint a honlapon szereplő programok a következők (vastagítással jelölöm azokat a programokat, amelyek kivonata szerepel a Mellékletben):

Probléma	Megadás	Tan-Hsueh	Scarf
Stabil partíció	véletlen	**thran.mws**	**scran.mws**
	adott	thadott.mws	scadott.mws
Stabil b-partíció	véletlen	bthran.mws	bscran.mws
	adott	**bthadott.mws**	**bscadott.mws**
Stabil allokáció	véletlen	-	ballocran.mws
	adott	-	**ballocadott.mws**

A problémák futása gyakorlatilag két alapgráfon kerül bemutatásra. Mindkettő 6 pontú, az egyik véletlenszerűen, a másik megadás útján keletkezett. Az ismertetett három probléma közül az első kettő, mind a Tan-Hsueh és ennek kiterjesztett változatával, mind pedig a Scarf algoritmussal meg van oldva. Érdekes összehasonlítani a két program futását ugyanazon az inputon.

Különlegesség, hogy a példák olyanok, hogy a problémák általánosításával váltakozva tartalmaznak ciklusokat. Tehát például az első gráfban alapesetben található páratlan ciklus (nem létezik stabil párosítás benne), de a pontkapacitásos esetben már nincs benne (létezik stabil b-párosítás benne), és végül az élkapacitásokat is figyelembe véve ismét lesz benne páratlan ciklus (nem létezik stabil allokáció benne). A másik gráf esetén pedig mindez pont fordítva van.

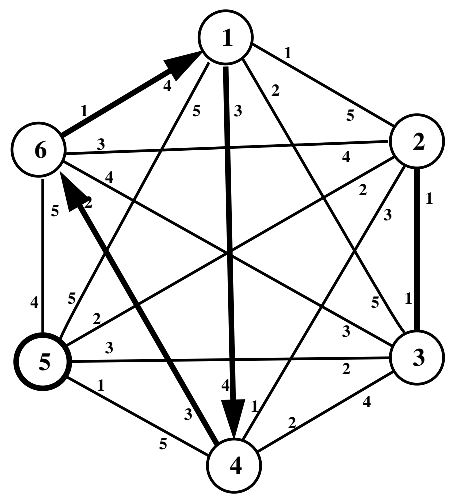
A kivonatok minden esetben tartalmazzák a kiinduló inputot, a lépések pontos leírását, és az ábrákon történő szemléltetését. A Scarf-algoritmus i. lépését leíró T_i mátrix második sora az x megoldásvektort mutatja, a nem-nulla elemek jelentik x tartóját, ezek a pontok és élek az ábrákon megvastagítva szerepelnek. Az e jelentése, hogy az adott pont vagy él súlya kicsi, de része a tartónak, ezt a gráfon szaggatott éllel jelölöm. A T_i mátrix harmadik sora azt jelenti, hogy az adott pont, vagy él melyik sorban dominál. A Scarf-algoritmus első alkalmazásánál a B és C mátrixokat nem csak a kezdőállapotukban ábrázolom. Bekeretézéssel jelölöm B mátrix esetén az aktuális bázis-vektorokat, C mátrix esetén pedig a sorminimumokat.

4.1 Stabil partíció keresése gráfokon

A gráf felépítése és preferenciái véletlenszerűen generálódtak ebben az esetben. A keletkezett 6 pontú teljes gráf pontjainak preferenciái a következők:

Személy	Preferencia lista
1	$[2, 3, 4, 6, 5]$
2	$[3, 5, 4, 6, 1]$
3	$[2, 5, 6, 4, 1]$
4	$[2, 3, 6, 1, 5]$
5	$[4, 2, 3, 6, 1]$
6	$[1, 4, 2, 3, 5]$

A keresett stabil partíció pedig a következő lesz:



Ábra 9: Stabil partíció a gráfban

Ezt először a Tan-Hsueh algoritmussal, majd a Scarf algoritmussal is megkeressük.

4.1.1 Tan-Hsueh algoritmussal

Az algoritmus lépései a következők:

1. Lépés:

1 ajánlatát senki sem fogadja el, 1 izolált pont lesz.

Stabil partíció: $< 1 >$

2. Lépés:

2 ajánlatát 1 elfogadja, 1 egyedül volt, a lépés véget ér.

Stabil partíció: $< 1, 2 >$

3. Lépés:

3 ajánlatát elsőként 2 fogadja el, 4 párja 1 egyedül marad,
1 ajánlatát senki sem fogadja el, 1 izolált pont lesz.

Stabil partíció: $< 1 >, < 2, 3 >$

4. Lépés:

4 ajánlatát elsőként 1 fogadja el, 1 egyedül volt, a lépés véget ér.

