

KAUNO TECHNOLOGIJOS UNIVERSITETAS
VYTAUTO DIDŽIOJO UNIVERSITETAS
VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

MINDAUGAS BRAŽĖNAS

**KETVIRTOS EILĖS FAZINIŲ SKIRSTINIŲ STRUKTŪRŲ
ANALIZĖ IR MARKOVO ATVYKIMŲ PROCESO PARAMETRŲ
PAIEŠKOS MAKSIMALAUS TIKĖTINUMO METODU
ALGORITMO IŠLYGIAGRETINIMAS**

Daktaro disertacijos santrauka
Gamtos mokslai, informatika (N 009)

2019, Kaunas

Disertacija rengta 2014–2018 metais Kauno technologijos universiteto Matematikos ir gamtos mokslų fakultete Matematinio modeliavimo katedroje.

Mokslinis vadovas:

Prof. habil. dr. Eimutis VALAKEVIČIUS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009).

Lietuvių kalbą redagavo: Giedrė Vadlugaite (UAB „Lingvobalt“)

Anglų kalbą redagavo: Aleksandra Drublionienė (UAB „Lingvobalt“)

Informatikos mokslo krypties disertacijos gynimo taryba:

prof. habil. dr. Rimantas BARAUSKAS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009) – **pirmininkas**;

prof. habil. dr. Raimondas ČIEGIS (Vilniaus Gedimino technikos universitetas, gamtos mokslai, informatika, N 009);

prof. dr. Gintaras PALUBECKIS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009);

prof. habil. dr. Minkydas RAGULSKIS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009);

doc. dr. Zsolt SAFFER (Vienos technologijos universitetas, gamtos mokslai, informatika, N 009).

Disertacija bus ginama viešame informatikos mokslo krypties disertacijos gynimo tarybos posėdyje 2019 m. birželio 18 d. 13 val. Kauno technologijos universiteto Rektorato salėje.

Adresas: K. Donelaičio g. 73-402, 44249 Kaunas, Lietuva.

Tel. +370 37 300 042; faks. +370 37 324 144; el. paštas doktorantura@ktu.lt

Disertacijos santrauka išsiųsta 2019 m. gegužės 17 d.

Su disertacija galima susipažinti internetinėje svetainėje <http://ktu.edu>, Kauno technologijos universiteto (K. Donelaičio g. 20, 44239 Kaunas), Vytauto Didžiojo universiteto (K. Donelaičio g. 52, 44244 Kaunas) ir Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, 10223 Vilnius) bibliotekose.

ŽYMĖJIMAI

Matricos vaizduojamos paryškintomis didžiosiomis raidėmis, pvz., \mathbf{Q} , konkretus matricos elementas nurodomas kaip $Q_{i,j}$. Vektoriai vaizduojami paryškintomis mažosiomis raidėmis, pvz., α, \mathbf{d} , o jų elementai nurodomi kaip α_i, \mathbf{d}_i . Matricos ir vektoriai gali būti numeruojami, pvz., $\mathbf{D}_0, \mathbf{P} \begin{bmatrix} v \\ u \end{bmatrix}, \mathbf{d}_1, \mathbf{a} \begin{bmatrix} v \\ u \end{bmatrix}$. Atsitiktiniai dydžiai vaizduojami didžiosiomis kaligrafinio šifro raidėmis, pvz., \mathcal{X} . Atsitiktiniai dydžiai gali būti indeksuojami, pvz., $\mathcal{B}_i, \mathcal{N}_{i,j}$. Atsitiktinių dydžių aibės vaizduojamos paryškintomis kaligrafinio šifro didžiosiomis raidėmis, pvz., $\mathcal{H} = \{\mathcal{B}_i, \mathcal{N}_{i,j}, \mathcal{Z}_i\}$. Atsitiktinių dydžių aibės vidurkiai nurodomi kaip pvz., $\mathbf{h} = \{b_i, n_{i,j}, z_i\}$. Atsitiktinių dydžių sąrašo pavyzdys – $\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2, \dots)$, o jo realizacijų sąrašas – $\mathbf{x} = (x_1, x_2, \dots)$. Parametrų aibės vaizduojamos tarp lenktų skliaustelių, pvz., (α, \mathbf{D}_0) .

Kiti žymėjimai:

r.a.d. – atstumo metrika, skaičiams $x, y \in \mathbb{R}$ apibrėžiama kaip $\frac{|y-x|}{|x|}$,

$\mathbb{P}(\dots)$ – tikimybė,

$\mathbb{E}[\dots]$ – tikėtina reikšmė, vidurkis,

\mathbf{Q} – Markovo grandinės intensyvumų matrica,

α – pradinių tikimybių vektorius (eilutė),

\mathbf{D}_0 – perėjimų, kurie negeneruoja įvykių, intensyvumų matrica,

\mathbf{D}_1 – perėjimų, kurie generuoja įvykius, intensyvumų matrica,

\mathbf{d}_1 – perėjimų į absorbuojančią būseną intensyvumų vektorius (stulpelis),

\mathbf{d} – perėjimų į absorbuojančią būseną intensyvumų vektorius (stulpelis),

\mathbf{r} – Erlango šakų dydžių vektorius,

λ – Erlango šakų intensyvumų vektorius,

π – pradinės Erlango šakos tikimybių vektorius (eilutė),

$\mathbf{\Pi}$ – Erlango šakų pasirinkimo tikimybių matrica,

$\mathbb{1}$ – vektorius (stulpelis) su vienetiniais elementais,

e_i – vektorius (eilutė) su nuliniiais elementais išskyrus i -tąjį elementą, kuris lygus 1.

Naudojamos santrumpos:

CTMC – tolydaus laiko Markovo grandinė,

DTMC – diskretaus laiko Markovo grandinė,

CHMM – tolydaus laiko paslėptoji Markovo grandinė,

PH – fazinis skirstinys,

ER – Erlango skirstinys,

HErD – Erlango hiperskirstinys,

MAP – Markovo atvykimų procesas,

TMAP – nutrūkstantis Markovo atvykimų procesas,

ER-CHMM – MAP arba TMAP procesas, turintis Erlango hiperskirstinio struktūrą,

EM – maksimalaus tikėtinumo metodas,

GPU – vaizdo procesorius,

CPU – centrinis procesorius,

llh – tikėtinumo funkcijos reikšmės logaritmas.

IVADAS

Tolydaus laiko baigtinės būsenų erdvės Markovo modeliai dešimtmečiais naudojami modeliuojant stochastinius procesus. Tokio tipo modeliavimo pranašumas – galimybė analiziškai apskaičiuoti modelio charakteristikas. Markovo proceso tolesnė raida priklauso tik nuo dabartinės būsenos (t. y. nepriklauso nuo ankstesnių būsenų). Kitaip tariant, procesas neturi atminties. Ši savybė lemia eksponentinio skirstinio savybių. Atsitiktinai sugeneruoto laiko (pagal eksponentinį skirstinį) likusio laiko skirstinys nepriklauso nuo praėjusio laiko ir turi tą patį skirstinį. Toks skirstinys vadinamas nesenstančiu ir nejaunėjančiu. Vis dėlto realaus pasaulio procesai nusakomi pačiais įvairiausiais skirstiniais. Tokių procesų nusakymo markovišku modeliu tikslumas priklauso nuo to, kaip tiksliai skirstiniai yra aproksimuojami eksponentinių skirstinių mišiniais ir sąsukomis. Šie skirstiniai vadinami faziniais, o juos sudarančios būsenos – fazėmis.

Bet kurį teigiamo atsitiktinio dydžio skirstinį galima (kiek norima) tiksliai aproksimuoti faziniu skirstiniu. Literatūroje daug dėmesio skiriama fazinių skirstinių reprezentacijos ir parametrų paieškos problemoms spręsti. Norint atlikti optimalią parametrų paiešką, reikia turėti fazinių skirstinių kanonines formas. Deja, šios kanoninės formos yra surastos faziniams skirstiniams tik su 2, 3 fazėmis. Aukštesnės eilės kanoninių formų radimas analiziniu būdu yra sudėtingas. Dėl šios priežasties nagrinėjamas įvairių fazinių skirstinių klasių pritaikomumas parametrų paieškai. Tačiau tokių klasių taikymas nėra optimalus ir aproksimuojant duotą skirstinį gali prireikti daugiau fazių. Kita vertus, parametrų paieška, naudojant tam tikras klases, yra efektyvi. Kuriant Markovo modelius susiduriama su problema – būsenų aibė sparčiai didėja naudojant fazinius skirstinius, turinčius daugiau fazių. Šią problemą galima sumažinti naudojant fazinius skirstinius su mažesniu fazių skaičiumi. Optimalus fazinių skirstinių radimas galimas tik turint jų kanonines formas.

Mokslinėje literatūroje nėra pateikta rezultatų apie bendrąsias fazinių skirstinių struktūras. Mes tiriamo keturių fazių skirstinius sugeneruodami visas galimas jų struktūras ir įvairiais būdais pašalindami perteklines. Gauta struktūrų aibė yra tirama empiriškai, panaudojant jas įvairių skirstinių aproksimavimui.

Faziniu skirstiniu (kaip ir bet kuriuo kitu skirstiniu) galima modeli-

uoti tarpusavyje nepriklausomus tarplaikius tarp stebimo reiškinių įvykio momentų. Vis dėlto pasitaiko atvejų, kai tarplaikiai tarpusavyje yra koreliuoti. Siekiant sukurti tikslesnį modelį, svarbu įvertinti tokias tarplaikių tarpusavio priklausomybes. Tai galima padaryti naudojant Markovo atvykimų procesus.

Dėl sudėtingo šių procesų nagrinėjimo analiziniu būdu, kanoninės jų formos surastos tik procesams su 2 fazėmis. Markovo atvykimo proceso parametrų paieška, lyginant su fazinių skirstinių parametrų paieška, yra sudėtingesnė, nes papildomai turi būti įvertinta ir tarplaikių koreliacija.

Markovo atvykimų procesams aktualesnė problema yra spartesnė parametrų paieška. Kad būtų tiksliau įvertinta tarplaikių koreliacija, parametrų paieškai turi būti naudojama kiek galima ilgesnė tarplaikių seka. Naudojant ilgesnę tarplaikių seką, parametrų paieškos laikas didėja. Dėl to skaičiavimus reikia išlygiagretinti. Deja, mokslinėje literatūroje nėra nagrinėjamas parametrų išlygiagretinimo klausimas. Siekdami užpildyti šią spragą, formalizavome ir palyginome įvairius maksimalaus tikėtimumo metodo algoritmo išlygiagretinimo variantus. Dalis idėjų buvo paimtos iš giminingos srities – tolydaus laiko paslėptųjų Markovo modelių parametrų paieškos algoritmų. Atlikome algoritmų sudėtingumo teorinę analizę. Algoritmus kūrėme C++ kalba, lygiagretiems skaičiavimams naudojome vaizdo procesorių. Rezultatus apibendrinome ir pateikėme tirtų algoritmų taikymo rekomendacijas.

Tyrimo objektas

Markovo procesai: faziniai skirstiniai ir Markovo atvykimų procesai.

Tyrimo tikslas

Ištirti aproksimavimą faziniais skirstiniais, turinčiais pakankamą parėjimų skaičių. Ištirti Markovo atvykimų proceso ir nutrūkstančio Markovo atvykimų proceso parametrų paieškos išlygiagretinimą.

Tyrimo uždaviniai

1. Sukurti fazinių skirstinių struktūrų generavimo algoritmą.
2. Tirti teigiamo atsitiktinio dydžio skirstinio aproksimavimą įvairių struktūrų klasių faziniais skirstiniais.

3. Formalizuoti Markovo atvykimų proceso parametru paieškos lygiagrečius algoritmus.
4. Pritaikyti maksimalaus tikėtinumo metodą nutrūkstančio Markovo atvykimų proceso parametru paieškai.
5. Sukurti Markovo atvykimų ir nutrūkstančių Markovo atvykimų procesų parametru paieškos lygiagrečius algoritmus. Spartesniai skaičiavimui naudoti grafinį procesorių.

Tyrimų metodai ir programinė įranga

1. Aprašomoji statistika naudojama tyrimų rezultatams analizuoti.
2. Visi skaičiavimai atlikti savarankiškai parašytais programomis C++ kalba. Naudotos kelios papildomos bibliotekos: EIGEN, BOOSTS.
3. Programos (veikimui pasitelkiant vaizdo procesorių) buvo sukurtos naudojant tam tikslui skirtą CUDA biblioteką.

Darbo mokslinis naujumas ir praktinė svarba

Irodyta, jog bet kuriam n -tosios eilės faziniam skirstiniui nusakyti pakanka $p = 2n - 1$ parametru. Tačiau, norint fazinius skirstinius panaudoti praktiškai, reikia turėti jų matricinės formos reprezentacijas. Matricinės formos reprezentaciją patogiau charakterizuoti perėjimų tarp būsenų skaičiumi m (maksimalus nepriklausomų parametru skaičius $p = m - 1$). Visa galimų fazinių skirstinių aibė gali būti padengta matricinės formos reprezentacijomis su $m \leq 2n + \Delta M_n$ perėjimais.

Kanoninės formos (reprezentacijų parametrizacijos) yra surastos tik eilių $n = 2, 3$ faziniams skirstiniams. Tad maksimalus papildomų perėjimų skaičius ΔM_n bendru atveju nėra žinomas (žinoma, tik $\Delta M_2 = 0, \Delta M_3 = 1$).

Mes tikimės, jog pakanka naudoti reprezentacijas be papildomų perėjimų (t. y. $m = 2n$ ir $p = 2n - 1$). Suformuluojame hipotezę, jog bet kuriam faziniam skirstiniui, kurio matricinės formos reprezentacija turi papildomų perėjimų (t. y. $\Delta m_n > 0$, kur $\Delta m_n \leq \Delta M_n$), galima surasti reprezentacijų be papildomų perėjimų ($m = 2n$) seką, kurios faziniai skirstiniai konverguoja į pradinį skirstinį. Analogiška hipotezė nėra suformuluota mokslinėje literatūroje. Hipotezę pagrindžiame empiriškai, tokiam atvejui, kai $n = 4$. Pasinaudoję mūsų sukurtu struktūrų generavimo algoritmu, gauname visą

fazinių skirstinių erdvę padengiančia struktūrų aibe. Šiomis struktūromis aproksimuojame literatūroje pateiktus skirstinius, remdamiesi gautais rezultatais pagrindžiamo hipotezę.

Mokslinėje literatūroje nagrinėjamas fazinių skirstinių efektyvios parametru paieškos klausimas naudojant tam tikrą struktūrų klasę. Tačiau pasirinkta viena struktūrų klasė negali padengti visos fazinių skirstinių aibės. Siekiant ieškoti parametru optimaliausiu būdu, reikia nagrinėti visas galimas struktūras. Mokslinėje literatūroje analiziniu būdu tai padaryta tik trečios eilės faziniams skirstiniams – surastos jų kanoninės formos. Ketvirtos eilės fazinių skirstinių analizinis tyrimas yra pernelyg komplikuoatas. Dėl to kaip alternatyvą pasiūlėme paprastesnį struktūrų generavimo algoritmą. Negalime garantuoti, jog algoritmo gauta struktūrų aibė yra minimali, tačiau ji gali būti naudojama išsamesniems tyrimams.

Tiesa, mūsų struktūrų generavimo algoritmas nėra efektyvus ir sunkiai praktiškai pritaikomas aukštesnių eilių ($n > 4$) fazinių skirstinių struktūrų generavimui dėl didelio pradinių struktūrų skaičiaus. Dėl to papildomai tiriame aproksimavimą faziniais skirstiniais, turinčiais atsitiktinai sugeneruotą struktūrą su $m = 2n$ perėjimais. Atvejui $n = 4$ rezultatai palyginami su analogiškais rezultatais naudojant sugeneruotą struktūrų aibę, kaip atspirties tašką. Mokslinėje literatūroje analogiško tyrimo rezultatai nėra pateikti, to priežastis – iki šiol nebuvo sukurtas struktūrų generavimo algoritmas.

Mokslinėje literatūroje nėra pateikta tyrimų, kuriais palyginamas aproksimavimas retomis (su $m = 2n$ perėjimais) struktūromis ir tankia struktūra, naudojant EM metodą. Darbe nustatėme aproksimavimo ypatumus abiem atvejais. Gauti rezultatai yra naudingi dėl jų taikymo. Tanki struktūra yra lanksti, tačiau dėl perteklinio parametru skaičiaus pasiekiamas labai lėtas konvergavimas dar nepasiekus labiausiai tikėtinu parametru. Aproksimuojant retų struktūrų aibę tikėtina, jog bent viena struktūra konverguos sparčiau ir bus gauti labiau tikėtini parametrai su mažesniu iteracijų skaičiumi. Be to, retų struktūrų parametru paieška gali būti atliekama lygiagrečiai, atskirai kiekvienai struktūrai. Tolesniems tyrimams galima kelti klausimą, kaip parinkti tinkamą, greitai konverguojančią struktūrą.

Markovo atvykimų proceso struktūrų analizė yra sudėtinga problema, kurios išsamiau nenagrinėjame. Dėmesį skiriame greitesnei Markovo atvykimų proceso parametru paieškai. Maksimalaus tikėtinumo metodo al-

goritmas dėl rekurentinių skaičiavimo formulių yra nuoseklus. Tačiau įvedus papildomus skaičiavimus algoritmą galima išlygiagretinti. Nepaisant to, mokslinėje literatūroje nėra aptariamas MAP procesų parametų paieškos išlygiagretinimas. Markovo atvykimų procesų klasė, nusakoma Erlango skirstinių mišiniu (ER-CHMM), atitinka tolydaus laiko paslėptąjį Markovo modelį (CHMM). Šio modelio parametrai gali būti ieškomi Baum-Welch algoritmu. Išgilinus į detales paaiškėjo, jog Baum-Welch algoritmo principus galima pritaikyti ir Markovo atvykimų proceso parametų paieškos išlygiagretinimui. Algoritmą galima formalizuoti įvairiais būdais, nuo kurių priklauso jo veikimo charakteristikos. Išskyrėme kelis algoritmų variantus (įskaitant ir vieną mūsų pasiūlytą) ir juos palyginome.

Nutrūkstantys Markovo atvykimų procesai (TMAP) nėra taip plačiai išanalizuoti kaip MAP procesai. Mokslinėje literatūroje nėra aptartas EM metodo taikymas TMAP proceso parametų paieškai. Mūsų darbo indėlis – EM metodo algoritmo išvedimas šio proceso parametų paieškai atlikti. EM metodą taikome panašiai kaip ir fazinių skirstinių (Markovo atvykimų procesų parametų paieškos atveju), skirtumas tik tas, jog įvertinamos atsitiktinio ilgio tarplaikių serijos. Ieškant parametų, tarpiniai skaičiavimai serijoms gali būti atlikti nepriklausomai, tad galimas tiesioginis išlygiagretinimas (t. y. neįvedant bendrą sudėtingumą padidinančių papildomų skaičiavimų). Sukūrėme nuoseklų bei lygiagretų algoritmus ir juos palyginome.

Pastaruoju metu sparčiai tobulinami vaizdo procesoriai (GPU). Be to, sukurtos programinės priemonės matematiniams skaičiavimams atlikti. Tačiau mokslinėje literatūroje neaptariamas GPU taikymas spartesnei MAP ir TMAP procesų parametų paieškai. Dėl to šių procesų parametų paieškos lygiagrečius algoritmus sukūrėme naudodami vaizdo procesorių. Darbo metu nustatėme išskylančias skaitinio skaičiavimo problemas ir suradome būdus, kaip jas išspręsti.

Praktinė disertacijos darbo reikšmė yra hipotezės apie fazinių skirstinių struktūras empirinis pagrindimas ir Markovo atvykimų procesų parametų paieškos išlygiagretinimas. Naudodami fazinius skirstinius baigtinės aptarnavimo eilės modelyje, palyginome, kaip skiriasi aproksimavimas tančia ir retomis struktūromis.

Disertacijos gynimui pateikiama:

1. Fazinių skirstinių struktūrų generavimo algoritmas.

2. Hipotezės, jog faziniai skirstiniai su $m = 2n$ perėjimais padengia beveik visą erdvę, empirinis pagrindimas, tokiam atvejui, kai $n = 4$.
3. Aproximavimo faziniais skirstiniais, turinčiais retą (su $m = 2n$ perėjimais) bei tankią struktūrą, palyginimas.
4. Markovo atvykimų proceso parametrų paieškos lygiagretūs algoritmai.
5. Maksimalaus tikėtinumo metodo pritaikymas nutrūkstančio Markovo atvykimų proceso parametrų paieškai.
6. Išlygiagretintų MAP, TMAP procesų parametrų paieškos algoritmų rezultatai. Eksperimentinių skaičiavimų naudojant vaizdo procesorių (GPU) rezultatai.

Darbo rezultatų aprobavimas

Disertacijos rezultatai publikuoti dvejuose moksliniuose (ISI) žurnaluose ir pristatyti dvejose tarptautinėse konferencijose bei keliose nacionalinėse konferencijose. Fazinių skirstinių struktūrų generavimo algoritmas, kartu su jo panaudojimo rezultatais aproksimuojant literatūroje pateiktus skirstinius, pristatytas ISI straipsnyje „On Structured Initial Solution Generation for Phase-Type Fitting with EM Method“. MAP procesų parametrų paieškos lygiagretūs algoritmai publikuoti ISI straipsnyje „Parallel Algorithms for Fitting Markov Arrival Processes“. Sudaryti TMAP procesų parametrų paieškos algoritmai ir jų rezultatai pristatyti tarptautinėje ASMTA16 konferencijos pranešime „Efficient Implementations of the EM-algorithm for Transient Markov Arrival Processes“. Fazinių skirstinių naudojimas konstruojant baigtinį aptarnavimo eilės modelį pristatytas IARA2014 konferencijos pranešime „Software Reliability Markovian Model Based on Phase-type Distributions“.

Disertacijos struktūra

Disertaciją sudaro įvadas, 4 pagrindiniai skyriai, išvados, literatūros ir autoriaus publikacijų sąrašai. Disertacijos teksto apimtis – 148 puslapiai. Tekste yra 54 iliustracijos, 44 lentelės ir 94 cituojamų šaltinių sąrašas (iš jų 10 šaltinių paskelbti 2014–2018 metais).

1. LITERATŪROS APŽVALGA

Apžvelgsime literatūrą, susijusią su faziniais skirstiniais ir Markovo atvykimų procesais, kuri naudojama disertacijos temai vystyti. Išsamesnė darbo konteksto apžvalga pateikta disertacijoje.

1.1. Faziniai skirstiniai

Stebimi stochastinio reiškinio generuojamų įvykių laiko momentai. Laiko tarpai tarp šių momentų vadinami tarplaikiais ir jų skirstinys interpretuojamas kaip pereinamos Markovo grandinės gyvavimo laiko skirstinys.

1 apibrėžimas. [1] *Atsitiktinio laiko, per kurį pasiekama pereinamos Markovo grandinės absorbuojanti būseną, skirstinys vadinamas faziniu skirstiniu.*

Anglų kalba pateiktoje literatūroje fazinį skirstinį įprasta nurodyti santrumpa PH (angl. *Phase-type*).

Fazinis skirstinys išsamiai nurodomas šios Markovo grandinės parametrais, intensyvumų matrica Q ir pradinių tikimybių vektoriumi α .

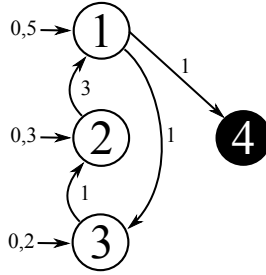
2 apibrėžimas. *Fazėmis vadinamos Markovo grandinės pereinamos būsenos.*

3 apibrėžimas. *Fazinio skirstinio eilė n vadinamas Markovo grandinės pereinamų būsenų skaičius.*

1 pavyzdys. Pateikta pereinama Markovo grandinė su parametrais,

$$Q = \begin{bmatrix} -2 & 0 & 1 & 1 \\ 3 & -3 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \alpha = [0,5 \quad 0,3 \quad 0,2 \quad 0]$$

nurodo trečios eilės fazinį skirstinį ir yra pavaizduota paveiksle 1.



1 pav. Pereinama Markovo grandinė.

Pirma, antra ir trečia būsenos yra pereinamos, o ketvirta – absorbuojanti. △

Intensyvumų matrica Q turi savybę – eilutės elementų suma lygi nuliui. Patogu matricą išreikšti blokais

$$Q = \begin{bmatrix} D_0 & d_1 \\ \mathbf{0} & 0 \end{bmatrix}.$$

4 apibrėžimas. *Parametrų rinkinys (α, D_0) vadinamas fazinio skirstinio matricinės formos reprezentacija.*

Paprastai teigiama, kai $\alpha_{n+1} = 0$.

Parametrų (α, D_0) pakanka fazinių skirstinių atitinkančiai Markovo grandinei nusakyti, nes $d_1 = -D_0 \mathbf{1}$. Čia $\mathbf{1}$ yra atitinkamo dydžio vektorius (stulpelis) su vienetiniais elementais.

