

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO  
ODDELEK ZA MATEMATIKO

Antonija Pršlja

**Primerjava in nadgradnja metod za računanje potenc  
prehodnih matrik markovskih verig**  
Doktorska disertacija



MENTOR: doc. dr. Damjana Kokol Bukovšek  
SOMENTOR: prof. dr. Matjaž Omladič

Ljubljana, 2015

Podpisana Antonija Pršlja izjavljam:

- da sem doktorsko disertacijo z naslovom *Primerjava in nadgradnja metod za računanje potenc prehodnih matrik markovskih verig* izdelala samostojno pod mentorstvom doc. dr. Damjane Kokol Bukovšek in somentorstvom prof. dr. Matjaža Omladiča in
- da Fakulteti za matematiko in fiziko Univerze v Ljubljani dovoljujem objavo elektronske oblike svojega dela na spletnih straneh.

Ljubljana, 9. avgust 2015

Podpis:

---

Iskrena hvala mentorici doc. dr. Damjani Kokol Bukovšek  
in sometorju prof. dr. Matjažu Omladiču  
za strokovno pomoč in nasvete,  
dr. Gregorju Šegi za vse plodne pogovore,  
moji najboljši prijateljici dr. Živi Štepančič za spodbudo  
in moji družini za podporo pri nastajanju dela.

---



# Povzetek

---

Markovske verige igrajo pomembno vlogo na različnih področjih, kot so biologija, finance in industrijska proizvodnja. Uporabljajo se za modeliranje, kako se sistem giblje med stanji v odvisnosti od časa. Pri tem je ključno, da je za napoved prehoda v novo stanje pomembna le informacija o trenutnem stanju, vso ostalo informacijo o zgodovini sistema pa lahko pozabimo. Kadar nas zanima analiza stroškov ali dobička določene operacije, se na markovskih verigah vpelje še dodatna informacija o ceni.

Če sta čas in množica stanj diskretna, govorimo o diskretnih markovskih verigah. V posebnem zaporedju prvih  $k$  členov diskretne markovske verige rečemo slučajni sprehod dolžine  $k$ . V realnih aplikacijah se pogosto izkaže potreba po izračunu povprečja cene slučajnega sprehoda dane dolžine  $z$  začetkom v določenem stanju. Ker analitičen izračun povprečja v praksi dostikrat ni možen, je potrebno zagotoviti vsaj dobre ocene. V doktorski disertaciji obravnavamo problem računanja ocene povprečja cene slučajnega sprehoda.

V osnovnem problemu računanja ocene povprečja je slučajni sprehod podan s prehodno matriko, neodvisno od časa, podobno je njegova cena določena s cenovno matriko, prav tako neodvisno od časa. V doktorski disertaciji ta problem posplošimo, kjer dopustimo, da se prehodna in cenovna matrika na določen čas spreminjata v odvisnosti od danih parametrov. Najprej predstavimo algoritem za reševanje osnovnega problema. Ta je temeljni gradnik v vseh treh nadaljnjih algoritmih, ki jih razvijemo za reševanje posplošenega problema. Glavna ideja v vseh predlaganih algoritmih temelji na uporabi metode Monte Carlo, ki omogoča, da oceno povprečja poiščemo s simulacijami.

Predlagane algoritme eksperimentalno ovrednotimo na testni množici podatkov. Monte Carlo algoritem za reševanje osnovnega problema primerjamo z algoritmom, ki uporablja metodo difuzijskih valčkov za računanje potenc matrik. Rezultati kažejo, da Monte Carlo algoritem prekaša algoritem z valčki v vseh primerih. Nazadnje med sabo primerjamo še algoritme za reševanje posplošenega problema. Prvi izmed predlaganih se izkaže za nedvomnega poraženca, medtem ko med drugima dvema ni večjih razlik. Vseeno dajemo prednost drugemu izmed predlaganih algoritmov.

**Math. Subj. Class. (2010):** 60J22, 91G60

**Ključne besede:** prehodna matrika, cenovna matrika, metoda Monte Carlo, slučajni sprehod, diskretna markovska veriga.



# Abstract

---

Markov chains play an important role in many fields as diverse as biology, finance and industrial production. They are used for modeling how a system moves between the states in time. The key property is that only the information about the current state is important for the prediction of moving into a new state, while all the information about the history of system can be forgotten. Whenever we are interested in the analysis of the price or profit of an operation, additional information about the cost is introduced to Markov chains.

If time and a set of states are discrete, we are talking about discrete-time Markov chains. In particular, a sequence of the first  $k$  items of discrete-time Markov chain is called a random walk of length  $k$ . In real applications there is often a need for calculating the average cost of a random walk of given length starting at certain state. Since analytical computation of the average is often not possible in practice, it is necessary to provide at least good estimates. In Thesis we consider the problem of computing estimates of the average cost of a random walk.

In the basic problem the random walk is given by a transition matrix that is time-independent and its cost is determined by the cost matrix that is also time-independent. In Thesis this problem is generalized such that we allow the transition matrix and the cost matrix to change at certain times depending on given parameters. First, we present an algorithm to solve the basic problem. It is a fundamental building block in all three subsequent algorithms that are developed to solve the generalized problem. The main idea in all proposed algorithms is to exploit Monte Carlo method, which allows to find estimates of the average by simulations.

The proposed algorithms are experimentally evaluated. Monte Carlo algorithm for solving basic problem is compared to the one that uses diffusion wavelets method for computing powers of matrices. Our results show that Monte Carlo algorithm outperformed the one using diffusion wavelets. Finally, the proposed algorithms for solving the generalized problem are compared to each other. The first of the proposed algorithms proves to be the clear loser, while the performance of the other two is not that different. However, the second one seems to have a slight advantage.

**Math. Subj. Class. (2010):** 60J22, 91G60

**Keywords:** transition matrix, cost matrix, Monte Carlo method, random walk, discrete Markov chain.





# Kazalo

---

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Motivacija	2
1.1.1	Sistem izposoje koles	2
1.1.2	Prevoz blaga s tovornjaki	3
1.2	Cilji in prispevki k znanosti	3
1.3	Pregled vsebine	4
<b>2</b>	<b>Markovske verige v diskretnem času</b>	<b>7</b>
2.1	Osnovne lastnosti markovskih verig	7
2.1.1	Prehodna porazdelitev in začetna porazdelitev	10
2.2	Klasifikacija stanj in verig	13
2.2.1	Nerazcepnost in periodičnost	13
2.2.2	Povrnljivost in minljivost	18
2.3	Stacionarna in limitna porazdelitev	22
<b>3</b>	<b>Metoda Monte Carlo</b>	<b>31</b>
3.1	Osnovni koncept	31
3.2	Generiranje vzorca	34
3.2.1	Inverzna transformacijska metoda	36
3.2.2	Generiranje markovskih verig	37
<b>4</b>	<b>Matematično upanje in potence prehodnih matrik</b>	<b>39</b>
4.1	Problem	39
4.1.1	Osnovni problem	40

4.1.2	Posplošitev osnovnega problema	41
4.2	Reševanje z metodo Monte Carlo	42
4.2.1	Reševanje posplošenega problema	44
4.3	Pristop, ki uporablja difuzijske valčke	51
<b>5</b>	<b>Eksperimentalni rezultati</b>	<b>55</b>
5.1	Testna množica podatkov	56
5.1.1	Simulacija umetnih podatkov	56
5.1.2	Simulacija sistema izmenjave koles	58
5.2	Testiranje metod in analiza rezultatov	59
5.2.1	Osnovni problem	59
5.2.2	Posplošeni problem	62
<b>6</b>	<b>Zaključek</b>	<b>67</b>
6.1	Nadaljnje delo	68
	<b>Literatura</b>	<b>69</b>
	<b>Stvarno kazalo</b>	<b>73</b>

---

# Uvod

Markovske verige so stohastični modeli, ki igrajo pomembno vlogo v številnih aplikacijah na različnih področjih kot na primer biologija [5, 37], finance [16, 31], kemija [26, 29], psihologija [1, 6] in industrijska proizvodnja [8, 10]. V grobem lahko rečemo, da se markovske verige uporabljajo za modeliranje, kako se sistem giblje med stanji v odvisnosti od časa. Prehodi med stanji sistema so določeni s pogojno verjetnostno porazdelitvijo, ki dodeli verjetnost prehoda v novo stanje glede na trenutno stanje. Torej je za napoved prehoda v novo stanje sistema pomembna le informacija o trenutnem stanju sistema, vso ostalo informacijo o zgodovini sistema pa lahko pozabimo.

Markovska veriga je natanko podana z vektorjem, ki opisuje začetno porazdelitev stanj, in prehodno matriko, ki določa pogojno verjetnostno porazdelitev prehodov. Velikokrat je podana samo prehodna matrika, pri čemer se potem tiho predpostavlja, da nas zanimajo vse začetne porazdelitve. Zaradi izreka o popolni verjetnosti zadostuje, da preučujemo samo takšne situacije, kjer je začetek markovske verige v poljubnem, vendar fiksnem stanju.

Markovski modeli se pogosto uporabljajo pri analizi stroškov ali dobička določene operacije. To pomeni, da na markovski verigi uvedemo cenovno funkcijo. Informacijo o cenovni funkciji podamo tako, da posameznemu prehodu priredimo njegovo ceno, ki jo shranimo v cenovno matriko. V nadaljevanju nas zanimajo markovske verige, opremljene z dodatno informacijo o ceni.

Kadar sta čas in množica stanj diskretna, govorimo o diskretnih markovskih verigah. V posebnem zaporedju prvih  $k$  členov diskretne markovske verige rečemo slučajni sprehod dolžine  $k$ . V realnih aplikacijah se često kaže potreba po izračunu povprečja cene slučajnega sprehoda dane dolžine z začetkom v določenem stanju – dve takšni aplikaciji bomo predstavili v naslednjem razdelku. Kot bomo videli v nadaljevanju, analitičen izračun povprečja v praksi dostikrat ni možen, zato je potrebno zagotoviti vsaj dobre aproksimacije. V disertaciji obravnavamo problem računanja ocene povprečja cene slučajnega sprehoda.

## 1.1 Motivacija

V tem razdelku obravnavamo potencialni aplikaciji, kjer je potrebno poznavanje povprečja cene slučajnega sprehoda: sistem izposoje koles in prevoz blaga s tovarnjaki. Pri sistemu izposoje koles izpostavimo še nekatere zanimive sorodne probleme.

### 1.1.1 Sistem izposoje koles

V mestih, kot so Pariz, Ljubljana in Benetke, je kolo postalo zelo popularno prevozno sredstvo, saj je njegova uporaba med drugim praktična, hitra, učinkovita, prijazna okolju in primerna tako za domačine kot turiste. Vsako mesto ima svojo različico sistema izposoje koles. Vsak sistem ima določeno število kolesarskih postajališč, enakomerno porazdeljenih po mestu v bližini mestnih znamenitosti.

Registriran uporabnik si lahko izposodi kolo na poljubnem postajališču in vrne kolo na katerem koli drugem postajališču. Prehodu iz enega postajališča v drugo preprosto rečemo vožnja. Različne kolesarske poti lahko večkrat dnevno prevozijo različni registrirani uporabniki. Kolesa so razpoložljiva podnevi in ponoči. Nasičenost poljubnega postajališča in povpraševanje po izposoji koles sta odvisna, zato sta morda različna po obdobjih posameznega dne.

Gibanje kolesa lahko opišemo z markovsko verigo, kjer kolesarska postajališča predstavljajo stanja, pripadajočo prehodno matriko pa lahko poiščemo takole. Predpostavimo, da za vsako kolo poznamo postajališči njegove izposoje in vrnitve skupaj s časom njegove izposoje in vrnitve. Opazujemo prevožene kolesarske poti med postajališči in po določenem času izračunamo frekvence pojavitev posameznih voženj iz vsakega postajališča do vsakega drugega postajališča. Tako iz zbranih podatkov zgradimo prehodno matriko.

Vsakemu prehodu iz poljubnega postajališča v poljubno postajališče določimo še ceno prehoda. Na primer, za ceno prehoda lahko vzamemo razdaljo med postajališčema, to je dolžino obstoječe kolesarske poti. Želeli bi izračunati povprečje prevoženih kilometrov kolesa po  $k$  vožnjah pri danem začetnem postajališču. To bi zagotovo zanimalo lastnike sistema na primer za oceno, kdaj je treba kolesa servisirati oziroma kdaj je treba kolo zamenjati.

V splošnem je uporaba koles (izposoja in vrnitev) različno porazdeljena skozi dan, zato lahko ločeno obravnavamo prevožene kolesarske poti v recimo treh različnih časovnih obdobjih dneva: dopoldan, popoldan in zvečer. Tako lahko konstruiramo tri različne prehodne matrike, ki opisujejo prehode koles med postajami v dopoldanskem, popoldanskem in večernem času. Z drugimi besedami, prehodna matrika se spreminja s časom. Prav tako lahko ceno prehoda obravnavamo ločeno po obdobjih. Na primer, dopoldan se uporabniki peljejo direktno na cilj, popoldan gredo med potjo še v trgovino in naredijo ovinek, zvečer pa krožijo naokrog. Ne samo, da se prehodna matrika torej spreminja s časom, tudi pripadajoča cenovna matrika se spreminja s časom.

## 1.2 Cilji in prispevki k znanosti

S sistemom izposoje koles so zanimivi še drugi sorodni problemi. S strani uporabnika bi bilo idealno, če bi si lahko kadar koli izposodil oziroma vrnil kolo na poljubni postaji. Vendar pa lahko na najbolj obiskanih postajališčih v prometni konici pride do pomanjkanja prostih koles ali pa prostih mest za vrnitev koles. Nastali problem se lahko rešuje s prerazporeditvijo koles po mestu z majhnimi tovornjaki – če se postajališče prekomerno prazni, se kolesa pripeljejo, če pa se postajališče prekomerno polni, se kolesa odpeljejo na bolj prazna postajališča. Tako se poskuša enakomerno porazdeliti kolesa po postajališčih, da zadostijo čim večjemu povpraševanju izposoje koles. Več o tem problemu si lahko bralec prebere v [9, 12]. Problem povpraševanja po uporabi in izposoji koles je podoben problemu razpoložljivih postelj v bolnišnici, kjer morajo pacienti v primeru prezasedenosti oditi v drugo bolnišnico [2, 3]. Podoben problem se pojavi tudi v sistemu video na zahtevo, in sicer pri dodeljevanju filmov večkratnim strežnikom [20, 21].

### 1.1.2 Prevoz blaga s tovornjaki

Osnovni koncept naslednje aplikacije je bil predstavljen v [19]. Recimo, da imamo v lasti avtoprevozniško podjetje, ki s tovornjaki dostavlja blago med različnimi mesti. Gibanje tovornjaka obravnavamo kot markovsko verigo, kjer mesta predstavljajo stanja, pripadajočo prehodno matriko pa lahko dobimo podobno kot v primeru sistema izposoje koles. Prehodu tovornjaka iz enega mesta v drugo enostavno rečemo prevoz. Ceno prevoza naj predstavljajo stroški, ki jih naredimo pri dostavi blaga iz enega mesta v drugo. Tokrat želimo izračunati povprečne stroške tovornjaka po  $k$  prevozih iz danega začetnega mesta.

Koncept lahko posplošimo. Ker mora voznik tovornjaka po osmih urah vožnje narediti osem ur premora, je prevoz blaga smiselno obravnavati v dveh delih – posebej zjutraj in pozno popoldan. Pripadajočo markovsko verigo v tem primeru podamo z dvema prehodnima matrikama. Podobno lahko tudi ceno prevoza obravnavamo posebej zjutraj in pozno popoldan.

## 1.2 Cilji in prispevki k znanosti

V luči razdelka 1.1 ločimo dva koncepta: osnovnega, kjer se prehodna in cenovna matrika s časom ne spreminjata, in splošnega, kjer dopuščamo možnost, da imamo za vsak čas  $n$  drugačno prehodno in cenovno matriko. Drugače rečeno, v osnovnem problemu prehodna in cenovna matrika nista odvisni od časa, medtem ko v splošnem sta. Posplošitev osnovnega koncepta tako, da bo modeliral aplikaciji v razdelku 1.1, predstavlja prvi cilj doktorske disertacije.

Reševanje osnovnega problema je bilo na kratko obravnavano v [19], kjer avtorji predlagajo algoritem, ki temelji na računanju potenc prehodnih matrik. Njihov pristop vključuje prevedbo problema na primerno integralsko obliko, diskretizacijo integrala z neko napako in nato računanje vsakega člena diskretizacije, zopet z neko napako, z metodo difuzijskih valčkov za izračun potenc prehodne matrike. Bolj

podroben opis njihovega algoritma, ki ga bomo krajše imenovali algoritem PDV, bomo podali v razdelku 4.3. Na tem mestu je potrebno izpostaviti, da se metoda difuzijskih valčkov zadovoljivo obnaša v primeru, ko ima matrika nizek numerični rang in je potenca oblike  $2^p$  za naravno število  $p$ . Algoritem PDV je torej omejen na primer, ko ima prehodna matrika nizek numerični rang in je dolžina prehoda enaka  $k = 2^p$ . Predstavitev drugačnega pristopa, ki je uporaben za poljubno prehodno matriko in poljubno dolžino  $k$ , je naslednji cilj doktorske disertacije.

Poleg tega algoritma PDV ni mogoče uporabiti za reševanje posplošenega problema. Glavni cilj doktorske je razvoj učinkovitih metod za reševanje slednjega problema. Ključna ideja pri vseh predlaganih pristopih temelji na uporabi metode Monte Carlo, ki omogoča, da oceno povprečja poiščemo s simulacijami.

V disertaciji so predstavljeni naslednji prispevki k znanosti:

- (i) *Posplošitev osnovnega problema.* V osnovnem problemu računanja ocene je slučajni prehod podan s prehodno matriko, neodvisno od časa, podobno je njegova cena določena s cenovno matriko, ki je prav tako neodvisna od časa (glej podrazdelek 4.1.1). Problem je posplošen tako, da se prehodna in cenovna matrika lahko na določen čas spreminjata v odvisnosti od danih parametrov (glej podrazdelek 4.1.2).
- (ii) *Razvoj metod za reševanje posplošenega problema.* Predstavljeni so algoritmi PovprečjeNehomogene1 (PN1), PovprečjeNehomogene2 (PN2) in PovprečjeNehomogene3 (PN3) za reševanje posplošenega problema (glej razdelek 4.2.1). Algoritem PN1 je neposredna posplošitev algoritma za reševanje osnovnega problema. Algoritem PN2 je nadgradnja algoritma PN1 in prevede problem na reševanje posebnega posplošenega problema. Algoritem PN3 je nadaljnja nadgradnja algoritma PN2 in reševanje posebnega posplošenega problema prevede na reševanje osnovnega problema.
- (iii) *Eksperimentalna analiza metod.* Vse metode so eksperimentalno ovrednotene na testni množici podatkov (glej razdelek 5.1). Najprej se osnovni algoritem Monte Carlo primerja z algoritmom PDV (glej podrazdelek 5.2.1). Nato se med sabo primerja algoritme PN1, PN2 in PN3 (glej razdelek 5.2.2).

Deli doktorske disertacije so bili predstavljeni na mednarodni konferenci International Workshop on Matrices and Statistics [34] in v znanstveni reviji [35].

### 1.3 Pregled vsebine

Uvodnemu poglavju doktorske disertacije sledi poglavje, ki podaja osnovne koncepte v teoriji diskretnih markovskih verig in so potrebna podlaga za razumevanje vsebine. Najprej predstavimo osnovne lastnosti markovskih verig, nato sledi njihova klasifikacija. Poglavje zaključimo z obravnavo limitne in stacionarne porazdelitve markovske verige.

### 1.3 Pregled vsebine

V tretjem poglavju predstavimo osnovno idejo metode Monte Carlo za izračun ocene povprečja, ki je temeljni gradnik naših algoritmov. Podamo formalno analizo natančnosti metode Monte Carlo, nato se posvetimo algoritmu za generiranje vzorca. Posebno pozornost namenimo generiranju markovskih verig.

Četrto in peto poglavje predstavljata osrednji del doktorske disertacije. Četrto poglavje je razdeljeno na tri glavne razdelke. V prvem razdelku formuliramo osnovni problem za eno prehodno in cenovno matriko, ki ga nato posplošimo na poljubno končno mnogo matrik. V drugem razdelku je predstavljena rešitev problema z uporabo metode Monte Carlo. Najprej obravnavamo osnovni problem, nato pa razvijemo tri algoritme za reševanje posplošenega problema. V tretjem razdelku bralca seznanimo še z algoritmom PDV. V petem poglavju eksperimentalno ovrednotimo metode, ki smo jih predstavili v četrtem. V ta namen najprej opišemo postopek za generiranje testne množice podatkov. Posebej predstavimo še postopek za generiranje delno umetnih podatkov, ki opisujejo sistem izmenjave in uporabe koles v Ljubljani. Poglavje zaključimo z rezultati testiranja in njihovo analizo.

Zadnje, šesto poglavje podaja opis opravljenega dela skupaj z ovrednotenjem rezultatov. Izpostavimo še predloge za nadaljnje delo.





---

# Markovske verige v diskretnem času

Recimo, da opazujemo poljuben sistem stanj v diskretnem času  $n = 0, 1, 2, \dots$ . Naj  $X_n$  označuje stanje, v katerem se sistem nahaja v času  $n$ . Predpostavimo, da smo trenutno v času  $n - 1$ , kar pomeni, da smo že obravnavali stanja  $X_0, X_1, \dots, X_{n-1}$ . Ali lahko uporabimo verjetnost za napoved stanja  $X_n$  v času  $n$ ? V splošnem je stanje  $X_n$  odvisno od celotne zgodovine sistema  $X_0, X_1, \dots, X_{n-1}$ . Situacija se precej poenostavi, če je stanje  $X_n$  pri dani celotni zgodovini  $X_0, X_1, \dots, X_{n-1}$  odvisno le od trenutnega stanja  $X_{n-1}$ . To pomeni, da lahko za napoved stanja  $X_n$  pozabimo vse o stanjih  $X_0, X_1, \dots, X_{n-2}$ , pomembna je le informacija o stanju  $X_{n-1}$ . Sistemu, ki zadošča tej lastnosti, pravimo, da zadošča *markovski lastnosti* (Andrey Markov, 1900).

V tem poglavju bomo podali osnovne definicije in lastnosti diskretnih markovskih verig. Pri tem večinoma sledimo [24] in [32]. Na tem mestu bralca opozorimo, da se bomo omejili na končne množice stanj, čeprav nekateri rezultati v nadaljevanju veljajo tudi za neskončno števne množice stanj. Bolj poglobljeno analizo lahko bralec najde v [4, 28].

## 2.1 Osnovne lastnosti markovskih verig

Naj množica  $T = \{0, 1, 2, \dots\}$  označuje *diskreten čas* in naj bo  $S$  končna  $N$ -elementna množica, ki jo poimenujemo *množica stanj*, njene elemente pa *stanja*.

*Slučajni proces* je vsako zaporedje  $\{X_n\}_{n \in T}$  slučajnih spremenljivk  $X_n, n \in T$ , katerih zaloga vrednosti leži v  $S$ . Očitno morajo biti vse slučajne spremenljivke diskretne. Zdaj lahko formalno definiramo markovsko verigo v diskretnem času.

**Definicija 2.1.1.** Slučajni proces  $\{X_n\}_{n \in T}$  z diskretno zalogo vrednosti  $S$  je *diskretna markovska veriga*, če zadošča naslednji *markovski lastnosti*

$$P(X_n = s_n \mid X_0 = s_0, \dots, X_{n-1} = s_{n-1}) = P(X_n = s_n \mid X_{n-1} = s_{n-1}),$$

za vsak  $n \in T$  in vsako končno zaporedje  $s_0, s_1, \dots, s_n$  stanj iz  $S$ .

**Opomba 2.1.2.** Poleg diskretnih markovskih verig poznamo tudi zvezne, ki pa ne bodo predmet naših obravnav. Zato bomo z markovsko verigo vedno mislili na diskretno.

Kot smo že v uvodu poglavja omenili, nam markovska lastnost pove, da novo stanje markovske verige ni odvisno od celotne zgodovine slučajnega procesa, ampak samo od trenutnega stanja.

**Definicija 2.1.3.** *Prehodna verjetnost*  $p_{ij}(n)$  iz poljubnega stanja  $s_i \in S$  v poljubno stanje  $s_j \in S$  v času  $n$  je pogojna verjetnost

$$p_{ij}(n) = P(X_n = s_j | X_{n-1} = s_i).$$

Markovska veriga je *homogena*, če so prehodne verjetnosti neodvisne od časa  $n$ :

$$p_{ij}(n) = P(X_n = s_j | X_{n-1} = s_i) = P(X_1 = s_j | X_0 = s_i), \quad n \in T. \quad (2.1)$$

V nasprotnem primeru rečemo, da je veriga *nehomogena*.

V nadaljevanju se bomo osredotočili na homogene markovske verige. V tem primeru poenostavimo oznako  $p_{ij}(n)$  in pišemo  $p_{ij}$ . Opazimo, da imamo  $N^2$  prehodnih verjetnosti  $p_{ij}$  na *prvem koraku*, ki jih lahko zložimo v matriko  $\mathbf{P}$  dimenzije  $N \times N$ :

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{bmatrix}.$$

Matriki  $\mathbf{P}$  rečemo *prehodna matrika*. Vsaka prehodna matrika  $\mathbf{P}$ , ki pripada homogeni markovski verigi, zadošča pogojema:

$$0 \leq p_{ij} \leq 1, 1 \leq i, j \leq N, \quad (2.2)$$

$$\sum_{j=1}^N p_{ij} = 1, 1 \leq i \leq N. \quad (2.3)$$

Matriki, ki zadošča pogojema (2.2) in (2.3), v linearni algebri rečemo tudi *stohastična matrika*.

Na modelu vremenske napovedi, ki je primer diskretne markovske verige, si pogledjmo, kako iz podanih pogojnih prehodnih verjetnosti zgradimo prehodno matriko markovske verige.

**Primer 2.1.4.** Radi bi napovedali, kakšno bo jutri vreme v Ljubljani. Na voljo imamo tri možne vremenske napovedi: sončno, oblačno ali deževno. Predpostavimo, da je jutrišnja napoved vremena odvisna samo od današnjega vremena. Poleg tega naj velja naslednje: če je danes sončno, bo jutri oblačno z verjetnostjo 0.35 in deževno z verjetnostjo 0.25; če je danes oblačno, bo jutri sončno z verjetnostjo 0.25

## 2.1 Osnovne lastnosti markovskih verig

in deževno z verjetnostjo 0.45; če je danes deževno, bo jutri sončno z verjetnostjo 0.55 in oblačno z verjetnostjo 0.35.

Naj bo  $X_n$  slučajna spremenljivka, ki opisuje vreme v Ljubljani na dan  $n$ , definirana kot:

$$X_n = \begin{cases} 1, & \text{če je sončno na dan } n, \\ 2, & \text{če je oblačno na dan } n, \\ 3, & \text{če je deževno na dan } n. \end{cases}$$

Potem je  $\{X_n\}_{n \in T}$  diskretna markovska veriga z množico stanj  $\{1, 2, 3\}$ . Iz danih podatkov razberemo, da je  $p_{12} = 0.35$  in  $p_{13} = 0.25$ . Iz enačbe (2.3) dobimo, da je  $p_{11} = 0.40$ . Podobno dobimo še preostale verjetnosti, ki jih nato zložimo v prehodno matriko:

$$\mathbf{P} = \begin{bmatrix} 0.40 & 0.35 & 0.25 \\ 0.25 & 0.30 & 0.45 \\ 0.55 & 0.35 & 0.10 \end{bmatrix}.$$

Informacije o prehodnih verjetnostih in prehodih med stanji lahko grafično predstavimo z usmerjenim grafom. *Usmerjen graf* sestavlja množica *vozlišč* (v našem primeru je to množica stanj  $S$ ) in množica *usmerjenih povezav*, kjer je usmerjena povezava definirana kot urejen par vozlišč  $(u, v)$ . Usmerjeno povezavo oblike  $(u, u)$  imenujemo *zanka*. *Usmerjen sprehod* dolžine  $k$  z začetkom v vozlišču  $v_0$  in koncem v vozlišču  $v_k$  je zaporedje  $k$  usmerjenih povezav oblike  $(v_0, v_1)(v_1, v_2) \cdots (v_{k-1}, v_k)$ . Usmerjen graf je *krepko povezan*, če za poljubni vozlišči  $u$  in  $v$  obstaja usmerjen sprehod z začetkom v  $u$  in koncem v  $v$ . Krepko povezanost lahko vidimo kot relacijo na usmerjenem grafu. Ni se težko prepričati, da je slednja relacija ekvivalenčna: njene ekvivalenčne razrede imenujemo *krepko povezane komponente*.