Stabil partíció: $< 1, 4 >, < 2, 3 >$

5. Lépés:

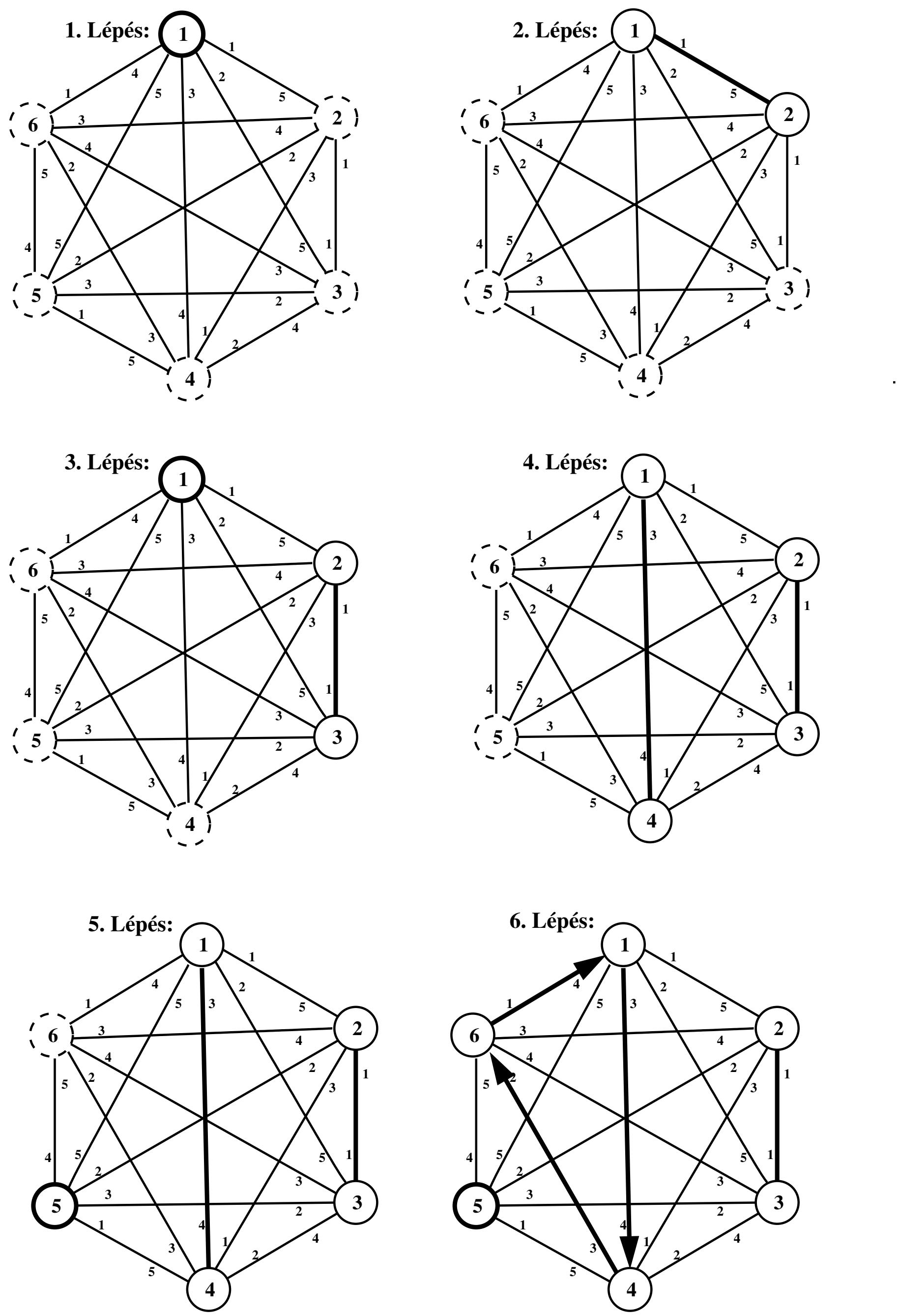
5 ajánlatát senki sem fogadja el, 5 izolált pont lesz.

Stabil partíció: $< 5 >, < 1, 4 >, < 2, 3 >$

6. Lépés:

6 ajánlatát 4 fogadja el elsőként, 4 párja 1 egyedül marad,
1 ajánlatát 6 fogadja el elsőként, 6 párja 4 egyedül marad,
4 ajánlatát 1 fogadja el elsőként, 1 párja 6 egyedül marad.

Stabil partíció: $< 5 >, < 2, 3 >, < 1, 6, 4 >$



Ábra 10: Tan-Hsueh algoritmus

4.1.2 Scarf algoritmussal

STABIL és PIVOT lépések váltakozó használatával 8 lépésben jutunk el egy új közös bázishoz.