1 teiginys. [1] *Fazinio skirstinio funkcija*

$$F(x) = 1 - \alpha e^{D_0 x} \mathbf{1},$$

ir tankio funkcija

$$f(x) = \alpha e^{D_0 x} d_1.$$

2 pavyzdys. Pagal apibrėžimą 4, pavyzdyje 1 pateikta Markovo grandinė

yra nusakoma tokiais (α, D_0) parametrais

$$D_0 = \begin{bmatrix} -2 & 0 & 1 \\ 3 & -3 & 0 \\ 0 & 1 & -1 \end{bmatrix}, \quad \alpha = [0,5 \quad 0,3 \quad 0,2]. \quad \triangle$$

5 apibrėžimas. *Fazinio skirstinio struktūra vadinamas parametru rinkinys $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)$, kuriuo nurodomi galimi perėjimai Markovo grandinėje.*

Struktūroje $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)$ galimas perėjimas nurodomas vienetu, negalimas – nuliu. Struktūra yra charakterizuojama galimų perėjimų skaičiumi m

$$m = \sum_{j=1}^n \dot{\alpha}_j + \sum_{i=1}^n \sum_{j=1}^n \dot{D}_{0i,j} + \sum_{i=1}^n \dot{d}_{1i}.$$

3 pavyzdys. Pavyzdžiuose 1, 2 pateikto fazinio skirstinio struktūra nurodoma taip

$$(\dot{\alpha}, \dot{D}_0, \dot{d}_1) = \left([1 \quad 1 \quad 1], \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right).$$

Ši struktūra turi $m = 7$ perėjimus. △

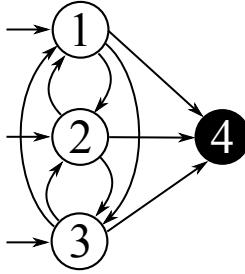
6 apibrėžimas. *Fazinio skirstinio tankia struktūra vadinama tokia struktūra, kurioje yra visi galimi perėjimai.*

Tanki struktūra turi maksimalų perėjimų skaičių, kuris yra lygus $m = n^2 + n$.

4 pavyzdys. Tankios struktūros, pavaizduotos paveiksle 2, parametrai

$$(\dot{\alpha}, \dot{D}_0, \dot{d}_1) = \left([1 \quad 1 \quad 1], \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right).$$

Perėjimų skaičius $m = n^2 + n = 12$. Matricos \dot{D}_0 pagrindinės įstrižainės elementai visada yra nuliniai, nes tolydaus laiko Markovo grandinėje perėjimas į tą pačią būseną nėra galimas.



2 pav. Pereinama Markovo grandinė, atitinkanti trečios eilės fazinio skirstinio tankią struktūrą.

△

7 apibrėžimas. *Matricinės formos reprezentacijos parametrizacija \mathbf{f} vadinama nusakytos Markovo grandinės perėjimų funkcinė priklausomybė nuo p parametrų.*

Kitaip tariant, $(\alpha, D_0) = \mathbf{f}_{(\dot{\alpha}, \dot{D}_0, \dot{d}_1)}(\mathbf{p})$, kur \mathbf{p} yra parametrų reikšmių vektorius.

Kiekvienai struktūrai galima sukurti itin daug parametrizacijų. Parinkus parametrų reikšmes gaunama konkreti matricinės formos reprezentacija.

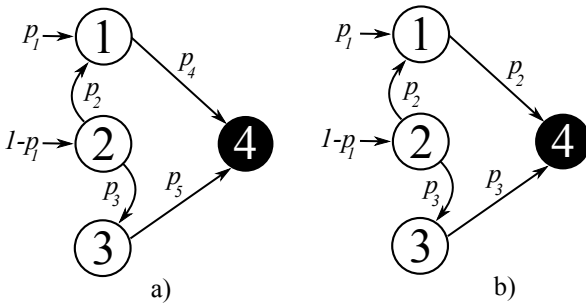
8 apibrėžimas. *Pilnaja matricinės formos reprezentacijos parametrizacija vadinama parametrizacija, turinti $p = m-1$ nepriklausomus parametrus.*

Apibrėžime 8 kalbama apie parametrų skaičių, kuris yra vienetu mažesnis nei perėjimų skaičius. Taip yra todėl, kad visada turime vieną priklausomą perėjimą, nulemtą pradinių tikimybių vektoriaus $\alpha \mathbb{1} = 1$.

5 pavyzdys. Struktūros ($m = 6$), nusakytos parametrais

$$(\dot{\alpha}, \dot{D}_0, \dot{d}_1) = \left([1 \ 1 \ 0], \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right),$$

dvi parametrizacijos pavaizduotos paveiksle 3. Parametrizacija (a) turi $p = 5$, o (b) turi $p = 3$ parametrus. Parametrizacija (a) yra pilnoji.



3 pav. Pavyzdyje 5 pateiktos struktūros dvi parametrizacijos.

△

1 teorema. [2] *Faziniam skirstiniui nurodyti pakanka $p = 2n - 1$ parametru.*

Iš teoremos 1 daroma išvada, kad pilnoji tankios struktūros parametrizacija su $p = n^2 + n - 1$ parametrais yra perteklinė. Tariaama, jog \mathcal{P}_n yra parametrizacijų su $p = 2n - 1$ parametrais aibė.

9 apibrėžimas. *n -tosios eilės fazinio skirstinio kanonine forma vadinama baigtinė aibė parametrizacijų \mathcal{C}_n su $p = 2n - 1$ parametrais, kuriomis vieninteliu būdu galima reprezentuoti kiekvieną galimą n -tosios eilės fazinį skirstinį.*

Turime $\mathcal{C}_n \subseteq \mathcal{P}_n$. Kanoninių parametrizacijų aibę \mathcal{C}_n tikslinga išreikšti kelių aibių sąjunga

$$\mathcal{C}_n = \mathcal{C}_n^{(2n)} \cup \dots \cup \mathcal{C}_n^{(M_n)},$$

kur $\mathcal{C}_n^{(m)}$ yra n -tosios eilės faziniai skirstiniai su m perėjimais, o M_n – maksimalus perėjimų skaičius. Čia $M_n = 2n + \Delta M_n$, ΔM_n – maksimalus papildomų perėjimų skaičius.

Kanoninės formos yra surastos tik antros ir trečios eilių faziniams skirstiniams [3, 4].

2 teorema. [3] *Trečios eilės fazinis skirstinys turi kanoninę formą, nusakytą trimis parametrizacijomis*

$$\alpha = [p_1 \quad p_2 \quad 1 - p_1 - p_2], \quad \mathbf{D}_0 = \begin{bmatrix} -p_3 & 0 & p_6 \\ p_4 & -p_4 & 0 \\ 0 & p_5 & -p_5 \end{bmatrix},$$

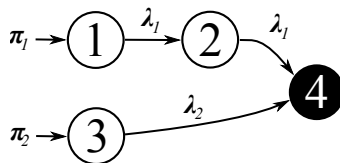
kai a) $p_6 = 0$, b) $p_3 = p_4$, c) $p_2 = 0$.

Parametrizacijos (a) ir (c) turi po $m = 6$ perėjimus, o parametrizacija (b) turi vieną papildomą perėjimą, t. y. $m = 7$, tad maksimalus papildomų perėjimų skaičius $\Delta M_3 = 1$.

Viena iš populiarių neciklinių struktūrų, turinčių fazinių skirstinių, klasė – Erlango hiperskirstiniai [5]. Erlango hiperskirstinys – tai R Erlango skirstinių mišinys. Individualus i -tasis skirstinys vadinamas Erlango šaka. Su π_i tikimybe pasirenkama i -toji Erlango šaka, kurios perėjimų tarp būsenų intensyvumas λ_i . Erlango skirstinių eilės nurodomos vektoriumi $\mathbf{r} = [r_1 \quad \dots \quad r_R]$. Paprastai tariama, jog \mathbf{r} yra fiksuotas, tuomet Erlango hiperskirstinys nurodomas parametrais (π, λ) . Iš \mathbf{r} galima gauti Erlango šakų pradinių būsenų numerius

$$s_i = 1 + r_1 + \dots + r_{i-1}.$$

6 pavyzdys. Panagrinėkime Erlango hiperskirstinį, kurio struktūra pateikta parametru $\mathbf{r} = [2 \quad 1]$. Skirstinį atitinkanti Markovo grandinė pavaizduota paveiksle 4.



4 pav. Pavyzdyje 6 pateikto Erlango hiperskirstinio Markovo grandinė.

Atitinkami fazinio skirstinio parametrai (α, D_0) yra

$$\alpha = [\pi_1 \quad 0 \quad \pi_2], D_0 = \begin{bmatrix} -\lambda_1 & \lambda_1 & 0 \\ 0 & -\lambda_1 & 0 \\ 0 & 0 & -\lambda_2 \end{bmatrix}. \quad \triangle$$

1.2. Fazinio skirstinio parametrų paieška maksimalaus tikėtinumo metodu

Stebimas atsitiktinis dydis suprantamas kaip pereinamos Markovo grandinės gyvavimo trukmės $\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_T)$. Konkrečios gyvavimo trukmės žymimos kaip $\mathbf{x} = (x_1, x_2, \dots, x_T)$. Užduotis yra surasti šios Markovo grandinės labiausiai tikėtinus parametrus (t. y. fazinį skirstinį, nusakomą parametrais (α, D_0)), su kuriais gyvavimo trukmių skirstinys kiek įmanoma labiau atitinka stebimo atsitiktinio dydžio skirstinį. Tai galima padaryti maksimizuojant parametrų tikėtinumo funkcijos reikšmę

$$\mathcal{L}(\alpha, D_0 | \mathcal{X}) = \prod_{k=1}^T f(x_k) = \prod_{k=1}^T \alpha e^{D_0 x_k} \mathbf{d}_1 \quad (1)$$

pritaikant maksimalaus tikėtinumo metodą [6].

Markovo grandinės gyvavimo trukmės \mathbf{x} yra vienintelė žinoma informacija. Norint vienareikšmiškai nustatyti Markovo grandinės parametrus, reikia turėti visą informaciją apie tai, kaip buvo sugeneruotos \mathbf{x} . Markovo procesas $\mathcal{J}_t^{(k)}$, generuojantis atsitiktinę gyvavimo trukmę \mathcal{X}_k , visiškai charakterizuojamas lankomų būsenų procesu

$$\mathcal{I}_0^{(k)}, \dots, \mathcal{I}_{\mathcal{M}^{(k)}-1}^{(k)}$$

ir atsitiktinėmis laiko trukmėmis

$$\mathcal{S}_0^{(k)}, \dots, \mathcal{S}_{\mathcal{M}^{(k)}-1}^{(k)},$$

kurios nusako praleistą laiką būsenoje kiekvieno apsilankymo metu. $\mathcal{M}^{(k)}$ yra perėjimų skaičius tarp būsenų prieš patenkant į absorbuojančią būseną. Taip pat turime

$$\mathcal{X}_k = \mathcal{S}_0^{(k)} + \dots + \mathcal{S}_{\mathcal{M}^{(k)}-1}^{(k)}.$$

Išsami informacija \mathcal{Y} apie konkrečias T realizacijas nurodoma taip:

$$\mathbf{y} = \left(i_0^{(1)}, \dots, i_{m^{(0)}-1}^{(1)}, s_0^{(1)}, \dots, s_{m^{(0)}-1}^{(1)}, \dots, i_0^{(T)}, \dots, i_{m^{(T)}-1}^{(T)}, s_0^{(T)}, \dots, s_{m^{(T)}-1}^{(T)} \right).$$

Remiantis turimomis gyvavimo trukmėmis \mathbf{x} , gauti labiau tikėtinus parametrus galima toliau nurodytu būdu. Su pasirinktais pradiniais parametrais sugeneruojamas pakankamas realizacijų kiekis. Atrinkta iš realizacijų, kurių gyvavimo trukmės sutampa su duotomis \mathbf{x} . Atrinkta išsami informacija naudojama labiau tikėtiniems parametrams apskaičiuoti. Vėl generuojamos realizacijos su patikslintais parametrais, procedūra tęsiama tol, kol parametų tikėtinumai nustoja didėti.

Iš tiesų parametų apskaičiavimui pakanka visą informaciją \mathcal{Y} charakterizuojančių statistikų

$$\mathcal{H} = \{ \mathcal{B}_i, \mathcal{N}_{i,j}, \mathcal{Z}_i \},$$

kur \mathcal{B}_i – realizacijų prasidedančių būsenoje i skaičius

$$\mathcal{B}_i = \sum_{k=1}^T \mathcal{B}_i[k], \quad \mathcal{B}_i[k] = \mathbb{1}_{\{I_0^{(k)}=i\}},$$

$\mathcal{N}_{i,j}$ – perėjimų iš būsenos i į būseną j skaičius

$$\mathcal{N}_{i,j} = \sum_{k=1}^T \mathcal{N}_{i,j}[k], \quad \mathcal{N}_{i,j}[k] = \sum_{l=0}^{\mathcal{M}^{(k)}} \mathbb{1}_{\{I_l^{(k)}=i, I_{l+1}^{(k)}=j\}}$$

ir \mathcal{Z}_i – visas laikas praleistas būsenoje i

$$\mathcal{Z}_i = \sum_{k=1}^T \mathcal{Z}_i[k], \quad \mathcal{Z}_i[k] = \sum_{l=0}^{\mathcal{M}^{(k)}} \mathcal{S}_l^{(k)} \mathbb{1}_{\{I_l^{(k)}=i\}}.$$

Konkretus statistikų \mathcal{H} įvertis \mathbf{h} duotiems parametrams $(\boldsymbol{\alpha}, \mathbf{D}_0)$ ir gyvavimo trukmėms \mathbf{x} randamas pagal toliau pateiktas formules. Realizacijų skaičius, prasidedančių būsenoje i , įvertis

$$b_i = \mathbb{E}[\mathcal{B}_i | \boldsymbol{\mathcal{X}}] = \sum_{k=1}^T \mathbb{E}[\mathcal{B}_i[k] | \mathcal{X}_k] = \sum_{k=1}^T \frac{\boldsymbol{\alpha}_i \mathbf{e}_i^\top e^{\mathbf{D}_0 \mathbf{x}_k} \mathbf{d}_1}{\boldsymbol{\alpha} e^{\mathbf{D}_0 \mathbf{x}_k} \mathbf{d}_1}, \quad (2)$$

peršokimų iš būsenos i į būseną j skaičiaus įvertis, $i = 1, \dots, n, j = 1, \dots, n, i \neq j$

$$\begin{aligned} n_{i,j} &= \mathbb{E}[\mathcal{N}_{i,j} \mid \mathcal{X}] = \sum_{k=1}^T \mathbb{E}[\mathcal{N}_{i,j}[k] \mid \mathcal{X}_k] \\ &= \sum_{k=1}^T \frac{\int_0^{x_k} \alpha e^{D_0 x_k} e_i D_{0i,j} e_j^\top e^{D_0(x_k-u)} \mathbf{d}_1 du}{\alpha e^{D_0 x_k} \mathbf{d}_1}, \end{aligned} \quad (3)$$

peršokimų iš būsenos i į absorbuojančią būseną $n + 1$ skaičiaus įvertis

$$\begin{aligned} n_{i,n+1} &= \mathbb{E}[\mathcal{N}_{i,n+1} \mid \mathcal{X}] = \sum_{k=1}^T \mathbb{E}[\mathcal{N}_{i,n+1}[k] \mid \mathcal{X}_k] \\ &= \sum_{k=1}^T \frac{\alpha e^{D_0 x_k} e_i \mathbf{d}_{1i}}{\alpha e^{D_0 x_k} \mathbf{d}_1}, \end{aligned} \quad (4)$$

praleisto laiko būsenoje i įvertis

$$\begin{aligned} z_i &= \mathbb{E}[\mathcal{Z}_i \mid \mathcal{X}] = \sum_{k=1}^T \mathbb{E}[\mathcal{Z}_i[k] \mid \mathcal{X}_k] \\ &= \sum_{k=1}^T \frac{\int_0^{x_k} \alpha e^{D_0 x_k} e_i e_j^\top e^{D_0(x_k-u)} \mathbf{d}_1 du}{\alpha e^{D_0 x_k} \mathbf{d}_1}. \end{aligned} \quad (5)$$

Tuomet labiausiai tikėtini parametrai nežinomos informacijos įverčio $\mathbf{h} = \{b_i, n_{i,j}, z_i\}$ atžvilgiu yra apskaičiuojami pagal formules [7]

$$\begin{aligned} \alpha_i &= \frac{b_i}{T}, & i &= 1, \dots, n \\ D_{0i,j} &= \frac{n_{i,j}}{z_i}, & i, j &= 1, \dots, n, i \neq j \\ \mathbf{d}_{1i} &= \frac{n_{i,n+1}}{z_i}, & i &= 1, \dots, n \\ D_{0i,i} &= -\mathbf{d}_{1i} - \sum_{j=1, j \neq i}^n D_{0i,j}, & i &= 1, \dots, n. \end{aligned} \quad (6)$$

Maksimalaus tikėtimumo metodo taikymas parametų paieškai pavaizduotas 1 algoritme.

1 algoritmas. Algoritmas fazinio skirstinio parametrų paieškai maksimalaus tikėtinumo metodu.

procedure *PH-EM-fitting* ($\hat{\alpha}, \hat{D}_0, \hat{d}_1$)

1. Pagal struktūrą $(\hat{\alpha}, \hat{D}_0, \hat{d}_1)$ sugeneruojami parametrai (α, D_0)
2. Apskaičiuojamas parametrų (α, D_0) tikėtinumas pagal (1)
3. **repeat**
4. Apskaičiuojamas įvertis h pagal (2), (3), (4), (5)
5. Apskaičiuojami nauji parametrai (α, D_0) pagal (6)
6. Apskaičiuojamas parametrų (α, D_0) tikėtinumas pagal (1)
7. **until** kol parametrų tikėtinumas didėja
8. **return** (α, D_0)

end procedure

Išsamios informacijos \mathcal{Y} pakankamų charakteristikų \mathcal{H} įverčio h apskaičiavimas yra sudėtingas – skaičiuojamos matricos eksponentės, sąsukos integralai. Pakankamų charakteristikų įvertinimas sprendžiant diferencialinių lygčių sistemą Rungės ir Kutos ketvirtos eilės metodu pateiktas [8]. Randomizacijos metodikos skaičiavimams pritaikymas pateiktas [9].

Tačiau Markovo grandinėse, turinčiose tam tikrą struktūrą, vykstančius procesus galima charakterizuoti paprasčiau. Viena iš tokių struktūrų klasių – Erlango hiperskirstinys. Yra įrodyta, jog Erlango hiperskirstiniu galima aproksimuoti bet kurį teigiamą atsitiktinio dydžio skirstinį [10].

1.3. Erlango hiperskirstinio parametrų paieška maksimalaus tikėtinumo metodu

Šiuo atveju Markovo procesas $\mathcal{J}_t^{(k)}$, generuojantis atsitiktinę gyvavimo trukmę \mathcal{X}_k , visiškai charakterizuojamas pradine būsena $\mathcal{I}_0^{(k)}$. Turint konkrečią pradinę būseną $i_0^{(k)}$, kuri buvo pasirinkta generuojant x_k ir \mathbf{r} , galima atkurti visą informaciją \mathbf{y} apie procesą. Dėl to užtenka, kad visą informaciją \mathcal{Y} nusakanti statistika yra $\mathcal{H} = (\mathcal{Q}_i[k])$, kur $\mathcal{Q}_i[k]$ – indikatorius, nurodantis ar \mathcal{X}_k buvo sugeneruotas i -tosios Erlango šakos

$$\mathcal{Q}_i[k] = \mathbb{1}_{\{\mathcal{I}_0^{(k)} = s_i\}}.$$

Įvertis \mathbf{h} surandamas pagal formulę

$$\begin{aligned} q_i[k] &= \mathbb{E}[Q_i[k] \mid \mathcal{X}_k] \\ &= \mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i \mid \mathcal{X}_k) = \frac{\mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i, \mathcal{X}_k)}{\mathbb{P}(\mathcal{X}_k)} = \frac{\boldsymbol{\pi}_i f_i(x_k)}{\sum_{j=1}^R \boldsymbol{\pi}_j f_j(x_k)}, \end{aligned} \quad (7)$$

čia $f_i(x)$ yra i -tosios Erlango šakos tankis

$$f_i(x) = \frac{(\boldsymbol{\lambda}_i x)^{r_i-1}}{(r_i-1)!} \boldsymbol{\lambda}_i e^{-\boldsymbol{\lambda}_i x}.$$

Toliau pateikiami įverčio \mathbf{h} atžvilgiu maksimalaus tikėtimumo parametrai

$$\boldsymbol{\pi}_i = \frac{1}{T} \sum_{k=1}^T q_i[k], \quad (8)$$

$$\boldsymbol{\lambda}_i = \frac{r_i \sum_{k=1}^T q_i[k]}{\sum_{k=1}^T q_i[k] x_k}. \quad (9)$$

Tikėtimumo funkcijos išraiška taip pat supaprastėja, ji yra

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\lambda} \mid \mathcal{X}) = \prod_{k=1}^T \sum_{i=1}^R \boldsymbol{\pi}_i f_i(x_k). \quad (10)$$

Toliau pateikiamas algoritmas 2, kuriuo apibendrinamas maksimalaus tikėtimumo metodo taikymas.

2 algoritmas. Algoritmas Erlango hiperskirstinio parametrų paieškai maksimalaus tikėtimumo metodu.

procedure *HerD-EM-fitting*(\mathbf{x}, \mathbf{r})

1. Atsižvelgiant į \mathbf{r} sugeneruojami parametrai $(\boldsymbol{\pi}, \boldsymbol{\lambda})$
2. Apskaičiuojamas $(\boldsymbol{\pi}, \boldsymbol{\lambda})$ tikėtimumas pagal (10)
3. **repeat**
4. Apskaičiuojamas įvertis \mathbf{h} pagal (7)
5. Apskaičiuojami nauji parametrai $(\boldsymbol{\pi}, \boldsymbol{\lambda})$ pagal (8), (9)
6. Apskaičiuojamas $(\boldsymbol{\pi}, \boldsymbol{\lambda})$ tikėtimumas pagal (10)
7. **until** kol parametrų tikėtimumas didėja
8. **return** $(\boldsymbol{\pi}, \boldsymbol{\lambda})$

end procedure

Šis metodas gali būti taikomas ir tuo atveju, kuomet pateiktos ne gyvavimo trukmės, bet jų skirstinys. Tokiu atveju diskretizavus skirstinį gaudamos gyvavimo trukmės x_k ir jas atitinkančios tikimybės w_k . Tikimybės w_k kaip svoriai gali būti įtrauktos į pakankamų charakteristikų įverčio h skaičiavimus.

1.4. Markovo atvykimų procesai

Generuojant gyvavimo trukmę pagal fazinį skirstinį pradinė būsena pasirenkama su tomis pat tikimybėmis α . Dėl to bet kuri gyvavimo trukmė nepriklauso nuo prieš tai sugeneruotos gyvavimo trukmės.

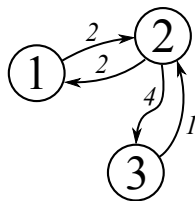
Tarkime, jog stebimas reiškinys, kuris atsitiktiniais laiko momentais t_0, t_1, \dots generuoja įvykius. Galima surasti tarplaikius (angl. *inter-arrivals*) $x_k = t_k - t_{k-1}$ tarp įvykių, kai $k = 1, 2, \dots$. Jeigu tarplaikis x_k nepriklauso nuo prieš jį buvusio tarplaikio x_{k-1} , galima juos interpretuoti kaip pereinamos Markovo grandinės gyvavimo trukmes ir naudoti fazinius skirstinius. Vis dėlto, jei priklausomybė egzistuoja, tarplaikius tikslinga interpretuoti kaip laiko tarpus tarp tam tikrų Markovo grandinėje įvykstančių perėjimų.

Įprastoje Markovo grandinėje visi perėjimai traktuojami vienodai, tačiau šiuo atveju reikia papildomai nurodyti perėjimus, kurie generuoja įvykius (angl. *arrivals*). Įprastų perėjimų intensyvumų matrica nurodoma kaip D_0 , įvykius generuojančių intensyvumų matrica – D_1 .

7 pavyzdys. Turime Markovo grandinę su trimis būsenomis

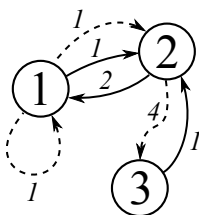
$$\alpha = [0,3 \quad 0 \quad 0,7], \quad Q = \begin{bmatrix} -2 & 2 & 0 \\ 2 & -6 & 4 \\ 0 & 1 & -1 \end{bmatrix},$$

kuri pavaizduota paveiksle 5.



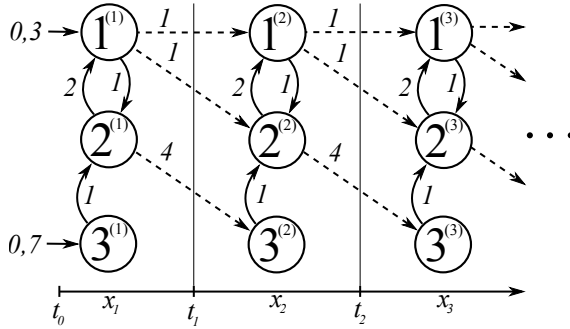
5 pav. Tolydaus laiko Markovo grandinė.

Toliau laisvai pasirenkami perėjimai, kurie generuoja įvykius. Šie perėjimai paveiksle 6 pavaizduoti punktyrinėmis rodyklėmis.



6 pav. Tolydaus laiko Markovo grandinės schema papildyta įvykius generuojančiais perėjimais.

