Eötvös Loránd Tudományegyetem
Természettudományi Kar

Vesecserére vonatkozó algoritmusok

SZAKDOLGOZAT

MATEMATIKA ALAPSZAK, ALKALMAZOTT MATEMATIKUS SZAKIRÁNY

Készítette
Dubán Dorina

Konzulens
Jankó Zsuzsanna

2015. május 19.

Tartalomjegyzék

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Jankó Zsuzsannának, aki szakértelmével, hasznos magyarázataival és a konzultációk során biztosított elengedhetetlen tanácsaival hatalmas segítséget nyújtott szakdolgozatom elkészüléséhez.
Hálával tartozom továbbá szüleimnek, testvéremnek és páromnak, akik nélkül ez a szakdolgozat nem jöhetett volna létre. Köszönöm nekik, hogy tanulmány során türelemmel és megértéssel támogattak, és minden helyzetben mellettem álltak.

Bevezető

Statisztikák szerint hazánkban 6-10 ezren járnak krónikus művesekezelésre. Közülük 700-an vannak transzplantációs várólistán. Évente körülbelül 300 átültetés történik. A számok is jól mutatják, hogy sokkal többen várnak vesére, mint ahányan bekerülnek a műtőbe. Önkéntesként sok vesére váró - vagy jobb esetben vese transzplantált gyermekkel találkozom.

A beültetendő szerv két forrásból származhat: élő és agyhalott donoroktól. A feltételezett beleegyezés elve szerint; ha életünkben nem tiltottuk meg, hogy szerveinket mások életének megmentése érdekében felhasználják, akkor halála esetén eltávolíthatóak a szervei.

Az élő donortól kapott vese beültetése orvosi szempontból optimálisabb. Míg agyhalott donortól kapott vesével a páciens 8-9 évig él, addig élő donortól kapott szerv esetén 15-20 évig. Mindkét esetben a túlélés esélye 90 % körül mozog. Ám sajnálatos módon a vese jóval ritkábban származik élő donortól. Egy közeli barátnak, rokonnak biztos sokan segítenének, akár azzal, hogy egy veséjüktől megválnak - ugyanis egy egészséges vesével még lehetett teljes életet élni. Sajnos a dolog nem ilyen egyszerű, hiszen a vesék nem teljesen kompatibilisek. A veseátültetés immunológiai feltételeit a későbbiekben részletesen tárgyalom. A "United Network for Organ Sharing" nevű program erre talált megoldást. Ha egy személy odaadná veséjét egy rokonának, de a vese nem megfelelő szerettének, akkor; ha az említett donor az egyik veséjét odaadományozza egy idegennek, akinek szüksége van rá, cserébe az átültetésre váró hozzátartozója is kap egy egészséges vesét. A program létrejöttében nagy szerepe volt Sommer Gentry, fiatal matematikusnőnek. A Harvard egyetemen 2002-től használtak egy algoritmust (TTCC) a donorok és a betegek párosítására, de tovább kívánták fejleszteni. A matematikusnő konstruált egy gráfot, mely csúcsai az inkompatibilis párok és élek akkor és csak akkor futnak két csúcs között, ha a csúcsokat jelentő párok közt lehetséges a vesecsere. Ezután a kapott gráfban kell maximális párosítást keresni, egyéb tényezőket figyelembe véve (földrajzi közelség, sürgősség, stb.)

Az első - a programon keresztül létrejött – cserére az amerikai Lebanonban, illetve St. Louis-ban került sor. Kathy Niedzwiecki Rebecca Burke-től kapott vesét. Cserébe Rebecca Burke vőlegényének (Ken Crowder) Kathy Niedzwiecki sógornője (Catherine Richard) adományozott szervet.
Ám létezik ennél sokkal hatékonyabb módszer is, ahol nem páros cserék, hanem egész csere-sorozatok jönnek létre. Tehát egy körben adják körbe a veséket. A gyakorlatban általában egy olyan donorral kezdődik, aki nem kíván cserébe vesét, így láncok jönnek létre. Ezt azért fontos megemlíteni, mert valaki visszalépésénél ebben az esetben senki sem járhat úgy, hogy vesét már adott és kiderül, hogy vesét nem is fog kapni. Gondoljunk bele ez az eset elég kellemetlenül érintené a szóban forgó donor-beteg párost, hiszen már újabb cserében sem tudnak részt venni (a donor már odaadta a vesét), de vesét sem kaptak (a visszalépés miatt). 2010-ben egy 21-es vesecsere jött létre a fent említett módon.

Akinek vesére van szüksége, vagy szívesen lenne donor, annak érdemes regisztrálnia a `www.matchingdonors.com` oldalon. Itt a beteg egy rövid leírást adhat meg magáról, a betegsége (vagy élete) történetéről. Továbbá megadhatja a vércsoportját, HLA adatait. Így a donor "böngészve" a betegek közt megtalálhatja azt, akivel kompatibilis lenne a veséje, illetve a számára legszimpatikusabb reménykedő. Az oldalnak köszönhetően jó pár vesetranszplantáció köszönhető.Illetve, ha a donor megkedvelt egy pácienst, de kiderül, hogy mégsem kompatibilis a veséje az illetővel, akkor együtt belevághatnak az előbb említett csereprogramba.

Dolgozatomban a páros cserék, és csere-sorozatok megkapásához vezető algoritmusokat hasonlítom össze. A szakdolgozat a következő módon épül fel:
A 1. fejezetben röviden bemutatom a vesecserére alkalmazható algoritmusokat. A továbbiakban ezen algoritmusokat hasonlítom össze különböző szempontok szerint. A dolgozat végén betekintést kapunk a vesecsere orvosi hátterébe is.

1. fejezet

Vesecserés algoritmusok röviden

Mint a bevezetőben említettem sokféle vesecserére vonatkozó algoritmus létezik. Ezek közül nézzük meg példák segítségével a fontosabbak, illetve pár érdekesebb algoritmus menetét!

1.1. Páros csere (pairwise exchange)

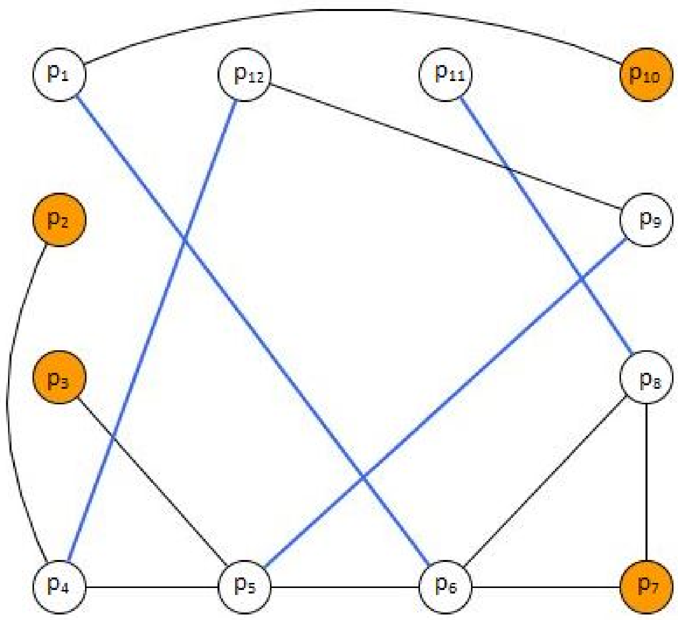
Definíció Egy gráfban M *párosítás*, ha M-ben szereplő pontok száma megegyezik az élek számának kétszeresével. Teljes párosításnak hívjuk az összes pontot lefedő párosítást. Másképp: párosításnak nevezzük azt a $\mu : N \to N$ függvényt, melyre

$$\mu(i) = j \Leftrightarrow \mu(j) = i \quad \forall i, j \epsilon N$$

Egy páros csere [9] kettő beteg-donor párt tartalmaz, az első beteg megkapja a második donornak veséjét, míg a második beteg megkapja az első donor veséjét. Ezt leírhatjuk egy gráffal, melyben párosítást keresünk. A p_i pont jelölje az i. beteget és az i. vesét is, két pont akkor legyen összekötve, ha köztük létre jöhet a csere. Így egy párosítás valóban páros cserét jelent.

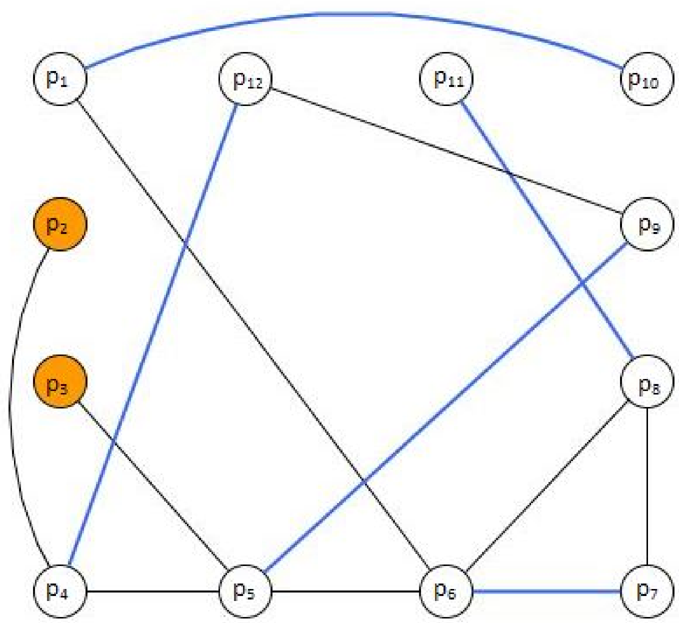
Definíció Legyen G gráf, M párosítás G-ben. Egy P utat javító útnak nevezünk egy M párosításra nézve, ha az út páratlan hosszú, kezdő és végpontjai nem párosítottak és minden páros sokadik éle eleme az M-nek, és páratlan sokadik éle nem eleme M-nek.
Javító utas algoritmus: Amíg találunk M-re vonatkozó P javító utat, addig M-et lecseréljük M'-re, ahol $M' = (M \backslash E(P)) \cup (E(P) \backslash M)$ (ahol $E(P)$ a javító út élei). Ha nem létezik M-re javító utas algoritmus, akkor M a maximális, azaz M nem bővíthető tovább.

Példa



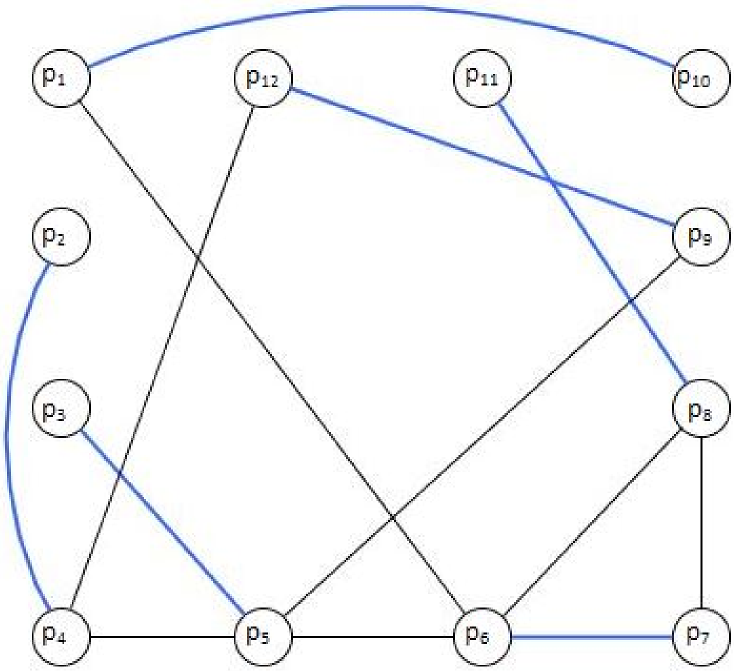
1.ábra

A kék élek egy párosítást alkotnak a gráfban. A párosítatlan pontokat sárgával jelöltük. Célunk javító út segítségével bővíteni a párosítást. Ha jobban megnézzük látjuk, hogy a $p_{10}p_1p_6p_7$ egy javító út, hiszen párosítatlan pontból indul, párosítatlan pontba végződik, és minden második éle van csak benne a párosításban. Az út mentén kicserélve a párosításban szereplő és nem szereplő éleket, kapjuk a 2.ábrát.



2.ábra

A kapott párosítást tovább tudjuk bővíteni $p_2p_4p_{12}p_9p_5p_3$ javító út segítségével, így a 3.ábrán látható teljes párosítást kapjuk.



3.ábra

1.1.1. Stabil párosítás

Most n beteg és n veséhez tartozó donor keres párt magának. A betegek nem rendelkeznek saját donorral. A betegek sorba helyezik a veséket kompatibilitásuk alapján (tegyük fel, hogy minden vesét be lehet ültetni mindenkinek, csak valamelyik jobban megfelelhet a másiknál), és hasonlóképpen minden veséhez hozzárendelünk egy preferencia-sorrendet a betegeken az alapján, hogy kihez passzolnak legjobban. Így mindannyian kialakítottak egy preferencia-sorrendet a másik csoport tagjain.

Definíció Minden beteg kialakít egy *preferencia sorrend*et az adott vesék között. Előre kerül a számára legmegfelelőbb vese, majd a sorrendjének vége felé haladva fokozatosan csökken a kompatibilitása az adott vesével. Egy vese minél hátrább van, annál kevésbé jó a páciensnek.

A stabil párosítási feladatban [13] ezen preferenciák alapján igyekszünk párokat kialakítani úgy, hogy mindenkinek a lehető legjobb legyen. Mi csak javasolhatunk nekik egy-egy párt, mind a beteg, mind a veséhez tartozó donor az általunk ideálisnak tartott párosítást elutasíthatja.
Egy adott párosításra nézve egy t beteg és egy k vese blokkoló párt alkot, akkor ha őket

összekötve, mindketten jobban éreznék magukat, mint az eredeti párjukkal. Egy párosítást stabilnak nevezünk, ha nincs benne blokkoló pár.
A Gale-Shapley [4] algoritmussal mindig találunk stabil párosítást a vesék és a betegek között:

- Mindegyik beteg megkeresi a neki legjobban megfelelő vesét beültetés céljából.
- Ha egy veséhez tartozó donort több beteg is felkeresett, akkor a donor megtartja a vesével legkompatibilisebb beteget (feltételesen), a többit pedig elutasítja.
- Minden következő lépésben a vese nélkül maradt betegek megkeresik a következő legkompatibilisebb vesét beültetés céljából (olyan donorokat, amitől még nem kapott elutasítást)
- Aki nem talált olyan vesét, ami elfogadta, az nem próbálkozik tovább.

Tétel 1 *A fenti algoritmus valóban stabil párosítást ad.*

Bizonyítás: Vegyünk egy tetszőleges $k - t$ élt a gráfban. Ha az algoritmus során t beteg megkereste a k vesét beültetés céljából, akkor a k vese ezek után mindenképp beültetésre kerül, ha nem t beteg kapja, akkor egy t-nél kompatibilisebb beteg. Ha t beteg nem kereste meg a k vesét, az azért lehet, mert talált k-nál kompatibilisebb vesét. Mindkét esetben látható, hogy $k - t$ nem blokkolja a párosítást, azaz a párosítás stabil. □

Most tekintsük a stabil párosításnak egy olyan változatát, ahol a betegek saját donorral érkeznek. Tehát valóban vesecsere történik, ha valaki kap, az ad is vesét. Ebben az esetben nem egy páros gráfunk lesz, hanem egy gráfunk, melynek egy csúcsa megegyezik egy beteggel és a hozzátartozó donorral, célunk ebben a gráfban stabil párosítást keresni. Ezt Irving algoritmusával [6] érhetjük el:

1. fázis

- Ha az i beteg megkeresi a j pácienst vesecsere céljából, akkor
  - A j beteg elutasítja i-t, ha már nála jobb ajánlatot kapott valaki mástól
  - A j beteg megtartja (feltételesen) i-t, abban az esetben ha ő az eddigi legjobb választása. Ebben az esetben a korábbi ajánlatát elutasítja.
- Egy i beteg ajánlatot tesz másoknak a preferencia listáján szereplős sorrend szerint. Ha valaki elfogadta ajánlatát, akkor abbahagyja a keresést. Ha az előbbi elfogadó beteg elutasítja ajánlatát (mert jobb ajánlatot kapott), akkor i folytatja a keresését. Ennek a fázisnak a végére:

  - Minden betegnek lesz egy (feltételes) ajánlata vagy
  - Van olyan k beteg, akit mindenki elutasított. Ebben az esetben mindenkinek van egy ajánlata, hiszen mindenki elutasította k-t. Ekkor nem létezik stabil párosítás.

- Ha az i beteg elfogadott egy kérést j-től, akkor eltávolítjuk i preferencia listájáról a j utáni betegeket, és ugyanígy j listájáról az i előtti betegeket. Tehát i-nek j lesz az utolsó, j-nek i lesz az első, mivel a kitörölt elemekkel nem lehetnek párosítva egy stabil párosításban. Már csak azt kell megmutatnunk, hogy hogyan kezeljük a csökkentett preferencia listákat, ha marad olyan lista, mely több mint egy pácienst tartalmaz. A második fázisban tovább csökkentjük a preferencia listákat.

2. fázis

- Legyen $a_1, ...a_n$ a betegeknek egy csoportra melyre a következő preferencia sorrendek állnak fenn:

$$\begin{aligned} &a_1 : b_1 \quad b_2 ... \\ &a_2 : b_2 \quad b_3 ... \\ &. \\ &. \\ &. \\ &a_n : b_n \quad b_1 ... \end{aligned}$$

  Ekkor

  - bármely stabil párosítás esetén a_i és b_i vagy minden i-re partnerek, vagy egyikre sem.
  - ha létezik olyan párosítás, amelyben a_i és b_i partnerek minden i-re, akkor olyan is létezik, amelyikben nem.

  Tehát ha létezik stabil párosítás, akkor olyan stabil párosítás is van, amelyben b_i minden i-re elutasítja a_i-t. Ekkor a_i ajánlatot tesz b_{i+1}-nek és b_{i+1} listájáról törölhetünk minden a_i-nél rosszabb jelentkezőt. Ezzel egy időben b_{i+1}-et is törölhetjük a jelentkezők listájáról.

- Ezt addig folytatjuk, amíg
  - Minden új preferencia listán egy ember szerepel vagy
  - Van olyan ember, akinek elfogynak az ajánlatot tevő lehetséges párjai. Ekkor nem létezik stabil párosítás.

1.2. Elsőbbségi algoritmus (priority algorithm)

Az elsőbbségi algoritmus [10] egy mohó algoritmus. Adott a betegek közt egy sürgősségi lista. A lista szerinti első beteget párosítsuk össze egy másik beteggel, ha kölcsönösen megfelelnek egymás donorjainak veséi egymásnak. Ha nem, akkor hagyjuk ki az első beteget. Így folytassuk az algoritmust, míg végül a lista szerinti utolsó embert párosítsuk, ha lehetséges.
Ebben az esetben is egy csere kettő beteg-donor párt tartalmaz, az első beteg megkapja a második donornak veséjét, míg a második beteg megkapja az első donor veséjét. Van egy gráfunk, melyben párosítást keresünk. p_i pont jelölje az i. beteget és az i. vesét is, két pont akkor legyen összekötve, ha köztük létre jöhet a csere. Így egy párosítás valóban páros cserét jelent. Azaz [10]

$\varepsilon^0 := M$ ahol M az összes párosítás halmaza
$\varepsilon^1 \subseteqq \varepsilon^0$ ahol

$$\varepsilon^1 = \begin{cases} \{\mu \epsilon \varepsilon^0 : \mu(1) \neq 1\} & \text{ha } \exists \mu \epsilon \varepsilon^0, \quad \text{hogy } \mu(1) \neq 1 \\ \varepsilon^0 & \text{egyébként} \end{cases}$$