Alapbázisok: $BS_0 = BP_0 = [1, 2, 3, 4, 5, 6]$, és a pont-kapacitás: $b = [1, 1, 1, 1, 1, 1]$

A B mátrix a kezdeti állapotban a következő:

$$B_0 = \begin{bmatrix} \boxed{1} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

A C mátrix:

$$C_0 = \begin{bmatrix} \boxed{0} & 25 & 24 & 23 & 22 & 21 & 5 & 4 & 3 & 1 & 2 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 \\ 26 & \boxed{0} & 24 & 23 & 22 & 21 & 1 & 19 & 18 & 17 & 16 & 5 & 3 & 4 & 2 & 11 & 10 & 9 & 8 & 7 & 6 \\ 26 & 25 & \boxed{0} & 23 & 22 & 21 & 20 & 1 & 18 & 17 & 16 & 5 & 14 & 13 & 12 & 2 & 4 & 3 & 8 & 7 & 6 \\ 26 & 25 & 24 & \boxed{0} & 22 & 21 & 20 & 19 & 2 & 17 & 16 & 15 & 5 & 13 & 12 & 4 & 10 & 9 & 1 & 3 & 6 \\ 26 & 25 & 24 & 23 & \boxed{0} & 21 & 20 & 19 & 18 & 1 & 16 & 15 & 14 & 4 & 12 & 11 & 3 & 9 & 5 & 7 & 2 \\ 26 & 25 & 24 & 23 & 22 & \boxed{0} & 20 & 19 & 18 & 17 & 5 & 15 & 14 & 13 & 3 & 11 & 10 & 2 & 8 & 4 & 1 \end{bmatrix}$$

1. Lépés: Az első lépés speciális, kimozdítjuk a rendszert a stabil helyzetéből!

$$T_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 1 & 0 & e & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

STABIL lépés: Ki: 1, Be: 12

$$C_1 = \begin{bmatrix} \mathbf{0} & 25 & 24 & 23 & 22 & 21 & 5 & 4 & 3 & 1 & 2 & \boxed{15} & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 \\ 26 & \boxed{0} & 24 & 23 & 22 & 21 & 1 & 19 & 18 & 17 & 16 & 5 & 3 & 4 & 2 & 11 & 10 & 9 & 8 & 7 & 6 \\ 26 & 25 & \boxed{0} & 23 & 22 & 21 & 20 & 1 & 18 & 17 & 16 & 5 & 14 & 13 & 12 & 2 & 4 & 3 & 8 & 7 & 6 \\ 26 & 25 & 24 & \boxed{0} & 22 & 21 & 20 & 19 & 2 & 17 & 16 & 15 & 5 & 13 & 12 & 4 & 10 & 9 & 1 & 3 & 6 \\ 26 & 25 & 24 & 23 & \boxed{0} & 21 & 20 & 19 & 18 & 1 & 16 & 15 & 14 & 4 & 12 & 11 & 3 & 9 & 5 & 7 & 2 \\ 26 & 25 & 24 & 23 & 22 & \boxed{0} & 20 & 19 & 18 & 17 & 5 & 15 & 14 & 13 & 3 & 11 & 10 & 2 & 8 & 4 & 1 \end{bmatrix}$$

PIVOT lépés: Be: 12, Ki: 2

$$B_1 = \begin{bmatrix} \boxed{1} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & \boxed{1} & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

<u>2. Lépés:</u>

$$T_2 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & e & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

STABIL lépés: Ki: 2, Be: 16

$$C_2 = \begin{bmatrix} 0 & 25 & 24 & 23 & 22 & 21 & 5 & 4 & 3 & 1 & 2 & \underline{15} & 14 & 13 & 12 & \boxed{11} & 10 & 9 & 8 & 7 & 6 \\ 26 & \mathbf{0} & 24 & 23 & 22 & 21 & 1 & 19 & 18 & 17 & 16 & \boxed{5} & 3 & 4 & 2 & 11 & 10 & 9 & 8 & 7 & 6 \\ 26 & 25 & \boxed{0} & 23 & 22 & 21 & 20 & 1 & 18 & 17 & 16 & 5 & 14 & 13 & 12 & 2 & 4 & 3 & 8 & 7 & 6 \\ 26 & 25 & 24 & \boxed{0} & 22 & 21 & 20 & 19 & 2 & 17 & 16 & 15 & 5 & 13 & 12 & 4 & 10 & 9 & 1 & 3 & 6 \\ 26 & 25 & 24 & 23 & \boxed{0} & 21 & 20 & 19 & 18 & 1 & 16 & 15 & 14 & 4 & 12 & 11 & 3 & 9 & 5 & 7 & 2 \\ 26 & 25 & 24 & 23 & 22 & \boxed{0} & 20 & 19 & 18 & 17 & 5 & 15 & 14 & 13 & 3 & 11 & 10 & 2 & 8 & 4 & 1 \end{bmatrix}$$

PIVOT lépés: Be: 16, Ki: 3

$$B_2 = \begin{bmatrix} \boxed{1} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & \boxed{1} & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & \boxed{1} & 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

<u>3. Lépés:</u>

$$T_3 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & e & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4. Lépés:

$$T_4 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 1 & 0 & 0 & 0 & e & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & e & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 \end{bmatrix}$$

5. Lépés:

$$T_5 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 1 & 0 & 0 & 0 & 1 & e & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & e & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 4 & 1 & 0 \end{bmatrix}$$