Atkreipiame dėmesį į perėjimą iš pirmos būsenos į pirmą būseną. Toks perėjimas tolydaus laiko Markovo grandinėje laikomas negalimu. Ši nesuderinamumą galima išspręsti nutarus, kad įvykį generuojantis perėjimas įvyksta į padubliuotas būsenas. Padubliuotų būsenų aibė vadinama lygiu (angl. *level*). Taip gaunama begalinė Markovo grandinė, pavaizduota paveiksle 7.



7 pav. Begalinė Markovo grandinė, gauta padubliuojant būsenas įvyki generuojančio perėjimo metu.

Ši begalinė Markovo grandinė nusakoma parametrais

$$\alpha = [0,3 \quad 0 \quad 0,7], \mathbf{D}_0 = \begin{bmatrix} -3 & 1 & 0 \\ 2 & -6 & 0 \\ 0 & 1 & -1 \end{bmatrix}, \mathbf{D}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix}.$$

Nuo to, kaip buvo sugeneruotas tarplaikis, priklauso pradinių tikimybių skirstinys, pagal kurį bus generuojamas kitas tarplaikis. Taip modelis gali įvertinti stebimas priklausomybes. \triangle

10 apibrėžimas. [11] *Markovo atvykimų procesu vadinama įvykių momentų seka, generuojama begalinės Markovo grandinės, kuri yra nurodyta parametrais $(\alpha, \mathbf{D}_0, \mathbf{D}_1)$.*

Tokie procesai anglų kalba parašytoje literatūroje žymimi santrumpa MAP (angl. *Markov arrival process*).

Procesui tęsiantis ilgesnį laiką, pradinių tikimybių vektoriaus α įtaka mažėja, o pasirenkamų būsenų tikimybės nusistovi. Dėl to tikslinga iš karto pradėti procesą su stacionariomis pradinėmis tikimybėmis $\bar{\alpha}$

$$\begin{cases} \bar{\alpha} \mathbf{P} = \bar{\alpha}, \\ \bar{\alpha} \mathbf{1} = 1, \end{cases}$$

kur $\mathbf{P} = (-\mathbf{D}_0)^{-1} \mathbf{D}_1$ yra tikimybių matrica, nurodanti, kaip pasirenkamos pradinės būsenos įvyki generuojančio perėjimo metu.

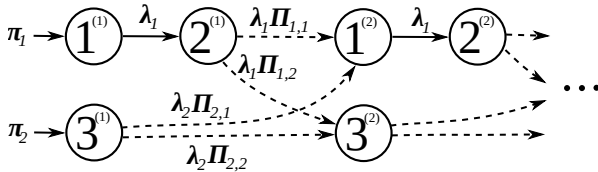
11 apibrėžimas. [11] *Stacionariu Markovo atvykimų procesu vadinama įvykių momentų seka, generuojama begalinės Markovo grandinės, kurios pradinio lygio būsenos pasirenkamos pagal stacionarų tikimybių vektorių. Procesas nurodomas parametrais (D_0, D_1) .*

8 pavyzdys. Pavyzdyje 7 pateiktą Markovo atvykimų procesą galima padaryti stacionarų, suradus stacionarų pradinių tikimybių vektorių, kuris šiuo atveju apytiksliai yra lygus

$$\bar{\alpha} \approx [0,167 \quad 0,167 \quad 0,667] . \quad \triangle$$

Įdomus atvejis – Markovo atvykimo procesai, turintys ER-CHMM struktūrą [12]. Ji yra analogiška HErD (Erlango) hiperskirstiniui, tik papildomai nurodoma Erlango šakų pasirinkimo tikimybių matrica Π . Procesas nurodomas parametrais (π, Π, λ) , o jo struktūra – Erlango šakų eilių vektoriumi r .

9 pavyzdys. ER-CHMM proceso schema su struktūra $r = [2 \quad 1]$ pavaizduota paveiksle 8.



8 pav. Begalinė ER-CHMM proceso Markovo grandinė.

△

Parinkus stacionarų pradinių tikimybių vektorių π , kuris yra lygčių sistemos

$$\begin{cases} \pi \Pi = \pi, \\ \pi \mathbb{1} = 1, \end{cases} \quad (11)$$

sprendinys, gaunamas stacionarus procesas, kuriam nurodyti pakanka parametrų (Π, λ) .

1.5. Markovo atvykimų proceso parametrų paieška

Markovo atvykimų proceso parametrų paieškai galima taikyti maksimalaus tikėtinumo metodą. Algoritmo konstravimo principai yra tokie pat kaip ir fazinių skirstinių parametrų paieškos atveju, jie yra pateikti [6].

Tarplaikis \mathcal{X}_k – tai praleistas laikas k -tajame lygmenyje. Kadangi seka $\mathcal{X} = (\mathcal{X}_1, \dots, \mathcal{X}_T)$ gali būti koreliuota, turime nagrinėti visą seką kartu. Šią atsitiktinių laikų seką generuoja nuosekliai vienas po kito vykstantys Markovo procesai

$$\mathcal{J}_t^{(1)}, \dots, \mathcal{J}_t^{(T)},$$

laiko momentu $t = \mathcal{X}_1 + \dots + \mathcal{X}_k$ pereinama nuo proceso $\mathcal{J}_t^{(k)}$ prie $\mathcal{J}_t^{(k+1)}$ ir t. t. Procesų seka visiškai charakterizuojama diskrečiu lankomų būsenų procesu

$$\mathcal{I}_0^{(1)}, \dots, \mathcal{I}_{\mathcal{M}^{(1)}-1}^{(1)}, \dots, \mathcal{I}_{\mathcal{M}^{(k)}-1}^{(k)}, \dots, \mathcal{I}_{\mathcal{M}^{(T)}-1}^{(T)}$$

ir diskrečiu laiko trukmių procesu

$$\mathcal{S}_0^{(1)}, \dots, \mathcal{S}_{\mathcal{M}^{(1)}-1}^{(1)}, \dots, \mathcal{S}_{\mathcal{M}^{(k)}-1}^{(k)}, \dots, \mathcal{S}_{\mathcal{M}^{(T)}-1}^{(T)},$$

kuris nusako praleistą laiką kiekvieno apsilankymo būsenoje metu, kur $\mathcal{M}^{(k)}$ yra perėjimų skaičius prieš įvykstant įvyki generuojančiam perėjimui pagal \mathbf{D}_1 . Tad išsami informacija \mathcal{Y} apie tarplaikį seką x_1, \dots, x_T yra

$$\mathbf{y} = \left(i_0^{(1)}, \dots, i_{m^{(0)}-1}^{(1)}, \dots, i_{m^{(T)}-1}^{(T)}, s_0^{(1)}, \dots, s_{m^{(0)}-1}^{(1)}, \dots, s_{m^{(T)}-1}^{(T)} \right).$$

Apžvelgiame ER-CHMM Markovo atvykimų procesų klasės parametrų paiešką maksimalaus tikėtinumo metodu.

Šiuo atveju užtektiną visą informaciją \mathcal{Y} charakterizuojanti statistika yra $\mathcal{H} = \{ \mathcal{Q}_i[k], \mathcal{Q}_{i,j}[k] \}$, kur $\mathcal{Q}_i[k]$ – indikatorius, nurodantis ar \mathcal{X}_k buvo sugeneruotas i -tosios Erlango šakos

$$\mathcal{Q}_i[k] = \mathbb{1}_{\{ \mathcal{I}_0^{(k)} = s_i \}},$$

$\mathcal{Q}_{i,j}[k]$ – indikatorius, nurodantis, jog sugeneravus \mathcal{X}_k su i -tąja Erlango šaka, toliau pasirenkama j -toji Erlango šaka

$$\mathcal{Q}_{i,j}[k] = \mathbb{1}_{\{ \mathcal{I}_0^{(k)} = s_i, \mathcal{I}_0^{(k+1)} = s_j \}}.$$

[vertis \mathbf{h} surandamas pagal

$$\begin{aligned}
 q_i[k] &= \mathbb{E}[Q_i[k] \mid \mathcal{X}] = \mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i \mid \mathcal{X}) \\
 &= \frac{\mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i, \mathcal{X})}{\mathbb{P}(\mathcal{X})} = \frac{\mathbf{a}_i[k-1]\mathbf{b}_i[k]}{\boldsymbol{\pi}\mathbf{b}[1]}, \\
 q_{i,j}[k] &= \mathbb{E}[Q_{i,j}[k] \mid \mathcal{X}] = \mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i, \mathcal{I}_0^{(k+1)} = \mathbf{s}_j \mid \mathcal{X}) \\
 &= \frac{\mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i, \mathcal{I}_0^{(k+1)} = \mathbf{s}_j, \mathcal{X})}{\mathbb{P}(\mathcal{X})} \\
 &= \frac{\mathbf{a}_i[k-1]f_i(x_k)\boldsymbol{\Pi}_{i,j}\mathbf{b}_j[k+1]}{\boldsymbol{\pi}\mathbf{b}[1]},
 \end{aligned} \tag{12}$$

čia $\mathbf{a}[k]$, $\mathbf{b}[k]$ yra pasirinkamų Erlango šakų tikėtinumai

$$\begin{aligned}
 \mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i, \mathcal{X}_1, \dots, \mathcal{X}_k) &= \mathbf{a}_i[k], \\
 \mathbb{P}(\mathcal{I}_0^{(k)} = \mathbf{s}_i, \mathcal{X}_k, \dots, \mathcal{X}_T) &= \mathbf{b}_i[k],
 \end{aligned}$$

išreiškiami rekurentinėmis formulėmis

$$\begin{aligned}
 \mathbf{a}_i[k] &= \begin{cases} \boldsymbol{\pi}_i, & k = 0, \\ \sum_{j=1}^R \mathbf{a}_j[k-1]f_j(x_k)\boldsymbol{\Pi}_{j,i}, & 1 \leq k \leq T, \end{cases} \\
 \mathbf{b}_j[k] &= \begin{cases} \sum_{i=1}^R f_j(x_k)\boldsymbol{\Pi}_{j,i}\mathbf{b}_i[k+1], & 1 \leq k \leq T, \\ 1, & k = T+1. \end{cases}
 \end{aligned} \tag{13}$$

10 pavyzdys. Tikimybė, jog generuojant tarplaikį \mathcal{X}_2 , buvo pasirinkta i -toji Erlango šaka, su sąlyga, jog papildomai žinoma tik \mathcal{X}_1 , yra

$$\mathbb{P}(\mathcal{I}_0^{(2)} = \mathbf{s}_i \mid \mathcal{X}_1, \mathcal{X}_2) = \frac{\mathbb{P}(\mathcal{I}_0^{(2)} = \mathbf{s}_i, \mathcal{X}_1, \mathcal{X}_2)}{\mathbb{P}(\mathcal{X}_1, \mathcal{X}_2)} = \frac{\mathbf{a}_i[2]}{\mathbf{a}[2]\mathbf{1}}. \quad \triangle$$

Maksimalaus tikėtinumo parametrai įvertio \mathbf{h} atžvilgiu surandami pagal formules

$$\boldsymbol{\lambda}_i = \frac{\mathbf{r}_i \sum_{k=1}^T q_i[k]}{\sum_{k=1}^T x_k q_i[k]}, \tag{14}$$

$$\boldsymbol{\Pi}_{i,j} = \frac{\sum_{k=1}^T q_{i,j}[k]}{\sum_{k=1}^T q_i[k]}. \tag{15}$$

Tikėtinumo funkcija išreiškiama per tikėtinumo vektorius tokiu būdu

$$\mathcal{L}(\mathbf{\Pi}, \boldsymbol{\lambda} \mid \mathcal{X}) = \mathbf{a}[T] \mathbb{1} = \boldsymbol{\pi} \mathbf{b}[1]. \quad (16)$$

Toliau pateikiamas algoritmas 3, kuriame nurodytas maksimalaus tikėtinumo metodo taikymas parametrų paieškai.

3 algoritmas. Algoritmas ER-CHMM proceso parametrų paieškai maksimalaus tikėtinumo metodu.

procedure *ER-CHMM-EM-fitting*(\mathbf{x}, \mathbf{r})

1. Atsižvelgiant į \mathbf{r} sugeneruojami parametrai $(\mathbf{\Pi}, \boldsymbol{\lambda})$
2. Apskaičiuojamas $(\mathbf{\Pi}, \boldsymbol{\lambda})$ tikėtinumas pagal (16)
3. **repeat**
4. Apskaičiuojamas įvertis h pagal (12)
5. Apskaičiuojami nauji parametrai $(\mathbf{\Pi}, \boldsymbol{\lambda})$ pagal (14), (15)
6. Apskaičiuojamas $(\mathbf{\Pi}, \boldsymbol{\lambda})$ tikėtinumas pagal (16)
7. **until** kol parametrų tikėtinumas didėja
8. **return** $(\mathbf{\Pi}, \boldsymbol{\lambda})$

end procedure

1.6. Nutrūkstantis Markovo atvykimų procesas

Markovo atvykimų procesas generuoja begalinę, galimai koreliuotą, tarplaikių seką. Nutrūkstantis Markovo atvykimų procesas generuoja baigtinę, galimai koreliuotą, tarplaikių seriją

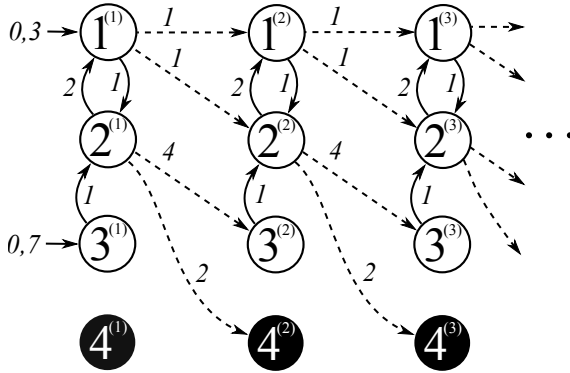
$$\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_K)$$

Nutrūkstantis Markovo atvykimų procesas anglų kalba pateiktoje literatūroje nurodomas santrumpa TMAP (angl. *Transient Markov Arrival Process*) ir yra pateiktas parametrais $(\boldsymbol{\alpha}, \mathbf{D}_0, \mathbf{D}_1)$. Skirtingai nuo Markovo atvykimų proceso, dabar turime $(\mathbf{D}_0 + \mathbf{D}_1) \mathbb{1} \neq \mathbf{0}$. Procesas iš bet kurio lygio i -tosios būsenos gali pereiti į absorbuojančią būseną su intensyvumu d_i , čia vektorius $\mathbf{d} = -(\mathbf{D}_0 + \mathbf{D}_1) \mathbb{1}$.

11 pavyzdys. Panagrinėkime nutrūkstantį Markovo atvykimų procesą, nurodytą parametrais

$$\boldsymbol{\alpha} = [0,3 \quad 0 \quad 0,7], \mathbf{D}_0 = \begin{bmatrix} -3 & 1 & 0 \\ 2 & -8 & 0 \\ 0 & 1 & -1 \end{bmatrix}, \mathbf{D}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix},$$

kurio begalinės Markovo grandinės schema pavaizduota paveiksle 9



9 pav. Begalinė nutrūkstančio Markovo atvykimų proceso Markovo grandinė, pateikta pavyzdyje 11.

Perėjimų į absorbuojančią būseną intensyvumų vektorius

$$\mathbf{d} = -(\mathbf{D}_0 + \mathbf{D}_1) \mathbf{1} = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}. \quad \triangle$$

Turimoms U tarplaikių serijoms

$$\mathbf{x} = \left(x_1^{(1)}, \dots, x_{K_1}^{(1)}, \dots, x_1^{(U)}, \dots, x_{K_U}^{(U)} \right)$$

proceso parametų $(\alpha, \mathbf{D}_0, \mathbf{D}_1)$ tikėtinumo funkcija užrašoma tokiu būdu:

$$\mathcal{L}(\alpha, \mathbf{D}_0, \mathbf{D}_1 \mid \mathcal{X}) = \prod_{u=1}^U \alpha e^{D_0 x_1^{(u)}} \mathbf{D}_1 \dots e^{D_0 x_{K_u}^{(u)}} \mathbf{d}. \quad (17)$$

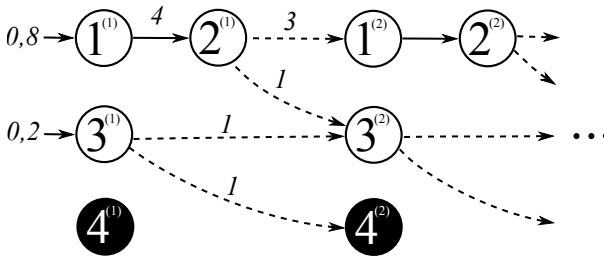
Kaip ir Markovo atvykimų proceso atveju, šiam procesui galima suteikti ER-CHMM struktūrą. Tokiu būdu procesui nusakyti pakanka parametų $(\pi, \mathbf{\Pi}, \lambda)$, kur π_i yra tikimybė, jog bus pasirinkta i -toji Erlango šaka, λ_i perėjimo tarp būsenų i -tojoje Erlango šakoje intensyvumai ir $\mathbf{\Pi}_{i,j}$ nurodo tikimybę, jog paliekant i -tosios Erlango šakos paskutinę būseną (įvyki generuojančio perėjimo metu) bus pasirinkta j -toji Erlango šaka. Taip pat

įvykį generuojančio perėjimo metu galima patekti į absorbuojančią būseną su tikimybe p_i , kur $p = \mathbb{1} - \mathbf{\Pi}\mathbb{1} = (\mathbf{I} - \mathbf{\Pi})\mathbb{1}$.

12 pavyzdys. Nagrinėjame konkretų nutrūkstantį Markovo atvykimų procesą, turintį ER-CHMM struktūrą, nurodytą parametrais

$$\lambda = [4 \quad 2], \mathbf{\Pi} = \begin{bmatrix} 0,75 & 0,25 \\ 0 & 0,5 \end{bmatrix}.$$

Šio proceso begalinė Markovo grandinė pavaizduota paveiksle 10.



10 pav. Pavyzdyje 12 pateikto proceso begalinė Markovo grandinė.

Iš pateiktų parametų galima surasti intensyvumus, su kuriais patenkama į absorbuojančią būseną

$$\tilde{\mathbf{d}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

taip pat ir tikimybes, jog iš i -tosios Erlango šakos bus pasiekta absorbuojanti būseną

$$\tilde{\mathbf{p}} = \begin{bmatrix} 0 \\ 0,5 \end{bmatrix}.$$

Šį procesą galima nurodyti ir bendresniu būdu – parametrais $(\alpha, \mathbf{D}_0, \mathbf{D}_1)$

$$\alpha = [0,8 \quad 0 \quad 0,2], \mathbf{D}_0 = \begin{bmatrix} -4 & 4 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \mathbf{D}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 3 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}. \quad \triangle$$

Mokslinėje literatūroje nėra pateikta maksimalaus tikėtimumo metodo algoritmų šio proceso parametų paieškai atlikti.

2. APROKSIMAVIMAS FAZINIAIS SKIRSTINIAIS

Šiame skyriuje tiriamie fazinių skirstinių struktūras. Tanki struktūra gali būti nusakoma $p = n^2 + n - 1$ parametrais, tačiau įrodyta, jog bet kuriam faziniam skirstiniui nurodyti pakanka $p = 2n - 1$ nepriklausomų parametrų [2]. Kyla klausimas – kokias struktūras naudoti? Pirmiausia aptarsime mūsų siūlomus struktūrų generavimo algoritmus, vėliau pereisime prie struktūrų empirinių tyrimų.

2.1. Fazinių skirstinių struktūrų generavimo algoritmas

Algoritmo idėja – sugeneruojamos visos galimos struktūros ir įvairiais būdais analizuojant šalinamos perteklinės struktūros. Struktūrų generavimo algoritmas turi du parametrus: n – fazinio skirstinio eilė, m – perėjimų skaičius matricinės formos reprezentacijos struktūroje.

4 algoritmas. Algoritmas fazinių skirstinių struktūroms generuoti.

procedure *Structure-generation*(n, m)

(1 žingsnis) Sugeneruojamos visos validžios struktūros su m perėjimais

(2 žingsnis) Pašalinamos perteklinės trivialiai ekvivalenčios struktūros

(3 žingsnis) * Pašalinamos apribotos struktūros arba paliekamos tik apribotos struktūros

(4 žingsnis) * Pašalinamos ciklinės arba neciklinės struktūros

(5 žingsnis) ** Jeigu $m = 2n$, pridedama Kokso skirstinio struktūra

(6 žingsnis) Pašalinamos struktūros, turinčios nekanoninės formos PH(2), PH(3) komponentą (-us)

(7 žingsnis) Pašalinamos struktūros, turinčios nekanoninės formos NMAP(2) komponentą (-us)

return likusių struktūrų aibė

end procedure

*Konfigūruojami algoritmo žingsniai. **Pasirenkamas algoritmo žingsnis. Išsamiau aptariame algoritme 4 pateiktus žingsnius.

Visų galimų struktūrų generavimas (1 žingsnis)

Generuojame visas struktūras, turinčias m perėjimus.

Tam apibrėžiame aibę \tilde{v} , sudarytą iš visų galimų vektorių su dvejetainiais elementais

$$\tilde{v} = \{[\dot{v}_1 \ \dot{v}_2 \ \dots \ \dot{v}_n] \mid \dot{v}_i \in \{0, 1\}, i = \overline{1, n}\},$$

ir dvejetainių matricių aibę $\tilde{\mathbf{A}}$

$$\tilde{\mathbf{A}} = \left\{ \left[\begin{array}{ccc} \dot{\mathbf{A}}_{1,1} & \cdots & \dot{\mathbf{A}}_{1,n} \\ \vdots & \ddots & \vdots \\ \dot{\mathbf{A}}_{n,1} & \cdots & \dot{\mathbf{A}}_{n,n} \end{array} \right] \middle| \dot{\mathbf{A}}_{i,j} \in \{0, 1\}, \dot{\mathbf{A}}_{i,i} = 0, i \neq j, i, j = \overline{1, n} \right\}.$$

Tuomet fazinių skirstinių struktūrų aibę su m perėjimais, kurią sudaro

$$\left\{ \left(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1 \right) \middle| \begin{array}{l} \dot{\boldsymbol{\alpha}} \in \tilde{\mathbf{v}}, \dot{\mathbf{D}}_0 \in \tilde{\mathbf{A}}, \dot{\mathbf{d}}_1 \in \tilde{\mathbf{v}}, \\ \sum_{i=1}^n \dot{\alpha}_i + \sum_{i=1}^n \sum_{j=1}^n \dot{\mathbf{D}}_{i,j} + \sum_{i=1}^n \dot{\mathbf{d}}_{1i} = m \end{array} \right\}$$

Tiesa, ne visos struktūros yra validžios.

12 apibrėžimas. *Struktūra yra validi fazinio skirstinio struktūra, jei: a) įmanoma aplankyti bet kurią iš būsenų, b) iš kiekvienos būsenos, galimai einant per kitas būsenas, įmanoma pasiekti absorbuojančią būseną.*

Iš sugeneruotų visų struktūrų atrenkame fazinių skirstinių struktūras.

Trivialiai ekvivalenčios struktūros (2 žingsnis)

Sugeneruota visų galimų fazinių struktūrų aibė turės ekvivalenčias struktūras.

Pažymime visų fazinių skirstinių su struktūra $(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)$ aibę

$$\text{PH}_{(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)}.$$

13 apibrėžimas. *Fazinių skirstinių struktūros $(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)^{(a)}$ ir $(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)^{(b)}$ yra ekvivalenčios, jeigu*

$$\text{PH}_{(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)^{(a)}} \subseteq \text{PH}_{(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)^{(b)},}$$

$$\text{PH}_{(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)^{(b)}} \subseteq \text{PH}_{(\dot{\boldsymbol{\alpha}}, \dot{\mathbf{D}}_0, \dot{\mathbf{d}}_1)^{(a)}.$$

14 apibrėžimas. [13] *Dvi fazinių skirstinių struktūros yra trivialiai ekvivalenčios, jei viena iš kitos gali būti gauta pernumeruojant būsenas arba (ir) pakeičiant visų perėjimų kryptis.*

Fazinio skirstinio struktūra su pernumeruotomis būsenomis nusako tą pačią fazinių skirstinių aibę [13]. Pernumeravę konkrečios fazinio skirstinio reprezentacijos būsenas gauname kitokią reprezentaciją, ja nusakoma Markovo grandinė. Tačiau šioje Markovo grandinėje laikas, per kurį pasiekiami absorbuojanti būsenai, turi tą patį skirstinį.

Formaliai duotam naujam būsenų numerių sąrašui (l_1, l_2, \dots, l_n) ekvivalenti struktūra $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(b)}$ gaunama iš $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(a)}$ tokiu būdu:

$$\begin{aligned}\dot{\alpha}_i^{(b)} &= \dot{\alpha}_{l_i}^{(a)}, \quad \dot{D}_{0,i,j}^{(b)} = \dot{D}_{0,l_i,l_j}^{(a)}, \\ \dot{d}_{1_i}^{(b)} &= \dot{d}_{1_{l_i}}^{(a)}, \quad i, j = \overline{1, n}.\end{aligned}$$

Jeigu fazinio skirstinio reprezentacijoje pakeičiame visų perėjimų kryptis (menama būsenai, iš kurios pereinama pasirinkus pradinę būseną, tampa absorbuojančia, o absorbuojanti – menamąja). Gauta reprezentacija nusako tą patį fazinį skirstinį, nes laiko, per kurį pasiekiami absorbuojanti būsenai abiejose Markovo grandinėse, skirstiniai yra tie patys [14].