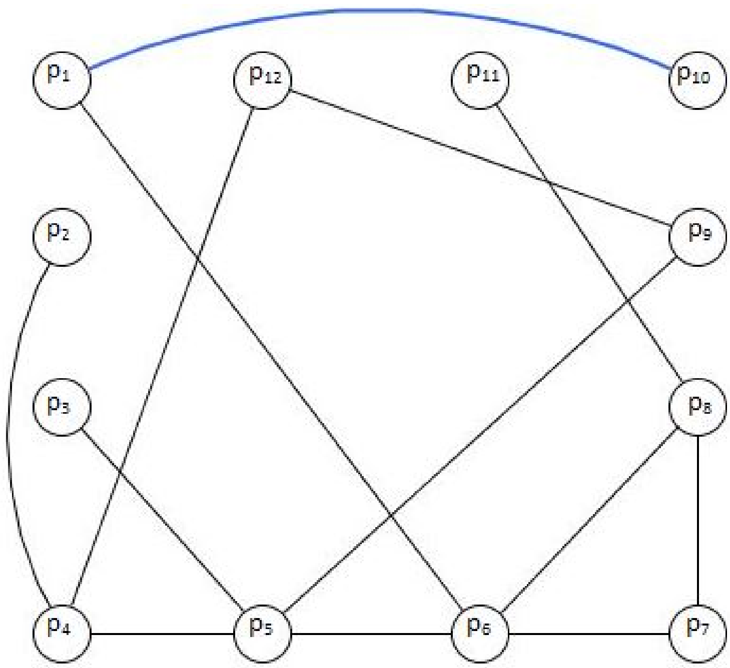
$k \leq n \quad \varepsilon^k \subseteqq \varepsilon^{k-1}$ ahol

$$\varepsilon^k = \begin{cases} \{\mu \epsilon \varepsilon^{k-1} : \mu(k) \neq k\} & \text{ha } \exists \mu \epsilon \varepsilon^{k-1}, \quad \text{hogy } \mu(k) \neq k \\ \varepsilon^{k-1} & \text{egyébként} \end{cases}$$

Megjegyzés 1 *Ha egy pontnak több párja is lehetne, akkor mindig azt a párt választom, aminek kevesebb megfelelő párja lehet, azaz azon pontokat akiknek kevesebb élük van. Az algoritmus ezt nem várja el, de könnyen végig lehet gondolni, hogy ebben az esetben működik a legoptimálisabban minden résztvevőt nézve.*

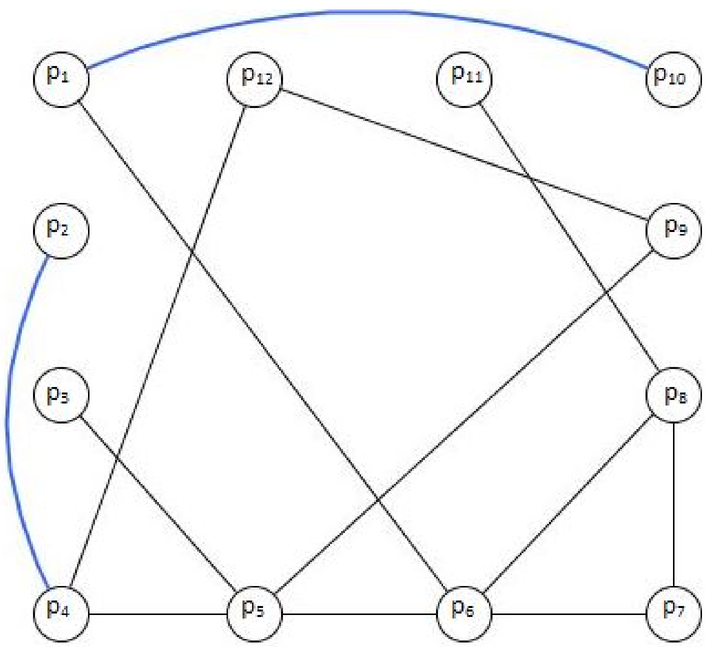
Példa

Legyen adva 12 pár beteg-donor a következő prioritási sorrendben: $p_1, p_2, ..., p_{11}, p_{12}$. Az algoritmus szerint először a p_1 pontot szeretnénk párosítani.



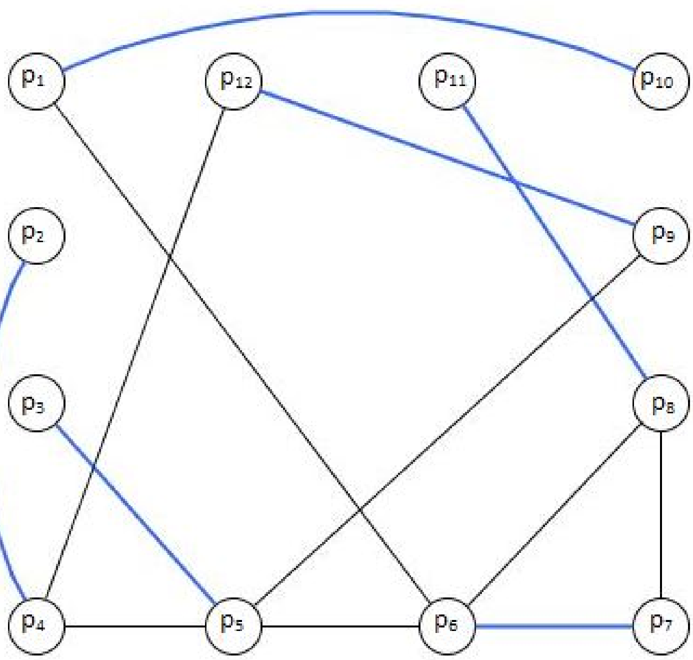
4.ábra

Ezután p_2 pontnak keresünk párt.



5.ábra

Az algoritmust így folytatjuk tovább, amíg lehetséges.



6.ábra

Az adott példában teljes párosítást kaptunk, ám ez közel sem igaz minden esetben. Gyakorlatban ez az algoritmus nem optimális minden résztvevőnek, hanem csak a lista elején állóknak.

1.3. TTC algoritmus (top trading cycles)

Számunkra igazán a TTCC (Top trading cycles and chains) algoritmus lesz érdekes, hiszen az ami valójában működik a vesecserés eljárások során. Ahhoz hogy megértsük a TTCC algoritmust, szükséges elődjéről a TTC algoritmusról [11] is pár szót szólnunk. A TTC algoritmus lényege:

1. Minden listán lévő beteg rámutat a számára legoptimálisabb vesére (donorra). Ekkor mindenképpen keletkezik legalább egy kör. Kör lehet az is, amikor a beteg a saját magához tartozó donorra mutat rá.

2. Minden körben végighajtjuk a megfelelő cseréket. Tehát a körökön belül mindenki megkapja a neki legmegfelelőbb donort. Majd ezeket a betegeket a kijelölt donorjaikkal együtt eltávolítjuk a listáról.

3. A folyamat addig zajlik, amíg nem marad beteg.

Ha jobban megfigyeljük a fenti algoritmust láthatjuk, hogy a végén mindenki kap magának donort, de azt nem vettük figyelembe, hogy lehetséges van olyan beteg, akinek nem megfelelő a donor. Tehát az orvosi feltételeket elhanyagoltuk. A TTCC algoritmus ezért jobb vesecserés szempontból, hiszen ott nincs minden betegnek egy preferencia sorrendje az összes donort nézve. Hanem vannak olyan betegek (nem is kevesen), akiknek nem megfelelő az összes donor.

1.4. TTCC algoritmus (Top trading cycles and chains)

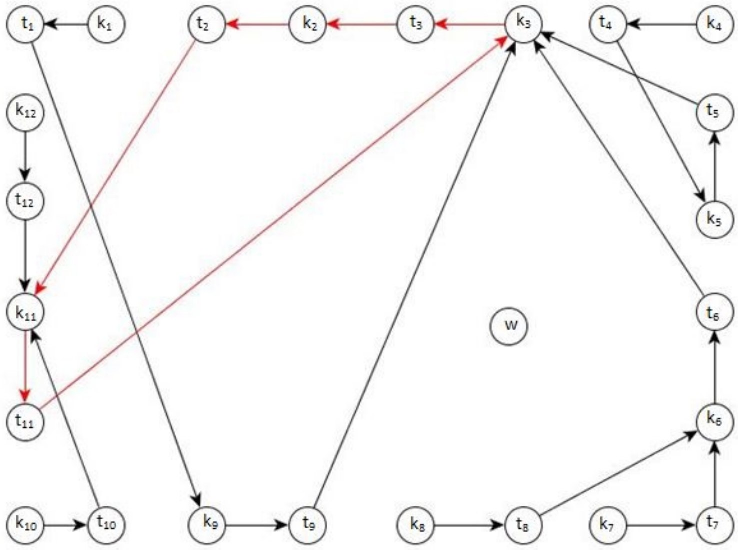
A TTCC algoritmust [11] több a vesecseréhez hasonló problémához is használhatjuk.
Legyen adva egy gráf a következőképpen:
Legyen t_i az i. beteg, k_i az i. vese - azaz a t_i beteg ismerősének veséje a k_i, w jelöli a várólistát. Tegyük fel, hogy mind az n betegnek az összes n vesén, adott egy rendezése. Ha az i. beteg preferencia sorrendjében előrébb van a k. vese a j. vesénél, azt a $j \prec_i k$ jelöli. Minden t_i beteg mutasson a neki legmegfelelőbb k_j vesére vagy w-re, illetve minden k_i vese mutasson a t_i betegre! A kapott gráfból hagyjuk el a köröket (azaz az egy körben lévő betegek kapják meg a számukra legjobb vesét), majd azon pontok akik így nem mutatnak sehova, mutassanak a következő "legjobb" választásukra. Ismét töröljük a köröket... Az algoritmus így folytatható. Amikor már nem található kör, akkor a gráf egy vagy több w-ben végződő láncból áll. Azaz ott is végrehajtható a csere, és az utolsó beteg várólistára kerül. Nézzünk meg egy konkrét példát a TTCC algoritmus futására! [9]

Példa

Legyen 12 párunk $(k_1, t_1), \ldots, (k_{12}, t_{12})$ a következő preferenciákkal:

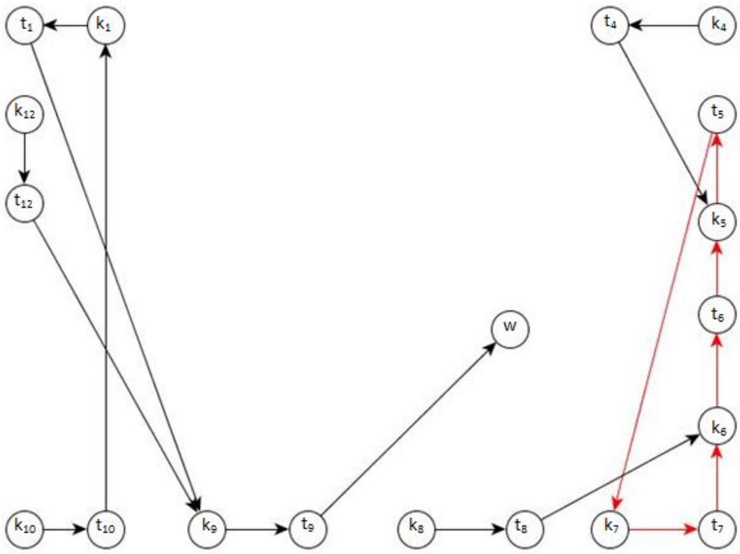
$$\begin{array}{llllllllllllllll}
t_1: k_9 & k_{10} & k_1 & & & & & & t_7: k_6 & k_1 & k_3 & k_9 & k_{10} & w \\
t_2: k_{11} & k_3 & k_5 & k_6 & k_2 & & & & t_8: k_6 & k_4 & k_{11} & k_2 & k_3 & k_8 \\
t_3: k_2 & k_4 & k_5 & k_6 & k_7 & k_8 & w & & t_9: k_3 & k_{11} & w \\
t_4: k_5 & k_9 & k_1 & k_6 & k_{10} & k_3 & w & & t_{10}: k_{11} & k_1 & k_4 & k_5 & k_6 & k_7 & w \\
t_5: k_3 & k_7 & k_{11} & k_4 & k_5 & & & & t_{11}: k_3 & k_6 & k_5 & k_{11} \\
t_6: k_3 & k_5 & k_8 & k_6 & & & & & t_{12}: k_{11} & k_3 & k_9 & k_8 & k_{10} & k_{12}
\end{array}$$

Minden t_i beteg mutasson a neki legmegfelelőbb k_j vesére vagy w-re, illetve minden k_i vese mutasson a t_i betegre!



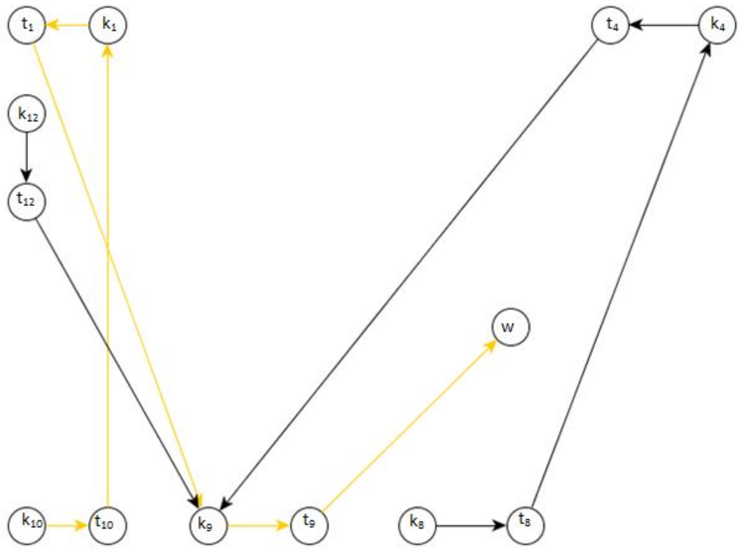
7.ábra

Hagyjuk el a pirossal jelölt kört, és akik nem mutatnak senkire mutassanak a második kedvencükre! Így kapjuk meg a 8. ábrát.



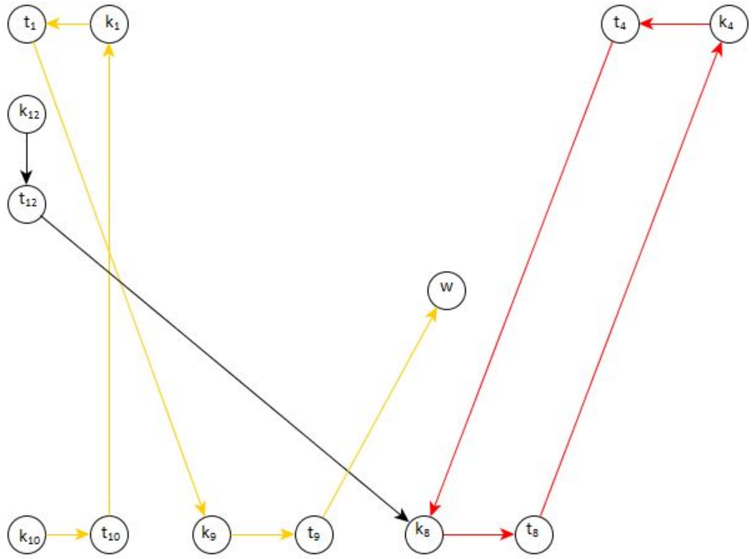
8.ábra

Hagyjuk el a kört, és akik nem mutatnak senkire mutassanak a maradékból kedvencükre! Így kapjuk meg a 9. ábrát.



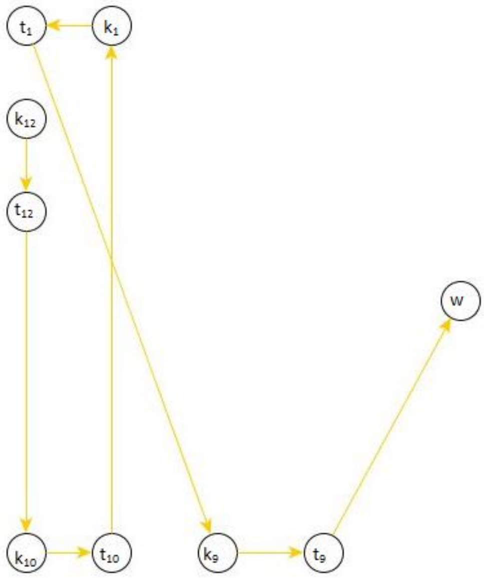
9.ábra

A sárgával jelölt élek láncot alkotnak.A következő lépésben azon pontoknak vegyük a preferencia sorrendjében következő pontot, amelyek eddig a sárgán jelölt lánc egy belső pontjára mutattak. Így kapjuk meg a 10. ábrát.



10.ábra

Így kaptunk egy újabb kört (piros élek), melyet elhagyva és a szükséges pontokat átirányítva a gráfunk már csak egyetlen láncból áll. Így kapjuk meg az 11. ábrát.



11.ábra

Tehát minden beteg kapott vesét, kivéve t_9, aki várólistára került. Illetve megmaradt a k_{12} vesénk.

1.5. Listás csere (indirect exchange mechanism)

A listás csere [9] lényege, hogy a váró listán szereplő betegek közül előre kerül az, akinek egy rokona/ismerőse másnak adott vesét. Értelemszerűen a rokon beteg ismerősének szeretne segíteni, de mivel ez nem volt lehetséges a különböző feltételek miatt, így más utat kellett választaniuk. A páros csere abban az esetben tökéletesen működik, amikor kettő beteg-donor párunk van. Ám a jelenlegi helyzetben egyetlen egy nem-kompatibilis beteg-donor párunk van és egy várólistánk, ahol szerepel a sok vesére váró reménykedő beteg.

1.6. Egyenlőségre törekvő algoritmus (egalitarian mechanism)

Sztochasztikus algoritmus (stochastic mechanism)

Legyen G a kompatibilis beteg-donor párok gráfja és $\mathcal{M}$ ezen gráfban szereplő párosítások halmaza. Továbbá legyen $\lambda = (\lambda_\mu)_{\mu\epsilon M}$ egy valószínűségi változó az $\mathcal{M}$ halmaz fölött. Minden $\mu\epsilon\mathcal{M}$ párosításhoz legyen $\lambda_\mu\epsilon[0,1]$ a μ párosítás valószínűség a λ-ban, és $\Sigma_{\mu\epsilon\mathcal{M}}\lambda_\mu = 1$. A sztochasztikus algoritmus [10] nem más, mint egy szisztematikus eljárás egy valószínűségi változó kiválasztásához minden problémára. Legyen adva egy valószínűségi változó, illetve egy szomszédsági mátrix: $A(\lambda) = [a_{i,j}(\lambda)]$, ami összefoglalja a valószínűségét annak, hogy az i beteg és a j beteg között párosítás legyen. Tehát minden lehetséges párhoz hozzárendelünk egy valószínűséget - annak a valószínűséget, hogy ők párban álljanak - és az ezt tartalmazó mátrix lesz az A mátrix, ahol a sorokban és az oszlopokban is ugyanazok a betegek szerepelnek. Továbbá minden pácienshez rendeljünk hozzá egy valószínűséget, mely azt jelöli, hogy mekkora az esélye annak, hogy ő részt vesz majd egy transzplantációban. Tehát minden beteghez tartozik egy indukált valószínűségi profil, ami gyakorlatilag egy vektor, mely tartalmazza az összes valószínűséget. A vektor első eleme az első beteggel való csere valószínűsége és így tovább. Ha az indukált valószínűségi profil $u(\lambda)$ vektor, akkor $(u(\lambda) = (u_i(\lambda))_{i \epsilon \mathcal{M}}$ és $u_i = \Sigma_{j \epsilon \mathcal{M}} a_{i,j}(\lambda)$, azaz u_i annak a valószínűsége, hogy az i-edik betegnek lesz párja. Erre az algoritmusra épül az egyenlőségre törekvő algoritmus.