6. Lépés:

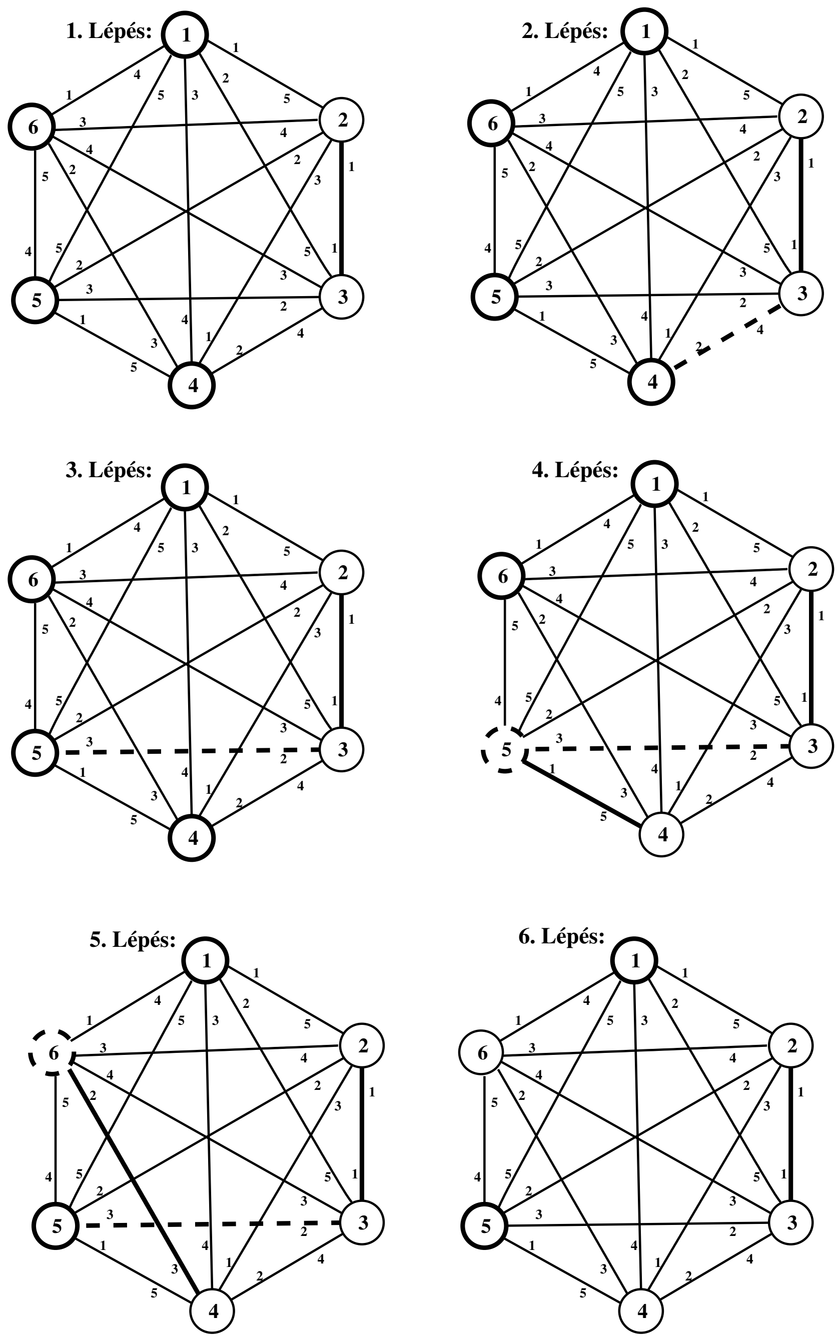
$$T_6 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & e & 0 & 0 & 1 & e \\ 0 & 0 & 0 & 0 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 4 & 1 \end{bmatrix}$$

7. Lépés:

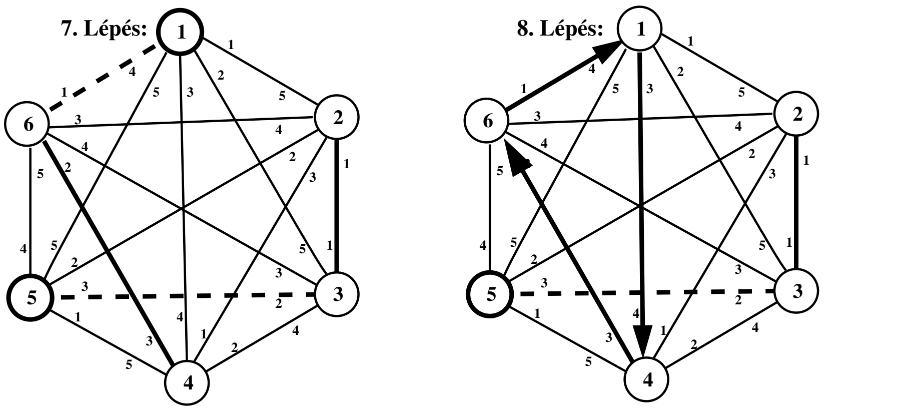
$$T_7 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & e & 1 & 0 & 0 & 0 & 0 & e & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 4 & 6 \end{bmatrix}$$

8. Lépés:

$$T_8 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 1 & 0 & 0 & 0 & 0 & e & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 4 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 6 & 0 \end{bmatrix}$$



Ábra 11: Stabil partíció keresése Scarf algoritmussal 1.