Formaliai struktūrai $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(a)}$ ekvivalenti atgalinio laiko struktūra $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(b)}$ gaunama taip:

$$\begin{aligned}\dot{\alpha}_i^{(b)} &= \dot{d}_{1_i}^{(a)}, \quad \dot{D}_{0,i,j}^{(b)} = \dot{D}_{0,j,i}^{(a)}, \\ \dot{d}_{1_i}^{(b)} &= \dot{\alpha}_i^{(a)}, \quad i, j = \overline{1, n}.\end{aligned}$$

15 apibrėžimas. *Trivialia fazinio skirstinio struktūros transformacija vadiname būsenų pernumeravimą ir laiko atgrėžimą.*

Apribotoji struktūra (3 žingsnis)

16 apibrėžimas. *Apribotą fazinio skirstinio struktūrą vadinama struktūra, kuri tenkina sąlygą $\forall i : \alpha_i d_{1_i} = 0$.*

Apribotąją struktūrą žymime trumpiniu ZID (angl. *Zero Initial Density*). Fazinių skirstinių, charakterizuojamų apribotąja struktūra, tankio funk-

cijos turi savybę $f(0) = 0$.

3 teorema. *Faziniam skirstiniui su apribotąja struktūra nurodyti pakanka $2n-2$ nepriklausomų parametrų.*

Teoriškai visas apribotąsias struktūras galima pašalinti, nes jos nėra tokios lanksčios kaip tos, kurios turi $m \geq 2n$ perėjimus.

17 apibrėžimas. *Fazinio skirstinio arba Markovo atvykimų proceso struktūra vadinama suvaržyta, jeigu ji turi lygiai tiek pat arba mažiau perėjimų nei minimalus nepriklausomų parametrų skaičius, reikalingas bet kuriam procesui nusakyti.*

Ciklinės struktūros (4 ir 5 žingsniai)

18 apibrėžimas. *Struktūra $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)$ nurodo neciklines fazinių skirstinių reprezentacijas, jeigu egzistuoja trivialios transformacijos, kurias pritaikius gaunama trikampė matrica \dot{D}_0 .*

Įrodyta [15], jog bet kuris fazinis skirstinys, turintis neciklinę struktūrą, gali būti transformuotas į surikiuotąjį Kokso (angl. *Coxian*) skirstinį. Tad visas neciklines struktūras galime pakeisti Kokso skirstinio struktūra, kuri yra [16]

$$\dot{\alpha} = [1 \quad 0 \quad \dots \quad 0], \dot{d}_1 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix},$$

$$\dot{D}_0 = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 \end{bmatrix}.$$

Fazinių skirstinių struktūrų komponentai

Fazinio skirstinio struktūrą $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)$ nagrinėjame tirdami jos komponentus.

19 apibrėžimas. *Fazinio skirstinio struktūros komponentu vadinamas pereinamų būsenų poaibis.*

Fazinio skirstinio struktūros $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)$ komponentas nurodomas jam priklausančių pereinamų būsenų sąrašu (l_1, \dots, l_r) . Jei komponentą sudaro r būsenos, likusių būsenų skaičius yra $q = n + 1 - r$. Markovo grandinės būsenas $(n + 1)$ galima pernumeruoti taip, jog intensyvumų matrica įgytų pavidalą

$$Q = \begin{bmatrix} C^* & E \\ X & C \end{bmatrix}, \quad (18)$$

čia C – matrica apibūdina perėjimus tarp komponento būsenų, matrica E nurodo, kokių būdu patenkama į komponento būsenas iš likusių būsenų, X nurodo, kaip išeinama iš komponentui priklausančių būsenų į likusias būsenas. Kadangi mus domina komponento struktūra, įvedame ją nusakančias matricas: \dot{C} – perėjimų tarp komponento būsenų struktūra, \dot{E} / \dot{X} – įėjimo / išėjimo matricų struktūra:

$$\dot{C} = \begin{bmatrix} \dot{C}_{1,1} & \dots & \dot{C}_{1,r} \\ \vdots & \ddots & \vdots \\ \dot{C}_{r,1} & \dots & \dot{C}_{r,r} \end{bmatrix}, \dot{E} = \begin{bmatrix} \dot{E}_{1,1} & \dots & \dot{E}_{1,r} \\ \vdots & \ddots & \vdots \\ \dot{E}_{q,1} & \dots & \dot{E}_{q,r} \end{bmatrix},$$

$$\dot{X} = \begin{bmatrix} \dot{X}_{1,1} & \dots & \dot{X}_{1,q} \\ \vdots & \ddots & \vdots \\ \dot{X}_{q,1} & \dots & \dot{X}_{q,q} \end{bmatrix},$$

čia $\dot{C}_{i,j} \in \{0, 1\}$, $i \neq j$, $i, j = \overline{1, r}$ nurodo perėjimus tarp komponento būsenų; $\dot{E}_{i,j} \in \{0, 1\}$, $i = \overline{1, q}$, $j = \overline{1, r}$ nurodo perėjimus iš likusių būsenų į komponento būsenas; $\dot{X}_{i,j} \in \{0, 1\}$, $i \neq j$, $i, j = \overline{1, q}$ nurodo perėjimus iš komponento būsenų į likusias būsenas.

Toliau apibrėžiame tokias matricų \dot{E} , \dot{X} charakteristikas:

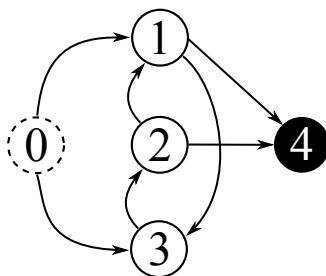
- E_c – būsenų skaičius, iš kurių galima patekti į komponento būsenų aibę;
- E_r – komponento būsenų skaičius, į kurias galima patekti iš likusių būsenų;
- X_c – būsenų skaičius, į kurias patenkama iš komponento būsenų aibės;
- X_r – komponento būsenų skaičius, iš kurių patenkama į likusias būsenas;
- E_{rank} / X_{rank} nurodo didžiausią galimą įėjimo / išėjimo matricų rangą.

$$\begin{aligned} E_{rank} &= \min(E_c, E_r), \\ X_{rank} &= \min(X_c, X_r). \end{aligned} \quad (19)$$

Apibrėžiame komponento perėjimų skaičių $m^{(c)}$

$$m^{(c)} = E_c + \sum_{i=1}^r \sum_{j=1}^r \dot{C}_{i,j} + X_c.$$

13 pavyzdys. Panagrinėkime paveiksle 12 pavaizduotos fazinio skirstinio struktūros komponentą, kurį sudaro (2, 3) būsenos.



11 pav. Pavyzdyje 13 nagrinėjama fazinio skirstinio struktūros schema.

Pernumeravę būsenas taip, kad Markovo grandinės intensyvumų matrica Q atitiktų (18), gauname

$$\dot{Q} = \begin{bmatrix} \dot{C}^* & \dot{E} \\ \dot{X} & \dot{C} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

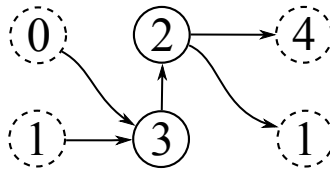
kur atitinkamai

$$\dot{C} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \dot{E} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \dot{X} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Iš matricių \dot{E} , \dot{X} surandamos charakteristikos

$$E_r = 2, E_c = 1, E_{rank} = 1, \\ X_r = 1, X_c = 2, X_{rank} = 1.$$

Paveiksle 12 pavaizduota gauto komponento struktūros schema.



12 pav. Pavyzdyje 13 nagrinėjamo komponento struktūros schema.

△

Nekanoninės formos PH(2), PH(3) komponentų atpažinimas (6 žingsnis)

Dviejų būsenų komponentas, charakterizuojamas $E_r = 1$ ir $X_r = 1$, gali būti traktuojamas kaip antros eilės ($n = 2$) fazinis skirstinys.

Jeigu $m^{(c)} = n + 1 = 3$, tuomet komponentas yra suvaržyta fazinio skirstinio reprezentacija ir nėra pakeičiamas (išlaikant tą patį perėjimų skaičių).

Yra įrodyta, jog bet kuris dviejų būsenų fazinis skirstinys gali būti transformuotas į kanoninę reprezentaciją – surikiuotą Kokso skirstinį [17]. Todėl, jeigu $m^{(c)} = 2n = 4$ ir struktūra nėra trivialiai ekvivalenti Kokso struktūrai, nagrinėjama struktūra yra perteklinė.

Jeigu $m^{(c)} > 2n = 4$, tuomet komponentas yra perteklinis ir tuo pačiu nagrinėjama struktūra yra perteklinė.

Jeigu trijų būsenų komponentas apibūdinamas charakteristikomis $E_r = 1$ ir $X_r = 1$, tuomet jis gali būti nagrinėjamas kaip trečios eilės ($n = 3$) fazinis skirstinys. Jeigu komponentą atitinkanti fazinio skirstinio reprezentacija nėra suvaržyta, reikia patikrinti, ar struktūra atitinka kurią nors iš kanoninių formų [3] struktūrų.

Jeigu $m^{(c)} < 2n = 6$, tuomet struktūra yra suvaržyta ir nėra pakeičiama – negalime daryti išvados apie nagrinėjamos struktūros perteklišumą.

Jeigu $m^{(c)} = 2n = 6$, patikrinama, ar komponentą atitinkančios fazinio skirstinio reprezentacijos struktūra gali būti trivialiai transformuota į kurią nors iš kanoninių formų struktūrų

$$\begin{aligned} (\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(1)} &= \left([1 \ 1 \ 1], \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right), \\ (\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(2)} &= \left([1 \ 0 \ 1], \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right). \end{aligned}$$

Jei ne – nagrinėjama fazinio skirstinio struktūra yra perteklinė.

Jeigu $m^{(c)} = 2n + \Delta M_3 = 7$ ir komponentą atitinkanti fazinio skirstinio reprezentacijos struktūra negali būti trivialiai transformuota į struktūrą

$$(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(3)} = \left([1 \ 1 \ 1], \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right),$$

originali fazinio skirstinio struktūra yra perteklinė.

Galiausiai, jeigu $m^{(c)} > 2n + \Delta M_3 = 7$, originali fazinio skirstinio struktūra yra perteklinė.

Perteklinės struktūros pašalinamos ir toliau nenagrinėjamos.

Nekanoninės formos NMAP(2) komponentų atpažinimas (7 žingsnis)

Dviejų būsenų komponentas, charakterizuojamas $E_r = 1$ ir $X_r = 2$, gali būti nagrinėjamas kaip nestacionarus Markovo atvykimų (NMAP(2)) procesas. Šio proceso kanoninė forma yra pateikta [18].

Tam, kad patikrintume, ar NMAP(2) nėra suvaržytas, įvedame komponento nuliu skaičiaus charakteristiką $z^{(c)}$

$$z^{(c)} = z_E + z_X + z_C,$$

čia z_E / x_X yra nulių skaičius matricose \dot{E} / \dot{X} neįskaitant eilučių / stulpelių, kurie turi tik nulinius elementus, ir z_C yra nulių skaičius matricoje \dot{C} , neįskaitant elementų, esančių pagrindinėje įstrižainėje.

Jeigu $z^{(c)} > 2$, komponentą atitinkantis Markovo atvykimų procesas yra suvaržytas ir nieko konkretaus negalime pasakyti apie nagrinėjamą fazinio skirstinio struktūrą.

Kitu atveju, jeigu $z^{(c)} \leq 2$ ir Markovo atvykimų proceso reprezentacijos struktūra negali būti trivialiai transformuota į vieną iš kanoninę formą [18] atitinkančių struktūrų

$$\begin{aligned} (\dot{\alpha}, \dot{D}_0, \dot{D}_1)^{(1)} &= \left([1 \ 1], \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \right), \\ (\dot{\alpha}, \dot{D}_0, \dot{D}_1)^{(2)} &= \left([1 \ 1], \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \right), \end{aligned}$$

originali fazinio skirstinio struktūra yra perteklinė. Markovo atvykimų procesų reprezentacijų struktūrų trivialios transformacijos apibrėžiamos analogiškai kaip ir fazinių skirstinių atveju.

2.2. Atsitiktinis fazinio skirstinių struktūrų generavimo algoritmas

Anksčiau aptarto fazinių skirstinių reprezentacijų struktūrų generavimo algoritmo trūkumas yra tas, kad generuojamos visos galimos pradinės struktūros, kurių gali būti labai daug. Kaip alternatyvą pateikiame paprastesnį algoritmą, kuris generuoja atsitiktines struktūras su savybėmis Θ . Savybių rinkinys Θ nurodo:

- ar struktūrą yra ciklinė / neciklinė;

- ar struktūra yra apribota (ZID);
- perėjimų skaičių m .

Pirmiausia sugeneruojama struktūra $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)$, turinti maksimalų perėjimų skaičių ir su savybėmis Θ . Tuomet pagal algoritmą 5 iš struktūros atsitiktinai pašaliname po vieną perėjimą, kol pasiekiamas norimas perėjimų skaičius m .

5 algoritmas. Algoritmas atsitiktinai pasirinkto perėjimo pašalinimui.

```

procedure Random-Transition-Removal  $\left( (\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(a)}, \Theta \right)$ 
1.  $\mathcal{T} := \emptyset$ 
2. for perėjimui  $t_i$  struktūroje  $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(a)}$  do
3.   Gaunama  $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(b)}$  pašalinant  $t_i$  iš  $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(a)}$ 
4.   if  $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(b)}$  yra validi ir turi savybes  $\Theta$  then
5.      $t_i$  įterpiamas į  $\mathcal{T}$ 
6.   end if
7. end for
8. Atsitiktinai pasirenkamas perėjimas  $t_k$  iš  $\mathcal{T}$ 
9. Gaunama  $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(b)}$  pašalinant  $t_k$  iš  $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(a)}$ 
10. return  $(\dot{\alpha}, \dot{D}_0, \dot{d}_1)^{(b)}$ 
end procedure

```

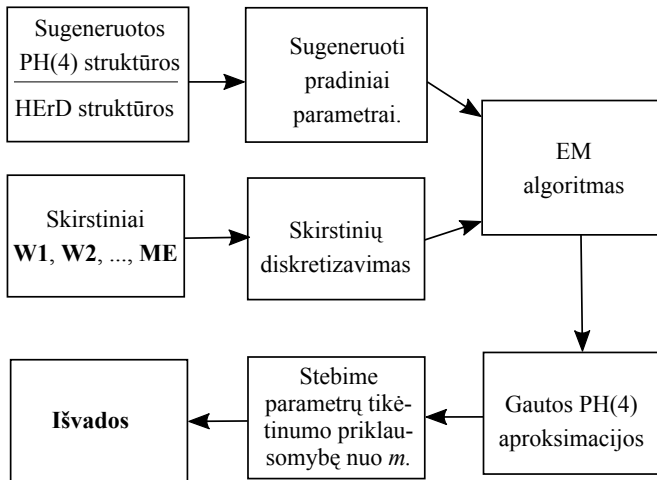
2.3. Empirinis hipotezės apie fazinio skirstinio parametrų kiekį pagrindimas

Žinoma, jog matricinė fazinio skirstinio reprezentacija yra perteklinė. Bet kuriam faziniam skirstiniui nurodyti pakanka $2n - 1$ nepriklausomų parametrų [2].

Remiantis kanonine forma faziniam skirstiniui su trimis fazėmis [3], galima tikėtis, jog ir aukštesnės eilės reprezentacijos turės papildomų perėjimų $\Delta M_n > 0$, kai $n > 3$. Dėl sudėtingo nagrinėjimo analiziniu būdu, dydis ΔM_n , kai $n > 3$, nėra žinomas. Formuluojuame hipotezę, jog kanoninės formos parametrizacijos su papildomais perėjimais ($\Delta m_n > 0$) reprezentuoja „išskirtinius“ fazinius skirstinius ir jų nėra daug. Šie „išskirtiniai“ faziniai skirstiniai turi labai panašius fazinius skirstinius, kuriuos galima reprezentuoti be papildomų perėjimų.

1 hipotezė. *Bet kuriam faziniam skirstiniui, kuris turi matricinės formos reprezentaciją su $m > 2n$ perėjimais, galima sukonstruoti reprezentaciją su $m = 2n$ perėjimais seką, kurias atitinkantys faziniai skirstiniai konverguoja į duotąjį.*

Hipotezę 1 pagrindžiamo empiriškai tam atvejui, kai $n = 4$, tyrimo schema pavaizduota paveiksle 13. Jeigu hipotezė 1 yra teisinga – fazinių skirstinių reprezentacijos (su pilnaja parametrizacija), turinčios $m = 2n$ perėjimus, turėtų duoti ne prastesnius rezultatus nei reprezentacijos su $m > 2n$ perėjimais.



13 pav. Hipotezės 1 empirinio pagrindimo schema.

Aproksimuojame skirstinius, pateiktus [19]. Struktūras generuojame mūsų pateiktu algoritmu ir stebime, kaip surastų parametų tikėtimumas priklauso nuo perėjimų skaičiaus m .

Pateikiame rezultatus, gautus aproksimuojant vieną iš skirstinių, Veibulo (**W1**), kurio tankio funkcija yra

$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}, & x \geq 0, \\ 0, & \text{kitais atvejis,} \end{cases}$$

su parametru reikšmėmis $\lambda = 1$ ir $k = 1,5$. Skirstinį diskretizavome adaptivių algoritmu į 69 stebėjimus intervale nuo 1,00129 iki 4,24. Intervalo galus pasirinkome rankiniu būdu taip, kad diskretizuoto skirstinio charakteristikos (vidurkis, standartinis nuokrypis, asimetriškumo koeficientas) būtų panašios į atitinkamas skirstinio charakteristikas (žr. lentelę 1).

1 lentelė. Skirstinio **W1** ir jo diskretizacijos charakteristikų palyginimas.

	Vidurkis	Stand. nuokr.	Asim. koef.
Skirstinys	0,90275	0,6129	1,072
Diskretizacija	0,90237	0,6115	1,0537
r.a.d.	$4,209 \cdot 10^{-4}$	$2,347 \cdot 10^{-3}$	$1,707 \cdot 10^{-2}$

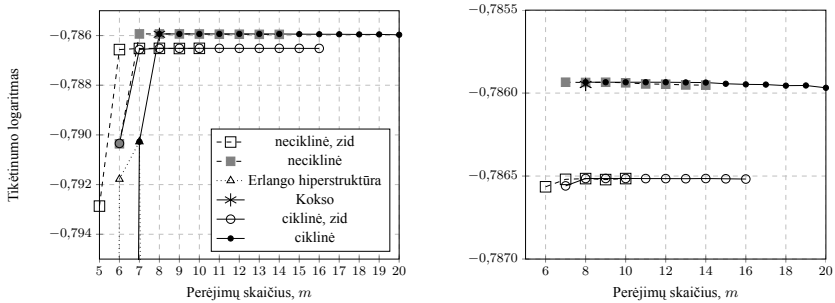
Iš tiesų tiksli skirstinio diskretizacija nėra būtina mūsų empiriniams pagrindimams.

Lentelėje 2 pateikiame sugeneruotų ciklinių struktūrų kiekius kiekvienam perejimų skaičiui m . Taip pat sugeneravome ir ciklines apribotas, neciklines, neciklines apribotas struktūras. Palyginimui pridėjome dar ir Erlango hiperskirstinius, Kokso skirstinius.

2 lentelė. Sugeneruotų PH(4) skirstinių ciklinių struktūrų kiekliai.

m	6	7	8	9	10
Kiekis	1	22	118	526	1241
m	11	12	13	14	15
Kiekis	1778	1874	1429	829	362
m	16	17	18	19	20
Kiekis	132	37	10	2	1

Tyrimo rezultatai skirstiniui **W1** pateikti paveiksle 14.



14 pav. Skirstinio W1 aproksimavimas įvairiomis PH(4) struktūromis.

Iš paveiksle 14 pateiktų grafikų matyti, jog mažiausias perėjimų skaičius, su kuriuo jau pasiekama didžiausia (vizualiai) tikėtimumo reikšmė, yra $m = 8$. Arba kai pilnosios parametrizacijos parametrų skaičius yra $p = 7$, o tai neprieštarauja hipotezės 1 teiginiui. Su nedideliais nuokrypiais ši tendencija kartojasi aproksimuojant ir kitus skirstinius. Disertacijoje pateikti išsamesni rezultatai, kuriais pagrindžiame hipotezę 1.

2.4. Aproksimavimo atsitikimai sugeneruotos struktūros faziniais skirstiniais tyrimas

Empiriškai pagrindus hipotezę apie pakankamą $m = 2p$ perėjimų kiekį (atvejui, kai $n = 4$), aktualu išsamiau patyrinėti aproksimavimą retos (angl. *sparse*) struktūros skirstiniais. Praktikoje, siekiant tiksliau aproksimuoti duotą skirstinį faziniu skirstiniu, dažnai gali prireikti daugiau nei keturių fazių. Kadangi mūsų struktūrų generavimo algoritmas nėra efektyvus dėl galimai nepašalinamų perteklinių struktūrų, kaip paprastą alternatyvą pateikėme atsitiktinių struktūrų generavimo algoritmą. Tačiau prieš mėginant atsitiktinai generuoti struktūras, tikslinga palyginti jų taikymą su sugeneruotomis struktūromis, kurios tikrai apima visą galimų fazinių skirstinių aibę.

Ankstesniame skyriuje skirstinius aproksimuojame ketvirtos eilės faziniais skirstiniais. Registruojame gautų parametrų įverčių tikėtimumą ir iteracijų skaičių, kurio prireikė konvergavimui pasiekti (iki pasiekiamas mažesnis tikėtimumo santykinis padidėjimas daugiau nei 10^{-8}). Tyrėme tokius fazinių skirstinių klasių atvejus: tankios struktūros (FULL), pagal

mūsų algoritmą sugeneruotos ciklinės ir neciklinės struktūros (GEN:cyc-2n, GEN:acyc-2n) bei atsitiktinai sugeneruotos ciklinės ir neciklinės struktūros (RND:cyc-2n, RND:acyc-2n). Visos struktūros, neįskaičiuojant tankiosios, turi po $m = 2n = 8$ perėjimus. Kiekvienai struktūrų klasei generavome po 1 180 pradinių parametrų rinkinių.

Gautus rezultatus aproksimuojant **W1** skirstinį pateikiame lentelėje 3.

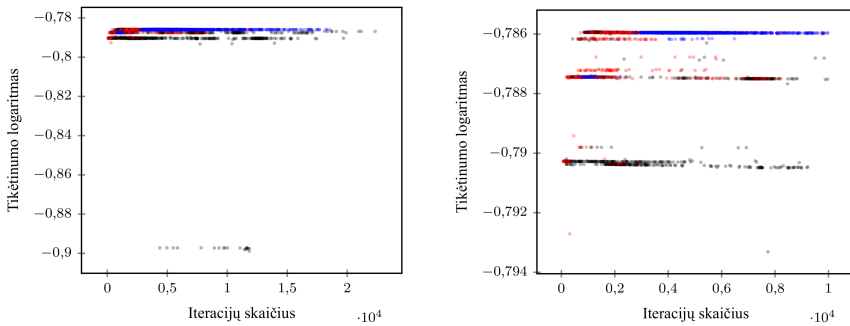
3 lentelė. Gautų parametrų įverčių tikėtinumo logaritmo ir iteracijų, aproksimuojant **W1** skirstinį ketvirtos eilės faziniu skirstiniu, skaitinės charakteristikos.

	Min.	Maks.	Vidurkis	Stand. nuokr.
GEN:cyc-2n iteracijų sk.	-1,4327 127	-0,7859 22331	-0,7911 4574,24	0,0234 4527,32
GEN:acyc-2n iteracijų sk.	-0,7927 76	-0,7859 12714	-0,7876 1889,29	0,0015 2151,44
RND:cyc-2n iteracijų sk.	-0,9031 121	-0,7859 24470	-0,7996 4773,32	0,0328 4573,64
RND:acyc-2n iteracijų sk.	-0,8972 10	-0,7859 9780	-0,7885 1132,02	0,0065 1476,19
FULL iteracijų sk.	-0,7874 727	-0,7859 18536	-0,7860 5827,76	0,0002 3051,21

Pastebėta tendencija, kad aproksimuojant tankios struktūros faziniu skirstiniu mažiausias iteracijų kiekis yra didesnis lyginant su kitomis struktūrų klasėmis. Tačiau bendrai aproksimuojant retomis struktūromis stebima tendencija (nors tai vienareikšmiškai ir neatsispindi šiame pavyzdyje), kad aproksimuojant retomis struktūromis gali prireikti ir kur kas daugiau iteracijų konvergavimui pasiekti. To galima priežastis – parinkus netinkamą struktūrą, globalaus ekstremumo pasiekimas gali būti negalimas. Kita vertus, parinkus tinkamą struktūrą, prie ekstremumo artėjama kur kas sparčiau nei naudojant tankią struktūrą. Klausimas – kaip nustatyti, kokia struktūra (ar jų aibė) geriausiai tinka konkrečiu atveju?

Jeigu darome prielaidą, kad netinkamai parinktos struktūros lėčiau konverguoja, tikslinga patyrinti, kaip siejasi gautų parametrų įverčių tikėtinumai su iteracijų skaičiumi. Šis sąryšis skirstiniui **W1** pavaizduotas

paveiksle 15.