Usmerjen graf predstavimo z *matriko sosednosti*, kjer je na  $(i, j)$ -tem mestu v matriki vrednost 1, če obstaja usmerjena povezava iz vozlišča  $i$  v vozlišče  $j$ , sicer zavzame vrednost 0. Očitno je, kako iz prehodne matrike markovske verige dobimo matriko sosednosti: če je  $p_{ij} > 0$ , potem obstaja usmerjena povezava iz stanja  $s_i$  v stanje  $s_j$ , zato na  $(i, j)$ -tem mestu v matriki sosednosti zapišemo 1, sicer povezava ne obstaja in je  $(i, j)$ -ti element enak 0.

**Primer 2.1.5.** Poglejmo si markovsko verigo z množico stanj  $S = \{1, 2, 3, 4, 5, 6\}$  in prehodno matriko

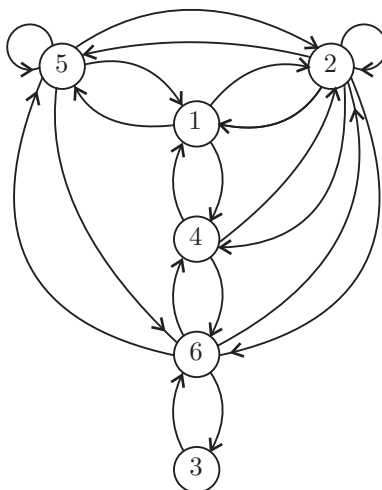
$$\mathbf{P} = \begin{bmatrix} 0 & 0.43 & 0 & 0.02 & 0.55 & 0 \\ 0.27 & 0.19 & 0 & 0.22 & 0.21 & 0.11 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0.19 & 0.77 & 0 & 0 & 0 & 0.04 \\ 0.22 & 0.04 & 0 & 0 & 0.08 & 0.66 \\ 0 & 0.35 & 0.15 & 0.48 & 0.02 & 0 \end{bmatrix},$$

Potem je

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

njena pripadajoča matrika sosednosti. Usmerjen graf (glej sliko 2.1), ki pripada tej matriki sosednosti, sestavlja množica vozlišč  $\{1, 2, 3, 4, 5, 6\}$  in množica usmerjenih povezav

$$\{(1, 2), (1, 4), (1, 5), (2, 1), (2, 2), (2, 4), (2, 5), (2, 6), (3, 6), (4, 1), (4, 2), (4, 6), (5, 1), (5, 2), (5, 5), (5, 6), (6, 2), (6, 3), (6, 4), (6, 5)\}.$$



Slika 2.1: Usmerjen graf, ki pripada markovski verigi v primeru 2.1.5.

### 2.1.1 Prehodna porazdelitev in začetna porazdelitev

Naj bo  $\{X_n\}_{n \in T}$  homogena diskretna markovska veriga z zalogo vrednosti v  $N$ -elementni množici  $S$ . Poglejmo si še eno pomembno karakteristiko markovske verige  $\{X_n\}_{n \in T}$  in sicer začetno porazdelitev, ki nam opiše začetek markovske verige. *Začetno porazdelitev* predstavimo z vektorjem

$$\begin{aligned} \pi^{(0)} &= [\pi_1^{(0)}, \pi_2^{(0)}, \dots, \pi_N^{(0)}] \\ &= [P(X_0 = s_1), P(X_0 = s_2), \dots, P(X_0 = s_N)]. \end{aligned}$$

Podobno za poljuben  $n \in T$  predstavimo *prehodno porazdelitev* na  $n$ -tem koraku z vektorjem

$$\begin{aligned} \pi^{(n)} &= [\pi_1^{(n)}, \pi_2^{(n)}, \dots, \pi_N^{(n)}] \\ &= [P(X_n = s_1), P(X_n = s_2), \dots, P(X_n = s_N)]. \end{aligned}$$

## 2.1 Osnovne lastnosti markovskih verig

Verjetnostim  $P(X_n = s_i)$ ,  $i = 1, 2, \dots, N$ , rečemo *brezpogojne verjetnosti stanj* na  $n$ -tem koraku. Če poznamo začetno porazdelitev  $\pi^{(0)}$  in prehodno matriko  $\mathbf{P}$ , lahko izračunamo poljubno prehodno porazdelitev  $\pi^{(n)}$  markovske verige. Včasih takšni porazdelitvi rečemo tudi *inducirana porazdelitev* (na  $n$ -tem koraku) markovske verige s prehodno matriko  $\mathbf{P}$  in začetno porazdelitvijo  $\pi^{(0)}$ .

Zanimajo nas torej brezpogojne verjetnosti stanj  $\pi_j^{(n)} = P(X_n = s_j)$  za vsak  $s_j \in S$  in  $n \in T$ . V ta namen zapišimo  $P(X_n = s_j)$  kot vsoto začetnih porazdelitev verjetnosti in pogojnih verjetnosti:

$$\begin{aligned} P(X_n = s_j) &= \sum_{i=1}^N P(X_n = s_j | X_0 = s_i) P(X_0 = s_i) \\ &= \sum_{i=1}^N \pi_i^{(0)} P(X_n = s_j | X_0 = s_i). \end{aligned} \quad (2.4)$$

Opazimo, da je dovolj poznati pogojno verjetnost  $P(X_n = s_j | X_0 = s_i)$ , ki jo imenujemo *prehodna verjetnost na  $n$ -tem koraku* markovske verige. Zaradi poenostavitve zapisa pišimo

$$p_{i,j}^{(n)} = P(X_n = s_j | X_0 = s_i).$$

Prehodne verjetnosti na  $n$ -tem koraku  $p_{ij}^{(n)}$  lahko zapišemo v *prehodno matriko  $n$ -tega koraka*  $\mathbf{P}^{(n)}$  dimenzije  $N \times N$ :

$$\mathbf{P}^{(n)} = \begin{bmatrix} p_{11}^{(n)} & p_{12}^{(n)} & \cdots & p_{1N}^{(n)} \\ p_{21}^{(n)} & p_{22}^{(n)} & \cdots & p_{2N}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1}^{(n)} & p_{N2}^{(n)} & \cdots & p_{NN}^{(n)} \end{bmatrix}.$$

Enakost (2.4) lahko zapišemo v matrični obliki kot produkt začetne porazdelitve verjetnosti in prehodne matrike  $n$ -tega koraka markovske verige:

$$\pi^{(n)} = \pi^{(0)} \cdot \mathbf{P}^{(n)}. \quad (2.5)$$

Obravnavajmo najprej posebna primera  $\mathbf{P}^{(0)}$  in  $\mathbf{P}^{(1)}$ . Za  $n = 0$  očitno velja

$$p_{ij}^{(0)} = P(X_0 = s_j | X_0 = s_i) = \begin{cases} 1, & \text{če } s_i = s_j, \\ 0, & \text{če } s_i \neq s_j. \end{cases} \quad (2.6)$$

Torej je  $\mathbf{P}^{(0)} = I$  identična matrika dimenzije  $N \times N$ . V primeru, ko je  $n = 1$ , pa zaradi (2.1) velja

$$p_{ij}^{(1)} = P(X_1 = s_j | X_0 = s_i) = p_{ij}, \quad (2.7)$$

od koder sledi, da je  $\mathbf{P}^{(1)}$  enaka kar dani začetni prehodni matriki  $\mathbf{P}$ . Uporabimo matematično indukcijo in pokažimo, da za poljuben  $n \in T$  velja zveza

$$\mathbf{P}^{(n)} = \mathbf{P}^n. \quad (2.8)$$

Bazo indukcije za  $n = 0$  in  $n = 1$  smo pravkar pokazali. Predpostavimo, da zveza velja za vse potence  $k \leq n$ . Računajmo:

$$\begin{aligned}
 p_{ij}^{(n)} &= P(X_n = s_j | X_0 = s_i) \\
 &= \sum_{k=1}^N P(X_n = s_j | X_{n-1} = s_k, X_0 = s_i) P(X_{n-1} = s_k | X_0 = s_i) \\
 &= \sum_{k=1}^N p_{ik}^{(n-1)} P(X_n = s_j | X_{n-1} = s_k, X_0 = s_i) \\
 &= \sum_{k=1}^N p_{ik}^{(n-1)} \underbrace{P(X_n = s_j | X_{n-1} = s_k)}_{\text{markovska lastnost}} \\
 &= \sum_{k=1}^N p_{ik}^{(n-1)} \underbrace{P(X_1 = s_j | X_0 = s_k)}_{\text{homogena veriga}} \\
 &= \sum_{k=1}^N p_{ik}^{(n-1)} p_{kj}. \tag{2.9}
 \end{aligned}$$

Dobljeno enakost (2.9) prepišimo v matrično obliko:

$$\mathbf{P}^{(n)} = \mathbf{P}^{(n-1)} \cdot \mathbf{P}.$$

Upoštevajmo sedaj indukcijsko predpostavko  $\mathbf{P}^{(n-1)} = \mathbf{P}^{n-1}$  in enakost (2.8) sledi.

Poglejmo si dve posledici enakosti (2.8). Če upoštevamo zadnjo enakost v (2.5), potem lahko prehodno porazdelitev  $\pi^{(n)}$  izrazimo s prehodno matriko  $\mathbf{P}$  in dano začetno porazdelitvijo  $\pi^{(0)}$ :

$$\pi^{(n)} = \pi^{(0)} \mathbf{P}^n.$$

Druga posledica pa nam da naslednjo enakost:

$$P(X_{n+m} = s_j | X_0 = s_0, \dots, X_n = s_i) = P(X_{n+m} = s_j | X_n = s_i) = p_{ij}^{(m)}. \tag{2.10}$$

Naslednja trditev je splošitev enakosti (2.4).

**Trditev 2.1.6.** *Prehodne verjetnosti  $n$ -tega koraka zadoščajo Chapman-Kolmogorovim enačbam:*

$$p_{ij}^{(n+m)} = \sum_{k=1}^N p_{ik}^{(n)} p_{kj}^{(m)}. \tag{2.11}$$

**Opomba 2.1.7.** Enačbe lahko zapišemo v matrični obliki:  $\mathbf{P}^{(n+m)} = \mathbf{P}^{(n)} \mathbf{P}^{(m)}$ .

**Dokaz:** Če zapišemo enačbe v matrično obliko kot v (2.1.7), rezultat sledi neposredno iz (2.8). ■

**Primer 2.1.8.** Gospa in gospod Novak čez natanko en teden praznujeta svojo 50. obletnico poroke. Praznovala bi jo rada na dvodnevem oddihu na morju. Na

## 2.2 Klasifikacija stanj in verig

morju bi rada preživela sedmi in osmi dan, pri čemer današnji dan šteje kot prvi dan. Celoten znesek paketa stane 1500 evrov. Razmišljata o zavarovanju potovanja, ki jima v primeru, da sta oba dneva deževna, povrnejo celoten znesek paketa, v vseh preostalih primerih pa se jima na povrne nič. Zavarovanje stane 150 evrov. Predpostavimo, da se vreme na morju spreminja kot v primeru 2.1.4. Ali naj par zavaruje potovanje, če vemo, da je danes sončno vreme na morju?

Naj bo  $R$  vrednost, ki ga par dobi od zavarovalniške družbe. Kot v primeru 2.1.4 naj slučajna spremenljivka  $X_n$  označuje vreme na  $n$ -ti dan. Ker je danes sončno vreme, velja  $X_1 = 1$  in

$$R = \begin{cases} 1500, & \text{če } X_7 = X_8 = 3, \\ 0, & \text{sicer.} \end{cases}$$

Od tod sledi:

$$\begin{aligned} E(R) &= R \cdot P(X_7 = X_8 = 3 | X_1 = 1) \\ &= 1500 \cdot P(X_8 = 3 | X_7 = 3, X_1 = 1) \cdot P(X_7 = 3 | X_1 = 1) \\ &= 1500 \cdot P(X_1 = 3 | X_0 = 3) \cdot P(X_6 = 3 | X_0 = 1) \\ &= 1500 \cdot p_{33} \cdot [\mathbf{P}^6]_{13} \\ &= 1500 \cdot p_{33} \cdot \begin{bmatrix} 0.40 & 0.35 & 0.25 \\ 0.25 & 0.30 & 0.45 \\ 0.55 & 0.35 & 0.10 \end{bmatrix}_{13}^6 \\ &= 1500 \cdot p_{33} \cdot \begin{bmatrix} 0.3913 & 0.3333 & 0.2754 \\ 0.3913 & 0.3333 & 0.2753 \\ 0.3913 & 0.3333 & 0.2754 \end{bmatrix}_{13} \\ &= 1500 \cdot 0.10 \cdot 0.2754 \\ &= 41.31. \end{aligned}$$

Torej bi v povprečju par dobil povrnjenih le 41,31 evrov, kar je manj od 150 evrov zavarovalnine, zato se takšno zavarovanje ne splača. Pozoren bralec bo opazil, da je verjetnost, da bo oba dneva deževalo, pri pogoju, da je danes sončno, enaka 2,7 odstotka.

## 2.2 Klasifikacija stanj in verig

Elemente množice  $S$ , torej stanja homogene diskretne markovske verige  $\{X_n\}_{n \in T}$  lahko pogosto klasificiramo na različne načine, odvisno od fundamentalnih lastnosti sistema. Glede na klasifikacijo stanj je možno določiti določene razrede verig.

### 2.2.1 Nerazcepnost in periodičnost

**Definicija 2.2.1.** Naj bosta  $s_i$  in  $s_j$  poljubni stanja v  $S$ . Stanje  $s_j$  je *dosegljivo* iz stanja  $s_i$ , če velja  $p_{ij}^{(n)} > 0$  za nek  $n \geq 0$ . V tem primeru pišemo  $s_i \rightarrow s_j$ .

**Opomba 2.2.2.** Stanje  $s_i$  je vedno dosegljivo iz stanja  $s_i$ , ker je  $p_{ii}^{(0)} = 1 > 0$ .

Stanje  $s_j$  je torej dosegljivo iz stanja  $s_i$ , če obstaja pozitivna verjetnost, da pridemo iz  $s_i$  v  $s_j$  v končno mnogo korakov. Če lahko pridemo iz  $s_j$  v  $s_i$  v enem koraku, rečemo, da je  $s_j$  *dosegljivo v enem koraku* iz  $s_i$ . Če pa  $s_j$  ni dosegljivo iz  $s_i$ , to med drugim pomeni tudi, da veriga, ki začne v  $s_i$ , nikoli ne obiše  $s_j$ .

Ker lahko prehodni matriki markovske verige priredimo usmerjen graf, lahko dosegljivost stanj interpretiramo v jeziku teorije grafov. Stanje  $s_j$  je dosegljivo iz stanja  $s_i$  natanko tedaj, ko v pripadajočem usmerjenem grafu obstaja (končen) usmerjen sprehod z začetkom v  $s_i$  in koncem v  $s_j$ . V posebnem,  $s_j$  je dosegljivo v enem koraku iz stanja  $s_i$  natanko tedaj, ko obstaja usmerjena povezava  $(s_i, s_j)$ .

**Definicija 2.2.3.** Poljubni stanja  $s_i$  in  $s_j$  v  $S$  sta *povezani* med seboj, če je stanje  $s_i$  dosegljivo iz stanja  $s_j$  in stanje  $s_j$  dosegljivo iz  $s_i$ . Tedaj pišemo  $s_i \leftrightarrow s_j$ .

Povezanost  $\leftrightarrow$  lahko interpretiramo kot relacijo na množici stanj. Ni se težko prepričati, da veljajo naslednje tri lastnosti:

- refleksivnost:  $s_i \leftrightarrow s_i$ ,
- simetričnost:  $s_i \leftrightarrow s_j \Rightarrow s_j \leftrightarrow s_i$ ,
- tranzitivnost:  $s_i \leftrightarrow s_j$  in  $s_j \leftrightarrow s_k \Rightarrow s_i \leftrightarrow s_k$ .

Torej je  $\leftrightarrow$  ekvivalenčna relacija, ki razbije množico  $S$  na disjunktne ekvivalenčne razrede, ki jih imenujemo *povezanostni razredi*. Na tem mestu bralca opozorimo, da povezanostni razredi sovpadajo s krepko povezanimi komponentami na pripadajočem usmerjenem grafu v smislu, da vozlišča v krepko povezani komponenti predstavljajo natanko povezanostni razred relacije  $\leftrightarrow$  na  $S$ .

**Primer 2.2.4.** Na sliki 2.2 je prikazan usmerjen graf, ki pripada neki markovski verigi z množico stanj  $S = \{1, 2, 3, 4, 5, 6\}$ . Vidimo, da graf oziroma markovska veriga razpade na štiri povezanostne razrede:

$$\{1\}, \quad \{2, 3\}, \quad \{4, 5\}, \quad \{6\}.$$

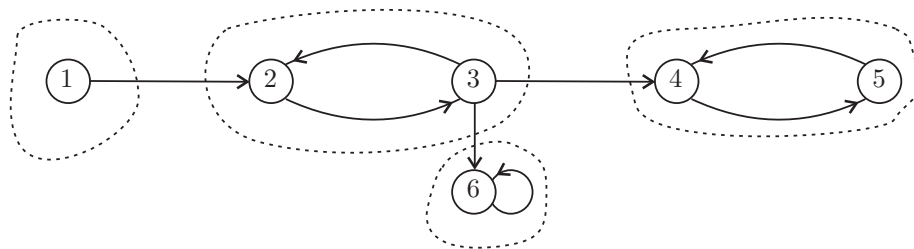
Opazimo še nekaj. Prva dva razreda se razlikujeta od zadnjih dveh. Če veriga pride recimo v razred  $\{4, 5\}$ , potem nikoli več ne zapusti tega razreda. To ne velja na primer za razred  $\{2, 3\}$ .

V luči zadnjega primera uvedemo pojem zaprte množice.

**Definicija 2.2.5.** Neprazna podmnožica stanj  $Z \subseteq S$  je *zaprta*, če nobeno stanje, ki leži izven  $Z$ , ni dosegljivo iz nobenega stanja v  $Z$ .



## 2.2 Klasifikacija stanj in verig



Slika 2.2: Usmerjen graf.

Če je  $Z$  zaprta množica in  $s_i \in Z$ , medtem ko  $s_j \notin Z$ , potem velja  $p_{ij}^{(n)} = 0$  za vse  $n \in T$ . Drugače rečeno, ko enkrat vstopimo v zaprto množico, je ne moremo več zapustiti.

Vsaka markovska veriga vsebuje vsaj eno zaprto množico, in sicer množico vseh stanj  $S$ . Zaprti povezanostni razredi so še posebej pomembni, ker razbijejo verigo na manjše, bolj obvladljive dele.

**Definicija 2.2.6.** Če markovska veriga ne vsebuje nobene prave zaprte podmnožice razen množice vseh stanj  $S$ , potem je veriga nerazcepna. V nasprotnem je razcepna.

Naslednja trditev podaja odnos med dosegljivostjo, povezanostjo, razcepnostjo in krepko povezanostjo. Dokaz prepuščamo bralcu.

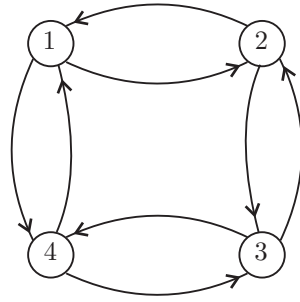
**Trditev 2.2.7.** Za markovsko verigo so ekvivalentne naslednje trditve:

- (i) Veriga je nerazcepna.
- (ii) Veriga vsebuje en sam povezanostni razred.
- (iii) Vsako stanje verige je dosegljivo iz vsakega drugega stanja verige.
- (iv) Poljubni dve stanji verige sta povezani.
- (v) Graf, prirejen markovski verigi, je krepko povezan.

**Opomba 2.2.8.** Iz računskega vidika je testiranje krepke povezanosti usmerjenega grafa enostavno: z uporabo algoritma za pregled grafa v globino lahko v linearnem času v velikosti vhodnih podatkov preverimo, ali je dani usmerjen graf krepko povezan.

**Primer 2.2.9.** Oglejmo si usmerjen graf na sliki 2.3, ki pripada neki markovski verigi z množico stanj  $S = \{1, 2, 3, 4\}$ . Očitno je ta markovska veriga nerazcepna. Vendar pa lahko v tem primeru obnašanje verige nadalje poenostavimo na podlagi naslednjega premisleka: če se veriga v nekem času nahaja v množici  $C_1 = \{1, 3\}$ , potem se bo v naslednjem koraku nahajala v množici  $C_2 = \{2, 4\}$  in obratno. Veriga torej alternira med dvema množicama: če je  $X_0 \in C_1$ , potem vemo, da velja

$$X_1 \in C_2, X_2 \in C_1, X_3 \in C_2, X_4 \in C_1, \dots$$



Slika 2.3: Usmerjen graf.

Naslednji pomemben koncept, ki si ga bomo ogledali, je (a)periodičnost. V nadaljevanju bomo največji skupni delitelj podmnožice  $X \subseteq \mathbb{N}$  naravnih števil označili z  $D(X)$ .

**Definicija 2.2.10.** Perioda  $d(s_i)$  stanja  $s_i \in S$  je definirana kot

$$d(s_i) = D(\{n \geq 1 \mid p_{ii}^{(n)} > 0\}),$$

pri čemer je  $d(s_i) = \infty$ , če je  $p_{ii}^{(n)} = 0$  za vse  $n \geq 1$ . Če je  $d(s_i) = 1$ , rečemo, da je stanje  $s_i$  aperiodično.

Perioda stanja  $s_i$  je torej največji skupni delitelj množice časov, ko se veriga s pozitivno verjetnostjo vrne v  $s_i$ , pri pogoju, da smo začeli v stanju  $X_0 = s_i$ .

**Trditev 2.2.11.** Stanje  $s_i$  je aperiodično natanko tedaj, ko obstaja  $n_0 \in \mathbb{N}$ , da je  $p_{ii}^{(n)} > 0$  za vse  $n > n_0$ .

**Dokaz:** Če obstaja  $n_0 \in \mathbb{N}$ , da je  $p_{ii}^{(n)} > 0$  za vse  $n > n_0$ , potem je  $s_i$  očitno aperiodično stanje.

Za dokaz obrata pokažimo najprej naslednjo pomožno trditev: za poljubno fiksno naravno število  $k$  obstaja naravno število  $n_0$ , da velja

$$\{n_0, n_0 + 1, n_0 + 2, \dots\} \subset \{n_1 k + n_2(k + 1) \mid n_1, n_2 \in \mathbb{N} \cup \{0\}\}.$$

Res. Fiksirajmo naravno število  $k$ . Če je  $n \geq k^2$ , potem obstajata nenegativni celi števila  $m, d \geq 0$ , da velja  $n - k^2 = mk + d$  in  $d < k$ . Torej lahko  $n$  zapišemo v obliki  $n = (k - d + m)k + d(k + 1)$ , zato je

$$n \in \{n_1 k + n_2(k + 1) \mid n_1, n_2 \in \mathbb{N} \cup \{0\}\}.$$

Z drugimi besedami,  $n_0 = k^2$  je iskano število in pomožna trditev je s tem dokazana. Naj bo  $s_i$  aperiodično stanje in naj bo  $J(i) = \{n \geq 1 \mid p_{ii}^{(n)} > 0\}$ , kjer je  $D(J(i)) = 1$ . Očitno je  $J(i)$  zaprta za seštevanje: ker velja

$$p_{ii}^{(n+m)} \geq p_{ii}^{(n)} p_{ii}^{(m)},$$

je  $n+m \in J(i)$ , če je  $n, m \in J(i)$ . Po pomožni trditvi je torej dovolj pokazati, da  $J(i)$  vsebuje dve zaporedni naravni števili  $k$  in  $k + 1$ . Recimo, da  $J(i)$  ne premore dveh

## 2.2 Klasifikacija stanj in verig

zaporednih naravnih števil. Naj  $t \geq 2$  označuje minimalno razdaljo med elementi v  $J(i)$ . Ker je  $D(J(i)) = 1$ , mora  $J(i)$  vsebovati naravno število  $n$ , oblike

$$n = mt + d, \tag{2.12}$$

za neko nenegativno celo število  $m$  in neko naravno število  $d \in \{1, 2, \dots, t - 1\}$ . Izberimo sedaj dve števili v  $J(i)$  na minimalni razdalji  $t$ , recimo  $n_1, n_1 + t \in J(i)$ . Ker je  $J(i)$  zaprta za seštevanje, je  $a(n_1 + t) \in J(i)$  in  $n + bn_1 \in J(i)$  za poljubni naravni števili  $a, b \in \mathbb{N}$ . Pokažimo, da lahko izberemo takšni naravni števili  $a$  in  $b$ , da je razdalja med  $a(n_1 + t)$  in  $n + bn_1$  manjša kot  $t$ . Upoštevajmo (2.12), pa imamo

$$a(n_1 + t) - n - bn_1 = (a - b)n_1 + (a - m)t - d$$

in za  $a = b = m + 1$  dobimo

$$a(n_1 + t) - n - bn_1 = t - d < t.$$

To je v protislovju s tem, da je minimalna razdalja med elementi v  $J(i)$  enaka  $t$ . Zaključimo, da  $J(i)$  premore dve zaporedni naravni števili. S tem je trditev dokazana. ■

**Izrek 2.2.12.** *Naj bo  $\{X_n\}_{n \in T}$  nerazcepna in naj bo  $s_k$  aperiodično stanje. Potem za poljubi stanji  $s_i, s_j \in S$  obstaja  $n_0 \in T$ , da je  $p_{ij}^{(n)} > 0$  za vse  $n > n_0$ .*

**Dokaz:** Fiksirajmo  $s_i, s_j \in S$ . Ker je  $\{X_n\}_{n \in T}$  nerazcepna, obstajata  $r, s \geq 0$ , da velja  $p_{ik}^{(r)}, p_{kj}^{(s)} > 0$ . Ker je  $s_k$  aperiodično stanje, obstaja  $n'_0 \in T$ , da je  $p_{kk}^{(n')} > 0$  za vse  $n' > n'_0$ . Naj bo  $n_0 = r + n'_0 + s$ . Potem je

$$p_{ij}^{(r+n'+s)} \geq p_{ik}^{(r)} p_{kk}^{(n')} p_{kj}^{(s)} > 0$$

za vse  $n = r + n' + s > n_0$ . ■

V splošnem imajo različna stanja lahko različno periodo. Toda izrek 2.2.12 med drugim pove, da se to lahko zgodi samo v primeru, ko je veriga razcepna.

**Posledica 2.2.13.** *Vsa stanja v povezanostnem razredu imajo enako periodo.*

**Definicija 2.2.14.** Markovska veriga je aperiodična, če so vsa stanja aperiodična. V nasprotnem je periodična.

V posebnem, če je markovska veriga nerazcepna, potem imajo vsa stanja enako periodo  $d$ , in lahko rečemo, da ima markovska veriga periodo  $d$ . Zadnja posledica porodi naslednji zadostni pogoj za aperiodičnost markovske verige.