Egyenlőségre törekvő algoritmus

Definíció Minden beteg kialakít egy *preferencia sorrend*et az adott vesék között. Előrekerül a számára legmegfelelőbb vese, majd a sorrendjének vége felé haladva fokozatosan csökken a kompatibilitása az adott vesével.

Az $a \succ_i b$ jelölés azt jelenti, hogy az a vese jobb a b vesénél az i. páciens szerint, azaz

1. Az i betegnek megfelel a j beteg által hozott donor: $j \succ_i i$
2. A j beteggel nem kompatibilis az i beteg donorjának veséje: $i \succ_i j$
3. A j és h beteg is rendelkeznek i-nek megfelelő donorral, vagy egyik sem: $j \sim_i h$

Tehát jelen esetben egy páciensnek egyformán megfelelőek a számára kompatibilis vesék.

Definíció Egy μ párosítás *Pareto-hatékony*, ha nem létezik olyan μ-től különböző η párosítás, amire $\eta(i) \succeq_i \mu(i)$ minden i-re és van olyan i melyre $\eta(i) \succ_i \mu(i)$.

Másképp egy rendszer Pareto-hatékony, ha a rendszerben nem hajtható végre olyan változtatás (Pareto-javítás), hogy a szereplők közül legalább egy jóléte nő és a többieké nem csökken. [12]

A jelenlegi modellben, mivel a beteg számára egyformán jók a kompatibilis vesék, egy párosítás pontosan akkor lesz Pareto-hatékony, ha maximális méretű. Ugyanis, ha lenne nagyobb méretű párosítás, találhatnánk alternáló utat két fedetlen pont között, és az ebben szereplő összes páciens legalább olyan jól járna, az eddig fedetlen páciensek szigorúan jobban.

Definíció *Alulkereslet páciens*nek (underdemanded patient) hívjuk azokat a betegeket, akikre létezik olyan Pareto-hatékony párosítás, mely őket párosítatlanul hagyja. Az alulkereslet páciensek halmazát jelöljük N^U-val.

Definíció *Túlkereslet*nek (overdemanded patient) nevezzük azokat a pácienseket, akik nem alulkeresletek és szomszédja egy alulkereslet páciensnek. A túlkereslet páciensek halmazát jelöljük N^O-val.

Definíció Az a beteg *tökéletesen párosított* (perfectly-matched patient), aki nem alulkereslet és nincs is alulkereslet szomszédja. A tökéletesen párosított páciensek halmazát jelöljük N^P-vel.

Gallai-Edmonds Dekompozíciós Lemma [5] [3]

Legyen μ egy Pareto hatékony párosítás az eredeti (N, R) gráfra és legyen (I, R_I) s túlkereslet páciensek törlésével kapott részgráf. Azaz $I = N \setminus N^0$ és

$$R_{ij} = \begin{cases} 1 & \text{ha i és j betegek kompatibilisek} \\ 0 & \text{egyébként} \end{cases}$$

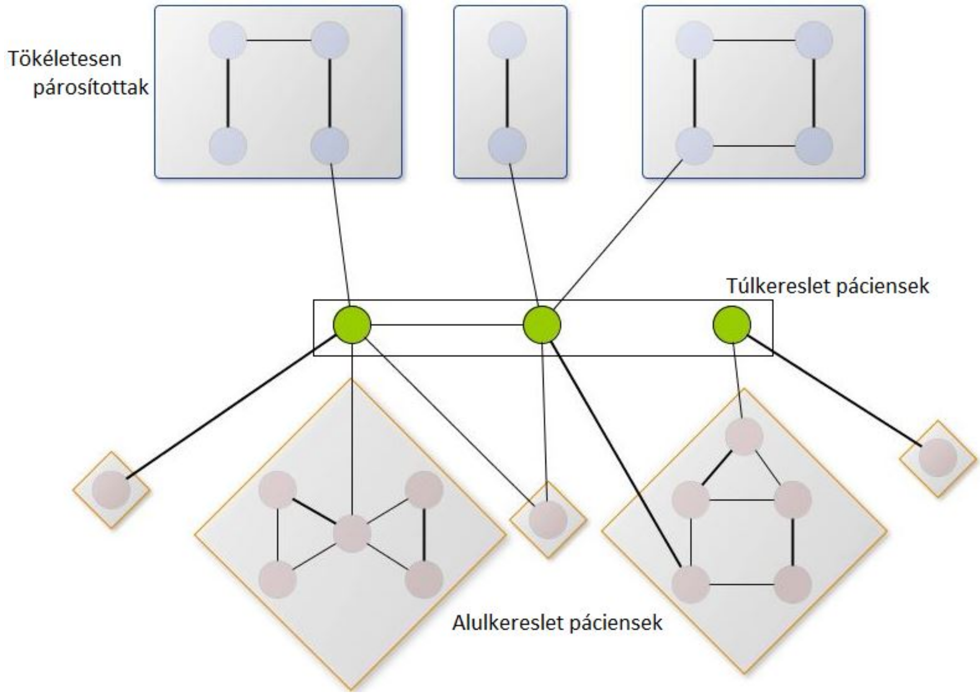
Tekintsük (I, R_I) komponenseit. Ekkor:

1. Minden túlkereslet páciens párosítva van egy alulkereslet pácienssel.

2. Ha a J halmaz (I, R_I) egy páros komponense, akkor $J \subseteq N^P$, azaz tökéletesen párosított páciensekből áll. Továbbá minden J-beli páciens μ-ben párosítva van egy másik J-beli pácienssel.

3. Ha a J halmaz (I, R_I) egy páratlan komponense, akkor $J \subseteq N^U$, azaz alulkereslet páciensekből áll. Bármelyik $x \in J$ beteget választva létezik teljes párosítás a $J \setminus x$ halmazon. A μ párosításra nézve egy kivételével minden J-beli páciens egy másik J-belivel áll párban, az utolsó páciens pedig vagy egy N^O-beli pácienssel áll párban, vagy párosítatlanul marad.

Példa egy Pareto-hatékony párosításra



Tehát a tétel azt mondja ki, hogy ha töröljük a gráfból N^0-t, akkor a maradék gráf páros komponensei N^P-beliek, míg a páratlan komponensei N^U-beliek.
Az egyenlőségre törekvő algoritmus [9] lényege, hogy - amennyire lehet - egyenlővé tegye a transzplantációra vonatkozó különböző esélyeket. Ha a sztochasztikus algoritmust elfogadnák a transzplantációs szervezetek, akkor ez az algoritmus nyújtaná az alapját annak, hogy hogyan tegyük az esélyeket egyenlővé, amellett, hogy hatékonyak legyünk.
$N = 1, 2..., n$ legyen a páciensek halmaza. $C(\mathcal{J}, I)$ jelölje a $\mathcal{J}$ páratlan komponenseinek szomszédjait az I halmazokon, ahol $\mathcal{J} \subseteq D$ és $I \subseteq N^0$. Definiáljuk a következő f függvényt:

$$f(\mathcal{J}, I) = \frac{|\cup_{j\epsilon\mathcal{J}}\, j| - (|\mathcal{J}| - |C(\mathcal{J}, I)|)}{|\cup_{j\epsilon\mathcal{J}}\, j|}$$

Ha $|\mathcal{J}| > |C(\mathcal{J}, I)|$, akkor legalább $|\mathcal{J}| - |C(\mathcal{J}, I)|$ alulkereslet páciens párosítatlanul marad. Emiatt $f(\mathcal{J}, I)$ lényegében a $\mathcal{J}$-beli alulkereslet páciensek átlagos hasznosságát adja meg a párosításokban. Míg a $f(\mathcal{J}, I)$ egy felső becslés, a $\mathcal{J}$-beli legkevésbé szerencsés hasznosságára. Az algoritmus a következő:

- első lépés

$$D_1 := argmin_{\mathcal{J} \subseteq D} f(J, N^0)$$

$$N_1^0 := C(D_1, N^0)$$

Azaz D_1-be kerültek azon alulkereslet páciensek, akik a legrosszabbul járnak. Hogy az ő esélyeiket növeljük, hozzárendeltük a szomszédaikat a túlkereslet páciensek közül, hogy amennyi hasznosságot el tudnak érni azt átlagban tényleg elérjék.

- k-adik lépés
$$D_k := argmin_{J \subseteq D \setminus \cup_{t=1}^{k-1} D_t} f(J, N^0 \setminus \cup_{t=1}^{k-1} N_t^0)$$
$$N_k^0 := C(D_k, N^0 \setminus \cup_{t=1}^{k-1} N_t^0)$$
Azaz az előző pontokban szereplő alulkereslet, illetve túlkereslet pontokat kivesszük a rendszerből, és a maradékban keressük meg a legkevésbé szerencséseket, ez lesz a D_k.

1.7. YRMH-IGYT algoritmus

Az YRMH-IGYT [1] algoritmus (You request my house - I got your turn) azon alapul, hogy vannak újonnan érkezők is a csere lebonyolítása közben. Először tekintsük az algoritmust a hagyományos szoba kiosztási problémára nézve, majd csak később vonunk párhuzamot a vesék és az szobák között. Tehát a probléma:
Egy kollégiumban vannak diákok, akiknek van szobájuk, és vannak diákok, akiknek még nincs szobájuk. Az közös minden diákban, hogy mindenkinek van egy preferencia rendezése a szobákon. Sorba állítjuk a diákokat egy véletlen sorrend szerint. Minden lépésben a soron következő diák rámutat a neki legjobban tetsző szobára. Ha az a szoba még nem foglalt, akkor beköltözik oda, ebben az esetben a szóban forgó diák és szoba többet nem vesz részt az algoritmusban,. Amennyiben a szoba foglalt, az előző tulajdonosa a lista elejére kerül, és választhat szobát, így egy út alakul ki, mely mentén végrehajtható a szobacsere. Az algoritmus így folytatódik, amíg mindenki nem kap szobát.
Most már bátran vonhatunk párhuzamot a vesék és a szobák között [12]. Az élődonorral rendelkező betegeket feleltessük meg a szobával rendelkező diákokkal, az élő donor a már foglalt szobákkal, a várólistán lévő betegeket a szobával nem rendelkező diákokkal és a nem élő donorból származott veséket a szabad szobákkal.

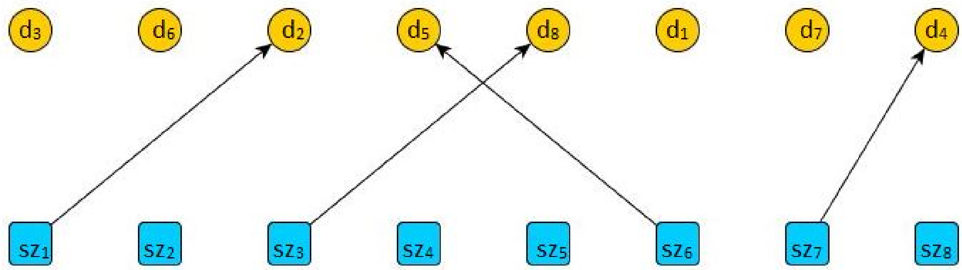
Szoba kiosztás	Vesecsere
szobával rendelkező szobák	élődonorral rendelkező páciensek
már foglalt szobák	élő donorok
újonnan érkező diákok	várólistás betegek
szabad szobák	halottból származó vesék

Tehát a YRMH-IGYT algoritmus gyakorlatilag nem más, mint a korábban említett elsőbbségi algoritmus kombinálva a listás cserével.

Példa

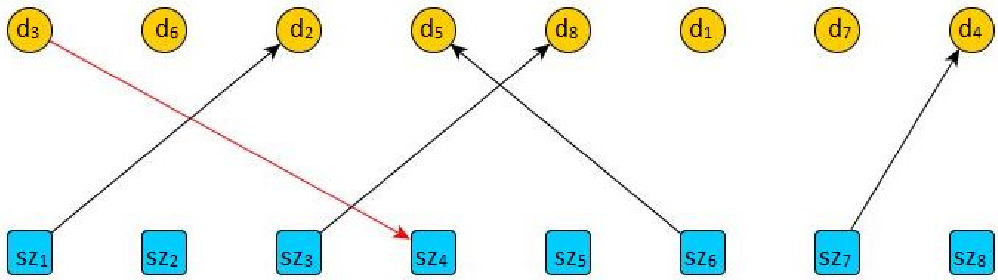
Az első ábrán látható páros gráf pontjai a diákok (d_i) és szobák (sz_i) alkotják, míg az élek az eredeti beosztást jelöli, tehát az új lakók érkezése előtt d_2 birtokolta az sz_1-es szobát. A véletlen sorrend is leolvasható a gráfról. A diákok sorrendje a következő: d_3 választ először, utána a d_6, majd rendre a d_2, d_5, d_8, d_1, d_7, d_4 diákok. A diákok szobákon kialakított preferencia sorrendjei a következőek:

$$
\begin{array}{llllllll}
d_1 : sz_1 & sz_2 & sz_3 & sz_4 & sz_5 & sz_6 & sz_7 & sz_8 \\
d_2 : sz_8 & sz_3 & sz_5 & sz_7 & sz_2 & sz_1 & sz_4 & sz_6 \\
d_3 : sz_4 & sz_8 & sz_7 & sz_6 & sz_5 & sz_2 & sz_3 & sz_1 \\
d_4 : sz_3 & sz_2 & sz_1 & sz_4 & sz_8 & sz_6 & sz_7 & sz_5 \\
d_5 : sz_1 & sz_3 & sz_5 & sz_7 & sz_4 & sz_2 & sz_8 & sz_6 \\
d_6 : sz_6 & sz_7 & sz_8 & sz_3 & sz_5 & sz_4 & sz_1 & sz_2 \\
d_7 : sz_3 & sz_2 & sz_1 & sz_4 & sz_5 & sz_8 & sz_7 & sz_6 \\
d_8 : sz_7 & sz_1 & sz_3 & sz_4 & sz_8 & sz_2 & sz_6 & sz_5
\end{array}
$$



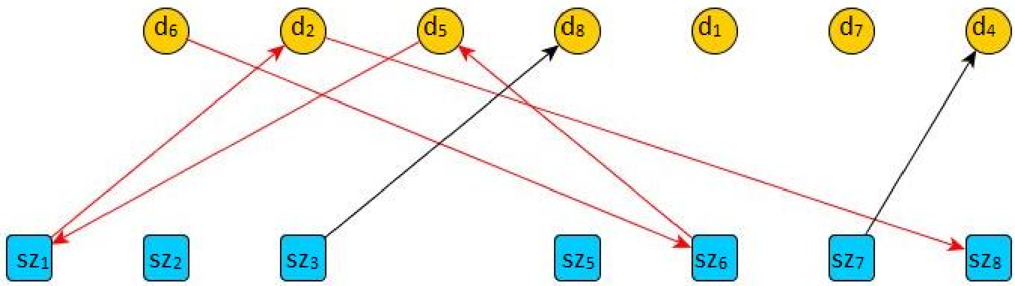
12.ábra

Ezután megkezdődik a szobák újraosztása az eddig szoba nélküli diákokat is figyelembe véve.



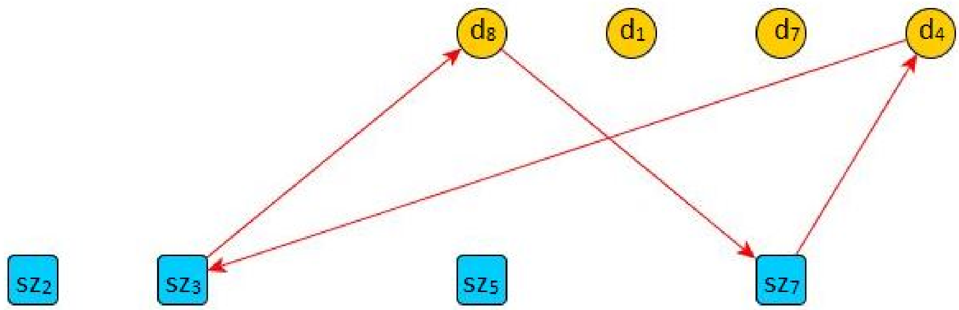
13.ábra

Mint a 13. ábrán látható, a d_3 diák elsősorban a sz_4 szobát szeretné, mivel az a szoba eredetileg is szabad volt, a diák gond nélkül megkaphatja, és a továbbiakban ezen szoba-diák páros nem vesz részt az eljárásban.



14.ábra

Az algoritmus következő lépésében a soron következő diáknak kell kiválasztania a neki legszimpatikusabb szobát. Ahogy a 14. ábra mutatja, a d_6 tanuló az sz_6 szobát választotta. Ám ez a szoba eredetileg foglalt volt a d_5 diák által. Így d_5 a sor elejére kerül és választhat szobát. A d_5 választása a szintén foglalt sz_1 szobára esett, tehát az sz_1 szoba eredeti tulajdonosa d_2 a sor elejére kerül. Ő az sz_8 szobát választja, ami szabad, így elkezdődhet a csere. A piros élek útján hajtsuk végre a szoba kiosztást. Világos, hogy senki se jár rosszabbul, mint az eredeti kiosztásnál. Az újonnan párosított szobákat és diákokat töröljük az eljárásból.



15.ábra

A 15. ábrán a d_8 (már korábban szobával rendelkező tanuló) választotta az sz_7 szobát. Ám az sz_7 szoba korábban d_4 diákhoz tartozott. Ezért d_4 a sor elejére kerül és ott kiválasztja a neki tetsző sz_3 szobát, melyet eddig d_8 birtokolt, tehát kaptunk egy kört. Ezúttal a csere a kör mentén hajtandó végre. A következő lépésben d_1 választ szobát a maradékok közül. Mivel d_1 preferencia sorrendjében az első megmaradt szoba az sz_2, ezért ezt választja. A szoba ezelőtt is szabad volt, így gond nélkül megkaphatja. Hasonlóan d_7 tanuló az sz_5 szobát kapja meg. Az algoritmussal kaptunk egy megfelelő szobakiosztást.

Tétel 2 ***Abdulkadiroglu és Sönmez*** *[1] Adott hozzárendelésnél az YRMH-IGYT algoritmussal ugyanazt az eredményt, mint a TTC algoritmus.*

Könnyen meggondolható, hogy a fenti állítás valóban igaz. Hiszen a TTC algoritmusnál a két diszjunkt halmazt a vesék és a betegek alkotják. Minden betegnek van egy donorja, hiszen ez volt a feltétele a cserének, így vegyük úgy, hogy minden vese (szoba) foglalt, és nincs újonnan érkező, erre alkalmazva az YRMH-IGYT algoritmust, pont a TTC-t kapjuk. Azzal a különbséggel, hogy a TTC algoritmusnál a betegek egyszerre mutatnak rá az általuk preferált vesére, de ez a lényegen nem változtat.

2. fejezet

Algoritmusok tulajdonságai

2.1. Hatékony algoritmusok

Definíció Egy vesecsere algoritmusról azt mondjuk, hogy hatékony, ha mindig Pareto-hatékony párosítást ad meg az aktuális betegek és a vesék között.

2.1.1. TTC algoritmus

Tétel 3 *Minden hozzárendeléshez a TTC algoritmus hatékony.*

Bizonyítás: [1] Tekintsük a TTC algoritmust.

- Minden beteg aki az első körben veséhez jutott (és emiatt nem vesz részt az algoritmus további lépéseiben), megkapta az ő első választott veséjét, azaz nyilvánvalóan nem járhat ennél jobban.
- Minden beteg aki a k. körben jutott veséhez, megkapta az ő neki legjobb vesét a maradék vesék közül. Ezek a betegek nem kaphatnak jobb vesét anélkül, hogy megsértenénk az előző körökben párosított betegek halmazát.

Ezzel beláttuk, hogy a TTC algoritmus hatékony. □

2.1.2. TTCC algoritmus

Tétel 4 *A TTCC algoritmus hatékony, ha [9] olyan szabállyal választunk a láncok közül, melynek lényege, hogy bármely kiválasztott lánc egy nemzárt körben marad, azaz továbbra is részt vesz az algoritmusban (megtartjuk azt) és a lánc végén lévő vesék is elérhetőek maradnak a továbbiakban. Például [12]:*

- *Legrövidebb w-láncot választjuk ki és azt megtartjuk.*

- *A legmagasabb prioritású párral rendelkező w-láncot választjuk és megtartjuk.*

Bizonyítás: [9] Vegyük a TTCC algoritmust a fent leírt lánc-választási szabállyal.

- Minden beteg aki az első körben veséhez jutott (és emiatt nem vesz részt az algoritmus további lépéseiben), megkapta az ő első választott veséjét, azaz nyilvánvalóan nem járhat ennél jobban.

- Minden beteg aki a k. körben jutott veséhez, megkapta az ő neki legjobb vesét a maradék vesék közül. Ezek a betegek nem kaphatnak jobb vesét anélkül, hogy megsértenénk az előző körökben párosított betegek halmazát.

Tehát a TTCC algoritmus hatékony. □

2.2. Taktikázásbiztos algoritmusok

Azt mondjuk, hogy egy hozzárendelés taktikázásbiztos, ha nem manipulálható taktikailag. Taktikai manipulálás alatt azt érjük, hogy az i. beteg az eljárás végén az a vesét kapná, viszont el tudja érni, hogy a'-t kapja úgy, hogy a valós $\prec_i$ preferenciái helyett egy másik $\prec_i'$-t közöl. [13].

2.2.1. Stabil párosítás

Tétel 5 *A Gale-Shapley algoritmus a betegek szempontjából taktikázásbiztos.*

Bizonyítás: [13] Indirekten tegyük fel, hogy van olyan beteg (t_1), aki sikeresen tud taktikázni. Ez azt jelenti, hogy vannak olyan $\pi = (\prec_{t_1}, \prec_{t_2}, ..., \prec_{t_n})$ preferenciák, és t_1-nek egy másik $\prec_{t_1}'$ preferenciája, hogy ha μ jelöli a betegoptimális stabil párosítást a π-re, μ' jelöli a betegoptimális stabil párosítást a $\pi' = (\prec_{t_1}', \prec_{t_2}, ..., \prec_{t_n})$-re, akkor $\mu(t_i) \prec_{t_i} \mu(t_1)'$, azaz t_i jobban jár, ha a hamis $\prec_{t_1}'$ preferencia sorrendet adja meg. Jelöljük Alg(π)-vel az eredeti π preferenciával elvégzett algoritmust, míg Alg(π')-vel a π'-vel elvégezve.
Tegyük fel, hogy $\mu'(t_j) \prec_{t_j} \mu(t_j)$. Ez azt jelenti, hogy Alg(π') során t_j-t $\mu(t_j)$ valamikor elutasítja. Legyen j az az index, amire először történik ilyen. Mivel $\mu(t_j)$ elutasítja t_j-t, ezért ajánlatot kapott valakitől, akitől Alg(π) során nem kap ajánlatot. De j választása miatt ez a beteg nem lehet t_l $l \neq 1$-re, másrészt t_1 sem, hiszen akkor $\mu'(t_1) = \mu(t_j)$ lenne és $t_1 - \mu(t_j)$ blokkoló él lenne μ-re. Tehát nem igaz, hogy $\mu'(t_j) \prec_{t_j} \mu(t_j)$, azaz $\prec_{t_1}'$-ben $\mu'(t_1)$ a legjobb.
Ebből következik, hogy ami az Alg(π') során megtörténik, az megtörténik az Alg(π) során is. Ezért nem t_1 az utoljára maradt páciens Alg(π)-ben, hiszen az utoljára kiválasztott

vesét csak egy beteg keresi meg, ezért ugyanaz a páciens keresi meg, aki Alg(π') során is. Tegyük fel, hogy Alg(π) során t_1 a k-adik lépésben keres meg utoljára egy vesét. Már csak be kell látni, hogy ha egy t_j beteg Alg(π) során az l-edik lépés után keres meg egy vesét, akkor $\mu'(t_j) = \mu(t_j)$ és $\mu'(t_1) = \mu(t_1)$.
Nevezzünk egy vese megkeresést beteljesülőnek, ha végül a beteg és a vese egy párt alkot. Indukcióval bizonyítunk, a beteljesülő megkeresések ideje szerint fordított sorrendben. Láttuk, hogy az Alg(π)-beli utolsó betegnek ugyanaz a párja μ-ben, mint μ'-ben. Tegyük fel, hogy a t_q páciens az r-edik lépésben keresi meg a $\mu(t_q)$ vesét és hogy az r. lépés utáni beteljesülő megkeresésekre igaz, hogy μ'-ben is párt alkotnak. Legyen T' azon páciensek halmaza, akiknek $\mu(t_q)$ kompatibilisebb, mint a saját μ-beli párjuk. Ha $T' = \emptyset$, akkor $\mu(t_q)$-t nem keresi meg t_q-n kívül más Alg(π) során, így Alg(π') során se, tehát $\mu(t_q) = \mu'(t_q)$. Ha $T' \neq \emptyset$, akkor legyen t_u a $\mu(t_q)$-nak legjobban megfelelő vese T'-ben. Vagyis $\mu(t_q)$ valamikor elutasítja t_u-t t_q miatt, vagy az r-edik lépésben, vagy később. Tehát t_u az r-edik lépés után keresi meg végső párját, így az indukciós feltevés miatt $\mu(t_u) = \mu'(t_u)$. Mivel $t_u \neq t_1$, ebből következik, hogy Alg(π')-ben t_u megkeresi $\mu(t_q)$-t, aki visszautasítja. $\mu(t_q)$ megkeresői csak T'-beliek vagy t_q lehetnek, tehát t_q miatt utasítja vissza. Tehát $\mu'(t_q) = \mu(t_q)$.
Ezzel beláttuk, hogy $\mu'(t_1) = \mu(t_1)$, ami ellentmond az indirekt feltevésnek, tehát a tételt beláttuk. □

2.2.2. Elsőbbségi algoritmus

Az elsőbbségi algoritmus nyilvánvalóan taktikázásbiztos, hiszen az első ember akire mutat, azt megkapja, így neki nem érdemes hazudnia. A második ember szintén megkapja az ő által választottat, így neki sem érdemes hazudnia. Így végigmehetünk az összes emberen és láthatjuk, hogy senkinek sem érdemes hazudnia. [8]

2.2.3. TTC algoritmus

Tétel 6 ***Roth (1982)*** *A TTC algoritmus taktikázásbiztos.*

Bizonyítás: [13]
A beteg-donor gráf megrajzolásánál hagyjuk ki az i beteget, tehát az i. betegből kilépő éleket ne húzzuk be! A többi betegből kilépő éleket az adott preferenciák szerint adjuk hozzá minden lépésben. Az algoritmus végére kapunk egy olyan fát, melynek minden éle i felé van irányítva, az i. beteg bárhogy választ ezután vesét, csak olyat kaphat, amely betegből él mutat felé - csak ebben az esetben alakul ki kör, ami mentén a csere végrehajtható. □

2.2.4. TTCC algoritmus

Tétel 7 *[9] Az, hogy a TTCC algoritmus taktikázásbiztos-e függ a w-lánc választásának elvétől.*
A következő választási stratégiák mellett a TTCC algoritmus taktikázásbiztos:

- *Válasszuk a legrövidebb w-láncot, majd töröljük ki.*
- *Állítsunk fel egy sorrendet a beteg-donor párok között. Válasszuk azt a w-láncot, mely a listán legelőrébb álló párból indul. Majd töröljük ki.*
- *Állítsunk fel egy sorrendet a beteg-donor párok között. Válasszuk azt a w-láncot, mely a listán legelőrébb álló párból indul. Majd tartsuk meg.*
- *Legyen egy elsőbbségi sorrend az alkalmas beteg-donor párok közt. Rendelkezzenek magasabb prioritással a 0-s vércsoportúak, mint a más vércsoportúak. A w-láncot úgy válasszuk, meg hogy a sorrendben előrébb álló párból induljon ki a lánc. Töröljük a láncot, ha a pár 0-s vércsoporttal rendelkezett, de a többi esetben tartsuk meg.*

Összefoglalva

Az alábbi összefoglaló táblázatban láthatóak a dolgozatomban szereplő egyes algoritmusok tulajdonságai, melyek közül több állítást is fent bizonyítottam.

Algoritmus	Hatékony	Taktikázásbiztos
Páros csere	$\surd$	$\surd$
Gale-Shapley algoritmus	$\surd$	$\surd$
Elsőbbségi algoritmus	$\surd$	$\surd$
TTC algoritmus	$\surd$	$\surd$
TTCC algoritmus	Nem minden esetben	Nem minden esetben
YRMH-IGYT algoritmus	$\surd$	$\surd$
Egyenlőségre törekvő algoritmus	$\surd$	$\surd$

3. fejezet

Legfeljebb k hosszú körrel rendelkező csere

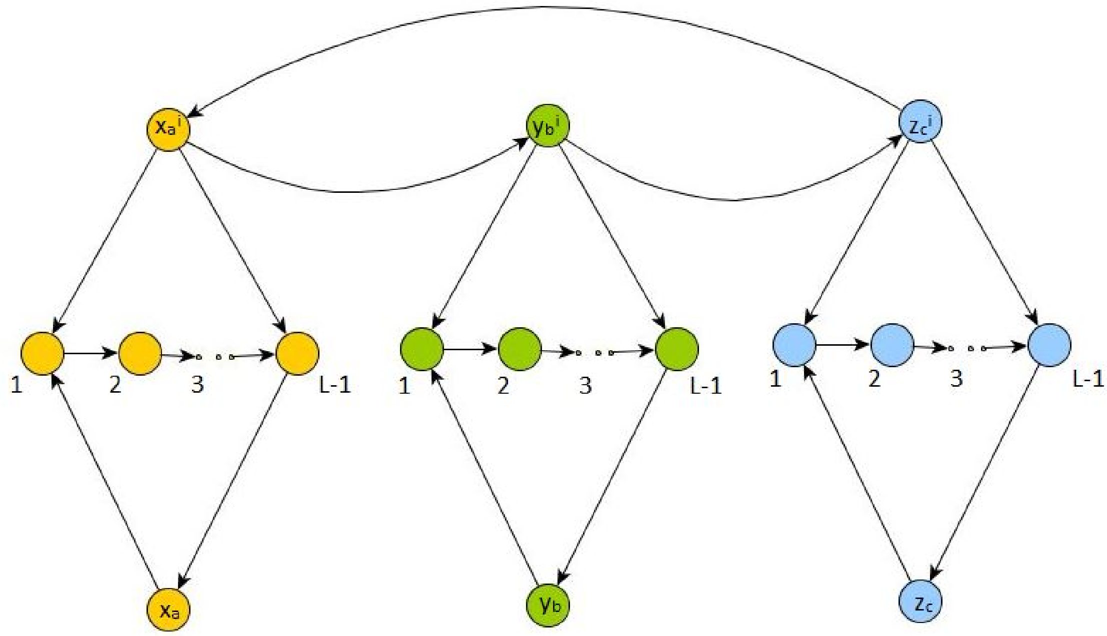
Egy vesecserénél általában a cél a transzplantációk számának maximalizálása a beteg-donor kompatibilitását figyelembe véve. Gyakorlatban az átültetéseket egyidőben kell végezni, hogy ne fordulhasson elő, hogy egy donor visszalép, ezért most csak a 2 és 3 hosszú körök mentén történő cseréket engedünk meg.

Definíció Legyen X, Y, Z diszjunkt q elemű halmazok és $T \subseteq X \times Y \times Z$ halmaz. M egy *3D-párosítás*, ha q darab diszjunkt, T-beli halmazból áll.

Tétel 8 *[2] Annak az eldöntése, hogy létezik-e 3D-párosítás NP-teljes.*

Tétel 9 *[2] Legyen adva egy $G = (V, E)$ gráf és egy $L \geq 3$ egész szám. Annak eldöntése, hogy G lefedhető-e legfeljebb L hosszú körökkel NP-teljes.*

Bizonyítás: [2] Az világos, hogy a probléma NP-beli, hiszen ha adott G lefedése körökkel, polinom időben tudjuk ellenőrizni, hogy teljes-e.
Az NP-nehézség belátásához vezessük vissza a problémát a 3D-párosításra.
Tekintsük a következő visszavezetést. Egy adott $T \subseteq X \times Y \times Z$ rendszerben konstruáljunk egy irányított gráfot, mely a 16. ábrán látható és csúcsai az X, Y, Z elemei.



16.ábra

Ezt polinomiális időben megtehetjük.
Legyen M egy teljes 3D-párosítás. Meg fogjuk mutatni, hogy a konstrukció ad egy rövid körökből álló teljes fedést.

- Ha $t_i = (x_a, y_b, z_c) \in M$. Vegyük bele a fedésbe a t_i konstrukciójából azt a három L-hosszú kört, mely tartalmazza x_a-t, y_b-t és z_c-t. Továbbá adjuk hozzá a $\{x_a^i, y_b^i, z_c^i\}$ kört.

- Ha $t_i = (x_a, y_b, z_c) \notin M$, Adjuk hozzá a konstrukciónkból (16. ábra) azt a három L-hosszú kört, mely tartalmazza x_a^i-t, y_b^i-t és z_c^i-t.

Világos, hogy ekkor az összes csúcs le van fedve, mert M particionálja $X \cup Y \cup Z$-t.

Megfordítva, tegyük fel, hogy van egy teljes lefedésünk legfeljebb L-hosszú körökből. Vegyük figyelembe, hogy a konstrukcióban csak 3 vagy L hosszú körök vannak, és egyetlen egy rövid kör sem tartalmaz két különböző konstrukcióban lévő csúcsot. Könnyű látni, hogy egy teljes fedésben minden t_i-hez tartozó részgráfban csak az előbb leírt $t_i \in M$ vagy $t_i \notin M$ eseteknél látott módon kaphatunk rövid köröket, ezért létezik teljes 3D-párosítás az eredeti rendszerben. $\square$

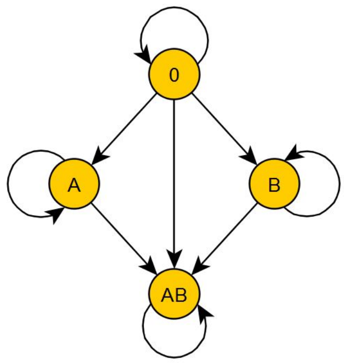
4. fejezet

Vesecserét befolyásoló tényezők

Mint a bevezetőben említettem sok tényezőn [7] múlik, hogy egy donor veséje kompatibilis-e a fogadó féllel. Íme néhány dolog, mely szükséges a sikeres vesetranszplantációhoz.

ABO kompatibilitás

A vértípus az egyik legfontosabb tényező, amely befolyásolja a vesecserét. Hiszen ugyanúgy ahogy nem adhat mindenki mindenkinek vért, nem adhat mindenki mindenkinek vesét. Aki nem adhat a másiknak vért, az nem adhat vesét sem. Az ábrán jól látható, hogy milyen vértípusú donor milyen vértípusú betegnek adhat vesét.



HLA kompatibilitás

A humán leukocita antigének (HLA) olyan fehérjék, amelyek a sejtek felületén elhelyezkedve segítik az immunrendszert a testazonos és testidegen sejtek azonosításában. Transzplantáció esetén donor-befogadó hisztokompatibilitási viszonyainak egyeztetése szükséges. Veseátültetés esetén mind HLA I, mind HLA II antigének fontosak, utóbbiból HLA-DR antigén a legjelentőségteljesebb.

Ezenfelül a túlélési időtartam hosszát befolyásolják a következőkben felsoroltak is (a teljesség igénye nélkül):

- Immunszuppresszió kezelés. Az immunszuppresszió (immunosuppresion) nem más mint a szervezet immunválaszának csökkentése egy külső beavatkozással szemben. Ezt gyógyszerekkel vagy egyéb módszerekkel idézik elő. Veseátültetés esetén azért alkalmaznak immunszuppressziót, hogy megelőzzék azt, hogy a szervezet a beültetett vesét kilökje. Fontos megjegyezni, hogy nincs rá vizsgálat, mely eldönti, hogy egy betegnek mennyi (van-e) szüksége immunszuppresszió kezelésre, tehát ezt az orvosnak kell eldöntenie. Ám a túl nagy mértékű kezelés ugyanakkora bajt okozhat, mint a túl kis mértékű kezelés.

- A donor életkora. A fejlett országokban nincs felső korhatár arra, hogy ki lehet donor. Azonban több betegség előfordulási valószínűsége is nő a kor előrehaladtával, ezért 75 év feletti donorral nem gyakran találkozhatunk.

- A donor betegségei. Van néhány betegség, melyekben szenvedő emberek nem lehetnek donorok. Például vírusos májgyulladás, AIDS, TBC, rosszindulatú daganatok, fertőző betegségek. Nyilvánvaló, hogy minél egészségesebb a donor annál "jobb" a veséje.

- Nagy jelentősége van annak is, hogy a vese élő vagy agyhalott donortól származik. Míg agyhalott donortól kapott vesével a páciens 8-9 évig él, addig élő donortól kapott szerv esetén 15-20 évig. A túlélés esélye mindkét esetben megegyezik.

- A páciens mennyi ideje vár vesére, illetve mennyi ideig kapott dialízist. A várakozási idő lehet 1-2 hónap, de akár több év is. Minél hamarabb kap valaki vesét, annál nagyobb az esélye a túlélésre. A dialízis csak akkor befolyásolja a túlélés idejét, ha a páciens 12 hónapnál hosszabb ideig kapott efajta kezelést.

- A páciens életkora, jó általános állapota és helyes életfelfogása szintén fontos szerepet játszik a transzplantáció sikerességében.

Tehát rengeteg szempontot figyelembe kell venni egy donor-páciens pár kialakításához. A tökéletes vesetranszplantációnak valószínűsége nagyon kicsi. Majdnem minden esetben van egy-egy tényező, ami nem az optimális. Tehát nem kell minden tényezőnek tökéletesnek lennie. Nyilván van egy fontossági sorrend, melyek alapján a döntés születik. A dolgozatban tárgyalt preferenciasorrendek ezen tényezők figyelembe vételével alakultak ki.

Irodalomjegyzék

[1] Atila Abdulkadiroglu and Tayfun Sönmez, "House Allocation with Existing Tenants", Journal of Economic Theory (1999) 88: 233-260

[2] Abraham, D., Blum, A., Sandholm, T.: "Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges." In: Proc. of the 8th ACM Conference on Electronic Commerce (EC), pp. 295-304 (2007)

[3] J.R. Edmonds, Maximum matching and a polyhedron with 0,1-vertices, J. Res. Nat. Bur. Standards Sect. B (1968), 125-130.

[4] D.Gale, L.S.Shapley, "College Admission and the stability of Marriage", The American Mathematical Monthly, Vol. 69 No 1 (1962), 9-15.

[5] T. Gallai, Maximale Systeme unabhänginger Kanten, Magyar Tud. Akad. Mat. Kutató Int. Közl., 9 (1965), 401-413.

[6] Robert W. Irving, "An Efficient Algorithm for the Stable Roommates Problem", Department of Mathematics, University of Saijord Salford MS 4WT, United Kingdom (1984)

[7] Barry D. Kahan, Claudio Ponticelli, "Principles and Practice of Renal Transplantation"

[8] Jonathan Levin, Stanford University, Lecture Note Economics 136: Undergraduate Market Design, "House Allocation and Kidney Exchange"

[9] Roth, Alvin E., Tayfun Sönmez and M. Utku Ünver. "Kidney Exchange" Quarterly Journal of Economics 119(2): 457-488 (2004)

[10] Alvin E. Roth, Tayfun Sönmez, M. Utku Ünver, "Pairwise kidney exchange", National Bureau of economic research (2014)

[11] L.S. Shapley and H. Scarf, "On Cores and Indivisibility. Journal of Mathematical Economics 1" (1974) 23-37

[12] Varga Gábor, "Vese csere program matematikai modellje" (2011)

[13] Végh László, Pap Júlia, Király Tamás, "Játékelmélet jegyzet" (2014)