15 pav. Ryšys tarp gautų parametų įverčių tikėtinumo ir iteracijų skaičiaus ketvirtos eilės faziniu skirstiniu aproksimuojant **W1** skirstinį.

Paveiksle 15 juodais taškais pažymėti rezultatai gauti aproksimuojant tankia struktūra, mėlyna – sugeneruotomis ciklinėmis struktūromis (GEN:cyc-2n) ir neciklinėmis struktūromis (GEN:acyc-2n). Rezultatų, gautų su atsitiktinai sugeneruotomis struktūromis, nepavaizdavome – jie labai panašūs į rezultatus, gautus naudojant atsitiktinai sugeneruotas struktūras.

Aiškliai matome kelias tikėtinumo reikšmes, prie kurių priartėjama. Tiesa, skiriasi iteracijų skaičius. Apibendrinant rezultatus galima teigti, kad tanki struktūra labai dažnai sugeba surasti labiausiai tikėtinus parametų įverčius. Tuo tarpu retos struktūros ne visada suranda labiausiai tikėtinus parametrus, tačiau juos gali surasti su mažesniu iteracijų skaičiumi. Tad kyla natūrali išvada, kad tinkamas retas struktūras galima mėginti atrinkti atsirenkant tas, kurių tikėtinumo reikšmė didžiausia atlikus nedidelį skaičių iteracijų. Šią idėją plačiau panagrinėjome paskutiniame skyriuje.

2.5. Išvados

Sukurtas fazinių skirstinių matricinės formos reprezentacijos struktūrų generavimo algoritmas. Tačiau negalima garantuoti, kad sugeneruota struktūrų aibė (turinti m perėjimus) yra minimali. Algoritmu iš pradžių sugeneruojamos visos galimos struktūros ir tuomet šalinamos perteklinės. Dėl eksponentiškai didėjančio pradinių struktūrų skaičiaus aukštesnių eilių ($n > 4$) struktūrų aibių generavimas yra sunkiai įgyvendinamas. Nepaisant

to, sugeneruotos ketvirtos eilės struktūrų aibės buvo sėkmingai panaudotos empiriškai pagrindžiant iškeltą hipotezę.

Hipotezė apie pakankamą perėjimų skaičių ($m = 2n$) fazinio skirstinio matricinės formos reprezentacijoje pagrįsta empiriškai tokiam atvejui, kai $n = 4$. Hipotezė pagrįsta empiriškai, maksimalaus tikėtinumo metodu faziniais skirstiniais aproksimuojant devynis skirstinius, mūsų algoritmu sugeneruotomis struktūrų aibėmis. Šie skirstiniai paskelbti mokslinėje literatūroje ir naudojami tiriant aproksimavimą faziniais skirstiniais. Hipotezė pagrįsta remiantis rezultatais, gautais analizuojant parametrų tikėtinumo priklausomybę nuo perėjimų skaičiaus m . Vienintelė priešara hipotezei gauta skirstinio **L1** atveju. Labiau tikėtini parametrai surasti naudojant struktūrą su vienu papildomu perėjimu (t. y. $m = 2n + 1 = 9$). Vis dėlto surastų parametrų įverčių tikėtinumo logaritmas yra tik 0,0000002 didesnis lyginant su tikėtinumo logaritmu gautu naudojant struktūrą be papildomų perėjimų (t. y. $m = 2n = 4$). Toks nuokrypis gali būti laikomas nereikšmingu, ypač galvojant apie skaičiavimo kompiuteriu specifika.

Tyrimo rezultatai (hipotezei pagrįsti) atskleidė dar keletą tendencijų. Mokslinėje literatūroje pastebėta, kad aproksimuojant neciklinės struktūros faziniais skirstiniais EM metodu gali būti gauti labiau tikėtini parametrai nei naudojant Kokso (angl. *Coxian*) skirstinį. Nepaisant fakto, jog pastarasis skirstinys padengia visą necikline struktūra reprezentuojamų fazinių skirstinių erdvę. Tai mėginama aiškinti tuo, kad Kokso struktūra nėra optimali EM metodo konvergavimo prasme. Mūsų eksperimentinio tyrimo rezultatuose taip pat stebima ši tendencija. Nagrinėjant tyrimo rezultatus pastebėta, jog daugeliu atvejų apribota struktūra (t. y. struktūra, kuri nurodo fazinius skirstinius, kurie gali turėti daugiausiai $p = 2n - 2$ nepriklausomus parametrus) suranda mažiau tikėtinus parametrų įverčius.

Hipotezė apie pakankamą perėjimų skaičių buvo pritaikyta tiriant aproksimavimą faziniais skirstiniais. Palyginta, kuo skiriasi aproksimavimas reta (su $m = 2n$ perėjimais) ir tankia struktūromis tam atvejui, kai $n = 4$. Analizuojant gautus rezultatus nustatyti tokie dėsningumai:

1. Dėl tankios struktūros lankstumo gautų parametrų įverčių tikėtinumas turi mažiausią dispersiją. Ieškant parametrų įverčių retomis struktūromis stebima atvirkštinė tendencija.
2. Aproksimuojant tankia struktūra dėl perteklinių parametrų konvergav-

imas yra lėtesnis ir priartėjus prie lėto konvergavimo zonos paieškos procedūra gali būti per anksti nutraukta.

3. Aproximuojant retomis struktūromis surastų parametrų tikėtinas priklausos nuo konkrečios struktūros ir aproksimuojamų duomenų. Jeigu struktūra yra tinkama, galima tikėtis greito konvergavimo (kur kas greitesnio nei su tankia struktūra). Jeigu parinkta mažiau tinkama struktūra, konvergavimas gali būti lėtas ir pasiekiami prastesni parametrų įverčiai.

Potencialiai retos struktūros gali būti naudojamos tuomet, kai reikia surasti labiau tikėtinus parametrų įverčius su mažesniu iteracijų skaičiumi. Deja, rezultatų patikimumas priklauso nuo patikrintų retų struktūrų skaičiaus. Todėl, siekiant praktiškai pasinaudoti sparčiu retų struktūrų konvergavimu, reikalinga metodika tinkamų struktūrų identifikavimui.

3. MARKOVO ATVYKIMŲ PROCESŲ PARAMETRŲ PAIEŠKOS IŠLYGIAGRETINIMAS

Užduotis – išlygiagretinti stacionaraus Markovo atvykimų proceso parametrų paiešką, vykdymui naudojant L skaičiavimo gijas (angl. *threads*). Stebimus tarplaikius x_1, \dots, x_T interpretuojame kaip laiko tarpus tarp ER-CHMM Markovo atvykimų procesų klasės generuojamų įvykių. Pasirenkame ER-CHMM procesų klasę dėl dviejų priežasčių: a) mažesnių skaičiavimo apimčių lyginant su bendruoju procesu, b) uždaviniui galima taikyti paslėptųjų Markovo modelių parametrų paieškos Baum-Welch [20] algoritmą. Mūsų darbo tikslas – formalizuoti ir palyginti galimus išlygiagretinimo algoritmų variantus.

3.1. Algoritmas P-1

Vykdam parametru (Π, λ) paiešką šiuo algoritmu, iteracijos metu tarplankiai x_1, \dots, x_T nuskaitomi vieną kartą.

Apibrėžiame stebėjimų sekos x_u, \dots, x_v tikėtinumų matricą

$$P[u^v] = P[u]P[u+1] \dots P[v] = \prod_{z=u}^v \begin{bmatrix} f_1(x_z)\Pi_{1,1} & \dots & f_1(x_z)\Pi_{1,R} \\ \vdots & \ddots & \vdots \\ f_R(x_z)\Pi_{R,1} & \dots & f_R(x_z)\Pi_{R,R} \end{bmatrix}$$

atvejams, kai $u > v$, apibrėžiame $P[u^v] = I$. Tikėtinumo vektoriai $\mathbf{a}[u]$, $\mathbf{b}[u]$ išreiškiami per $P[u^v]$ tokiu būdu:

$$\mathbf{a}[u] = \boldsymbol{\pi} P[u^u] \text{ ir } \mathbf{b}[u] = P[u^T] \mathbf{1}. \quad (20)$$

Tarplaikių seką x_1, \dots, x_T padalijame į L serijas po K tarplaikių. Pastutinėje serijoje gali būti ir mažiau tarplaikių, $T - (L - 1)K$.

Pažymime visos ℓ -tosios serijos tikėtinumų matricą $U[\ell] = P[(\ell-1)K+1]^\ell$, tuomet ℓ -tosios serijos tikėtinumo vektorius galime išreikšti taip:

$$\mathbf{a}[u] = \boldsymbol{\pi} \left(\prod_{z=1}^{\ell-1} \mathbf{U}[z] \right) \mathbf{P}[(\ell-1)K+1], \quad (21)$$

$$\mathbf{b}[u] = \mathbf{P}[\ell K] \left(\prod_{z=\ell+1}^L \mathbf{U}[z] \right) \mathbf{1}. \quad (22)$$

Pažymime ℓ -tosios serijos atidarantį $\boldsymbol{\pi}^\ell$ ir uždarantį $\mathbf{1}^\ell$ vektorius kaip

$$\boldsymbol{\pi}^\ell = \mathbf{a}[\ell K] = \boldsymbol{\pi} \mathbf{P}[\ell K] = \boldsymbol{\pi} \prod_{z=1}^{\ell} \mathbf{U}[z],$$

$$\mathbf{1}^\ell = \mathbf{b}[\ell K + 1] = \boldsymbol{\pi} \mathbf{P}[\ell K + 1] = \prod_{z=\ell+1}^L \mathbf{U}[z] \mathbf{1}.$$

Dabar tikėtinumo vektorius galime apskaičiuoti kiekvienoje iš serijų atskirai, pagal

$$\mathbf{a}[u] = \boldsymbol{\pi}^\ell \mathbf{P}[(\ell-1)K+1] \quad \text{ir} \quad \mathbf{b}[u] = \mathbf{P}[\ell K] \mathbf{1}^\ell,$$

kai $(\ell - 1)K + 1 \leq u \leq \ell K$.

Perrašome parametrų $(\boldsymbol{\Pi}, \boldsymbol{\lambda})$ skaičiavimo formules (14), (15) atskirai pažymėdami sumas

$$\boldsymbol{\lambda}_i = \frac{\sum_{u=1}^T \mathbf{r}_i \mathbf{a}_i[u-1] \mathbf{b}_i[u]}{\sum_{u=1}^T x_u \mathbf{a}_i[u-1] \mathbf{b}_i[u]} = \frac{\mathbf{r}_i S_i^{(1)}}{S_i^{(2)}}, \quad (23)$$

$$\boldsymbol{\Pi}_{i,j} = \frac{\sum_{u=1}^{T-1} \mathbf{a}_i[u-1] f_i(x_u) \boldsymbol{\Pi}_{i,j} \mathbf{b}_j[u+1]}{\sum_{u=1}^{T-1} \mathbf{a}_i[u-1] \mathbf{b}_j[u]} = \frac{S_{i,j}^{(3)}}{S_i^{(3)}}, \quad (24)$$

iš čia turime, jog

$$S_i^{(1)} = S_i^{(3)} \mathbf{a}_i[T-1] \mathbf{b}_i[T] \quad \text{bei} \quad S_i^{(3)} = \sum_{j=1}^R S_{i,j}^{(3)}.$$

Todėl pakanka išlygiagretinti tik sumų

$$S_i^{(2)} \quad \text{ir} \quad S_{i,j}^{(3)}$$

skaičiavimą. Tarplaikių skaičių, esančių priešais ℓ -tąją seriją, pažymėję kaip $\iota_\ell = (\ell - 1)K$, pertvarkome sumą $S_i^{(2)}$

$$\begin{aligned}
S_i^{(2)} &= \sum_{\ell=1}^{L-1} \sum_{u=1}^K \mathbf{a}_i[\iota_\ell + u - 1] x_{\iota_\ell + u} \mathbf{b}_i[\iota_\ell + u] \\
&+ \sum_{u=1}^{T-(L-1)K} \mathbf{a}_i[\iota_L + u - 1] x_{\iota_L + u} \mathbf{b}_i[\iota_L + u] \\
&= \sum_{\ell=1}^{L-1} \pi^{\ell-1} \underbrace{\sum_{u=1}^K \mathbf{P}[\begin{smallmatrix} \iota_\ell + u - 1 \\ \iota_\ell + 1 \end{smallmatrix}] \mathbf{e}_i^\top x_{\iota_\ell + u} \mathbf{e}_i \mathbf{P}[\begin{smallmatrix} \ell K \\ \iota_\ell + u \end{smallmatrix}]}_{\bar{\Omega}_i^{(2)}[\ell, K] = \Omega_i^{(2)}[\ell]} \mathbb{1}^{\ell+1} \\
&+ \pi^{L-1} \underbrace{\sum_{u=1}^{T-(L-1)K} \mathbf{P}[\begin{smallmatrix} \iota_L + u - 1 \\ \iota_L + 1 \end{smallmatrix}] \mathbf{e}_i^\top x_{\iota_L + u} \mathbf{e}_i \mathbf{P}[\begin{smallmatrix} T \\ \iota_L + u \end{smallmatrix}]}_{\bar{\Omega}_i^{(2)}[L, T-(L-1)K] = \Omega_i^{(2)}[L]} \mathbb{1},
\end{aligned} \tag{25}$$

čia

$$\bar{\Omega}_i^{(2)}[\ell, z] = \sum_{u=1}^z \mathbf{P}[\begin{smallmatrix} \iota_\ell + u - 1 \\ \iota_\ell + 1 \end{smallmatrix}] \mathbf{e}_i^\top x_{\iota_\ell + u} \mathbf{e}_i \mathbf{P}[\begin{smallmatrix} \iota_\ell + z \\ \iota_\ell + u \end{smallmatrix}].$$

Užrašome sumą $S_{i,j}^{(3)}$

$$\begin{aligned}
S_{i,j}^{(3)} &= \sum_{\ell=1}^L \sum_{u=1}^K \mathbf{a}_i[\ell_\ell + u - 1] \mathbf{P}_{i,j}[\ell_\ell + u] \mathbf{b}_j[\ell_\ell + u + 1] \\
&+ \sum_{u=1}^{T-(L-1)K-1} \mathbf{a}_i[\ell_L + u - 1] \mathbf{P}_{i,j}[\ell_L + u] \mathbf{b}_j[\ell_L + u + 1] \\
&= \sum_{\ell=1}^{L-1} \pi^{\ell-1} \underbrace{\sum_{u=1}^K \mathbf{P}[\begin{smallmatrix} \ell_\ell+u-1 \\ \ell_\ell+1 \end{smallmatrix}] \mathbf{e}_i^\top \mathbf{e}_i \mathbf{P}[\ell_\ell + u] \mathbf{e}_j^\top \mathbf{e}_j \mathbf{P}[\begin{smallmatrix} \ell_\ell+K \\ \ell_\ell+u+1 \end{smallmatrix}]}_{\bar{\Omega}_{i,j}^{(3)}[\ell,K]=\Omega_{i,j}^{(3)}[\ell]} \mathbb{1}^{\ell+1} \\
&+ \pi^{L-1} \underbrace{\sum_{u=1}^{T-(L-1)K-1} \mathbf{P}[\begin{smallmatrix} \ell_L+u-1 \\ \ell_L+1 \end{smallmatrix}] \mathbf{e}_i^\top \mathbf{e}_i \mathbf{P}[\ell_L + u] \mathbf{e}_j^\top \mathbf{e}_j \mathbf{P}[\begin{smallmatrix} T \\ \ell_L+u+1 \end{smallmatrix}]}_{\bar{\Omega}_{i,j}^{(3)}[L,T-(L-1)K-1]=\Omega_{i,j}^{(3)}[L]} \mathbb{1},
\end{aligned} \tag{26}$$

čia

$$\bar{\Omega}_{i,j}^{(3)}[\ell, z] = \sum_{u=1}^z \mathbf{P}[\begin{smallmatrix} \ell_\ell+u-1 \\ \ell_\ell+1 \end{smallmatrix}] \mathbf{e}_i^\top \mathbf{e}_i \mathbf{P}[\ell_\ell + u] \mathbf{e}_j^\top \mathbf{e}_j \mathbf{P}[\begin{smallmatrix} \ell_\ell+z \\ \ell_\ell+u+1 \end{smallmatrix}].$$

Remdamesi paslėptųjų Markovo modelių parametų paieškos algoritmu [20], įrodome tokią teoremą.

4 teorema. *Kiekvienam z , $0 < z \leq K$, kai $\ell < L$, bei $0 < z \leq T - (L - 1)K$, kai $\ell = L$, matricos $\bar{\Omega}_i^{(2)}[\ell, z]$ ir $\bar{\Omega}_{i,j}^{(3)}[\ell, z]$ tenkina tokius rekursyvius sąryšius*

$$\bar{\Omega}_i^{(2)}[\ell, z] = \bar{\Omega}_i^{(2)}[\ell, z - 1] \mathbf{P}[\ell_\ell + z] + \mathbf{P}[\begin{smallmatrix} \ell_\ell+z-1 \\ \ell_\ell+1 \end{smallmatrix}] \mathbf{e}_i^\top x_z \mathbf{e}_i \mathbf{P}[\ell_\ell + z]$$

ir

$$\bar{\Omega}_{i,j}^{(3)}[\ell, z] = \bar{\Omega}_{i,j}^{(3)}[\ell, z - 1] \mathbf{P}[\ell_\ell + z] + \mathbf{P}[\begin{smallmatrix} \ell_\ell+z-1 \\ \ell_\ell+1 \end{smallmatrix}] \mathbf{e}_i^\top \mathbf{e}_i \mathbf{P}[\ell_\ell + z] \mathbf{e}_j^\top \mathbf{e}_j$$

su pradinėmis reikšmėmis $\bar{\Omega}_i^{(2)}[\ell, 0] = \mathbf{0}$ ir $\bar{\Omega}_{i,j}^{(3)}[\ell, 0] = \mathbf{0}$.

Toliau pateikiame algoritmo vykdymo etapus.

1. Matricų $\mathbf{U}[\ell]$, $\mathbf{\Omega}_i^{(2)}[\ell]$ ir $\mathbf{\Omega}_{i,j}^{(3)}[\ell]$ lygiagretus apskaičiavimas kiekvienai serijai $\ell = 1, \dots, L$.
2. Atidarančių / uždarančių tikėtinumo vektorių $\boldsymbol{\pi}^\ell$ ir $\mathbb{1}^\ell$ skaičiavimas pagal formules

$$\boldsymbol{\pi}^\ell = \begin{cases} \boldsymbol{\pi}, & \ell = 0, \\ \boldsymbol{\pi}^{\ell-1} \mathbf{U}[\ell], & \ell \geq 1, \end{cases} \quad \text{ir} \quad \mathbb{1}^\ell = \begin{cases} \mathbb{1}, & \ell = L + 1, \\ \mathbf{U}[\ell] \mathbb{1}^{\ell+1}, & \ell \leq L, \end{cases}$$

ir tuomet sumų $S_i^{(2)}$, $S_{i,j}^{(3)}$, $i, j = 1, \dots, R$ skaičiavimas pagal

$$S_i^{(2)} = \sum_{\ell=1}^L \boldsymbol{\pi}^{\ell-1} \mathbf{\Omega}_i^{(2)}[\ell] \mathbb{1}^{\ell+1}, \quad S_{i,j}^{(3)} = \sum_{\ell=1}^L \boldsymbol{\pi}^{\ell-1} \mathbf{\Omega}_{i,j}^{(3)}[\ell] \mathbb{1}^{\ell+1},$$

kurios panaudojamos apskaičiuojant naujus parametrų įverčius $\boldsymbol{\lambda}_i$, $\mathbf{\Pi}_{i,j}$.

3.2. Algoritmas P-2

Kita algoritmo versija – P-2 gaunama iš P-1 algoritmo, pastebėjus tai kad, vietoj veiksmų su matricomis $\mathbf{\Omega}_i^{(2)}$ ir $\mathbf{\Omega}_{i,j}^{(3)}$ galime dirbti su vektoriais. Kita algoritmo savybė – per duotų tarplaikių seką pereinama du kartus.

1. Matricų $\mathbf{U}[\ell]$ lygiagretus apskaičiavimas kiekvienai serijai $\ell = 1, \dots, L$.
2. Atidarančių / uždarančių tikėtinumo vektorių $\boldsymbol{\pi}^\ell$ ir $\mathbb{1}^\ell$ skaičiavimas pagal formules

$$\boldsymbol{\pi}^\ell = \begin{cases} \boldsymbol{\pi}, & \ell = 0, \\ \boldsymbol{\pi}^{\ell-1} \mathbf{U}[\ell], & \ell \geq 1, \end{cases} \quad \text{ir} \quad \mathbb{1}^\ell = \begin{cases} \mathbb{1}, & \ell = L + 1, \\ \mathbf{U}[\ell] \mathbb{1}^{\ell+1}, & \ell \leq L, \end{cases}$$

3. Vektorių $\boldsymbol{\omega}_i^{(2)}[\ell] = \boldsymbol{\pi}^\ell \mathbf{\Omega}_i^{(2)}[\ell]$ ir $\boldsymbol{\omega}_{i,j}^{(3)}[\ell] = \boldsymbol{\pi}^\ell \mathbf{\Omega}_{i,j}^{(3)}[\ell]$ lygiagretus skaičiavimas kiekvienai serijai $\ell = 1, \dots, L$.
4. Sumų $S_i^{(2)}$ ir $S_{i,j}^{(3)}$ skaičiavimas pagal

$$S_i^{(2)} = \sum_{\ell=1}^L \boldsymbol{\omega}_i^{(2)}[\ell] \mathbb{1}^{\ell+1}, \quad S_{i,j}^{(3)} = \sum_{\ell=1}^L \boldsymbol{\omega}_{i,j}^{(3)}[\ell] \mathbb{1}^{\ell+1},$$

ir naujų parametrų įverčių $\boldsymbol{\lambda}_i$, $\mathbf{\Pi}_{i,j}$ skaičiavimas.

Kuomet Erlango šakų tankiai $f_i(x)$ yra išsaugomi atmintyje, turime algoritmo modifikaciją P-2-D.

3.3. Algoritmas P-3

Trečiasis algoritmas skiriasi nuo ankstesnių tuo, jog sumos $S_i^{(2)}$ ir $S_{i,j}^{(3)}$ skaičiuojamos tiesiogiai iš tikėtinumo vektorių $\mathbf{a}[u]$, $\mathbf{b}[u]$.

1. Kaip ir P-2 algoritmo atveju, kartu skaičiuojame matricas $\mathbf{U}[\ell]$ kiekvienai serijai $\ell = 1, \dots, L$.
2. Atidarantių / uždarančių tikėtinumo vektorių $\boldsymbol{\pi}^\ell$ ir $\mathbf{1}^\ell$ skaičiavimas pagal formules

$$\boldsymbol{\pi}^\ell = \begin{cases} \boldsymbol{\pi}, & \ell = 0, \\ \boldsymbol{\pi}^{\ell-1} \mathbf{U}[\ell], & \ell \geq 1, \end{cases} \quad \text{ir} \quad \mathbf{1}^\ell = \begin{cases} \mathbf{1}, & \ell = L + 1, \\ \mathbf{U}[\ell] \mathbf{1}^{\ell+1}, & \ell \leq L. \end{cases}$$

3. Lygiagretus tikėtinumo vektorių $\mathbf{a}[u]$ ir $\mathbf{b}[u]$ skaičiavimas kiekvienai serijai $\ell = 1, \dots, L$ pagal

$$\mathbf{a}[u] = \boldsymbol{\pi}^\ell \mathbf{P}[\iota_{\ell+1}^u], \text{ ir } \mathbf{b}[u] = \mathbf{P}[\iota_u^{\ell+K}] \mathbf{1}^{\ell+1}. \quad (27)$$

4. Lygiagretus tarpinių sumų $S_i^{(2)}[\ell]$ ir $S_{i,j}^{(3)}[\ell]$ skaičiavimas pagal

$$S_i^{(2)}[\ell] = \sum_{u=1}^K \mathbf{a}_i[\iota_\ell + u - 1] x_{\iota_\ell + u} \mathbf{b}_i[\iota_\ell + u] \quad (28)$$

$$S_{i,j}^{(3)}[\ell] = \sum_{u=1}^K \mathbf{a}_i[\iota_\ell + u - 1] \{ \mathbf{P}[\iota_\ell + u] \}_{i,j} \mathbf{b}_j[\iota_\ell + u + 1],$$

pasinaudojant anksčiau apskaičiuotais tikėtinumo vektoriais $\mathbf{a}[u]$, $\mathbf{b}[u]$.

5. Nuoseklus sumų $S_i^{(2)}$ ir $S_{i,j}^{(3)}$ skaičiavimas pagal

$$S_i^{(2)} = \sum_{\ell=1}^L S_i^{(2)}[\ell], \text{ ir } S_{i,j}^{(3)} = \sum_{\ell=1}^L S_{i,j}^{(3)}[\ell], \quad (29)$$

kurios naudojamos naujų parametru $\boldsymbol{\lambda}_i$, $\boldsymbol{\Pi}_{i,j}$ įterčių apskaičiavimui.

Kuomet Erlango šakų tankiai $f_i(x)$ yra išsaugomi atmintyje, turime algoritmo modifikaciją P-3-D.