**Posledica 2.2.15.** *Nerazcepna markovska veriga  $\{X_n\}_{n \in T}$  je aperiodična, če obstaja stanje  $s_i \in S$ , da velja  $p_{ii} > 0$ .*

**Opomba 2.2.16.** Pogoj je ekvivalenten temu, da v pripadajočem usmerjenem grafu markovske verige obstaja zanka.

**Primer 2.2.17.** Naj bo  $\{X_n\}_{n \in T}$  markovska veriga z množico stanj  $S = \{1, 2, 3\}$  in pripadajočo prehodno matriko

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0.5379 & 0 & 0.4621 \\ 0 & 1 & 0 \end{bmatrix}.$$

Očitno je dana markovska veriga nerazcepna, zato je dovolj poračunati periodo poljubnega stanja, na primer, stanja  $s_1$ :

$$d(s_1) = D(\{2, 4, \dots\}) = 2, \text{ ker je } p_{11}^{(1)} = 0, p_{11}^{(2)} > 0, p_{11}^{(3)} = 0, p_{11}^{(4)} > 0, \dots$$

Markovska veriga je torej periodična s periodo 2.

Ko združimo nerazcepnost in aperiodičnost skupaj s standardno predpostavko, da je množica stanj končna, dobimo naslednji pomemben rezultat.

**Izrek 2.2.18.** *Naj bo  $\{X_n\}_{n \in T}$  markovska veriga z zalogo vrednosti v končni  $N$ -elementni množici  $S$  in naj bo  $\mathbf{P}$  njena pripadajoča prehodna matrika. Potem je markovska veriga  $\{X_n\}_{n \in T}$  nerazcepna in aperiodična natanko tedaj, ko obstaja naravno število  $n$ , da ima matrika  $\mathbf{P}^n$  vse elemente neničelne.*

**Dokaz:** Če obstaja naravno število  $n$ , da ima matrika  $\mathbf{P}^n$  vse elemente neničelne, potem je očitno, da je veriga nerazcepna. Poleg tega ima  $\mathbf{P}^l$  vse elemente neničelne za vsak  $l \geq n$ , zato je po trditvi 2.2.11 veriga tudi aperiodična.

Pokažimo še obrat. Predpostavimo, da je  $\{X_n\}_{n \in T}$  nerazcepna in aperiodična. Fiksirajmo stanji  $s_i, s_j \in S$ . Po izreku 2.2.12 obstaja  $m_{ij} \in T$ , da je  $p_{ij}^{(n')} > 0$  za vse  $n' > m_{ij}$ . Vzemimo

$$n = \max\{m_{ij} \mid i, j = 1, 2, \dots, N\} + 1$$

in rezultat sledi. ■

Za učinkovito računanje periode nerazcepne markovske verige oziroma za učinkovito testiranje aperiodičnosti problem zopet interpretiramo v jeziku teorije grafov: perioda je enaka največjemu skupnemu delitelju množice dolžin vseh usmerjenih ciklov v pripadajočem usmerjenem grafu markovske verige. Če združimo algoritem za pregled grafa v širino skupaj z algoritmom za računanje največjega skupnega delitelja, se izkaže, da lahko v linearnem času v velikosti vhodnih podatkov izračunamo periodo.

## 2.2.2 Povrnjivost in minljivost

Eno izmed temeljnih vprašanj, ki nas ponavadi zanima, je, ali je vrnitev v določeno stanje gotov dogodek (povrnjivost) ali ne (minljivost), in v primeru, ko je vrnitev zanesljiva, ali je povprečni čas vrnitve končen. Naj bo  $\{X_n\}_{n \in T}$  markovska veriga s pripadajočo prehodno matriko  $\mathbf{P}$ .

**Definicija 2.2.19.** Slučajno spremenljivko

$$T_i = \min\{n \geq 1 \mid X_n = s_i\},$$

pri čemer je  $T_i = \infty$ , če  $X_n \neq s_i$  za vse  $n \geq 1$ , imenujemo čas prvega prihoda v stanje  $s_i$ .

**Definicija 2.2.20.** Stanje  $s_i \in S$  je povrnljivo, če je vrnitev v  $s_i$  gotov dogodek:

$$P(T_i < \infty \mid X_0 = s_i) = 1;$$

sicer je stanje  $s_i$  minljivo.

Naslednji izrek karakterizira povrnljiva (minljiva) stanja v smislu prehodnih verjetnosti na  $n$ -tem koraku.

**Izrek 2.2.21.** (i) Stanje  $s_i$  je povrnljivo natanko tedaj, ko velja

$$\sum_{n=0}^{\infty} p_{ii}^{(n)} = \infty.$$

(ii) Stanje  $s_i$  je minljivo natanko tedaj, ko velja

$$\sum_{n=0}^{\infty} p_{ii}^{(n)} < \infty.$$

**Dokaz:** Naj

$$I_{\{X_n=s_j\}} = \begin{cases} 1, & \text{če } X_n = s_j, \\ 0, & \text{sicer} \end{cases}$$

označuje indikatorsko funkcijo slučajne spremenljivke  $X_n$ . Opozorimo, da je  $I_{\{X_n=s_j\}}$  slučajna spremenljivka. Nadalje, naj slučajna spremenljivka  $N_i$  označuje število obiskov stanja  $s_i$ , ki jo lahko zapišemo z indikatorskimi funkcijami kot

$$N_i = \sum_{n=0}^{\infty} I_{\{X_n=s_i\}}.$$

Potem je povprečno število obiskov stanja  $s_i$  pri pogoju, da smo začeli v  $s_i$ , enako

$$\begin{aligned} E(N_i \mid X_0 = s_i) &= E\left(\sum_{n=0}^{\infty} I_{\{X_n=s_j\}} \mid X_0 = s_i\right) = \sum_{n=0}^{\infty} E(I_{\{X_n=s_j\}} \mid X_0 = s_i) \\ &= \sum_{n=0}^{\infty} P(X_n = s_i \mid X_0 = s_i) = \sum_{n=0}^{\infty} p_{ii}^{(n)}. \end{aligned}$$

Izračunajmo  $E(N_i \mid X_0 = s_i)$  še na drug način. Pišimo  $f_i = P(T_i < \infty \mid X_0 = s_i)$  in pokažimo, da je pogojna porazdelitev spremenljivke  $N_i$  glede na  $X_0 = s_i$  enaka geometrijski porazdelitvi s parametrom  $1 - f_i$ . Res. Opazimo, da je ob vsakem

obisku stanja  $s_i$  verjetnost, da se ne bomo nikoli več vrnili v  $s_i$ , enaka  $1 - f_i$ , zato velja

$$P(N_i = n | X_0 = s_i) = f_i^{n-1}(1 - f_i).$$

Spomnimo, da je matematično upanje geometrijske spremenljivke s parametrom  $1 - f_i$  enako  $\frac{1}{1-f_i}$ , in zato

$$E(N_i | X_0 = s_i) = \frac{1}{1 - f_i}.$$

Torej velja

$$\sum_{n=0}^{\infty} p_{ii}^{(n)} = \frac{1}{1 - f_i}$$

in rezultat sledi. ■

Z uporabo zadnjega izreka lahko pokažemo, da so vsa stanja v povezanostnem razredu povrnjljiva (minljiva), brž ko je eno stanje znotraj razreda povrnjljivo (minljivo).

**Izrek 2.2.22.** *Stanja v povezanostnem razredu so bodisi vsa povrnjljiva bodisi vsa minljiva.*

**Dokaz:** Vzemimo poljubi stanji  $s_i$  in  $s_j$  v povezanostnem razredu in predpostavimo, da je  $s_i$  minljivo. Potem obstajata  $n, m \geq 0$ , da velja  $p_{ij}^{(n)} > 0$  in  $p_{ji}^{(m)} > 0$ . Poleg tega za vsak  $r \geq 0$  velja

$$p_{ii}^{(n+r+m)} \geq p_{ij}^{(n)} p_{jj}^{(r)} p_{ji}^{(m)}$$

in zato

$$\sum_{r=0}^{\infty} p_{jj}^{(r)} \leq \frac{1}{p_{ij}^{(n)} p_{ji}^{(m)}} \sum_{r=0}^{\infty} p_{ii}^{(n+r+m)} < \infty$$

po izreku 2.2.21. Torej je  $s_j$  prav tako minljivo po izreku 2.2.21. ■

Pri povrnjljivih stanjih se lahko vprašamo še, kako se obnaša povprečni čas prve vrnitve. Glede na to ločimo dve vrsti povrnjljivih stanj.

**Definicija 2.2.23.** Povrnjljivo stanje  $s_i$  je neničelno, če je povprečni čas prve vrnitve v  $s_i$  končen:

$$E(T_i | X_0 = s_i) < \infty;$$

sicer je ničelno.

Če je množica stanj končna, velja naslednja pomembna trditev.

**Trditev 2.2.24.** *Naj bo  $\{X_n\}_{n \in T}$  nerazcepna markovska veriga s končno  $N$ -elementno množico stanj  $S$  in naj bo  $\mathbf{P}$  pripadajoča prehodna matrika. Potem za poljubni stanji  $s_i$  in  $s_j$  velja*

- (i)  $P(T_j < \infty | X_0 = s_i) = 1$ ,
- (ii)  $E(T_j | X_0 = s_i) < \infty$ .

## 2.2 Klasifikacija stanj in verig

**Dokaz:** Če pokažemo, da je  $P(T_j = \infty | X_0 = s_i) = 0$ , bo točka (i) dokazana. Ker je veriga nerazcepna, za poljubni stanji  $s_i$  in  $s_j$  obstajata pozitivni števili  $r_{ij}$  in  $\epsilon_{ij}$ , da velja

$$\mathbf{P}_{ij}^{r_{ij}} = \epsilon_{ij}.$$

Naj bo  $r = \max\{r_{ij} | i, j = 1, 2, \dots, N\}$  in naj bo  $\epsilon = \min\{\epsilon_{ij} | i, j = 1, 2, \dots, N\}$ . Torej je za poljubno vrednost spremenljivke  $X_k$  verjetnost, da obiščemo stanje  $s_j$  med časom  $k$  in  $k + r$  vsaj  $\epsilon$ . Fiksirajmo poljubni stanji  $s_i$  in  $s_j$ . Očitno velja

$$P(T_j > r | X_0 = s_i) \leq P(X_1 \neq s_j, X_2 \neq s_j, \dots, X_r \neq s_j) \leq 1 - \epsilon.$$

Podobno velja

$$\begin{aligned} P(T_j > 2r | X_0 = s_i) &= P(T_j > 2r | T_j > r, X_0 = s_i)P(T_j > r | X_0 = s_i) \\ &\leq P(X_{r+1} \neq s_j, X_{r+2} \neq s_j, \dots, X_{2r} \neq s_j)P(T_j > r | X_0 = s_i) \\ &\leq (1 - \epsilon)^2. \end{aligned}$$

Če iteracijsko nadaljujemo, dobimo za vsak  $l$  neenakost

$$P(T_j > lr | X_0 = s_i) \leq (1 - \epsilon)^l. \quad (2.13)$$

Pošljimo sedaj  $l$  v zadnji neenakosti proti neskončnosti in (i) sledi.

Lastnost (ii) pokažemo z direktnim izračunom:

$$\begin{aligned} E(T_j | X_0 = s_i) &= \sum_{k=0}^{\infty} P(T_j > k | X_0 = s_i) \\ &= \sum_{l=0}^{\infty} \sum_{k=lr}^{(l+1)r-1} P(T_j > k | X_0 = s_i) \\ &\leq \sum_{l=0}^{\infty} \sum_{k=lr}^{(l+1)r-1} P(T_j > lr | X_0 = s_i) = r \sum_{l=0}^{\infty} P(T_j > lr | X_0 = s_i) \\ &\leq r \sum_{l=0}^{\infty} (1 - \epsilon)^l = r \frac{1}{1 - (1 - \epsilon)} = \frac{r}{\epsilon} < \infty, \end{aligned}$$

kjer smo v zadnji vrstici upoštevali neenakost (2.13). ■

**Opomba 2.2.25.** Točka (i) v Trditvi 2.2.24 pove, da je obisk stanja  $s_j$  v končno mnogo korakov gotov dogodek pri pogoju, da smo začeli v stanju  $s_i$ . Točka (ii) v Trditvi 2.2.24 pa pove, da je povprečni čas prvega obiska stanja  $s_j$  končen pri pogoju, da smo začeli v stanju  $s_i$ .

Naslednja rezultata sta neposredni posledici zadnje trditve.

**Posledica 2.2.26.** *Nerazcepna markovska veriga s končno množico stanj ima vsa stanja povrnljivo neničelna.*

**Posledica 2.2.27.** Za nerazcepno markovsko verigo s končno  $N$ -elementno množico stanj  $S$  velja

$$P(T_j < \infty) = 1$$

za vsa stanja  $s_j \in S$ .

**Dokaz:** Iz markovske lastnosti sledi

$$P(T_j < \infty) = \sum_{i=1}^N P(X_0 = s_i)P(T_j < \infty | X_0 = s_i).$$

Toda  $P(T_j < \infty | X_0 = s_i) = 1$  po točki (i) trditve 2.2.24 za vsak  $s_i \in S$  in rezultat sledi. ■

**Opomba 2.2.28.** Obisk stanja  $s_j$  v končno mnogo korakov je gotov dogodek, ne glede na to, v katerem stanju smo začeli.

### 2.3 Stacionarna in limitna porazdelitev

V tem razdelku bomo obravnavali eno izmed centralnih vprašanj v teoriji markovskih procesov: Kaj lahko povemo o markovskih verigah, ko se s časom približujemo neskončnosti? Ali lahko najdemo limitne izreke?

Če je  $\{X_n\}_{n \geq 0}$  poljubna netrivialna markovska veriga, potem bodo vrednosti spremenljivke  $X_n$  kolebale neskončno časa za  $n \rightarrow \infty$ , zato ne moremo pričakovati, da bi obstajala limita za  $X_n$ . Lahko pa pričakujemo, da se bo porazdelitev spremenljivke  $X_n$  sčasoma ustalila. Pomembno vlogo pri tem igra stacionarna porazdelitev.

**Definicija 2.3.1.** Naj bo  $\{X_n\}_{n \geq 0}$  markovska veriga s končno  $N$ -elementno množico stanj  $S$  in naj bo  $\mathbf{P}$  pripadajoča prehodna matrika. Vektor  $\pi = (\pi_1, \pi_2, \dots, \pi_N)$  je stacionarna porazdelitev za markovsko verigo, če velja:

- (i)  $\pi_i \geq 0$  za  $i = 1, 2, \dots, N$ , in  $\sum_{i=1}^N \pi_i = 1$  in
- (ii)  $\pi_j = \sum_{i=1}^N \pi_i p_{ij}$  za  $j = 1, 2, \dots, N$ .

**Opomba 2.3.2.** Lastnost (i) preprosto pomeni, da je  $\pi$  verjetnostna porazdelitev na  $S$ . Iz lastnosti (ii) sledi naslednje: če je začetna porazdelitev  $\pi^{(0)}$  enaka  $\pi$ , potem porazdelitev  $\pi^{(1)}$  verige v času 1 zadošča

$$\pi_j^{(1)} = \sum_{i=1}^N \pi_i^{(0)} p_{ij} = \sum_{i=1}^N \pi_i p_{ij} = \pi_j \text{ za } j = 1, 2, \dots, N.$$

Če induktivno nadaljujemo, vidimo, da za poljuben  $n$  velja  $\pi_j^{(n)} = \pi_j$  za  $j = 1, 2, \dots, N$ . Stacionarno porazdelitev lahko torej interpretiramo kot fiksno točko markovske verige:  $\pi^{(n)} = \pi$  za vsak  $n \geq 0$ .



**Opomba 2.3.3.** Lastnost (ii) lahko v matrični obliki zapišemo kot  $\pi \mathbf{P} = \pi$ . Vidimo, da je stacionarna porazdelitev  $\pi$  natanko levi lastni vektor prehodne matrike  $P$  pri lastni vrednosti 1.

Najprej se posvetimo naslednjima vprašanjema:

- (i) Kdaj sploh obstaja stacionarna porazdelitev?
- (ii) Če obstaja, kako je z njeno enoličnostjo?

Začnimo z obstojem stacionarne porazdelitev. Če je markovska veriga s končno množico stanj nerazcepna, potem vedno obstaja stacionarna porazdelitev. O tem govori naslednji izrek.

**Izrek 2.3.4.** *Za vsako nerazcepno markovsko verigo s končno množico stanj obstaja stacionarna porazdelitev.*

**Dokaz:** Fiksirajmo stanje  $s_k \in S$  in naj bo

$$\rho_j = \sum_{n=0}^{\infty} P(X_n = s_j, T_k > n \mid X_0 = s_k)$$

povprečno število obiskov stanja  $s_j$  med dvema zaporednima obiskoma stanja  $s_k$ . Pokažimo, da je

$$\pi = \left( \frac{\rho_1}{E(T_k \mid X_0 = s_k)}, \frac{\rho_2}{E(T_k \mid X_0 = s_k)}, \dots, \frac{\rho_N}{E(T_k \mid X_0 = s_k)} \right)$$

stacionarna porazdelitev za markovsko verigo. Pokažimo najprej, da tako definiran vektor zadošča točki (i) v definiciji 2.3.1. Očitno je  $\pi_i \geq 0$  za vsak  $i = 1, 2, \dots, N$ . Po točki (ii) iz trditve 2.2.24 je  $E(T_k \mid X_0 = s_k) < \infty$ . Poleg tega velja

$$\begin{aligned} E(T_k \mid X_0 = s_k) &= \sum_{n=0}^{\infty} P(T_k > n \mid X_0 = s_k) \\ &= \sum_{n=0}^{\infty} \sum_{i=1}^N P(X_n = s_i, T_k > n \mid X_0 = s_k) \\ &= \sum_{i=1}^N \sum_{n=0}^{\infty} P(X_n = s_i, T_k > n \mid X_0 = s_k) \\ &= \sum_{i=1}^N \rho_i. \end{aligned}$$

Torej je

$$\sum_{i=1}^N \pi_i = \frac{1}{E(T_k \mid X_0 = s_k)} \sum_{i=1}^N \rho_i = 1.$$

## 2 Markovske verige v diskretnem času

Pokažimo še, da tako definiran vektor zadošča točki (ii) v definiciji 2.3.1. Dovolj je videti, da velja

$$\rho_j = \sum_{i=1}^N \rho_i p_{ij} \text{ za } j = 1, 2, \dots, N.$$

Predpostavimo najprej, da je  $j \neq k$ . Najprej opazimo, da je za  $n = 1, 2, \dots$ , dogodek  $\{T_k > n - 1\}$  odvisen samo od dogodkov  $X_0, X_1, \dots, X_{n-1}$ . Torej v času  $n - 1$  velja markovska lastnost

$$P(X_{n-1} = s_i, X_n = s_j, T_k > n - 1 \mid X_0 = s_k) = P(X_{n-1} = s_i, T_k > n - 1 \mid X_0 = s_k)p_{ij}.$$

Za  $j \neq k$  velja še  $P(X_0 = s_j, T_k > n \mid X_0 = s_k) = P(X_n = s_j, T_k = n \mid X_0 = s_k) = 0$ , od koder sledi

$$\begin{aligned} \rho_j &= \sum_{n=0}^{\infty} P(X_n = s_j, T_k > n \mid X_0 = s_k) \\ &= \sum_{n=1}^{\infty} P(X_n = s_j, T_k > n \mid X_0 = s_k) \\ &= \sum_{n=1}^{\infty} P(X_n = s_j, T_k > n - 1 \mid X_0 = s_k) \\ &= \sum_{n=1}^{\infty} \sum_{i=1}^N P(X_{n-1} = s_i, X_n = s_j, T_k > n - 1 \mid X_0 = s_k) \\ &= \sum_{n=1}^{\infty} \sum_{i=1}^N P(X_{n-1} = s_i, T_k > n - 1 \mid X_0 = s_k)p_{ij} \\ &= \sum_{i=1}^N p_{ij} \sum_{n=1}^{\infty} P(X_{n-1} = s_i, T_k > n - 1 \mid X_0 = s_k) \\ &= \sum_{i=1}^N p_{ij} \sum_{m=0}^{\infty} P(X_m = s_i, T_k > m \mid X_0 = s_k) \\ &= \sum_{i=1}^N p_{ij} \rho_i = \sum_{i=1}^N \rho_i p_{ij}. \end{aligned}$$

Ostane nam še primer, ko je  $j = k$ . Potem je očitno, da je  $\rho_k = 1$ . Po točki (i)

iz trditve 2.2.24 velja še je  $P(T_k < \infty | X_0 = k) = 1$  in zato

$$\begin{aligned}
 \rho_k = 1 &= P(T_k < \infty | X_0 = s_k) = \sum_{n=1}^{\infty} P(T_k = n | X_0 = s_k) \\
 &= \sum_{n=1}^{\infty} \sum_{i=1}^N P(X_{n-1} = s_i, T_k = n | X_0 = s_k) \\
 &= \sum_{n=1}^{\infty} \sum_{i=1}^N P(X_{n-1} = s_i, T_k > n-1 | X_0 = s_k) p_{ik} \\
 &= \sum_{i=1}^N p_{ik} \sum_{n=1}^{\infty} P(X_{n-1} = s_i, T_k > n-1 | X_0 = s_k) \\
 &= \sum_{i=1}^N p_{ik} \sum_{m=0}^{\infty} P(X_m = s_i, T_k > m | X_0 = s_k) \\
 &= \sum_{i=1}^N \rho_i p_{ik}.
 \end{aligned}$$

■

**Primer 2.3.5.** Naj bo

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ 0 & 0 & 1 \end{bmatrix}$$

prehodna matrika, ki pripada markovski verigi  $\{X_n\}_{n \in T}$ . Ni se težko prepričati, da je za poljuben  $\alpha \in [0, 1]$  vektor

$$\pi = [\alpha, 0, 1 - \alpha]$$

stacionarna porazdelitev za  $\{X_n\}_{n \in T}$ . Torej ima veriga  $\{X_n\}_{n \in T}$  neskončno mnogo stacionarnih porazdelitev.

V primeru 2.3.5 smo videli, da markovska veriga nima nujno enolične stacionarne porazdelitve. Opozorimo, da je v tem primeru markovska veriga razcepna, saj recimo stanji 1 in 3 nista povezani. Naslednji izrek pove, da natanko razcepnost preprečuje enoličnost stacionarne porazdelitve.

**Izrek 2.3.6.** Vsaka nerazcepna markovska veriga s končno množico stanj ima natanko eno stacionarno porazdelitev.

**Dokaz:** Naj bo  $\{X_n\}_{n \in T}$  nerazcepna markovska veriga s končno  $N$ -elementno množico stanj. Po izreku 2.3.4 obstaja stacionarna porazdelitev  $\pi$  za  $\{X_n\}_{n \in T}$ . Naj bo začetna porazdelitev  $\pi^{(0)}$  enaka  $\pi$ ; tako je proces stacionaren. Izberimo poljubno stanje  $s_i \in S$  in naj bo

$$m_i = E(T_i | X_0 = s_i)$$

povprečni čas prve vrnitve v  $s_i$ . Pokažimo, da velja

$$m_i \pi_i = 1.$$

Ker je povprečni čas prve vrnitve  $m_i$  enolično določen, bo to pomenilo, da je tudi stacionarna porazdelitev enolično določena. Pišimo

$$m_i = \sum_{n=1}^{\infty} P(T_i \geq n | X_0 = s_i)$$

in upoštevajmo, da velja  $\pi_i = \pi_i^{(0)} = P(X_0 = s_i)$ . Tedaj je

$$\begin{aligned} m_i \pi_i &= \sum_{n=1}^{\infty} P(T_i \geq n | X_0 = s_i) P(X_0 = s_i) \\ &= \sum_{n=1}^{\infty} P(T_i \geq n, X_0 = s_i) \\ &= P(X_0 = s_i) + \sum_{n=2}^{\infty} P(X_0 = s_i, X_1 \neq s_i, X_2 \neq s_i, \dots, X_{n-1} \neq s_i). \end{aligned}$$

Sedaj upoštevajmo, da za poljubna dogodka  $A$  in  $B$  velja

$$P(A \cap B) = P(A) - P(A \cap B^C).$$

Vzemimo  $A = \{X_1 \neq s_i, X_2 \neq s_i, \dots, X_{n-1} \neq s_i\}$  in  $B = \{X_0 = s_i\}$  in dobimo

$$\begin{aligned} m_i \pi_i &= P(X_0 = s_i) + \sum_{n=2}^{\infty} \left( P(X_1 \neq s_i, \dots, X_{n-1} \neq s_i) - P(X_0 \neq s_i, \dots, X_{n-1} \neq s_i) \right) \\ &= P(X_0 = s_i) + \sum_{n=2}^{\infty} \left( P(X_0 \neq s_i, \dots, X_{n-2} \neq s_i) - P(X_0 \neq s_i, \dots, X_{n-1} \neq s_i) \right), \end{aligned}$$

kjer smo v zadnji vrstici zaradi stacionarnega procesa lahko indekse zmanjšali za ena. Pišimo

$$a_n = P(X_0 \neq s_i, X_1 \neq s_i, \dots, X_n \neq s_i),$$

pa dobimo

$$\begin{aligned} m_i \pi_i &= P(X_0 = s_i) + \sum_{n=2}^{\infty} (a_{n-2} - a_{n-1}) \\ &= P(X_0 = s_i) + a_0 - a_1 + a_1 - a_2 + a_2 - a_3 + \dots \end{aligned}$$

Opazimo, da je  $n$ -ta delna vsota  $S_n$  zadnje vrste enaka

$$S_n = P(X_0 = s_i) + a_0 - a_n,$$

zato je

$$m_i \pi_i = P(X_0 = s_i) + a_0 - \lim_{n \rightarrow \infty} a_n.$$

Ker je  $a_0 = P(X_0 \neq s_i)$ , velja

$$P(X_0 = s_i) + a_0 = P(X_0 = s_i) + P(X_0 \neq s_i) = 1.$$

Nazadnje pokažimo še, da je  $\lim_{n \rightarrow \infty} a_n = 0$ . Res. Opazimo, da  $\lim_{n \rightarrow \infty} a_n$  predstavlja verjetnost, da veriga nikoli ne obišče stanja  $s_i$ . Toda po posledici 2.2.27 je

$$P(T_i < \infty) = 1,$$

kar pomeni, da je obisk stanja  $s_i$  gotov dogodek. Z drugimi besedami,  $\lim_{n \rightarrow \infty} a_n = 0$ . S tem je izrek dokazan. ■

V resnici nas zanimajo limitne porazdelitve.

**Definicija 2.3.7.** Porazdelitev  $\pi^* = [\pi_1^*, \pi_2^*, \dots, \pi_N^*]$  je limitna porazdelitev za markovsko verigo  $\{X_n\}_{n \in T}$ , če za vsako začetno porazdelitev  $\pi^{(0)}$  verige velja

$$\pi_j^* = \lim_{n \rightarrow \infty} \pi_j^{(n)} = \lim_{n \rightarrow \infty} P(X_n = s_j) \text{ za vsak } j = 1, 2, \dots, N.$$

**Opomba 2.3.8.** Definicija limitne porazdelitve je dobra. Ni se namreč težko prepričati, da je  $\sum_{i=1}^N \pi_i^* = 1$ , zato  $\pi^*$  res predstavlja porazdelitev.

Limitna porazdelitev je tesno povezana s stacionarno porazdelitvijo.

**Trditev 2.3.9.** Če limitna porazdelitev obstaja, potem je enaka stacionarni porazdelitvi.

**Dokaz:** Recimo, da limitna porazdelitev  $\pi^*$  obstaja. Pokažimo, da zadošča pogojem (i) in (ii) definicije 2.3.1. V resnici je dovolj pokazati točko (ii). Po eni strani uporabimo formulo za popolno verjetnost in računajmo:

$$\begin{aligned} P(X_{n+1} = s_j) &= \sum_{i=1}^N P(X_{n+1} = s_j | X_n = s_i) P(X_n = s_i) \\ &= \sum_{i=1}^N P(X_1 = s_j | X_0 = s_i) P(X_n = s_i) \\ &= \sum_{i=1}^N P(X_n = s_i) p_{ij}, \quad s_j \in S. \end{aligned} \tag{2.14}$$

Nato levo in desno stran enakosti (2.14) pošljemo proti neskončnosti in zamenjamo vrstni red limite in vsote na desni strani. Po drugi strani, ker  $\pi^*$  obstaja, velja

$$\lim_{n \rightarrow \infty} P(X_n = s_j) = \lim_{n \rightarrow \infty} P(X_{n+1} = s_i) = \pi_j^*.$$

Slednje vstavimo v (2.14) in rezultat sledi. ■

Oglejmo si primer markovske verige, za katero limitna porazdelitev ne obstaja.