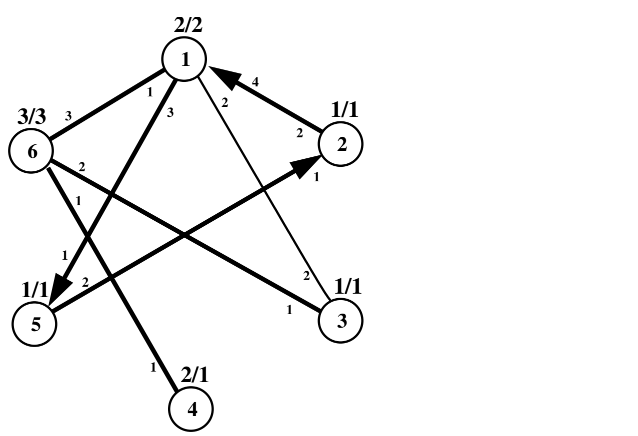
Ábra 12: Stabil partíció keresése Scarf algoritmussal 2.

4.2 Stabil b-partíció keresése nem páros gráfon

Adott egy 6 pontú gráf a preferencia listáival és $b = [2, 1, 1, 2, 1, 3]$ kapacitással:

Személy	Preferencia lista
1	$6, 3, 2, 5$
2	$5, 1$
3	$6, 1$
4	6
5	$1, 2$
6	$4, 3, 1$

A stabil b-partíció a következő lesz:



Ábra 13: Stabil b-partíció: $< 4 >, < 1, 6 >, < 3, 6 >, < 4, 6 >, < 1, 2, 5 >$

4.2.1 Tan-Hsueh kiterjesztett változatával

Az algoritmus lépései a következők:

1. Lépés:

1.ütem: 1 ajánlatát senki sem fogadja el, 1 telítetlen pont lesz.

Stabil b-partíció: $< 1 >$

2. Lépés:

1.ütem: 2 ajánlatát 1 elfogadja, 1 telítetlen volt, a lépés véget ér.

Stabil b-partíció: $< 1, 2 >$

3. Lépés:

1.ütem: 3 ajánlatát elsőként 1 fogadja el, 1 telítetlen volt, a lépés véget ér.

Stabil b-partíció: $< 1, 2 >, < 1, 3 >$

4. Lépés:

1.ütem: 4 ajánlatát senki sem fogadja el, 4 telítetlen pont lesz.

Stabil b-partíció: $< 4 >, < 1, 6 >, < 3, 6 >, < 4, 6 >, < 1, 2, 5 >$

5. Lépés:

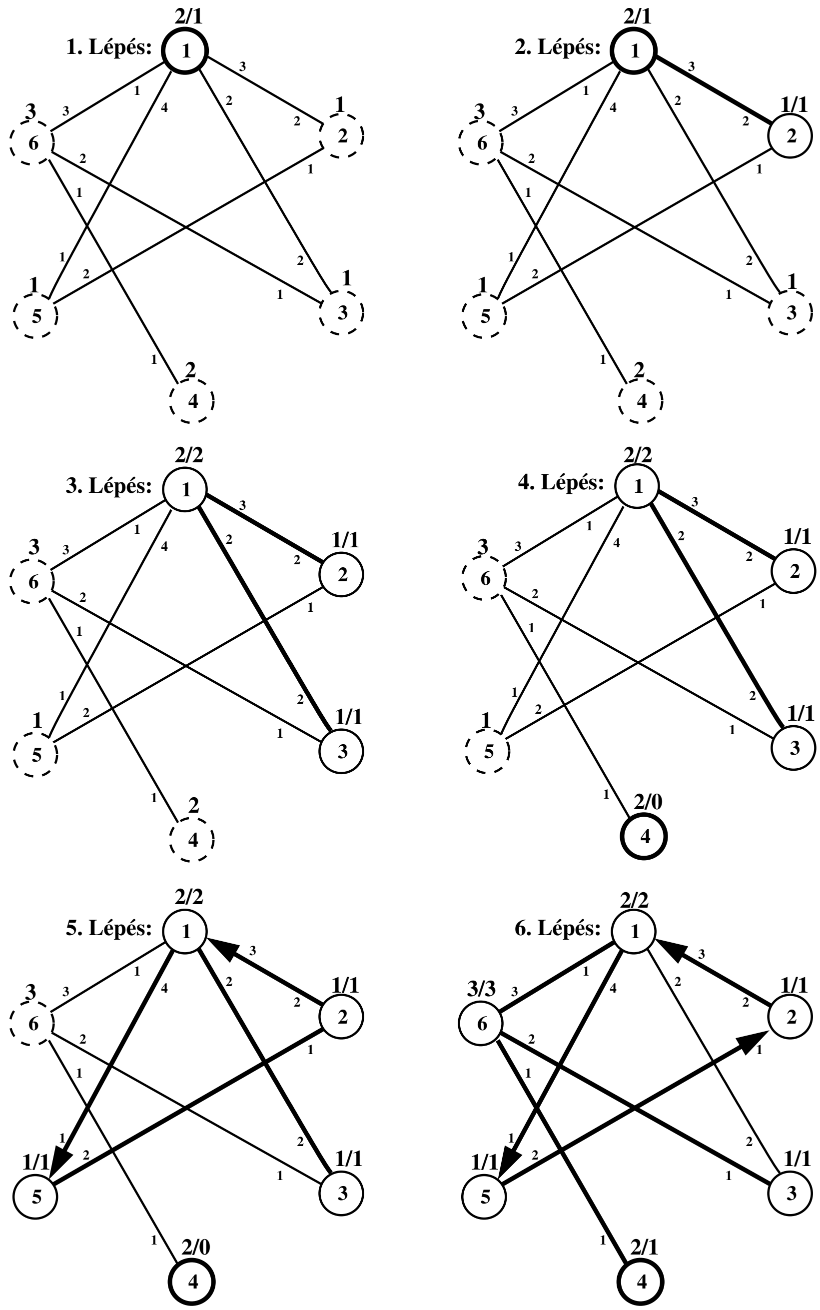
1.ütem: 5 ajánlatát 2 fogadja el elsőként, 2 legrosszabb párja 1 telítetlen marad,
1.ütem: 1 ajánlatát 5 fogadja el elsőként, 5 legrosszabb párja 2 telítetlen marad,
1.ütem: 2 ajánlatát 1 fogadja el elsőként, 1 legrosszabb párja 5 telítetlen marad.

Stabil b-partíció: $< 4 >, < 1, 3 >, < 1, 2, 5 >$

6. Lépés:

1.ütem: 6 ajánlatát 4 fogadja el elsőként, 4 telítetlen volt, új ütem,
2.ütem: 6 ajánlatát 3 fogadja el elsőként, 3 legrosszabb párja 1 telítetlen marad,
2.ütem: 1 párba lép 2-vel és 1 párba lép 5-el, új ütem.
3.ütem: 6 ajánlatát 1 fogadja el elsőként, 1 legrosszabb párja 5 egyedül marad.
3.ütem: 5 ajánlatát 2 fogadja el elsőként, 2 legrosszabb párja 1 telítetlen marad,
3.ütem: 1 ajánlatát 5 fogadja el elsőként, 5 legrosszabb párja 2 telítetlen marad,
3.ütem: 2 ajánlatát 1 fogadja el elsőként, 1 legrosszabb párja 5 telítetlen marad.

Stabil b-partíció: $< 4 >, < 1, 6 >, < 3, 6 >, < 4, 6 >, < 1, 2, 5 >$



Ábra 14: Tan-Hsueh algoritmus b-párosításra

4.2.2 Scarf algoritmussal

$$B_0 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

$$C = \begin{bmatrix}
0 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 3 & 4 & 2 & 5 & 8 & 7 & 6 \\
25 & 0 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 4 & 11 & 10 & 9 & 5 & 7 & 6 \\
25 & 24 & 0 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 4 & 10 & 9 & 8 & 5 & 6 \\
25 & 24 & 23 & 0 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 5 \\
25 & 24 & 23 & 22 & 0 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 5 & 9 & 4 & 7 & 6 \\
25 & 24 & 23 & 22 & 21 & 0 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 3 & 8 & 4 & 5 \\
25 & 24 & 23 & 22 & 21 & 20 & 0 & 18 & 17 & 16 & 15 & 14 & 13 & 1 & 11 & 10 & 9 & 8 & 7 & 6 \\
25 & 24 & 23 & 22 & 21 & 20 & 19 & 0 & 17 & 16 & 15 & 14 & 13 & 12 & 1 & 10 & 9 & 8 & 7 & 6 \\
25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 0 & 16 & 15 & 14 & 13 & 12 & 11 & 1 & 9 & 8 & 7 & 6 \\
25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 0 & 15 & 14 & 13 & 12 & 11 & 10 & 1 & 8 & 7 & 6 \\
25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 0 & 14 & 13 & 12 & 11 & 10 & 9 & 1 & 7 & 6 \\
25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 0 & 13 & 12 & 11 & 10 & 9 & 8 & 1 & 6 \\
25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 0 & 12 & 11 & 10 & 9 & 8 & 7 & 1
\end{bmatrix}$$

<u>1. Lépés</u>

$$T_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 1 & 2 & e & 3 & 1 & 1 & 1 & 1 & e & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

2. Lépés

$$T_2 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 0 & 2 & e & 2 & 1 & 1 & 1 & 1 & e & e & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \end{bmatrix}$$

3. Lépés

$$T_3 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 0 & 1 & e & 1 & 1 & 1 & 1 & 1 & e & e & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 0 & 0 & 0 & 2 & 3 & 1 \end{bmatrix}$$

4. Lépés

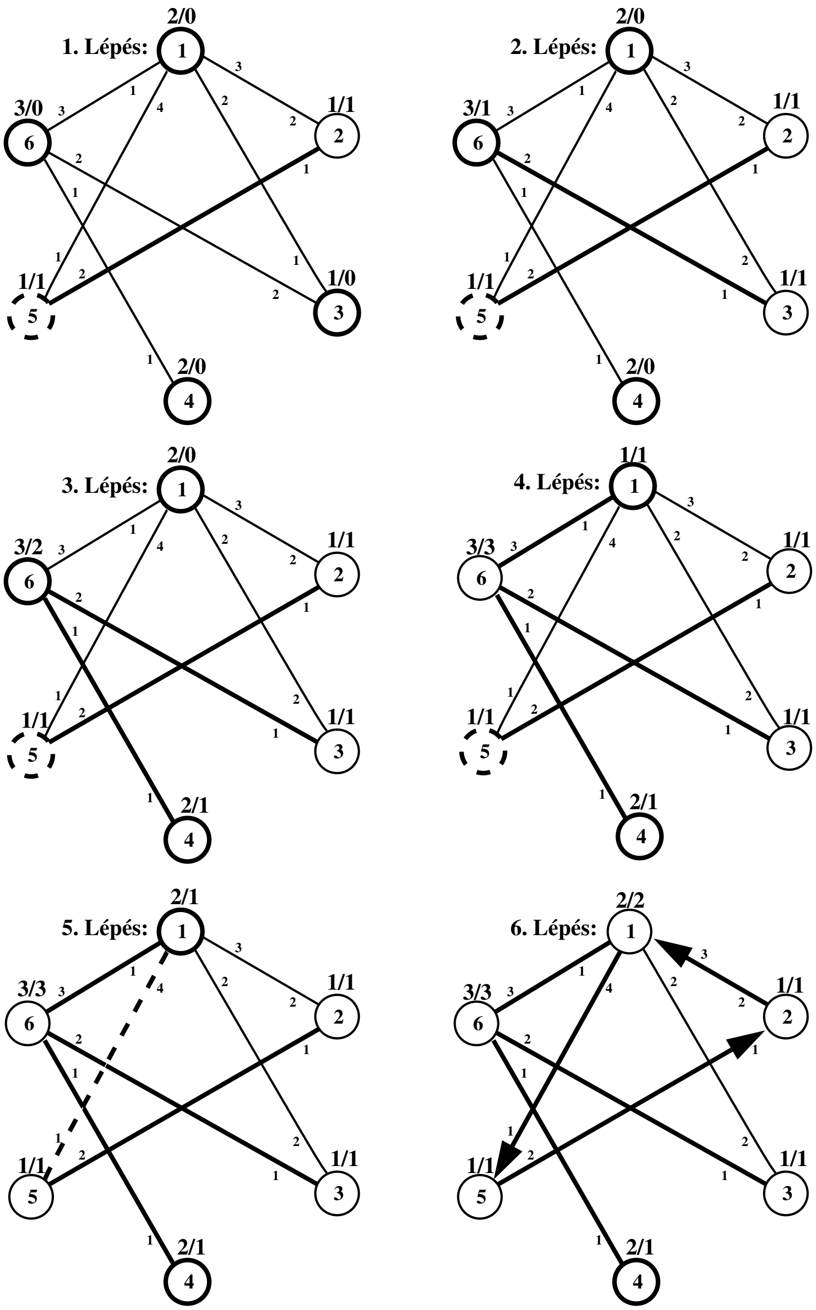
$$T_4 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 1 & 0 & 0 & 1 & e & 0 & 1 & 1 & 1 & e & e & e & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 13 \end{bmatrix}$$

5. Lépés

$$T_5 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & e & e & e & 0 & 0 & 0 & e & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 4 & 5 & 0 & 7 & 8 & 9 & 10 & 11 & 12 & 0 & 0 & 0 & 1 & 6 & 2 & 3 & 13 \end{bmatrix}$$

6. Lépés

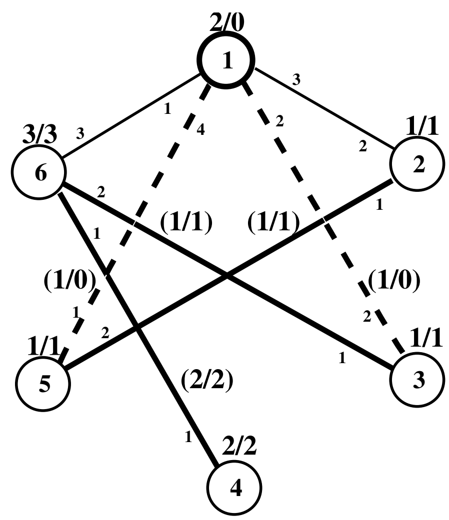
$$T_6 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1/2 & 1 & 1/2 & e & 1/2 & e & 0 & 1/2 & 0 & 1/2 & 1 & 1/2 & 1 & 1 \\ 0 & 0 & 0 & 4 & 0 & 0 & 7 & 8 & 9 & 10 & 11 & 12 & 0 & 2 & 0 & 1 & 6 & 5 & 3 & 13 \end{bmatrix}$$



Ábra 15: Scarf algoritmus b-párosításra

4.3 Stabil allokáció-partíció keresése Scarf-algoritmussal

A pont- és élkapacitások: $bo := [2, 1, 1, 2, 1, 3]$ és $be := [1, 2, 2, 3, 1, 2, 3]$



Ábra 16: Stabil allokáció

1. Lépés

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 1 & 2 & e & 3 & 1 & 2 & 2 & 3 & e & 2 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

2. Lépés

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 0 & 2 & e & 2 & 1 & 2 & 2 & 3 & e & 1 & 3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \end{bmatrix}$$

3. Lépés

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 0 & e & e & 0 & 1 & 2 & 2 & 3 & e & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 0 & 0 & 0 & 2 & 3 & 1 \end{bmatrix}$$

4. Lépés

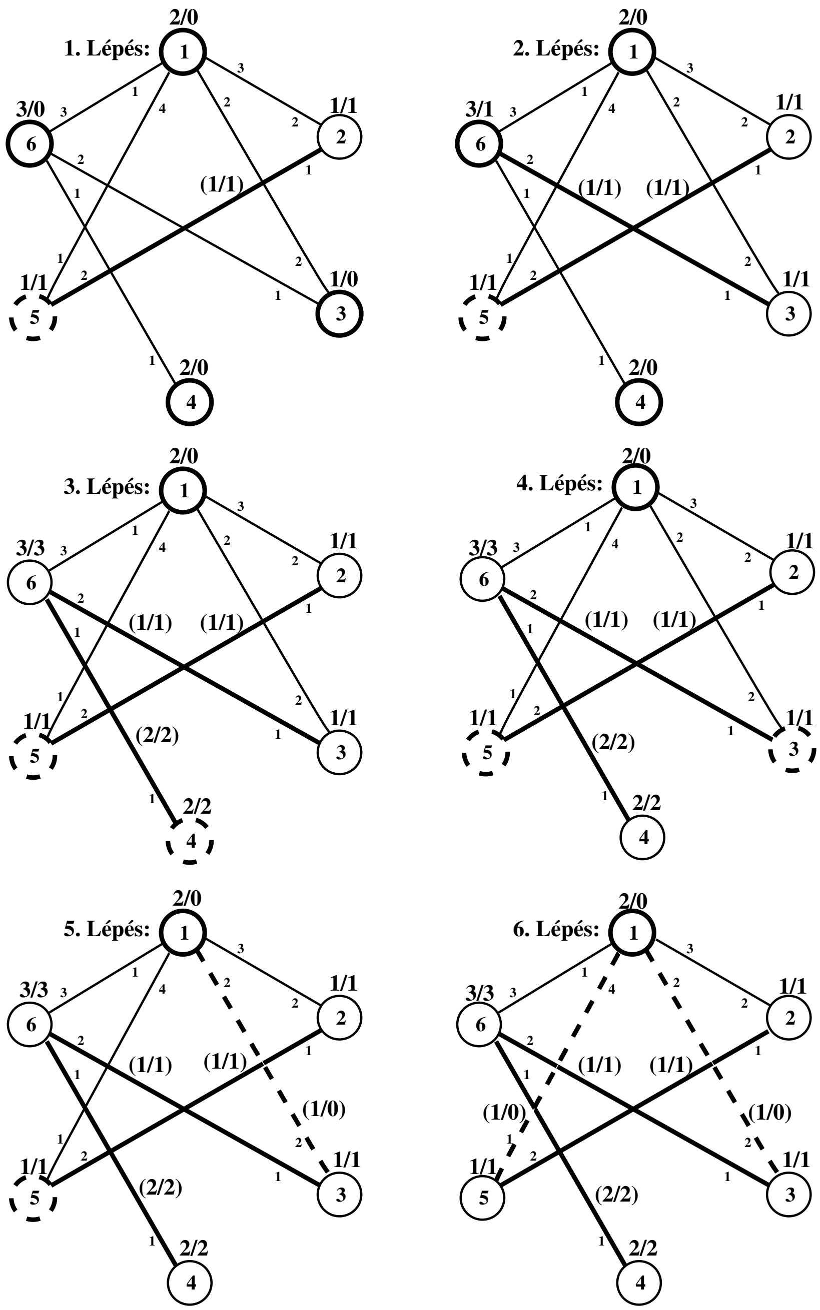
$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & e & 0 & e & 0 & 1 & 2 & 2 & 3 & e & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 3 & 4 & 5 & 0 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 0 & 0 & 0 & 2 & 6 & 1 \end{bmatrix}$$

5. Lépés

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 0 & 0 & e & 0 & 1 & 2 & 2 & 3 & e & 1 & 1 & 0 & e & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 3 & 0 & 5 & 0 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 1 & 0 & 0 & 2 & 6 & 4 \end{bmatrix}$$

6. Lépés

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 3 & e & 1 & 1 & 0 & e & e & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 5 & 0 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 0 & 3 & 1 & 0 & 2 & 6 & 4 \end{bmatrix}$$



Ábra 17: Stabil allokáció keresése a Scarf algoritmussal