3.4. Analizinis algoritmų palyginimas

Pateikti algoritmai skiriasi naudojamos atminties kiekiu ir skaičiavimo sudėtingumu.

Algoritmų naudojamos atminties analizė pateikta lentelėje 4.

4 lentelė. Algoritmų vykdymui reikalingas atminties kiekis, matuojamas kintamųjų skaičiumi.

Duomenys	Nuoseklus alg.	P-1 alg.	P-2 alg.	P-3 alg.
Parametrai r, Π, λ, π	$R^2 + 3R$	$R^2 + 3R$	$R^2 + 3R$	$R^2 + 3R$
Tarplankiai x_u	T	T	T	T
Sumos $S_{i,j}^{(3)}, S_i^{(n)}, n=1, 2, 3$	$R^2 + 3R$	$R^2 + 3R$	$R^2 + 3R$	$R^2 + 3R$
Vektoriai $\pi^\ell, \mathbb{1}^\ell$		$2LR(+2L)$	$2LR(+2L)$	$2LR(+2L)$
Matricos $U[\ell]$		$LR^2(+L)$	$LR^2(+L)$	$LR^2(+L)$
Dalinės sumos		$\Omega_i^{(2)}[\ell]: LR^3(+L)$	$\omega_i^{(2)}[\ell]: LR^2(+L)$	$S_i^{(2)}[\ell]: LR(+L)$
		$\Omega_{i,j}^{(3)}[\ell]: LR^4(+L)$	$\omega_{i,j}^{(3)}[\ell]: LR^3(+L)$	$S_{i,j}^{(3)}[\ell]: LR^2(+L)$
Vektoriai $a[u], b[u]$	$2TR(+2T)$			$2TR(+2T)$
Šakų tankiai $f_i(x)$	TR		P-2-D tik: TR	P-3-D tik: TR
Bendras sudėtingumas	$\mathcal{O}(TR)$	$\mathcal{O}(T + LR^4)$	$\mathcal{O}(T + LR^3)$	$\mathcal{O}(TR + LR^2)$

Lentelėje 4 skliausteliuose nurodytas papildomas kintamųjų kiekis, reikalingas reikšmių kodavimui dvejetainiais. Nuoseklaus algoritmo atveju didžioji naudojamos atminties dalis tenka tikėtinumo vektorių $a[u]$, $b[u]$ bei Erlango šakų tankių saugojimui ir priklauso nuo T , R . Tuo tarpu P-1 algoritmas didžiąją atminties dalį panaudoja matricių $\Omega_i^{(2)}[\ell]$, $\Omega_{i,j}^{(3)}[\ell]$ saugojimui. Tik šį kartą reiklumas atminčiai priklauso nuo serijų kiekio L ir yra mažesnis nei nuoseklaus algoritmo, kai $R \ll L \ll T$. Algoritmas P-2 vietoj matricių atmintyje laiko vektorius $\omega_i^{(2)}[\ell]$, $\omega_{i,j}^{(3)}[\ell]$, tad jo reiklumas atminčiai yra dar mažesnis nei P-1. Algoritmo modifikacija P-1-D papildomai naudoja atmintį Erlango šakų tankiams saugoti, tačiau bendras reiklumas atminčiai yra mažesnis nei nuoseklaus algoritmo. Algoritmas P-3 (ir jo modifikacija P-3-D) naudoja daugiausiai atminties.

Toliau lentelėje 5 pateikiama algoritmų sudėtingumo analizė, kurioje išskiriami nuoseklūs ir lygiagretūs skaičiavimų etapai.

5 lentelė. Skaičiavimo sudėtingumo eilė pagal atliekamų daugybos operacijų skaičių.

Skaičiavimai	Nuoseklus alg.	P-1 alg.	P-2-D alg.	P-3-D alg.	
Šakų tankiai $f_i(x)$	RT	RT/L	RT/L	RT/L	P1
Matricos $U[\ell]$		R^3T/L	R^3T/L	R^3T/L	
Dalinės sumos		$\Omega_{i,j}^{(3)}[\ell]: R^5T/L$			
Vektoriai $\pi^\ell, \mathbb{1}^\ell$		R^2L	R^2L	R^2L	S
Šakų tankiai $f_i(x)$			P-2 only: RT/L	P-3 only: RT/L	P2
Dalinės sumos			$\omega_{i,j}^{(3)}[\ell]: R^4T/L$		
$\alpha[u], b[u]$	R^2T			R^2T/L	
Šakų tankiai $f_i(x)$				P-3 only: RT/L	P3
Dalinės sumos				$S_{i,j}^{(3)}[\ell]: R^2T/L$	
Π, λ, π	R^2T	R^4L	R^3L		
Bendras sudėtingumas	$\mathcal{O}(R^2T)$	$\mathcal{O}(R^5T/L)$	$\mathcal{O}(R^4T/L)$	$\mathcal{O}(R^3T/L)$	

Kaip ir atminties kiekio analizėje, skaičiavimo sudėtingumą nurodo R, L ir T . Nuoseklaus algoritmo sudėtingumas yra $\mathcal{O}(R^2T)$, o lygiagrečių algoritmų atveju vertiname skaičiavimo sudėtingumą, tenkantį kiekvienai iš L gijų. Algoritmuose P-3, P-3-D didžiausia sudėtingumo dalis tenka matricių $U[\ell]$ skaičiavimui ir yra $\mathcal{O}(R^3T/L)$. Algoritmų P-2, P-2-D skaičiavimo sudėtingumas didesnis – $\mathcal{O}(R^4T/L)$. Tačiau kai šakų kiekis R yra nedidelis, šie algoritmai gali būti greitesni palyginti su P-3 (P-3-D) dėl efektyvesnio atminties naudojimo.

3.5. Praktinis algoritmų palyginimas

Praktiniam algoritmų palyginimui lygiagrečias jų versijas realizavome panaudodami CUDA biblioteką skaičiavimams atlikti su vaizdo procesoriumi (GPU). Disertacijoje aptariame būdus, kaip sumažinti skaitinių paklaidų įtaką. Visų programų (įskaitant ir tiriamosios) kodai pateikti <https://github.com/minbraz/parmap>.

20 apibrėžimas. *Kerneliu vadinama C++ kalba parašyta programos dalis, kuri yra vykdoma naudojant vaizdo procesorių.*

Tinkamiausias lygiagrečiai vykdomų gijų skaičius $L_{opt}(R)$ priklauso nuo konkretaus kernelio naudojamų išteklių, pvz., registrų skaičiaus. Kernelio registrų skaičiaus priklausomybė nuo šakų skaičiaus R yra labai sudėtinga – nulemta to, kaip parašytas programos kodas ir kaip jis yra sukompiliuojamas. Dėl to naudojame fiksuotą (visiems R) ir gana didelį gijų

skaičių L . Kadangi patikimai negalime nustatyti pakankamų gijų skaičiaus L , skaičiavimus pakartojame su $L = kL^*$, $k = 1, 2, 4$, čia $L^* = 1920$ – mūsų vaizdo procesoriaus (GTC 1070) CUDA branduolių skaičius.

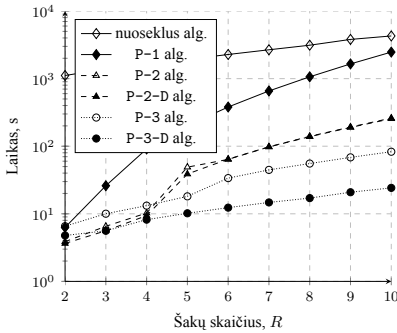
Testavimui sugeneravome $T = 50\,000\,000$, tarplaikių imitaciniu būdu modeliudami stacionarų Markovo atvykimų procesą, nurodytą parametrais

$$\mathbf{r} = [3 \quad 3 \quad 2 \quad 1 \quad 1], \mathbf{\lambda} = [1 \quad 3 \quad 2 \quad 1 \quad 3], \mathbf{\Pi} = \frac{1}{10} \begin{bmatrix} 1 & 2 & 1 & 4 & 2 \\ 3 & 1 & 2 & 1 & 3 \\ 2 & 3 & 1 & 1 & 3 \\ 4 & 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 3 & 2 \end{bmatrix}.$$

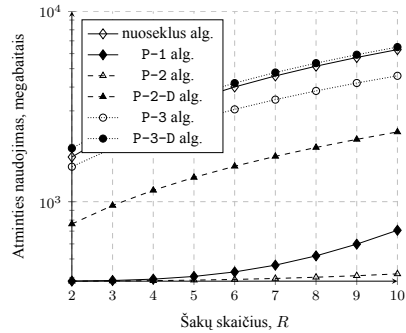
Gautai tarplaikių sekai ieškome labiausi tikėtinų stacionaraus Markovo atvykimų proceso parametrų visoms galimoms ER-CHMM struktūroms su 10 būsenų (ir $R \geq 2$). Atlikome po 100 kiekvieno algoritmo iteracijų. Vykdyto laikai pateikti lentelėje 6 ir pavaizduoti paveiksle 16. Algoritmų naudojamas atminties kiekis pateiktas paveiksle 17.

6 lentelė. Algoritmų realizacijų vykdymo laikas sekundėmis, kuomet gijų skaičius yra $L = 2L^*$.

R	Nuo-seklus alg.	P-1 alg.	P-2 alg.	P-2-D alg.	P-3 alg.	P-3-D alg.
2	1111,521	6,297	3,870	3,644	6,557	4,775
3	1395,008	26,018	6,539	5,611	10,033	5,607
4	1605,503	91,137	10,267	9,399	13,206	8,195
5	1923,365	201,369	48,999	38,758	18,121	10,181
6	2276,253	380,215	63,872	64,392	33,734	12,336
7	2671,527	657,038	97,559	98,059	44,649	14,734
8	3122,541	1063,828	138,394	140,038	55,413	17,078
9	3803,840	1650,562	189,625	192,288	68,378	20,859
10	4278,771	2469,702	255,107	260,547	82,898	24,186



16 pav. Algoritmų realizacijų vykdymo laikas, kai $L = 2L^*$.



17 pav. Algoritmų realizacijų atminties naudojimas, kai $L = 2L^*$.

Iš lentelėje 6 pateikto vykdymo laiko matome, jog P-2-D algoritmo realizacija yra greitesnė už P-3-D tik tuomet, kai $R = 2$. Taip yra dėl to, kad kuomet R yra nedidelis, kiekviena gija gali sutalpinti visus (arba didesnę dalį) darbinis kintamuosius registrų atmintyje, dėl to paspartėja operacijų atlikimas. Kai $R > 4$, pastebimas žymus vykdymo laiko trukmės padidėjimas, nes didžioji dalis gijos darbinų kintamųjų nebesutelpa registrų atmintyje ir tenka naudoti lėtesnę atmintį.

Kuomet $R > 2$, greičiausia yra P-3-D algoritmo realizacija.

3.6. Išvados

Pristatyti trys lygiagretūs EM metodo algoritmai Markovo atvykimų proceso parametrų paieškai atlikti. Algoritmai P-1 ir P-2 gauti pritaikius paslėptųjų Markovo modelių paieškos Baum-Welch algoritmą. Algoritmas P-1 skaičiavimus atlieka matricomis, tuo tarpu algoritme P-2 naudojami vektoriai. Dėl to algoritmo P-2 skaičiavimo sudėtingumas, $\mathcal{O}(R^5 T/L)$, yra viena eile aukštesnis lyginant su algoritmo P-1, $\mathcal{O}(R^4 T/L)$. Trečiuoju algoritmu P-3 labiau tikėtini parametrų įverčiai skaičiuojami tiesiogiai panaudojant tikėtinumo vektorius; algoritmo skaičiavimo sudėtingumas yra $\mathcal{O}(TR + LR^2)$.

Atlikta algoritmų reiklumo atminčiai sudėtingumo analizė. Algoritmo P-1 reiklumo atminčiai (vertinant kintamųjų skaičių) sudėtingumas yra $\mathcal{O}(T + LR^4)$. Algoritmas P-2 yra mažiau reiklus atminčiai, nes vietoj ma-

trijų skaičiavimai atliekami vektoriais; reiklumo atminčiai sudėtingumas $\mathcal{O}(T + LR^3)$. Algoritmas P-3 yra labiausiai reiklus atminčiai ir yra $\mathcal{O}(TR + LR^2)$. Taip yra todėl, kad atmintyje turi būti saugomi tikėtinumo vektoriai.

Algoritmų realizacijų veikimo sparta priklauso nuo konkrečios skaičiavimo platformos. Algoritmas P-1 iteracijos metu duomenis nuskaityti vieną kartą, tai gali būti svarbu, jeigu duomenų nuskaitymo greitis nėra didelis. Kitu atveju reikėtų rinktis tarp algoritmų P-2 ir P-3. Remiantis sudėtingumo analize, algoritmas P-2 yra mažiausiai reiklus atminčiai ir gali būti gana greitas, kuomet Erlango šakų skaičius nėra didelis. Priešingai, algoritmo P-3 reiklumas yra didžiausias, tačiau jo skaičiavimo sudėtingumo priklausomybė nuo Erlango šakų skaičiaus yra mažiausia. Siekiant taikyti vieną iš algoritmų, reikia įvertinti skaičiavimo platformos išteklius ir aproksimuojamų duomenų kieki.

Siekiant parodyti, kaip algoritmai veikia praktiškai, jie buvo realizuoti veikimui panaudojant vaizdo procesorių (GPU). Palyginimui naudojamai nuosekliai algoritmo versijai taikyta tikslesnio skaičių sumavimo metodika. Visoms algoritmų realizacijoms, siekiant išvengti skaitinių problemų, pritaikyta skaičiaus kodavimo dvejetainio laipsniais metodika. Papildomai realizuotos dvi algoritmų modifikacijos: P-2-D ir P-3-D, kuriose apskaičiuoti Erlango šakų tankiai išsaugomi atmintyje vėlesniam naudojimui. Šios modifikacijos padidino algoritmų reiklumą atminčiai. Algoritmai palyginti aproksimuojant sugeneruotą 50 000 000 tarplaikių duomenis, atliekant po 100 iteracijų. Analizuojant eksperimentinių skaičiavimų rezultatus nustatyta:

1. Algoritmo modifikacija P-2-D yra mažiau reikli atminčiai, lyginant su P-2, tačiau skaičiavimo laikas skiriasi nedaug.
2. Algoritmo modifikacijoje P-3-D, dėl Erlango šakų tankių išsaugojimo atmintyje buvo pasiektos trumpiausios vykdymo trukmės.

4. NUTRŪKSTANČIO MARKOVO ATVYKIMŲ PROCESO PARAMETRŲ PAIEŠKA

4.1. Maksimalaus tikėtimumo metodo pritaikymas parametrų paieškai

Maksimalaus tikėtimumo (EM) metodą pritaikome nutrūkstančio Markovo atvykimų proceso parametrų paieškai. Apsiribojame tik ER-CHMM struktūrą turinčiais procesais. Mūsų algoritmas nuo jau aptarto Markovo atvykimų procesų parametrų paieškos algoritmo skiriasi tuo, jog jis įvertina tarplaikių sekos ilgį. Kadangi tarplaikių sekos yra baigtinės (atsitiktinio ilgio), pradinių tikimybių skirstinio įtaka yra reikšminga. Dėl to šį skirstinį taip pat įvertiname (palyginimui – Markovo atvykimų procesų atveju naudojome stacionarų pradinių tikimybių skirstinį).

Stebimi duomenys – tai U atsitiktinių dydžių (tarplaikių) serijos

$$\mathcal{X} = \left(\mathcal{X}_1^{(1)}, \mathcal{X}_2^{(1)}, \dots, \mathcal{X}_{\mathcal{K}_1}^{(1)}, \dots, \mathcal{X}_1^{(U)}, \mathcal{X}_2^{(U)}, \dots, \mathcal{X}_{\mathcal{K}_U}^{(U)} \right).$$

Galimai koreliuoti tarplaičiai interpretuojami kaip laiko tarpai tarp nutrūkstančio Markovo atvykimų proceso generuojamų įvykių laiko momentų. Užduotis – surasti labiausiai tikėtinus proceso parametrus stebimų tarplaikių atžvilgiu.

Tarkime, kad atsitiktinių laikų serijas generuoja nuosekliai vienas po kito vykstantys tolydaus laiko Markovo procesai

$$\mathcal{J}_t^{(1,1)}, \dots, \mathcal{J}_t^{(1,\mathcal{K}_1)}, \dots, \mathcal{J}_t^{(U,1)}, \dots, \mathcal{J}_t^{(U,\mathcal{K}_U)},$$

kur laiko momentu $t = \mathcal{X}_1^{(1)} + \dots + \mathcal{X}_k^{(u)}$ pereinama nuo proceso $\mathcal{J}_t^{(u,k)}$ prie $\mathcal{J}_t^{(u,k+1)}$, ir t. t. Procesų sekos visiškai charakterizuojamos diskrečiu lankomų būsenų procesu

$$\mathcal{I}_0^{(1,1)}, \dots, \mathcal{I}_{\mathcal{M}^{(1,1)}-1}^{(1,1)}, \dots, \mathcal{I}_m^{(u,k)}, \dots, \mathcal{I}_{\mathcal{M}^{(U,\mathcal{K}_U)}-1}^{(U,\mathcal{K}_U)}$$

ir diskrečiu atsitiktinių laikų trukmių procesu

$$\mathcal{S}_0^{(1,1)}, \dots, \mathcal{S}_{\mathcal{M}^{(1,1)}-1}^{(1,1)}, \dots, \mathcal{S}_m^{(u,k)}, \dots, \mathcal{S}_{\mathcal{M}^{(U,\mathcal{K}_U)}-1}^{(U,\mathcal{K}_U)},$$

kuris nusako praleistą laiką kiekvieno apsilankymo metu. Čia $\mathcal{M}^{(u,k)}$ yra perėjimų skaičius prieš įvykstant įvyki generuojančiam perėjimui pagal \mathbf{D}_1 . Visą informaciją apie tarplaikių seką $\mathbf{x} = (x_1^{(1)}, \dots, x_{K_U}^{(U)})$ sudaro

$$\mathbf{y} = \left(i_0^{(1,1)}, \dots, i_{m^{(1,1)}-1}^{(1,1)}, \dots, i_m^{(u,k)}, \dots, i_{m^{(U,K_U)}-1}^{(U,K_U)}, \right. \\ \left. s_0^{(1,1)}, \dots, s_{m^{(1,1)}-1}^{(1,1)}, \dots, s_m^{(u,k)}, \dots, s_{m^{(U,K_U)}-1}^{(U,K_U)} \right).$$

Papildomai išskiriame procesą

$$\mathcal{G}_1^{(1)}, \dots, \mathcal{G}_{K_1}^{(1)}, \dots, \mathcal{G}_k^{(u)}, \dots, \mathcal{G}_{K_U}^{(U)},$$

kur $\mathcal{G}_k^{(u)}$ – Erlango šakos, generuojančios tarplaiki $\mathcal{X}_k^{(u)}$, numeris.

Jeigu turėtume $\mathbf{g} = (g_1^{(1)}, g_2^{(1)}, \dots, g_{K_U}^{(U)})$ reikšmes konkrečiai tarplaiikių sekai $\mathbf{x} = (x_1^{(1)}, x_2^{(1)}, \dots, x_{K_U}^{(U)})$, tikėtinumo funkcijos išraiška būtų

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda} \mid \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{G}}) = \prod_{u=1}^U \prod_{k=1}^{K_u} f_{g_k^{(u)}}(x_k^{(u)}),$$

o maksimalaus tikėtinumo parametrus rastume pagal [21]

$$\boldsymbol{\lambda}_i = \frac{\sum_{u=1}^U \sum_{k=1}^{K_u} \mathbf{r}_i \mathbb{1}_{\{g_k^{(u)}=i\}}}{\sum_{u=1}^U \sum_{k=1}^{K_u} x_k^{(u)} \mathbb{1}_{\{g_k^{(u)}=i\}}}, \\ \boldsymbol{\Pi}_{i,j} = \frac{\sum_{u=1}^U \sum_{k=1}^{K_u-1} \mathbb{1}_{\{g_k^{(u)}=i, g_{k+1}^{(u)}=j\}}}{\sum_{u=1}^U \sum_{k=1}^{K_u} \mathbb{1}_{\{g_k^{(u)}=i\}}}, \\ \boldsymbol{\pi}_i = \frac{1}{U} \sum_{u=1}^U \mathbb{1}_{\{g_1^{(u)}=i\}}.$$

Tačiau informacijos, apie tai, kokia Erlango šaka buvo pasirenkama generuojant kiekvieną iš tarplaiikių, neturime. Sukonstruojame pakankamą statistiką \mathcal{H} , kuri charakterizuoja visą informaciją \mathcal{Y} , taip

$$\mathcal{H} = \left\{ \mathcal{Q}_i^{(u)}[k], \mathcal{Q}_{i,j}^{(u)}[k] \right\},$$

$\mathcal{Q}_i^{(u)}[k]$ – indikatorius, nurodantis, ar $\mathcal{X}_k^{(u)}$ buvo sugeneruotas i -tosios Erlango šakos

$$\mathcal{Q}_i^{(u)}[k] = \mathbb{1}_{\{\mathcal{I}_0^{(u,k)} = s_i\}},$$

$\mathcal{Q}_{i,j}^{(u)}[k]$ – indikatorius, nurodantis, jog sugeneravus $\mathcal{X}_k^{(u)}$ su i -tąja Erlango šaka, toliau pasirenkama j -toji Erlango šaka

$$\mathcal{Q}_{i,j}^{(u)}[k] = \mathbb{1}_{\{\mathcal{I}_0^{(u,k)} = s_i, \mathcal{I}_0^{(u,k+1)} = s_j\}}.$$

Prieš išvedant formules statistikos \mathcal{H} įverčio \mathbf{h} radimui, apibrėžiame tikėtinumo vektorius

$$\mathbf{a}^{(u)}[k] = \boldsymbol{\pi} \mathbf{P}^{(u)}[k], \quad \mathbf{b}^{(u)}[k] = \mathbf{P}^{(u)}[K_u] \mathbb{1},$$

čia $\mathbf{P}^{(u)}[s]$ yra stebėjimų sekos $x_l^{(u)}, \dots, x_s^{(u)}$ tikėtinumo matrica

$$\mathbf{P}^{(u)}[s] = \prod_{k=l}^s \mathbf{P}^{(u)}[k],$$

kur $\mathbf{P}^{(u)}[k]$ yra tarplaiikiui $x_k^{(u)}$ apibrėžta Erlango pasirenkamų šakų tikėtinumo matrica

$$\mathbf{P}^{(u)}[k] = \begin{bmatrix} f_1(x_k^{(u)}) \boldsymbol{\Pi}_{1,1} & \dots & f_1(x_k^{(u)}) \boldsymbol{\Pi}_{1,R} \\ \vdots & \ddots & \vdots \\ f_R(x_k^{(u)}) \boldsymbol{\Pi}_{R,1} & \dots & f_R(x_k^{(u)}) \boldsymbol{\Pi}_{R,R} \end{bmatrix}.$$

Tuomet parametrų tikėtinumo funkcija užrašoma taip:

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda} \mid \mathcal{X}) = \prod_{u=1}^U \boldsymbol{\pi} \left(\prod_{k=1}^{K_u} \mathbf{P}^{(u)}[k] \right) \mathbb{1}, \quad (30)$$

o įvertis \mathbf{h} surandamas pagal formules

$$\begin{aligned}
q_i^{(u)}[k] &= \mathbb{E}[\mathcal{Q}_i^{(u)}[k] \mid \mathcal{X}] \\
&= \mathbb{P}(\mathcal{I}_0^{(u,k)} = \mathbf{s}_i \mid \mathcal{X}) = \frac{\mathbb{P}(\mathcal{I}_0^{(u,k)} = \mathbf{s}_i, \mathcal{X})}{\mathbb{P}(\mathcal{X})} \\
&= \frac{\left(\mathbf{a}_i^{(u)}[k-1] \mathbf{b}_i^{(u)}[k] \right) \prod_{v \neq u} \boldsymbol{\pi} \mathbf{b}^{(v)}[1]}{\prod_{v=1}^U \boldsymbol{\pi} \mathbf{b}^{(v)}[1]} \\
&= \frac{\mathbf{a}_i^{(u)}[k-1] \mathbf{b}_i^{(u)}[k]}{\boldsymbol{\pi} \mathbf{b}^{(u)}[1]}, \tag{31}
\end{aligned}$$

$$\begin{aligned}
q_{i,j}^{(u)}[k] &= \mathbb{E}[\mathcal{Q}_{i,j}^{(u)}[k] \mid \mathcal{X}] \\
&= \mathbb{P}(\mathcal{I}_0^{(u,k)} = \mathbf{s}_i, \mathcal{I}_0^{(u,k+1)} = \mathbf{s}_j \mid \mathcal{X}) \\
&= \frac{\mathbb{P}(\mathcal{I}_0^{(u,k)} = \mathbf{s}_i, \mathcal{I}_0^{(u,k+1)} = \mathbf{s}_j, \mathcal{X})}{\mathbb{P}(\mathcal{X})} \\
&= \frac{\mathbf{a}_i^{(u)}[k-1] f_i(x_k) \boldsymbol{\Pi}_{i,j} \mathbf{b}_j^{(u)}[k+1]}{\boldsymbol{\pi} \mathbf{b}^{(u)}[1]}.
\end{aligned}$$

Pasinaudoję visą informaciją \mathbf{y} charakterizuojančios pakankamos statistikos įverčių \mathbf{h} , surandame maksimalaus tikėtimumo parametrus pagal formules

$$\boldsymbol{\lambda}_i = \frac{\mathbf{r}_i \sum_{u=1}^U \sum_{k=1}^{K_u} q_i^{(u)}[k]}{\sum_{u=1}^U \sum_{k=1}^{K_u} x_k^{(u)} q_i^{(u)}[k]}, \tag{32}$$

$$\boldsymbol{\Pi}_{i,j} = \frac{\sum_{u=1}^U \sum_{k=1}^{K_u-1} q_{i,j}^{(u)}[k]}{\sum_{u=1}^U \sum_{k=1}^{K_u-1} q_i^{(u)}[k]}, \tag{33}$$

$$\boldsymbol{\pi}_i = \frac{1}{U} \sum_{u=1}^U q_i^{(u)}[1]. \tag{34}$$

Parametrų įverčiai aptartu maksimalaus tikėtimumo metodu ieškomi pagal algoritmą 6.