**Primer 2.3.10.** Naj bo

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0.1 & 0 & 0.9 \\ 0 & 1 & 0 \end{bmatrix}$$

prehodna matrika, ki pripada markovski verigi  $\{X_n\}_{n \in T}$ . Elementarni izračun pokaže, da je

$$\mathbf{P}^{2n} = \begin{bmatrix} 0.1 & 0 & 0.9 \\ 0 & 1 & 0 \\ 0.1 & 0 & 0.9 \end{bmatrix}, \quad n \geq 1,$$

in

$$\mathbf{P}^{2n-1} = \begin{bmatrix} 0 & 1 & 0 \\ 0.1 & 0 & 0.9 \\ 0 & 1 & 0 \end{bmatrix}, \quad n \geq 1.$$

Za poljubno začetno porazdelitev  $\pi^{(0)}$  potem dobimo

$$\pi^{(2n)} = [0.1(\pi_1^{(0)} + \pi_3^{(0)}), \pi_2^{(0)}, 0.9(\pi_1^{(0)} + \pi_3^{(0)})], \quad n \geq 1,$$

in

$$\pi^{(2n-1)} = [0.1\pi_2^{(0)}, \pi_1^{(0)} + \pi_3^{(0)}, 0.9\pi_2^{(0)}], \quad n \geq 1.$$

Očitno obstajajo začetne porazdelitve  $\pi^{(0)}$ , za katere ne obstajajo limite

$$\lim_{n \rightarrow \infty} \pi_j^{(n)}, \quad j = 1, 2, \dots, N.$$

Ker morajo po definiciji 2.3.7 obstajati zgornje limite za poljubno začetno porazdelitev  $\pi^{(0)}$ , limitna porazdelitev ne obstaja.

Opozorimo, da je v primeru 2.3.10 markovska veriga nerazcepna, zato ima veriga enolično stacionarno porazdelitev, kljub temu pa limitna porazdelitev ne obstaja. kateremu dodatnemu pogoju mora torej zadoščati markovska veriga, da bo limitna porazdelitev obstajala? Izkaže se, da je dovolj, če poleg nerazcepnosti predpostavimo, da je markovska veriga aperiodična. To pove naslednji izrek.

**Izrek 2.3.11.** *Za vsako nerazcepno aperiodično markovsko verigo s končno množico stanj obstaja limitna porazdelitev.*

**Dokaz:** Naj bo  $\{X_n\}_{n \in T}$  poljubna nerazcepna aperiodična markovska veriga s končno množico stanj, naj bo  $\mathbf{P}$  pripadajoča prehodna matrika in naj bo  $\pi^{(0)}$  poljubna začetna porazdelitev. Po izreku 2.3.4 obstaja stacionarna porazdelitev  $\pi$  za  $\{X_n\}_{n \in T}$ . Potem lahko konstruiramo markovsko verigo  $\{Y_n\}_{n \in T}$ , neodvisno od  $\{X_n\}_{n \in T}$ , s prehodno matriko  $\mathbf{P}$  in začetno porazdelitvijo  $\pi$ . Poglejmo si proces  $\{W_n\}_{n \in T} = \{(X_n, Y_n)\}_{n \in T}$ . Ni se težko prepričati, da je  $\{W_n\}_{n \in T}$  markovska veriga na končni množici stanj  $S \times S$ , s prehodnimi verjetnostmi

$$\tilde{p}_{(i,k)(j,l)} = p_{ij}p_{kl}$$

### 2.3 Stacionarna in limitna porazdelitev

in začetno porazdelitvijo

$$\pi_{(i,k)}^{(0)} = \pi_i^{(0)} \pi_k.$$

Ker je veriga  $\{X_n\}_{n \in T}$  nerazcepna in aperiodična, po izreku 2.2.12 sledi, da za poljubna stanja  $s_i, s_j, s_k, s_l$  velja

$$\tilde{p}_{(i,k)(j,l)}^{(n)} = p_{ij}^{(n)} p_{kl}^{(n)} > 0$$

za dovolj velike čase  $n$ ; torej je  $\{W_n\}_{n \in T}$  tudi nerazcepna veriga. Fiksirajmo stanje  $s_b \in S$  in naj bo

$$T_{(b,b)} = \min\{n \geq 1 \mid X_n = Y_n = s_b\}$$

čas prvega prihoda v stanje  $(s_b, s_b)$ . Po posledici 2.2.27 velja  $P(T_{(b,b)} < \infty) = 1$ . Z drugimi besedami,

$$\lim_{n \rightarrow \infty} P(T_{(b,b)} > n) = 0.$$

Poleg tega se ni težko prepričati, da sta porazdelitvi procesov  $\{X_n\}_{n \in T}$  in  $\{Y_n\}_{n \in T}$  identični za vsak čas  $n \geq T_{(b,b)}$ . Naj bo  $X_0 = s_i$  in  $Y_0 = s_j$  za poljubni stanji  $s_i, s_j \in S$ . Velja torej:

$$\begin{aligned} p_{ik}^{(n)} &= P(X_n = s_k \mid X_0 = s_i) \\ &= P(X_n = s_k \mid T_{(b,b)} \leq n, X_0 = s_i) P(T_{(b,b)} \leq n) \\ &\quad + P(X_n = s_k \mid T_{(b,b)} > n, X_0 = s_i) P(T_{(b,b)} > n) \\ &= P(Y_n = s_k \mid T_{(b,b)} \leq n, Y_0 = s_j) P(T_{(b,b)} \leq n) \\ &\quad + P(X_n = s_k \mid T_{(b,b)} > n, X_0 = s_i) P(T_{(b,b)} > n) \\ &\leq P(Y_n = s_k \mid Y_0 = s_j) + P(T_{(b,b)} > n) \\ &= p_{jk}^{(n)} + P(T_{(b,b)} > n). \end{aligned}$$

V tem premisleku lahko vlogi  $s_i$  in  $s_j$  zamenjamo, zato velja

$$|p_{ik}^{(n)} - p_{jk}^{(n)}| \leq P(T_{(b,b)} > n).$$

Sledi  $\lim_{n \rightarrow \infty} (p_{ik}^{(n)} - p_{jk}^{(n)}) = 0$ . Toda ob upoštevanju, da je  $\sum_{i=1}^N \pi_i = 1$  in  $\pi_k = \sum_{i=1}^N \pi_i p_{ik}^{(n)}$ , dobimo

$$\begin{aligned} \pi_k - p_{jk}^{(n)} &= \sum_{i=1}^N \pi_i p_{ik}^{(n)} - \sum_{i=1}^N \pi_i p_{jk}^{(n)} \\ &= \sum_{i=1}^N \pi_i (p_{ik}^{(n)} - p_{jk}^{(n)}). \end{aligned}$$

Od tod sledi

$$\begin{aligned} \lim_{n \rightarrow \infty} (\pi_k - p_{jk}^{(n)}) &= \lim_{n \rightarrow \infty} \sum_{i=1}^N \pi_i (p_{ik}^{(n)} - p_{jk}^{(n)}) \\ &= \sum_{i=1}^N \pi_i \lim_{n \rightarrow \infty} (p_{ik}^{(n)} - p_{jk}^{(n)}) = 0 \end{aligned}$$

## 2 Markovske verige v diskretnem času

oziroma

$$\lim_{n \rightarrow \infty} p_{jk}^{(n)} = \pi_k.$$

Nazadnje upoštevajmo še Chapman-Kolmogorove enačbe

$$\lim_{n \rightarrow \infty} P(X_n = s_k) = \lim_{n \rightarrow \infty} \sum_{j=1}^N \pi_j^{(0)} p_{jk}^{(n)} = \sum_{j=1}^N \pi_j^{(0)} \lim_{n \rightarrow \infty} p_{jk}^{(n)} = \sum_{j=1}^N \pi_j^{(0)} \pi_k = \pi_k$$

in izrek je dokazan. ■



---

# Metoda Monte Carlo

Nekateri problemi so preveč kompleksni, da bi jih obravnavali analitično. V takšnih primerih se pogosto zatečemo k računalniškim simulacijam. V grobem lahko rečemo, da je računalniška simulacija umetno generiranje podatkov, ki mu sledi načrtovanje modela in izvajanje numeričnih eksperimentov. Pogosto je pomembno, da se nekaterim spremenljivkam v modelu dodeli slučajne vrednosti. Ključno vlogo pri tem igrajo generatorji slučajnih števil. Metodo Monte Carlo lahko poenostavljeno opišemo kot tehniko, kjer za izračun numeričnih rezultatov uporabljamo slučajna števila.

Metoda Monte Carlo se je prvič pojavila med drugo svetovno vojno pri načrtovanju vodikove atomske bombe. Predlagala sta jo John von Neumann in Stanislaw Ulam, ko sta reševala problem difuzije nevtronov. Skupaj z drugimi raziskovalci so uporabili simulacijo pri mnogih problemih v zvezi z jedrskim orožjem in tako vzpostavili temelje Monte Carlo simulacije. S hitrim razvojem računalniške tehnologije v osemdesetih in devetdesetih letih dvajsetega stoletja je metoda doživela razcvet.

Danes je metoda Monte Carlo precej razširjeno orodje na različnih področjih: v fiziki za načrtovanje nuklearnih reaktorjev, v medicini za načrtovanje obsevanj, v energetiki in metalurgiji za obravnavo toplotnega sevanja, v prometu za analizo prometnih tokov, v ekonomiji za napovedovanje dogodkov na borzi in v računalniški grafiki za generiranje slik. To so le nekatere izmed številnih aplikacij Monte Carlo tehnik.

V tem poglavju bomo predstavili osnovno idejo metode Monte Carlo. Pri tem večinoma sledimo [14] in [24]. Več o konceptu metode Monte Carlo in njeni uporabi je moč najti v [7, 17, 18, 23, 25, 27, 36].

## 3.1 Osnovni koncept

Kadarkoli simuliramo slučajni proces, lahko aproksimacijo matematičnega upanja izračunamo iz povprečja simulacij. Natančneje, naj bodo  $\underline{X}_1, \underline{X}_2, \dots, \underline{X}_M$  neodvisni, enako porazdeljeni slučajni vektorji, ki simulirajo nek verjetnostni model, in

naj slučajni vektor  $\underline{X}$  označuje generično realizacijo tega modela. Torej so vsi slučajni vektorji  $\underline{X}_i$ ,  $i = 1, 2, \dots, M$ , enako porazdeljeni kot  $\underline{X}$ . Zaporedju takšnih slučajnih vektorjev rečemo *vzorec porazdelitve*. Poljubni realni zvezni funkciji  $g$  na  $\underline{X}_i$  rečemo *vzorčna funkcija*. Opozorimo, da je  $g(\underline{X}_i)$  slučajna spremenljivka. Recimo, da je dana realna funkcija  $g$  na  $\underline{X}$  in da želimo izračunati matematično upanje  $\mu = E(g(\underline{X}))$ . Pogosto se zgodi, da  $\mu$  ne znamo oziroma ne moremo izračunati analitično. Monte Carlo je metoda za oceno matematičnega upanja.

**Definicija 3.1.1.** Monte Carlo aproksimacija matematičnega upanja  $\mu$  je vzorčno povprečje

$$\hat{\mu} = \frac{1}{M} \sum_{i=1}^M g(\underline{X}_i).$$

**Opomba 3.1.2.** Vzorčno povprečje  $\hat{\mu}$  je slučajna spremenljivka, ki ji rečemo tudi cenilka za matematično upanje. Cenilka je nepristranska, saj velja

$$E(\hat{\mu}) = E\left(\frac{1}{M} \sum_{i=1}^M g(\underline{X}_i)\right) = \frac{1}{M} \sum_{i=1}^M E(g(\underline{X}_i)) = \frac{1}{M} \sum_{i=1}^M E(g(\underline{X})) = E(g(\underline{X})).$$

Pomembno vlogo pri oceni igra krepki zakon velikih števil, ki pove naslednje: če so  $X_1, X_2, \dots$  neodvisne, enako porazdeljene slučajne spremenljivke, pri čemer je  $E(X_1) < \infty$ , potem velja

$$P\left(\lim_{M \rightarrow \infty} \frac{X_1 + X_2 + \dots + X_M}{M} = E(X_1)\right) = 1.$$

Ker so slučajni vektorji  $\underline{X}_1, \underline{X}_2, \dots, \underline{X}_M$  neodvisni in enako porazdeljeni, so tudi slučajne spremenljivke  $g(\underline{X}_1), g(\underline{X}_2), \dots, g(\underline{X}_M)$  neodvisne in enako porazdeljene. Poleg tega ima vsaka slučajna spremenljivka  $g(\underline{X}_i)$ ,  $i = 1, 2, \dots, M$ , matematično upanje enako  $\mu$ . Od tod lahko sklepamo, da velja

$$P\left(\lim_{M \rightarrow \infty} \frac{g(\underline{X}_1) + g(\underline{X}_2) + \dots + g(\underline{X}_M)}{M} = \mu\right) = 1,$$

brž ko  $\mu$  obstaja. Z drugimi besedami, če je vzorec dovolj velik, potem je majhna verjetnost, da se vzorčno povprečje  $\hat{\mu}$  precej razlikuje od matematičnega upanja  $\mu$ .

Če združimo vse povedano skupaj, potem je osnovna ideja metode Monte Carlo naslednja:

- (i) generiramo vzorec  $\underline{X}_1, \underline{X}_2, \dots, \underline{X}_M$  porazdelitve slučajnega vektorja  $\underline{X}$ , in
- (ii) izračunamo vzorčno povprečje

$$\hat{\mu} = \frac{1}{M} \sum_{i=1}^M g(\underline{X}_i).$$

### 3.1 Osnovni koncept

Krepki zakon velikih števil nam torej ponuja recept za določanje ocene matematičnega upanja, vendar pa nam ne pove, kako natančna je metoda. Kako velik mora biti vzorec v praksi, da bo napaka majhna? Pomembna lastnost metode Monte Carlo – poleg tega, da poišče aproksimacijo za matematično upanje – je, da lahko dobimo oceno nezanesljivosti v aproksimaciji. To nam omogoča centralni limitni izrek, ki pove naslednje: če so  $X_1, X_2, \dots$  neodvisne, enako porazdeljene slučajne spremenljivke, ki imajo končno matematično upanje  $E(X_i)$  in končno varianco  $\text{var}(X_i) = \sigma^2(X_i)$ , potem je asimptotična porazdelitev slučajne spremenljivke

$$\frac{\sqrt{M}}{\sigma(X_1)} \left( \frac{X_1 + X_2 + \dots + X_M}{M} - E(X_1) \right)$$

enaka standardni normalni porazdelitvi.

Najprej opozorimo, da je Monte Carlo stohastična metoda, kar pomeni, da ob ponovnih izvršitvah algoritma pri istih vhodnih podatkih tipično dobimo različne izhodne rezultate. Torej imamo opravka s stohastično napako. Predpostavimo, da obstaja varianca slučajne spremenljivke  $g(\underline{X})$  in s  $\sigma$  označimo njen standardni odklon. Iz centralnega limitnega izreka sledi, da se porazdelitev slučajne spremenljivke

$$\frac{\sqrt{M}}{\sigma} (\hat{\mu} - \mu)$$

asimptotično približuje standardni normalni porazdelitvi. Torej za vsak  $z^* \geq 0$  in za dovolj velik vzorec velja naslednja aproksimacija

$$P\left(-z^* \leq \frac{\sqrt{M}}{\sigma} (\hat{\mu} - \mu) \leq z^*\right) \approx P(-z^* \leq Z \leq z^*)$$

oziroma

$$P\left(\hat{\mu} - z^* \frac{\sigma}{\sqrt{M}} \leq \mu \leq \hat{\mu} + z^* \frac{\sigma}{\sqrt{M}}\right) \approx P(-z^* \leq Z \leq z^*), \quad (3.1)$$

kjer je  $Z$  standardna normalna slučajna spremenljivka. Če izberemo parameter  $z^*$ , da velja  $P(-z^* \leq Z \leq z^*) = 1 - \alpha$ , potem lahko zaključimo, da  $\mu$  leži znotraj intervala

$$\left[ \hat{\mu} - z^* \frac{\sigma}{\sqrt{M}}, \hat{\mu} + z^* \frac{\sigma}{\sqrt{M}} \right]$$

z (aproksimativno) verjetnostjo  $1 - \alpha$ . Takšnemu intervalu rečemo *aproksimativni*  $(1 - \alpha)$ -*interval zaupanja*, parametru  $\alpha$  pa *stopnja zaupanja*. Tipične vrednosti za  $\alpha$  so 0.1, 0.05 ali 0.01.

V praksi so potrebne nadaljnje aproksimacije. Prvič, standardni odklon  $\sigma$ , ki je potreben, da določimo interval zaupanja, je pogosto neznan, zato ga moramo oceniti z *vzorčnim standardnim odklonom*

$$\hat{\sigma} = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (g(Y_i) - \hat{\mu})^2},$$

normalno porazdelitev pa zamenjati s Studentovo  $t$ -porazdelitvijo. Dobro je znano, da se Studentova  $t$ -porazdelitev za velike vzorce približuje normalni porazdelitvi. Posledično je za vsak  $t^* \geq 0$  in za dovolj velik vzorec aproksimacija (3.1) ekvivalentna aproksimaciji

$$P\left(\hat{\mu} - t^* \frac{\hat{\sigma}}{\sqrt{M}} \leq \mu \leq \hat{\mu} + t^* \frac{\hat{\sigma}}{\sqrt{M}}\right) \approx P(-t^* \leq T \leq t^*),$$

kjer je  $T$  slučajna spremenljivka s Studentovo  $t$ -porazdelitvijo z  $M - 1$  prostostnimi stopnjami. Za parameter  $t^*$ , ki ustreza enačbi  $P(-t^* \leq T \leq t^*) = 1 - \alpha$ , tako dobimo aproksimativni  $(1 - \alpha)$ -interval zaupanja

$$\left[\hat{\mu} - t^* \frac{\hat{\sigma}}{\sqrt{M}}, \hat{\mu} + t^* \frac{\hat{\sigma}}{\sqrt{M}}\right]. \quad (3.2)$$

Drugič, kot smo že omenili zgoraj, nas zanima velikost vzorca  $M$  v odvisnosti od želene natančnosti in intervala zaupanja. Če želimo, da ima končni aproksimativni interval zaupanja dolžino  $2\epsilon|\mu|$  za nek  $\epsilon > 0$  (to pomeni, da je relativna napaka enaka  $\epsilon$ ), potem iz (3.2) sledi

$$\epsilon|\mu| = t^* \frac{\hat{\sigma}}{\sqrt{M}}.$$

Nazadnje,  $\mu$  aproksimiramo z  $\hat{\mu}$  in od tod dobimo oceno za velikost vzorca

$$M \approx \left(\frac{t^* \hat{\sigma}}{\hat{\mu} \epsilon}\right)^2. \quad (3.3)$$

V praksi potrebno velikost vzorca ocenimo tako, da za izbrano relativno napako  $\epsilon$  in stopnjo zaupanja  $\alpha$  najprej generiramo manjši vzorec velikosti  $M_0$ . Nato za ta vzorec poračunamo vzorčno povprečje in vzorčni standardni odklon ter iz (3.3) poračunamo oceno za  $M$ . Na koncu na novo generiramo vzorec velikosti  $M$ .

## 3.2 Generiranje vzorca

Generiranje vzorca porazdelitve slučajnega vektorja je ključni del metode Monte Carlo. Recimo, da želimo generirati slučajni vektor  $(X_0, X_1, \dots, X_k)$  s skupno porazdelitveno funkcijo  $F_{(X_0, X_1, \dots, X_k)}(x_0, x_1, \dots, x_k)$ . Potem tipično najprej generiramo slučajno spremenljivko  $X_0$  z robno porazdelitveno funkcijo  $F_{X_0}(x_0)$ , nato generiramo slučajno spremenljivko  $X_1$  s pogojno porazdelitveno funkcijo  $F_{X_1|X_0}(x_1)$ , nato slučajno spremenljivko  $X_2$  s pogojno porazdelitveno funkcijo  $F_{X_2|X_0, X_1}(x_2)$  in tako naprej. Če znamo torej generirati slučajno spremenljivko s poljubno porazdelitvijo, znamo generirati tudi slučajni vektor s poljubno porazdelitvijo, brž ko znamo iz skupne porazdelitvene funkcije poiskati robne in pripadajoče pogojne porazdelitvene funkcije. Kot bomo videli v nadaljevanju, lahko takšen pristop uporabimo za generiranje markovskih verig.

Poglejmo si, kako generiramo vzorec poljubne porazdelitve slučajne spremenljivke  $X$ . Kaj točno to sploh pomeni? To pomeni, da je za dovolj velik vzorec

### 3.2 Generiranje vzorca

$X_1, X_2, \dots, X_M$  delež spremenljivk  $X_i$ , ki padejo v poljuben interval  $[a, b]$ , blizu verjetnosti  $P(X \in [a, b])$ , to je

$$\frac{|\{i \mid X_i \in [a, b], i = 1, 2, \dots, M\}|}{M} \approx P(X \in [a, b]).$$

Precej tehnik obstaja za generiranje vzorca dane porazdelitve. Mogoče najbolj pogosto uporabljena metoda za generiranje vzorca porazdelitve diskretne slučajne spremenljivke, ki ima dva enako verjetna dogodka, je met kovanca. Podobno se lahko uporabi met kocke za generiranje vzorca porazdelitve diskretne slučajne spremenljivke, ki ima šest enako verjetnih dogodkov.

Pomembno vlogo pri računalniškem generiranju igrajo vzorci enakomerne porazdelitve na enotskem intervalu  $[0, 1]$  slučajne spremenljivke  $U$ , ki jim rečemo *vzorci slučajnih števil*. Generiranje vzorca porazdelitve slučajne spremenljivke  $X$  namreč sestoji iz naslednjih dveh korakov:

- (i) generiraj vzorec  $U_1, U_2, \dots, U_M$  slučajnih števil,
- (ii) prevedi vzorec  $U_1, U_2, \dots, U_M$  slučajnih števil v vzorec  $X_1, X_2, \dots, X_M$  porazdelitve spremenljivke  $X$ .

Vzorec povsem slučajnih števil je težko generirati. V večini programskih jezikov imamo dostop do generatorja slučajnih števil. No, če smo natančni, zadnji stavek ni čisto resničen vsaj zaradi naslednjih dveh razlogov:

- (a) Vzorec  $U_1, U_2, \dots, U_M$ , ki nam ga vrne generator slučajnih števil, nima enakomerne porazdelitve na enotskem intervalu  $[0, 1]$ . Tipično imajo spremenljivke v vzorcu končen binaren (ali decimalen) zapis in so zato racionalne. Toda izkaže se, da je slučajno število, ki ima porazdelitev resnično enako enakomerni porazdelitvi na  $[0, 1]$ , iracionalno z verjetnostjo 1.
- (b) Vzorec  $U_1, U_2, \dots, U_M$  ni niti slučajen, saj ga dobimo z determinističnim algoritmom. Zato včasih (in bolj točno) generatorjem slučajnih števil rečemo *generatorji pseudoslučajnih števil*.

Pomanjkljivost (b) predstavlja precej večjo oviro, saj se zdi, da pomanjkljivost (a) ni velik problem, ko je število binarnih ali decimalnih števk dovolj veliko, recimo 32 bitov. Skozi desetletja je bilo precej več pozornosti namenjeno reševanju pomanjkljivosti (b). Danes obstajajo generatorji, ki zelo dobro odpravljajo to pomanjkljivost. Zaradi tega razloga bomo preprosto predpostavili, da imamo dostop do generiranja slučajnih števil. Čeprav se ne bomo bolj podrobno poglobljali v problem (psevo)slučajnosti, pa bomo imeli vedno v mislih, da je potencialni vir napak pri računalniškem generiranju.

Za prevedbo vzorca slučajnih števil v vzorec poljubno porazdeljene slučajne spremenljivke obstajajo različne metode, kot so na primer inverzna transformacijska metoda, metoda sprejema in zavrnitve, polarna metoda, metoda trakov in mrežna metoda. Najpogosteje se uporablja inverzna transformacijska metoda, ki si jo bomo sedaj pogledali bolj natančno.

### 3.2.1 Inverzna transformacijska metoda

Čeprav lahko inverzno transformacijsko metodo uporabimo za generiranje vzorca poljubne porazdelitve tako zvezne kot diskretne slučajne spremenljivke, pa za naše potrebe zadošča, če se omejimo na diskretne spremenljivke. Recimo, da želimo generirati diskretno slučajno spremenljivko  $X$ , ki zavzame diskretne vrednosti  $x_1 < x_2 < \dots < x_N$  z verjetnostmi

$$P(X = x_i) = p_i, \quad i = 1, 2, 3, \dots, N,$$

in ima porazdelitveno funkcijo enako

$$F_X(x) = P(X \leq x) = \sum_{x_j \leq x} p_j.$$

Natanko to naredi naslednji algoritem:

- (i) generiraj slučajno število  $U$ ,
- (ii) poišči  $x_i$ , da je  $F_X(x_{i-1}) \leq U < F_X(x_i)$ ,
- (iii) vrni  $X = x_i$ .

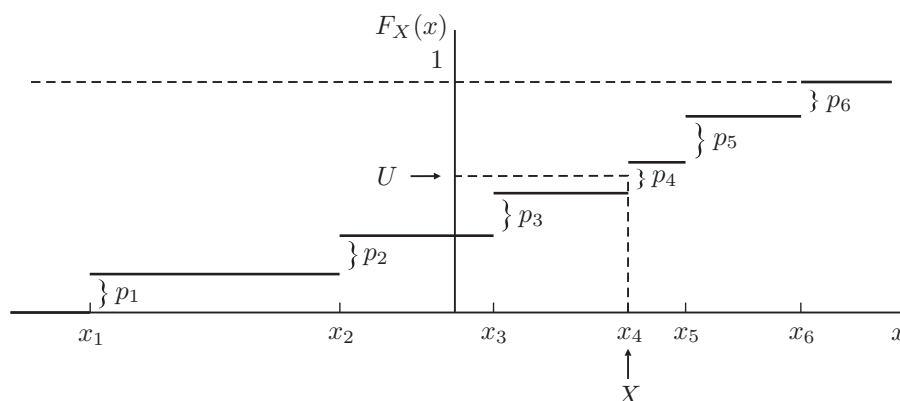
Naslednji račun utemeljuje pravilnost algoritma:

$$P(X = x_i) = P(F_X(x_{i-1}) \leq U < F_X(x_i)) = F_X(x_i) - F_X(x_{i-1}) = \sum_{j=1}^i p_j - \sum_{j=1}^{i-1} p_j = p_i.$$

Grafična ilustracija algoritma je prikazana na sliki 3.1. Če definiramo posplošen inverz  $F_X^{-1}(x)$  porazdelitvene funkcije  $F_X(x)$  kot

$$F_X^{-1}(x) = \min\{y \mid F_X(y) \geq x\},$$

potem je  $X = F_X^{-1}(U)$ . Od tod tudi ime inverzna metoda.



Slika 3.1: Inverzna transformacijska metoda.

V koraku (ii) je potrebno poiskati vrednost  $x_j$ . Na voljo imamo več računskih možnosti. Če se naivno sprehodimo čez vse diskretne vrednosti, potem  $x_j$  v najslabšem primeru najdemo v linearnem času glede na število  $N$ . Boljša ideja je iskanje z bisekcijo. Naslednji algoritem je prirejen za iskanje vrednosti  $x_i$  z bisekcijo v primeru, da so diskretne vrednosti  $x_j$ ,  $j = 1, 2, \dots, N$ , cela števila:

- (i) namesti  $a := x_1$  in  $b := x_N$ ,
- (ii) dokler  $a < b - 1$  ponavljaj
  - (a) vzemi celo število  $d$ , ki je najbližje  $\frac{a+b}{2}$ ,
  - (b) če je  $F_X(d) < U$ , potem  $a := d$ , sicer  $b := d$ ,
- (iii) vrni  $x_i = b$ .

Z bisekcijo bomo tako  $x_j$  v najslabšem primeru našli v logaritemskem času glede na število  $N$ . Poglejmo si sedaj, kako inverzno transformacijsko metodo uporabimo za generiranje markovske verige.

### 3.2.2 Generiranje markovskih verig

Kako generiramo homogeno diskretno markovsko verigo  $\{X_n\}_{n \in T}$  s končno množico stanj  $S = \{s_1, s_2, \dots, s_N\}$  in pripadajočo prehodno matriko  $\mathbf{P}$  ter začetno porazdelitvijo  $\pi^{(0)}$ ? Uporabimo pristop, ki smo ga opisali v uvodu razdelka 3.2, skupaj z inverzno transformacijsko metodo in recept je na dlani.

Preden podamo algoritem, uvedimo še pojem kumulativne matrike. *Kumulativna matrika*  $Q$  glede na prehodno matriko  $\mathbf{P}$  je  $N \times N$  matrika, kjer je  $i$ -ta vrstica  $Q[i]$  enaka kumulativni porazdelitvi  $i$ -te vrstice  $\mathbf{P}[i]$ :

$$Q = \begin{bmatrix} p_{11} & p_{11} + p_{12} & \dots & p_{11} + p_{12} + \dots + p_{1(N-1)} & 1 \\ p_{21} & p_{21} + p_{22} & \dots & p_{21} + p_{22} + \dots + p_{2(N-1)} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{N1} & p_{N1} + p_{N2} & \dots & p_{N1} + p_{N2} + \dots + p_{N(N-1)} & 1 \end{bmatrix}.$$

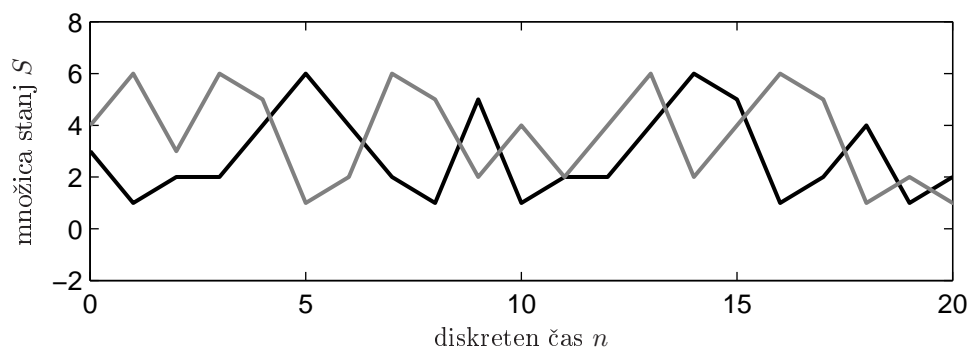
Naj bo  $Y_i$  diskretna slučajna spremenljivka s porazdelitveno funkcijo  $F_{Y_i}(x)$ , ki je porojena iz  $i$ -te vrstice kumulativne matrike  $Q$ :

$$F_{Y_i}(x) = \begin{cases} 0, & \text{če } x < 1, \\ Q_{ij}, & \text{če } j - 1 < x \leq j \text{ za nek } j \in \{1, 2, \dots, N\}, \\ 1, & \text{če } x > N. \end{cases}$$

Sedaj imamo pripravljeno vse, da podamo algoritem za generiranje prvih  $k$  korakov markovske verige  $\{X_n\}_{n \in T}$ :

- (i) generiraj  $X_0$  s porazdelitvijo  $\pi^{(0)}$  in namesti  $n := 0$ ,
- (ii) dokler  $n < k$  ponavljaj
  - $s_i := X_n$ , generiraj  $Y_i$  s porazdelitvijo  $F_{Y_i}(x)$ ,  $n := n + 1$ ,  $X_n := Y_i$ ,
- (iii) vrni  $X_0, X_1, \dots, X_k$ .

**Primer 3.2.1.** Vzemimo markovsko verigo z množico stanj  $S = \{1, 2, 3, 4, 5, 6\}$  iz primera 2.1.5. Slika 3.2 ilustrira generiranje prvih dvajsetih korakov te verige z različnima začetnima porazdelitvima  $[0, 0, 1, 0, 0, 0]$  in  $[0, 0, 0, 1, 0, 0]$ . Seveda bi pri ponovnem generiranju v splošnem dobili drugačen graf.



Slika 3.2: Generiranje prvih dvajsetih korakov markovske verige iz primera 2.1.5 pri dveh različnih začetnih pogojih.



---

# Matematično upanje in potence prehodnih matrik

Poglavje začnemo z notacijo in osnovnimi pojmi, ki so podlaga za opis raziskovalnega problema. V razdelku 4.1.1 formuliramo osnovni problem, ki ga v razdelku 4.1.2 posplošimo. Sledi reševanje problema z uporabo metode Monte Carlo. V razdelku 4.2 najprej obravnavamo osnovni problem, nato pa podamo tri pristope za reševanje posplošenega problema. Nazadnje v razdelku 4.3 povzamemo še metodo za reševanje osnovnega problema, ki so jo predlagali v [19].

## 4.1 Problem

Naj bo  $\{X_n\}_{n \in T}$  diskretna (ne nujno homogena) markovska veriga s končno množico stanj  $S = \{s_1, s_2, \dots, s_N\}$  in pripadajočo prehodno matriko  $\mathbf{P}$  (morda odvisno od časa  $n$ ), ki opisuje gibanje delca med stanji. Za fiksno naravno število  $k$  nas zanima prvih  $k$  korakov, ki jih naredi delec, to je, oglejmo si prvih  $k+1$  členov  $X_0, X_1, \dots, X_k$  markovske verige  $\{X_n\}_{n \in T}$ . Potem lahko gibanje delca opišemo s slučajnim vektorjem  $(X_0, X_1, \dots, X_k)$ , ki mu rečemo *slučajni sprehod dolžine  $k$* . Poljubni realizaciji  $(s_{j_0}, s_{j_1}, \dots, s_{j_k})$  vektorja  $(X_0, X_1, \dots, X_k)$ , to je poljubnemu elementu njegove zaloge vrednosti, zaradi enostavnosti prav tako rečemo (slučajni) sprehod dolžine  $k$  glede na  $\mathbf{P}$  in ga označimo z

$$W^{\mathbf{P}} = s_{j_0} s_{j_1} \cdots s_{j_k}.$$

Poleg tega naj bo dana realna funkcija  $c$  (morda odvisna od časa  $n$ ) na slučajnem sprehodu, ki opisuje njegovo ceno. Cenovna funkcija  $c$  je enolično določena z  $N \times N$  cenovno matriko  $C$  (morda odvisno od časa  $n$ ), kjer element  $c_{ij} \in \mathbb{R}$  označuje ceno prehoda iz stanja  $s_i$  v stanje  $s_j$ . Natančneje, funkcijo  $c$  lahko zapišemo v naslednji obliki

$$c(s_{j_0}, s_{j_1}, \dots, s_{j_k}) = \gamma(s_{j_0}, s_{j_1}) + \gamma(s_{j_1}, s_{j_2}) + \cdots + \gamma(s_{j_{k-1}}, s_{j_k}), \quad (4.1)$$

kjer je  $\gamma$  realna funkcija, definirana kot  $\gamma(s_i, s_j) = c_{ij}$ ,  $i, j = 1, 2, \dots, N$ .

Nadalje, privzemimo, da markovska veriga  $\{X_n\}_{n \in T}$  začne v izbranem začetnem stanju  $s_{j_0} \in S$ . Z drugimi besedami, začetna porazdelitev  $\pi^{(0)}$  naj bo enaka

$$\pi_i^{(0)} = \begin{cases} 1, & \text{če } i = j_0, \\ 0, & \text{sicer.} \end{cases}$$

Bralca spomnimo, da so s prehodno matriko  $\mathbf{P}$  in začetno porazdelitvijo  $\pi^{(0)}$  vse prehodne porazdelitve  $\pi^{(n)}$  verige  $\{X_n\}_{n \in T}$  enolično določene, s tem pa tudi porazdelitev slučajnega vektorja  $(X_0, X_1, \dots, X_k)$ . Njegova skupna verjetnostna funkcija je tedaj enaka

$$P(X_0 = s_{j_0}, \dots, X_k = s_{j_k}) = P(X_1 = s_{j_1} | X_0 = s_{j_0}) \cdots P(X_k = s_{j_k} | X_{k-1} = s_{j_{k-1}}),$$

kjer smo upoštevali začetno porazdelitev  $P(X_0 = s_{j_0}) = \pi_{j_0}^{(0)} = 1$ . Pri naših predpostavkah torej slučajni vektor  $(X_0, X_1, \dots, X_k)$  opisuje gibanje delca v prvih  $k$  korakih začenši v  $s_{j_0}$ . Zalogo vrednosti vektorja  $(X_0, X_1, \dots, X_k)$ , torej množico vseh sprehodov dolžine  $k$  začenši v  $s_{j_0}$ , označimo s  $S_{j_0}^k$ .

### 4.1.1 Osnovni problem

Predpostavimo, da je markovska veriga  $\{X_n\}_{n \in T}$  homogena, torej je prehodna matrika  $\mathbf{P}$  neodvisna od časa  $n$ . Poleg tega privzemimo, da je tudi cenovna matrika  $C$  neodvisna od časa  $n$ . Verjetnost sprehoda  $W^{\mathbf{P}} = s_{j_0} s_{j_1} \cdots s_{j_k}$  označimo s  $p_{W^{\mathbf{P}}}$  in je enaka

$$p_{W^{\mathbf{P}}} = P(X_0 = s_{j_0}, X_1 = s_{j_1}, \dots, X_k = s_{j_k}) = p_{j_0 j_1} p_{j_1 j_2} \cdots p_{j_{k-1} j_k}.$$

Podobno ceno vzdolž sprehoda  $W^{\mathbf{P}} = s_{j_0} s_{j_1} \cdots s_{j_k}$  označimo s  $c_{W^{\mathbf{P}}}$  in je po (4.1) enaka

$$c_{W^{\mathbf{P}}} = c_{j_0 j_1} + c_{j_1 j_2} + \cdots + c_{j_{k-1} j_k}.$$

Radi bi izračunali matematično upanje cene slučajnega sprehoda  $(X_0, X_1, \dots, X_k)$  dolžine  $k$  začenši v  $s_{j_0}$ :

$$E(c(X_0, \dots, X_k)) = \sum_{W^{\mathbf{P}} \in S_{j_0}^k} c_{W^{\mathbf{P}}} p_{W^{\mathbf{P}}}. \quad (4.2)$$

**Opomba 4.1.1.** Če želimo analitično izračunati upanje  $E(c(X_0, \dots, X_k))$ , potem je v splošnem problem ta, da moč množice  $S_{j_0}^k$  raste eksponentno s parametrom  $k$ . Še več, moč množice stanj  $S$  je prav tako lahko velika v praksi. Torej z naraščajočima parametroma  $k$  in  $n$  postane analitičen izračun nepraktičen tako v smislu časovne kot tudi prostorske zahtevnosti.

**Opomba 4.1.2.** Kaj lahko povemo o obnašanju porazdelitve slučajne spremenljivke  $c(X_0, \dots, X_k)$  za dovolj velike vrednosti  $k$ ? Če upoštevamo zvezo (4.1), lahko slučajno spremenljivko  $c(X_0, \dots, X_k)$  zapišemo kot

$$c(X_0, \dots, X_k) = \gamma(X_0, X_1) + \gamma(X_1, X_2) + \cdots + \gamma(X_{k-1}, X_k)$$

vsoto slučajnih spremenljivk  $\gamma(X_n, X_{n+1})$ ,  $n = 0, 1, \dots, k-1$ . Opazimo, da je slučajni proces  $\{\gamma(X_n, X_{n+1})\}_{n \in T}$  spet markovska veriga. Če bi bile spremenljivke  $\gamma(X_n, X_{n+1})$  neodvisne in enako porazdeljene, bi lahko uporabili centralni limitni izrek, ki bi zagotovil, da se porazdelitev spremenljivke  $c(X_0, \dots, X_k)$  za dovolj velike vrednosti  $k$  približuje normalni porazdelitvi. Ker te predpostavke niso nujno izpolnjene, splošnega centralnega limitnega izreka na zaporedju  $\{\gamma(X_n, X_{n+1})\}_{n \in T}$  ne moremo uporabiti. Vendar pa obstaja posplošena verzija centralnega limitnega izreka za markovske verige (glej na primer [33]), ki ga lahko uporabimo na  $\{\gamma(X_n, X_{n+1})\}_{n \in T}$ . Ta pove, da se pod določenimi pogoji porazdelitev spremenljivke

$$\gamma(X_0, X_1) + \gamma(X_1, X_2) + \dots + \gamma(X_{k-1}, X_k)$$

za dovolj velike vrednosti  $k$  približuje normalni porazdelitvi.

### 4.1.2 Posplošitev osnovnega problema

Predpostavimo sedaj, da je markovska veriga  $\{X_n\}_{n \in T}$  nehomogena, in naj bo pripadajoča  $N \times N$  prehodna matrika  $\mathbf{P}(n)$  enaka

$$\mathbf{P}(n) = \begin{cases} \mathbf{P}_1, & \text{če } k(l_1 + l_2) + 1 \leq n \leq k(l_1 + l_2) + l_1, k \in \mathbb{N} \cup \{0\}, \\ \mathbf{P}_2, & \text{če } k(l_1 + l_2) + l_1 + 1 \leq n \leq (k+1)(l_1 + l_2), k \in \mathbb{N} \cup \{0\}, \end{cases} \quad (4.3)$$

kjer sta  $l_1$  in  $l_2$  fiksni naravni števili,  $\mathbf{P}_1$  in  $\mathbf{P}_2$  pa  $N \times N$  stohastični matriki, neodvisni od časa  $n$ . Gibanje delca je torej opisano z odsekovno homogeno markovsko verigo: na prvih  $l_1$  korakih je določeno s prehodno matriko  $\mathbf{P}_1$ , na naslednjih  $l_2$  korakih s prehodno matriko  $\mathbf{P}_2$ , nato se vzorec ponavlja. Podobno naj bo tudi  $N \times N$  cenovna matrika  $C(n)$  odvisna od časa  $n$  in naj bo oblike

$$C(n) = \begin{cases} C_1, & \text{če } k(l_1 + l_2) + 1 \leq n \leq k(l_1 + l_2) + l_1, k \in \mathbb{N} \cup \{0\}, \\ C_2, & \text{če } k(l_1 + l_2) + l_1 + 1 \leq n \leq (k+1)(l_1 + l_2), k \in \mathbb{N} \cup \{0\}, \end{cases} \quad (4.4)$$

kjer sta  $C_1$  in  $C_2$  realni  $N \times N$  matriki, neodvisni od časa  $n$ .

Oglejmo si slučajni sprehod  $(X_0, X_1, \dots, X_{(l_1+l_2)k})$  dolžine  $(l_1 + l_2)k$  začeni v  $s_{j_0}$ . V ta namen uvedemo definicijo produkta dveh sprehodov. Za sprehoda

$$W^{\mathbf{P}_1} = s_{j_0} s_{j_1} \dots s_{j_{l_1}} \quad \text{in} \quad W^{\mathbf{P}_2} = s_{j_{l_1}} s_{j_{l_1+1}} \dots s_{j_{l_1+l_2}}$$

je njun produkt definiran kot

$$W^{\mathbf{P}_1} W^{\mathbf{P}_2} = s_{j_0} \dots s_{j_{l_1}} s_{j_{l_1}} \dots s_{j_{l_1+l_2}}.$$

Potem je verjetnost  $p_{W^{\mathbf{P}_1} W^{\mathbf{P}_2}}$  sprehoda  $W^{\mathbf{P}_1} W^{\mathbf{P}_2}$  enaka

$$p_{W^{\mathbf{P}_1} W^{\mathbf{P}_2}} = p_{W^{\mathbf{P}_1}} \cdot p_{W^{\mathbf{P}_2}},$$

medtem ko je cena  $c_{W^{\mathbf{P}_1} W^{\mathbf{P}_2}}$  vzdolž sprehoda  $W^{\mathbf{P}_1} W^{\mathbf{P}_2}$  enaka

$$c_{W^{\mathbf{P}_1} W^{\mathbf{P}_2}} = c_{W^{\mathbf{P}_1}} + c_{W^{\mathbf{P}_2}},$$

kjer sta ceni  $c_{W^{\mathbf{P}_1}}$  in  $c_{W^{\mathbf{P}_2}}$  zaporedoma poračunani glede na cenovno matriko  $C_1$  oziroma  $C_2$ . Poljuben sprehod

$$W^{\mathbf{P}_1, \mathbf{P}_2} = s_{j_0} s_{j_1} \cdots s_{j_{(l_1+l_2)k}} \in S_{j_0}^{(l_1+l_2)k}$$

lahko potem zapišemo kot produkt sprehodov

$$W^{\mathbf{P}_1, \mathbf{P}_2} = \prod_{j=1}^k W_{2j-1}^{\mathbf{P}_1} W_{2j}^{\mathbf{P}_2}. \quad (4.5)$$

Verjetnost  $p_{W^{\mathbf{P}_1, \mathbf{P}_2}}$  sprehoda  $W^{\mathbf{P}_1, \mathbf{P}_2}$  je enaka

$$p_{W^{\mathbf{P}_1, \mathbf{P}_2}} = \prod_{j=1}^k p_{W_{2j-1}^{\mathbf{P}_1}} \cdot p_{W_{2j}^{\mathbf{P}_2}},$$

njegova cena pa

$$c_{W^{\mathbf{P}_1, \mathbf{P}_2}} = \sum_{j=1}^k (c_{W_{2j-1}^{\mathbf{P}_1}} + c_{W_{2j}^{\mathbf{P}_2}}).$$

Zanima nas matematično upanje cene slučajnega sprehoda  $(X_0, X_1, \dots, X_{(l_1+l_2)k})$  dolžine  $(l_1 + l_2)k$  začenši v  $s_{j_0}$ :

$$E(c(X_0, X_1, \dots, X_{(l_1+l_2)k})) = \sum_{W^{\mathbf{P}_1, \mathbf{P}_2} \in S_{j_0}^{(l_1+l_2)k}} c_{W^{\mathbf{P}_1, \mathbf{P}_2}} p_{W^{\mathbf{P}_1, \mathbf{P}_2}}. \quad (4.6)$$

**Opomba 4.1.3.** Problem lahko na očiten način posplošimo na končno mnogo matrik.

## 4.2 Reševanje z metodo Monte Carlo

Osredotočimo se najprej na osnovni problem, kjer je markovska veriga homogena, podana s prehodno matriko  $\mathbf{P}$ . Kot smo omenili v opombi 4.1.1, analitičen izračun matematičnega upanja v (4.2) praktično pogosto ni možen. Lahko pa poiščemo dobro oceno z metodo Monte Carlo, ki smo jo spoznali v tretjem poglavju. Vse, kar moramo narediti, je, da generiramo dovolj velik vzorec slučajnih sprehodov  $W^{\mathbf{P}}$  dolžine  $k$  z začetkom v  $s_{j_0}$  in za oceno vzamemo vzorčno povprečje cen. Za kasnejšo referenco formalno podajmo algoritem `CenaSprehoda( $\mathbf{P}, C, s_{j_0}, k$ )`, ki generira slučajni sprehod (glede na prehodno matriko  $\mathbf{P}$ ) dolžine  $k$  z začetkom v  $s_{j_0}$  in vrne njegovo ceno (glede na matriko  $C$ ) skupaj s končnim stanjem sprehoda (v tem trenutku informacije o končnem stanju ne potrebujemo, jo pa bomo v nadaljevanju pri reševanju posplošenega problema).

Podajmo še formalno osnovni algoritem `PovprečjeHomogene( $\mathbf{P}, C, s_{j_0}, k, M$ )`, ki generira vzorec  $M$  slučajnih sprehodov (glede na prehodno matriko  $\mathbf{P}$ ) dolžine  $k$  z začetkom v  $s_{j_0}$  in vrne povprečje njihovih cen  $\hat{\mu}$  (glede na cenovno matriko  $C$ ).

---

**Algoritem** CenaSprehoda( $\mathbf{P}$ ,  $C$ ,  $s_{j_0}$ ,  $k$ )

---

**Vhod:** prehodna matrika  $\mathbf{P} \in \mathbb{R}^{N \times N}$ ,  
 cenovna matrika  $C \in \mathbb{R}^{N \times N}$ ,  
 začetno stanje  $s_{j_0}$ ,

dolžina  $k \in \mathbb{N}$  slučajnega sprehoda

**Izhod:** cena  $c \in \mathbb{R}$  vzdolž generiranega slučajnega sprehoda dolžine  $k$  začenši v  $s_{j_0}$ ,  
 končno stanje  $s_{j_k}$  sprehoda

$Q :=$  kumulativna matrika glede na  $\mathbf{P}$ ;

$c := 0$ ,  $i := j_0$ ;

**for**  $n = 1$  to  $k$  **do**

$U :=$  slučajno število na  $[0, 1]$ ;

$a := 1$ ,  $b := N$ ;

**while**  $a < b - 1$  **do**

$d :=$  celo število, ki je najbližje  $\frac{a+b}{2}$ ;

**if**  $Q[i, d] < U$  **then**

$a := d$ ;

**else**

$b := d$ ;

**end if**

**end while**

$c := c + C(i, b)$ ;

$i := b$ ;

**end for**

$j_k := i$ ;

**return**  $c$ ,  $s_{j_k}$ ;

---



---

**Algoritem** PovprečjeHomogene( $\mathbf{P}$ ,  $C$ ,  $s_{j_0}$ ,  $k$ ,  $M$ )

---

**Vhod:** prehodna matrika  $\mathbf{P} \in \mathbb{R}^{N \times N}$ ,  
 cenovna matrika  $C \in \mathbb{R}^{N \times N}$ ,

začetno stanje  $s_{j_0}$ ,

dolžina  $k \in \mathbb{N}$  slučajnega sprehoda,

velikost  $M$  vzorca

**Izhod:** vzorčno povprečje cen  $\hat{\mu}$

$\hat{\mu} := 0$ ;

**for**  $j = 1$  to  $M$  **do**

$c, \_ :=$  CenaSprehoda( $\mathbf{P}$ ,  $C$ ,  $s_{j_0}$ ,  $k$ );

$\hat{\mu} := \hat{\mu} + c$ ;

**end for**

$\hat{\mu} := \hat{\mu}/M$ ;

**return**  $\hat{\mu}$ ;

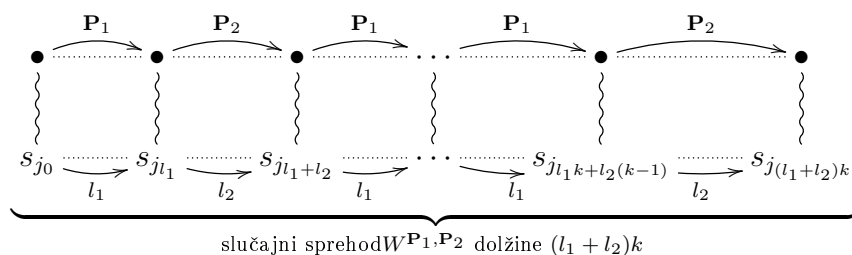
---

### 4.2.1 Reševanje posplošenega problema

Ker je homogena markovska veriga poseben primer nehomogene markovske verige, je posplošeni problem vsaj tako težak kot osnovni problem, zato se bomo tudi v tem primeru zadovoljili že z dobro oceno matematičnega upanja v (4.6). V nadaljevanju predlagamo tri pristope, ki temeljijo na metodi Monte Carlo. Vse tri metode na nek način uporabljajo algoritem CenaSprehoda. Čeprav so zaradi enostavnosti vsi pristopi predstavljeni za dve prehodni oziroma cenovni matriki, pa jih lahko posplošimo za končno mnogo matrik.

#### Prvi pristop

Sledimo osnovni ideji: generiramo vzorec slučajnih sprehodov  $W^{\mathbf{P}_1, \mathbf{P}_2}$  dolžine  $(l_1 + l_2)k$  z začetkom v  $s_{j_0}$  kot v (4.5) in za oceno zopet vzamemo vzorčno povprečje cen. Sprehode  $W^{\mathbf{P}_1, \mathbf{P}_2}$  skupaj z njihovimi cenami generiramo tako, da izmenično uporabljamo algoritem CenaSprehoda za prehodni matriki  $\mathbf{P}_1$ ,  $\mathbf{P}_2$  in cenovni matriki  $C_1, C_2$ . Grafično lahko generiranje sprehoda  $W^{\mathbf{P}_1, \mathbf{P}_2}$  ponazorimo z naslednjim diagramom:



Podajmo še formalni algoritem `PovprečjeNehomogene1( $\mathbf{P}_1, \mathbf{P}_2, C_1, C_2, s_{j_0}, l_1, l_2, k, M$ )`, ki generira vzorec  $M$  slučajnih sprehodov (glede na prehodni matriki  $\mathbf{P}_1$  in  $\mathbf{P}_2$ ) dolžine  $(l_1 + l_2)k$  z začetkom v  $s_{j_0}$  in vrne povprečje njihovih cen  $\hat{\mu}$  (glede na cenovni matriki  $C_1$  in  $C_2$ ).

#### Drugi pristop

Za trenutek se ustavimo pri osnovnem problemu. Naj bo markovska veriga homogena, podana s prehodno matriko  $\mathbf{P}$ , radi pa bi izračunali matematično upanje cene (glede na matriko  $C$ , ki določa ceno  $c$ ) slučajnega sprehoda  $(X_0, X_1, \dots, X_k)$  dolžine  $k$  z začetkom v  $s_{j_0}$  kot v (4.2). Pokažimo, da lahko vso potrebno informacijo za izračun zakodiramo na slučajnem sprehodu  $(Y_0, Y_1)$  dolžine 1 začenši v  $s_{j_0}$  in sicer s prehodno matriko  $\mathbf{P}^k$  ter primerno cenovno matriko  $C'$ , ki določa ceno  $c'$  na  $(Y_0, Y_1)$ . Kaj moramo vzeti za elemente matrike  $C'$ , da bo veljalo

$$E(c'(Y_0, Y_1)) = E(c(X_1, X_2, \dots, X_k))?$$

Ker nas (trenutno) zanimajo samo sprehodi, ki se začnejo v  $s_{j_0}$ , si pogledjmo, kako izgleda  $j_0$ -ta vrstica matrike  $C'$ . Odgovor nam da naslednja trditev.

---

**Algoritem** PovprečjeNehomogene1( $\mathbf{P}_1, \mathbf{P}_2, C_1, C_2, s_{j_0}, l_1, l_2, k, M$ )

---

**Vhod:** prehodni matriki  $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{N \times N}$ ,  
 cenovni matriki  $C_1, C_2 \in \mathbb{R}^{N \times N}$ ,  
 začetno stanje  $s_{j_0}$ ,  
 $l_1, l_2, k \in \mathbb{N}$ ,  
 velikost  $M$  vzorca

**Izhod:** vzorčno povprečje cen  $\hat{\mu}$

```

 $\hat{\mu} := 0;$ 
 $začetek := j_0;$ 
for  $j = 1$  to  $M$  do
  for  $n = 1$  to  $k$  do
     $c, konec := \text{CenaSprehoda}(\mathbf{P}_1, C_1, začetek, l_1);$ 
     $\hat{\mu} := \hat{\mu} + c, začetek := konec;$ 
     $c, konec := \text{CenaSprehoda}(\mathbf{P}_2, C_2, začetek, l_2);$ 
     $\hat{\mu} := \hat{\mu} + c, začetek := konec;$ 
  end for
end for
 $\hat{\mu} := \hat{\mu}/M;$ 
return  $\hat{\mu};$ 

```

---

**Trditev 4.2.1.** Pri zgornji notaciji in predpostavkah je vrednost  $c'_{j_0j}$  na  $(j_0, j)$ -tem mestu matrike  $C'$  enaka pogojnemu matematičnemu upanju

$$c'_{j_0j} = E(c(X_0, \dots, X_k) | X_k = s_j) \quad (4.7)$$

slučajne spremenljivke  $c(X_0, X_1, \dots, X_k)$  pri pogoju, da smo končali v  $s_j$ .

**Dokaz:** Pokazati moramo, da je  $E(c'(Y_0, Y_1)) = E(c(X_1, X_2, \dots, X_k))$ . Uporabimo Bayesovo formulo na matematičnem upanju in računajmo:

$$\begin{aligned}
 E(c'(Y_0, Y_1)) &= \sum_{j=1}^N E(c'(Y_0, Y_1) | Y_1 = s_j) P(Y_1 = s_j) \\
 &= \sum_{j=1}^N c'(Y_0 = s_{j_0}, Y_1 = s_j) P(Y_0 = s_{j_0} | Y_1 = s_j) P(Y_1 = s_j) \\
 &= \sum_{j=1}^N c'_{j_0j} \frac{P(Y_1 = s_j | Y_0 = s_{j_0}) P(Y_0 = s_{j_0})}{P(Y_1 = s_j)} P(Y_1 = s_j) \\
 &= \sum_{j=1}^N c'_{j_0j} P(Y_1 = s_j | Y_0 = s_{j_0}) \quad (4.8)
 \end{aligned}$$

Opazimo, da velja

$$P(Y_1 = s_j | Y_0 = s_{j_0}) = \mathbf{P}_{j_0j}^k = P(X_k = s_j), \quad (4.9)$$

kjer smo v zadnji enakosti uporabili, da je  $P(X_0 = s_{j_0}) = 1$ . Upoštevajmo (4.9) skupaj s predpostavko (4.7) v (4.8) in dobimo

$$\begin{aligned} E(c'(Y_0, Y_1)) &= \sum_{j=1}^N c'_{j_0j} P(Y_1 = s_j | Y_0 = s_{j_0}) \\ &= \sum_{j=1}^N E(c(X_0, \dots, X_k) | X_k = s_j) P(X_k = s_j) \\ &= E(c(X_1, X_2, \dots, X_k)). \end{aligned}$$

S tem je trditev dokazana. ■

**Opomba 4.2.2.** Za izračun poljubne vrstice  $i$  matrike  $C'$  uporabimo isti recept s to spremembo, da je začetno stanje sprehodov enako  $s_i$ .

Naj  $S_{j_0j}^k$  označuje množico vseh sprehodov dolžine  $k$  z začetkom v  $s_{j_0}$  in koncem v  $s_j$ . Elementaren izračun pokaže, da lahko vrednost  $c'_{j_0j}$  v (4.7) zapišemo v obliki

$$c'_{j_0j} = \frac{\sum_{W \in S_{j_0j}^k} c_{W\mathbf{P}} p_{W\mathbf{P}}}{\sum_{W \in S_{j_0j}^k} p_{W\mathbf{P}}}. \quad (4.10)$$

V praksi to pomeni, da moramo za izračun vrednosti  $c'_{j_0j}$  zgraditi množico  $S_{j_0j}^k$  vseh sprehodov dolžine  $k$ , ki začnejo v  $s_{j_0}$  in končajo v  $s_j$ . Torej moramo za izračun cele  $j_0$ -te vrstice matrike  $C'$  zgraditi natanko množico  $S_{j_0}^k$  vseh sprehodov dolžine  $k$  z začetkom v  $s_{j_0}$ . Spomnimo, da je to natanko množica, ki nastopa v osnovnem problemu (4.2). Kot smo že omenili, je analitično reševanje osnovnega problema v praksi težko, če sta parametra  $N$  in še posebej  $k$  velika. Če pa sta parametra dovolj majhna, lahko problem rešimo analitično. V ta namen formalno podajmo algoritem GradnjaSprehodov( $\mathbf{P}$ ,  $s_{j_0}$ ,  $k$ ), ki vrne množico  $S_{j_0}^k$ . Bralca opozorimo, da je slednji algoritem modifikacija algoritma za iskanje v širino [13].

Ko imamo množico  $S_{j_0}^k$  zgrajeno, lahko izračunamo vrednosti  $c'_{j_0j}$  matrike  $C'$  za  $j_0$ -to vrstico. Množico  $S_{j_0}^k$  enostavno razbijemo na disjunktno unijo

$$S_{j_0}^k = \bigcup_{j=1}^N S_{j_0j}^k$$

množic  $S_{j_0j}^k$ . Nato za vsako množico  $S_{j_0j}^k$  posebej uporabimo formulo v (4.10) in izračunamo vrednosti  $c'_{j_0j}$ . Pozoren bralec bo opazil, da velja še

$$\mathbf{P}_{j_0j}^k = \sum_{W \in S_{j_0j}^k} p_{W\mathbf{P}}.$$

To pomeni, da matrike  $\mathbf{P}^k$  ni potrebno posebej računati, ampak jo dobimo za-stonj med samo gradnjo matrike  $C'$ . Za nadaljnjo referenco podajmo še algoritem GradnjaMatrik( $\mathbf{P}$ ,  $C$ ,  $k$ ), ki z uporabo algoritma GradnjaSprehodov zgradi matriki  $\mathbf{P}^k$  in  $C'$ . Na tem mestu opozorimo, da je v praksi gradnja matrike  $C'$  tako težka, kot je reševanje  $N$  osnovnih problemov.



---

**Algoritem** GradnjaSprehodov( $\mathbf{P}, s_{j_0}, k$ )

---

**Vhod:** prehodna matrika  $\mathbf{P} \in \mathbb{R}^{N \times N}$ ,  
začetno stanje  $s_{j_0}$ ,  
dolžina  $k \in \mathbb{N}$  slučajnega sprehoda

**Izhod:** množica  $S_{j_0}^k$

$S_{j_0}^k := \{\}$ ;

$S :=$  prazna vrsta;

Vstavi ( $s_{j_0}$ ) v vrsto  $S$ ;

**while**  $S$  ni prazna **do**

    Vzemi prvi element  $W$  iz vrste  $S$ ;

    Recimo, da je  $W = (s_{j_0}, s_{j_1}, \dots, s_{j_m})$ ;

**if**  $m = k$  **then**

$S_{j_0}^k := S_{j_0}^k \cup \{W\}$ ;

**else**

**for**  $i = 1$  to  $N$  **do**

**if**  $\mathbf{P}[m, i] > 0$  **then**

                Vstavi ( $s_{j_0}, s_{j_1}, \dots, s_{j_m}, s_i$ ) v vrsto  $S$ ;

**end if**

**end for**

**end if**

**end while**

**return**  $S_{j_0}^k$ ;

---



---

**Algoritem** GradnjaMatrik( $\mathbf{P}, C, k$ )

---

**Vhod:** prehodna matrika  $\mathbf{P} \in \mathbb{R}^{N \times N}$ ,  
cenovna matrika  $C \in \mathbb{R}^{N \times N}$ ,  
dolžina  $k \in \mathbb{N}$  slučajnega sprehoda

**Izhod:** prehodna matrika  $\mathbf{P}^k$ , cenovna matrika  $C'$

$\mathbf{P}^k :=$  ničelna matrika dimenzije  $N \times N$ ;

$C' :=$  ničelna matrika dimenzije  $N \times N$ ;

**for**  $i = 1$  to  $N$  **do**

$S_i^k :=$  GradnjaSprehodov( $\mathbf{P}, s_i, k$ );

**for**  $W \in S_i^k$  **do**

        Recimo, da je  $W = (s_i, s_{j_1}, \dots, s_{j_{k-1}}, s_j)$ ;

$p_{W\mathbf{P}} := p_{s_i s_{j_1}} \cdots p_{s_{j_{k-1}} s_j}$ ,  $c_{W\mathbf{P}} := c_{s_i s_{j_1}} + \cdots + c_{s_{j_{k-1}} s_j}$ ;

$\mathbf{P}^k[i, j] := \mathbf{P}^k[i, j] + p_{W\mathbf{P}}$ ,  $C'[i, j] := C'[i, j] + c_{W\mathbf{P}} p_{W\mathbf{P}}$ ;

**end for**

**for**  $j = 1$  to  $N$  **do**

$C'[i, j] := C'[i, j] / \mathbf{P}^k[i, j]$ ;

**end for**

**end for**

**return**  $\mathbf{P}^k, C'$ ;

---

**Opomba 4.2.3.** Algoritma GradnjaSprehodov in GradnjaMatrik lahko uporabimo tudi v primeru, ko sta matriki  $\mathbf{P}$  in  $C$  odvisni od časa  $n$ . V algoritmu enostavno zamenjamo  $\mathbf{P}$  in  $C$  zaporedoma s  $\mathbf{P}(n)$  in  $C(n)$ . To dejstvo bomo izkoristili v tretjem pristopu.

**Opomba 4.2.4.** Če  $C'$  ne moremo izračunati analitično, potem lahko poiščemo njen približek recimo tako, da poiščemo  $N$  ocen za osnovni problem z metodo Monte Carlo. Natančneje, za vsako vrstico  $i, i = 1, 2, \dots, N$ , z uporabo algoritma CenaSprehoda( $\mathbf{P}, C, s_i, k$ ) generiramo na primer  $M$  slučajnih sprehodov dolžine  $k$  z začetkom v  $s_i$ , nato jih uredimo v  $N$  skupin  $\mathcal{S}_j$  glede na končna stanja  $s_j, j = 1, 2, \dots, N$ , in definiramo

$$c'_{ij} = \frac{\sum_{W \in \mathcal{S}_j} c_{WP}}{|\mathcal{S}_j|}.$$

Podajmo formalni algoritem PribližekCenovne( $\mathbf{P}, C, k, M$ ), ki za računanje vsake vrstice  $M$ -krat uporabi algoritem CenaSprehoda, in vrne približek matrike  $C'$ . V tem primeru matriko  $\mathbf{P}^k$  izračunamo posebej.

---

**Algoritem** PribližekCenovne( $\mathbf{P}, C, k, M$ )

---

**Vhod:** prehodna matrika  $\mathbf{P} \in \mathbb{R}^{N \times N}$ ,  
 cenovna matrika  $C \in \mathbb{R}^{N \times N}$ ,  
 dolžina  $k \in \mathbb{N}$  slučajnega sprehoda,  
 število ponovitev  $M$

**Izhod:** približek cenovne matrike  $C'$

$C' :=$  ničelna matrika dimenzije  $N \times N$ ;

$V :=$  ničelna matrika dimenzije  $N \times N$ ;

**for**  $i = 1$  to  $N$  **do**

**for**  $j = 1$  to  $M$  **do**

$c, konec :=$  CenaSprehoda( $\mathbf{P}, C, s_i, k$ );

$C'[i, konec] := C'[i, konec] + c; V[i, konec] := V[i, konec] + 1;$

**end for**

**end for**

**for**  $i, j \in \{1, 2, \dots, N\}$  **do**

**if**  $C'[i, j] \neq 0$  **then**

$C'[i, j] := C'[i, j]/V[i, j];$

**end if**

**end for**

**return**  $C'$ ;

---

Vrnimo se zdaj nazaj k reševanju posplošenega problema, kjer imamo nehomogeno markovsko verigo  $\{X_n\}_{n \in T}$ , podano s prehodnima matrikama  $\mathbf{P}_1$  in  $\mathbf{P}_2$  kot v (4.3), in pripadajoči cenovni matriki  $C_1$  in  $C_2$  kot v (4.4). Oglejmo si prvih  $l_1 + l_2$  korakov markovske verige  $\{X_n\}_{n \in T}$  z začetkom v  $s_{j_0}$ . Ideja je, da informacijo o prvih  $l_1$  korakih glede na  $\mathbf{P}_1$  in pripadajočo  $C_1$  zakodiramo s sprehodom dolžine 1 glede na

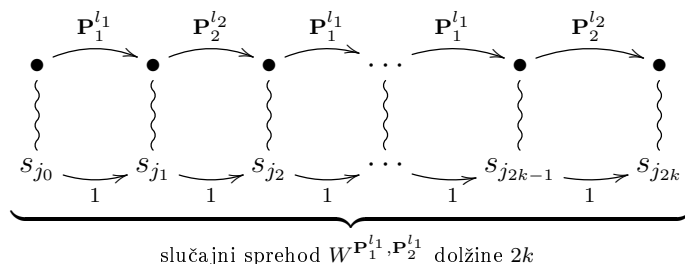
prehodno matriko  $\mathbf{P}_1^{l_1}$  in pripadajočo cenovno matriko  $C_{l_1}$ , kot smo opisali zgoraj. Podobno lahko informacijo o naslednjih  $l_2$  korakih glede na  $\mathbf{P}_2$  in  $C_2$  zakodiramo s prehodno matriko  $\mathbf{P}_2^{l_2}$  in cenovno matriko  $C_{l_2}$ . Tako verigo  $\{X_n\}_{n \in T}$  opišemo z nehomogeno markovsko verigo  $\{Y_n\}_{n \in T}$  s prehodno matriko

$$\mathbf{P}'(n) = \begin{cases} \mathbf{P}_1^{l_1}, & \text{če } n = 2k - 1, k \in \mathbb{N}, \\ \mathbf{P}_2^{l_2}, & \text{če } n = 2k, k \in \mathbb{N}, \end{cases} \quad (4.11)$$

ceno  $c'$  na slučajnem sprehodu  $(Y_0, Y_1, \dots, Y_{2k})$  pa s cenovno matriko

$$C'(n) = \begin{cases} C_{l_1}, & \text{če } n = 2k - 1, k \in \mathbb{N}, \\ C_{l_2}, & \text{če } n = 2k, k \in \mathbb{N}. \end{cases} \quad (4.12)$$

Sedaj iščemo oceno za matematično upanje  $E(c'(Y_0, Y_1, \dots, Y_{2k}))$ , ki je jasno zelo poseben primer posplošenega problema, ko je  $l_1 = l_2 = 1$ . Od tu naprej nadaljujemo z reševanjem kot v prvem pristopu. Grafično lahko generiranje sprehoda  $W^{\mathbf{P}_1^{l_1}, \mathbf{P}_2^{l_2}}$  dolžine  $2k$  ponazorimo z naslednjim diagramom:



Podajmo še formalni algoritem `PovprečjeNehomogene2`( $\mathbf{P}_1, \mathbf{P}_2, C_1, C_2, s_{j_0}, l_1, l_2, k, M$ ), ki zgradi prehodni matriki  $\mathbf{P}_1^{l_1}, \mathbf{P}_2^{l_2}$  kot v (4.11) ter cenovni matriki  $C_{l_1}, C_{l_2}$  kot v (4.12), generira vzorec  $M$  slučajnih sprehodov (glede na  $\mathbf{P}_1^{l_1}, \mathbf{P}_2^{l_2}$ ) dolžine  $2k$  z začetkom v  $s_{j_0}$  in vrne povprečje njihovih cen  $\hat{\mu}$  (glede na  $C_{l_1}, C_{l_2}$ ).

---

**Algoritem** `PovprečjeNehomogene2`( $\mathbf{P}_1, \mathbf{P}_2, C_1, C_2, s_{j_0}, l_1, l_2, k, M$ )

---

**Vhod:** prehodni matriki  $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{N \times N}$ ,  
 cenovni matriki  $C_1, C_2 \in \mathbb{R}^{N \times N}$ ,  
 začetno stanje  $s_{j_0}$ ,  
 $l_1, l_2, k \in \mathbb{N}$ ,  
 velikost  $M$  vzorca

**Izhod:** vzorčno povprečje cen  $\hat{\mu}$

$\mathbf{P}_1^{l_1}, C_{l_1} := \text{GradnjaMatrik}(\mathbf{P}_1, C_1, l_1)$ ;

$\mathbf{P}_2^{l_2}, C_{l_2} := \text{GradnjaMatrik}(\mathbf{P}_2, C_2, l_2)$ ;

**return**  $\hat{\mu} := \text{PovprečjeNehomogene1}(\mathbf{P}_1^{l_1}, \mathbf{P}_2^{l_2}, C_{l_1}, C_{l_2}, s_{j_0}, 1, 1, k, M)$ ;

---

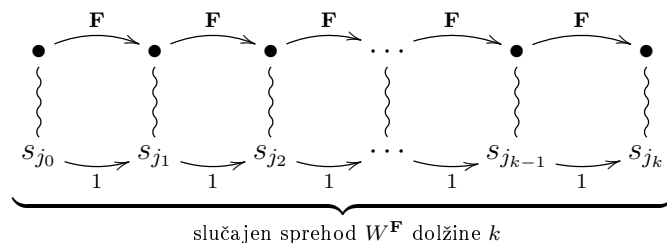
### Tretji pristop

Ta pristop se od drugega razlikuje v načinu računanja ocene za matematično upanje  $E(c'(Y_0, Y_1, \dots, Y_{2k}))$ . Če smo pri drugem pristopu to oceno izračunali neposredno z

uporabo prvega pristopa, pa tu ponovno uporabimo idejo o kodiranju informacije, ki smo jo uporabili v prvem delu drugega pristopa.

Kot v drugem pristopu najprej izračunamo matriki  $\mathbf{F}_1 = \mathbf{P}_1^{l_1}$  in  $\mathbf{F}_2 = \mathbf{P}_2^{l_2}$ , ki določata prehodno matriko  $\mathbf{P}'(n)$  markovske verige  $\{Y_n\}_{n \in T}$ , ter matriki  $C_{l_1}$  in  $C_{l_2}$ , ki določata cenovno matriko  $C'(n)$  in ceno  $c'$  na prehodu  $(Y_0, Y_1, \dots, Y_{2k})$ . Sedaj si oglejmo prva dva koraka verige  $\{Y_n\}_{n \in T}$  z začetkom v  $s_{j_0}$ . Podobno kot prej lahko vso potrebno informacijo o prvih dveh korakih zakodiramo na slučajnem prehodu  $(Z_0, Z_1)$  dolžine 1 začenši v  $s_{j_0}$ , in sicer s prehodno matriko  $\mathbf{F} = \mathbf{F}_1 \cdot \mathbf{F}_2$  ter cenovno matriko  $C''$ , ki določa ceno  $c''$  na  $(Z_0, Z_1)$ . Cenovno matriko  $C''$  skupaj s prehodno matriko  $\mathbf{F}$  zopet zgradimo z algoritmom  $\text{GradnjaMatrik}(\mathbf{P}'(n), C'(n), 2)$ .

Verigo  $\{Y_n\}_{n \in T}$  tako opišemo s homogeno markovsko verigo  $\{Z_n\}_{n \in T}$  s prehodno matriko  $\mathbf{F}$ , ceno  $c''$  na slučajnem prehodu  $(Z_1, Z_2, \dots, Z_k)$  pa s cenovno matriko  $C''$ . Sedaj iščemo oceno za matematično upanje  $E(c''(Z_0, Z_1, \dots, Z_k))$ , ki je očitno primer osnovnega problema. Torej smo posplošeni problem prevedli na osnovnega. Grafično lahko generiranje prehoda  $W^{\mathbf{F}}$  dolžine  $k$  ponazorimo z naslednjim diagramom:



Podajmo še formalni algoritem  $\text{PovprečjeNehomogene3}(\mathbf{P}_1, \mathbf{P}_2, C_1, C_2, s_{j_0}, l_1, l_2, k, M)$ , ki najprej zgradi prehodni matriki  $\mathbf{P}_1^{l_1}, \mathbf{P}_2^{l_2}$  kot v (4.11) ter cenovni matriki  $C_{l_1}, C_{l_2}$  kot v (4.12), nato zgradi prehodno matriko  $\mathbf{F}$  in cenovno matriko  $C''$  ter generira vzorec  $M$  slučajnih sprehodov (glede na  $\mathbf{F}$ ) dolžine  $k$  z začetkom v  $s_{j_0}$  in vrne povprečje njihovih cen  $\hat{\mu}$  (glede na  $C''$ ).

---

**Algoritem**  $\text{PovprečjeNehomogene3}(\mathbf{P}_1, \mathbf{P}_2, C_1, C_2, s_{j_0}, l_1, l_2, k, M)$

---

**Vhod:** prehodni matriki  $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{N \times N}$ ,  
 cenovni matriki  $C_1, C_2 \in \mathbb{R}^{N \times N}$ ,  
 začetno stanje  $s_{j_0}$ ,  
 $l_1, l_2, k \in \mathbb{N}$ ,  
 velikost  $M$  vzorca

**Izhod:** vzorčno povprečje cen  $\hat{\mu}$

$\mathbf{P}_1^{l_1}, C_{l_1} := \text{GradnjaMatrik}(\mathbf{P}_1, C_1, l_1)$ ;  
 $\mathbf{P}_2^{l_2}, C_{l_2} := \text{GradnjaMatrik}(\mathbf{P}_2, C_2, l_2)$ ;  
 $\mathbf{P}'(n) :=$  prehodna matrika, določena z matrikama  $\mathbf{P}_1^{l_1}$  in  $\mathbf{P}_2^{l_2}$  kot v (4.11);  
 $C'(n) :=$  cenovna matrika, določena z matrikama  $C_{l_1}$  in  $C_{l_2}$  kot v (4.12);  
 $\mathbf{F}, C'' := \text{GradnjaMatrik}(\mathbf{P}'(n), C'(n), 2)$ ;  
**return**  $\hat{\mu} := \text{PovprečjeHomogene}(\mathbf{F}, C'', s_{j_0}, k, M)$ ;

---

**Opomba 4.2.5.** Če sta parametra  $l_1$  in  $l_2$  prevelika, lahko izračunamo približek cenovnih matrik  $C_{l_1}$  in  $C_{l_2}$  kot v drugem pristopu. Isto idejo lahko uporabimo tudi pri cenovni matriki  $C''$ , če je analitično računanje prezahtevno.

### 4.3 Pristop, ki uporablja difuzijske valčke

Oceno matematičnega upanja  $E(c(X_1, X_2, \dots, X_k))$  v osnovnem problemu lahko poiščemo tudi s pristopom, ki so ga predlagali v [19] in temelji na uporabi difuzijskih valčkov za izračun  $n$ -te potence prehodne matrike markovske verige [11]. V tem razdelku bomo predstavili ta pristop.

Naj bo  $f: \mathbb{R} \rightarrow \mathbb{R}$  funkcija. V [19] predlagajo oceno za

$$E(f(c(X_1, X_2, \dots, X_k))) = \sum_{j_1, j_2, \dots, j_k \in S} f(c_{j_0 j_1} + \dots + c_{j_{k-1} j_k}) p_{j_0 j_1} \dots p_{j_{k-1} j_k}. \quad (4.13)$$

**Opomba 4.3.1.** Oceno za matematično upanje  $E(c(X_1, X_2, \dots, X_k))$  dobimo, če v (4.13) za  $f$  vzamemo identično funkcijo  $f(x) = x$ .

Prva ideja je, da vsoto v (4.13) zapišemo v integralsko obliko. Naj bo  $P_C(t)$   $n \times n$  matrika, ki je odvisna od realne spremenljivke  $t$ , njeni elementi pa naj bodo oblike  $e^{i c_{ij} t} p_{ij}$ . Nadalje, naj  $\mathbf{1}$  označuje  $n \times 1$  vektor enic in naj  $e_{j_0}^T$  označuje  $1 \times n$  enotski vektor z enico na  $j_0$ -ti komponenti. Opazimo, da je  $(u, v)$ -ti element  $n \times n$  matrike  $[P_C(t)]^k$  oblike

$$\sum_{j_1, j_2, \dots, j_{k-1}} e^{i(c_{uj_1} + c_{j_1 j_2} + \dots + c_{j_{k-1} v})t} p_{uj_1} p_{j_1 j_2} \dots p_{j_{k-1} v}.$$

Elementarni račun pokaže, da je produkt  $e_{j_0}^T [P_C(t)]^k \mathbf{1}$  enak

$$e_{j_0}^T [P_C(t)]^k \mathbf{1} = \sum_{j_1, j_2, \dots, j_{k-1}, j_k \in S} e^{i(c_{j_0 j_1} + c_{j_1 j_2} + \dots + c_{j_{k-1} j_k})t} p_{j_0 j_1} \dots p_{j_{k-1} j_k}. \quad (4.14)$$

Uporabimo neunitarno inverzno Furierovo transformacijo

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(t) e^{ixt} dt,$$

kjer je  $\hat{f}(x) = \int_{-\infty}^{\infty} \hat{f}(t) e^{-itx} dx$ . Upoštevajmo še (4.14) in računajmo

$$\begin{aligned} E(f(X)) &= \sum_{j_1, j_2, \dots, j_k \in S} f(c_{j_0 j_1} + c_{j_1 j_2} + \dots + c_{j_{k-1} j_k}) p_{j_0 j_1} p_{j_1 j_2} \dots p_{j_{k-1} j_k} \\ &= \sum_{j_1, j_2, \dots, j_k \in S} \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(t) e^{i(c_{j_0 j_1} + \dots + c_{j_{k-1} j_k})t} dt \right) p_{j_0 j_1} \dots p_{j_{k-1} j_k} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(t) \sum_{j_1, j_2, \dots, j_k \in S} e^{i(c_{j_0 j_1} + \dots + c_{j_{k-1} j_k})t} p_{j_0 j_1} \dots p_{j_{k-1} j_k} dt \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(t) e_{j_0}^T [P_C(t)]^k \mathbf{1} dt \\ &= \frac{1}{2\pi} \cdot e_{j_0}^T \cdot \int_{-\infty}^{\infty} \hat{f}(t) [P_C(t)]^k \mathbf{1} dt. \end{aligned}$$

Zdaj, ko imamo matematično upanje v (4.13) zapisano v integralski obliki, v članku [19] predlagajo, da diskretiziramo integral z neko napako in nato vsak člen diskretizacije, zopet z neko napako, izračunamo z metodo difuzijskih valčkov.

Kot smo omenili v opombi 4.3.1, osnovni problem dobimo, če za funkcijo  $f$  izberemo  $f(x) = x$ . Pripadajoča Furierova transformacija funkcije  $f$  je v tem primeru enaka

$$\hat{f}(t) = 2\pi i \delta'(t),$$

kjer je  $\delta'$  prvi odvod Diracove delta funkcije. Ker Diracova delta funkcija zavzame vrednost nič povsod razen v 0, je diskretizacija trivialna. Z uporabo enakosti

$$\int_{-\infty}^{\infty} \delta'(t)g(t)dt = -g'(0)$$

dobimo

$$\begin{aligned} E(X) &= \frac{1}{2\pi} e_{j_0}^T \int_{-\infty}^{\infty} 2\pi i \delta'(t) [P_C(t)]^k \mathbf{1} dt \\ &= i e_{j_0}^T \left( \int_{-\infty}^{\infty} \delta'(t) [P_C(t)]^k dt \right) \mathbf{1} \\ &= -i e_{j_0}^T ([P_C(t)]^k)'|_{t=0} \mathbf{1} \\ &= e_{j_0}^T \left[ \sum_{i=1}^k P_C(0)^{i-1} (DP_C(t))|_{t=0} P_C(0)^{k-i} \right] \mathbf{1}, \end{aligned} \quad (4.15)$$

kjer  $D$  ozančuje operator  $\frac{1}{i} \frac{d}{dt}$  z elementi matrike  $DP_C(t)$  oblike  $e^{ic_{ij}t} c_{ij} p_{ij}$ . V članku [19] za vrednost  $k$  predlagajo  $k = 2^p$  za  $p > 0$ , saj metoda difuzijskih valčkov najboljše deluje za potence števila dva. Potem lahko vsoto (4.15) zapišemo kot

$$E(X) = e_{j_0}^T \sum_{i=1}^{2^p-1} \left( P_C(0)^{i-1} DP_C(0) P_C(0)^{k-i} + P_C(0)^{k-i} DP_C(0) P_C(0)^{i-1} \right) \mathbf{1}. \quad (4.16)$$

Opozorimo, da je  $P_C(0)$  natanko prehodna matrika  $\mathbf{P}$ , matrika  $DP_C(0)$  pa je produkt po komponentah matrik  $C$  in  $\mathbf{P}$ .

Potence  $P_C(0)^j$  v zgornji vsoti (4.16) izračunamo na naslednji način. Najprej potenco  $j$  zapišemo kot vsoto potenc dvojke:  $j = 2^{p_1} + 2^{p_2} + \dots + 2^{p_r}$ . Vsako potenco dvojke matrike  $P_C(0)$  izračunamo z metodo difuzijskih valčkov kot je predlagano v [11]. Nazadnje  $P_C(0)^j$  izračunamo kot

$$P_C(0)^j = P_C(0)^{2^{p_1}} P_C(0)^{2^{p_2}} \dots P_C(0)^{2^{p_r}}.$$

Opazimo, da je  $2^p - 1$  največja potenco, ki nastopa v (4.16). Ker je

$$2^p - 1 = \sum_{i=0}^{p-1} 2^i,$$

je za izračun  $P_C(0)^{2^p-1}$  potrebno poznati vse matrike  $P_C(0)^{2^i}$ ,  $i = 1, 2, \dots, p-1$  (za  $i = 0$  matriko že poznamo). Zato najprej izračunamo vse potence dvojke in

---

**Algoritem** PovprečjeDifuzijskiValčki( $\mathbf{P}, C, s_{j_0}, p$ )
 

---

**Vhod:** prehodna matrika  $\mathbf{P} \in \mathbb{R}^{N \times N}$ ,  
 cenovna matrika  $C \in \mathbb{R}^{N \times N}$ ,  
 začetno stanje  $s_{j_0}$ ,  
 $p \in \mathbb{N}$

**Izhod:** ocena  $\tilde{\mu}$  za  $\mu$

$D :=$  produkt po komponentah matrik  $\mathbf{P}$  in  $C$ ;

$M :=$  ničelna matrika dimenzije  $N \times N$ ;

$\mathcal{P} :=$  prazen seznam dolžine  $p$ ;  $\mathcal{P}[0] := \mathbf{P}$ ;

$\mathbf{1} := n \times 1$  vektor enic;  $e_{j_0}^T := 1 \times n$  enotski vektor z enico na  $j_0$ -ti komponenti;

**for**  $i = 1$  to  $p - 1$  **do**

  Izračunaj  $\mathbf{P}^{2^i}$  z uporabo difuzijskih valčkov;

$\mathcal{P}[i] := \mathbf{P}^{2^i}$ ;

**end for**

**for**  $i = 1$  to  $2^{p-1}$  **do**

  Naj bo  $i - 1 = 2^{p_1} + 2^{p_2} + \dots + 2^{p_r}$ ;

$A := \mathcal{P}[p_1] \cdot \mathcal{P}[p_2] \cdot \dots \cdot \mathcal{P}[p_r]$ ;

  Naj bo  $2^p - i = 2^{p'_1} + 2^{p'_2} + \dots + 2^{p'_l}$ ;

$B := \mathcal{P}[p'_1] \cdot \mathcal{P}[p'_2] \cdot \dots \cdot \mathcal{P}[p'_l]$ ;

$M := M + ADB + BDA$ ;

**end for**

**return**  $\tilde{\mu} := e_{j_0}^T \cdot M \cdot \mathbf{1}$ ;

---

jih shranimo. S tem prihranimo pri času, ko računamo ostale potence. Združimo vse skupaj v algoritem PovprečjeDifuzijskiValčki( $\mathbf{P}, C, s_{j_0}, p$ ), ki z uporabo metode difuzijskih valčkov za računanje potenc vrne oceno  $\tilde{\mu}$  za matematično upanje  $\mu$ .

**Opomba 4.3.2.** Metoda difuzijskih valčkov za računanje potenc matrik se zadovoljivo obnaša v primeru, ko ima matrika nizek numerični rang. Razložimo bolj natančno, kaj to pomeni. Spomnimo se najprej, da je rang matrike  $A$  definiran kot število neodvisnih stolpcev v  $A$ . Toda ob prisotnosti napak – recimo pri aproksimaciji, diskreditizaciji ali zaokroževanju – ta definicija ni uporabna, ker bi bilo možno stolpce matrike  $A$ , ki so iz matematičnega vidika strogo neodvisni, iz praktičnega vidika smatrati kot skoraj linearno odvisne. Grobo rečeno je numerični rang enak številu stolpcev matrike  $A$ , ki so, glede na neko toleranco, praktično linearno neodvisni. Formalno je *numerični  $\delta$ -rang*  $r_\delta$  matrike  $A \in \mathbb{R}^{N \times N}$ , glede na dano toleranco  $\delta > 0$ , definiran kot

$$r_\delta = \min\{\text{rang}(B) \mid \|A - B\|_2 \leq \delta\},$$

kjer  $\|\cdot\|_2$  označuje spektralno oziroma 2-normo. Rečemo, da ima matrika  $A$  nizek numerični rang  $r_\delta$ , če velja  $r_\delta \ll N$ , to je, numerični rang  $r_\delta$  je precej manjši kot red  $N$  matrike  $A$ . Več o numeričnem rangju si lahko bralec prebere v [22].

Kot smo že omenili, lahko algoritem PovprečjeDifuzijskiValčki uporabimo samo za reševanje osnovnega problema, pa še to pod določenimi pogoji. Zato ga bomo v naslednjem poglavju primerjali le z algoritmom PovprečjeHomogene.





---

## Eksperimentalni rezultati

V tem poglavju ekperimentalno ovrednotimo predstavljene metode. Da poenostavimo zapis, algoritme PovprečjeHomogene, PovprečjeNehomogene1, PovprečjeNehomogene2 in PovprečjeNehomogene3 krajše zapišemo zaporedoma kot PH, PN1, PN2 in PN3. Poleg tega algortima GradnjaMatrik in PribližekCenovne krajše zapišemo zaporedoma kot GM in PC.

Pri algoritmu PN2 ločimo dve varianti glede na to, ali cenovni matriki  $C_{l_1}$  in  $C_{l_2}$  izračunamo z algoritmom GM ali pa uporabimo algoritem PC in poiščemo njuna približka:

- (i) PN2(GM) označuje, da smo v algoritmu PN2 uporabili algoritem GM,
- (ii) PN2(PC) označuje, da smo v algoritmu PN2 uporabili algoritem PC.

Podobno v algoritmu PN3 – poleg tega, da ločimo, ali smo za  $C_{l_1}$  in  $C_{l_2}$  uporabili algoritem GM oziroma PC – nadalje ločimo še dve varianti glede na to, ali cenovno matriko  $C''$  izračunamo z algoritmom GM ali pa uporabimo algoritem PC in poiščemo njuna približka. Skupaj dobimo štiri možne variante:

- (i) PN3(GM-GM) označuje, da smo v PN3 za  $C_{l_1}$ ,  $C_{l_2}$  in  $C''$  uporabili GM,
- (ii) PN3(GM-PC) označuje, da smo v PN3 za  $C_{l_1}$ ,  $C_{l_2}$  uporabili GM, za  $C''$  pa PC,
- (iii) PN3(PC-PC) označuje, da smo v PN3 za  $C_{l_1}$ ,  $C_{l_2}$  in  $C''$  uporabili PC,
- (iv) PN3(PC-GM) označuje, da smo v PN3 za  $C_{l_1}$ ,  $C_{l_2}$  uporabili PC, za  $C''$  pa GM.

Nadalje, algoritem, ki uporablja difuzijske valčke, krajše zapišemo kot PDV. Vse algoritme smo implementirali v programskem okolju MATLAB in jih testirali na računalniku s procesorjem Intel® Pentium® CPU N3520 @ 2.16GHz in 4 GB pomnilnika. Pri implementaciji algoritma PDV smo za izračun potenc matrike uporabili obstoječo kodo, dano v [11].

## 5.1 Testna množica podatkov

Za eksperimentalno ovrednotenje potrebujemo testno množico prehodnih in cenovnih matrik. V naslednjem podrazdelku opišemo postopek za generiranje umetnih podatkov. Poleg tega v podrazdelku 5.1.2 posebej predstavimo še postopek za generiranje delno umetnih podatkov, ki opisujejo sistem izmenjave in uporabe koles v Ljubljani.

### 5.1.1 Simulacija umetnih podatkov

#### Prehodna matrika

Prehodno matriko  $P$  dimenzije  $N \times N$  generiramo tako, da je vsak element matrike  $\mathbf{P}$  neničelen z verjetnostjo  $\frac{1}{\sqrt{N}}$  in za vsako vrstico  $i$  velja  $\sum_{j=1}^N p_{ij} = 1$ . Podajmo še formalno algoritem `GenerirajPrehodno( $N$ )`, ki za poljubno naravno število  $N$  vrne prehodno matriko  $\mathbf{P}$ .

Tako generirano matriko lahko uporabimo za testiranje naših metod, vendar pa je za uporabo algoritma PDV potrebno zagotoviti še dodaten pogoj: matrika  $\mathbf{P}$  mora imeti nizek numerični rang. V ta namen naredimo še dodatni test, ali ima prehodna matrika nizek numerični rang pri neki specifični toleranci. Matrike s takimi lastnostmi nato uporabimo pri testiranju.

#### Cenovna matrika

Fiksirajmo poljubni pozitivni realni števili  $c, d > 0$ . Za vsak neničelni element prehodne matrike  $\mathbf{P}$  generiramo pripadajoči element cenovne matrike  $C$  iz porazdelitve naslednje slučajne spremenljivke

$$Y = c \cdot |\log(X)| - d,$$

kjer je  $X$  zvezna enakomerno porazdeljena slučajna spremenljivka na intervalu  $[0, 1]$ . Ni se težko prepričati, da ima slučajna spremenljivka  $c \cdot |\log(X)|$  eksponentno porazdelitev s parametrom  $\frac{1}{c}$ . Slučajno spremenljivko  $c \cdot |\log(X)|$  nato premaknemo za konstanto  $-d$  z razlogom, da je lahko kakšen element matrike  $C$  tudi negativen. Podajmo še formalni algoritem `GenerirajCenovno( $\mathbf{P}, c, d$ )`, ki za dano prehodno matriko  $\mathbf{P}$  ter parametra  $c$  in  $d$  vrne pripadajočo cenovno matriko  $C$ .

#### Podatki

Z uporabo algoritmov `GenerirajPrehodno` in `GenerirajCenovno` smo generirali prehodne in cenovne matrike naslednjih dimenzij:

- (i) za osnovni problem  $317 \times 317$ ,  $411 \times 411$  in  $1000 \times 1000$ ,
- (ii) za posplošeni problem  $235 \times 235$  ( $l_1 = 2$ ,  $l_2 = 3$ ) in  $500 \times 500$  ( $l_1 = 2$ ,  $l_2 = 1$ ).

Pri generiranju cenovnih matrik smo uporabili konstanti  $c = 100$  in  $d = 10$ .

---

**Algoritem** GenerirajPrehodno( $N$ )

---

**Vhod:** naravno število  $N$ **Izhod:** prehodna matrika  $\mathbf{P}$  dimenzije  $N \times N$  $\mathbf{P} :=$  ničelna matrika dimenzije  $N \times N$ ;**while** matrika  $\mathbf{P}$  ima ničelno vrstico **do**  **for**  $i, j \in \{1, 2, \dots, N\}$  **do**     $U :=$  slučajno število na  $[0, 1]$ ;    **if**  $U < \frac{1}{\sqrt{N}}$  **then**       $\mathbf{P}[i, j] := 1$ ;    **end if**  **end for****end while** $vsota :=$  ničelni vektor velikosti  $N$ ;**for**  $i, j \in \{1, 2, \dots, N\}$  **do**  **if**  $\mathbf{P}[i, j] == 1$  **then**     $U :=$  slučajno število na  $[0, 1]$ ;  $\mathbf{P}[i, j] := U$ ;  $vsota[i] := vsota[i] + U$ ;  **end if****end for****for**  $i, j \in \{1, 2, \dots, N\}$  **do**   $\mathbf{P}[i, j] := \mathbf{P}[i, j] / vsota[i]$ ;**end for****return**  $\mathbf{P}$ ;

---

**Algoritem** GenerirajCenovno( $\mathbf{P}, c, d$ )

---

**Vhod:** prehodna matrika  $\mathbf{P}$  dimenzije  $N \times N$ ,**Vhod:** pozitivni realni števili  $c$  in  $d$ **Izhod:** cenovna matrika  $C$  dimenzije  $N \times N$ **for**  $i = 1$  to  $N$  **do**  **for**  $j = 1$  to  $N$  **do**    **if**  $\mathbf{P}[i, j] > 0$  **then**       $U :=$  slučajno število na  $[0, 1]$ ;       $C[i, j] := c |\log(U)| - d$ ;    **end if**  **end for****end for****return**  $C$ ;

### 5.1.2 Simulacija sistema izmenjave koles

Želimo simulirati sistem izmenjave koles v Ljubljani, kjer je trenutno 36 kolesarskih postajališč. Predstavimo, kako generiramo prehodni matriki, ki opisujeta vožnjo kolesa med postajališči, in pripadajočo cenovno matriko, ki v tem primeru predstavlja realne razdalje med postajališči. Bralca opozorimo, da smo poskušali pridobiti realne podatke o izposoji koles registriranih uporabnikov ljubljanskega sistema, žal pa smo bili pri tem neuspešni.

#### Prehodni matriki

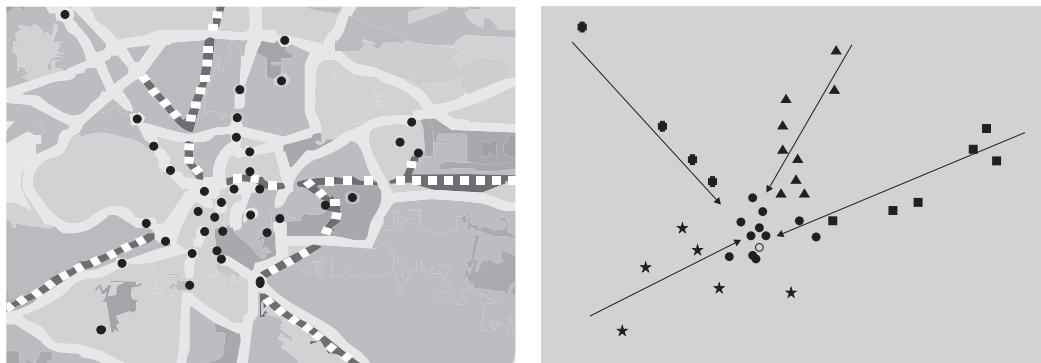
Generiranje matrik temelji na naslednjih predpostavkah. Potreba po izposoji koles je porazdeljena tako, da so v jutranjih urah in zgodaj popoldan postajališča v centru mesta predvsem prihodna postajališča, pozno popoldan pa postanejo predvsem odhodna postajališča. Poleg tega se večina ljudi nagiba h krajšim vožnjam in se zato daljše vožnje zgodijo s precej manjšo verjetnostjo. V jutranjih urah je večja verjetnost, da si kolesar izposodi kolo s postajališč na obrobju mesta in kolesari proti postajališčem v centru mesta. Ravno nasprotno se zgodi v popoldanskih urah.

Upoštevajoč zgornje predpostavke bolj natančno predstavimo generiranje prehodnih matrik  $\mathbf{P}_1$  in  $\mathbf{P}_2$ , ki zaporedoma opisujeta potrebo po izposoji koles zjutraj in popoldan. Najprej generiramo matriki  $Q_1$  in  $Q_2$  tako, da vsak njun element generiramo iz enakomerne porazdelitve na intervalu  $[a, b]$ , kjer interval  $[a, b]$  izberemo glede na lokacijo postajališč, čas potovanja in smer potovanja. V ta namen postajališča najprej razdelimo v pet skupin: skupino v centru mesta (na sliki 5.1 je to skupina postajališč, označena s krogci) in štiri skupine v predmestju (na sliki 5.1 so to skupine postajališč, označene s križci, trikotniki, kvadratki in zvezdicami). Intervale nato določimo na naslednji način:

- (i) če sta postajališči  $A$  in  $B$  v skupini v centru, potem je  $a = 0.6$  in  $b = 0.9$  (za obe matriki  $Q_1, Q_2$  in obe smeri),
- (ii) če se postajališče  $A$  nahaja v eni od predmestnih skupin, medtem ko je postajališče  $B$  v neki drugi predmestni skupini, potem je  $a = 0$  in  $b = 0.01$  (za obe matriki  $Q_1, Q_2$  in obe smeri),
- (iii) če se postajališči  $A$  in  $B$  nahajata v isti predmestni skupini, potem je  $a = 0$  in  $b = 0.2$  (za obe matriki  $Q_1, Q_2$  in obe smeri),
- (iv) če se postajališče  $A$  nahaja v skupini v centru, medtem ko je postajališče  $B$  v predmestni skupini, potem je
  - (a)  $a = 0$  in  $b = 0.3$  (za matriko  $Q_1$ , če je smer iz  $A$  v  $B$ , in za matriko  $Q_2$ , če je smer iz  $B$  v  $A$ ),
  - (b)  $a = 0.7$  in  $b = 1$  (za matriko  $Q_1$ , če je smer iz  $B$  v  $A$ , in za matriko  $Q_2$ , če je smer iz  $A$  v  $B$ ).

Poleg tega je verjetnost, da se kolo vrne nazaj na isto postajališče, iz katerega je začelo, neničelna. V tem primeru vzamemo  $a = 0.01$  in  $b = 0.35$  za obe matriki

## 5.2 Testiranje metod in analiza rezultatov



Slika 5.1: Leva slika prikazuje realne položaje kolesarskih postajališč in glavne poti v Ljubljani, medtem ko desna slika prikazuje delitev postajališč v pet skupin.

$Q_1$  in  $Q_2$ . Nazadnje prehodni matriki  $P_1$  in  $P_2$  dobimo zaporedoma iz matrik  $Q_1$  in  $Q_2$  tako, da vsako njuno vrstico normaliziramo.

### Cenovna matrika

Za cenovno matriko  $C$  smo vzeli matriko razdalj. Natančneje, s pomočjo Google zemljevida smo za vsaki postajališči  $s_i$  in  $s_j$  vzeli najkrajšo obstoječo kolesarsko pot med  $s_i$  in  $s_j$  in vrednost, merjeno v kilometrih, shranili na  $(i, j)$ -to mesto v  $C$ . Ker je verjetnost, da se bo kolesar vrnil nazaj v isto postajališče, neničelna, smo vrednosti  $c_{ii}$  generirali iz enakomerne porazdelitve na intervalu  $[1.1, 2.2]$ .

## 5.2 Testiranje metod in analiza rezultatov

Vse pristope med seboj primerjamo glede na realen čas, potreben za izračun ocene. V podrazdelku 5.2.1 na treh umetnih primerih primerjamo algoritem PH z algoritmom PDV. Nato v podrazdelku 5.2.2 na dveh umetnih primerih in sistemu izmenjave koles med seboj primerjamo algoritem PN1, dve varianti algoritma PN2 (PN2(GM), PN2(PC)) in štiri variante algoritma PN3 (PN3(GM-GM), PN3(GM-PC), PN3(PC-PC), PN3(PC-GM)). Če kakšna izmed variant PN2(GM), PN3(GM-GM), PN3(GM-PC) ali PN3(PC-GM) v konkretnem primeru ni vključena v primerjavo, pomeni, da je bilo iskanje v širino prezahtevno in algoritma GM nismo mogli uporabiti.

### 5.2.1 Osnovni problem

Rezultati za primere dimenzij  $317 \times 317$ ,  $411 \times 411$  oziroma  $1000 \times 1000$  so podani zaporedoma v tabelah 5.1-5.3. Prvi stolpec prikazuje dolžino  $k$  slučajnih sprehodov, ki jih generiramo. Naslednjih pet stolpcev se nanaša na algoritem PH. Ker je PH stohastičen algoritem, smo ga za vsako vrednost  $k$  pognali desetkrat in v drugem, tretjem ter četrtem stolpcu zaporedoma predstavili minimalen, maksimalen in

## 5 Eksperimentalni rezultati

povprečni rezultat vseh zagonov. Peti stolpec opisuje povprečni izvršilni čas (v sekundah) desetih zagonov na izbranih podatkih. Velikost vzorca  $M$  je predstavljena v šestem stolpcu. Predzadnja stolpca se nanašata na algoritem PDV. Za razliko od PH je PDV deterministični algoritem, zato smo ga za vsako vrednost  $k$  pognali enkrat in v sedmem stolpcu predstavili rezultat, ki ga le-ta vrne, medtem ko je izvršilni čas (v sekundah) prikazan v osmem stolpcu. V zadnjem stolpcu je absolutna razlika med povprečnim izvršilnim časom algoritma PH in izvršilnim časom algoritma PDV izražena v odstotkih izvršilnega časa algoritma PDV.

Naslednji parametri so bili uporabljeni v algoritmu PH:

- (i) velikost testnega vzorca  $M_0 = 10^4$  (v vseh treh primerih),
- (ii) stopnja značilnosti  $\alpha = 0.01$  (v vseh treh primerih),
- (iii) relativna napaka  $\epsilon = 10^{-3}$  (v primerih dimenzij  $317 \times 317$  in  $411 \times 411$ ) in  $\epsilon = 10^{-4}$  (v primeru dimenzije  $1000 \times 1000$ ).

Toleranca, ki smo jo uporabili pri metodi difuzijski valčki, je integrirano število v programu MATLAB in sicer  $2.2 \cdot 10^{-16}$ .

Tabela 5.1: Primerjava realnih časov algoritmov PH in PDV za prehodno matriko in pripadajočo cenovno matriko dimenzije  $317 \times 317$ .

$k$	PH					PDV		pohitritev
	$\hat{\mu}_{\min}$	$\hat{\mu}_{\max}$	$\hat{\mu}_p$	$t_p(s)$	$M$	$\tilde{\mu}$	$t(s)$	
$2^5$	2861	2866	2865	8	258932	2867	247	97%
$2^6$	5671	5680	5676	8	130612	5678	289	97%
$2^7$	11323	11337	11331	8	65454	11334	350	98%
$2^8$	22671	22695	22684	8	32762	22685	399	98%
$2^9$	45371	45447	45414	8	16201	45403	462	98%
$2^{10}$	90784	90954	90872	8	8162	90870	567	99%

Tabela 5.2: Primerjava realnih časov algoritmov PH in PDV za prehodno matriko in pripadajočo cenovno matriko dimenzije  $411 \times 411$ .

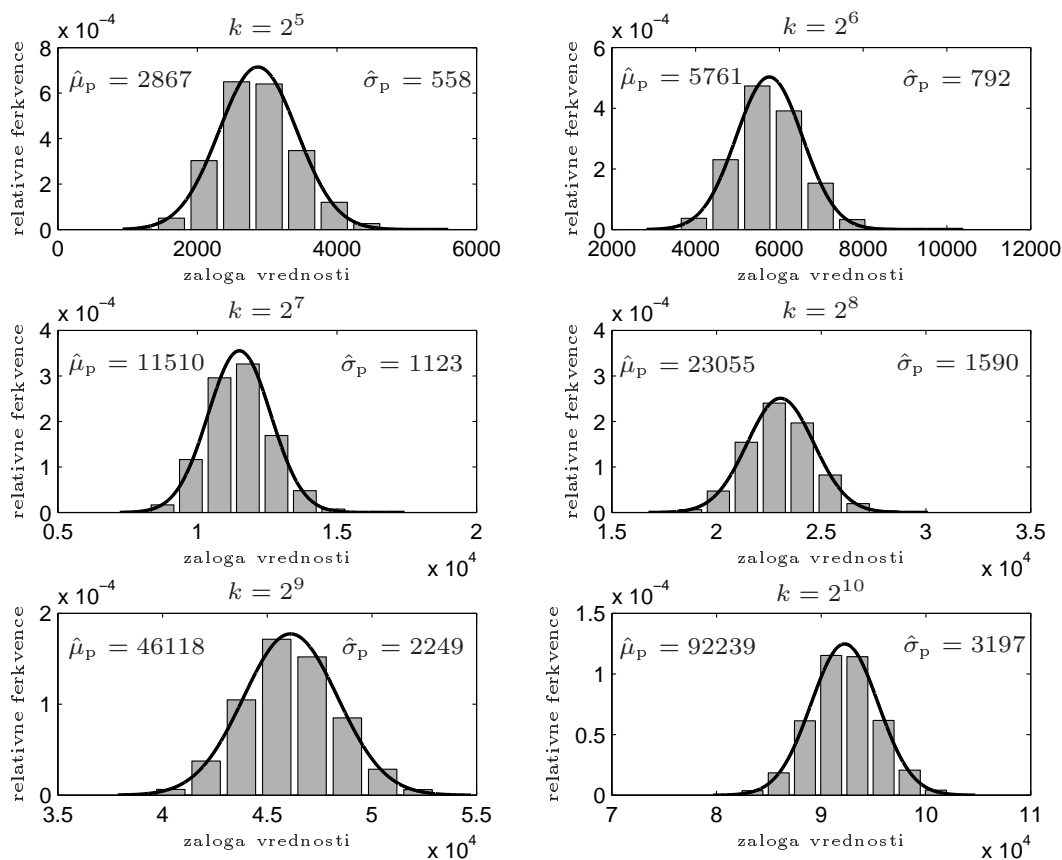
$k$	PH					PDV		pohitritev
	$\hat{\mu}_{\min}$	$\hat{\mu}_{\max}$	$\hat{\mu}_p$	$t_p(s)$	$M$	$\tilde{\mu}$	$t(s)$	
$2^5$	2865	2869	2867	10	252622	2869	204	95%
$2^6$	5758	5766	5761	10	125414	5762	252	96%
$2^7$	11493	11510	11500	10	64222	11504	301	97%
$2^8$	23046	23073	23055	10	31737	23048	366	97%
$2^9$	46086	46138	46118	11	15798	46114	420	97%
$2^{10}$	92154	92309	92239	11	7966	92243	515	98%

## 5.2 Testiranje metod in analiza rezultatov

Tabela 5.3: Primerjava realnih časov algoritmov PH in PDV za prehodno matriko in pripadajočo cenovno matriko dimenzije  $1000 \times 1000$ .

$k$	PH					PDV		pohitritev
	$\hat{\mu}_{\min}$	$\hat{\mu}_{\max}$	$\hat{\mu}_p$	$t_p(s)$	$M$	$\tilde{\mu}$	$t(s)$	
$2^5$	2906	2907	2907	905	25980000	2908	1806	50%
$2^6$	5796	5797	5797	1026	12892788	5797	2167	53%
$2^7$	11639	11642	11640	1110	6604388	11639	2715	59%
$2^8$	23252	23255	23253	1083	3211753	23255	3215	66%
$2^9$	46529	46536	46534	1115	1627952	46531	3962	72%
$2^{10}$	92958	92974	92968	1088	806173	92967	5336	80%

Na sliki 5.2 so za primer dimenzije  $411 \times 411$  za vsak  $k = 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}$  podani tudi histogrami relativnih frekvenc vzorcev, ki smo jih generirali z algoritmom PH. Za vsak histogram je podano povprečje  $\hat{\mu}_p$  desetih vzorčnih povprečji in povprečje  $\hat{\sigma}_p$  desetih vzorčnih standardnih odklonov. Spomnimo, da smo v opombi 4.1.2 nakazali, da se porazdelitev slučajne spremenljivke  $c(X_0, X_1, \dots, X_k)$  pod določenimi pogoji približuje normalni porazdelitvi. To se lepo vidi tudi na sliki 5.2.



Slika 5.2: Vzorčne porazdelitve slučajne spremenljivke  $c(X_0, X_1, \dots, X_k)$  za primer dimenzije  $411 \times 411$ .

Kot lahko vidimo iz tabel 5.1-5.3, je očitno, da algoritem PH prekaša algoritem PDV. Za rezultate v tabelah 5.1-5.2, izračunane z algoritmom PH, obstaja odstotek verjetnosti, da se le-ti razlikujejo od prave vrednosti  $\mu$  na tretji številki, ker smo za stopnjo značilnosti vzeli  $\alpha = 0.01$ , za relativno napako pa  $\epsilon = 10^{-3}$ . Podobno za rezultate v tabeli 5.3, izračunane z algoritmom PH, obstaja odstotek verjetnosti, da se le-ti razlikujejo od prave vrednosti  $\mu$  na četrti številki, ker smo za stopnjo značilnosti vzeli  $\alpha = 0.01$ , za relativno napako pa  $\epsilon = 10^{-4}$ .

Opazimo, da so ocene, izračunane z obema algoritmoma, precej podobne, vendar pa je izvršilni čas, potreben za izračun, pri algoritmu PDV precej večji. Z naraščajočim parametrom  $k$  tudi časi algoritma PDV naraščajo, medtem ko so pri algoritmu PH precej konstantni pri fiksni relativni napaki  $\epsilon$  in stopnji značilnosti  $\alpha$ .

### 5.2.2 Posplošeni problem

Rezultati za primer dimenzije  $235 \times 235$  oziroma  $500 \times 500$  so podani zaporedoma v tabelah 5.4 in 5.5, medtem ko so rezultati za sistem izmenjave koles podani v tabeli 5.6. Prvi stolpec prikazuje dolžino  $k$  slučajnih sprehodov, ki jih generiramo, medtem ko drugi stolpec opisuje algoritme PN1, PN2(GM), PN2(PC), PN3(GM-PC), PN3(PC-PC) in PN3(PC-GM), ki jih primerjamo. Opozorimo, da varianta PN3(GM-GM) zaradi časovne prezahtevnosti iskanja v širino ni vključena v primerjavo v nobenem primeru. Zaradi enakega razloga varianta PN3(PC-GM) ni vključena v primerjavo v primerih dimenzij  $235 \times 235$  in  $500 \times 500$ , medtem ko varianta PN3(GM-PC) ni vključena v primerjavo v primeru izmenjave koles. Ker so vsi predlagani algoritmi za reševanje posplošenega problema stohastični, je zopet primerno, da jih poženemo večkrat. Ta test zaradi časovnega obsega ni bil izveden, zato smo jih pognali le enkrat na izbranih podatkih. Tretji stolpec predstavlja rezultate  $\hat{\mu}$ , ki jih vrnejo algoritmi. Izvršilni čas (v sekundah), potreben za izračun rezultatov posameznega algoritma, je predstavljen v četrtem stolpcu. Velikost vzorca  $M$  za posamezen algoritem je predstavljena v petem stolpcu, medtem ko je velikost  $M_0$  testnega vzorca podana v šestem stolpcu. V zadnjem stolpcu je absolutna razlika izvršilnih časov algoritmov X in PN1 izražena v odstotkih izvršilnega časa algoritma PN1 za  $X = \text{PN2(GM)}, \text{PN2(PC)}, \text{PN3(GM-PC)}, \text{PN3(PC-PC)}, \text{PN3(PC-GM)}$ .

Naslednji parametri so bili uporabljeni v algoritmih:

- (i) stopnja značilnosti  $\alpha = 0.01$  (v vseh treh primerih),
- (ii) relativna napaka  $\epsilon = 10^{-4}$  (v vseh treh primerih).

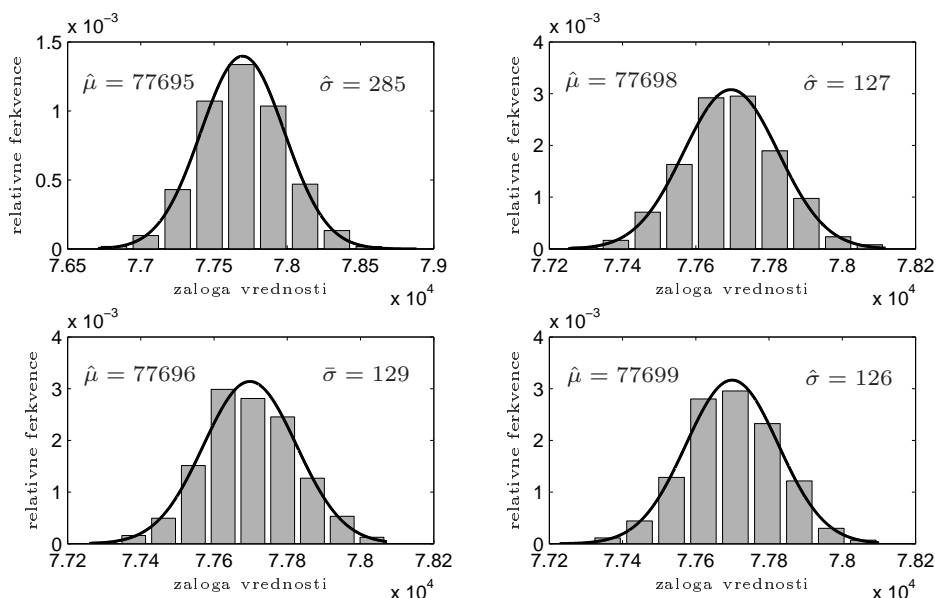
V tabelah 5.7-5.9 so za vse tri primere zaporedoma predstavljeni še izvršilni časi, ki so bili potrebni za izračun cenovnih matrik  $C_{l_1}$ ,  $C_{l_2}$  (prvi štirje stolpci) in  $C''$  (zadnji trije stolpci) v različicah algoritmov PN2 oziroma PN3 (ti časi so seveda že všteti v končne čase, ki so prikazani v tabelah 5.4-5.6). Prvi stolpec opisuje možna algoritma GM in PC, uporabljena v PN2. Drugi stolpec predstavlja cenovni matriki  $C_{l_1}$  in  $C_{l_2}$ , za kateri je v tretjem stolpcu podan izvršilni čas (v sekundah), potreben za njun izračun. Velikost vzorca  $M$ , ki smo ga generirali z algoritmom



## 5.2 Testiranje metod in analiza rezultatov

PC, je predstavljena v četrtem stolpcu. Peti stolpec opisuje možna algoritma GM in PC, uporabljena v PN3 za izračun cenovne matrike  $C'''$ , medtem ko je izvršilni čas (v sekundah), potreben za njen izračun, predstavljen v šestem stolpcu. Zadnji stolpec zopet opisuje velikost vzorca  $M$ , ki smo ga generirali z algoritmom PC, za izračun cenovne matrike  $C'''$ . Bralca opozorimo, da so cenovne matrike  $C_{l_1}$ ,  $C_{l_2}$  in  $C'''$  neodvisne od parametra  $k$ . To pomeni, da v primerih ni bilo potrebno za vsak parameter  $k$  posebej računati cenovnih matrik  $C_{l_1}$ ,  $C_{l_2}$  in  $C'''$ . Ko smo jih prvič izračunali, smo jih enostavno shranili in jih nato ponovno uporabili.

Na sliki 5.3 so za primer sistema izmenjave koles za  $k = 2^{12}$  podani še histogrami relativnih frekvenc vzorcev, ki smo jih generirali z algoritmi PN1, PN2(PC), PN3(PC-PC) in PN3(PC-GM). Za vsak histogram je podano tudi vzorčno povprečje  $\hat{\mu}$  in vzorčni standardni odklon  $\hat{\sigma}$ .



Slika 5.3: Vzorčne porazdelitve slučajne spremenljivke  $c(X_0, X_1, \dots, X_k)$  za primer dimenzije  $36 \times 36$  in  $k = 2^{12}$ , dobljene z algoritmi PN1 (zgoraj levo), PN2(PC) (zgoraj desno), PN3(PC-PC) (spodaj levo) in PN3(PC-GM) (spodaj desno).

V smislu natančnosti ocene  $\hat{\mu}$  so vsi algoritmi primerljivi, saj se vsi rezultati razlikujejo na četrti številki. To je v skladu s pričakovanji, saj smo za relativno napako izbrali  $\epsilon = 10^{-4}$ . Kar se tiče časovne primerjave, pa so opazne razlike med njimi. Najbolj odstopa algoritem PN1, ki je daleč najpočasnejši. Med verzijami algoritmov PN2 in PN3 razlike niso tako ekstremne. Opozorimo, da so vse variante algoritma PN2 in PN3, ki uporabljajo algoritem GM, pri večjih vrednostih parametrov  $l_1$  in  $l_2$  oziroma večjem redu  $N$  praktično neuporabne. Za ilustracijo smo vseeno vzeli tudi manjše vrednosti za parametra  $l_1$  in  $l_2$ , da smo lahko med seboj primerjali več variant. V nasprotnem sta primerni le verziji PN2(PC) in PN3(PC-PC).

V večini primerov je algoritem PN2 (oziroma njegove verzije) nekoliko hitrejši od PN3 (oziroma njegovih verzij). Glavni razlog je v tem, da dodatna gradnja cenovne matrike  $C'''$  v algoritmu PN3, ki je časovno zelo potratna, prinese zelo malo

5 Eksperimentalni rezultati

Tabela 5.4: Primerjava realnih časov algoritmov PN1, PN2(GM), PN2(PC), PN3(GM-PC) in PN3(PC-PC) za dve prehodni matriki in pripadajoči cenovni matriki dimenzij  $235 \times 235$  ter  $l_1 = 2$ ,  $l_2 = 3$ .

$k$	Alg.	$\hat{\mu}$	$t(s)$	$M$	$M_0$	pohitritev
$2^{10}$	PN1	461675	817	162140	$10^4$	
	PN2(GM)	461685	175	71329	$10^4$	79%
	PN2(PC)	461722	323	71390	$10^4$	60%
	PN3(GM-PC)	461686	326	6864	$10^3$	60%
	PN3(PC-PC)	461667	473	6828	$10^3$	42%
$2^{11}$	PN1	923429	829	80262	$10^4$	
	PN2(GM)	923437	174	35471	$10^4$	79%
	PN2(PC)	923399	328	36331	$10^4$	60%
	PN3(GM-PC)	923399	326	3355	$10^3$	61%
	PN3(PC-PC)	923354	473	3341	$10^3$	43%
$2^{12}$	PN1	1846837	825	40477	$10^4$	
	PN2(GM)	1846889	175	17752	$10^4$	79%
	PN2(PC)	1846729	320	17789	$10^4$	61%
	PN3(GM-PC)	1846791	326	1722	$10^3$	60%
	PN3(PC-PC)	1846550	473	1700	$10^3$	43%

Tabela 5.5: Primerjava realnih časov algoritmov PN1, PN2(GM), PN2(PC), PN3(GM-PC) in PN3(PC-PC) za dve prehodni matriki in pripadajoči cenovni matriki dimenzij  $500 \times 500$  ter  $l_1 = 2$ ,  $l_2 = 1$ .

$k$	Alg.	$\hat{\mu}$	$t(s)$	$M$	$M_0$	pohitritev
$2^{10}$	PN1	274694	1160	267769	$10^4$	
	PN2(GM)	274698	621	218246	$10^4$	46%
	PN2(PC)	274721	881	220267	$10^4$	24%
	PN3(GM-PC)	274703	378	38251	$10^4$	67%
	PN3(PC-PC)	274732	601	38182	$10^4$	48%
$2^{11}$	PN1	549430	1163	135134	$10^4$	
	PN2(GM)	549428	645	112647	$10^4$	45%
	PN2(PC)	549397	885	112427	$10^4$	24%
	PN3(GM-PC)	549430	378	19259	$10^4$	67%
	PN3(PC-PC)	549482	601	19385	$10^4$	48%
$2^{12}$	PN1	1098924	1185	67310	$10^4$	
	PN2(GM)	1098835	629	55241	$10^4$	47%
	PN2(PC)	1098907	891	55781	$10^4$	25%
	PN3(GM-PC)	1098823	379	9675	$10^3$	68%
	PN3(PC-PC)	1099037	600	9497	$10^3$	49%

## 5.2 Testiranje metod in analiza rezultatov

Tabela 5.6: Primerjava realnih časov algoritmov PN1, PN2(PC), PN3(PC-PC) in PN3(PC-GM) za dve prehodni matriki in pripadajoči cenovni matriki dimenzij  $36 \times 36$  ter  $l_1 = 7$ ,  $l_2 = 5$ .

$k$	Alg.	$\hat{\mu}$	$t(s)$	$M$	$M_0$	pohitritev
$2^8$	PN1	4855	369	147401	$10^4$	
	PN2(PC)	4855	152	30292	$10^4$	59%
	PN3(PC-PC)	4854	184	27831	$10^4$	50%
	PN3(PC-GM)	4855	158	28180	$10^4$	57%
$2^{10}$	PN1	19422	356	37031	$10^4$	
	PN2(PC)	19422	152	7820	$10^3$	57%
	PN3(PC-PC)	19423	184	7083	$10^3$	48%
	PN3(PC-GM)	19424	158	7221	$10^3$	56%
$2^{12}$	PN1	77695	350	9117	$10^3$	
	PN2(PC)	77696	152	1892	$10^3$	57%
	PN3(PC-PC)	77698	184	1774	$10^3$	47%
	PN3(PC-GM)	77699	158	1857	$10^3$	55%

Tabela 5.7: Realni časi, potrebni za konstrukcijo cenovnih matrik  $C_2$ ,  $C_3$  in  $C''$  v primeru dimenzije  $235 \times 235$ .

PN2				PN3		
Alg.	$C'$	$t(s)$	$M$	Alg.	$t(s)$	$M$
GM	$C_2$	2	-	PC	287	$7 \cdot 10^5$
	$C_3$	31	-			
PC	$C_2$	60	$1.5 \cdot 10^5$	PC	288	$7 \cdot 10^5$
	$C_3$	119	$2 \cdot 10^5$			

Tabela 5.8: Realni časi, potrebni za konstrukcijo cenovnih matrik  $C_2$  in  $C''$  v primeru dimenzije  $500 \times 500$ .

PN2				PN3		
Alg.	$C'$	$t(s)$	$M$	Alg.	$t(s)$	$M$
GM	$C_2$	11	-	PC	318	$2.5 \cdot 10^5$
	$C$	-	-			
PC	$C_2$	257	$2.5 \cdot 10^5$	PC	295	$2.5 \cdot 10^5$
	$C$	-	-			

## 5 Eksperimentalni rezultati

Tabela 5.9: Realni časi, potrebni za konstrukcijo cenovnih matrik  $C_7$ ,  $C_5$  in  $C''$  v primeru dimenzije  $36 \times 36$ .

PN2				PN3		
Alg.	$C'$	$t(s)$	$M$	Alg.	$t(s)$	$M$
PC	$C_7$	94	$7 \cdot 10^5$	PC	28	$7 \cdot 10^5$
	$C_5$	31	-			
PC	$C_7$	49	$5 \cdot 10^5$	GM	2	-
	$C_5$	119	$2 \cdot 10^5$			

k nadaljnjemu računanju, saj s to dodatno gradnjo problem le razpolovimo – v luči tega, da se nova dolžina sprehoda le razpolovi. Zato dajemo prednost algoritmu PN2.

Za konec dodajmo, da lahko vse algoritme za reševanje posplošenega problema seveda uporabimo tudi za reševanje osnovnega primera. Opazimo, da v tem primeru vsi algoritmi za reševanje posplošenega problema sovpadajo z algoritmom PH. Posledično vse tabele v razdelku 5.2.1 posredno primerjajo tudi vse algoritme za reševanje posplošenega problema.

---

## Zaključek

V doktorski disertaciji obravnavamo problem računanja povprečja cene slučajnega sprehoda končne dolžine z danim začetnim stanjem. Ker je analitičen izračun povprečja – razen v zelo omejenih in majhnih primerih – praktično nemogoč, se v disertaciji osredotočimo na učinkovite metode za ocenjevanje povprečja. Glavna ideja temelji na uporabi metode Monte Carlo, ki omogoča, da oceno poiščemo s pomočjo slučajnih števil in simulacij: enostavno generiramo vzorec slučajnih sprehodov glede na dane prehodne matrike in za oceno vzamemo povprečje cen sprehodov v vzorcu. Bistvo je, da se z večanjem vzorca ocena približuje pravi vrednosti.

Osnovni problem, kjer je slučajni sprehod podan z eno prehodno matriko, njegova cena pa z eno cenovno matriko, posplošimo na končno mnogo prehodnih in cenovnih matrik. Najprej predstavimo algoritem PH za reševanje osnovnega problema. Ta je temeljni gradnik v vseh treh nadaljnjih algoritmih PN1, PN2 in PN3, ki jih razvijemo za reševanje posplošenega problema. Algoritem PN1 je neposredna posplošitev algoritma PH, medtem ko je algoritem PN2 nadgradnja algoritma PN1. Ideja je, da se tisti deli podsprehodov, ki jih opisuje posamezna prehodna matrika, zakodirajo na sprehodih dolžine ena s primerno potenco prehodne matrike. Skladno s tem je potrebno izračunati tudi novo pripadajočo cenovno matriko. Izkaže se, da je za analitičen izračun nove cenovne matrike potrebno rešiti  $N$  (v praksi manjših) osnovnih problemov, kjer je  $N$  red prehodne matrike. Podamo algoritem GM, ki z iskanjem v širino zgradi novo cenovno matriko. Ker vedno obstaja možnost, da nove cenovne matrike ne moremo izračunati analitično, v tem primeru predstavimo algoritem PC, ki z uporabo metode Monte Carlo poišče njen približek. Ko imamo nove prehodne in cenovne matrike izračunane, se problem prevede na zelo poseben posplošen problem, ki ga rešimo z algoritmom PN1. Algoritem PN3 je nadaljnja nadgradnja algoritma PN2 in reševanje zelo posebnega posplošenega problema prevede na reševanje osnovnega problema. V ta namen še enkrat uporabimo idejo o kodiranju informacije. Dobljeni osnovni problem nazadnje rešimo z algoritmom PH.

Vse predlagane algoritme smo eksperimentalno ovrednotili. Algoritem PH smo primerjali z algoritmom PDV, ki so ga predlagali avtorji v članku [19] in temelji na računanju potenc prehodnih matrik z metodo difuzijskih valčkov. Ker se metoda difuzijskih valčkov zadovoljivo obnaša v primeru, ko ima matrika nizek numerični

rang, smo se omejili samo na prehodne matrike z nizkim numeričnim rangom. Rezultati primerjav kažejo, da algoritem PH prekaša algoritem PDV v vseh primerih. Čeprav je njuna natančnost zelo primerljiva, pa so časovni rezultati algoritma PH precej boljši. Nazadnje smo med sabo primerjali še algoritem PN1, dve varianti algoritma PN2 (PN2(GM), PN2(PC)) in tri variante algoritma PN3 (PN3(GM-PC), PN3(PC-PC), PN3(PC-GM)). Izkaže se, da je v varianti PN3(GM-GM) iskanje v širino časovno prezahtevno, zato je nismo vključili v primerjave. Podobno sta bili varianti PN3(PC-GM) in PN3(GM-PC) vključeni v primerjavo le v tistih primerih, kjer iskanje v širino ni bilo časovno prezahtevno. Primerjave kažejo, da so vsi algoritmi podobno natančni, zopet pa obstajajo časovne razlike. Najbolj očitna razlika je pri algoritmu PN1, ki je daleč najpočasnejši. Ostale verzije algoritmov PN2 in PN3 so časovno precej podobne, vseeno pa dajemo prednost algoritmu PN2.

## 6.1 Nadaljnje delo

Čeprav je koncept metode Monte Carlo preprost, vsestranski, prilagodljiv in enostaven za uporabo, pa njena počasna konvergenca lahko predstavlja pomankljivost v izračunih, ki vključujejo visoko natančnost. Iz centralnega limitnega izreka sledi, da je konvergenca metode Monte Carlo približno reda  $M^{-\frac{1}{2}}$ , kjer je  $M$  velikost vzorca. Če želimo izboljšati konvergenco, je ena izmed možnih strategij, da slučajna števila (oziroma psevdoslučajna števila), ki igrajo ključno vlogo pri generiranju vzorca, zamenjamo z determinističnimi števili. To vodi do kvazi Monte Carlo metode, ki jo lahko preprosto opišemo kot deterministično verzijo metode Monte Carlo v smislu, da slučajne vzorce v metodi Monte Carlo zamenjamo z dobro izbranimi determinističnimi vzorci. Glavni cilj je, da izberemo takšne deterministične vzorce, za katere je deterministična napaka manjša kot stohastična napaka metode Monte Carlo. S primerno izbranimi determinističnimi vzorci je lahko v nekaterih primerih red konvergence enak  $M^{-1}$ . Več o metodi kvazi Monte Carlo si lahko bralec prebere v [15, 30].

V posplošenem problemu smo uporabili konstantno število korakov  $l_1$  in  $l_2$  glede na prehodni matriki  $\mathbf{P}_1$  oziroma  $\mathbf{P}_2$ . V luči aplikacije sistema izposoje koles bi bilo zanimivo, če bi dopustili spremenljivo število izposoje koles za različna časovna obdobja dneva. Zdi se, da implementacija tega v algoritmu PN1 ne predstavlja težav, brž ko lahko uspešno generiramo  $l_1$  in  $l_2$  iz neke v naprej dane porazdelitve. Modifikacija algoritmov PN2 in PN3 v tej smeri bi pomenila, da se časovna kot tudi prostorska zahtevnost algoritmov poveča. Motivacija za nadaljnje delo se zlasti navezuje na zadnji problem.

# Literatura

---

- [1] J. Abbot, J. Austerweil in T. L. Griffiths, *Human memory search as a random walk in a semantic network*, Advances in Neural Information Processing Systems **25** (2012) 3050–3058.
- [2] K. Au, G. Byrnes, C. Bain, M. Fackrell, C. Brand, D. Campbell in P. Taylor, *Predicting Overflow in an Emergency Department*, IMA Journal of Management Mathematics **20** (2009) 39–49.
- [3] C. Bain, P. Taylor, G. McDonnell in A. Georgiou, *Myths of ideal hospital occupancy*, Medical Journal of Australia **192** (2010) 42–43.
- [4] E. Behrends, *Introduction to Markov Chains: With Special Emphasis on Rapid Mixing*, Vieweg, Wiesbaden, 2000.
- [5] M. Borodovsky, J. McIninch, E. Koonin, K. Rudd, C. Medigue in A. Danchin, *Detection of new genes in a bacterial genome using Markov models for three gene classes*, Nucleic Acids Research **23** (1995) 3554–3562.
- [6] S. Brooks, *Markov chain Monte Carlo method and its application*, Journal of the Royal Statistical Society: Series D (the Statistician), **47** (1998) 69–100.
- [7] S. Brooks, A. Gelman, G. L. Jones, in X. Meng, *Handbook of Markov Chain Monte Carlo*, Chapman & Hall, New York, 2011.
- [8] J. Buzacott in J. Shanthikumar, *Stochastic Models of Manufacturing Systems*, International ed., Prentice-Hall, New Jersey, 1993.
- [9] L. Caggiani in M. Ottomanelli, *A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems*, Procedia - Social and Behavioral Sciences **87** (2013) 203–210.
- [10] W. Ching, *Iterative Methods for Queuing and Manufacturing Systems*, Springer, London, 2001.
- [11] R. R. Coifman in M. Maggioni, *Diffusion wavelets*, Applied and Computational harmonic Analysis **21** (2006) 53–94.
- [12] E. Côme in L. Oukhrillou, *Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Vélib' System of Paris*, ACM Transactions on Intelligent Systems and Technology **5** (2014) 39:1–39:21.

- [13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, in C. Stein, *Introduction to Algorithms (2nd ed.)*, MIT Press, Cambridge, MA, 2001.
- [14] W. L. Dunn, *Exploring Monte Carlo methods*, Elsevier, Amsterdam, 2012.
- [15] P. L'Ecuyer, C: Lecot, in B. Tuffin, *A Randomized quasi-Monte Carlo Simulation Method for Markov Chains*, *Operations Research* **56** (2008) 958–975.
- [16] R. J. Elliott in P. E. Kopp, *Mathematics of financial markets*, Springer-Verlag, New York, 2005.
- [17] W. R. Gilks, *Markov chain Monte Carlo in practice*, Chapman & Hall, London, 1996.
- [18] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Applications of Mathematics, Springer, New York, 2003.
- [19] M. J. Goldberg in S. Kim, *Applications of some formulas for finite Markov chains*, *Applied and Computational Harmonic Analysis* **30** (2011) 37–46.
- [20] J. Guo, Y. Wang, K. S. Tang, S. Chan, E. W. M. Wong, P. Taylor in M. Zukerman, *Evolutionary optimization of file assignment for a large-scale video-on-demand system*, *IEEE Transactions on Knowledge and Data Engineering* **20** (2008) 836–850.
- [21] J. Guo, E. W. M. Wong, S. Chan, P. Taylor, M. Zukerman in K. S. Tang, *Combination load balancing for video-on-demand systems*, *IEEE Transactions on Circuits and Systems for Video Technology* **18** (2008) 937–948.
- [22] P. C. Hansen, *Rank-deficient and discrete ill-posed problems : numerical aspects of linear inversion*, SIAM, Philadelphia , 1998.
- [23] M. J. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, *Biometrika* **57** (1970) 97–109.
- [24] O. Häggström, *Finite Markov chains and algorithmic applications*, Cambridge University Press, Cambridge, 2002.
- [25] A. Hobolth, *A Markov chain Monte Carlo expectation maximization algorithm for statistical analysis of DNA sequence evolution with neighbour-dependent substitution rates*, *Journal of Computational and Graphical Statistics* **17** (2008) 138–164.
- [26] G. W. King, *Stochastic methods in statistical mechanics*, Monte Carlo method, National Bureau of Standards (1951) 12–18.
- [27] A. J. Kleywegtt, A. Shapiro in T. Homem-De-Mello, *Society for Industrial and Applied Mathematics*, In: Technical report, Schuijbroek **12** (2001) 479-502.
- [28] V. G. Kulkarni, *Introduction to Modeling and Analysis of Stochastic systems*, Springer, New York, 2011.
- [29] E. W. Montroll, *Markoff chains and the excluded volume effect in polymer chains*, *J. Chem. Phys.* **18** (1950) 734–743.
- [30] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.



## Literatura

- [31] R. Norberg, *A time-continuous Markov chain interest model with applications to insurance*, J. Appl. Stoch. Models and Data Anal. (1995) 245–256.
- [32] J. R. Norris, *Markov Chains*, Cambridge University Press, Cambridge, 1997.
- [33] S. C. Port, *Theoretical probability for applications*, Wiley, New York, 1994.
- [34] A. Pršlja, *Random walks relative to multiple transition matrices*, International Workshop on Matrices and Statistics, Ljubljana, Slovenija, junij 2014.
- [35] A. Pršlja, *Random walks relative to multiple transition matrices*, Operators and Matrices, sprejeto za objavo.
- [36] C. Robert in G. Casella, *Monte Carlo Statistical Methods (2nd ed.)*, Springer-Verlag, New York, 2004.
- [37] S. L. Salzberg, A. L. Delcher, S. Kasif in O. White, *Microbial gene identification using interpolated Markov models*, Nucleic Acids Research **26** (1998) 554–548.



# Stvarno kazalo

---

- aproksimativni interval zaupanja, 33
- graf
  - kreepko povezan, 9
  - usmerjen, 9
  - usmerjen sprehod, 9
- markovska lastnost, 7
- markovska veriga
  - aperiodična, 17
  - diskretna, 7
  - homogena, 8
  - nehomogena, 8
  - nerazcepna, 15
  - periodična, 17
  - razcepna, 15
- matrika
  - cenovna, 39
  - kumulativna, 37
  - prehodna, 8
  - soednosti, 9
  - stohastična, 8
- numerični rang, 53
  - nizek, 53
- porazdelitev
  - inducirana, 11
  - limitna, 27
  - prehodna, 10
  - stacionarna, 22
  - začetna, 10
- povezanostni razred, 14
- prehodna verjetnost, 8
- psevdoslučajno število, 35
- slučajni proces, 7
- slučajni sprehod, 39
- slučajno število, 35
- stanje
  - aperiodično, 16
  - dosegljivo, 13
  - minljivo, 19
  - perioda, 16
  - periodično, 16
  - povrnljivo, 19
    - neničelno, 20
    - ničelno, 20
- stopnja zaupanja, 33
- vzorčni standardni odklon, 33
- vzorčno povprečje, 32
- vzorec porazdelitve, 32
- zaprtá množica, 14