Eötvös Loránd Tudományegyetem
Természettudományi Kar

Elérési idők tabu Markov-láncok esetén

BSc Szakdolgozat

Készítette: Osváth Tibor Attila

Matematika BSc, Matematikai elemző szakirány

Témavezető: Pröhle Tamás

Eötvös Loránd Tudományegyetem, Természettudományi Kar

Valószínűségelméleti és Statisztika Tanszék

Budapest, 2020.

NYILATKOZAT

Név: Osváth Tibor Attila

ELTE Természettudományi Kar, szak: matematika alapszak

NEPTUN azonosító: JXX096

Szakdolgozat címe:
Elérési idők tabu Markov-láncok esetén

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2020.05.29

Osváth Tibor Attila

a hallgató aláírása

Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek Pröhle Tamásnak, aki mindenben segített, és mindig számíthattam a segítségére. Szeretném még megköszönni családomnak, a töretlen támogatásukat, és a végtelen türelmüket, amivel hozzám viszonyultak.

Tartalomjegyzék

Bevezetés

A matematikában a Markov-lánc egy olyan diszkrét sztochasztikus folyamatot jelent, amely Markov-tulajdonságú. Nevét Andrej Markovról kapta, aki hírnevét a sztochasztikus folyamatokban végzett kutatásainak köszönheti. Markov-tulajdonság azt jelenti, hogy a jelen állapot mellett a jövőbeli állapot nem függ a múltbeli állapotoktól. Semmilyen múltban történő esemény, nem hat, nem ad előrejelzést a jövőre nézve.

Egy egyszerű példa erre a következő. Ketten játszanak egy kockával. Mindenki addig dobhat míg 6-ost nem dob. Ha valaki 6-ost dob, akkor át kell adnia a kockát a másiknak. Azt vizsgáljuk, hogy a k. dobás után kinél van a kocka: A-nál vagy B-nél. Ez egy 2 állapotú Markov-lánc. Az, hogy a $k+1$. lépés után kinél van a kocka, ha tudjuk, hogy a k. lépés után kinél volt, nem függ attól, hogy kinél volt a $(k-1)$., $(k-2)$., $(k-3)$., $\ldots 0$. dobás után. Másként mondva, az, hogy kinél lesz a kocka a következő dobás után, csak attól függ, hogy most kinél van, és hogy ő 6-ost dob-e vagy sem.

A matematika ezen ágát számos más tudomány is alkalmazza. A Markovi rendszerek jelentős részei a fizikának, különösképp a statisztikus mechanikának. A gazdaságban a dinamikus makroökonómiának is nélkülözhetetlen része. A Markov-láncokat használhatjuk a statisztika egyes folyamatainak modellezésére is.

Az egyik legismertebb alkalmazása a társasjátékokhoz köthető. Az ismert gyerekjáték, a „Kígyók és létrák” is megközelíthető a Markov-láncok felöl. Minden egyes körnél a játékos egy meghatározott mezőn áll (adott állapotban van) és megvannak a rögzített valószínűségek (mit dob a kockával) a következő lehetséges mezőre (állapotba) jutáshoz.

A szakdolgozatom első részében a Markov-lánc fogalmát fogom definiálni és az alapvető definíciókat fogom kimondani evvel kapcsolatban (elérhetőség, visszatérőség stb.).

A második részben a `markovchain` csomag alapjait fogom bemutatni, továbbá megnézem azokat a függvényeket amik a várható elérési idővel kapcsolatban állnak.

A harmadik részben bevezetem a tabu valószínűségek fogalmát. Az első felében kimondok vele kapcsolatos definíciókat és tételeket, majd a második felében egy másik irányból, rekurzióval közelítem meg az elérési valószínűségeket.

Végül a negyedik részben a tabu Markov-láncok segítségével szimulálok különböző átjárókat és értelmezem a kapott eredményeket.

1. A Markov-láncokról általánosságban

Legyen az $x(t) \equiv x_t$, $t = 0, 1, ...$ Markov lánc lehetséges állapotainak (az x_t lehetséges értékeinek) a halmaza az $S = \{0, 1, 2, ..., n\}$ véges vagy $S = \{0, 1, 2, ...\}$ megszámlálhatóan végtelen diszkrét halmaz.

A fejezethez a [2], [5], [6] forrásokat használtam.

1. Definíció. *S értékű véges (vagy megszámlálhatóan végtelen) sok értéket felvevő $x_0, x_1, x_2, \dots$ valószínűségi változók sorozatát **Markov-láncnak** nevezzük, ha a valószínűségi változók között a következő összefüggés fennáll. Minden $x_{t+1}, x_t, ...$ valószínűségi változó minden lehetséges $i_{t+1}, i_t, ... \in S$ értékére teljesül, hogy*

$$\mathbb{P}(x_{t+1} = i_{t+1} | x_t = i_t,\ x_{t-1} = i_{t-1}, \dots,\ x_0 = i_0) = \mathbb{P}(x_{t+1} = i_{t+1} | x_t = i_t)$$

Ez a definíció egyaránt érvényes véges és végtelen állapottér esetén. Mi a továbbiakban csak véges állapotterű Markov folymatokkal foglalkozunk. A Markov-tulajdonság értelmezhető folytonos időben is, azaz, ha t $\in \mathbb{R}^+$ vagy $\mathbb{R}$. Viszont az előző definíció csak a diszkrét idejű Markov-láncokra igaz. Mi a továbbiakban csak véges állapotterű, diszkrét idejű Markov folyamatokkal foglalkozunk, amelyeket röviden Markov-láncoknak nevezünk.

A Markov-lánc időben *homogén*, ha

$$\mathbb{P}(x_{t+1} = i_{t+1} | x_t = i_t) = P(x_1 = i_{t+1} | x_0 = i_t) \quad \forall t \geq 0,$$

és minden lehetséges i_{t+1} és i_t esetén. Azaz, ha az átmenetvalószínűségek a t időponttól függetlenek.

Esetünkben $S = \{1, 2, ..., n\}$. Jelölje az $n \times n$ méretű $\mathbf{P}$ mátrix a Markov lánc átmenet mátrixát. Azaz ha a $\mathbf{P}$ mátrix i. sor j. eleme p_{ij} akkor tetszőleges $t = 0, 1, ...$ időpontra legyen

$$p_{i,j} = \mathbb{P}(x_{t+1} = j | x_t = i)$$

A Markov-tulajdonság azt jelenti, hogy ha az x_t eloszlását az n elemű q vektor adja meg, akkor az x_{t+1} eloszlása a $j = 1, 2, ..., n$ lehetséges értékekre:

$$\mathbb{P}(x_{t+1} = j) = \sum_{i=1}^{n} \mathbb{P}(x_{t+1} = j | x_t = i) \mathbb{P}(x_t = i) = \left(q^T \cdot \mathbf{P}\right)_j$$

A **P** mátrix neve átmenet-valószínűségmátrix vagy egylépéses átmenet-valószínűségmátrix. Ennek az n-edik hatványa:

$$\mathbf{P}^n = p_{i,j}^{(n)}$$

ahol

$$p_{i,j}^{(n)} = \mathbb{P}(x_{t+n} = j | x_t = i)$$

Ez a kapcsolat teszi lehetővé azt, hogy a Markov-láncok vizsgálatára fel tudjuk használni pl. a mátrix hatványozást.

1.1. Az állapotok osztályozása

Elérhetőség

2. Definíció. *Azt mondjuk, hogy az i állapotból a j elérhető ($i \to j$) ha van olyan $n \geq 0$, amelyre $P_{ij}^{(n)} > 0$. Ez reflexív($P_{ii}^{(0)} = 0$) és tranzitív(Chapman-Kolmogorov egyenlet)*

1. Állítás. *Legyenek $p_{ij}^{(n)} = P(X_{n+m} = j | X_m = i)$ az n-edrendű átmenetvalószínűségek. Legyen i egy olyan állapot, amelyre $P(X_m = i) > 0$. Ekkor teljesül a Chapman-Kolmogorov egyenlet:*

$$p_{ik}^{(n+m)} = \sum_{j \in S} p_{ij}^{(n)} p_{jk}^{(m)}$$

Bizonyítás.

$$P(X_{n+m} = k | X_0 = i) \stackrel{(1)}{=} \sum_{j \in S} P(X_{n+m} = k, X_n = j | X_0 = i) \stackrel{(2)}{=}$$

$$\stackrel{(2)}{=} \sum_{j \in S} P(X_{n+m} = k | X_n = j, X_0 = i) P(X_n = j | X_0 = i) \stackrel{(3)}{=} \sum_{j \in S} p_{jk}^{(m)} p_{ij}^{(n)}$$

1. Megjegyzés. *(1) teljes valószínűség tétele miatt érvényes.*

(2) ekvivalens átalakítás.

(3) Markov-tulajdonság miatt teljesül.

□

3. Definíció. *Azt mondjuk, hogy i és j közlekednek ha $i \to j$ és $j \to i$. Ez ekvivalenciareláció, tehát osztályokra bontja az állapotteret (csak az átmenetmátrixtól függ). A Markov-lánc irreducibilis, ha az állapottere egyetlen osztályból áll.*

4. Definíció. *Az i állapot lényeges, ha $i \to j$ esetén $j \to i$ is teljesül.*

Az állapotokra értelmezett valamely tulajdonság osztálytulajdonság, ha egy osztálynak vagy minden eleme ilyen tulajdonságú, vagy egy sem.

5. Definíció. *Az $\{n \geq 0, P_{ii}^{(n)} > 0\}$ halmaz legnagyobb közös osztója az i periódusa, jele: $d(i)$. Ha $d(i) = 1$ akkor az állapot aperiodikus.*

2. Állítás. *Egy osztály minden állapotának ugyanannyi a periódusa.*

Bizonyítás. Legyenek $k, l \in C$ azonos osztálybeliek. Ekkor létezik n, m amelyekre teljesül, hogy $P_{kl}^{(n)} > 0$ és $P_{lk}^{(m)} > 0$. Ebből a kettőből következik, hogy $P_{kk}^{(n+m)} > 0$. Ha valamely s-re $P_{kk}^{(s)} > 0$, akkor ezek alapján $P_{kk}^{(n+m+s)} > 0$. Emiatt $d(k)|(n+m)$ és $d(k)|(n+m+s)$ az oszthatóság szabályai miatt, pedig a kettő különbségét $d(k)|s$ is osztja. Tehát $d(k)$ közös osztója az ilyen s számoknak azaz $d(k)|d(l)$. Mivel k és l szerepe felcserélhető, az állítást beláttuk. □

Visszatérőség

Legyen $f_{ij}^{(0)} = 0,\ f_{ij}^{(n)} = \mathbb{P}(X_n = j, X_k \neq j : k = 1, 2 \ldots n-1 | X_o = i),\ n \geq 1$
Ez annak a valószínűsége, hogy az i állapotból indulva a lánc először az n-edik lépésben ér el a j állapotba. Legyen $f_{ij}^{*} = \sum_{n=1}^{\infty} f_{ij}^{(n)}$ ez azt jelenti, hogy az i állapotból indulva a lánc előbb-utóbb elér a j állapotba.

6. Definíció. *Az i állapot visszatérő(vagy rekurrens), ha $f_{ii}^{*} = 1$, egyébként pedig átmeneti vagy tranziens.*

2. A markovchain csomag

A szakdolgozatom elkészítéséhez hozzátartozik egy programozási feladat, amit az R programnyelvben implementáltam. Ahhoz, hogy meg tudjunk ismerkedni a programmal, először nézzük meg a `markovchain` csomagot. A `markovchain` csomagot Giorgio Alfredo Spedicato hozta létre 2016-ban. A célja, hogy a benne található függvények segítségével könnyebben tudjuk kezelni a diszkrét Markov-láncokat.

Forrásként leginkább [3], [4] eredményeire támaszkodtam.

2.1. S4 osztály

Az S4 osztályt a `setClass()` függvény határozza meg. Az objektumok komponenseit slotoknak nevezzük. Az osztály meghatározásakor be kell állítanunk a nevét és a slotokat (a slot osztályával együtt). Egy egyszerű példán bemutatjuk, az S4 osztályok kezelési módját.

```
setClass("tanulo",
slots=list(nev="character", eletkor="numeric", atlag="numeric"))
s <- new("tanulo",nev="Tibor", eletkor=22, atlag=4)
s
An object of class "tanulo"
Slot "nev":
# "Tibor"
Slot "eletkor":
# 22
Slot "atlag":
# 4
isS4(s)
#TRUE
s@nev
# "Tibor"
s@eletkor
```

```
# 22
s@atlag
# 4
```

Itt definiáltunk egy új osztályt annak bemutatására, hogy hogyan működik az objektumorientált környezetben az osztály-metódus pár. Az osztály neve `tanulo`, és 3 slotja van, aminek a nevei a következők: `nev`, `eletkor`, `atlag`. Az S4-es objektumokat a `new()` függvénnyel hozhatunk létre. Azt, hogy egy objektum tényleg S4-es objektum az `isS4()` függvénnyel ellenőrizhetjük. Csakúgy, mint egy lista összetevőjéhez a dollárjel használatával lehet hozzáférni, az objektum egy slotjához a @ jellel lehet hozzáférni.

```
s@nev<-"Attila"
s
An object of class "tanulo"
Slot "nev":
# "Attila"
Slot "eletkor":
# 22
Slot "atlag":
# 4
```

Egy slot módosítását direkt módon is el lehet végezni. Itt a `nev`et módosítottam Attilára. Ekkor a többi slot értéke változatlan maradt.

```
a<-1
a
# 1
class(a)<-"Tibi"
a
# 1
attr(,"class")
# "Tibi"
 print.Tibi<-function(u){cat("Tibi rendes diák" )}
```

```
 print.Tibi
function(u){cat("Tibi rendes diák" )}
a
Tibi rendes diák> class(a)<-NULL
a
# 1
```

Ezen felül saját módszereket is könnyedén írhatunk. Ebben az esetben print.Tibi a print függvénynek egy metódusa lett. Eredetileg `a`-t egynek inicializáljuk az osztályát pedig "Tibi"-nek állítjuk be. A print.Tibi függvény meghívásakor a "Tibi rendes diák" szöveg íratódik ki. Mivel az `a`-nak osztálya Tibi ezért a értéke kiíratáskor a print.Tibi függvény lesz, de ha `a` osztályát megszüntetjük akkor `a`-nak az eredeti értékét fogja kiírni ami 1.

2.2. committorAB

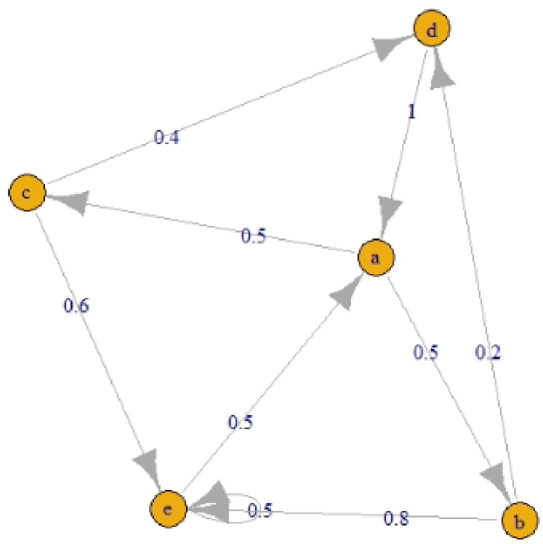
Az elérési valószínűségeket committor valószínűségeknek nevezik, ebből jön a függvény neve: committorAB. A függvény arra jó, hogy ha egy példát már kézzel kiszámoltunk, akkor a segítségével már könnyen tudjuk ellenőrizni az eredmény helyességét. A committorAB függvény azt a valószínűséget fogja visszaadni, hogy a folyamat hamarabb ér el egy állapotot az A halmazból, mint a B halmaz bármely állapotát.
A függvény 4 paraméterrel rendelkezik: committorAB(Mátrix,A,B,p)

Paraméterek

Mátrix	egy `markovchain` osztály objektum
A	állapotok egy halmaza
B	állapotok egy halmaza
p	kezdeti állapot (alapból a program 1-nek veszi)

A függvény megold egy lineáris algebrai egyenletrendszert (az egyenletrendszer formájáról később lesz szó), amivel kiszámolja, hogy a folyamat hamarabb elér egy valamilyen A-beli állapotot, mithogy elérné bármelyik B-belit. Ha a kezdeti állapot nincs feltüntetve, akkor egy vektort ad vissza a valószínűségekkel, ha pedig fel van tüntve akkor egy számot. Egy konkrét példán ez könnyebben szemléltethető.

```
Mátrix <- matrix(c(0,0,0,1,0.5,
                   0.5,0,0,0,0,
                   0.5,0,0,0,0,
                   0,0.2,0.4,0,0,
                   0,0.8,0.6,0,0.5),
                 nrow = 5)
MC <- new("markovchain", states=c("a","b","c","d","e"),transitionMatrix=Mátrix)
plot(MC)
committorAB(MC,c(5),c(3))
```



Először feltöltjük a *Mátrix*-ot, amiből a program egy 5x5-ös mátrixot csinál, nrow=5 parancs segítségével a program soronként tölti fel a beadott adatokkal. Majd létrehozunk egy új *MC* objektumot aminek az osztálya `markovchain`, állapotai az "a", "b", "c", "d", "e" állapotok, továbbá az átmenet-valószínűségmátrix lesz az előbb létrehozott *Mátrix*. A függvény meghívásakor az A halmaz az 5-ös állapotból áll, a B halmaz pedig a 3-as állapotból, kezdeti értéket most nem adtunk meg. A *Mátrix* gráfos reprezentációja a szöveg fölött található.

```
        a         b         c         d         e
0.4444444 0.8888889 0.0000000 0.4444444 1.0000000
```

Mit is jelent akkor a végeredmény?
A célállapotunk ebben az esetben az `"e"` állapot, a tabu állapotunk pedig a `"c"` állapot. Tehát pl. annak a valószínűsége, hogy a `"c"` állapotból hamarabb elérjük az `"e"` állapotot, mint a `"c"`-t nyilvánvalóan 0 lesz, mivel a kezdeti állapotunk a `"c"` volt. Nézzük meg a másik triviális esetet. A megoldásvektor 5. eleme azt adja meg, hogy az `"e"` állapotból indulva mennyi a valószínűsége annak, hogy a folyamat hamarabb éri el az `"e"` állapotot, mint a `"c"`-t. Ez nyilván 1 lesz mivel, a folyamatot az 5-ös állapotból indítottuk. A többi állapot esetén lesz "érdekes" a kapott valószínűség. Az `"a"` állapotból $\frac{4}{11}$ a valószínűsége annak, hogy az `"e"` állapotot hamarabb elérjük, mint a `"c"`-t. Ezt hasonlóan látjuk a `"b"` és `"d"` állapotra is amire rendre $\frac{8}{11}$ és $\frac{4}{11}$ lesz a keresett valószínűség.

2.3. Egyéb módszerek

Ebben az alfejezetben a `markovchain` csomag pár módszerét fogom ismertetni.

- meanFirstPassageTime(mc,cel)

  A függvény 2 paraméterrel rendelkezik:

  Paraméterek

  mc egy `markovchain` objektum

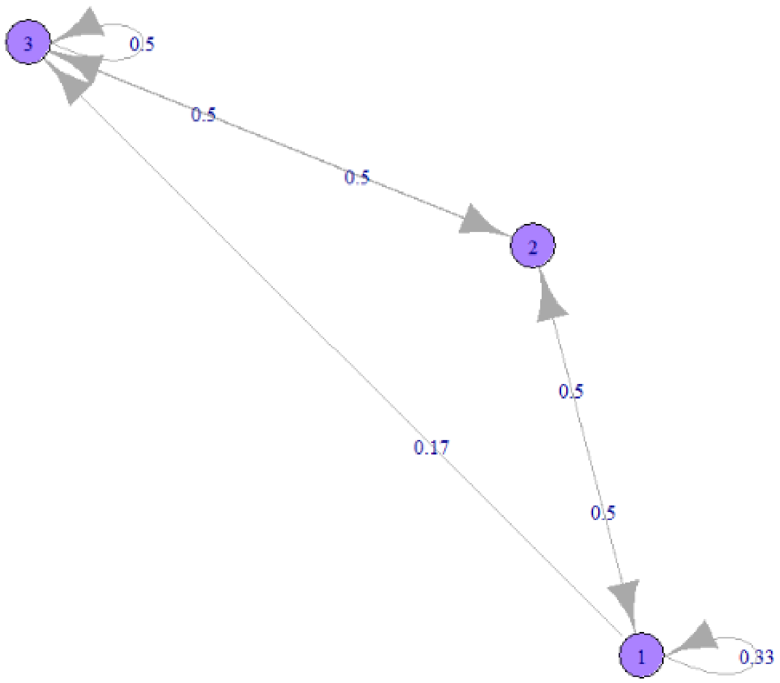
  cel célállapotok (alapból üres)

  Egy ergodikus Markov-lánc esetén a következőt számolja ki:

  - Ha a végcél üres akkor egy olyan mátrixot ad ki, ami az összes (i, j) párra kiszámolja a várható elérési időt, ahol i a kezdőállapot és j a végállapot, m állapot esetén a mátrix mérete $m * m$
  - Ha a végcél nem üres, akkor a várható elérési időt adja ki a célállapotba többi állapotból. Ha összesen m állapotunk van akkor egy $(m - 1)$ hosszú vektort ad vissza.

Példa: Vegyük a következő Markov-láncot:

$$P_{ij} = \begin{pmatrix} \frac{2}{6} & \frac{3}{6} & \frac{1}{6} \\ \frac{3}{6} & 0 & \frac{3}{6} \\ 0 & \frac{3}{6} & \frac{3}{6} \end{pmatrix}$$




```
mc <- new("markovchain", states = c("1","2","3"),
transitionMatrix = m)
meanFirstPassageTime(mc,"3")
# 1   2
 3.6 2.8
```

A program a 3-as állapotba számolja ki az elérési időket az 1-es és 2-es állapotokból. Legyen $\{\phi(i) = i\ (i = 1, 2, 3)$ állapotból a 3-as állapot első elérési ideje$\}$. Ekkor a következő egyenletrendszert kell majd megoldanunk:

$$
\begin{aligned}
\phi(1) &= 1 + \frac{2}{6}\phi(1) + \frac{3}{6}\phi(2) + \frac{1}{6}\phi(3) \\
\phi(2) &= 1 + \frac{3}{6}\phi(1) + \frac{3}{6}\phi(3) \\
\phi(3) &= 0
\end{aligned}
$$

Ebből

$$
\begin{aligned}
\frac{4}{6}\phi(1) &= 1 + \frac{3}{6}\phi(2) \\
\phi(2) &= 1 + \frac{3}{6}\phi(1)
\end{aligned}
$$

Amiből egyértelműen látszik, hogy a megoldása $\phi(1) = \frac{18}{5}$ és $\phi(2) = \frac{14}{5}$ tehát a kézzel kiszámolt eredmény megegyezik az R által számolt eredménnyel.

- meanRecurrenceTime(mc)

  Várható visszatérési idő. Bemenetként egy Markov-láncot vár, a kimenete pedig egy vektor lesz, amiben a rekurrens állapotok fognak szerepelni. Ha a Markov-lánc ergodikus (irreducibilis), akkor az összes állapot benne lesz a vektorban az eredeti sorrendben (mivel ugye minden állapot rekurrens). Azt számolja ki, hogy mennyi a várható ideje annak, hogy egy rekurrens állapotból indulva először visszatérünk ugyanabba a rekurrens állapotba. Ez viszonylag közel áll a várható elérési időhöz, ezért az előző példán fogom szemléltetni a kiszámítását.

  Jelölje t_j^* az első visszatérési időt a j állapotra.

```
meanFirstPassageTime (mc)
#  1 2   3
1 0 2 3.6
2 4 0 2.8
3 6 2 0.0
meanRecurrenceTime(mc)
#  1   2   3
  4.0 3.0 2.4
```

Az előző példához képest most annyi változtatás történt, hogy a várható elérési idő függvény esetében nincs célpont megadva, ezért outputként egy mátrixot kapunk. A mátrix i. sorának j. eleme azt fogja jelenti, hogy az i állapotból indulva várhatóan a j állapotba hány lépésben jutunk el. A mátrix i. sorának j. elemét jelölje m_{ij}.

Ekkor a következő egyenletrendszert tudjuk felírni:

$$t_1^* = 1 + \sum_{j=1}^{3} p_{1j} m_{j1} = 1 + \frac{2}{6} * 0 + \frac{3}{6} * 4 + \frac{1}{6} * 6 = 4$$
$$t_2^* = 1 + \sum_{j=1}^{3} p_{2j} m_{j2} = 1 + \frac{3}{6} * 2 + 0 * 0 + \frac{3}{6} * 2 = 3$$
$$t_3^* = 1 + \sum_{j=1}^{3} p_{3j} m_{j3} = 1 + 0 * 3.6 + \frac{3}{6} * 2.8 + \frac{3}{6} * 0 = 2.4$$

Az első egyenlet pontosan azt jelenti, hogy egyet lép a folyamat egy valamilyen j majd a j állapotból várhatóan m_{j1} lépésben fog visszatérni az 1-es állapotba. A másik kettő egyenlet ugyanaz csak a 2 és 3 állapotra alkalmazva az előzőt.

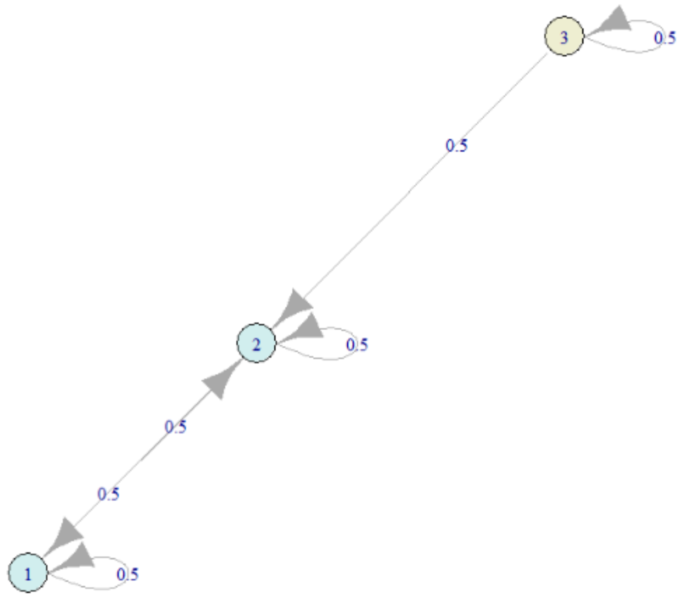
Tehát megkaptuk a program által kiszámolt értékeket.

- meanAbsorptionTime(mc)

  Várható elnyelési idő. A függvény a Markov-láncot várja bemeneti paraméternek. Megadja, hogy egy tranziens állapotból várhatóan hány lépésben jutunk el valamelyik rekurrens állapotba.

  Példa: Vegyük a következő Markov-láncot

$$P_{ij} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$




```
recurrentStates(mc)
# "1" "2"
transientStates(mc)
# "3"
meanAbsorptionTime(mc)
# 3
  2
```

Tehát a rekurrens állapotok az 1 és 2 és egy tranziens állapot van a 3. Könnyedén látszik, hogy a 3 tranziens, mivel az $1, 2$ állapotból nem vezet irányított él 3-ba. Tehát kevesebb, mint 1 valószínűséggel lesz visszatérő, mivel ha már $3 \to 2$-be átmegy, már nem fog tudni visszatérni. $\{\phi(i) = i\ (i = 1, 2, 3)$ állapotból várhatóan hány lépés alatt jut el egy tranziens állapotba$\}$.

Ekkor a következő egyenletrendszert kell megoldanunk:

$$\phi(3) = 1 + \frac{1}{2}\phi(3) + \frac{1}{2}\phi(2)$$

$$\phi(2) = 0$$

$$\phi(1) = 0$$

Ennek hasonlóan $\phi(3) = 2$ az eredménye, mint a program esetén, vagyis várhatóan 2 lépés alatt jutunk el egy tranziens állapotból egy rekurrens állapotba.

3. Tabu valószínűségek

Ahhoz, hogy mélyebben megértsük a Markov-láncokat, most bevezetjük az átmenetvalószínűségek fogalmát tabu állapotokkal. Legyen H egy tetszőleges halmaz.

Forrásként az [1]-et fogom használni.

7. Definíció. $HP_{ij}^{(n)} = \mathbb{P}\{X_n(\omega) = j; X_\mu(\omega) \notin H, 0 < \mu < n| \ X_0(\omega) = i\}, \qquad n \geq 1.$

Ez szavakkal kifejezve azt jelenti, hogy az i állapotból elindulva n lépés múlva eljutunk a j állapotba, úgy, hogy közben a folyamat sohasem volt a H egyik állapotában sem (végpontokat kivéve), vagy máshogy: a folyamat végig az $(I-H)$ halmazon belül marad. H lesz a tabu halmazunk, és a benne levő állapotok a tabu állapotok. H lehet üres, ez esetben a jelölésből kihagyjuk. Ha H egyetlen egy k állapotból áll, akkor jelöljük a valószínűséget $kP_{ij}^{(n)}$-nel. Ha a taboo halmaz a H halmaz és a k uniójából áll, akkor a következő valószínűséget jelöljük $\{k, H\}P_{ij}^{(n)}$-nel.
Legyen $HP_{ij}^{(0)} = \delta_{ij}$, ha $i \notin H$; $= 0$, ha $i \in H$.

8. Definíció. $HP_{ij}^{*} = \sum_{n=1}^{\infty} HP_{ij}^{(n)}$; *mely azt adja meg, hogy i-ből indulva, várhatóan hányszor jár a lánc j-ben, míg H-ba beér (a beérést is beszámítva).*

Ha a sorozat divergens akkor az értéke ∞ lesz. Ez persze azon várható alkalmak száma, hogy a Markov-lánc i-ből indulva hamarabb éri el a j állapotot, mint a H valamelyik állapotát. A ∞-re vonatkozó szokásos konvenciókat ugyanúgy alkalmazhatjuk, így

$$c + \infty = \infty, \qquad |c| \cdot \infty = \infty, \qquad \frac{c}{\infty} = 0, \qquad \frac{\infty}{|c|} = \infty, \qquad \frac{\infty}{-|c|} = -\infty$$

ahol $c \neq 0$ véges konstans;

$$\infty \cdot \infty = \infty, \qquad 0 \cdot \infty = 0;$$

de a $\infty - \infty$ vagy a $\frac{\infty}{\infty}$ tiltott művelet.

Most meg fogunk adni 2 formulát, ami a kiindulópontja lesz az összes tételnek.

1. Tétel. *(Általános dekompozíciós formulák) Ha $n \geq 1$ és $k \notin H$, akkor*

$$HP_{ij}^{(n)} = \{k, H\}P_{ij}^{(n)} + \sum_{\nu=1}^{n-1} \{k, H\}P_{ik}^{(\nu)} HP_{kj}^{(n-\nu)}; \tag{1}$$

$$HP_{ij}^{(n)} = \{k,H\}P_{ij}^{(n)} + \sum_{\nu=1}^{n-1} HP_{ik}^{(\nu)}\{k,H\}P_{kj}^{(n-\nu)}. \tag{2}$$

Bizonyítás. Az első tagja az összegnek mindkét esetben azt jelenti, hogy a k állapot sosincs érintve mialatt a folyamat i-ből j-be megy n lépés alatt. Első esteben a szumma pedig azt a valószínűséget jelenti, hogy a k állapot először a ν. lépésben van érintve az átmenet közben, a második esetben pedig azt mutatja hogy a k állapot utoljára a ν. lépésben van érintve az átmenet közben. □

2. Következmény. *Ha $k \notin H$, akkor*

$$HP_{ij}^{*} = \{k,H\}P_{ij}^{*} + \{k,H\}P_{ik}^{*}HP_{kj}^{*}; \tag{3}$$

$$HP_{ij}^{*} = \{k,H\}P_{ij}^{*} + HP_{ik}^{*}\{k,H\}P_{kj}^{*}; \tag{4}$$

Bizonyítás. Használjuk fel az előző dekompozíciós formulákat és szumázzuk őket $n = 1$-től $n = N$-ig

$$\sum_{n=1}^{N} HP_{ij}^{(n)} = \sum_{n=1}^{N}\{k,H\}P_{ij}^{(n)} + \sum_{\nu=1}^{N-1}\{k,H\}P_{ik}^{(\nu)}\sum_{n=1}^{N-\nu} HP_{kj}^{(n)} \tag{5}$$

$$\sum_{n=1}^{N} HP_{ij}^{(n)} = \sum_{n=1}^{N}\{k,H\}P_{ij}^{(n)} + \sum_{\nu=1}^{N-1} HP_{ik}^{(\nu)}\sum_{n=1}^{N-\nu}\{k,H\}P_{kj}^{(n)} \tag{6}$$

$N \to \infty$ esetén hozzájutunk a bizonyítandó állításhoz. Konvergens vagy divergens sorozatok esetén is érvényes a bizonyítás a ∞-re vonatkozó konvenciók alapján. □

3. Tétel. *A 3 mennyiség:*

$$HP_{ij}^{*}, \qquad \{j,H\}P_{ij}^{*}, \qquad \{i,H\}P_{ij}^{*}$$

mindegyike vagy pozitív(esetleg ∞) vagy 0.

Bizonyítás. Ha $j \notin H$ akkor vegyük $k = j$-t (3)-ban , ez alapján:

$$HP_{ij}^{*} = \{j,H\}P_{ij}^{*} + \{j,H\}P_{ij}^{*}HP_{jj}^{*} = \{j,H\}P_{ij}^{*}(1 + HP_{jj}^{*}). \tag{7}$$

Ha $j \in H$ akkor $HP_{ij}^{*} = \{j,H\}P_{ij}^{*}$. Ennélfogva, ha az egyik 0, úgy a másik is 0 lesz. Hasonlóan, ha $i \notin H$, akkor vegyük $k = i$-t a (4) -ben és így kapjuk:

$$HP_{ij}^{*} = \{i,H\}P_{ij}^{*} + HP_{ii}^{*}\ \{i,H\}P_{ij}^{*} = \{i,H\}P_{ij}^{*}(1 + HP_{ii}^{*}). \tag{8}$$

A többi hasonlóan következik az előzőekből. □

9. Definíció. *$j \rightsquigarrow H$ ha létezik olyan $k \in H$ amire $j \rightsquigarrow k$*

4. Lemma. *(Töplitz) Legyen $\{a_n, n \geq 0\}$, egy nemnegatív sorozat, $\{b_n, n \geq 0\}$ egy valós számokból álló sorozat. Ha*

$$\lim_{n\to\infty} \frac{a_n}{\sum\limits_{\nu=0}^{n} a_\nu} = 0$$

akkor

$$\lim_{n\to\infty} \frac{\sum\limits_{\nu=0}^{n} a_\nu b_{n-\nu}}{\sum\limits_{\nu=0}^{n} a_\nu} = \lim_{n\to\infty} b_n$$

5. Tétel. *Ha $j \in H$ vagy $j \rightsquigarrow H$, akkor $HP^*_{ij} < \infty$.*

Bizonyítás. Ha $j \in H$ akkor $HP^{(n)}_{ij} \leq j, P^{(n)}_{ij} = f^{(n)}_{ij}$. Ezért $HP^*_{ij} \leq f^*_{ij} \leq 1$. Legyen $k \in H,\ k \neq j$, és $j \rightsquigarrow k$ úgy, hogy $kP^{(m)}_{jk} = f^{(m)}_{jk} > 0$ valamiyen $m \geq 1$-re. Ezek alapján

$$kP^{(n)}_{ij} kP^{(m)}_{jk} \leq kP^{(n+m)}_{ik} = f^{(n+m)}_{ik}$$

Ebből következik, hogy

$$HP^*_{ij} \leq kP^*_{ij} \leq \frac{1}{f^{(m)}_{jk}} \sum_{n=1}^{\infty} f^{(n+m)}_{ik} \leq \frac{1}{f^{(m)}_{jk}}.$$

□

Ezekből elő tudunk állítani 2 általános formulát. Osszuk le az (5)-t és a (6)-t rendre $\sum\limits_{\nu=1}^{N} HP^{(\nu)}_{kj}$-nel és $\sum\limits_{\nu=1}^{N} HP^{(\nu)}_{ik}$-nel alkalmazzuk a Töplitz-lemmát így hozzájutunk az alábbi 2 formulához.

$$\lim_{N\to\infty} \frac{\sum\limits_{n=1}^{N} HP^{(n)}_{ij}}{\sum\limits_{n=1}^{N} HP^{(n)}_{kj}} = \lim_{N\to\infty} \frac{\sum\limits_{n=1}^{N} \{k,H\}P^{(n)}_{ij}}{\sum\limits_{n=1}^{N} HP^{(n)}_{kj}} + \{k,H\}P^*_{ik}; \tag{9}$$

$$\lim_{N\to\infty} \frac{\sum\limits_{n=1}^{N} HP^{(n)}_{ij}}{\sum\limits_{n=1}^{N} HP^{(n)}_{ik}} = \lim_{N\to\infty} \frac{\sum\limits_{n=1}^{N} \{k,H\}P^{(n)}_{ij}}{\sum\limits_{n=1}^{N} HP^{(n)}_{ik}} + \{k,H\}P^*_{kj}; \tag{10}$$

feltéve, hogy $k \notin H$ és a limesz a jobb oldalon létezik. Ez a formula általános, de könnyen egyszerűbbé tehetjük néhány feltételezés segítségével.

10. Definíció. *Azt mondjuk, hogy* $i \overset{H}{\rightsquigarrow} j$, *ha* $HP^*_{ij} > 0$.

11. Definíció. *Azt mondjuk, hogy* i *és* j *közlekedhető a* H *taboo halmaz alatt, ha* $i \overset{H}{\rightsquigarrow} j$ *és* $j \overset{H}{\rightsquigarrow} i$ *fennáll. Ezt* $i \overset{H}{\leftrightsquigarrow} j$*-val jelöljük.*

6. Tétel. *Ha* $i \notin H, j \notin H$ *és* $i \overset{H}{\rightsquigarrow} j$, *akkor*

$$\lim_{N\to\infty} \frac{1+\sum_{n=1}^{N} HP^{(n)}_{jj}}{1+\sum_{n=1}^{N} HP^{(n)}_{ii}} = \frac{\{i,H\}P^*_{ij}}{\{j,H\}P^*_{ij}} = \lim_{N\to\infty} \frac{1+\sum_{n=1}^{N} \{i,H\}P^{(n)}_{jj}}{1+\sum_{n=1}^{N} \{j,H\}P^{(n)}_{ii}} \tag{11}$$

Bizonyítás. Mivel $HP^*_{ij} > 0$ és a feltevés alapján $0 < \{j,H\}P^*_{ij} < \infty$ és $\{i,H\}P^*_{ij} > 0$ a **4.Tétel** és **5.Tétel** alapján. Írjunk be $k = i$-t a (10)-be és így a következőt kapjuk

$$\lim_{N\to\infty} \frac{\sum_{n=1}^{N} HP^{(n)}_{ij}}{\sum_{n=1}^{N} HP^{(n)}_{ii}} = \frac{\{i,H\}P^*_{ij}}{HP^*_{ii}} + \{i,H\}P^*_{ij} \tag{12}$$

ennek következtében

$$\lim_{N\to\infty} \frac{\sum_{n=1}^{N} HP^{(n)}_{ij}}{1+\sum_{n=1}^{N} HP^{(n)}_{ii}} = \lim_{N\to\infty} \frac{\sum_{n=1}^{N} HP^{(n)}_{ii}}{\sum_{n=0}^{N} HP^{(n)}_{ii}} \left(\frac{1}{HP^*_{ii}} + 1\right) \{i,H\}P^*_{ij} = \{i,H\}P^*_{ij} \tag{13}$$

Hasonlóan, ha $k = j$-t írunk be a (9)-be akkor megkapjuk

$$\lim_{N\to\infty} \frac{\sum_{n=1}^{N} HP^{(n)}_{ij}}{1+\sum_{n=1}^{N} HP^{(n)}_{jj}} = \{j,H\}P^*_{ij} \tag{14}$$

Mert ha $HP^*_{ii} = \infty$, akkor a limesz a jobb oldalon 1 lesz, míg ha $HP^*_{ii} < \infty$, akkor a tényezők kiesnek a zárójelen belül. Ha vesszük a (13) és (14) hányadosát, akkor az első egyenletet kapjuk meg a **6. Tétel** állításában. A tétel második egyenlőségét nem bizonyítom. □

7. Következmény. *Ha* $i \notin H$, $j \notin H$ *és* $i \overset{H}{\leftrightsquigarrow} j$ *akkor*

$$\lim_{N\to\infty} \frac{1+\sum_{n=1}^{N} HP^{(n)}_{jj}}{1+\sum_{n=1}^{N} HP^{(n)}_{ii}} = \frac{\{i,H\}P^*_{ij}}{\{j,H\}P^*_{ij}} = \frac{\{i,H\}P^*_{ji}}{\{j,H\}P^*_{ji}} = \frac{1+\{i,H\}P^*_{jj}}{1+\{j,H\}P^*_{ii}} = \frac{Hf^*_{ji}\{i,H\}f^*_{ij}}{Hf^*_{ij}\{j,H\}f^*_{ji}} \tag{15}$$

ahol a limesz pozitív és véges. Ezen felül ha a következő 4 állapotból i, j, k, l bármelyik kettő közlekedhető a taboo H halmaz alatt és $i \notin H,\ j \notin H,\ k \notin H,\ l \notin H$ vagyis egyik sincs H-ban, akkor

$$\lim_{N\to\infty} \frac{\sum\limits_{n=1}^{N} {}_HP_{ij}^{(n)}}{\sum\limits_{n=1}^{N} {}_HP_{kl}^{(n)}} \tag{16}$$

létezik és véges.

12. Definíció. $m_{ij} = \sum\limits_{n=1}^{\infty} n f_{ij}^{(n)}$

Ez az átlagos első elérés i-ből j-be, feltéve, hogy $f_{ij}^{*} = 1$ (a lánc előbb-utóbb elér i-ből j-be. Ehhez a definícióhoz fog kapcsolódni a következő tétel.

8. Tétel. *Ha* $f_{ij}^{*} = 1$*, akkor*

$$\sum_k {}_jP_{ik}^{*} = m_{ij}$$

Bizonyítás. Definíció szerint a következő érvényes:

$$\sum_k {}_jP_{ik}^{(n)} = P\{X_\nu(\omega) \neq j\ 0 < \nu < n | X_0(\omega) = i\} = \sum_{\nu=n}^{\infty} f_{ij}^{(\nu)}$$

Emiatt felírhatjuk a következőt:

$$\sum_k {}_jP_{ik}^{*} \overset{(1)}{=} \sum_k \sum_{n=1}^{\infty} {}_jP_{ik}^{(n)} \overset{(2)}{=} \sum_{n=1}^{\infty} \sum_k {}_jP_{ik}^{(n)} \overset{(3)}{=} \sum_{n=1}^{\infty} \sum_{\nu=n}^{\infty} f_{ij}^{(\nu)} \overset{(4)}{=} \sum_{\nu=1}^{\infty} \nu f_{ij}^{(\nu)} \overset{(def)}{=} m_{ij}$$

2. Megjegyzés. *(1) a* ${}_jP_{ik}^{*}$ *definíciója miatt teljesül.*

(2) a nemnegatív sorozatok átrendezhetősége miatt érvényes.

(3) a bizonyítás első lépése miatt igaz.

(4) triviális.

(5) pedig m_{ij} *definíciójából követezik.*

□

Egy másik megközelítés

Legyen az állapotok két diszjunkt részhalmaza A illetve B, az összes többi állapot legyen C. Először azt vizsgáljuk, hogy mennyi annak a valószínűsége, hogy a folyamat állapota az idő teltével úgy válik egy A-beli állapottá, hogy az előző állapotok egyike sem B-beli. Utóbb megvizsgáljuk különböző indulóállapotok mellett, annak várható időtartamát, hogy a lánc állatpota A-belivé váljon a B elkerülése mellett. Megvizsgáljuk, hogyan alakulnak a mondott valószínűségek különböző egyszerű strukturájú gráfok esetén.

3.1. Elérési valószínűségek

3.1.1. Egy állapothalmaz elérésének a valószínűsége

Legyen A az S lehetséges állapotok egy tetszőleges részhalmaza. Legyen $\tau(A,k) \equiv \tau_A(k)$ az A állapothalmaz első elérésének időpontja, ha $x(0) = k$, ahol $k \in S$, azaz egy tetszőleges induló állapot. Azaz legyen

$$\tau_A(k) = inf\{t \geq 0 : x(0) = k \text{ és } x(t) \in A\}$$

A τ lehetséges értékei $1, 2, 3, ...$, és 0, ha $k \in A$ és ∞, ha az A a k-ból nem érhető el.

Alkalmazzuk a következő jelölést:

$$h(A,k) \equiv h_A(k) = \mathbb{P}(\tau_A(k) < \infty)$$

Azaz $h_A(k)$ annak a valószínűségét jelöli, hogy az A állapothalmaz a k állapotból véges idő alatt elérhető. A $h_A(k)$, $k = 1, ..., n$ valószínűségekből alkotott n dimenziós vektort h_A fogja jelölni. A következő állítást bizonyítjuk:

9. Tétel. *Elérési valószínűség*
A $h_A(k)$ elérési valószínűség, $k \in S$-re a következő egyenlet rendszer minimális megoldása:

$$\begin{aligned} \textit{ha } k \in A \textit{ akkor:} \qquad h_A(k) &= 1 \\ \textit{ha } k \notin A \textit{ akkor:} \qquad h_A(k) &= \big(\mathbf{P} \cdot h_A\big)(k) = \sum_{j \in S} p_{k,j} h_A(j) \end{aligned}$$

A minimalitás azt jelenti, hogy minden $k \in S$-re $h_A(k) \leq u(k)$, ha n dimenziós u vektor a fenti egyenlet rendszer egy tetszőleges megoldása.

Bizonyítás.
Előbb belátjuk, hogy az elérési valószínűségek tényleg megoldásai a fenti egyenleteknek. A $k \in A$ eset nyilvánvaló: ha $k \in A$, akkor k-ból 1 valószínűséggel érhető el az A. Ha $k \notin A$ akkor $\tau_A(i) > 0$, emiatt érvényes a következő átalakítás sor első lépése, a következőben az

$x(t)$ Markov tulajdonságát használjuk ki:

$$\begin{aligned}
h_A(k) &= \mathbb{P}\big(\tau_A(k) < \infty\big) \\
&= \sum_{j\in S} \mathbb{P}\big(\tau_A(k) < \infty \text{ és } x(1) = j\big) \\
&= \sum_{j\in S} \mathbb{P}\big(\tau_A(k) < \infty \mid x(1) = j\big) \cdot \mathbb{P}\big(x(1) = j \mid x(0) = k\big) \\
&= \sum_{j\in S} p_{k,j} h_A(j)
\end{aligned}$$

Belátjuk, hogy a $h_A(k)$ az egyenletrendszer minimális megoldása.

Legyen $w(k)$, $k \in S$ az egyenletrendszer egy tetszőleges megoldása.

A $k \in A$ eset ismét nyilvánvaló. Ha $k \notin A$ akkor osszuk az egyenlőség szerinti szummát két részre:

$$\begin{aligned}
w(k) = \sum_{j\in S} p_{k,j} w(j) &= \sum_{j\in A} p_{k,j} w(j) + \sum_{j\in S\setminus A} p_{k,j} w(j) \\
&= \sum_{j\in A} p_{k,j} + \sum_{j\in S\setminus A} p_{k,j} w(j) \\
&= \mathbb{P}(x_1 \in A | x_0 = k) + \sum_{j\in S\setminus A} p_{k,j} w(j)
\end{aligned}$$

Ugyanis ha $j \in A$ akkor $w(j) = 1$. E felírás szummájában az $w(j)$-re alkalmazzuk ugyanezt a felbontást, de felhasználva, hogy minden $j \in S$-re
$\mathbb{P}(x_1 \in A | x_0 = j) = \mathbb{P}(x_2 \in A | x_1 = j)$:

$$\begin{aligned}
w(k) &= \mathbb{P}(x_1 \in A | x_0 = k) + \sum_{j\in S\setminus A} p_{k,j} \Big(\mathbb{P}(x_2 \in A | x_1 = j) + \sum_{i\in S\setminus A} p_{j,i} w(i)\Big) \\
&= \mathbb{P}(x_1 \in A | x_0 = k) + \sum_{j\in S\setminus A} p_{k,j} \mathbb{P}(x_2 \in A | x_1 = j) + \sum_{j\in S\setminus A} \sum_{i\in S\setminus A} p_{k,j} p_{j,i} w(i) \\
&= \mathbb{P}(x_1 \in A | x_0 = k) + \mathbb{P}(x_2 \in A, x_1 \notin A | x_0 = k) + \sum_{j\in S\setminus A} \sum_{i\in S\setminus A} p_{k,j} p_{j,i} w(i)
\end{aligned}$$

Mint látható, az első valószínűség annak a valószínűsége, hogy a lánc első lépésre az A-ba jut, a második pedig annak a valószínűsége, hogy a második lépés az első, amikor a lánc egy A-beli állapotba kerül. A dupla szumma pedig, az $w(i)$ tényező nélkül annak a valószínűsége, hogy a lánc állapota sem az első sem a második lépésre nem jut az A állapothalmazba.

Az így nyert kifejezés harmadik tagjában megismételhetjük a helyettesítést. Így az $w(k)$ egy olyan négy tagú felbontását nyerjük, amely a lánc 3 egymás utáni lépésének lehetséges eredményei szerinti. Ha a helyettesítést n-szer megismételjük, akkor egy olyan $n+1$ tagú felbontást kapunk, ahol az első n tag rendre annak a valószínűsége, hogy a lánc $1, 2, ..., n$ lépés után ér először A-ba. Mivel a helyettesítést akárhányszor megtehetjük, és az utolsó tag mindig nemnegatív, adódik, hogy $w_A(k) \geq h_A(k)$ ami az álltást bizonyítja. $\square$

3.1.2. Elérési valószínűség kikerülési feltétellel

Legyen A és a B az S lehetséges állapotok két diszjunkt részhalmaza. Azt vizsgáljuk, mennyi az A elérésének a valószínűsége a B kikerülése mellett. Alkalmazzuk a következő jelölést:

$$h(A, B, k) \equiv h_{A|\overline{B}}(k) = \mathbb{P}(\tau_A(k) < \tau_B(k))$$

Azaz $h_{A|\overline{B}}(k)$ annak a valószínűségét jelöli, hogy az x_t Markov lánc a k állapotból indulva hamarabb kerül egy A mint egy B-beli állapotba. A $h_{A|\overline{B}}(k)$, $k = 1, ..., n$ valószínűségekből alkotott n dimenziós vektort $h_{A|\overline{B}}$ fogja jelölni. A következő állítást bizonyítjuk:

10. Tétel. *Elérési valószínűség kikerülési feltétel mellett*

A $h_{A|\overline{B}}$ elérési valószínűség vektor a következő egyenlet rendszer minimális megoldása:

$$\widetilde{\mathbf{P}} \cdot h_{A|\overline{B}} = u$$

Ahol u egy olyan n elemű oszlop vektor, amely minden sorában 0, kivéve az A állapot halmaznak megfelelőket, ahol 1. A $\widetilde{\mathbf{P}}$ pedig a $\mathbf{P}$ átmenet valószínűség mátrix úgy módosított változata, hogy előbb a $\mathbf{P}$ mátrixból kivonunk egy identitás mátrixot, majd a mátrix A-nak és B-nek megfelelő sorait 0-val töltjük fel, kivéve az adott sorokbeli diagonális elemeket, amelyek helyére 1-eket írunk.

3. Megjegyzés. *Az egyenlet formája, ha az állapotokat olyan sorrendben vesszük, hogy előbb a c darab $S \setminus (A \cup B)$ állapotot, majd a tiltott B állapotokat végül pedig az A állapotokat soroljuk fel:*

$$\widetilde{\mathbf{P}} \cdot h_{A|\overline{B}} \equiv \begin{pmatrix} p_{1,1}-1 & \dots & p_{1,c} & p_{1,c+1} & \dots & \dots & \dots & \dots & p_{1,n} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ p_{c,1} & \dots & p_{c,c}-1 & p_{c,c+1} & \dots & \dots & \dots & \dots & p_{c,n} \\ 0 & \dots & 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 1 \end{pmatrix} \cdot h_{A|\overline{B}} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

4. Megjegyzés. *Nyilvánvaló, hogy az elkerülési feltétel nélküli valószínűségek – az előző tétel szerint – egy ugyanilyen formájú egyenlet megoldásai, kivéve azt, hogy abban az esetben, a B állapothalmaz egy bizonyosan üres halmaz.*

Bizonyítás.
Vegyük észre, hogy a $\widetilde{\mathbf{P}}$ is egy átmenet-valószínűségmátrix. A $\mathbf{P}$ mátrix szerinti Markov lánc úgy módosított változatának átmenet mátrixa, hogy az A és a B állapotok elnyelő állapotok.

Nyilvánvaló, hogy a módosított mátrix szerinti egyenlet megoldása:

$$\begin{aligned} \text{ha } k \in A \text{ akkor:} \qquad h_{A|\overline{B}}(k) &= 1 \\ \text{ha } k \in B \text{ akkor:} \qquad h_{A|\overline{B}}(k) &= 0 \end{aligned}$$

Ami a feladat feltételei szerint szükségszerű. Ha $k \notin (A \cup B)$ akkor a $h_{A|\overline{B}}(k)$-ra ugyanazok az egyenletek érvényesek mint amelyeket előzőleg a $h_A(k)$-ra felírtunk. A 2. Megjegyzés szerinti állapot sorrendet és jelölést alkalmazva: $k = 1, ..., c$-re:

$$h_{A|\overline{B}}(k) = \sum_{j \in S} p_{k,j} h_{A|\overline{B}}(j)$$

□

4. Átjáró

4.1. Fix nyelőkkel

Vegyünk egy átjárót, más néven egy véges négyzetrácsot. A Markov-lánc az a véletlen sorozat, hogy az idő múlásával melyik állapot az aktuális. Az aktuális lánc állapotainak a halmaza a négyzetrács rácspontjai lesznek. A rácsnak legyen összesen m sora és n oszlopa. Ekkor $m \cdot n$ állapota lesz, vagyis az átmenetmátrixnak $m \cdot n$ sora és $m \cdot n$ oszlopa lesz. Az állapotsorszámokat felülről lefelé és balról jobbra töltjük ki. Későbbiekben a mezőkre állapotsorszámaikkal hivatkozom.
Egy konkrét $m = 4$ $n = 5$ esetben ez így néz ki:

sorszám	[, 1]	[,2]	[, 3]	[, 4]	[, 5]
[1,]	1	5	9	13	17
[2,]	2	6	10	14	18
[3,]	3	7	11	15	19
[4,]	4	8	12	16	20

Az átmenet a szomszédokba egyenletes eloszlású lesz. Vagyis ha a jelenben az átjáró csúcspontjában van a folyamat, akkor $\frac{1}{2}$ valószínűséggel lép át egy szomszédos állapotba. Tehát az 1-es sorszámú állapotból $(\frac{1}{2}, \frac{1}{2})$ valószínűséggel lép át az 5-be vagy a 2-be. Hasonlóan, ha a folyamat aktuálisan egy élközép pontban van, akkor $\frac{1}{3}$, ha pedig egy belső pontban van, akkor $\frac{1}{4}$ valószínűséggel lép egy szomszédos pontba, mivel ezeknek rendre 3 illetve 4 szomszédjuk van. Attól Markov, hogy ez az utóbbi valószínűség mindig csak attól függ, hogy melyik állapot az aktuális.

típus	[, 1]	[,2]	[, 3]	[, 4]	[, 5]
[1,]	1	6	6	6	3
[2,]	5	9	9	9	8
[3,]	5	9	9	9	8
[4,]	2	7	7	7	4

9 féle állapottípust különböztetünk meg, az $1, 2, 3, 4$ állapotok lesznek a 4 csúcsállapot.

$5, 6, 7, 8$ jelöljük az élközépi állapotokat attól függően, hogy melyik oldalon vannak és 9 jelöljük a belső pontokat.

A tabu Markov-lánc pedig a következő lesz. A kezdeti állapotok a "felénk" eső állapotok lesznek, $m = 4$ $n = 5$ esetben ezek a $4, 8, 12, 16, 20$ állapotok. A célállapotok az átjáró "másik végén" lesznek, vagyis az $1, 5, 9, 13, 17$ állapotok. A nyelő állapotaink pedig az átjáró 2 szélén fognak elhelyezkedni, ezek a $2, 3, 18, 19$ sorszámmal jelöltek lesznek. (K: kezdeti, C: cél, NY: nyelő)

sorszám	[, 1]	[,2]	[, 3]	[, 4]	[, 5]
[1,]	C	C	C	C	C
[2,]	NY	6	10	14	NY
[3,]	NY	7	11	15	NY
[4,]	K	K	K	K	K

Azt fogom megvizsgálni, hogy mennyi a különböző állapotokból a célpontok elérési valószínűsége, azzal a feltétellel, hogy a folyamat nem érinti a nyelő állapotokat. Ekkor egy mátrixban lesznek eltárolva az elérési valószínűségek:

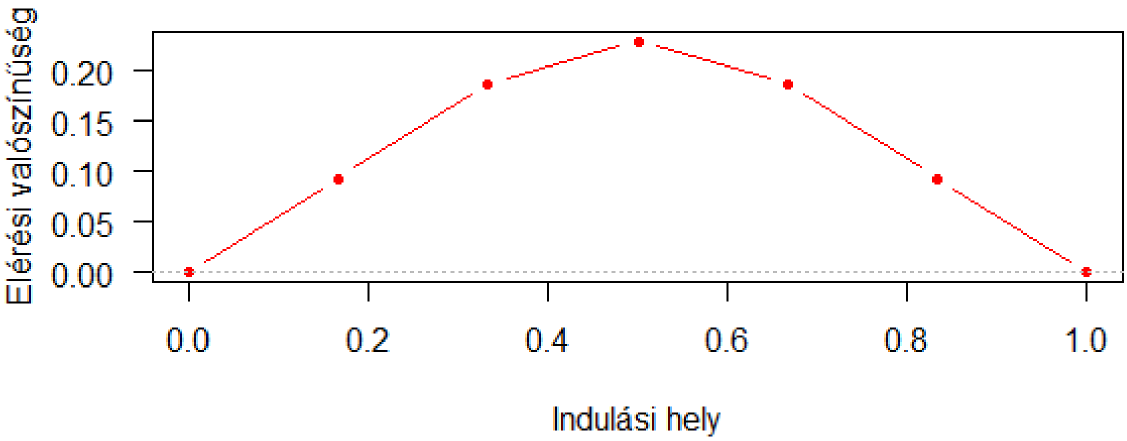
sorszám	[, 1]	[,2]	[, 3]	[, 4]	[, 5]
[1,]	1	1	1	1	1
[2,]	0	0.447	0.552	0.447	0
[3,]	0	0.237	0.313	0.237	0
[4,]	0.093	0.186	0.229	0.186	0.093

A triviális esetek a nyelő pontok és a célpontok. Annak a valószínűsége, hogy bármelyik nyelő pontból eljutunk a célpontok valamelyikébe azzal a feltétellel, hogy nem érintünk nyelő állapotot, nyilvánvalóan 0 lesz, mivel nyelő állapotból indultunk. Hasonlóan, ha egy célpontból akarunk eljutni egy célpontba úgy, hogy nem érint a folyamat nyelő állapotot 1 valószínűséggel bekövetkezik, mivel a kiinduló állapot egy cél állapot volt. A $[4, 1]$ sorszámú start állapotból ez a valószínűség 0.093, a $[3, 3]$ sorszámú belső pontból 0.313. A belső pontok nem túl érdekesek számunkra, ugyanúgy viselkednek, mint a start állapotok, ezért elég csak a start állapotokkal foglalkozni. (**3.1.2**)

Továbbá meg lehet figyelni, hogy a mátrix a 3. oszlopra szimmetrikus, ami abból következik, az átjáró és a nyelő pontok is szimmetrikusan helyezkednek el. A valószínűségek kívülről befelé fognak monoton növekedni, ergo a középső elemnél/elemeknél lesz a legnagyobb, mivel a nyelő pontoknak már egyre kisebb hatása lesz a középső elemekre.

Innentől kezdve csak a start pontokból fogjuk megfigyelni a valószínűségeket, az elérési valószínűség mátrix utolsó sorából. Továbbá az ábrázoláskor az átjáró le lesz *normálva*, tehát a $[0,1]$ intervallumon fogjuk ábrázolni a hozzátartozó valószínűségeket. Az 1. ábra az (5.1) programkóddal készült

Így az előző táblázat utolsó sora ábrázolva az alábbi módon fog kinézni:



1. ábra. 4x5-s átjáró

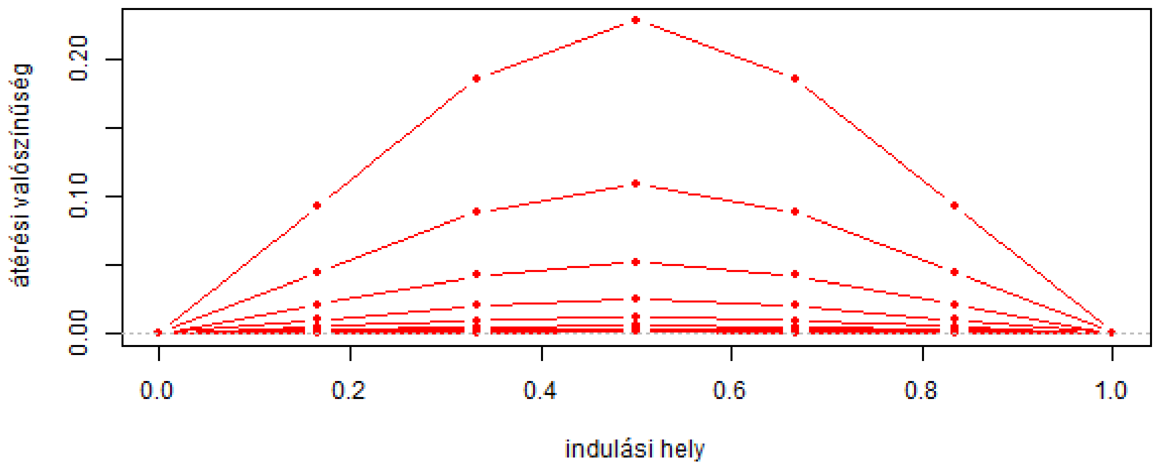
4.2. Aszimptotikus vizsgálat

Most vizsgáljuk meg mi történik aszimptotikusan. Rögzítsük az egyik paramétert miközben a másikat változtatjuk (növeljük). Nyilvánvalóan nem tudjuk azt vizsgálni, hogy a tábla valamelyik paraméterben ∞ nagy legyen, de érdemes megvizsgálni, azokat az eseteket amikor az m/n hányados viszonylag nagy, vagy ha kicsi.

- Legyen $n = 5$ és $m = 4 : 12$, tehát n rögzítve van és m-et növeljük.

  Ekkor az értékek táblázatba foglalva az alábbiak:

sorszám	[, 1]	[,2]	[, 3]	[, 4]	[, 5]
$m = 4$	0.093	0.186	0.228	0.186	0.093
$m = 5$	0.044	0.088	0.109	0.088	0.044
$m = 6$	0.0209	0.0418	0.0520	0.0418	0.0209
$m = 7$	0.010	0.198	0.246	0.198	0.010
$m = 8$	0.005	0.009	0.012	0.009	0.005
$m = 9$	0.00222	0.00443	0.00552	0.00443	0.00222
$m = 10$	0.00105	0.00210	0.00261	0.00210	0.00105
$m = 11$	0.00050	0.00099	0.00124	0.00099	0.00050
$m = 12$	0.00023	0.00047	0.00059	0.00047	0.00023

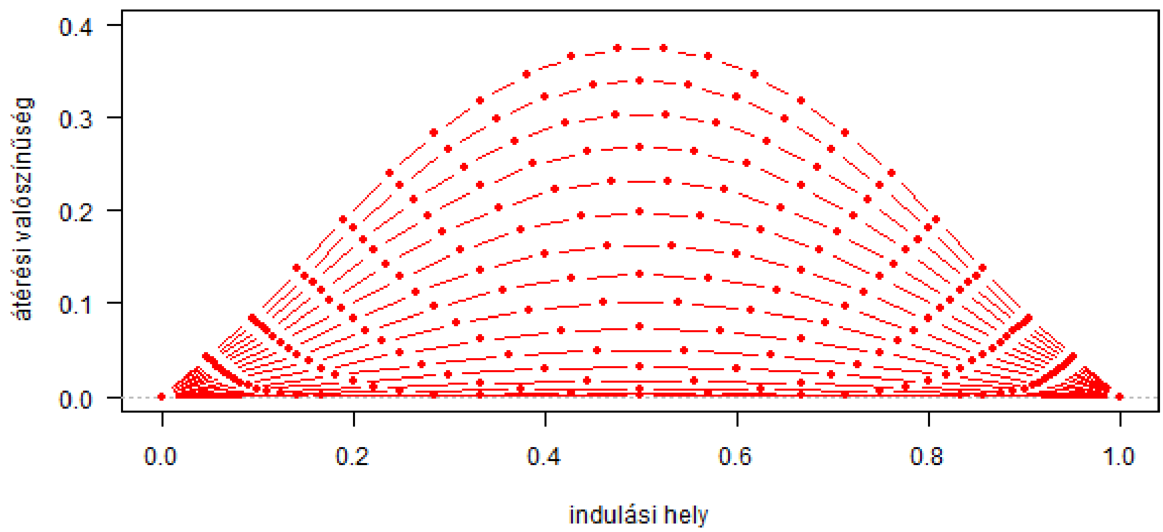


2. ábra. 5 széles átjáró változó hosszúsággal

A táblát *hosszabbítottuk*, vagyis a start és cél pontok egyre messzebb kerültek egymástól. Emellett a nyelő pontok száma is növekedett. Minél hosszabb a tábla, annál nagyobb utat kell bejárni, hogy eljussunk egy célpontba, tehát nagyobb m értékekhez kisebb elérési valószínűségek fognak tartozni.

Ha $m \to \infty$-hez, akkor a valószínűségek tartanának 0-hoz. Érdemes megfigyelni, hogy $m = 6$ esetén az elérési valószínűségek, mind kisebbek, mint $\frac{1}{10}$, $m = 9$ esetén már $\frac{1}{100}$-nál is kisebbek, $m = 12$ esetén pedig már $\frac{1}{1000}$-t sem érik el. Tehát 3 soronként, nagyjából tizedére csökkennek az egymáshoz tartozó valószínűségek.

Továbbá a második oszlophoz tartozó értékek nagyjából 2-szer akkorák, mint az első oszlopban levők.

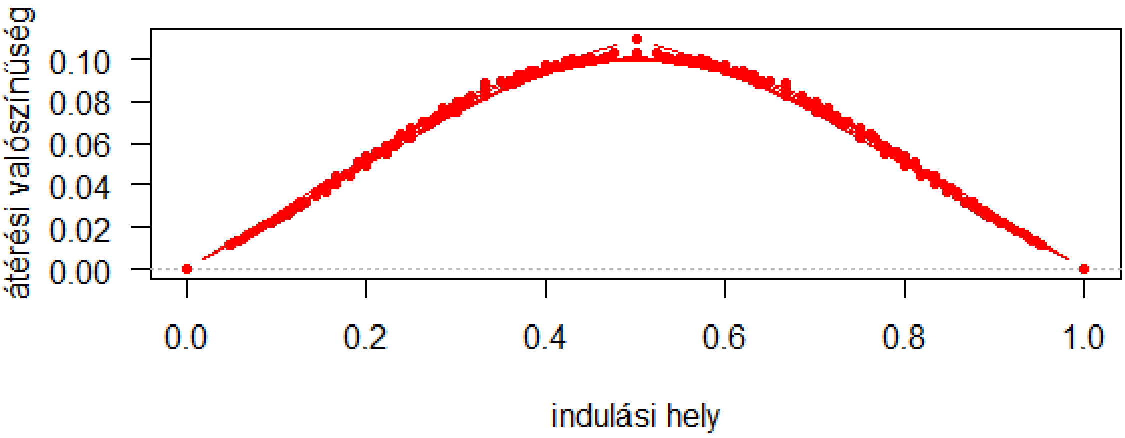


3. ábra. 12 hosszú átjáró változó szélességgel

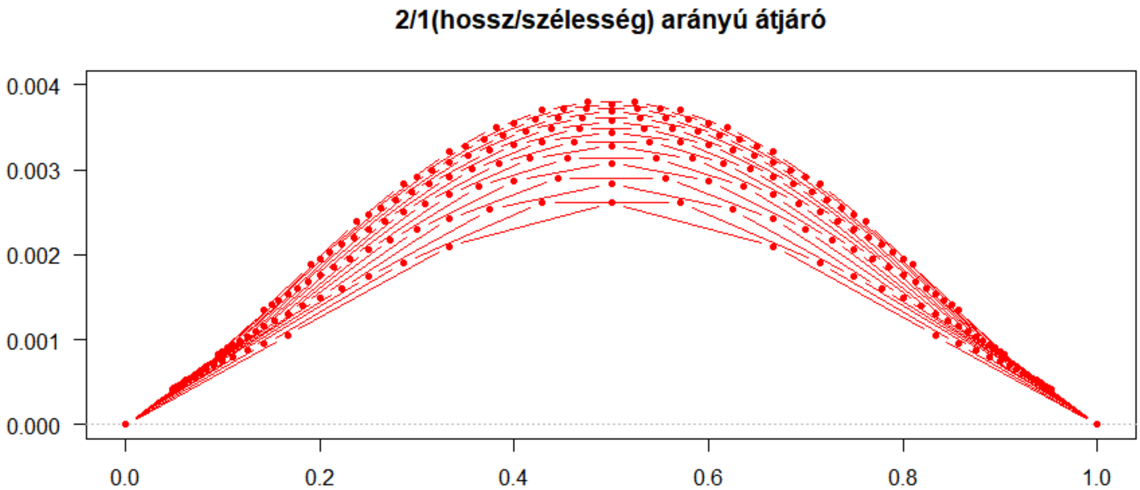
A táblát ebben az esetben *szélesítettük*, azaz m-et rögzítjük és n-et növeljük egyesével. A 3.ábrán $m = 12$, $n = 5 : 20$ volt. Ilyenkor a nyelő pontok száma változatlan marad, a start és a cél pontok száma pedig növekszik. Várhatóan így az történt amit várni lehetett, kívülről-befelé nőnek a valószínűségek, és a tábla szélesítésével a görbék pontjai egyre közelebb fognak kerülni az 1 valószínűséghez. Az alábbi táblázatban kigyűjtöttem csak a középső 5 darab célpontra az elérési valószínűségeket $\{m = 12; n = 5, 20, 100, 200\}$ esetén. Az alábbi táblázat is mutatja, hogy, ha m rögzített és $n \to \infty$ akkor 1-hez tartanak a valószínűségek. A 2. és 3. ábra (5.2) programkóddal készültek. Az adatok 5 tizedesjegyre vannak kerekítve.

n	[, 1]	[,2]	[, 3]	[, 4]	[, 5]
$n = 5$	0.00023	0.00047	0.00059	0.00047	0.00023
$n = 20$	0.34588	0.36439	0.37371	0.37371	0.36439
$n = 100$	0.99688	0.99699	0.99705	0.99705	0.99699
$n = 200$	1	1	1	1	1

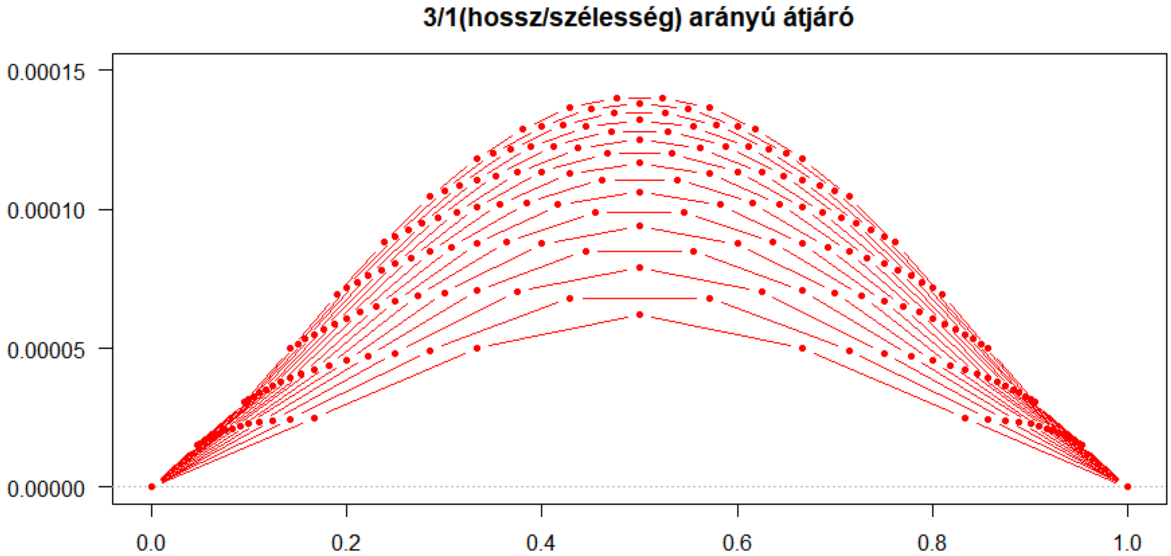
4.2.1. Adott arányú tábla



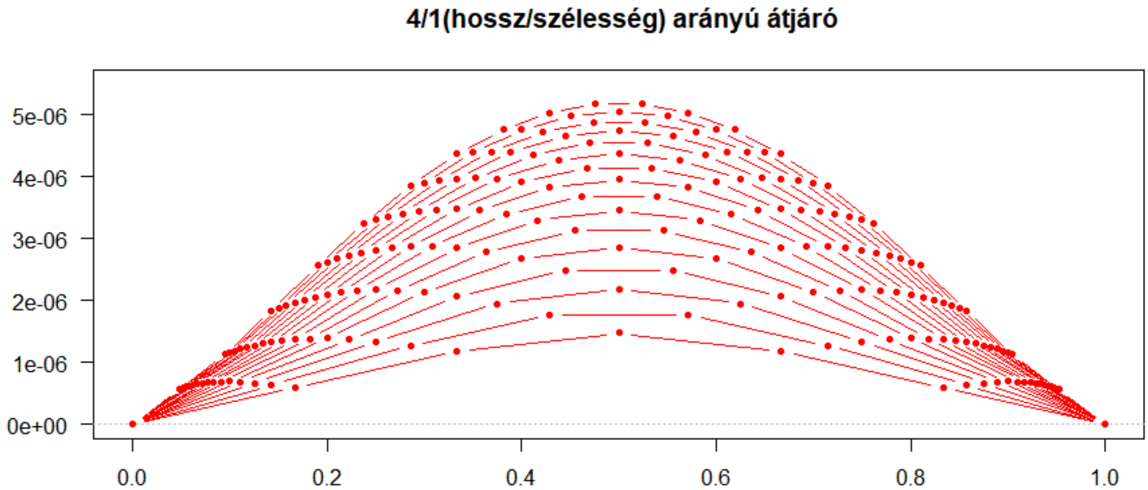
4. ábra. Négyzetes átjáró



5. ábra. **x: Indulási hely, y: Elérési valószínűség**



6. ábra. **x: Indulási hely, y: Elérési valószínűség**



7. ábra. **x: Indulási hely, y: Elérési valószínűség**

Ebben az esetben azt vizsgáltuk meg, hogy az elérési valószínűség görbék, hogy viselkednek különböző hossz/szélesség arányú átjárók esetén. 4 ilyen táblát vizsgáltunk 1/1, 2/1, 3/1 , 4/1 aránnyal. Miközben a k méretét növeltük 5-től 20-ig.

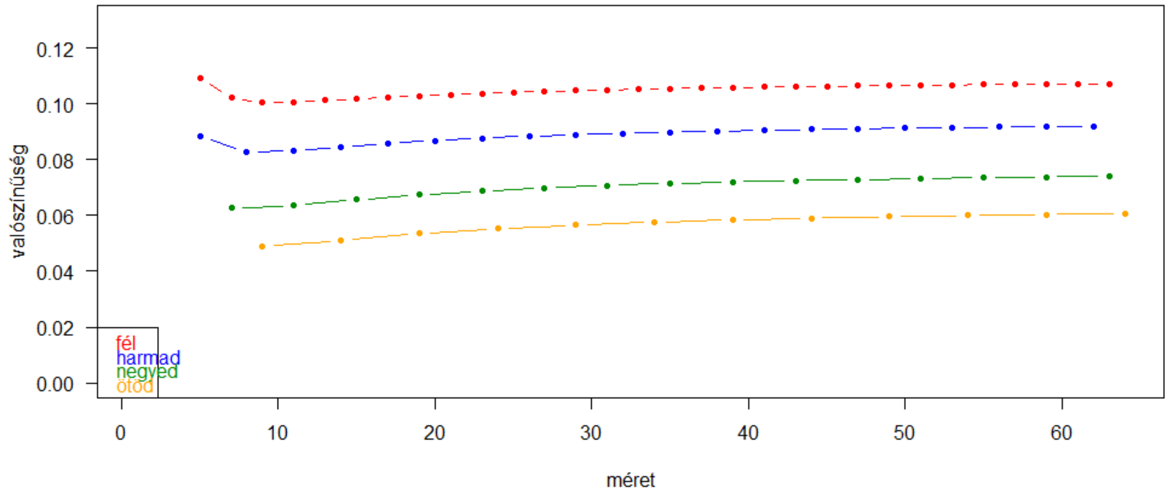
Ez azt jelenti, hogy első esetben 5×5, 6×6,..., 20×20 méretű átjárók vannak egy grafikonon ábrázolva. A második esetben, ugyanilyen logika mentén 10×5, 12×6, $14 \times 7, \ldots$ 40×20-ig vannak a táblák együtt ábrázolva, harmadik, negyedik eset hasonlóan az eddigiekből.

Ha a grafikonokat egymással hasonlítjuk össze, akkor kiderül, hogy minél nagyobb ez az arány annál kisebbek lesznek az elérési valószínűségek. Például egy 5×5-s táblában nagyságrendekkel nagyobb lesz a valószínűség, mint egy 15×5-s esetben.

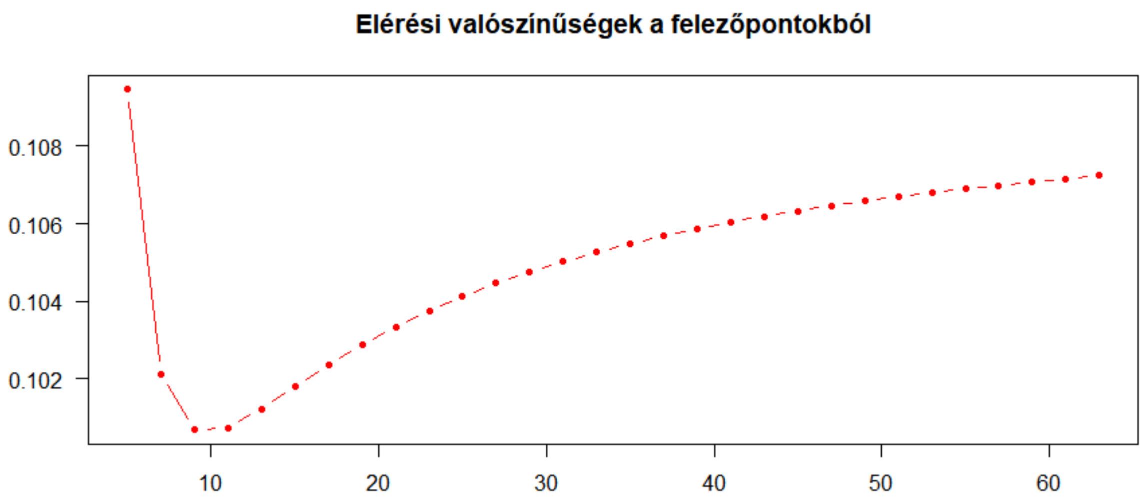
Továbbá, ha külön-külön vizsgáljuk a grafikonokat, akkor megfigyelhető mindegyik esetben, hogy az azonos struktúrájú gráfok esetén a nagyobb tábla esetén az elérési valószínűségek kicsit meg fognak nőni a méret növelésével. Sőt annál jobban fognak változni, minél nagyobb a hossz/szélesség arány. Vagyis, ha például megnézzük, hogy a táblánk 5×5-ről változik 20×20-ra, akkor arányaiban nem lesz akkora növekedés, mintha 20×5-ről 80×20-ra változtatnánk, habár mindkét esetben a táblák hosszát és szélességét $(4-4)$-szeresére növeltük.

A (4.), (5.), (6.), (7.) ábrák a (5.2.1) programkóddal készültek.

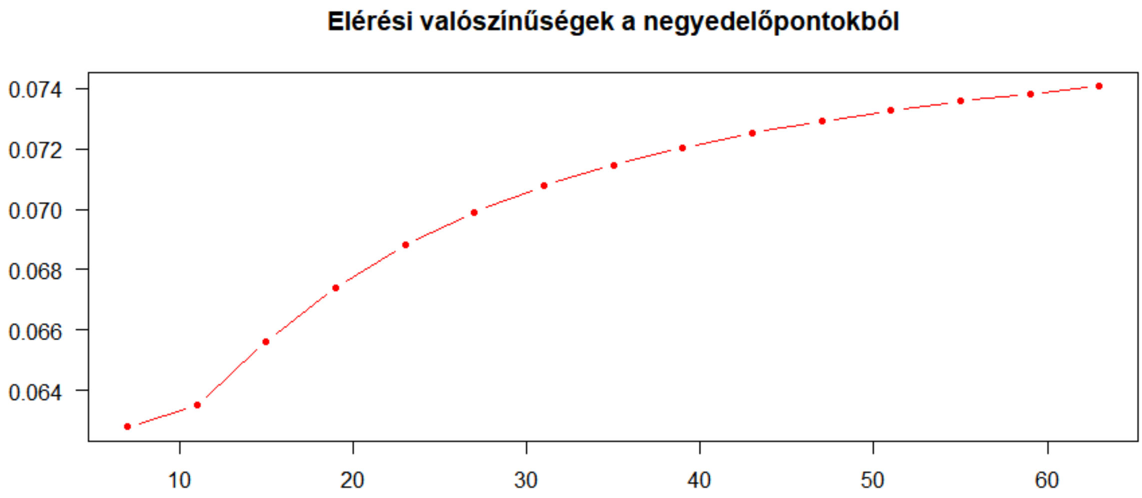
4.2.2. Négyzetes átjáró osztópontokkal



8. ábra. Négyzetes átjáró esetén az osztópontokból az elérési valószínűségek



9. ábra. **x-tengely: Méret, y-tengely: Elérési valószínűség**



10. ábra. **x-tengely: Méret, y-tengely: Elérési valószínűség**

n	3	4	...	31	32
$2n-1$	5	7	...	61	63

1. táblázat. Felezőpontok

n	2	3	...	20	21
$3n-1$	5	8	...	59	62

2. táblázat. Harmadolópontok

n	2	3	...	15	16
$4n-1$	7	11	...	59	63

3. táblázat. Negyedelőpontok

n	2	3	...	12	13
$5n-1$	9	14	...	59	64

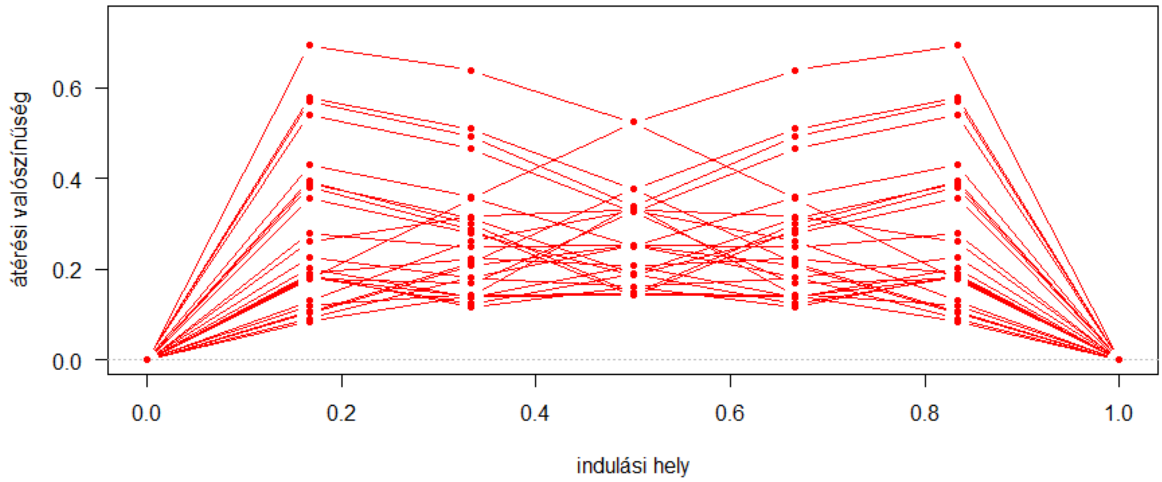
4. táblázat. Ötödölőpontok

Ebben az esetben a különböző osztópontokból (felező, harmadoló, negyedelő, ötödölő) vizsgáltam az elérési valószínűségeket négyzetes átjárók esetén. A fenti 4 táblázat első sora az osztópont sorszámát, a második sora a tábla méretét mutatja. Ez azt jelenti, hogy például a felezőpontok esetén egy 7×7-s méretű táblánál start sor 4. eleme lesz a középső elem (felezési pont). A hozzárendelési szabály alapján látható, hogy n egyesével van növelve, emellett a tábla mérete 2, 3, 4, 5-tel növekszik, attól függően, hogy melyik osztópont. A felezőpontból indulás minden másodikra, harmadoló csak minden harmadikra van, stb.

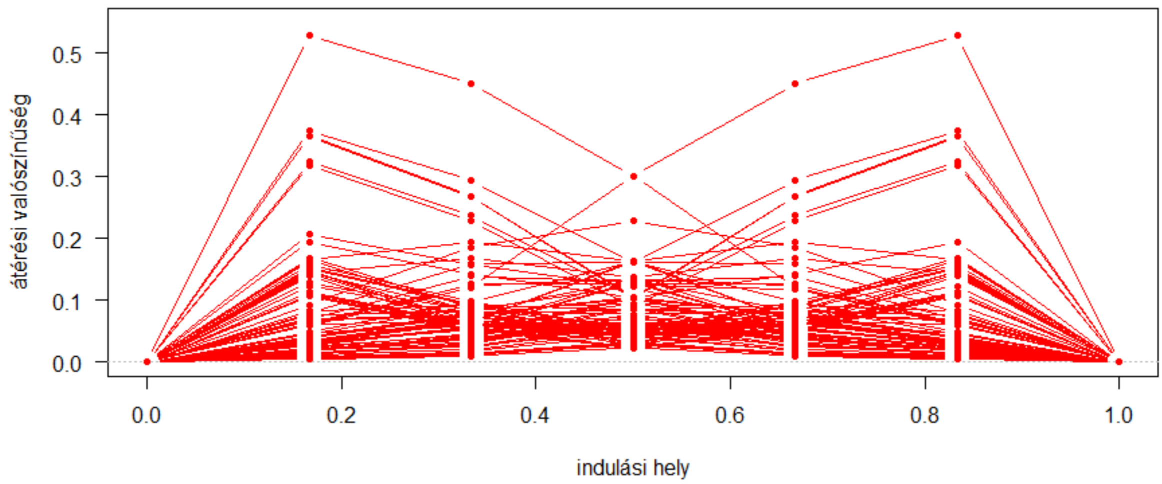
Az együttes ábrából látszik, hogy a felezőpontok elérési valószínűsége a legnagyobb, az ötödölőpontoké pedig a legkisebb. Ennek az a magyarázata, hogy fix méretű átjáró esetén a valószínűségek kívülről befelé növekednek. Továbbá, ha a tábla mérete legalább 10×10, akkor a tábla növelésével mindegyik esetben a valószínűségek növekednek és konvergálnak egy adott számhoz. A felező és a harmadolópontok esetén a tábla kis mérete miatt megfigyelhető egy kezdeti csökkenés, amíg a tábla mérete viszonylag kicsi (5×5 vagy 8×8), de ez nem játszik szerepet az aszimptotikus vizsgálatban.

A (8.), (9.), (10.) ábrák a (5.2.2) programkóddal készültek.

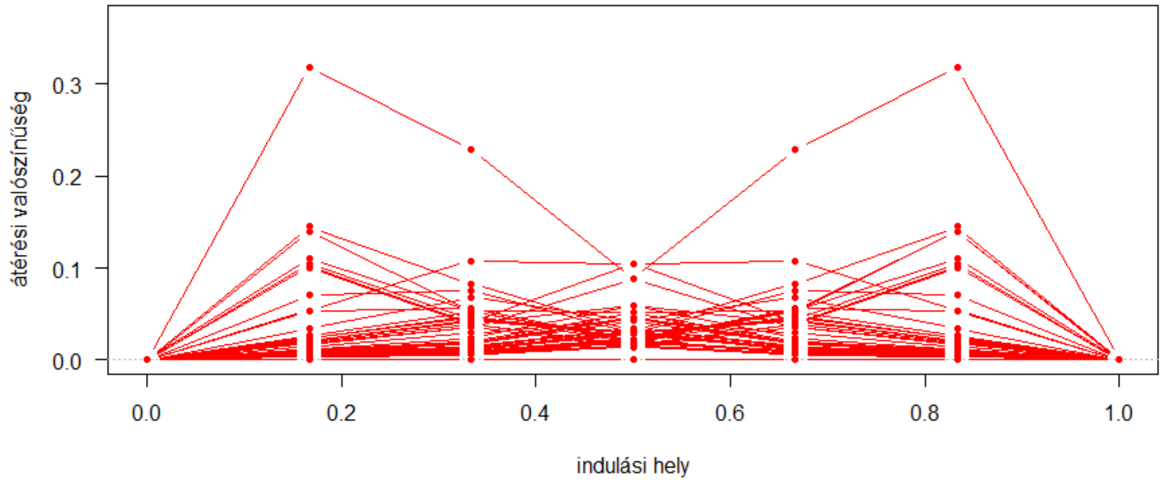
4.3. Fix átjáró különböző helyzetű nyelőkkel



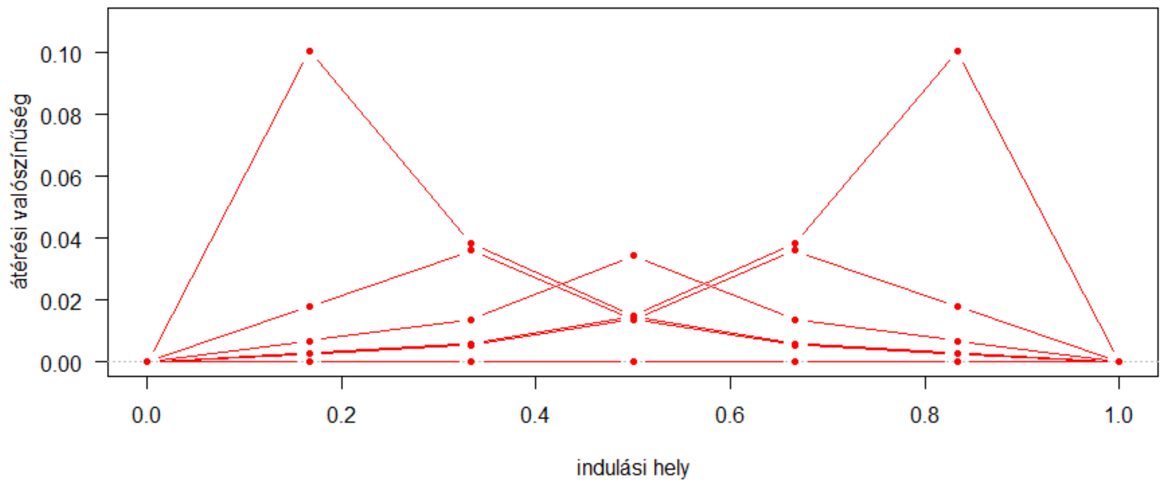
11. ábra. 4 × 5 átjáró (1-1) nyelővel



12. ábra. 4 × 5 átjáró (2-2) nyelővel



13. ábra. 4×5 átjáró (3-3) nyelővel



14. ábra. 4×5 átjáró (4-4) nyelővel

Itt egy konkrét táblára (4×5) vizsgáltam meg, mi történik, ha a nyelő pontok számát és helyét változtatom. Mindegyik esetben kirajzolja az összes elérési valószínűség görbét.

A nyelő pontokra egy kikötést tettem, hogy a második és harmadik sorból ugyanannyit választok ki, így hasonló marad a gráf strukturálisan. A fentebbi ábrákon rendre: 1-1, 2-2, 3-3, 4-4 nyelő pontot választottam ki. Az első esetben $\binom{5}{1}\binom{5}{1} = 25$ a második esetben $\binom{5}{2}\binom{5}{2} = 100$ ábra van, és így tovább.

Triviális eset amikor 2×5 nyelő állapotunk van (ezt nem ábrázoltam), ugyanis ez úgy lehet, hogy a 2. és 3.sorban csak nyelő pontok vannak, tehát bármelyik célpont elérési valószínűsége 0 lesz.

Továbbá, ha a 17. ábrát vizsgáljuk a 25 lefutásból csak 5 darab lesz, ahol nem mindegyik elérés 0, ezek pontosan azok az esetek, amikor a nyelő pontok egymás alatt helyezkednek el. Az alábbi táblázat egy ilyen esetet mutat. Itt csak akkor jut el a folyamat egy célpontba, ha végigmegy a 2. oszlopon.

sorszám	[, 1]	[,2]	[, 3]	[, 4]	[, 5]
[1,]	C	C	C	C	C
[2,]	NY	9	NY	NY	NY
[3,]	NY	9	NY	NY	NY
[4,]	K	K	K	K	K

Hasonlóan az eddigiekhez, a görbe akkor lesz szimmetrikus az $x = 0,5$ egyenesre, ha a nyelő pontok is szimmetrikusan helyezkednek el. Nyilvánvalóan mivel fix táblánk van, ha növeljük a nyelő pontok számát az elérési valószínűségek csökkenni fognak.

5. Függelék

5.1. Átjáró fix nyelőkkel

```
%m*n-es átjáró létrehozása
library(markovchain)
folyoso <- function(hossza,szelte,rajz=FALSE)
{
  m <- hossza
  n <- szelte
  allapot_nevek <- apply(cbind("[",rep(1:m,n),",",rep(1:n,each=m),"]"),1,
                       function(u) paste(u,collapse = ""))
  %állapot sorszám mátrix létrehozása
  allapot_sorszam <- matrix(1:(n*m),m,n)

  %állapot típus mátrix létrehozása
  allapot_tipus <- matrix(9,m,n)
  allapot_tipus[c(1,m),] <- c(6,7)
  allapot_tipus[,c(1,n)] <- rep(c(5,8),each=m)
  allapot_tipus[cbind(c(1,m,1,m),c(1,1,n,n))] <- 1:4

  % valószínűségek a különböző típusú csúcsokból
  p <- c(1/2,1/2,1/2,1/2,1/3,1/3,1/3,1/3,1/4)

   % az [i,j] állapotból melyik sorszámú állapotokba lehet menni
  hova_mehet <- function(tipus,i=2,j=2)
    {
      oda <-
       switch(paste(tipus),
                    "1" = c(allapot_sorszam[2,1],
                           allapot_sorszam[1,2]),
```

```
                    "2"= c(allapot_sorszam[m-1,1],
                           allapot_sorszam[m,2]),
                    "3"= c(allapot_sorszam[1,n-1],
                           allapot_sorszam[2,n]),
                    "4"= c(allapot_sorszam[m,n-1],
                           allapot_sorszam[m-1,n]),
                    "5"= c(allapot_sorszam[i-1,1],
                           allapot_sorszam[i+1,1],
                           allapot_sorszam[i  ,2]),
                    "6"= c(allapot_sorszam[1,j-1],
                           allapot_sorszam[2,j  ],
                           allapot_sorszam[1,j+1]),
                    "7"= c(allapot_sorszam[m  ,j-1],
                           allapot_sorszam[m-1,j  ],
                           allapot_sorszam[m  ,j+1]),
                    "8"= c(allapot_sorszam[i  ,n-1],
                           allapot_sorszam[i-1,n],
                           allapot_sorszam[i+1,n]),
                    "9"= c(allapot_sorszam[i,j-1],
                           allapot_sorszam[i-1,j],
                           allapot_sorszam[i+1,j],
                           allapot_sorszam[i,j+1]),
                   NA)
   return(oda)
  }

% átmenet-valószínűségmátrix feltöltése
P <- matrix(0,m*n,m*n)
for(i in 1:m)for(j in 1:n)
  {
```

```
     tipus <- allapot_tipus[i,j]
     sorszam <- allapot_sorszam[i,j]
     P[sorszam,hova_mehet(tipus,i,j)] <- p[tipus]
    }

 %a Markov-lánc felépítése
  MC <- new("markovchain",transitionMatrix=P)

  %a speciális állapotok kijelölése
  cel <- ((1:n)-1)*m+1
  nyel <- c(2:(m-1),((n-1)*m+2):(n*m-1))
  start <- (1:n)*m

  hp <- committorAB(MC,cel,nyel) % elérési valószínűség az összes állapotból
  hps <- matrix(hp,m,n)[m,] % elérési valószínűség a start állapotokból

  % ábra létrehozása
  if(rajz)
   {
    x <- seq(0,1,length=n+2)
    plot(x,c(0,hps,0),t='b',col="red",pch=20,las=1,
           xlab="Indulási hely",ylab="Elérési valószínűség")
    abline(h=0,col="gray",lty=3)
   }

  return(hps)
}
% 4 hosszú 5 széles átjáró
folyoso(4,5) %elérési valószínűségek
folyoso(4,5,TRUE) %rajzzal
```

5.2. Aszimptotikus vizsgálat

```
%5 széles 4:12 hosszú átjáró
n <- 5
hpm <- matrix(NA,0,n)
for(m in 4:12) hpm<-rbind(hpm,folyoso(m,n))
round(hpm,5) % az értékek egy táblázatba foglalva

%egy ábrán a különböző hosszúságú átjárók
x <- seq(0,1,length=n+2)
matplot(x,t(cbind(0,hpm,0)),t="b",lty=1,col="red",pch=20,
            xlab="indulási hely",
            ylab="elérési valószínűség")
abline(h=0,col="gray",lty=3)

%12 hosszú 5:20 széles átjáró
m <- 12
hpl <- list(NULL)
for(n in 5:20) hpl <- c(hpl,list(folyoso(m,n)))
hpl <- hpl[-1]

% segédfüggvény a normált valószínűségek ábrázolására
prob.curve <- function(u)
  {
    x <- seq(0,1,length=length(u)+2)
    u <- c(0,u,0)
    points(x,u,t="b",lty=1,col="red",pch=20)
  }

  % egy ábrán a 16 darab átjáró
  par(mar=c(4,5,3,1))
```

```
  plot(.5,0,xlim=c(0,1),ylim=c(0,.4),t="n",las=1,
            xlab="indulási hely",
            ylab="elérési valószínűség")
  abline(h=0,col="gray",lty=3)
  w <- sapply(hpl,"prob.curve")
```

5.2.1. Adott arányú tábla

```
%Négyzetes átjáró 5:20 hosszúság esetén
% többi eset folyoso(2*k,k),folyoso(3*k,k),folyoso(4*k,k)
hpl <- list(NULL)
for(k in 5:20) hpl <- c(hpl,list(folyoso(k,k)))
hpl <- hpl[-1]

%ábrázolása
 plot(.5,0,xlim=c(0,1),ylim=c(0,.11),t="n",las=1,
            xlab="indulási hely",
            ylab="elérési valószínűség")
  abline(h=0,col="gray",lty=3)
  w <- sapply(hpl,"prob.curve")
```

5.2.2. Négyzetes átjáró osztópontokkal

```
%felezőpontok

%vizsgált hossz
h.50 <- 30 %többi eset h.33<-20,h.25<-15,h.20<-12
felezo <- vector("numeric",h.50) %többi eset h.33, h.25,h.20
mn50 <- seq(5,by=2,length=h.50) % méret
pont <- seq(3,by=1,length=h.50) % a felező pontok
rbind(mn50,pont)
```

```
for(k in 1:h.50)
   felezo[k] <- folyoso(mn50[k],mn50[k])[pont[k]]
   %felezőpontok ábrázolása
   plot(mn50,felezo,
     xlab="méret",ylab="valószínűség",
     t="b",col="red",pch=20,las=1)
%együtt az ábrák
plot(.5,0,t="n",las=1,
     xlim=c(1,64),ylim=c(0,0.13),
     xlab="méret",ylab="elérési valószínűség",)
points(mn50,felezo   ,col="red"   ,t="b",pch=20)
points(mn33,harmadolo,col="blue"  ,t="b",pch=20)
points(mn25,negyedelo,col="green4",t="b",pch=20)
points(mn20,otodolo  ,col="orange",t="b",pch=20)
legend("bottomleft",c("fél","harmad","negyed","ötöd"),
       text.col=c("red","blue","green4","orange"))
```

5.3. Fix átjáró különböző helyzetű nyelőkkel

Nagyon hasonló a fix nyelős esethez, annyi különbséggel, hogy először a nyelőpontok számát beállítom (1-1), (2-2), (3-3) vagy (4,4) majd az adott esetre megnézem a nyelőpontok összes lehetséges elhelyezkedését és lefuttatom a folyoso függvényt.
Ebben az esetben a programkódot nem részletezem.

Hivatkozások

[1] KAI LAI CHUNG, *Markov Chains with Stationary Transition Probabilities ,Springer-Verlag, New York,* 1967, 43-49, (19-23)

[2] CSISZÁR VILLŐ, *Diszkrét és folytonos paraméterű Markov láncok*

(https://web.cs.elte.hu/~villo/ml/ML.pdf)
Elérés időpontja: 2020. április, (7-8)

[3] https://cran.r-project.org/web/packages/markovchain/markovchain.pdf
Elérés időpontja: 2020. április, (11-18)

[4] https://www.datamentor.io/r-programming/s4-class/
Elérés időpontja: 2020. április, (9-11)

[5] https://hu.wikipedia.org/wiki/Markov-lánc
Elérés időpontja: 2020. április, (5)

[6] https://math.bme.hu/~szbalazs/oktatas/sztoch_info/het_4_Markov.pdf
Elérés időpontja: 2020. április, (6)