6 algoritmas. Algoritmas nutrūkstančio Markovo atvykimų proceso, turinčio ER-CHMM struktūrą, parametrų paieškai atlikti maksimalaus tikėtinumo metodu.

procedure *TMAP-EM-fitting*(\mathbf{x}, \mathbf{r})

1. Atsižvelgiant į \mathbf{r} sugeneruojami parametrai $(\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda})$
2. Apskaičiuojamas $(\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda})$ tikėtinumas pagal (30)
3. **repeat**
4. Apskaičiuojamas įvertis \mathbf{h} pagal (31)
5. Apskaičiuojami nauji parametrai $(\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda})$ pagal (32), (33), (34)
6. Apskaičiuojamas $(\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda})$ tikėtinumas pagal (30)
7. **until** kol parametrų tikėtinumas didėja
8. **return** $(\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda})$

end procedure

4.2. Parametrų paieškos algoritmo išlygiagretinimas

Pateikto algoritmo išlygiagretinimas gali būti gautas neįvedant papildomų skaičiavimų. Įverčio \mathbf{h} elementai gali būti skaičiuojami nepriklausomai kiekvienai u -tajai tarplaičių serijai $x_1^{(u)}, \dots, x_{K_u}^{(u)}$. Maksimalaus tikėtinumo parametrų formulėse (32), (33) esančių sumų skaičiavimą galima taip pat išlygiagretinti.

Turime L skaičiavimo gijų. Prieš pradėdami skaičiavimus, kiekvienai ℓ -tajai gijai priskiriame konkrečias stebėjimų serijas

$$\Phi^{(\ell)} = (u_1, u_2, \dots)$$

taip, kad visoms gijoms tektų kiek įmanoma panašesnis tarplaičių $x_k^{(u)}$ skaičius.

Toliau apibrėžiame dalines sumas

$$\begin{aligned} S_i^{(1)}[\ell] &= \sum_{u \in \Phi^{(\ell)}} \sum_{k=1}^{K_u} q_i^{(u)}[k], & S_i^{(2)}[\ell] &= \sum_{u \in \Phi^{(\ell)}} \sum_{k=1}^{K_u} x_k^{(u)} q_i^{(u)}[k], \\ S_{i,j}^{(3)}[\ell] &= \sum_{u \in \Phi^{(\ell)}} \sum_{k=1}^{K_u-1} q_{i,j}^{(u)}[k], & S_i^{(3)}[\ell] &= \sum_{u \in \Phi^{(\ell)}} \sum_{k=1}^{K_u-1} q_i^{(u)}[k], \end{aligned} \quad (35)$$

per kurias išreiškiame maksimalaus tikėtinumo formules parametrus $\boldsymbol{\lambda}$ ir $\boldsymbol{\Pi}$

$$\boldsymbol{\lambda}_i = \frac{\mathbf{r}_i \sum_{\ell=1}^L S_i^{(1)}[\ell]}{\mathbf{r}_i \sum_{\ell=1}^L S_i^{(2)}[\ell]}, \quad \boldsymbol{\Pi}_{i,j} = \frac{\sum_{\ell=1}^L S_{i,j}^{(3)}[\ell]}{\sum_{\ell=1}^L S_i^{(3)}[\ell]}. \quad (36)$$

Išlygiagretinta parametrų įverčių paieška maksimalaus tikėtinumo metodu pateikta algoritme 7.

7 algoritmas. Lygiagretus algoritmas nutrūkstančio Markovo atvykimų proceso, turinčio ER-CHMM struktūrą, parametrų paieškai maksimalaus tikėtinumo metodu.

procedure *TMAP-EM-parallel-fitting*(\mathbf{x}, \mathbf{r})

1. Atsižvelgiant į \mathbf{r} sugeneruojami parametrai ($\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda}$)
2. Apskaičiuojamas ($\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda}$) tikėtinumas pagal (30)
3. **repeat**
4. **parallel for** $\ell = 1$ to L **do**
5. Apskaičiuojami įverčio \mathbf{h} elementai priklausantys $\Phi^{(\ell)}$ serijoms pagal (31)
6. **end for**
7. **parallel for** $\ell = 1$ to L **do**
8. Apskaičiuojamos dalinės sumos pagal (35)
9. **end for**
10. Apskaičiuojami nauji parametrai $\boldsymbol{\Pi}, \boldsymbol{\lambda}$ pagal (36), ir $\boldsymbol{\pi}$ pagal (34)
11. Apskaičiuojamas ($\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda}$) tikėtinumas pagal (30)
12. **until** kol parametrų tikėtinumas didėja
13. **return** ($\boldsymbol{\pi}, \boldsymbol{\Pi}, \boldsymbol{\lambda}$)

end procedure

4.3. Eksperimentiniai skaičiavimai

Nuoseklų ir lygiagretų algoritmus realizavome C++ programavimo kalba. Lygiagretiems skaičiavimams atlikti su vaizdo procesoriais (GPU) naudojome CUDA biblioteką. Kaip ir Markovo atvykimų proceso parametrų paieškos atveju, išmėginome tris variantus, kuomet visi skaičiavimai padalijami į $L = kL^*$, $k = 1, 2, 4$ dalis. Čia $L^* = 1920$ yra mūsų vaizdo procesoriaus CUDA branduolių kiekis.

Laisvai pasirinkome proceso parametrus

$$\mathbf{r} = [3 \quad 3 \quad 2 \quad 1 \quad 1], \boldsymbol{\lambda} = [1 \quad 3 \quad 2 \quad 1 \quad 3],$$

$$\boldsymbol{\Pi} = \frac{1}{10} \begin{bmatrix} 1 & 2 & 1 & 3 & 2 \\ 3 & 1 & 2 & 1 & 2 \\ 2 & 2 & 1 & 1 & 3 \\ 3 & 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 2 & 2 \end{bmatrix}. \quad (37)$$

Imitaciniu modeliavimu sugeneravome tarplaikių serijų rinkinį, kurį sudaro 50 000 000 tarplaikiai.

Atsitiktinai sugeneruotiems parametrų pritaikėme anksčiau aprašytus nuoseklų ir lygiagretų algoritmus, kuriuose vykdomė po 100 iteracijų. Sugeneravome po vieną pradinį parametrų rinkinį kiekvienai iš galimų ER-CHMM struktūrų su 10 būsenų ($R \geq 2$).

Lentelėje 7 pateikiame parametrų, gautų atlikus 100 iteracijų, tikėtinumo funkcijos logaritmuotas reikšmes. Akivaizdu, jog 100 iteracijų nepakako pasiekti parametrų (37) tikėtinumo funkcijos reikšmės $-79212363,236$. Tačiau svarbiausia, kad tikėtinumo funkcijos reikšmė labai nesiskiria skaičiuojant nuosekliu ir lygiagrečiais algoritmais.

7 lentelė. Parametrų, gautų atlikus 100 paieškos algoritmo iteracijų, tikėtinumo funkcijos logaritmuotas reikšmės.

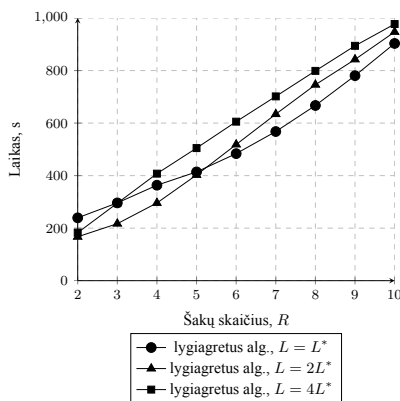
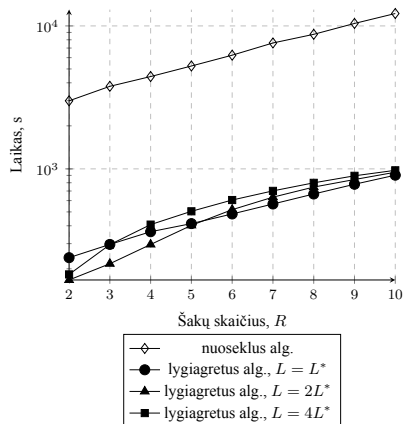
R	Nuoseklus alg.	Lygiagretus alg., $L = L^*$	Lygiagretus alg., $L = 2L^*$	Lygiagretus alg., $L = 4L^*$
2	-79611116	-79611134	-79611136	-79611134
3	-79452224	-79452257	-79452234	-79452230
4	-79425934	-79425984	-79425953	-79425941
5	-79410651	-79410690	-79410670	-79410664
6	-79443226	-79443274	-79443258	-79443239
7	-79476124	-79476131	-79476120	-79476133
8	-79516724	-79516726	-79516734	-79516719
9	-79516761	-79516762	-79516747	-79516750
10	-79683891	-79683290	-79683406	-79683404

Algoritmo vykdymo laikas sekundėmis pateiktas lentelėje 8 ir paveiksle 18. Pagreitėjimas, skaičiuojant lygiagrečiu algoritmu, siekia nuo 13 iki 18 kartų, priklausomai nuo šakų skaičiaus R . Tai nėra daug lyginant su rezultatais, gautais atliekant analogiškus eksperimentus su Markovo atvykimų proceso parametrų paieška. To priežastis – skirtingi serijų ilgiai, kiekvienu laiko momentu kiekviena iš skaičiavimo gijų atlieka skirtingas skaičiavimo operacijas. Dėl to neefektyviai išnaudojama skaičiavimo platforma (GPU), kuri yra pritaikyta sparčiai atlikti tas pačias operacijas (su skirtingais duomenimis).

8 lentelė. Nutrūkstančio Markovo atvykimų proceso parametru paieškos 100 iteracijų vykdymo laikas sekundėmis.

R	Nuoseklus alg.	Lygiagretus alg., $L = L^*$	Lygiagretus alg., $L = 2L^*$	Lygiagretus alg., $L = 4L^*$
2	2992,35	239,27	167,06	182,72
3	3779,80	296,15	216,99	295,09
4	4423,85	363,43	295,82	407,54
5	5242,63	414,14	402,48	504,83
6	6239,32	483,69	518,32	605,46
7	7598,72	567,83	634,03	701,60
8	8732,11	667,00	745,96	798,75
9	10404,82	780,53	842,25	894,20
10	12209,27	903,22	947,06	977,08

18 pav. Nutrūkstančio Markovo atvykimų proceso parametru paieškos 100 iteracijų vykdymo laikas sekundėmis.



4.4. Išvados

Sudarytos EM metodo formulės nutrūkstančio Markovo atvykimų (TMAP) proceso, turinčio Erlango hiperstruktūrą, parametru paieškai atlikti. Realizuotos nuosekli ir lygiagreti algoritmo procedūros. Pastaroji pritaikyta skaičiavimui panaudojant vaizdo procesorių (GPU). Buvo sugeneruoti 50 000 000

tarplaikiai ir panaudoti ieškant TMAP proceso parametrų. Gautos tokios išvados:

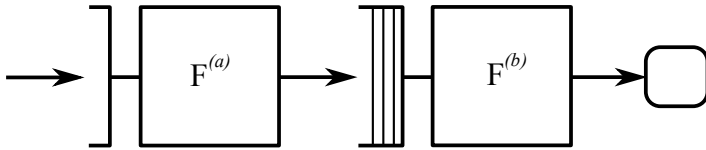
- Nuoseklaus algoritmo realizacijos atveju, siekiant išvengti paklaidų, reikėjo taikyti skaičių sumavimo metodiką.
- Tiek nuosekliai, tiek lygiagrečiai algoritmų realizacijoms reikėjo pritaikyti skaičiaus kodavimo dvejetainio laipsniais metodiką.
- Dėl duomenų specifikos (įvairaus ilgio tarplaikių sekų) nepavyko užtikrinti, kad būtų efektyviai atliekamos atminties ir skaičiavimo operacijos. Dėl to lygiagretaus algoritmo realizacija buvo ne tokia sparti (tik 15–20 kartų spartesnė, lyginant su MAP proceso parametrų paieška).

5. BAIGTINĖS APTARNAVIMO EILĖS MODELIS

Šiame skyriuje pateikiame baigtinės aptarnavimo eilės modelį. Skirstinius aproksimuojame faziniais skirstiniais dviem būdais: tankia struktūra ir retosiomis struktūromis (su $m = 2n$ perėjimais). Tiriama, kaip skiriasi aproksimavimas tankia / retosiomis struktūromis ir kaip tai lemia apskaičiuotas modelio charakteristikas.

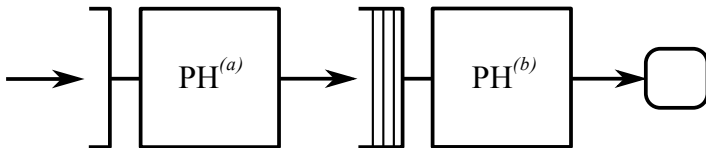
5.1. Modelio aprašas

Aprašome baigtinės aptarnavimo eilės modelį. Tariame, jog paraiškų skaičius baigtinis ir yra lygus L . Priimame, jog laiko intervalai tarp paraiškų priėmimo turi skirstinį $F^{(a)}$. Priimtoms paraiškos aptarnaujamos po vieną per atsitiktinį laiko tarpą, kurio skirstinys yra $F^{(b)}$. Kol paraiška yra aptarnaujama, naujai priimtoms paraiškoms patalpinamos į laukimo eilę (FIFO tipo). Modelio schema pavaizduota paveiksle 19.



19 pav. Baigtinės aptarnavimo eilės modelio schema.

Tam, kad galėtume sudaryti Markovo modelį, skirstinius $F^{(a)}$, $F^{(b)}$ pakeičiame kiek galima tiksliau juos atitinkančiais faziniais skirstiniais $PH^{(a)}$, $PH^{(b)}$. Taip gauname kitą modelį (paveikslas 20), kuris aproksimuoja pradinio modelio elgseną.



20 pav. Baigtinės aptarnavimo eilės modelio aproksimacija, gauta panaudojant fazinius skirstinius.

Modelio konstravimą sudaro du etapai.

Pirmiausia sugeneruojame visų pasiekiamų būsenų aibę

$$\mathcal{V} = \{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n_{\mathcal{V}})}\}.$$

Būsena apibrėžiama vektoriumi

$$\mathbf{v} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3 \quad \mathbf{v}_4],$$

čia $\mathbf{v}_1 \in \{1, \dots, n^{(a)}\}$ nurodo fazinio skirstinio $\text{PH}^{(a)}$ aktyvios fazės numerį ($\mathbf{v}_1 > 0$), $\mathbf{v}_1 = 0$ nurodo, jog fazinis skirstinys yra neaktyvus; analogiškai $\mathbf{v}_3 \in \{1, \dots, n^{(b)}\}$ nurodo fazinio skirstinio $\text{PH}^{(b)}$ aktyvios fazės numerį ($\mathbf{v}_3 > 0$) ir $\mathbf{v}_3 = 0$ nurodo, jog fazinis skirstinys yra neaktyvus; $\mathbf{v}_2 \in \{0, 1, \dots, Fc - 1\}$ nurodo eilėje laukiančių paraiškų skaičių; $\mathbf{v}_4 \in \{0, 1, \dots, Fc\}$ nurodo aptarnautų paraiškų skaičių.

Būsenų aibės generavimą pradedame nuo pradinės būsenos

$$[1 \quad 0 \quad 0 \quad 0].$$

Pritaikę paraiškų priėmimo ir aptarnavimo įvykius, iš kiekvienos jau surastos būsenos sukonstruojame naujas būsenas. Joms vėl pritaikome šiuos įvykius ir taip tęsiame, kol gauname vis naujų būsenų. Toliau būsenų aibę praplečiame pritaikydami perėjimo tarp fazinių skirstinių fazių įvykius ir gauname visų pasiekiamų būsenų aibę. Detalus algoritmas pateiktas disertacijoje.

Sudarome perėjimų tarp būsenų intensyvumų matricą \mathbf{Q} . Iš tiesų perėjimų intensyvumai tarp modelio būsenų priklauso tik nuo konkrečių naudojamų fazinių skirstinių $\text{PH}^{(a)}$, $\text{PH}^{(b)}$. Imame kiekvieną iš modelio būsenų, surandame iš jos pasiekiamas būsenas ir nustatome perėjimo intensyvumus. Išsamus intensyvumų matricos \mathbf{Q} sudarymo algoritmas pateiktas disertacijoje.

Galiausiai modeliavimo rezultatai gaunami pradinių tikimybių vektoriui α apskaičiuojant modelio būsenų tikimybes praėjus laiko tarpui t pagal $\alpha e^{\mathbf{Q}t}$.

5.2. Eksperimentiniai skaičiavimai

Pasirenkame lengvai faziniais skirstiniais aproksimuojamus skirstinius $F^{(a)}$, $F^{(b)}$. Taip galėsime sulyginti Markovo modeliavimo ir imitacinio

modeliavimo rezultatus (disertacijoje pateikimas papildomas pavyzdys, kuriame skirstinys $F^{(a)}$ yra sunkiai aproksimuojamas).

Pasirenkame Veibulo skirstinius, kurio tankio funkcija turi formą

$$F(x) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}, & x \geq 0, \\ 0, & x \leq 0. \end{cases}$$

Laiko intervalams tarp paraiškų atvykimų modeliuoti pasirenkame skirstinį su parametrais $k^{(a)} = 0.9$, $\lambda^{(a)} = 0.5$ (**W3**). Paraiškų aptarnavimo laikų skirstiniui parenkame parametrus $k^{(b)} = 2$, $\lambda^{(b)} = 0.9$ (**W4**). Pastaruoju atveju skirstinys (**W4**) yra mažiau panašus į eksponentinį (nes $k^{(b)}$ reikšmė yra labiau nutolusi nuo vieneto nei $k^{(a)}$), todėl aproksimuosime didesniu fazių skaičiumi.

Parametrų paieškos procedūrą atliekame etapais. Pirmuose etapuose fiksuojamas iteracijų skaičius (maksimalaus tikėtinumo metodo). Tolesni etapai: iš ankstesnio etapo atrenkami labiausiai tikėtini fazinių skirstinių parametrai ir jie toliau gerinami maksimalaus tikėtinumo metodu. Tokia strategija yra paranki aproksimuojant retomis struktūromis, nes taip galima atmesti potencialiai netinkančias struktūras. Paskutiniame etape iteracijų skaičius neribojamas ir jos atliekamos tol, kol parametrų įverčių tikėtimumas konverguoja (t. y. kol jo sąlyginis padidėjimas didesnis nei 10^{-8}). Šis paskutinis etapas palankus aproksimavimui tankia struktūra, nes išnauojamas jos universalumas. Tiesa, aproksimuojant tankia struktūra gali būti pasiekama lėto konvergavimo sritis ir dėl to tolesnė paieška nutraukiama (nesuradus labiau tikėtinų parametrų).

Skirstinį **W3** aproksimuojame ketvirtos eilės fazinių skirstiniu, o **W4** – trečios eilės fazinių skirstiniu. Paieškos procedūrų rezultatai pateikti lentelėse 9, 10.

9 lentelė. W3 skirstinio aproksimavimo ketvirtos eilės faziniu skirstiniu procedūros rezultatai.

Etapas	Struktūrų klasė	Parametrų rinkinių skaičius	Didžiausia 11h reikšmė	Kiek 11h reikšmė didesnė naudojant RND?
#1, 1 min, 100 it.	FULL	538	-0,348543	$3,6997 \cdot 10^{-4}$
	RND	491	-0,348174	
#2, 1 min, 200 it.	FULL	261	-0,348421	$2,6294 \cdot 10^{-4}$
	RND	288	-0,348158	
#3, 1 min, 300 it.	FULL	162	-0,348304	$1,9764 \cdot 10^{-4}$
	RND	184	-0,348107	
#4, 3 min	FULL	5	-0,348039	$-1,7561 \cdot 10^{-5}$
	RND	23	-0,348056	

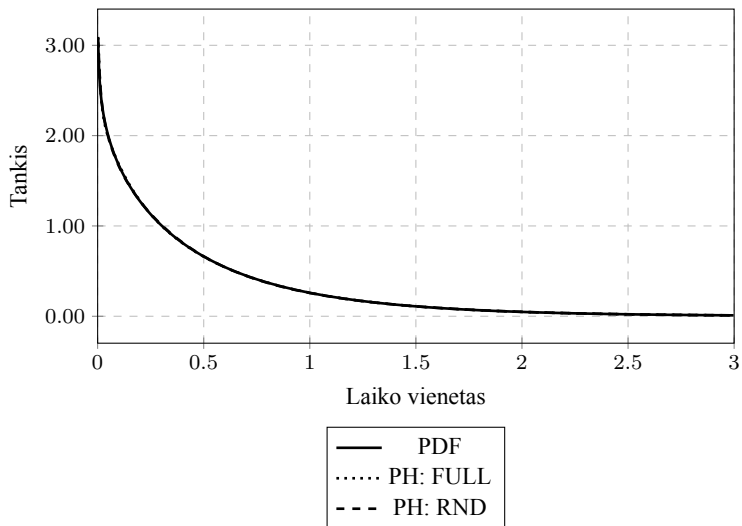
10 lentelė. W4 skirstinio aproksimavimo ketvirtos eilės faziniu skirstiniu procedūros rezultatai.

Etapas	Struktūrų klasė	Parametrų rinkinių skaičius	Didžiausia 11h reikšmė	Kiek 11h reikšmė didesnė naudojant RND?
#1, 1 min, 100 it.	FULL	220	-0,498372	$6,1015 \cdot 10^{-3}$
	RND	112	-0,492270	
#2, 1 min, 200 it.	FULL	113	-0,492634	$1,8886 \cdot 10^{-3}$
	RND	86	-0,490746	
#3, 1 min, 300 it.	FULL	76	-0,491580	$1,1552 \cdot 10^{-3}$
	RND	57	-0,490425	
#4, 3 min	FULL	4	-0,490216	$3,0481 \cdot 10^{-3}$
	RND	7	-0,490186	

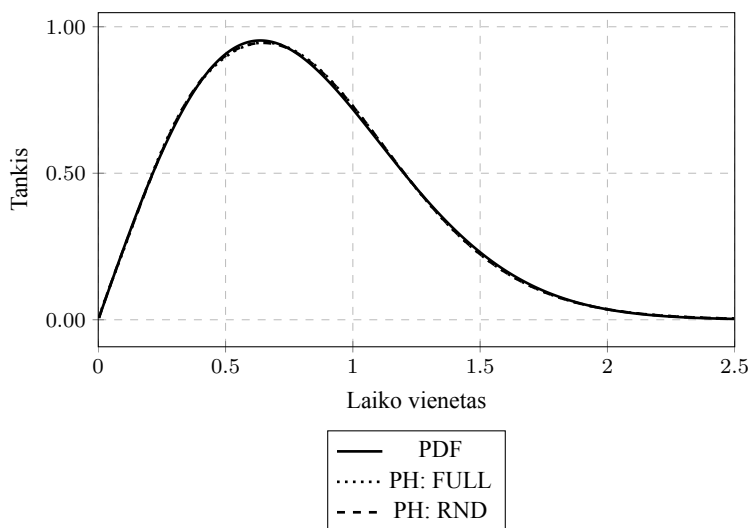
Iš rezultatų, pateiktų lentelėse 9, 10, pastebima tendencija, kad aproksimuojant retomis struktūromis iš pradžių gaunami labiau tikėtini parametru įverčiai. Tačiau paiešką tęsiant skirtumas tarp rastų parametrų įverčių tikėtinumų retomis ir tankia struktūromis mažėja. Gali kad, jog su visa struktūra randami ir labiau tikėtini parametrų įverčiai, jei išmėginamas nepakanamas retų struktūrų skaičius.

Skirstinių **W3**, **W4** ir jų fazinių skirstinių aproksimacijų tankio funk-

cijos pavaizduotos paveiksluose 21, 22.



21 pav. Skirstinio **W3** ir jo fazinio skirstinių aproksimacijų tankio funkcijos.



22 pav. Skirstinio **W4** ir jo fazinio skirstinių aproksimacijų tankio funkcijos.

Kaip matyti, mums pavyko skirstinius **W3**, **W4** aproksimuoti faziniais skirstiniais gana gerai.

Pasirenkame bendrą paraiškų skaičių, $L = 3$.

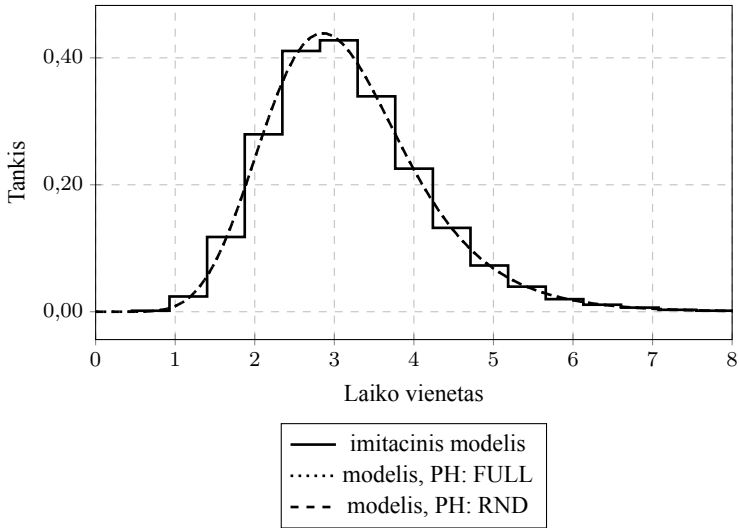
Tam, kad galėtume patikrinti, ar mūsų sudarytas Markovo modelis (modeliuojamos baigtinės paraiškų eilės modelio aproksimacija) tinkamai veikia, realizavome imitacinį modelį. Imitacinis modelis 100 000 kartų vis iš naujo generuoja proceso eigą (su **W3**, **W4** skirstiniais), iš kurių vėliau išskaičiuojamos dominančios charakteristikos.

Viena iš tokių charakteristikų – laikas, per kurį yra aptarnaujamos visos paraiškos, kurios skaitinės charakteristikos pateiktos lentelėje 11. Kadangi Markovo modelis yra sugeriančioji tolydaus laiko Markovo grandinė (absorbuojanti būseną pasiekama kai aptarnaujama paskutinė paraiška), galime sudaryti paraiškų aptarnavimo fazinį skirstinį. Paveiksle 23 pavaizduota šio skirstinio tankio funkcija.

11 lentelė. Paraiškų aptarnavimo laiko skaitinės charakteristikos.

	Vidurkis	Stand. nuokrypis	Asim. koef.
Imitacinis modelis	3,18208	1,01053	0,91849
Modelis, PH: FULL	3,18378	1,01482	0,91021
Modelis, PH: RND	3,18368	1,01295	0,89380

23 pav. Paraiškų aptarnavimo laiko skirstinio tankio funkcija.



5.3. Išvados

Palygintas aproksimavimas faziniais skirstiniais naudojant retas (su $m = 2n$ perėjimais) bei tankią struktūrą konstruojant baigtinės eilės modelį. Modeliavimo rezultatai praktiškai nesiskiria nuo to, kokio tipo struktūra naudojama. Tačiau pastebėti du dalykai. Pirma, naudojant retas struktūras, labiau tikėtini parametrai surandami anksčiau. Antra, nepaisant to, jog buvo išmėgintas ganėtinai nedidelis atsitiktinai sugeneruotų struktūrų skaičius (ypač atvejui $n = 17$), surastų parametrų įverčių tikėtinumai yra ganėtinai panašūs į analogiškus rezultatus, gautus naudojant tankią struktūrą. Remiantis šiuo pastebėjimu galima numanyti, kad minimali struktūrų aibė, kuri padengia visą fazinių skirstinių erdvę, yra nedidelė.

Siekiant geriau išnaudoti retų struktūrų konvergavimą, reikia sukurti tinkamos struktūros identifikavimo metodiką.

IŠVADOS

1. Sukurtas fazinių skirstinių matricinės formos reprezentacijos struktūrų generavimo algoritmas.
2. Remiantis empiriniais tyrimais (atvejui $n = 4$) pagrįsta hipotezė apie tai, jog iš praktinės pusės, pakanka fazinių skirstinių reprezentuoti su $2n$ perėjimais.
3. Aproximuojant retomis (su $2n$ perėjimais) fazinių skirstinių struktūromis labiau tikėtini parametrai randami greičiau. Aproximuojant tankia struktūra gali būti surasti ne ką prasteni parametru įverčiai, tačiau tam gali prireikti daugiau skaičiavimo resursų bei paieška gali būti per anksti nutraukta dėl lėto konvergavimo. Aproximuojant reta struktūra konvergavimo greitis priklauso nuo struktūros pasirinkimo. Tinkamai parinktos struktūros atveju konvergavimo greitis spartesnis (lyginant su tankia struktūra).
4. Pritaikius papildomus skaičiavimus nuoseklus maksimalaus tikėtinumo metodo algoritmas Markovo atvykimų proceso (MAP) parametru paieškai gali būti išlygiagretintas. Buvo nagrinėjama ER-CHMM procesų klasė. Panaudojant Baum-Welch algoritmo principus suformuluotas lygiagretus algoritmas P-2, kuris naudoja kur kas mažiau atminties nei nuoseklus algoritmas. Taip pat, pateikta kita lygiagretaus algoritmo versija P-3-D, kurios sudėtingumas didėja lėčiausiai, didinant Erlango šakų skaičių.
5. Suformuluotos maksimalaus tikėtinumo metodo algoritmas nutrūkstančio Markovo atvykimų proceso (TMAP) parametru paieškai.
6. MAP ir TMAP parametru paieškos algoritmai realizuoti C++ programavimo kalba. Lygiagretūs skaičiavimai atlikti panaudojant vaizdo procesorių (GPU) panaudojant CUDA biblioteką. Tyrimai parodė, jog MAP proceso parametru lygiagreti paieška kur kas greitesnė, nei analogiška TMAP proceso parametru paieška. Taip yra dėl skaičiavimo architektūros specifikos ir to, jog TMAP proceso atveju bet kuriuo laiko momentu skaičiavimo gijos atlieka skirtingas operacijas.

LITERATŪROS SARAŠAS

1. NEUTS, M. F. *Matrix Geometric Solutions in Stochastic Models, an Algorithmic Approach*. The Johns Hopkins University Press, 1981.
2. BLADT, M. and B. F. NIELSEN. Moment Distributions of Phase Type. *Stochastic Models*. Taylor & Francis Group, LLC, 2011, vol. 27, no. 4, pp. 651–663. doi:10.1080/15326349.2011.614192.
3. HORVÁTH, G. and M. TELEK. A Canonical Representation of Order 3 Phase Type Distributions. In: *Formal Methods and Stochastic Models for Performance Evaluation: Fourth European Performance Engineering Workshop, EPEW 2007, Berlin, Germany, September 27-28, 2007. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 48–62. ISBN 978-3-540-75211-0. doi:10.1007/978-3-540-75211-0_5.
4. HORVÁTH, G. and M. TELEK. On the Canonical Representation of Phase Type Distributions. *Performance Evaluation*. 2009.
5. FANG, Y. Hyper-Erlang Distribution Model and its Application in Wireless Mobile Networks. *Wireless Networks*. Springer, 2001, vol. 7, no. 3, pp. 211–219. doi:10.1023/a:1016617904269.
6. DEMPSTER, A. P., N. M. LAIRD and D. B. RUBIN. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*. 1977.
7. ALBERT, A. Estimating the infinitesimal generator of a continuous time, finite state markov process. *Ann. Math. Statist.* The Institute of Mathematical Statistics, 1962, vol. 33, no. 2, pp. 727–753. doi: 10.1214/aoms/1177704594.
8. ASMUSSEN, S., O. NERMAN and M. OLSSON. Fitting Phase-Type Distributions via the EM Algorithm. *Scandinavian Journal of Statistics*. Blackwell Publishing on behalf of Board of the Foundation of the Scandinavian Journal of Statistics, 1996, vol. 23, no. 4, pp. 419–441. ISSN 03036898. doi:10.2307/4616418.
9. OKAMURA, H., T. DOHI and K. S. TRIVEDI. A refined EM algorithm for PH distributions. *Perform. Eval.* Elsevier, 2011, vol. 68, no. 10, pp. 353–381. ISSN 0166-5316. doi:10.1016/j.peva.2011.04.001.

10. FANG, Y. and I. CHLAMTAC. Teletraffic analysis and mobility modeling of PCS networks. *IEEE Transactions on Communications*. IEEE, 1999, vol. 47, no. 7, pp. 1062–1072. doi:10.1109/26.774856.
11. NEUTS, M. F. A Versatile Markovian Point Process. *Journal of Applied Probability*. Applied Probability Trust, 1979, vol. 16, no. 4, pp. 764–779. doi:10.2307/3213143.
12. OKAMURA, H. and T. DOHI. Faster Maximum Likelihood Estimation Algorithms for Markovian Arrival Processes. In: *Sixth International Conference on the Quantitative Evaluation of Systems*. Budapest, Hungary: IEEE, 2009, pp. 73–82. ISBN 978-0-7695-3808-2. doi: 10.1109/QEST.2009.28.
13. Ó CINNÉIDE, C. A. On non-uniqueness of representations of phase-type distributions. *Communications in Statistics Stochastic Models*. 1989.
14. ANDERSEN, A. T., M. F. NEUTS and B. F. NIELSEN. On the Time Reversal of Markovian Arrival Processes. *Stochastic Models*. Taylor & Francis Group, LLC, 2004, vol. 20, no. 2, pp. 237–260. doi:10.1081/STM-120034131.
15. CUMANI, A. . *Microelectronics Reliability*. 1982.
16. COX, D. R. A use of complex probabilities in the theory of stochastic processes. *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press, 1955, vol. 51, no. 2, pp. 313–319. doi:10.1017/s0305004100030231.
17. OSOGAMI, T. and M. HARCHOL-BALTER. Necessary and Sufficient Conditions for Representing General Distributions by Coxians. In: *Computer Performance Evaluation. Modelling Techniques and Tools: 13th International Conference, TOOLS 2003, Urbana, IL, USA, September 2-5, 2003. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 182–199. ISBN 978-3-540-45232-4. doi: 10.1007/978-3-540-45232-4_12.
18. MÉSZÁROS, A. and M. TELEK. Canonical Form of Order-2 Non-stationary Markov Arrival Processes. In: *Computer Performance Engineering: 12th European Workshop, EPEW 2015, Madrid, Spain, Au-*

gust 31 - September 1, 2015, Proceedings. Cham, 2015, pp. 163–176. ISBN 978-3-319-23267-6. doi:10.1007/978-3-319-23267-6_11.

19. BOBBIO, A. and M. TELEK. A benchmark for PH estimation algorithms: results for acyclic-PH. *Stochastic Models*. 1994.
20. TURIN, W. Unidirectional and parallel Baum-Welch algorithms. *IEEE Trans. Speech and Audio Processing*. IEEE Signal Processing Society, 1998, vol. 6, no. 6, pp. 516–523. ISSN 1063-6676. doi:10.1109/89.725318.
21. ANDERSON, T. W. and L. A. GOODMAN. Statistical Inference about Markov Chains. *The Annals of Mathematical Statistics*. 1957, vol. 28, no. 1. doi:10.1214/aoms/1177707039.

MOKSLINIŲ PUBLIKACIJŲ SĄRAŠAS

Straipsniai Mokslinės informacijos instituto (ISI) duomenų bazėse referuojamuose leidiniuose (pagrindinių ISI žurnalų sąrašas)

1. BRAŽĖNAS, Mindaugas; HORVÁTH, Gábor; TELEK, Miklós. Parallel algorithms for fitting Markov arrival processes // Performance evaluation. Amsterdam : Elsevier. ISSN 0166-5316. eISSN 1872-745X. 2018, vol. 123-124, p. 50–67. DOI: 10.1016/j.peva.2018.05.001.

2. BRAŽĖNAS, Mindaugas; VALAKEVIČIUS, Eimutis. On structured initial solution generation for phase-type fitting with EM method // Information technology and control = Informacinės technologijos ir valdymas. Kaunas : Technologija. ISSN 1392-124X. eISSN 2335-884X. 2018, vol. 47, iss. 2, p. 197–208. DOI: 10.5755/j01.itc.47.2.18169.

Straipsniai Mokslinės informacijos instituto (ISI) duomenų bazėse referuojamuose konferencijų leidiniuose (ISI Proceedings)

3. BRAŽĖNAS, Mindaugas; HORVÁTH, Gábor; TELEK, Miklós. Efficient implementations of the EM-algorithm for transient Markovian arrival processes // Analytical and stochastic modelling techniques and applications : proceedings of the 23rd international conference, ASMTA 2016, Cardiff, UK, August 24-26, 2016 / editors: Sabine Wittevrongel, Tuan Phung-Duc. Cham : Springer, 2016. ISBN 9783319439037. eISBN 9783319439044. p. 107–122. DOI: 10.1007/978-3-319-43904-4_8.

4. BRAŽĖNAS, Mindaugas; VALAKEVIČIUS, Eimutis. Software reliability Markovian model based on phase-type distribution. In: ICSEA 2014: the Ninth International Conference on Software Engineering Advances, October 12-16, 2014, Nice, France. Nice: IARIA, 2014, p. 591–597. ISBN 9781612083674.

TRUMPA INFORMACIJA APIE DISERTACIJOS AUTRIŲ

Gimė: 1989 m. kovo 9 d. Utenos rajone.

Išsilavinimas:

2008–2012 m. – Kauno technologijos universiteto Matematikos ir gamtos mokslų fakultetas, matematikos bakalauro laipsnis.

2012–2014 m. – Kauno technologijos universiteto Matematikos ir gamtos mokslų fakultetas, matematikos magistro laipsnis.

2014–2018 m. – Kauno technologijos universiteto Matematikos ir gamtos mokslų fakultetas, informatikos (09P) doktorantūros studijos.

Mokslinių interesų sritys:

Markovo procesų parametrų paieška, Markovo modelių konstravimas, algoritmų išlygiagretinimas.

El. paštas:

mindaugas.brazenas@ktu.lt

RESUME

Continuous time finite state space Markovian models have been used to model stochastic processes for decades. This type of modelling is attractive due to its analytical tractability. The Markovian process progression depends only on the current system state (i.e. does not depend on the past system states). This property is known as the memoryless property, which in our case is inherited from the exponential distribution properties. The remaining holding time of exponential distribution does not depend on the elapsed time and has the same exponential distribution. More formally, such distribution is non-aging nor aging. However, the real life phenomena exhibit behaviours that are governed by a wide range of distributions. The accuracy of modelling non-Markovian processes depends on the accuracy of distribution approximation by exponential distribution mixtures and convolutions. Such distributions are said to be of Phase-type (PH).

Any distribution, of a positive random variable, can be approximated by a Phase-type distribution. In scientific literature, a lot of effort is being invested in solving PH representational and parameter searching problems. Canonical forms are significant for effective parameter fitting and application. However, these forms are known only for low order PH(n) distributions, i.e. $n = 2, 3$. For higher orders the analytical analysis of general PH distribution structures is complicated. Instead, properties of various PH sub-classes are investigated and used for parameter fitting. Despite the fact that these sub-classes are not optimal and might require more phases to approximate a given distribution, it has been found that these distribution sub-classes are more applicable for practical use. One of the most limiting Markovian modelling problems is a rapid state space increase to the point where it becomes impractical. This problem can be tackled by using more dense, preferably canonical, structures.

Phase-type distributions are used to model independent inter-arrival times; however, in practice certain processes show some kind of dependency. In general, dependency structure can be very complex and the model ability to capture it is essential for obtaining realistic statistical properties. For this purpose, the extension of PH distributions, the Markov Arrival Processes (MAPs) are used. Due to even more complex analytical structural analysis, canonical forms are known only for MAP(n) of order $n = 2$. The

parameter fitting problem is more complex than PH fitting, since autocorrelation must be captured.

In the scientific literature there are not many results of general PH distribution structures, because of the complexity of analytical analysis. We investigate PH(4) structures by generating all possible structures and eliminating the redundant ones based on current results in the field. The obtained generated structure sub-classes are investigated empirically by approximating various benchmark distributions by PH distributions of these structures.

A more pressing issue of MAPs is the fitting effectiveness. More specifically, in order to capture the autocorrelation structure, long data traces must be used. There are no documented parallel algorithms for parallel MAP fitting. In order to fill this gap we have implemented various parallel algorithms, including their modifications, and analyzed their execution properties. Some ideas for parallelization have been taken from the analogous CHMM (Continuous time Hidden Markov Model) fitting procedures. Also, we have derived an algorithm for transient Markov arrival process (TMAP) fitting by applying the Expectation-maximization method. Finally, for comparison purposes, we have implemented a parallel MAP and TMAP parameter fitting algorithm for GPU computing architecture.

Research Object

Phase-type distributions, Markov arrival processes and transient Markov arrival processes parameter fitting using the Expectation-maximization method.

Research Aim

To investigate general Phase-type distribution fitting with a sufficient number of transitions, and to develop parallel algorithms for Markov arrival processes and transient Markov arrival processes fitting.

Research Tasks

1. To develop an algorithm for Phase-type structure generation.
2. To research Phase-type fitting of various structure sub-classes.
3. To develop parallel algorithms for Markov arrival process fitting and compare their execution properties.

4. To develop an Expectation-maximization algorithm and its parallel version for transient Markov arrival process fitting.
5. To implement parallel algorithms for Markov arrival processes and transient Markov arrival processes fitting for execution on a GPU device using the CUDA library. To solve numerical issues, which arise when large data sets are fitted.

Methods and software

1. The descriptive statistics have been used to analyse research results.
2. Computations have been performed by the programs written in the C++ programming language. Several libraries have been used, i.e. ‘Eigen’ for matrix computations and ‘Boosts’ for statistical distributions.
3. The research programs have been built by the ‘CMake’ tool in Linux environment. For programs to be executed on a GPU device the CUDA library has been used.

Scientific Novelty and Practical Relevance

Canonical forms for $\text{PH}(n)$ distributions are known only for orders of $n = 2, 3$. By applying a Laplace–Stieltjes transformation to a PH distribution density function, it can be shown that $\text{PH}(n)$ distribution can be specified by $p = 2n - 1$ independent parameters. However, for practical applicability the PH representation of the matrix form is required. It is convenient to characterize the structure of the matrix form representation by a number of transitions m (the maximum number of independent parameters is $p = m - 1$). The full PH distribution space can be covered by matrix form representations of structures with $m = 2n + \Delta m_n$ transitions, where $0 \leq \Delta m_n \leq \Delta M_n$. In general the maximum number of additional transitions ΔM_n is not known, except for $\Delta M_2 = 0$ and $\Delta M_3 = 1$. We expect that structures with $m = 2n$ transitions (i.e. $p = 2n - 1$ parameters) are sufficient for practical fitting purposes. We formulate a hypothesis that a PH distribution, which has a matrix form representation with additional transitions (i.e. $0 < \Delta m_n \leq \Delta M_n$), can be closely approximated by a similar PH distribution without additional transitions (i.e. $\Delta m_n = 0$). An equivalent hypothesis has not been formulated in the scientific literature. We validate it empirically, for case $n = 4$. To achieve that we have developed a Phase-type distribution matrix form representation structure generation algorithm.

The algorithm outputs a set of structures that are used to approximate nine distributions, presented in the related scientific literature. Based on the fitting results, via EM method, the hypothesis is validated.

In the scientific literature the usage of certain Phase-type distribution matrix form representation structure classes for more effective fitting is being investigated. However, one specific structure class can not cover the whole parameter space. In order to search for parameters in an optimal manner, canonical forms must be used. The canonical forms, by an analytical approach, are discovered by Phase-type distributions of order $n \leq 3$. We have proposed an alternative combinatorial approach to generate a structure set. It cannot be guaranteed, that the obtained structure set is minimal, however, it can be used for further investigations in search of canonical forms.

However, our Phase-type structure generation algorithm is not effective and is hardly applicable for higher orders ($n > 4$), due to a large number of initial structures. Thus, we have compared Phase-type fitting using the generated structures ($n = 4$) versus the randomly generated sparse ($m = 2n$) structures. This comparison in scientific literature could not be done before, because of the absence of an analogous structure generation algorithm.

In the scientific literature there are no documented investigations of fitting with general sparse (with $m = 2n$ transitions) structures versus the full structure. We have carried out this investigation and obtained insights which are beneficial from a practical point of view. The full structure is flexible, but due to a redundant number of parameters slow convergence can be reached, which prevents one from reaching the most likely parameters. On the other hand, by fitting a set of sparse structures, there is a chance that one or more structures will converge faster and will reach more likely parameter estimates with a smaller number of iterations. Moreover, fitting with a set of sparse structures can be parallelized. For further research one might raise the question of how to determine a suitable structure given trace data.

Markov arrival processes (MAPs) structure analysis is a complex problem. We attempted to solve a more practical problem for faster fitting. The forward-backward algorithm of expectation maximization (EM) method is serial by nature, since likelihood vectors are evaluated recursively. Paral-

lelism can be achieved by increasing computational complexity. However, there are no documented attempts to parallelize MAP fitting in scientific literature. When the ER-CHMM structure is used the Baum-Welch algorithm (for CHMM fitting) can be adapted. However, there are various options to formulate the algorithm. In addition, we suggest our own algorithm, too. All algorithms have been implemented and their execution properties compared.

The transient Markov arrival process (TMAP) differs from Markov arrival process in a sense that it models finite sequences of inter-arrivals (further – runs), while MAPs generate an infinite sequence of inter-arrivals. EM method application for TMAP fitting is not present in the scientific literature. Thus, we have developed an algorithm to search for the maximum likelihood parameter estimates of such a process. It can be observed that part of computations for each run can be done independently. Therefore, TMAP fitting can be directly parallelized. We have implemented and compared serial and parallel versions of the algorithm.

The transient Markov arrival processes (TMAPs) fitting algorithm performs a number of independent computations for each run (the finite sequences of inter-arrivals) and the results are merged. Therefore, TMAP fitting can be easily parallelized; however, each thread is likely to perform different computational operations at any given moment, which is a limiting factor in effective parallel hardware usage.

Recently, with advances in the GPU (Graphics Processing Unit) hardware development and applicability for general computing, it is tempting to utilize it for faster parameter fitting. However, there are no documented attempts to utilize a GPU for faster MAP and TMAP process fitting in the scientific literature. Therefore, we have implemented the developed algorithms for execution in the GPU, which is required to find out how certain numerical issues should be solved.

The practical significance of the research results is as follows: In situations where general Phase-type distributions with a large number of phases must be fitted, sparse structures with $m = 2n$ transitions can be used to obtain more likely parameter estimates faster. Especially, in this case, a set of sparse structures is fitted in parallel. If a certain real life phenomena generates a finite sequence of events at random time instances, the collected

inter-arrival data can be used to fit a TMAP process. If a non-terminating sequence is observed, a MAP process can be fitted. Using our developed algorithms this fitting can be done faster, which is of significant importance given that the inter-arrival data is massive.

Approbation of the Research Results

The thesis work has been presented in 2 scientific papers (ISI) and 2 international conferences and several national conferences. PH(n) structure generation algorithm along with some fitting research has been presented in the ISI paper ‘On Structured Initial Solution Generation for Phase-Type Fitting with EM Method’. The presented work has been extended by adding more thorough research to find out how different sub-classes of generated structures perform fitting. The MAP fitting algorithms and results on fitting performance have been presented in the ISI paper ‘Parallel algorithms for fitting Markov arrival processes’. The derived EM algorithm for TMAP fitting with research of fitting performance has been presented at the ASMTA16 conference, and a conference proceedings paper with the title ‘Efficient implementations of the EM-algorithm for transient Markov arrival processes’ has been published. Finally, a simple finite queue model has been published in the IARA2014 conference proceedings with title ‘Software reliability Markovian model based on phase type distributions’. We have revisited the presented works and used them in the thesis as a practical application context for PH fitting.

The Structure and Volume of the Dissertation

This doctoral dissertation consists of an introduction, 4 major sections, a conclusion, a list of references, and a list of the author’s publications. The total length of the dissertation is 148 pages. The thesis features 54 figures, 44 tables and a list of 94 cited sources (10 sources have been published from 2014 to 2018).

CONCLUSIONS

1. A Phase-type distribution matrix form representation structure algorithm has been developed. It is not guaranteed that the generated structure sets are minimal and that the algorithm is not practical for higher order ($n > 4$) structure generation. However, it managed to generate structures of order $n = 4$ for the following research.
2. The hypothesis that $2n$ transitions are sufficient for Phase-type fitting is validated empirically for the case $n = 4$, using structure sets generated with our algorithm.
3. PH(n) fitting of sparse representations with $2n$ transitions finds more likely parameter estimates earlier. However, the full structure is capable of approaching good parameter estimates, but it requires more computational effort, and can be terminated too early due to slow convergence. The convergence of sparse structures depends on structure choice. In case of suitable structure, convergence is much faster (compared to full structure). In order to use the advantage of sparse structure fast convergence, a methodology for suitable structure should be developed.
4. With additional computational complexity the EM forward-backward algorithm for MAP fitting can be parallelized. The implementations based on Baum-Welch algorithm is memory efficient, while an algorithm based on parallel likelihood vector computation is faster.
5. An algorithm based on Expectation Maximization method for Transient Markov Arrival Process (TMAP) fitting has been formulated and tested.
6. MAP fitting performed 300-500 times faster on our GPU, while a simple TMAP parallelization implementation for GPU gave only a 10-20 times speed increase. The MAP parallelization is more efficient for execution on a GPU device (compared to TMAP fitting implementation) because of its uniform nature of computations.

UDK 519.217.2 (043.3)

SL344. 2019-05-02, 5,5 leidyb. apsk. I. Tiražas 50 egz.

Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas