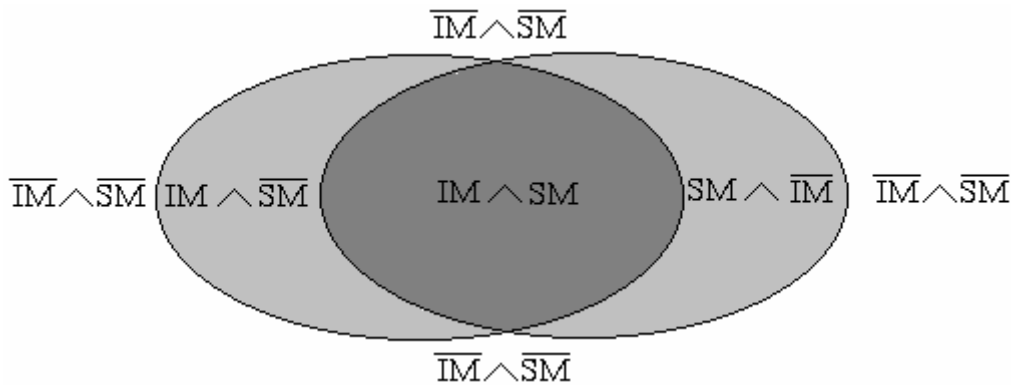


1.14. STATISTINIS MODELIAVIMAS

1.14.1. Statistinis ir imitacinis modeliavimas

Stochastinių sistemų ir procesų imitaciniame modeliavime labai plačiai naudojami statistinio modeliavimo metodai, pavyzdžiui, taip vadinamasis Monte-Karlo metodas. Galima sakyti, jog **stochastinių sistemų ir procesų imitacinis modeliavimas tiesiog neįmanomas be statistinio modeliavimo.**

Dėl tokio artimo ryšio tarp imitacinio ir statistinio modeliavimo sąvokų kartais atsiranda ir kolizijų, kada šios sąvokos nebeskiriamos vieną nuo kitos. Tačiau šios sąvokos, nors ir labai artimos, nėra pilnai sutampančios, nes, iš vienos pusės, galimas **sudėtingų determinuotų sistemų imitacinis modeliavimas**, nereikalaujantis statistinio modeliavimo metodų panaudojimo (297 pav. sritis $\overline{IM} \wedge \overline{SM}$, kuriai priklauso, pavyzdžiui, medžiagų sandėlio, aprašyto 1.4. ir 1.10 skyriuose, funkcionavimo imitacinis modeliavimas), iš kitos pusės – galimas **statistinio modeliavimo metodų panaudojimas, apskaičiuojant, pavyzdžiui, skaičių π , kartotinius integralus ir pan.**, vargu ar galima pavadinti imitavimu (297 pav. sritis $SM \wedge \overline{IM}$).



297 pav.

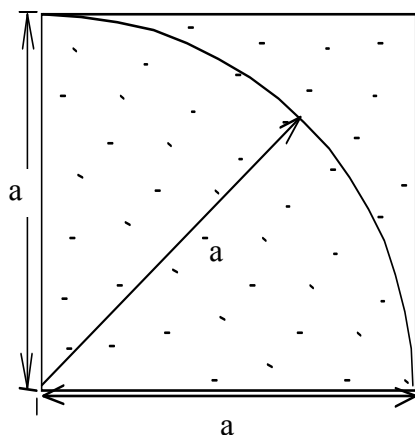
Tiktai srityje $IM \wedge SM$, imituojant stochastinius procesus ir sudėtingų nepilnai determinuotų sistemų funkcionavimą, IM ir SM sąvokos sutampa.

1.14.2. Statistinio modeliavimo esmė ir pavyzdžiai

Statistinio modeliavimo ir jo taikymų esmę gali paašškinti toks natūrinis eksperimentas. Paimkime lapą kartono, nubrėžkime jame kvadratą su kraštine a ir ketvirtį apskritimo su spinduliu $R=a$, kaip parodyta 298 pav.

Išneškime šį lapą trumpam laikui į lauką, dulkiant smulkiam lietui ar sningant smulkiam sniegui. Palaukime, kol kvadratas ir skritulio dalis pasidengs mažyčiais drėgmės taškučiais ir suskaičiuokime jų skaičius n_1 ir n , kur n_1 – taškučių skaičius skritulio ketvirtyje, n – taškučių skaičius visame kvadrato plote. Kadangi skaičiai n_1 ir n , kai drėgmės taškučiai yra atsitiktinai ir vidutiniškai beveik tolygiai išsibarstę kartono paviršiuje, su nedidele paklaida turi būti proporcingi skritulio ketvirčio ir

kvadrato plotams, tai gauname apytikrę lygybę $\frac{n_1}{n} \approx \frac{\frac{\pi R^2}{4}}{a^2} = \frac{\pi}{4}$, iš kurios išplaukia, jog



298 pav.

$$\pi \approx 4 \frac{n_1}{n}.$$

Tokiu būdu mūsų eksperimentas leidžia rasti apytikrę skaičiaus π reikšmę. Šiame pavyzdėlyje panaudotas metodas, vadinamas **Monte-Karlo metodu**, remiasi tuo, jog taškučių atsitiktinio tolygaus pasiskirstymo tankio vidurkiai kvadrato ir skritulyje turi būti apytikriai vienodi ir tų vidurkių lygybė tuo tikslesnė, kuo didesni kvadrato ir skritulio plotai ir kuo mažesni patys taškučiai, tiksliau, kuo daugiau tų taškučių tuose plotuose (žiūr. Čebyšovo nelygybę ir didžiųjų skaičių dėsnį [164, 165]).

Tuo pačiu statistikos dėsniu remiasi ir plačiai žinomi Biufono eksperimentai [164], leidžiantys, mėtant adatą, apskaičiuoti apytikrę π reikšmę. Tai irgi **fizinio statistinio modeliavimo**, sprendžiant matematinius uždavinius, pavyzdys. Fizinį modeliavimą* (ne tik statistinį, bet ir determinuotą) galima taikyti ne tik matematinių, bet ir techninių, biologinių ir pan. problemų sprendimui.

Vietoj fizinio modeliavimo galima taikyti taip pat ir **matematinį analoginį modeliavimą***, naudojantis jau mums žinomu izomorfizmo dėsniu ir analoginėmis schemomis bei analoginėmis skaičiavimo mašinomis, kurios gali būti ir specialiomis ir universaliosiomis.

Tačiau šiame skyrelyje mus domins ne fizinis ir ne matematinis analoginis, o **matematinis skaitmeninis modeliavimas**, t. y. modeliavimas, kuris vykdomas skaitmeninėmis diskretaus veikimo mašinomis – kompiuteriais.

Skaitmeniniam statistiniam modeliavimui, kurį šiame skyrelyje toliau vadinsime sutrumpintai tiesiog **statistiniu modeliavimu**, reikalingos atsitiktinės skaičių sekos arba programos, sugebančios generuoti tokias sekas.

Atsitiktinėmis skaičių sekomis vadinsime skaičių sekas, kuriose skaičiai yra išsidėstę atsitiktinai, nes kokio nors skaičiaus atsiradimas kokioje nors tos sekos vietoje yra **atsitiktinis įvykis**. Skaičius, sudarančius tokias sekas, vadinsime **atsitiktiniais skaičiais**. Kitaip tariant, atsitiktiniais skaičiais vadinsime skaičius, kurių konkrečios reikšmės yra gaunamos atsitiktinai. Atsitiktinį procesą, kurio metu gaunamos **atsitiktinio dydžio** konkrečių reikšmių atsitiktinės sekos, vadinsime **atsitiktinių skaičių generavimu**, nors tiksliau tai turėtų vadintis **atsitiktinės skaičių sekos generavimu**.

Jei mes turime atsitiktinę skaičių seką, kurioje skaičiai pasiskirstę pagal kokį nors žinomą dėsnį, tai iš jos po tam tikrų matematinių pertvarkymų galime gauti atsitiktinę skaičių seką, pasiskirsčiusią pagal kitą, šiuo laiku mums reikalingą, pasiskirstymo dėsnį. Dažniausiai tos pirminės atsitiktinės skaičių sekos, iš kurios kompiuteris gamina visas kitas visų kitų mums reikalingų pasiskirstymų sekas, vaidmenį atlieka atsitiktinė vidutiniškai tolygiai pasiskirsčiusių atkarpoje $[0; 1)$ skaičių seka.

Aprašysiu vieną natūrinį eksperimentą, kurį galima būtų panaudoti tokios atsitiktinės skaičių sekos sudarymui. Pažymėsiu, jog šio eksperimento paskirtis – ne praktinėms reikmėms pritaikyta atsitiktinių skaičių sekų „gamyba“, o atsitiktinių skaičių generavimo proceso esmės paaiškinimas ir iliustravimas.

* Apie fizinį ir matematinį analoginį modeliavimą galima ir naudinga plačiau pasiskaityti knygoje [166, 167, 168, 169, 170].

1.14.3. Atsitiktinės skaičių, vidutiniškai tolygiai pasiskirsčiusių intervale $[0;1)$, sekos generavimo pavyzdys

Tarkime, mums reikia generuoti atsitiktinių vidutiniškai tolygiai pasiskirsčiusių sveikų teigiamų vienaženklų skaičių 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 atsitiktinę seką.

Tarkime, mes turime pakankamai storą ir pakankamai naują knygą, kurios puslapiai sunumeruoti ir kuriai galioja teiginys, jog tikimybė atsitiktinai atversti tą knygą bet kurioje vietoje, yra vienoda.

Tarkime, jog mes turime dar ir monetą, kuri yra pakankamai simetriška, t. y. tokia, kad tikimybė šiai monetai po pamėtėjimo nukristi herbu arba skaitmenimis į viršų yra vienoda ir lygi 0,5.

Įvykdysime seriją eksperimentų, kurių kiekvienas susidės iš dviejų etapų. Pirmajame etape mes atsitiktinai atversime knygą kurioje nors atsitiktinoje vietoje. Antrajame etape mes mestelėsime monetą ir nustatysime, kaip ji nukrito. Jei moneta nukrito herbu į viršų, žiūrėsime į kairiąją atverstos knygos puslapį ir, perskaitę to puslapio numerį, užrašysime lape to numerio paskutinį skaitmenį. Jei moneta nukris skaičiais į viršų, tai mes tą patį padarysime su dešiniuoju atverstos knygos puslapio numerio paskutiniuoju skaitmeniu.

Pakartoję šias procedūras pakankamai daug kartų, mes lape gausime atsitiktinę sveikų skaičių seką, kurioje skaičiai 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 bus pasiskirstę vidutiniškai tolygiai (žinoma, jei seka pakankamai ilga ir mūsų knyga bei moneta yra idealios, o taip pat idealiai atsitiktiniai yra knygos atvertimo ir monetos metimo procesai).

Jei modifikuosime mūsų eksperimentus, laikydami, jog dabar kiekvienas eksperimentas susideda iš dviejų elementarių eksperimentų, kurių pirmasis nustato dviženklės dešimtainės trupmenos dešimtųjų dalių skaitmenį, o antrasis tos trupmenos šimtųjų dalių skaitmenį, tai po serijos tokių eksperimentų mes gausime atsitiktinę seką, kurioje bus vidutiniškai tolygiai ir atsitiktinai pasiskirstę skaičiai 0,00; 0,01; 0,02; 0,03; 0,04; 0,05; 0,06; 0,07; 0,08; 0,09; 0,10; 0,11; ...; 0,99.

Jei kiekvieną eksperimentą sudarys $n=3$ elementarūs eksperimentai, tai gausime atsitiktinę seką, kurioje atsitiktinai ir vidutiniškai tolygiai bus pasiskirstę triženkliai skaičiai 0,000; 0,001; ...; 0,999.

Didinant n , **diskretus tolygus atsitiktinis n -ženklų dešimtainių trupmenų pasiskirstymas artės prie tolydaus tolygaus atsitiktinių skaičių pasiskirstymo atkarpoje $[0;1)$.**

Atsitiktinių skaičių sekų generavimui galima naudoti ne tik čia aprašytus, bet ir įvairius kitus atsitiktinius procesus, pavyzdžiui, radioaktyviųjų elementų skilimo procesus, termoelektroninės emisijos iš pastovia srove kaitinamo katodo fluktuacijas, įvairius atsitiktinius triukšmus ir pan. Šie procesai yra labiau patogūs praktiniam naudojimui negu knygos vartymas bei monetos mėtymas. Įvairiuose lošimuose dažnai naudojamos kauliuko mėtymu, rutuliukų, pažymėtų skaičiais, iškritimu ir pan.

1.14.4. Pseudoatsitiktiniai skaičiai ir jų generavimas

Fiziniai atsitiktinių skaičių generatoriai nėra plačiai vartojami statistiniame modeliavime, kai šis modeliavimas vykdomas kompiuteriuose, nes tokie generatoriai nėra patogūs: visų pirma reikia turėti pačius generatorius ir užtikrinti, jog jie veikia gerai, t. y., generuoja tikrai atsitiktinius procesus, tikrai atitinkančius tą pasiskirstymą, kuris mums reikalingas; antra, reikia turėti atsitiktinių procesų charakteristikų matavimo priemones ir užtikrinti, kad matavimo paklaidos nedeformuos generuojamo

atsitiktinių skaičių pasiskirstymo taip, jog jis taps kitu pasiskirstymu (t. y. pasiskirstymu, kuriam galioja jau kitas pasiskirstymo dėsnis, mums nepageidautinas ar net nežinomas); trečia, reikia turėti interfeisą tarp matavimo priemonių ir kompiuterio, o visa tokia technika be visa ko turi dar ir kainą bei reikalauja papildomų eksploataavimo išlaidų.

Žinoma, viskas būtų daug paprasčiau, jei, projektuojant ir gaminant kompiuterius, į jų vidinę struktūrą specialiai būtų įjungiami atsitiktinių procesų generatoriai, generuojantys atsitiktinius skaičius, tačiau dabar dauguma plačiai naudojamų kompiuterių tokių generatorių neturi.

Todėl kompiuteriniame statistiniame modeliavime vietoj fizinių atsitiktinių skaičių generatorių žymiai dažniau (praktiškai beveik visada) vartojami **matematiniai pseudoatsitiktinių skaičių generatoriai**.

Pseudoatsitiktiniais skaičiais vadinami tokie skaičiai, kurie nėra pilnai atsitiktiniai, nes eilinis pseudoatsitiktinių skaičių sekos skaičius gaunamas pagal tam tikrą algoritmą iš jau anksčiau gautų skaičių ir todėl yra tų skaičių funkcija. Tačiau tas algoritmas parenkamas taip, jog gauti pseudoatsitiktiniai skaičiai su tam tikru tikslumu ir tam tikrose ribose galėtų būti panaudoti kaip atsitiktiniai, nes tenkintų atsitiktinių skaičių pasiskirstymo kriterijus. Kitaip tariant, **pseudoatsitiktinių skaičių generatorius imituoja atsitiktinių skaičių generavimą** ir gamina skaičių sekas, panašias į atsitiktinių skaičių sekas. Šis faktas rodo dar į vieną ryšį tarp statistinio ir imitacinio modeliavimo: **kompiuterinis statistinis modeliavimas naudoja imitaciniu modeliavimu pseudoatsitiktinių skaičių gamybą**.

Tiems skaitytojams, kurie norėtų sužinoti apie pseudoatsitiktinių ir atsitiktinių skaičių generavimo algoritmus ir apie gautų skaičių sekų tikrinimo kriterijus bei metodus šiek tiek daugiau negu parašyta šioje mokymo priemonėje, rekomenduojau pasiskaityti apie tai knygoje [171, 172, 160, 163]. Pseudoatsitiktinių skaičių generavimo idėją skaičių sandaugos vidurio išskyrimo metodu iliustruosiu pavyzdžiu, paimtu iš knygos [161].

Tarkime, mums reikalinga tolygiai pasiskirsčiusių atkarpoje $[0;1)$ keturženklų pseudoatsitiktinių skaičių seka. Pradžioje imame du teigiamus keturženklus nelyginius skaičius – daugiklius, pavyzdžiui, 5167 ir 3729. Pirmąjį iš jų vartosime kaip pastovų visai generuojamai sekai daugiklį, o antrąjį – kaip „pradinį dauginamąjį“. Pseudoatsitiktinių skaičių generavimo algoritmą sudaro šie veiksmas.

1. Randame dauginamojo ir daugiklio sandaugą – aštuonženklį skaičių: $3729 \times 5167 = 19267743$

2. Išskiriame sandaugoje paskutinius (dešiniuosius) keturis skaitmenis kaip naują dauginamojo reikšmę: 7743.

3. Išskiriame keturis vidurinius sandaugos skaitmenis ir užrašome juos dešimtaine keturženkle trupmena kaip pirmąjį generuojamos pseudoatsitiktinių skaičių sekos skaičių: 0,2677.

4. Vykdomė su gautuoju pseudoatsitiktiniu skaičiumi veiksmus, reikalingus statistiniame modeliavime ir, jei reikia naujų pseudoatsitiktinių skaičių, grįžtame prie pirmojo generavimo algoritmo žingsnio ir kartojame dauginimo ir išskyrimo veiksmus (tik su nauja dauginamojo reikšme).

Aprašytąjį algoritmą ir jo veikimo rezultatus (pirmąjį dešimtuką pseudoatsitiktinių skaičių) vaizduoja 101 lentelė.

Šią lentelę galima tęsti, surandant, pavyzdžiui, vienuoliktąjį pseudoatsitiktinį skaičių, lygų 0,6606, dvyliktąjį, lygų 0,1363 ir t.t, tačiau ne iki begalybės, nes anksčiau ar vėliau skaičiai pradės kartotis.

101. lentelė

Dauginamasis	Sandauga	Pseudoatsitiktiniai skaičiai
3729	$3729 \times 5167 = 19\,26\,77\,43$	$\sqrt{0,2677}$
7743	$7743 \times 5167 = 40\,00\,80\,81$	$\sqrt{0,0080}$
8081	$8081 \times 5167 = 41\,75\,45\,27$	$\sqrt{0,7545}$
4527	$4527 \times 5167 = 23\,39\,10\,09$	0,3910
1009	$1009 \times 5167 = 5\,21\,35\,03$	0,2135
3503	$3503 \times 5167 = 18\,10\,00\,01$	0,1000
1	$1 \times 5167 = 51\,67$	0,0051
5167	$5167 \times 5167 = 26\,69\,78\,89$	0,6978
7889	$7889 \times 5167 = 40\,76\,24\,63$	0,7624
2463	$2463 \times 5167 = 12\,72\,63\,21$	0,7263

Kai dauginamieji ir daugiklis yra nelyginiai skaičiai, turintys baigtinį ženklų skaičių n , tai kintantysis dauginamasis, tęsiant generavimą, pasikartos ne vėliau negu po $0,5 \cdot 10^n$ žingsnių, nes egzistuoja tik $0,5 \cdot 10^n$ skirtingų dešimtainių nelyginių n -ženklų skaičių.

Todėl pseudoatsitiktinių skaičių x_i sekoje $x_1, x_2, \dots, x_i, \dots$ būtinai atsiras toks skaičius x_r , lygus tos sekos skaičiui x_j , kur $1 \leq j \leq r \leq 0,5 \cdot 10^n$, kad po to sektų skaičiai $x_{r+1}, x_{r+2}, \dots$, tenkinantys lygybes: $x_{r+1} = x_{j+1}, x_{r+2} = x_{j+2}, \dots, x_{r+p} = x_{j+p} = x_r = x_j, x_{r+p+1} = x_{j+p+1} = x_{r+1} = x_{j+1}, \dots, x_{r+p+m} = x_{j+m} = x_{j+m}, \dots, x_{r+kp+m} = x_{j+m}, \dots$, t. y. generuojamųjų pseudoatsitiktinių skaičių seka $x_j, x_{j+1}, x_{j+2}, \dots, x_{j+p-1}$ kartosis, pradedant nuo $x_j = x_r$, su periodu $p = r - j \leq 0,5 \cdot 10^n$.

Tikrumoje kartojimosi periodas bus trumpesnis negu $0,5 \cdot 10^n$. Pavyzdžiui, mūsų nagrinėjamu atveju negalėsime gauti dauginamųjų, kurių paskutinis skaitmuo būtų 5, todėl p negalės būti didesnis už $0,4 \cdot 10^n$. Jeigu mes būtume pasirinkę pradinę dauginamojo reikšmę taip, kad to dauginamojo paskutinis skaitmuo būtų 5, tai kartojimosi periodas būtų dar trumpesnis ir neviršytų 10^{n-1} , nes tada visi dauginamieji baigtusi skaitmeniu 5 ir būtų dalūs iš 5. Panašų periodo sutrumpėjimą gautume, jei būtume pasirinkę tokias pradines dauginamojo ir daugiklio reikšmes, kad abiejų skaičių paskutiniai skaitmenys būtų vienodi ir lygūs 1 arba 6.

Tai rodo, jog **periodo ilgis** p priklauso ne tik nuo ženklų skaičiaus n , bet ir nuo to, kokie buvo pasirinktieji pradinis dauginamasis ir daugiklis. Juos stengiamasi parinkti taip, kad ne tik periodo ilgis būtų kuo didesnis, bet kad ir sekos **pradinė aperiodinė atkarpa** $x_1, x_2, \dots, x_i, \dots, x_r$ būtų kuo ilgesnė, nes, kai modeliuojant reikia atlikti daug nepriklausomų atsitiktinių eksperimentų, imituojuant atsitiktinius įvykius, tai reikia, kad generuojama pseudoatsitiktinių nesikartojančių skaičių seka būtų pakankamai ilga. Į tai reikia atsižvelgti, programuojant pseudoatsitiktinių skaičių generatorių, t. y., pasirenkant n ir daugiklio bei pradinę dauginamojo reikšmes.

Tačiau pseudoatsitiktinių skaičių seka, imituojanti atsitiktinumą ir iš tikrųjų nebūdama tikrai atsitiktinė, turi ne vien tik neigiamų savybių. Ji gali būti panaudota tam, kad statistinį modeliavimą būtų galima pakartoti visiškai identiškomis sąlygomis, ko negalima padaryti, naudojantis tikrai atsitiktiniais skaičiais.

Čia aprašytasis pseudoatsitiktinių skaičių, tolygiai pasiskirsčiusių intervale $[0;1)$ generavimo metodas nėra vienintelis. Jis buvo čia paminėtas ir aprašytas todėl, kad yra naudojamas žinomos JAV firmos IBM sukurtoje masinio aptarnavimo sistemų funkcionavimo imitacinio modeliavimo sistemoje GPSS. Šis skaičių **sandaugos vidurio išskyrimo metodas** – tai patobulinta žymaus matematiko Neumano pasiūlyto **kvadrato vidurio metodo** modifikacija.

Yra nemažai kitų metodų ir jų pagrindu sukurtų generatorių. Su jais galima susipažinti specialioje literatūroje bei jau minėtose knygos [160, 171, 172].

Čia paminėsiu tik keletą tokių metodų.

Taip vadinamame **multiplikaciniame kongruentiškame pseudoatsitiktinių skaičių generatoriuje** pasirenkami teigiami sveiki skaičiai β_0 , k ir M , iš kurių skaičius M – pats didžiausias. Naudojantis tais skaičiais, generuojami skaičiai: $\beta_1=[k\beta_0](\text{mod}M)$, $\beta_2=[k\beta_1](\text{mod}M)$, ..., $\beta_i=[k\beta_{i-1}](\text{mod}M)$, $\beta_{i+1}=[k\beta_i](\text{mod}M)$, ...

Iš sekos β_i , kur $i=1, 2, 3, \dots$, naudojantis formule $\gamma_i=\beta_i M^{-1}$, gaunama teigiamų pseudoatsitiktinių skaičių γ_i seka, kurioje visi skaičiai yra apytikriai tolygiai pasiskirstę intervale $[0; 1)$ ir pakankamai gerai tenkina skaičių atsitiktinumą kriterijus.

Esant pakankamai dideliui kompiuteryje apdorojamų skaičių ženklių skaičiui n ir atitinkamai parinkus skaičių k ir M reikšmes, galima gauti pakankamai ilgą pseudoatsitiktinių skaičių $\gamma_i = \frac{[\beta_{i-1}k](\text{mod}M)}{M}$ pradinę aperiodinę atkarpą $\gamma_1, \gamma_2, \gamma_3, \dots$,

$\gamma_j, \dots, \gamma_{r-1}$ bei sekos periodą $p=r-j$. Pavyzdžiui, kai $\beta_0=1$, $k=23$ ir $M=10^8+1$, tai $p=5882352$, o kai $\beta_0=1$, $k=5^{13}$ ir $M=2^{36}$ tai $p \approx 2^{34}$. Dar ilgesnis periodas p gaunamas, kai $\beta_0=1$, $k=5^{17}$, $M=2^{42}$. Tuomet $p \approx 10^{12}$.

Periodą p galima dar padidinti, pereinant prie taip vadinamų **mišriųjų kongruentiškųjų generatorių**, kurie skiriasi nuo grynai multiplikacinių tuo, kad apskaičiuojant β_i , naudojamos ne formule $\beta_i=[k\beta_{i-1}](\text{mod}M)$, o formule $\beta_i=[k\beta_{i-1}+C](\text{mod}M)$, kur C – pastovus teigiamas sveikas skaičius, mažesnis už M .

Kai $C=0$, tai iš mišraus generatoriaus gauname jau minėtąjį multiplikacinį kongruentiškąjį generatorių.

Tolygiai pasiskirsčiusių intervale $[0, 1)$ pseudoatsitiktinių skaičių generavimui knygoje [160] rekomenduojamas algoritmas, pavadintas **likinių (reziduų) algoritmu**. Jį aprašo formulės: 1) $\beta_0=2^{-m}$, 2) $\beta_{n+1}=\{5^{2p+1} \cdot \beta_n\}$, kur m – dvejetainių skilčių skaičius kompiuteryje, naudojant pseudoatsitiktinių skaičių generavimui; p – koks nors sveikas teigiamas skaičius, pavyzdžiui, $p=8$; figūriniai skliausteliai $\{ \}$ reiškia trupmeninę skaičiaus, esančio skliausteliuose, dalį.

Naudojantis šiuo metodu, kurį kartais vadina dar ir **multiplikatyvaus daviklio metodu**, vietoj $\beta_0=2^{-m}$ galima naudotis ir bet kuria kita dvejetainė trupmena, besibaigiančia dvejetainiais skaitmenimis 01, o vietoje rekurentinės formulės $\beta_{n+1}=\{5^{2p+1} \cdot \beta_n\}$ – naudotis formule $\beta_n=\{\beta_0 \cdot 5^{n(2p+1)}\}$.

Pseudoatsitiktinius skaičius, tolygiai pasiskirsčiusius intervale $[0; 1)$, galima generuoti ir naudojantis dvejetainio skaičiaus postūmiu kompiuterio registre bei pastumtojo skaičiaus sudėtimi su prieš tai buvusiu nepastumtojo skaičiaus reikšme. Tokiu būdu gautąjį pseudoatsitiktinį skaičių vėl stumdant ir sudedant, gaunamas antras pseudoatsitiktinis skaičius ir t. t. Šį metodą, vadinamą **chaotiško sumaišymo metodu**, gana paprasta realizuoti kompiuteryje. Kartais postūmio ir sudėties operacijos

papildomos dar ir daugybos operacija. Kai kompiuteryje vaizduojami skaičiai turi pakankamai didelį dvejetainių ženklų skaičių n , tai, atitinkamai parinkus pradinį skaičių ir postūmio dydį, galima gauti pakankamai ilgas nesikartojančių skaičių sekas, kuriose jie pasiskirstę beveik tolygiai ir tenkina atsitiktinumo kriterijus. Pavyzdžiui, kai $n=42$ ir postūmio dydis – 7 dvejetainiai ženklai į kairę, tai gaunamos pradinės aperiodinės atkarpos ilgis – apie 3×10^5 pseudoatsitiktinių skaičių, nors blogiau pasirinkus pradinį skaičių šios atkarpos ilgis gali būti ir žymiai mažesnis.

Pseudoatsitiktinių skaičių sekos periodo ilgį galima padidinti, **naudojantis ne vienu, o dviem generatoriais**: po tam tikro sugeneruotų vienu generatoriumi pseudoatsitiktinių skaičių skaičiaus keičiant pirmąjį generatorių antruoju, po to antrąjį – pirmuoju ir t. t.

Pavyzdžiui, naudojantis anksčiau aprašytu sandomos vidurio metodu, periodo p ilgį galima padidinti, laikas nuo laiko keičiant pastovų daugiklį. Jau minėtoje masinio aptarnavimo procesų imitacinio modeliavimo sistemoje GPSS naudojamos net aštuonios pastovaus daugiklio („branduolio“) reikšmės: 37584381, 1909996635, 1964463183, 1235671459, 1480745561, 442596621, 340029185 ir 2030226625. Jos gali būti keičiamos, pavyzdžiui, **pasirenkamos atsitiktinai (pseudoatsitiktinai)**, generuojant eilinių pseudoatsitiktinių skaičių.

Dar viename pseudoatsitiktinių skaičių generavimo metode, vadinamame **kvadratų metodu**, pasirenkami teigiami skaičiai a ir b taip, kad $a < b$ ir $a + b = 1$ (pavyzdžiui, $a=0,4$, $b=0,6$), bei pradinė kintamojo β_i reikšmė β_0 : $0 < \beta_0 < 1$. Po to generuojama skaičių β_i seka, naudojantis algoritmu:

$$\beta_i = \begin{cases} \beta_{i-1}^2, & \text{kai } \beta_{i-1}^2 \geq b, \\ \beta_{i-1}^2 + a, & \text{kai } \beta_{i-1}^2 < b. \end{cases}$$

Iš skaičių β_i , kur $i=1, 2, 3, \dots$, gaunami pseudoatsitiktiniai skaičiai $\gamma_i = \frac{\beta_i - b}{1 - b}$. Nesunku įsitikinti, jog skaičiai γ_i apytikriai tolygiai pasiskirstę atkarpoje

$[0;1)$ ir neblogai tenkina atsitiktinumo kriterijus. Gaunamų pseudoatsitiktinių skaičių sekos kokybė dar pagerėja, kai vietoje ką tik aprašyto algoritmo, pradedant

nuo $i=3$, naudojamas algoritmu: $\beta_3 = \frac{1}{3}(\beta_2 + \beta_1 + \beta_0)$, $\beta_4 = \begin{cases} \beta_3^2, & \text{kai } \beta_3^2 \geq b, \\ \beta_3^2 + a, & \text{kai } \beta_3^2 < b, \end{cases}$

$$\beta_5 = \frac{1}{3}(\beta_4 + \beta_3 + \beta_2), \beta_6 = \begin{cases} \beta_5^2, & \text{kai } \beta_5^2 \geq b, \\ \beta_5^2 + a, & \text{kai } \beta_5^2 < b \end{cases} \text{ ir t. t., } \gamma_i = \frac{\beta_i - b}{1 - b}.$$

Kvadratų metodo trūkumas – jo realizavimui reikia daugiau kompiuterio operacijų negu kitiems anksčiau aprašytiems metodams.

Statistiniame modeliavime gali būti naudojamos ir įvairiomis iš anksto sudarytomis atsitiktinių ar pseudoatsitiktinių skaičių lentelėmis, patalpinamomis į kompiuterio atmintį. **Lentelinis metodas** turi tą trūkumą, kad, sprendžiant sudėtingus statistinio modeliavimo uždavinius ir norint pasiekti pakankamą tikslumą, reikia turėti daug kompiuterinės atminties.

Pseudoatsitiktinių skaičių generatorių generuojamų skaičių sekų kokybę, kai imitaciniame modeliavime naudojamos nepatikrintu generatoriumi, tikslinga patikrinti dar prieš pradedant modeliavimą, nes, naudojantis nekokybiškais generatoriais, galima gauti rezultatus, turinčius **sisteminę paklaidą**, kurios nepavyks pašalinti, didinant statistinių eksperimentų su modeliu skaičių.

Norint patikrinti pseudoatsitiktinių skaičių pasiskirstymo intervale $[0;1)$ tolygumą, galima naudotis χ^2 , Kolmogorovo bei kitais kriterijais. Tikrinant

pseudoatsitiktinių skaičių pasiskirstymo „atsitiktinumą“ galima naudotis serijų bei įvairiais kitais koreliacinės analizės metodais (pavyzdžiui, apskaičiuojant koreliacijos koeficientus, pseudoatsitiktinių skaičių porų sandaugų sumas ir pan.).

Tokie generatorių kokybės patikrinimai užima nemažai laiko, tačiau sudarant kokybiškus pseudoatsitiktinių skaičių generatorius, jie visuomet daromi. Todėl imitaciniame modeliavime patartina naudotis jau patikrintais solidžių firmų sudarytais generatoriais, nors, žinoma, galima kurti ir „nuosavos konstrukcijos“ generatorius, po to patikrinant gautų pseudoatsitiktinių skaičių sekų kokybę.

1.14.5. Atsitiktinių įvykių ir diskrečių atsitiktinių skaičių, pasiskirsčiusių su duotomis tikimybėmis, generavimas

Turėdami tolygiai pasiskirsčiusių atkarpoje $[0;1)$ atsitiktinių ar pseudoatsitiktinių skaičių generatorius, mes galime modeliuoti diskrečius atsitiktinius įvykius A_i , kurių įvykimo tikimybės p_i yra žinomos.

Tarkime, mums statistinio modeliavimo metu reikia nustatyti, kurio iš atsitiktinių įvykių $A_1, A_2, \dots, A_i, \dots, A_n$, kur $\sum_{i=1}^n p_i = 1$ ir $\forall i \forall j i \neq j p(A_i \wedge A_j) = 0$ įvykimą dabartiniu imitavimo momentu reikia modeliuoti. Tuo tikslu atkarpą $[0;1)$ suskirstome į atkarpėles $[0;p_1), [p_1; p_1+p_2), [p_1+p_2; p_1+p_2+p_3), \dots, \left[\sum_{i=1}^{n-1} p_i; 1\right)$. Po to kreipiamės į tolygiai pasiskirsčiusių atkarpoje $[0;1)$ pseudoatsitiktinių skaičių generatorių, gauname eilinį atsitiktinį skaičių ξ ir žiūrime, į kurią iš atkarpėlių jis pateko. Jei skaičius ξ patenka į atkarpėlę $\sum_{i=1}^{j-1} p_i \leq \xi < \sum_{i=1}^j p_i$, tai reiškia, jog šiuo metu reikia imituoti įvykio A_j įvykimą. Nesunku pastebėti, jog taip elgiantis įvykių A_j modeliavimo dažniai bus apytikriai proporcingi atkarpėlių $\left[\sum_{i=1}^{j-1} p_i; \sum_{i=1}^j p_i\right)$ ilgiams, t. y., tikimybėms p_j .

Tokiu būdu mes iš tolygaus skirstinio atkarpoje $[0;1)$ galime gauti bet kokią diskretų vienmatį skirstinį su duotomis tikimybėmis.

Pailiustruosiu tai konkrečiu pavyzdžiu, imituodamas lošimą, kurio metu iš urnos su dviejų spalvų rutuliais atsitiktinai traukiami keturi rutuliai, įvertinamas ištrauktų tam tikros spalvos rutulių skaičius ir po to ištrauktieji rutuliai vėl grąžinami į urną, kad rutulių traukimą būtų galima pakartoti.

Tarkime, kad urnoje vieną penktadalį rutulių sudaro balti rutuliai, o likusius keturis penktadalius – juodi rutuliai.

Tarkime, jog lošėjas, norėdamas dalyvauti lošime, t. y. atsitiktinai traukti iš urnos keturis rutulius, perka bilietą, už kurį sumoka 4 litus.

Jei jis ištraukia visus keturis juodus rutulius, tai jis nieko neišlošia. Jei ištraukia vieną baltą rutulį ir tris juodus, tai gauna 1 litą. Jei ištraukia du baltus ir du juodus rutulius, tai 5 litus. Jei ištraukia tris baltus rutulius ir vieną juodą rutulį, tai išlošia 20 litų, o jei ištraukia visus keturis baltus rutulius, tai išlošia 1000 litų.

Pasinaudodami pseudoatsitiktinių skaičių generatoriumi, kurio darbo rezultatų lentelę mes sudarėme 1.14.4. skyrelyje (žiūr. 101. lentelę), prateisdami tą lentelę imituosime šį lošimą, t. y. generuosime skaičių, rodančių lošėjų išlošimus litais, atsitiktinę seką.

Šiuo atveju galimi penki atsitiktiniai įvykiai: 1) A_0 , kai ištraukiami keturi juodi rutuliai ir lošėjo išlošis $x=0$; 2) A_1 , kai ištraukiamas tik vienas baltas rutulys ir lošėjo išlošis $x=1$ Lt; 3) A_2 , kai ištraukiami du balti bei du juodi rutuliai ir lošėjo išlošis $x=5$ Lt; 4) A_3 , kai ištraukiami trys balti rutuliai ir lošėjo išlošis $x=20$ Lt; 5) A_4 , kai ištraukiami visi keturi balti rutuliai ir lošėjo išlošis $x=1000$ Lt.

Šių įvykių skirstinys – binominis, todėl įvykio A_i tikimybė $P\{A_i\} = C_4^i p^i (1-p)^{4-i} = C_4^i 0,2^i \cdot 0,8^{4-i}$.

Apskaičiuosime tas tikimybes:

- 1) $P\{A_0\} = P\{x=0\} = C_4^0 0,8^4 = 0,4096$;
- 2) $P\{A_1\} = P\{x=1\} = C_4^1 0,2 \cdot 0,8^3 = 0,4096$;
- 3) $P\{A_2\} = P\{x=5\} = C_4^2 0,2^2 \cdot 0,8^2 = 0,1536$;
- 4) $P\{A_3\} = P\{x=20\} = C_4^3 0,2^3 \cdot 0,8 = 0,0256$;
- 5) $P\{A_4\} = P\{x=1000\} = C_4^4 0,2^4 = 0,0016$.

Suskirstysime atkarpą $[0; 1)$ į intervalus: $[0,0000; 0,4096)$, $[0,4096; 0,8192)$, $[0,8192; 0,9728)$, $[0,9728; 0,9984)$, $[0,9984; 1,0000)$.

Imituosime 40 pirmųjų bandymų išlošti tą pageidaujamą 1000 litų. Tuo tikslu generuosime 40 pseudoatsitiktinių skaičių ξ_i , tolygiai pasiskirsčiusių atkarpoje $[0; 1)$, ir pagal jų patekimą į mūsų sudarytas atkarpeles spręsimė apie išlošio x_i dydį.

Pasinaudodami 101. lentelės sudarymo algoritmu ir pratęsdami tą lentelę, gauname:

$\xi_1=0,2677$; $\xi_2=0,0080$; $\xi_3=0,7545$; $\xi_4=0,3910$; $\xi_5=0,2135$; $\xi_6=0,1000$; $\xi_7=0,0051$; $\xi_8=0,6978$;
 $\xi_9=0,7624$; $\xi_{10}=0,7263$; $\xi_{11}=0,6606$; $\xi_{12}=0,1363$; $\xi_{13}=0,9086$; $\xi_{14}=0,5550$; $\xi_{15}=0,0468$; $\xi_{16}=0,3784$;
 $\xi_{17}=0,6559$; $\xi_{18}=0,9141$; $\xi_{19}=0,4998$; $\xi_{20}=0,0861$; $\xi_{21}=0,6685$; $\xi_{22}=0,1416$; $\xi_{23}=0,6857$; $\xi_{24}=0,5194$;
 $\xi_{25}=0,2803$; $\xi_{26}=0,5656$; $\xi_{27}=0,9403$; $\xi_{28}=0,8963$; $\xi_{29}=0,4952$; $\xi_{30}=0,1939$; $\xi_{31}=0,2598$; $\xi_{32}=0,6727$;
 $\xi_{33}=0,3074$; $\xi_{34}=0,3546$; $\xi_{35}=0,9800$; $\xi_{36}=0,2428$; $\xi_{37}=0,7207$; $\xi_{38}=0,0457$; $\xi_{39}=0,7670$; $\xi_{40}=0,6185$.

Kadangi $\xi_1 \in [0; 0,4096)$, tai $x_1=0$. Analogiškai gauname, jog ir $x_2=0$. Kadangi $\xi_3 \in [0,4096; 0,8192)$, tai $x_3=1$. Kadangi ξ_4, ξ_5, ξ_6 ir ξ_7 , vėl priklauso intervalui $[0; 0,4096)$, tai $x_4=x_5=x_6=x_7=0$. Kadangi ξ_8, ξ_9, ξ_{10} ir ξ_{11} priklauso intervalui $[0,4096; 0,8192)$, tai $x_8=x_9=x_{10}=x_{11}=1$. Analogiškai gauname, jog $x_{12}=0$, $x_{13}=5$, $x_{14}=1$, $x_{15}=0$, $x_{16}=0$, $x_{17}=1$, $x_{18}=5$, $x_{19}=1$, $x_{20}=0$, $x_{21}=1$, $x_{22}=0$, $x_{23}=1$, $x_{24}=1$, $x_{25}=0$, $x_{26}=1$, $x_{27}=5$, $x_{28}=5$, $x_{29}=1$, $x_{30}=1$, $x_{31}=0$, $x_{32}=1$, $x_{33}=0$, $x_{34}=0$, $x_{35}=20$, $x_{36}=0$, $x_{37}=1$, $x_{38}=0$, $x_{39}=1$, $x_{40}=1$.

Kaip matome, iš 40 lošėjų, sumokėjusių po 4 Lt už bilietą, nieko neišlošė 18 lošėjų, po 1 Lt išlošė 17 lošėjų, po 5 Lt – 4 lošėjai, 20 Lt – 1 lošejas. 1000 Lt neišlošė nė vienas lošejas, nes 40 lošimų – per mažas bandymų skaičius, kad būtų pakankama tikimybė įvykti tokiam įvykiui, kurio tikimybė yra $P\{x=1000\}=0,0016=\frac{1}{625}$ (t. y. vidutiniškai tik viename iš 625 bandymų tai galėjo atsitikti).

Kadangi šiuo atveju $P\{x=0\}=P\{x=1\}$, tai ir išlošimų po 1 Lt ir nieko neišlošimų skaičiai buvo apytikriai lygūs.

40 bandymų – per mažas skaičius, kad visos statistinės proporcijos būtų pakankamai artimos atitinkamų tikimybių santykiams. Pavyzdžiui, $\frac{P\{x=1\}}{P\{x=5\}} \approx 2,7$ žymiai skiriasi nuo santykio $\frac{17}{4}=4,25$.

Tačiau imitavimo rezultatams būdingi tie patys bruožai, kaip ir realiems lošimams – loterijoms: dauguma žmonių, nusipirkę bilietus nieko neišlošia arba išlošia nedideles sumas. Dideles sumas išlošia tik nedaugelis, tačiau viltis išlošti didelę sumą vilioja ir todėl žmonės, pirkdami loterijų bilietus ir beveik nieko neišlošdami, sukaupia dideles sumas, kurios atitenka nedidelei daliai laimingųjų ir duoda nemažą pelną loterijų bei lošimo namų savininkams.

Pavyzdžiui, šiame lošime išlošio matematinis vidurkis $\bar{x} = \sum_{i=1}^n x_i P\{x_i\} = 0 \cdot 0,4096 + 1 \cdot 0,4096 + 5 \cdot 0,1536 + 20 \cdot 0,0256 + 1000 \cdot 0,0016 = 0,4096 + 0,7680 + 0,5120 + 1,600 = 3,2896$.

Jis, kaip ir realiuose lošimuose, yra mažesnis už bilieto kainą, nes kitaip lošimo organizatoriai ne tik neturėtų pelno, bet ir nepadengtų lošimo organizavimo išlaidų.

Todėl dalyvavimas lošimuose, žiūrint į juos statistiškai, yra nuostolingas ir atskiriems lošiantiems asmenims, ir visuomenei. Atskiriems lošiantiems asmenims jis yra nuostolingas todėl, kad galimų išlošių statistinis vidurkis visada yra mažesnis už lošimo dalyvio galimų išlaidų statistinį vidurkį. Visuomenei lošimai nuostolingi todėl, kad didina turtinę nelygybę visuomenėje ir skatina nesveikas aistras, kurios kartais priveda ir prie tragiškų pasekmių. Tačiau lošimai kai kam atneša didelius pelnus ir todėl visuomenėse, kuriose žmonių veiklos variklis – didelių pelnų siekimas, jie toleruojami.

Norėčiau pažymėti dar vieną imitacinio modeliavimo savybę, kuri pasireiškė ir ką tik mūsų atliktame eksperimente: analizuodami imitavimo eigą ir jo rezultatus, mes kartais pastebime tai, į ką paprastai nekreipiame dėmesio.

Mūsų ką tik nagrinėtu atveju, tai – „didžioji prizo“ neišlošimo efektas.

Lošimų organizatoriai, stengdamiesi privilioti daugiau klientų, siūlo už palyginti nedidelę ir visiems priimtina kainą išgyti galimybę išlošti išpūdingą sumą, neišpėdami savo klientų, kad tokio išlošio tikimybė labai maža, ir todėl gana dažnai „didysis prizas“, viliojęs lošėjus, taip ir lieka neišloštas.

Toks reiškinys daugumoje atvejų reikšmingai padidina lošimo organizatorių pelną. Jis tampa žymiai didesnis negu apskaičiuotas normatyvinis vidutinis pelnas.

Mūsų nagrinėjamu atveju apskaičiuotas vidutinis pelnas sudaro $\frac{4,0000 - 3,2896}{4,0000} \cdot 100\% = 17,76\%$, o imitavimo metu dėl to, kad „didysis prizas“ liko neišloštas, gautas pelnas sudarė $\frac{40,4 - 17,1 - 4,5 - 1,20}{40,4} \cdot 100\% \approx 64,38\%$. Žinoma, grynas pelnas yra mažesnis, nes reikėtų atsižvelgti ir

į lošimo organizavimo išlaidas. Tačiau pelno padidėjimas, neišlošus „didžiojo prizo“, vis tik lieka gana reikšmingas. Rezultatas nelabai pasikeistų net žymiai padidinus lošimų skaičių, nes tokie atvejai, kai beveik būtų išlošiamas „didysis prizas“, pasitaikytų retai.

Pateiksiu dar vieną pavyzdėlį, kai eksperimentai su imitaciniais modeliais padėjo eksperimentatoriams akyliau ir giliau pažvelgti į analizuojamus reiškinius.

Dar gyvendamas ir dirbdamas Vakarų Sibire, aš buvau pastebėjęs, kad dideliuose pramoniniuose miestuose rytais, kai didelės minios žmonių skuba į darbą, autobusai ir troleibusai, aptarnaujantys ilgus maršrutus, pradeda važinėti „paketais“ (po kelis iškart, vienas paskui kitą), nebesilaikydami tvarkaraščių ir stipriai vėluodami, nors tuomet individualių automobilių Vakarų Sibiro miestuose buvo dar nedaug ir transporto kamščiai, taip būdingi dabartiniams miestams, transporto judėjimui tuomet dar netrukdė.

Kai kurie nepatenkinti keleiviai tokį reiškinį tuomet aiškino autobusų ir troleibusų vairuotojų nedrausmingumu: esą, pastarieji galinėse maršruto stotelėse lošia kortomis arba domino ir todėl išvažiuoja iš tų stotelių ne pagal tvarkaraštį, o grupėmis.

Tokių atvejų tuomet irgi kartais pasitaikydavo, tačiau pagrindinė šio reiškinio priežastis man paaiškėjo tik tada, kai su studentais pabandžiau modeliuoti eismo judėjimą.

Dėl transporto priemonių stokos, tais laikais stotelėse rytais būdavo didelės grūstys, nemaža dalis laukiančių keleivių netilpdavo į atvažiuosį autobusą ar troleibusą, vairuotojai ilgai vargdavo, kol sugebėdavo uždaryti autobusų bei troleibusų duris, o dažnai to padaryti ir nesugebėdavo: taip ir važiuodavo su kabančiais tarpduriuose keleiviais.

Mums, t. y. man ir mano studentams, buvo pavesta ištirti keleivių srautus ir pasiūlyti, kaip racionaliau paskirstyti turimas transporto priemones, kurių tuomet labai trūko, permetant jas iš mažiau apkrautų maršrutų į labiau apkrautus, ir tokiu būdu pagerinti keleivių aptarnavimą.

Surinkę statistinius duomenis apie tai, kiek keleivių ir kokiais tempais renkasi tose ar kitose maršrutų stotelėse, mes pradėjome imituoti tų keleivių pervežimą, laikydami, jog transporto priemonės iš savo pradinių stotelių, kuriuose įrengti dispečerių punktai, išvažiuoja nustatyta tvarka, o tarpinėse stotelėse stengiasi užtrukti tik tiek, kiek užtrunka keleivių įsėdimas. Įsėdimo trukmę apskaičiuodavome, atsižvelgdami į prognozuojamą stotelėje susirinkusių keleivių skaičių ir į tai, kad tuo atveju, kai keleivių daug, dėl susidariusios grūsties, keleivių įsėdimo procesas sulėtėja.

Mus nustebino, kad mūsų modeliuojami autobusai ir troleibusai po keliolikos stotelių rytais pradėdavo judėti „paketais“, panašiai kaip ir realūs autobusai ir troleibusai. Šiek tiek panagrinėję šį reiškinį, mes nustatėme jo priežastį.

Keleivių, atvykstančių į stoteles, srautus, mes imituodavome, naudodamiesi atsitiktinių skaičių generatoriais, atsižvelgdami į tų srautų statistines charakteristikas, kurias buvome nustatę statistinių tyrimų metu. Imituojamų srautų vidutinės charakteristikos sutapdavo su mūsų anksčiau nustatytomis realių srautų vidutinėmis charakteristikomis, tačiau atskirais imitavimo momentais tie srautai galėdavo būti ir didesni, ir mažesni už vidutinius.

Tarkime, kokioje nors stotelėje atsitiktinai susirinko žymiai daugiau keleivių negu vidutinis jų skaičius. Tuomet keleivių įsėdimas užtrunka ilgiau negu vidutiniškai. Į kitą stotelę autobusus atvyksta šiek tiek pavėlavęs, todėl čia randa daugiau keleivių ne tik atsitiktinai: per ilgesnį laukimo laiką jų susirinko daugiau. Autobuso vėlavimas dar padidėja, ir tas vėlavimo laikas kaupiasi, nes kitose stotelėse vėluojantis autobusas randa vis daugiau keleivių.

Kadangi vėluojantis autobusas išvažiuoja iš stotelės vėliau negu buvo numatyta, tai laiko tarpas tarp šio autobuso išvažiavimo iš tos stotelės ir kito autobuso atvažiavimo į šią stotelę sumažėja. Sumažėja ir keleivių, susikaupiančių per tą laiką stotelėje, skaičius. Todėl sutrumpėja jų susodinimo laikas, ir antrasis autobusas pradeda vyti vėluojantį pirmąjį autobusą. Ir štai jau du autobusai atvažiuoja į kokią nors stotelę kartu ar beveik kartu. Net jei antrasis autobusas aplenkė pirmąjį (nors jo vairuotojas norėdamas išvengti vėluojančių keleivių „šturmo“ ir durų laužymo, greičiausiai, to nedarys), tai jie tik pasikeistų vaidmenimis, o autobusų vėlavimas ir toliau didėtų, dėl ko šiuos du autobusus pavytų trečiasis autobusas. Štai, jau ir autobusų „paketas“.

Šį pavyzdėlį pateikiau todėl, kad parodyčiau, jog imituojant sudėtingus procesus galima susidurti su reiškiniais, kurie nebuvo specialiai programuojami. Imituojant sudėtingų sistemų

funkcionavimą tai pasitaiko gana dažnai. Tai rodo, jog imitacinis modeliavimas ir eksperimentai su matematiniais modeliais gali būti ne tik sprendinių paieškos, bet ir sudėtingų procesų tyrimo bei jų pažinimo priemonė. Norint šia priemone efektyviai naudotis, tikslinga jos taikymą racionalizuoti ir kompiuterizuoti.

Norėdami generuoti diskrečiai pasiskirsčiusius su tikimybėmis p_i dydžius x_i kompiuteriu, t. y. norėdami gauti su minėtomis tikimybėmis diskrečiai pasiskirsčiusių dydžių η_j failą, galime pasinaudoti šiuo **standartiniu algoritmu**:

1) išrikiuojame (sunumeruojame) tikimybes p_i ir dydžius x_i tikimybių reikšmių mažėjimo tvarka: $p_0 \geq p_1 \geq p_2 \geq \dots \geq p_{n-1} \geq p_n$, kur $\sum_{i=0}^n p_i = 1$; tokia tvarka išrikiuotas

tikimybes p_i ir dydžius x_i įrašome į kompiuterio atmintį kaip du vienamačius failus arba kaip vieną porų (p_i, x_i) failą; į kompiuterio atmintį taip pat įrašome mums reikalingų generuotinių atsitiktinių dydžių η_j skaičių N (arba koki nors kitą pakankamai didelį sveiką skaičių), išskiriame kompiuterio atmintyje vietą generuojamų dydžių η_j užrašymui (jei tie dydžiai η_j bus tuoj pat panaudojami, tai kompiuterio atmintį galima ir pataupyti, neišskiriant joje vietos dydžių η_j užrašymui); indeksui j priskiriame pradinę reikšmę, lygią 1, t. y. įvykdome priskyrimo operaciją $j:=1$ ir perduodame programos valdymą į antrąjį algoritmo bloką;

2) kreipiamės į atsitiktinių (pseudoatsitiktinių) tolygiai pasiskirsčiusių intervale $[0;1)$ dydžių ξ_j generatorių (pavyzdžiui, į vieną iš 1.14.4. skyrelyje aprašytų generatorių) ir generuojame eilinį atsitiktinį dydį ξ_j ; gautąją ξ_j reikšmę priskiriame programos pagalbiniam kintamajam M , kitam pagalbiniam kintamajam P – pradinę reikšmę p_0 , o indeksui i – pradinę reikšmę, lygią 0, t. y. įvykdome priskyrimo operacijas: $M:=\xi_j$, $P:=p_0$, $i:=0$; po to valdymą perduodame į trečiąjį algoritmo bloką;

3) apskaičiuojame naują kintamojo M reikšmę $M:=M-P$ ir, jei $M < 0$, tai valdymą perduodame ketvirtajam, o jei $M \geq 0$ – penktajam blokui;

4) surandame dydžių x_i faile (kurį sudarėme pirmajame bloke) eilinį dydį x_i ir priskiriame jo reikšmę eiliniam dydžiui η_j ($\eta_j:=x_i$), įrašydami tą reikšmę į generuojamų diskrečiai pasiskirsčiusių su tikimybėmis p_i atsitiktinių dydžių η_j sąrašo j -ją vietą (arba perduodami dydį η_j į atitinkamą imitacinio modeliavimo algoritmo, kuriam reikalingi tie dydžiai η_j , bloką); jei $j < N$, tai įvykdome operaciją $j:=j+1$ ir valdymą perduodame antrajam blokui, o jei $j=N$, tai skaičių η_j generavimo algoritmas baigia darbą;

5) įvykdome operaciją $i:=i+1$ ir iš tikimybių p_i failo iškviečiame eilinės tikimybės p_i reikšmę p_i^* , kad ją priskirtume pagalbiniam programos kintamajam P , t. y. įvykdome operaciją $P:=p_i^*$; po to programos valdymą perduodame į trečiąjį bloką.

Tikimybių p_i išrikiavimas jų mažėjimo tvarka neturi įtakos dydžių η_j pasiskirstymui, nes dydžiui ξ_j pasiskirstę intervale $[0;1)$ tolygiai, tačiau tai turi įtakos čia aprašyto standartinio algoritmo darbo trukmei, kadangi sumažina ciklą tarp trečiojo ir penktojo blokų skaičių: dažniau vyksta perdavimai iš trečiojo bloko į ketvirtąjį negu

į penktąjį bloką, nes skaičiai ξ_i dažniau patenka į ilgesnius intervalus $\left[\sum_{i=0}^k p_i; \sum_{i=0}^{k+1} p_i \right)$ negu į trumpesnius. Todėl ir tikslinga tuos intervalus išdėstyti atkarpoje $[0;1)$ jų trumpėjimo tvarka, t. y. taip, jog, didėjant k , intervalų ilgiai mažėtų.

Daugelyje diskrečių skirstinių (binominiame, Puasono, geometriname, hipergeometriname ir t. t.) dydžiai x_i yra sveiki skaičiai i , išsidėstę į natūralią eilutę $i=0, 1, 2, 3, \dots, n$. Tokiais atvejais tampa nebereikalingas x_i reikšmių failo formavimas, jo įsiminimas ir kreipimasis į jį ketvirtajame standartinio algoritmo bloke, kas leidžia taupyti ne tik kompiuterio atmintį, bet ir jo darbo laiką.

Generuojant tokius skirstinius galima neįsiminti ir tikimybių p_i reikšmių failo, įsimenant tik kokią nors pradinę tikimybę (pavyzdžiui p_0) ir santykį $r(i) = \frac{p_{i+1}}{p_i}$, leidžiantį apskaičiuoti eilinės tikimybės p_{i+1} reikšmę, naudojantis rekurentine formule $p_{i+1} = r(i) \cdot p_i$.

Surasime $r(i)$ formules minėtiems skirstiniams.

1. Binominiam skirstiniui su parametrais p ir n tikimybė $p_i = P(\eta=i) = C_n^i p^i (1-p)^{n-i}$, todėl $r(i) = \frac{C_n^{i+1} p^{i+1} (1-p)^{n-i-1}}{C_n^i p^i (1-p)^{n-i}} = \frac{p}{1-p} \cdot \frac{n-i}{i+1}$.

2. Puasono skirstiniui su parametru λ tikimybė, kad per laiko vienetą įvyks lygiai i įvykių, priklausančių Puasono įvykių srautui, turinčiam vidutinį intensyvumą λ , yra lygi $p_i = \frac{\lambda^i}{i!} e^{-\lambda}$, todėl $r(i) = \frac{\lambda^{i+1} i!}{(i+1)! \lambda^i} = \frac{\lambda}{i+1}$.

3. Analogiškai gauname, jog geometriniam skirstiniui su parametru p tikimybė $p_i = P(\eta=i) = p(1-p)^i$ ir todėl $r(i) = 1-p$, o hipergeometriniam skirstiniui su parametrais n , n_1 ir m tikimybė $p_i = P(\eta=i) = \frac{C_n^i \cdot C_{n-n_1}^{m-i}}{C_n^m}$, kur $i \in [\max(0; n_1 + m - n); \min(n_1, m)]$, o

$$r(i) = \frac{(n-i)(m-i)}{(i+1)(n-n_1-m+i+1)}.$$

Atsižvelgiant į čia išvardintas tokių sveikaskaitinių skirstinių savybes, aukščiau aprašytąjį standartinį atsitiktinių dydžių η_j generavimo algoritmą galima modifikuoti.

Pirmajame bloke nebereikėtų rikiuoti ir įsiminti dydžių p_i ir x_i , o pakaktų įsiminti dydžius p_0 ir N bei santykio $r(i)$ formulę.

Ketvirtajame bloke nebereikėtų ieškoti dydžio x_i tų dydžių faile ir pakaktų įvykdyti tik labai paprastą priskyrimo operaciją $\eta_j := i$, paliekant visas kitas bloko operacijas be pasikeitimų.

Penktajame bloke nebereikėtų ieškoti eilinės p_i reikšmės tų reikšmių faile, o pakaktų po operacijos $i := i+1$ operaciją $P := p_i$ pakeisti operacija $P := P \cdot r(i)$.

Toks standartinio algoritmo patobulinimas leistų sutaupyti šiek tiek kompiuterio atminties ir jo darbo laiko, kas yra svarbu, kuriant tas programas, kurios bus dažnai naudojamos.

Tačiau taip tobulindami standartinį algoritmą, mes susiduriame su tuo faktu, kad ne visiems skirstiniams ir ne visada $r(i) < 1$, t. y., ne visos tikimybės p_i didėjant i , mažėja, o tai reiškia, jog, naudodamiesi patobulintu algoritmu, mes ne visada minimizuosime ciklą tarp trečiojo ir penktojo blokų skaičių. Pavyzdžiui, Puasono skirstinyje, kai $\lambda > 1$, tikimybės p_i , indeksui i didėjant pradedant nuo $i=0$, pradžioje taip pat didėja, nes pradžioje $r(i) = \frac{\lambda}{i+1} > 1$. Kai i tampa lygus $[\lambda]$, kur $[\lambda]$ – sveikoji

parametro λ dalis, tikimybės p_i reikšmė pasiekia maksimumą $\frac{\lambda^{[\lambda]}}{[\lambda]!} e^{-\lambda}$ ir tik po to, didėjant i , pradeda mažėti.

Norėdami sumažinti ciklą tarp trečiojo ir penktojo blokų skaičių, tokiais atvejais turėtume pradėti tikimybių p_i reikšmių peržiūrą ne būtinai nuo $i=0$, o nuo tos i reikšmės, kuri atitinka maksimalią p_i reikšmę (pavyzdžiui, generuojant Puasono

skirstinį, – nuo $i^* = [\lambda]$, ir po to intervalų $\left[\sum_{i=0}^k p_i; \sum_{i=0}^{k+1} p_i \right)$ peržiūrą vykdyti į dešinę ir į

kairę nuo $i=i^*$. Skaitytojas pats gali sugalvoti, kaip tuo atveju reikėtų modifikuoti generavimo algoritmą, o jei tai jam nepavyktų, – gali pasiskaityti knygoje [160].

Kai reikia modeliuoti dvimatį (bendrai, n -matį) diskretų pasiskirstymą, tai reikia atitinkamai suskirstyti atkarpas $[0;1)$ ant dviejų (bendrai, n) koordinatinių ašių ir generuoti du atsitiktinius skaičius ξ ir η (bendrai, n skaičių $\xi_1, \xi_2, \dots, \xi_n$), nustatant atkarpėles, į kurias tie skaičiai patenka, ir nustatant atitinkamus dvimatės (n -matės) erdvės taškus bei tuos įvykius, kurie tuos taškus (ir tas atkarpėles) atitinka.

Kai kurių specialių diskrečių srautų generavimui (pavyzdžiui, Puasono, Erlango ir kitų įvykių) gali būti naudojami ir specialūs metodai, apie kuriuos bus papasakota vėlesniuose skyreliuose (žiūr. 1.14.6. ir 1.14.9. skyrelius).

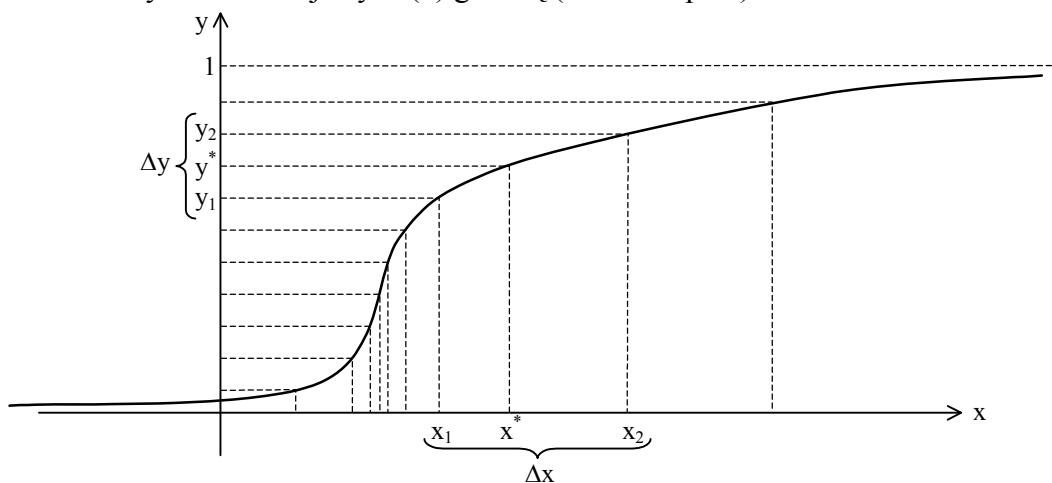
1.14.6. Tolydziai pasiskirsčiusių atsitiktinių bei pseudoatsitiktinių skaičių sekų generavimas atvirkštinės funkcijos metodu

Iš tolygaus tolydaus skirstinio atkarpoje $[0;1)$ galima gauti ne tik diskrečių, bet ir įvairių tolydžių skirstinių. Tai galima padaryti įvairiais būdais. Aprašysiu vieną jų – **atvirkštinės funkcijos metoda**.

Tarkime, jog mes turime tolygiai pasiskirsčiusių atkarpoje $[0;1)$ atsitiktinių skaičių y generatorių, generuojantį skaičius, turinčius pakankamai daug ženklų, kad pasiskirstymą būtų galima laikyti pakankamai artimu tolydžiam, ir norime gauti atsitiktinių skaičių x tolydų pasiskirstymą su tankiu $f(x)$, t. y., su integraline to

pasiskirstymo funkcija $F(x) = \int_{-\infty}^x f(t) dt$, kur $F(x^*)$ lygi tikimybei $P(x < x^*)$.

Sudarysime funkcijos $y=F(x)$ grafiką (žiūr. 299 pav.).



299 pav.

Funkcijos $y=F(x)$ reikšmė gali keistis nuo 0 iki 1. Generuosime atsitiktinius skaičius y , tolygiai pasiskirsčiusius atkarpoje $[0;1)$ su tankiu $\phi_1(y)=1$. Kiekvienai atsitiktinei y_i reikšmei rasime tokią atsitiktinio dydžio x_i reikšmę, kuri tenkintų lygybę $y_i=F(x_i)$. Tai galima padaryti analiziškai, jei žinome funkciją $F(x)$ ir galime išspręsti lygtį $y=F(x)$, t. y. rasti $x_i=F^{-1}(y_i)$, kur F^{-1} yra funkcija, atvirkštinė funkcijai F (iš čia ir metodo pavadinimas). Dydžio x reikšmę, atitinkančią atsitiktinę dydžio y reikšmę, galima rasti ir grafiškai (žiūr. 299 pav., kur y reikšmei y^* randama x reikšmė x^*). Nesunku pastebėti, jog, kai atsitiktinio dydžio y pasiskirstymas yra tolygus su tankiu

$\varphi_1(y)=1$, tai atsitiktinio dydžio x pasiskirstymas gali būti ir netolygus. Rasime dydžio x pasiskirstymo tankį $\varphi_2(x)$ ir įrodysime, jog jis lygus duotajam tankiui $f(x)$.

Atkarpoje $0 \leq y < 1$ paimsime atkarpėlę $\Delta y = y_2 - y_1$ ir rasime ją atitinkančią atkarpėlę $\Delta x = x_2 - x_1$ (žiūr. 299 pav.). Kai $\Delta y \rightarrow 0$, tai ir $\Delta x \rightarrow 0$.

Tikimybė $P(y^* \in \Delta y)$, jog atsitiktinis skaičius y^* pateks į atkarpėlę Δy , lygi $\int_{y_1}^{y_2} \varphi_1(y) dy$. Su ta pačia tikimybe skaičius x^* pateks į atkarpėlę Δx . Todėl $\int_{x_1}^{x_2} \varphi_2(x) dx = \int_{y_1}^{y_2} \varphi_1(y) dy$.

Kadangi $\varphi_1(y)=1$, tai $\int_{x_1}^{x_2} \varphi_2(x) dx = y_2 - y_1 = F(x_2) - F(x_1) = \int_{-\infty}^{x_2} f(x) dx - \int_{-\infty}^{x_1} f(x) dx = \int_{x_1}^{x_2} f(x) dx$.

Lygybė $\int_{x_1}^{x_2} \varphi_2(x) dx = \int_{x_1}^{x_2} f(x) dx$ galioja **bet kuriai x reikšmių porai x_2, x_1** , kai $x_2 \geq x_1$, todėl iš šios integralų lygybės išplaukia pointegracinių funkcijų lygybė $\varphi_2(x) = f(x)$, nes iš lygybės $\forall x \left[\int_x^{x+\Delta x} \varphi_2(x) dx = \int_x^{x+\Delta x} f(x) dx \right]$ išplaukia lygybė

$$\forall x \left[\lim_{\Delta x \rightarrow 0} \frac{\int_x^{x+\Delta x} \varphi_2(x) dx}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{\int_x^{x+\Delta x} f(x) dx}{\Delta x} \right], \text{ t. y. } \forall x [\varphi_2(x) = f(x)], \text{ ką ir reikėjo įrodyti.}$$

Sudarysime pseudoatsitiktinių skaičių x , **eksponentiškai pasiskirsčiusių intervale $[0; +\infty)$** su tankiu $f(x) = \begin{cases} 0, & \text{kai } x < 0 \\ \lambda e^{-\lambda x}, & \text{kai } x \geq 0 \end{cases}$ **generavimo atvirkštinės funkcijos metodu algoritmą.**

1. Sprendžiame lygtį $y = F(x)$, t. y. lygtį $y = \int_0^x \lambda e^{-\lambda x} dx$ ir surandame $x = F^{(-1)}(y)$. Kadangi $y = (-e^{-\lambda x}) \Big|_0^x = 1 - e^{-\lambda x}$, tai $x = \frac{1}{\lambda} \ln \frac{1}{1-y}$.

2. Naudodamiesi jau turimu pseudoatsitiktinių skaičių, tolygiai pasiskirsčiusių intervale $[0; 1)$, generatoriumi, generuojame pseudoatsitiktinių skaičių y .

3. Įstatome y reikšmę, gautą antrajame algoritmo žingsnyje, į formulę $x = \frac{1}{\lambda} \ln \frac{1}{1-y}$, gautą pirmajame algoritmo žingsnyje, ir apskaičiuojame artutinę x reikšmę.

4. Kartojame antro ir trečiojo žingsnių porą tiek kartų, kiek pseudoatsitiktinių skaičių x norime gauti.

Sudarysime dydžio x , pasiskirsčiusio pagal eksponentinį dėsnį, kai $\lambda=0,5$, atsitiktinių reikšmių lentelę (žiūr. 102 lentelę), pasinaudodami 101 lentele.

Atsitiktinių dydžių x sekos, kurioje tų dydžių reikšmės būtų pasiskirsčiusios pagal eksponentinį dėsnį, generavimo algoritmą buvo galima truputį supaprastinti, naudojantis tais atvejais, kai $y \neq 0$, vietoj formulės $x = \frac{1}{\lambda} \ln \frac{1}{1-y}$ paprastesne formule $x = \frac{1}{\lambda} \ln \frac{1}{y}$. Tokį supaprastinimą galima daryti todėl, kad tais atvejais, kai $y \neq 0$, atsitiktiniai dydžiai y ir $1-y$ priklauso tam pačiam skirstiniui, kuriame atsitiktiniai dydžiai tolygiai pasiskirstę intervale $(0;1)$

Tolygiai pasiskirsčiusio atkarpoje $[0,1)$ dydžio y reikšmės	$1-y$	$\frac{1}{1-y}$	$\ln \frac{1}{1-y}$	Eksponentiškai pasiskirsčiusio su tankiu $f(x)=\lambda e^{-\lambda x}$ dydžio $x=\frac{1}{\lambda} \ln \frac{1}{1-y}$ reikšmės
0,2677	0,7323	1,3655605	0,311565	0,62313
0,0080	0,9920	1,0080645	0,008032	0,016064
0,7545	0,2455	4,0733197	1,404459	2,808918
0,3910	0,6090	1,6420361	0,495937	0,991874
0,2135	0,7865	1,2714558	0,240162	0,480324
0,1000	0,9000	1,1111111	0,10536	0,21072
0,0051	0,9949	1,0051261	0,005113	0,010226
0,6978	0,3022	3,3090668	1,196667	2,393334
0,7624	0,2376	4,2087542	1,437167	2,874334
0,7263	0,2737	3,6536353	1,295724	2,591448

Galima dar labiau sutrumpinti kompiuterio laiko sąnaudas, generuojant iškart po du ar net po tris eksponentinio skirstinio skaičius η_j ir tokiu būdu sumažinant tokios sudėtingos funkcijos kaip logaritmas $\ln$ reikšmės apskaičiavimo dažnį.

Pavyzdžiui, generuojant po du skaičius η_j , elgiamasi taip:

1) generuojami trys tolydziai pasiskirstę intervale $[0,1)$ atsitiktiniai dydžiai ξ_1, ξ_2, ξ_3 ;

2) sudaroma seka ξ'_0, ξ'_1, ξ'_2 , kurioje $\xi'_0=0, \xi'_1=\xi_3, \xi'_2=1$;

3) apskaičiuojamos generuojamų eksponentiškai pasiskirsčiusių atsitiktinių dydžių reikšmės η_1 ir η_2 :

$$\eta_1=(\xi'_0-\xi'_1)\ln(\xi_1\cdot\xi_2)=(0-\xi_3)\ln(\xi_1\cdot\xi_2)=-\xi_3\ln(\xi_1\cdot\xi_2);$$

$$\eta_2=(\xi'_1-\xi'_2)\ln(\xi_1\cdot\xi_2)=(\xi_3-1)\ln(\xi_1\cdot\xi_2).$$

Tokiu būdu gaunami du skaičiai η_1 ir η_2 , logaritmo $\ln(\xi_1\cdot\xi_2)$ reikšmę skaičiuojant tik vieną kartą. Generuojant naują dydžių η_j porą, pakanka generuoti tik vieną naują dydį ξ_4 , nes anksčiau generuotus dydžius ξ_2 ir ξ_3 galima panaudoti kaip naujus ξ_1 ir ξ_2 , o ξ_4 – kaip ξ_3 . Taip slenkant generuojamų skaičių ξ_i seka pirmyn, galima gauti vis naujas atsitiktinių eksponentiškai pasiskirsčiusių skaičių η_j poras, skaičiuojant logaritminės funkcijos reikšmę tik po vieną kartą.

Kai norima generuoti po tris skaičius η_j , pradžioje generuojama 5 atsitiktinių dydžių ξ_i seka $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5$ ir sudaroma papildoma seka $\xi'_0, \xi'_1, \xi'_2, \xi'_3$, kurioje $\xi'_0=0, \xi'_1=\min(\xi_4, \xi_5), \xi'_2=\max(\xi_4, \xi_5), \xi'_3=1$.

Po to apskaičiuojamos eksponentiškai pasiskirsčiusių atsitiktinių dydžių η_1, η_2, η_3 reikšmės:

$$1) \eta_1=(\xi'_0-\xi'_1)\ln(\xi_1\cdot\xi_2\cdot\xi_3)=-\min(\xi_4, \xi_5)\ln(\xi_1\cdot\xi_2\cdot\xi_3);$$

$$2) \eta_2=(\xi'_1-\xi'_2)\ln(\xi_1\cdot\xi_2\cdot\xi_3)=[\min(\xi_4, \xi_5)-\max(\xi_4, \xi_5)]\ln(\xi_1\cdot\xi_2\cdot\xi_3);$$

$$3) \eta_3=(\xi'_2-\xi'_3)\ln(\xi_1\cdot\xi_2\cdot\xi_3)=[\max(\xi_4, \xi_5)-1]\ln(\xi_1\cdot\xi_2\cdot\xi_3).$$

Tokiu būdu gauname tris skaičius η_j , skaičiuodami logaritmo $\ln(\xi_1\cdot\xi_2\cdot\xi_3)$ reikšmę tik vieną kartą.

Skaitytojui, kurį domintų šių patobulintų eksponentinio skirstinio generavimo algoritmų teorinis pagrindimas, galima rekomenduoti jau minėtą gana išsamią statistinio modeliavimo metodams ir jų taikymui skirtą knygą [160]. Toje knygoje jis ras gana daug įdomių ir naudingų žinių, o taip pat ir atsitiktinių skaičių generavimo algoritmų, kurie neaprašyti šioje mokymo priemonėje.

Turėdami laiko intervalų seką, kurių trukmės sudaro eksponentinį skirstinį, galime sudaryti ir

Puasono įvykių srautą, kurio įvykių įvykimo tikimybės tenkina lygybę $p_k(t)=\frac{(\lambda t)^k}{k!}e^{-\lambda t}$, kur $p_k(t)$ –

tikimybė, kad per laiko intervalą t įvyks lygiai k įvykių, λ – vidutinis įvykių srauto intensyvumas, t. y. vidutinis įvykių, įvykstančių per laiko vienetą, skaičius. Tokia galimybė išplaukia iš diskretaus Puasono srauto savybės, kad **laiko intervalai tarp gretimų Puasono įvykių pasiskirstę tolydziai ir sudaro**

eksponentinį skirstinį su tikimybinio tankiu $f(t)=\begin{cases} 0, & \text{kai } t < 0, \\ \lambda e^{-\lambda t}, & \text{kai } t \geq 0. \end{cases}$

Nesunku įrodyti ir atvirkštinį teiginį: jei spindulyje Ox į dešinę nuo jo pradžios O išdėstytume vieną po kitos atkarpos, kurių ilgai $\tau_i=x_i-x_{i-1}$ būtų atsitiktinio dydžio τ , pasiskirsčiusio su tikimybinio

tankiu $f(\tau) = \begin{cases} 0, & \text{kai } \tau < 0, \\ \lambda e^{-\lambda\tau}, & \text{kai } \tau \geq 0 \end{cases}$, atsitiktinės reikšmės, tai taškai x_i , kur $i=1, 2, 3, \dots$, būtų išsibarstę spindulyje Ox taip, jog tikimybė $p_k(\Delta\ell)$, kad į bet kurią to spindulio atkarpą, turinčią ilgį $\Delta\ell$, pateks lygiai k taškų x_i , bus lygi $\frac{(\lambda\Delta\ell)^k}{k!} e^{-\lambda\Delta\ell}$. Čia pateiktų teiginių įrodymus galima rasti tikimybių teorijos vadovėliuose, tačiau, prieš kreipiantis į tuos vadovėlius, būtų naudinga pabandyti tuos įrodymus atrasti savarankiškai.

Baigdamas skyrelį 1.14.6., skirtą atvirkštinės funkcijos metodo aprašymui, norėčiau **atkreipti dėmesį į atsitiktinių dydžių tikimybių skirstinių integralinių funkcijų savybę, kuria pasinaudojant ir buvo sukurtas atvirkštinės funkcijos metodo algoritmas**, leidžiantis tolygų intervale $[0,1]$ skirstinį transformuoti į tikimybinių skirstinių, turintį duotąją integralinę funkciją $y=F(x)$ bei duotąjį tikimybinių tankį $f(x)$.

Galima įrodyti, jog integralinės funkcijos $y=F(x)=\int_{-\infty}^x f(x)dx = P\{X < x\}$ reikšmė y , atitinkanti

atsitiktinio dydžio X reikšmę x , kurios pasiskirstymo tikimybinių tankis yra $f(x)$, atsitiktinai kintant tai atsitiktinio dydžio X reikšmei x , keičiasi taip, kad jos, t. y. funkcijos $y=F(x)$ reikšmės, pasiskirstymo tikimybinių tankis $g(y)$ yra pastovus ir lygus 1, nepriklausomai nuo to, koks yra atsitiktinio dydžio X reikšmės x pasiskirstymo tankis $f(x)$. Trumpai tariant, **kaip bebūtų pasiskirsčiusios atsitiktinio dydžio**

X reikšmės x , integralinės funkcijos $F(x)=\int_{-\infty}^x f(x)dx$ reikšmės y pasiskirsto vienodai – intervale

$[0,1]$ su tikimybiniu tankiu $g(y)=1$.

Įrodysiu šį (iš pirmo žvilgsnio galintį pasirodyti keistoku) teiginį, išsivedamas atsitiktinio dydžio Y reikšmės $y=F(x)=\int_{-\infty}^x f(z)dz$ integralinę funkciją $G(y)=\int_0^y g(y)dy = P\{Y < y\}$, kurią sudaręs

galėsiu apskaičiuoti tikimybinių tankį $g(y)=\frac{dG(y)}{dy}$.

Kadangi $G(y)$ ir $F(y)$ yra integralinės funkcijos, tai jos – nemažėjančios funkcijos, reiškiančios tikimybes $P\{Y < y\}$ ir $P\{X < x\}$ atitinkamai. Abi tos funkcijos, didėjant jų argumentų y ir x reikšmėms, keičiasi nuo 0 iki 1. Todėl šioms funkcijoms yra teisingi šie predikatai: 1) $(x_2 > x_1) \Rightarrow [F(x_2) \geq F(x_1)]$; 2) $(y_2 > y_1) \Rightarrow [G(y_2) \geq G(y_1)]$; 3) $[F(x_2) > F(x_1)] \Rightarrow (x_2 > x_1)$; 4) $[G(y_2) > G(y_1)] \Rightarrow (y_2 > y_1)$.

Iš šių predikatų ir funkcijų $F(x)$ ir $G(y)$ apibrėžimų išplaukia lygybių seka: $G(y)=P\{Y < y\}=P\{F(X) < F(y)\}=P\{X < y\}=F(y)=y$.

Kadangi $G(y)=y$, tai $g(y)=\frac{dG(y)}{dy}=\frac{dy}{dy}=1$, ką ir reikėjo įrodyti.

Vadinasi, integralinės funkcijos $y=F(x)$ perveda bet kurio atsitiktinio dydžio X reikšmių x tikimybinių skirstinių, turintį bet kuri tikimybinių tankį $f(x)$, į atsitiktinio dydžio Y reikšmių y , tolygiai pasiskirsčiusių intervale $[0,1]$, tikimybinių skirstinių. **Taikydami atvirkštinės funkcijos metodą, mes atliekame procedūrą F^{-1} , atvirkštinę procedūrai F , gaudami iš $y=F(x)$ funkcija $x=F^{-1}(y)$, t. y. iš tikimybinių skirstinio, turinčio tikimybinių tankį $g(y)=1$, gauname tikimybinių skirstinį, turintį tikimybinių tankį $f(x)$.**

Čia ką tik įrodyto teiginio teisingumo įrodymą galėjome gauti ir pasinaudodami tikimybių teorijos teorema (žiūr. [165]), teigiančia, jog jei atsitiktinis dydis Z yra atsitiktinio dydžio X funkcija $z=\varphi(x)$ ir atsitiktinio dydžio X reikšmių x pasiskirstymo tankis yra $f(x)$, tai atsitiktinio dydžio Z reikšmių z pasiskirstymo tikimybinių tankis $g(z)$ yra lygus $f(\varphi^{-1}(z)) \cdot \left| \frac{d\varphi^{-1}(z)}{dz} \right|$.

Laikydami, jog $z=y$, o $\varphi(x)=F(x)=\int_{-\infty}^x f(x)dx$, suradę atvirkštinę funkciją $\varphi^{-1}(z)=F^{-1}(y)$ bei jos

išvestinės $\frac{dF^{-1}(y)}{dy}$ modulį, gautume, jog $g(y)=1$.

Siūlyčiau skaitytojui, pabandyti tai padaryti ir įsitikinti, jog naudojantis minėtąja teorema, pritaikant ją šiam daliniam atvejui, įrodymo procedūra būtų gerokai ilgesnė ir painesnė už mūsų ką tik

pateiktą teiginio $g(y)=1$ įrodymą. Ji taptu dar ilgesnė, jei prijungtume ir minėtosios teoremos, pasiskolintos iš [165], įrodymą. Mums gi teiginio, iš kurio išplaukia atvirkštinės funkcijos algoritmo idėja, įrodymui prireikė tik vienos eilutės lygybių ir nelygybių, gautų iš integralinės funkcijos ir tikimybinio tankio apibrėžimų.

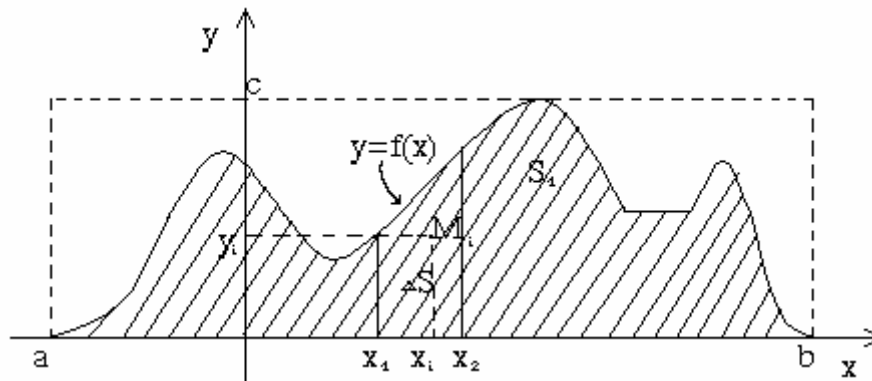
Šis pavyzdėlis rodo, kad **sąvokų apibrėžimus naudinga ne tik žinoti, bet ir suprasti, mokėti jais kūrybiškai naudotis**. Tada loginiai samprotavimai gali padėti išvengti sudėtingų skaičiavimų, pakeisti juos.

Atvirkštinės funkcijos metodas yra patogus tada, kai nesunku rasti generuojamo skirstinio integralinę funkciją $F(x)$ ir jos atvirkštinę funkciją $F^{(-1)}(y)$, t. y. nesunku išspręsti lygtį $y=F(x)$.

Kai integralinę funkciją $F(x)$ ar jos atvirkštinę funkciją $F^{(-1)}(x)$ rasti sunku ar neįmanoma, galima pasinaudoti **Neumano metodu**, kuris yra beveik universalus, nes leidžia generuoti bet kokią skirstinį baigtiniame intervale $[a;b]$, jei tik žinomas to skirstinio tankis $f(x)$, $x \in [a;b]$. Už universalumą, žinoma, tenka „mokėti“: generuojant tolydžių skirstinių atsitiktinius dydžius Neumano metodu, reikia daugiau tolygiai pasiskirsčiusių intervale $[0;1]$ atsitiktinių ar pseudoatsitiktinių skaičių ir todėl tenka įvykdyti žymiai daugiau operacijų negu naudojantis atvirkštinės funkcijos metodu.

1.14.7 Neumano metodas

Tarkime, mums reikia generuoti atsitiktinius dydžius x , pasiskirsčiusius atkarpoje $[a;b]$ tolydžiai su tankiu $f(x)$, kurio grafikas $y=f(x)$ vaizduojamas 300 pav.



300 pav.

Randomai $c = \max f(x)$ ir sudarome stačiakampį su viršūnėmis taškuose $(a;0)$, $(b;0)$, $(b;c)$, $(a;c)$.

Kai atsitiktinis dydis η_i bus tolydžiai ir tolygiai pasiskirstęs atkarpoje $[0;1]$ su tankiu $\varphi_1(\eta)=1$, tai atsitiktinis dydis $y_i=c\eta$ bus tolydžiai ir tolygiai pasiskirstęs atkarpoje $[0;c]$ su tankiu $\varphi_2(y) = \frac{1}{c}$.

Kai atsitiktinis dydis ξ bus tolydžiai ir tolygiai pasiskirstęs atkarpoje $[0;1]$ su tankiu $\varphi_1(\xi)=1$, tai atsitiktinis dydis $x_i=a+(b-a)\xi$ bus tolydžiai ir tolygiai pasiskirstęs atkarpoje $[a;b]$ su tankiu $\varphi_3(x) = \frac{1}{b-a}$.

Turėdami dydžius x_i ir y_i , galime surasti tašką M_i , turintį šias koordinatas. Jis būtinai pateks į mūsų sudarytą stačiakampį, tačiau gali patekti, o gali ir nepatekti į užstričiuotą sritį po kreive $y=f(x)$. Tuo atveju, kai taškas M_i patenka į užstričiuotą sritį, jo abscisę x_i priskiriame generuojamai atsitiktinių skaičių x^* sekai, o kai M_i nepatenka į užstričiuotą plotą – tašką M_i atmetame kaip netenkinantį skaičių x_i .

patekimo į aibę $\{x^*\}$ sąlygos $y_i \leq f(x_i)$. Todėl kartais Neumano metodą vadina **eliminavimo metodu** [160].

Kartodami šią procedūrą, galime gauti reikiamo ilgio pseudoatsitiktinių skaičių x^* seką.

Ši procedūra vadinama **Neumano algoritmu**. Jį sudaro šie žingsniai.

1. Generuojame atsitiktinį (pseudoatsitiktinį) dydį η_1 , tolydziai ir tolygiai pasiskirsčiusį atkarpoje $[0;1)$.

2. Randame atsitiktinį (pseudoatsitiktinį) dydį $y_1 = c\eta_1$.

3. Generuojame atsitiktinį (pseudoatsitiktinį) dydį ξ_1 tolydziai ir tolygiai pasiskirsčiusį atkarpoje $[0;1)$.

4. Randame atsitiktinį (pseudoatsitiktinį) dydį $x_1 = a + (b-a)\xi_1$.

5. Tikriname sąlygą $f(x_1) \geq y_1$, t. y. sąlygą $f(a + (b-a)\xi_1) \geq c\eta_1$.

Jei ši sąlyga išpildyta, tai reiškia, jog taškas $M_1(x_1, y_1)$ priklauso sričiai, kuri 300 pav. padengta štrichais. Tada $x_1 = a + (b-a)\xi_1$ reikšmę įrašome į atsitiktinių dydžių x^* , pasiskirsčiusių atkarpoje $[a; b)$ su reikiamu tankiu $f(x)$, sąrašą – aibę $\{x^*\}$.

Jei ši sąlyga neišpildyta, tai taškas $M_1(x_1, y_1)$ priklauso stačiakampio sričiai, kuri 300 pav. nepadengta štrichais. Tokią $x_1 = a + (b-a)\xi_1$ reikšmę mes į generuojamų atsitiktinių dydžių x^* , pasiskirsčiusių su tankiu $f(x)$, sąrašą neįtraukiame.

6. Kartodami $1 \div 5$ žingsnius, mes galime didinti mums reikalingų atsitiktinių dydžių x^* sekos ilgį (suprantama, kad ne kiekvienas $1 \div 5$ žingsnių pakartojimas papildo mūsų sąrašą nauja atsitiktinio dydžio x^* reikšme), t. y. galime gauti reikalingą x^* reikšmių skaičių.

Nesunku įrodyti, jog x^* reikšmės sąrašas bus pasiskirsčiusios su tankiu $\varphi_4(x^*) \approx f(x)$. Įrodymo esmę sudaro šie samprotavimai, išreiškiantys Neumano metodo idėją:

1. Po pakankamai didelio eksperimentų skaičiaus visas stačiakampio, pavaizduoto 300 pav., plotas bus padengtas apytikriai tolygiai išsidėsčiusiais taškais M_i , kur $M_i(x_i, y_i) = M_i(a + (b-a)\xi_i; c\eta_i)$.

2. Į sąrašą $\{x^*\}$ pateks abscisės tik tų taškų, kurie priklausys sričiai, padengtai štrichais. Todėl tame sąrašas bus daugiau tokių x_i^* reikšmių, kurioms atitinka didesnė funkcijos $f(x_i)$ reikšmė, t. y. tankis $\varphi_4(x^*)$ turės būti apytikriai proporcingas tankiui $f(x)$. O mums kaip tik to ir reikia. Belpieka tik įrodyti, kad tankiai $\varphi_4(x^*)$ ir $f(x)$ – ne tik apytikriai proporcingi, bet ir apytikriai lygūs.

Įrodymas, kad $\varphi_4(x^*) \approx f(x)$, galėtų atrodyti taip:

1) išskirsime **bet kurioje 300 pav. užštrichuotoje srityje S_1** bet koki plotelį ΔS , apribotą linijomis $y=0$, $y=f(x)$, $x=x_1$, $x=x_2$, kur x_1 ir x_2 – bet kurios x reikšmės, tenkinančios sąlygą $a \leq x_1 < x_2 \leq b$;

2) tarsime, jog pseudoatsitiktiniai dydžiai x^* pasiskirstę intervale $[a; b)$ su tankiu $\varphi_4(x)$, todėl tikimybė $P(a \leq x^* < b) = \int_a^b \varphi_4(x) dx = 1$, o tikimybė $P(x_1 \leq x^* < x_2) =$

$$= \int_{x_1}^{x_2} \varphi_4(x) dx ;$$

3) kadangi, esant pakankamai dideliu taškų M_i skaičiui, jų išsibarstymas stačiakampio plote (o tuo pačiu ir plote S_1) bus apytikriai tolygus, tai iš skaičių x^* gavimo algoritmo (Neumano algoritmo) išplaukia, jog $\frac{P(x_1 \leq x^* < x_2)}{P(a \leq x^* < b)} = \frac{P(\Delta S)}{P(S_1)} \approx \frac{\Delta S}{S_1}$, t. y. tikimybės $P(x_1 \leq x^* < x_2)$ ir $P(a \leq x^* < b)$, būdamos lygios taškų M_i patekimo į

atitinkamus plotelius ΔS ir S_1 tikimybėms $P(\Delta S)$ ir $P(S_1)$, bus proporcingos tiems ploteliams ΔS ir S_1 ;

4) plotas $S_1 = \int_a^b f(x)dx = 1$, nes bet kokiam tikimybiniam tankiui galioja lygybė

$$\int_{-\infty}^{\infty} f(x)dx = 1, \text{ o mūsų atveju intervaluose } -\infty < x < a \text{ ir } b \leq x < +\infty \text{ } f(x)=0 \text{ ir todėl}$$

$$\int_{-\infty}^{\infty} f(x)dx = \int_a^b f(x)dx ;$$

5) kadangi plotelis (plotas) $\Delta S = \int_{x_1}^{x_2} f(x)dx$, tai iš (3), (2) ir (4) gauname

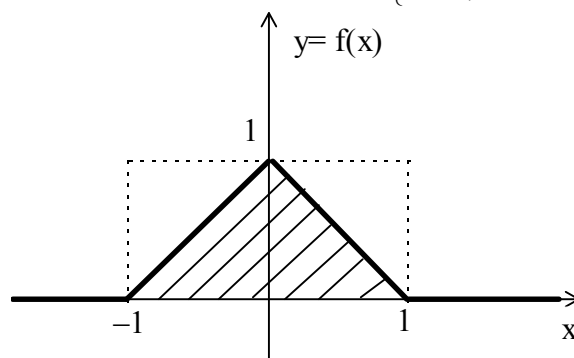
$$\int_{x_1}^{x_2} \varphi_4(x)dx \approx \int_{x_1}^{x_2} f(x)dx ;$$

6) kadangi, kaip išplaukia iš (1), **lygybė (5) galioja bet kuriems intervalo $[a;b]$ taškams $x_2 > x_1$** (bet kurioje to intervalo vietoje, o taip pat ir tuo atveju kai $\Delta x = x_2 - x_1$ artėja prie 0), tai iš lygybės (5) gauname lygybę $\varphi_4(x) \approx f(x)$, ką ir reikėjo įrodyti.

Apytikrė lygybė $\varphi_4(x) \approx f(x)$ bus tuo artimesnė lygybei $\varphi_4(x) = f(x)$, kuo daugiau taškų M_i (ir tuo pačiu – skaičių x^*) bus generuota ir kuo geresni bus pseudoatsitiktinių skaičių ξ_i ir η_i generatoriai, t. y. kuo tų skaičių pasiskirstymas bus artimesnis tolygiam tolydziam pasiskirstymui intervale $[0; 1)$.

Pailiustruosiu Neumano algoritmo taikymą konkrečiais pavyzdžiais ir tuo pačiu palyginsiu jį su atvirkštinės funkcijos algoritmo taikymu.

Tarkime, jog mums reikia generuoti pseudoatsitiktinių skaičių seką, kurioje skaičiai x būtų pasiskirstę su „trikampio“ skirstinio tikimybinio tankiu $f(x) = \begin{cases} 0, & \text{kai } x < -1, \\ 1 - |x|, & \text{kai } -1 \leq x < 1, \text{ (žiūr. 301 pav.)} \\ 0, & \text{kai } x \geq 1 \end{cases}$



301 pav.

Skaičius ξ_i ir η_i imsime iš 101 lentelės, t. y. naudosimės mūsų turimu pseudoatsitiktinių skaičių, tolygiai pasiskirsčiusių intervale $[0; 1)$, generatoriumi.

Kadangi mūsų nagrinėjamu atveju $a = -1$ ir $b = 1$, tai iš $\xi_1 = 0,2677$ gauname $x_1 = a + (b - a)\xi_1 = -1 + 2\xi_1 = -0,4646$.

Kadangi mūsų nagrinėjamu atveju $c = 1$, tai $y_1 = \eta_i$ ir iš lygybės $\eta_1 = 0,0080$ gauname $y_1 = 0,0080$.

$f(x_1)=1-|x_1|=1-0,4646=0,5354$. Ir bendrai, kai $-1 \leq x_i < 1$, o tokią nelygybę dydis $x_i = -1 + 2\xi_i$ visada tenkina, nes $0 \leq \xi_i < 1$, iš lygybių $f(x_i)=1-|x_i|$ ir $x_i = -1 + 2\xi_i$ gauname, kad

$$f(x_i) = \begin{cases} 2\xi_i, & \text{kai } \xi_i < \frac{1}{2}, \\ 2(1-\xi_i), & \text{kai } \xi_i \geq \frac{1}{2}. \end{cases}$$

Kadangi $y_1 < f(x_1)$, nes $0,0080 < 0,5354$, tai taškas M_1 patenka į užstrichuotą sritį, ir $x_1 = -0,4646$ priskiriamas sekai $\{x^*\}$, t. y. $x_1^* = -0,4646$.

Paėmę $\xi_2 = 0,7545$ ir $\eta_2 = 0,3910$, gauname: $x_2 = 0,5090$, $y_2 = 0,3910$, $f(x_2) = 0,4910$.

Kadangi $y_2 \leq f(x_2)$, tai M_2 patenka į užstrichuotą sritį, ir x_2 į aibę $\{x^*\}$: $x_2^* = 0,5090$.

Paėmę $\xi_3 = 0,2135$ ir $\eta_3 = 0,1000$, gauname: $x_3 = -0,5730$, $y_3 = 0,1000$, $f(x_3) = 0,4270$.

Kadangi ir šiuo atveju $y_3 < f(x_3)$, tai $x_3^* = x_3 = -0,5730$.

Paėmę $\xi_4 = 0,0051$ ir $\eta_4 = 0,6978$, gauname: $x_4 = -0,9898$, $y_4 = 0,6978$, $f(x_4) = 0,0102$.

Šiuo atveju $y_4 > f(x_4)$, todėl taškas M_4 nepatenka į užstrichuotą sritį, o x_4 reikšmė atmetama, t. y. neįtraukiama į seką $\{x^*\}$, kurioje po šio žingsnio pasilieka kol kas tik trys anksčiau gauti skaičiai.

Paėmę $\xi_5 = 0,7624$ ir $\eta_5 = 0,7263$, gauname: $x_5 = 0,5248$, $y_5 = 0,7263$, $f(x_5) = 0,4752$.

Kadangi $y_5 > f(x_5)$, tai x_5 taip pat atmetame ir sekoje $\{x^*\}$ išlieka kol kas tik trys skaičiai.

Paėmę $\xi_6 = 0,6606$ ir $\eta_6 = 0,1363$, gauname: $x_6 = 0,3212$, $y_6 = 0,1363$, $f(x_6) = 0,6788$.

Kadangi $y_6 < f(x_6)$, tai x_6 priskiriame sekai $\{x^*\}$, t. y. $x_6^* = x_6 = 0,3212$.

Tęsdami šio algoritmo darbą, galime gauti tokio ilgio skaičių x^* seką $\{x^*\}$, kokio mums reikia, jei tik tas ilgis ne didesnis už tą, nuo kurio prasidės pseudoatsitiktinių skaičių kartojimasis. Tokiais atvejais tektų arba keisti dauginamuosius, t. y. keisti vieną pseudoatsitiktinių skaičių generatorių kitu, arba didinti ženklų skaičių pradiniuose dauginamuosiuose. Tęsdami skaičių x^* generavimą ir po to, kai prasidėtų pseudoatsitiktinių skaičių ξ ir η kartojimasis, mes gautume periodiškai besikartojančių skaičių x^* seką, kuri jau nebeatitiktų duotojo tankio $f(x)$, nesančio periodine funkcija.

Pseudoatsitiktinių skaičių x seką, pasiskirsčiusią su tikimybinio tankiu

$$f(x) = \begin{cases} 0, & \text{kai } x < -1, \\ 1 - |x|, & \text{kai } -1 \leq x < 1, \\ 0, & \text{kai } x \geq 1, \end{cases} = \begin{cases} 0, & \text{kai } x < -1, \\ 1 + x, & \text{kai } -1 \leq x \leq 0, \\ 1 - x, & \text{kai } 0 \leq x < 1, \\ 0, & \text{kai } x \geq 1, \end{cases} \quad \text{galima generuoti ir}$$

nesinaudojant Neumano metodu, o naudojantis atvirkštinės funkcijos metodu, nes šiuo atveju nesunku rasti integralinę funkciją $y = F(x)$ ir atvirkštinę jai funkciją

$$x = F^{(-1)}(y) : y = F(x) = \int_{-\infty}^x f(x) dx = \begin{cases} 0, & \text{kai } x < -1, \\ \frac{1}{2}(x+1)^2, & \text{kai } -1 \leq x \leq 0, \\ 1 - \frac{1}{2}(x-1)^2, & \text{kai } 0 \leq x < 1, \\ 1, & \text{kai } x \geq 1, \end{cases}$$

$$x = F^{(-1)}(y) = \begin{cases} -1 + \sqrt{2y}, & \text{kai } 0 \leq y \leq \frac{1}{2}, \\ 1 - \sqrt{2-2y}, & \text{kai } \frac{1}{2} \leq y < 1. \end{cases}$$

Pasinaudodami 101-ja lentele ir priimdami, kad joje gauti pseudoatsitiktiniai skaičiai yra tikimybės $y = F(x)$, gauname ieškomąją skaičių seką:

$$x_1 = -1 + \sqrt{2 \cdot 0,2677} \approx -0,2683;$$

$$x_2 = -1 + \sqrt{2 \cdot 0,0080} \approx -0,8735;$$

$$x_3 = 1 - \sqrt{2 - 2 \cdot 0,7545} \approx 0,2993;$$

$$x_4 = -1 + \sqrt{2 \cdot 0,3910} \approx -0,1157;$$

$$x_5 = -1 + \sqrt{2 \cdot 0,2135} \approx -0,3465;$$

$$x_6 = -1 + \sqrt{2 \cdot 0,1000} \approx -0,5528;$$

$$x_7 = -1 + \sqrt{2 \cdot 0,0051} \approx -0,8990;$$

$$x_8 = 1 - \sqrt{2 - 2 \cdot 0,6978} \approx 0,2226;$$

$$x_9 = 1 - \sqrt{2 - 2 \cdot 0,7624} \approx 0,3107;$$

$$x_{10} = 1 - \sqrt{2 - 2 \cdot 0,7263} \approx 0,2601;$$

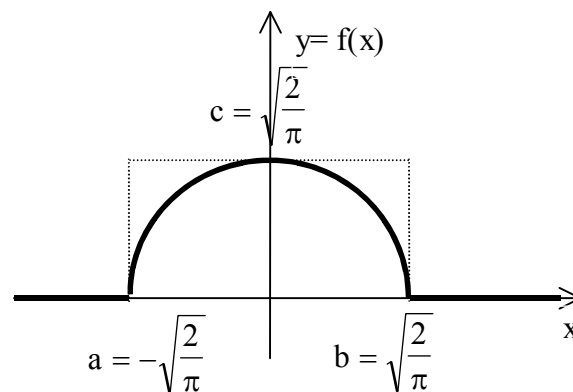
$$x_{11} = 1 - \sqrt{2 - 2 \cdot 0,6606} \approx 0,1761;$$

$$x_{12} = -1 + \sqrt{0,1363} \approx -0,4779 \text{ ir t. t.}$$

Lygindami Neumano ir atvirkštinės funkcijos metodus, matome, kad, naudodamiesi Neumano metodu, mes iš dvylikos pseudoatsitiktinių skaičių, paimtų iš 101-sios lentelės, sugebėjome gauti tik keturis mums reikalingus pseudoatsitiktinius skaičius x^* , pasiskirsčiusius su duotuoju tankiu, kai tuo tarpu naudodamiesi atvirkštinės funkcijos metodu, iš dvylikos tų pačių pseudoatsitiktinių skaičių y mes sugebėjome gauti visą dvyliktuką mums reikalingų skaičių x .

Tai rodo, jog tais atvejais, **kai galima pasinaudoti atvirkštinės funkcijos metodu, tiksliau yra naudotis juo, o ne Neumano metodu**. Pastaruoju metodu tikslinga naudotis tik tada, kada pasinaudoti atvirkštinės funkcijos metodu nėra galimybės.

Pavyzdžiui, jei mums reikėtų generuoti pseudoatsitiktinių skaičių seką, priklausančią „pusapskritimio“ skirstiniui, turinčiam tankį $f(x) = \sqrt{\frac{2}{\pi} - x^2}$, kur $x \in \left[-\sqrt{\frac{2}{\pi}}; \sqrt{\frac{2}{\pi}}\right]$ (žiūr. 302 pav.), tai, nors ir sugebėtume rasti šio skirstinio integralinę funkciją $y=F(x)$, tačiau pasinaudoti atvirkštinės funkcijos metodu mes negalėtume, nes nesugebėtume išspręsti lygties $y=F(x)$ kintamojo x atžvilgiu, t. y. nesugebėtume rasti atvirkštinės funkcijos $x=F^{(-1)}(y)$.



302 pav.

Integruodami tankio funkciją $f(x) = \sqrt{\frac{2}{\pi} - x^2}$, $x \in \left[-\sqrt{\frac{2}{\pi}}; \sqrt{\frac{2}{\pi}}\right]$, gautume:

$$y=F(x) = \int_{-\infty}^x \sqrt{\frac{2}{\pi} - x^2} dx = \int_{-\sqrt{\frac{2}{\pi}}}^x \sqrt{\frac{2}{\pi} - x^2} dx = \frac{1}{2} + \frac{x}{\sqrt{2\pi}} \sqrt{1 - \frac{\pi x^2}{2}} + \frac{1}{\pi} \arcsin x \sqrt{\frac{\pi}{2}}.$$

Rasti analizinį šios lygties $y=F(x)$ sprendinį $x=F^{(-1)}(y)$, kaip matome, yra neįmanoma, o artutinio sprendinio paieškos pareikalautų tiek skaičiavimų, kad pasidarytų efektyviau naudotis Neumano, o ne atvirkštinės funkcijos metodu.

Naudodamiesi Neumano metodu, tankio funkcija $f(x)=\sqrt{\frac{2}{\pi}-x^2}$ ir 101-ja lentelę, gauname:

$$\begin{aligned} 1) \quad \xi_1=0,2677, \eta_1=0,0080, x_1=-\sqrt{\frac{2}{\pi}}+2\sqrt{\frac{2}{\pi}}\xi_1 &= \sqrt{\frac{2}{\pi}}(-1+2\xi_1)=\sqrt{\frac{2}{\pi}}(-1+2\cdot 0,2677)\approx \\ \approx -0,370697; y_1=C\eta_1 &= \sqrt{\frac{2}{\pi}}\cdot 0,0080\approx 0,006383; f(x_1)=\sqrt{\frac{2}{\pi}-x_1^2}=\sqrt{\frac{2}{\pi}-\left[\sqrt{\frac{2}{\pi}}(-1+2\xi_1)\right]^2}= \\ =2\sqrt{\frac{2}{\pi}}\sqrt{\xi_1-\xi_1^2} &=2\sqrt{\frac{2}{\pi}}\sqrt{0,2677(1-0,2677)}\approx 0,706543. \end{aligned}$$

Kadangi $y_1 < f(x_1)$, tai $x_1 = -0,370697$ priskiriame sekai $\{x^*\}$, t. y. $x_1^* = -0,370697$.

$$\begin{aligned} 2) \quad \xi_2=0,7545, \eta_2=0,3910; x_2 &= -\sqrt{\frac{2}{\pi}}+2\sqrt{\frac{2}{\pi}}\xi_2=\sqrt{\frac{2}{\pi}}(-1+2\cdot 0,7545)\approx 0,406123; \\ y_2 &= \sqrt{\frac{2}{\pi}}\cdot 0,3910\approx 0,311973; f(x_2)=\sqrt{\frac{2}{\pi}-x_2^2}=2\sqrt{\frac{2}{\pi}}\sqrt{\xi_2(1-\xi_2)}=2\sqrt{\frac{2}{\pi}}\sqrt{0,7545\cdot 0,2455}\approx 0,686792. \end{aligned}$$

Kadangi $y_2 < f(x_2)$, tai $x_2 \approx 0,406123$ priskiriame sekai $\{x^*\}$, t. y. $x_2^* = 0,406123$.

$$3) \quad \xi_3=0,0051, \eta_3=0,1000, x_3 \approx -0,457188, y_3 \approx 0,0797885, f(x_3) \approx 0,653911.$$

Kadangi $y_3 < f(x_3)$, tai $x_3 \approx -0,457188$ priklauso $\{x^*\}$, t. y. $x_3^* = -0,457188$.

$$4) \quad \xi_4=0,2135, \eta_4=0,6978, x_4 \approx -0,789746, y_4 \approx 0,556764, f(x_4) \approx 0,113670.$$

Kadangi $y_4 > f(x_4)$, tai $x_4 \approx -0,789746$ nepriklauso $\{x^*\}$.

$$5) \quad \xi_5=0,7624, \eta_5=0,7263, x_5 \approx 0,418730, y_5 \approx 0,579504, f(x_5) \approx 0,679180.$$

Kadangi $y_5 < f(x_5)$, tai $x_5 \approx 0,418730$ priklauso $\{x^*\}$, t. y. $x_5^* = 0,418730$.

$$6) \quad \xi_6=0,6606, \eta_6=0,1363, x_6 \approx 0,256281, y_6 \approx 0,108752, f(x_6) \approx 0,377803.$$

Kadangi $y_6 < f(x_6)$, tai $x_6 \approx 0,256281$ priklauso $\{x^*\}$, t. y. $x_6^* = 0,256281$.

Šią procedūrą tęsdami iki tol, kol neprasidės sekos kartojimasis, gauname pradinę sekos atkarpą,

$$\text{kurioje skaičiai bus pasiskirstę su tankiu, apytikriai atitinkančiu tankį } f(x)=\sqrt{\frac{2}{\pi}-x^2}, x \in \left[-\sqrt{\frac{2}{\pi}}; \sqrt{\frac{2}{\pi}}\right].$$

Jei tos atkarpos ilgis mūsų reikmėms pasirodytų per trumpas, sekos formavimą galėtume pratęsti, pakeisdami buvusį pseudoatsitiktinių skaičių generatorių nauju, taip pat generuojančiu skaičius ξ_i ir η_i , tolygiai pasiskirsčiusius intervale $[0; 1)$.

Sugeneravę n porų skaičių ξ_i, η_i , galėtume gauti skaičių x^* , tenkinančių „pusapskritimio“ dėsnį,

$$\text{aibę (seką), kurioje būtų apytikriai } n_1 \approx \frac{\pi R^2}{R \cdot 2R} n = \frac{\pi}{4} n \approx 0,7845n \text{ skaičių. Pavyzdžiui, sugeneravę } 70 \text{ porų}$$

skaičių ξ_i ir η_i , gautume apie 55 skaičius x^* , kai tuo tarpu, gamindami skaičius x^* , tenkinančius anksčiau išnagrinėtąjį „trikampio“ pasiskirstymo dėsnį, iš 70 porų skaičių ξ_i, η_i , būtume gavę apytikriai tik 35 skaičius x^* .

Kaip jau buvo minėta anksčiau, Neumano metodas, būdamas „gremėzdiškesnis“ už atvirkštinės funkcijos metodą, yra žymiai universalesnis už pastarąjį. Tačiau ir Neumano metodas nėra absoliučiai universalus. Šį metodą galima taikyti, kai **generuojamo skirstinio rėžiai a ir b yra baigtiniai**, ko nereikalauja atvirkštinės funkcijos metodas.

Todėl **egzistuoja skirstiniai, kurių teoriškai nepavyksta generuoti nei atvirkštinės funkcijos, nei Neumano metodu**. Tokio skirstinio pavyzdys – gerai žinomas ir plačiai taikomas normalusis skirstinys $N(a; \sigma)$, o taip pat standartinė (centruota ir normalizuota) jo modifikacija – standartinis normalusis skirstinys $N(0;1)$.

Tokių (ir kitų panašių) skirstinių generavimui taikomi **specialūs metodai**. Su vienu iš jų – **sumavimo metodu** – taikomu normalaus skirstinio generavimui, susipažinsime kitame skyrelyje. Ten susipažinsime ir su kai kuriais kitais specialiais metodais.

Žinoma, atsižvelgdami į tai, kad, esant didelėms $|x|$ reikšmėms, normalaus skirstinio $N(0;1)$ tankis $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ tampa apytikriai lygus 0 ir artėja prie 0, didėjant $|x|$, mes galime normalaus skirstinio praktiniam generavimui taikyti ir Neumano metodą, pasirinkdami $(-a)=b$, kur $b \gg 3$.

1.14.8. Sumavimo ir kiti specialūs metodai

Sumavimo metodas remiasi tikimybių teorijos centrine ribine teorema [164, 165], iš kurios išplaukia, jog nepriklausomų atsitiktinių bet kaip pasiskirsčiusių dydžių sumos skirstinys, didėjant dėmenų skaičiui, artėja prie normalaus skirstinio.

Praktiniam statistiniam modeliavimui reikalingų normaliai pasiskirsčiusių atsitiktinių (pseudoatsitiktinių) skaičių generavimui pakanka ne taip jau didelio dėmenų skaičiaus m . Gana dažnai naudojamas $m=12$, nes, pirma, tokio sumos dėmenų skaičiaus jau pakanka, kad gautas skirstinys būtų artimas normaliam, ir, antra, esant tokiai m reikšmei susipaprastina skaičiavimo formulės.

Norėdami gauti atsitiktinių (pseudoatsitiktinių) dydžių z_i seką su skirstiniu, artimu standartiniam normaliam skirstiniui $N(0,1)$ su tankiu $f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$, dydį z_i

galime apskaičiuoti, naudodamiesi formule $z_i = \sqrt{\frac{12}{m}} \left[\sum_{j=1}^m \xi_{(i-1)m+j} - 0,5m \right]$, kur ξ_r

atsitiktinis (pseudoatsitiktinis) skaičius, tolygiai paskirstęs atkarpoje $[0;1)$, r – skaičiaus ξ_r eilės numeris jų sekoje, gautoje tuos skaičiaus generuojant, m – atsitiktinių skaičių ξ_r sumos dėmenų skaičius.

Pavyzdžiui, kad gautume skaičių z_1 , reikia sudėti skaičius $\xi_1, \xi_2, \dots, \xi_{m-1}, \xi_m$; tam, kad gautume skaičių z_2 , reikia generuoti ir sudėti skaičius $\xi_{m+1}, \xi_{m+2}, \dots, \xi_{2m-1}, \xi_{2m}$; tam, kad gautume skaičių z_3 , reikia generuoti ir sudėti skaičius $\xi_{2m+1}, \xi_{2m+2}, \dots, \xi_{3m-1}, \xi_{3m}$ ir t. t.

Dydis $0,5m$ atimamas iš sumos $\sum_{r=1}^m \xi_r$ tam, kad dydžių z pasiskirstymo centrą gautume taške $\bar{z}=0$, nes $M[\xi] = \int_0^1 \xi f(\xi) d\xi = \int_0^1 \xi \cdot 1 \cdot d\xi = \left. \frac{\xi^2}{2} \right|_0^1 = 0,5$, o atsitiktinių skaičių ξ sumos $\sum_{r=1}^m \xi_r$ vidurkis lygus tų skaičių vidurkių sumai ir todėl $M \left[\sum_{j=1}^m \xi_j \right] = 0,5m$.

$$\text{Todėl } M[z_i] = M \left\{ \sqrt{\frac{12}{m}} \left[\sum_{j=1}^m \xi_{(i-1)m+j} - 0,5m \right] \right\} = 0.$$

$$\text{Kadangi } M[\xi^2] = \int_0^1 \xi^2 f(\xi) d\xi = \left. \frac{\xi^3}{3} \right|_0^1 = \frac{1}{3}, \quad \sigma_\xi = \sqrt{D_\xi} = \sqrt{M(\xi^2) - [M(\xi)]^2} =$$

$$= \sqrt{\frac{1}{3} - \frac{1}{4}} = \sqrt{\frac{1}{12}} \text{ ir nepriklausomų atsitiktinių dydžių sumos dispersija lygi tų dydžių}$$

dispersijų sumai, tai, naudodamiesi formule $z_i = \sqrt{\frac{12}{m}} \left[\sum_{j=1}^m \xi_{(i-1)m+j} - 0,5m \right]$, gauname, jog

$$\sigma_{z_i} = 1.$$

Kai $m=12$, tai dydžio z_i apskaičiavimo formulė tampa paprastesnė:

$$z_i = \sum_{j=1}^{12} \xi_{12(i-1)+j} - 6.$$

Kai reikia generuoti atsitiktinių (pseudoatsitiktinių) dydžių x_i pasiskirstymą, artimą normaliam pasiskirstymui $N(a; \sigma)$ su tankiu $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}$, tai dydį x_i galima apskaičiuoti, naudojantis formule $x_i = a + \sigma z_i$, kur z_i – atsitiktinis (pseudoatsitiktinis) dydis, apskaičiuotas naudojantis formule $z_i = \sqrt{\frac{12}{m}} \left[\sum_{j=1}^m \xi_{(i-1)m+j} - 0,5m \right]$.

Sumavimo metodo trūkumas – reikia gana didelio atsitiktinių (pseudoatsitiktinių) skaičių, tolygiai pasiskirsčiusių intervale $[0; 1)$, skaičiaus $m \cdot n$, kad gautume normaliai pasiskirsčiusių atsitiktinių (pseudoatsitiktinių) skaičių seką, turinčią reikalingą ilgį n .

L. I. Boljševs yra pasiūlęs dydžio $\eta_i = \sqrt{\frac{12}{m}} \sum_{j=1}^m \xi_{(i-1)m+j}$, naudojamo sumavimo metodo formulėse, transformaciją į dydį $\varsigma_i = \eta_i - \frac{1}{20m} (3\eta_i - \eta_i^3)$, kuri leidžia žymiai sumažinti sumos dėmenų skaičių m , o tuo pačiu ir skaičių $m \cdot n$, nepabloginant, o neretai net pagerinant dydžių z_i pasiskirstymo asimptotinį normalumą.

Yra sukurta daug specialių metodų, besiremiančių, pavyzdžiui, funkcijų $f(t)$ ir $F(x) = \int_{-\infty}^x f(t) dt$ aproksimavimu ir atvirkštinės funkcijos bei Neumano metodo taikymu, o taip pat ir kitomis idėjomis. Tokie metodai kartais naudojami ir normaliai pasiskirsčiusių atsitiktinių (pseudoatsitiktinių) skaičių generavimui, nes jie sunaudoja mažiau tolygiai pasiskirsčiusių intervale $[0; 1)$ pseudoatsitiktinių skaičių ξ .

Aprašysiu vieną tokį specialų algoritmą, kuris remiasi taip vadinamu **daugiamačio izotropinio vektoriaus metodu**.

Atsitiktinį vektorių $\vec{\omega} = (\omega_1, \omega_2, \dots, \omega_n)$ vadina izotropiniu, kai n -matės erdvės taškai $\frac{\vec{\omega}}{|\vec{\omega}|}$ yra tolygiai ir nepriklausomai pasiskirstę n -mačio rutulio, turinčio centrą

koordinatų pradžioje ir spindulį $R=1$, paviršiuje. Pavyzdžiui, kai $n=2$, tai atsitiktiniai izotropiniai vektoriai $\vec{\omega} = (x, y) = (\rho, \varphi)$ kirs apskritimą $x^2 + y^2 = 1$ taškuose, kurie bus tolygiai išsidėstę tame apskritime. Kitaip tariant, kampas φ bus atsitiktinis tolygiai išsibarstęs intervale $[0; 2\pi)$ dydis.

Kai $n=2m$, kur $m>0$ – koks nors sveikas skaičius, tai mums reikalingi atsitiktiniai (pseudoatsitiktiniai) skaičiai η_j , priklausantys normaliajam standartiniam skirstiniui $N[0; 1]$, naudojantis daugiamačio izotropinio vektoriaus metodu, gaunami taip:

1) generuojami atsitiktiniai (pseudoatsitiktiniai) tolygiai pasiskirstę intervale $[0; 1)$ dydžiai ξ_j ir apskaičiuojami dydžiai $\gamma_j = 1 - 2\xi_j$, $j=1, 2, \dots, n$;

2) apskaičiuojamas dydis $d^2 = \sum_{j=1}^n \gamma_j^2$ ir, jei $d^2 > 1$, tai grįžtama į pirmąjį bloką, generuojamos naujos ξ_j reikšmės ir apskaičiuojamos naujos γ_j reikšmės ir tai kartojama

iki tol, kol gaunamas atvejis kada $d^2 \leq 1$; tuomet apskaičiuojami dydžiai

$$\omega_j = \frac{\gamma_j}{d} = \frac{\gamma_j}{\sqrt{\sum_{j=1}^n \gamma_j^2}};$$

3) apskaičiuojamos mums reikalingų atsitiktinių (pseudoatsitiktinių) skaičių η_j , priklausančių skirstiniui $N[0;1]$ reikšmės: $\eta_j = \sqrt{-2 \ln(\xi_1 \cdot \xi_2 \cdot \dots \cdot \xi_m)} \omega_j$, $j=1, 2, \dots, n=2m$;

4) kai mums reikalingi atsitiktiniai (pseudoatsitiktiniai) skaičiai x_j , priklausantys skirstiniui $N[a;\sigma]$, tai juos randame naudodamiesi formule $x_j = \eta_j \sigma + a$;

5) kai mums reikia daugiau negu $n=2m$ normaliai pasiskirsčiusių skaičių, tai grįžtame į pirmąjį bloką ir kartojame visą procedūrą iki tol kol gauname mums pakankamą tokių skaičių kiekį.

Praktikoje dydžiai η_j dažniausiai generuojami poromis, t. y., pasirenkant $n=2$, nes tuomet daugumoje atvejų sąlyga $d^2 \leq 1$ tenkinama iš pirmojo karto ir eliminavimas, eikvojantis dydžius ξ_j ir neduodantis naujų dydžių η_j , pasitaiko palyginti, retai, o didėjant n tokių eliminavimo atvejų pasitaikymo dažnis didėja.

Dydžiai η_1 ir η_2 tuomet apskaičiuojami naudojantis formulėmis $\eta_1 = \sqrt{-2 \ln \xi_1} \omega_1$, $\eta_2 = \sqrt{-2 \ln \xi_1} \omega_2$ arba formulėmis $\eta_1 = \sqrt{-2 \ln \xi_1} \sin 2\pi \xi_2$, $\eta_2 = \sqrt{-2 \ln \xi_1} \cos 2\pi \xi_2$.

Čia aprašyto metodo pagrindimą ir įrodymą, jog tokiu būdu gauti dydžiai η_j priklausys skirstiniui $N[0;1]$, skaitytojas gali rasti knygoje [160]. Ten jis gali rasti ir gana dažnai statistiniame modeliavime pasitaikančių **beta ir gama – skirstinių generavimo algoritmus** (tame tarpe ir austrų matematiko Jonkos pasiūlytus „egzotiškus“ algoritmus).

Specialūs metodai yra sukurti ne tik tolydžių, bet ir diskrečių skirstinių gavimui.

Pavyzdžiui, dydžius, priklausančius geometriniam skirstiniui $p_i = P(x_i = i) = p(1-p)^i$, galima generuoti ne tik taip, kaip buvo aprašyta 1.14.5. skyrelyje, bet ir iš viso išvengiant intervalų $\left[\sum_{i=0}^k p_i, \sum_{i=0}^{k+1} p_i \right)$ peržiūros bei ciklą tarp algoritmo blokų, t. y. naudojantis formule $x_i = i = \left\lceil \frac{\ln \xi}{\ln(1-p)} \right\rceil$, kurioje skliausteliai $\lceil \cdot \rceil$ išskiria jų viduje esančio skaičiaus sveikąją dalį.

Įrodysime tai, t. y. įrodysime, jog tikimybė, kad, naudojantis formule $x_i = \left\lceil \frac{\ln \xi}{\ln(1-p)} \right\rceil$, skaičius x_i bus lygus i su tikimybe $p_i = p(1-p)^i$.

Kitaip tariant, įrodysime, jog $P\left\{i \leq \frac{\ln \xi}{\ln(1-p)} < i+1\right\} = p(1-p)^i$.

1) $P\left\{i \leq \frac{\ln \xi}{\ln(1-p)} < i+1\right\} = P\left\{i \leq \frac{-\ln \xi}{-\ln(1-p)} < i+1\right\} = P\{-i \ln(1-p) \leq -\ln \xi < -(i+1) \ln(1-p)\} = P\{-\ln(1-p)^i \leq -\ln \xi < -\ln(1-p)^{i+1}\} = P\{\ln(1-p)^i > \ln \xi \geq \ln(1-p)^{i+1}\}.$

2) kadangi $e > 1$, tai iš $\ln(1-p)^{i+1} \leq \ln \xi < \ln(1-p)^i$ išplaukia $(1-p)^{i+1} \leq \xi < (1-p)^i$, todėl gauname, jog $P\left\{i \leq \frac{\ln \xi}{\ln(1-p)} < i+1\right\} = P\{(1-p)^{i+1} \leq \xi < (1-p)^i\}$.

3) kadangi pseudoatsitiktiniai dydžiai ξ yra tolygiai pasiskirstę intervale $[0;1]$ su tankiu $f(\xi)=1$, tai $P\{(1-p)^{i+1} \leq \xi < (1-p)^i\} = \int_{(1-p)^{i+1}}^{(1-p)^i} f(\xi) d\xi = \int_{(1-p)^{i+1}}^{(1-p)^i} d\xi = (1-p)^i - (1-p)^{i+1} = (1-p)^i p$, kas atitinka geometrinio skirstinio tikimybę $p_i = P(x_i=i) = p(1-p)^i$, ką ir reikėjo įrodyti.

Egzistuoja specialūs binominio, Puasono ir kitų diskrečių skirstinių generavimo algoritmai, kuriuos skaitytojas gali rasti jau minėtoje knygoje [160].

Dar daugiau specialių generavimo algoritmų pateikiama knygoje [163]. **Ten galima rasti ne tik normalaus, eksponentinio, beta, gama, bet ir logonormalaus, Veibulo, Raiso ir kitų skirstinių generavimo algoritmus.**

Normalaus skirstinio generavimui ten rekomenduojami du algoritmai A1 ir A2.

Pirmasis algoritmas naudojasi formulėmis $\eta_1 = \sqrt{-2 \ln \xi_1} \cos 2\pi \xi_2$, $\eta_2 = \sqrt{-2 \ln \xi_1} \sin 2\pi \xi_2$ ir sutampa su algoritmu, aprašytu knygoje [160].

Antrasis algoritmas dažnai būna efektyvesnis už pirmąjį, nes nereikalauja kosinusų ir sinusų skaičiavimo. Juo naudojantis, pradžioje apskaičiuojami dydžiai $V1=2\xi_1-1$ ir $V2=2\xi_2-1$, kas dėl skirstinio $N[0;1]$ simetriškumo koordinačių pradžios taško atžvilgiu yra dydžių $\gamma_1=1-2\xi_1$ ir $\gamma_2=1-2\xi_2$, naudojamų knygoje [160], analogija. Po to surandamas dydis $S=(V1)^2+(V2)^2$, kas yra dydžio d^2 analogija. Kai $S \geq 1$, tai grįžtama prie naujų dydžių ξ_1 ir ξ_2 generavimo, naujų dydžių $V1$, $V2$ ir S apskaičiavimo, o kai $S < 1$, tai apskaičiuojami dydžiai $\eta_1 = \sqrt{-\frac{2 \ln S}{S}} \cdot V1$ ir

$$\eta_2 = \sqrt{-\frac{2 \ln S}{S}} \cdot V2.$$

Nesunku įrodyti, jog dydžiai η_1 ir η_2 priklauso skirstiniui $N[0;1]$.

Su kai kuriais kitais specialiais metodais susipažinsime antrojoje šios mokymo priemonės dalyje, nagrinėdami GPSS [161]. Apie atsitiktinių dydžių generavimą, naudojantis kompiuteriu rašoma ir knygoje [165] (žiūr. 135÷137 p.p. ir 7-ją lentelę). Šia knyga galima naudotis kaip savotišku žinyne. Joje skaitytojas gali rasti statistinės informacijos apdorojimo metodų, kurie mums bus reikalingi tolesniuose šio skyriaus skyreliuose, aprašymą, o jos pirmajame priede – trumpą informaciją apie matematinės statistikos taikomųjų programų paketus.

1.14.9. Erlango skirstiniai, jų generavimas ir taikymas. Gama skirstinys

Įvairių empirinių skirstinių generavimui neretai naudojami Erlango skirstiniai, iš kurių, atitinkamai parinkus Erlango skirstinio E_k eilę k , galima gauti skirstinius, pakankamai artimus, duotiesiems empiriniams skirstiniams.

Erlango k -osios eilės skirstinį galima gauti iš Puasono skirstinio, apie kurį jau buvo rašyta 1.14.6. skyrelyje, išrenkant iš puasoninio įvykių srauto $A_1, A_2, A_3, \dots$ tik įvykius $A_k, A_{2k}, A_{3k}, \dots$ ir praleidžiant visus kitus įvykius (po $k-1$ įvykių tarp gretimų Erlango įvykių A_{ik} ir $A_{(i+1)k}$).

Iš to, kas čia pasakyta, išplaukia, jog Erlango pirmosios eilės skirstinio E_1 įvykių srautas sutampa su Puasono įvykių srautu ir laiko intervalai tarp tokių įvykių pasiskirstę eksponentiškai.

Pažymėję laiko intervalų $T^{(k)}$ tarp gretimų Erlango srauto E_k įvykių skirstinio tikimybinį tankį $f_k(t)$, galime sudaryti lygybę $\int_0^t f_k(z) dz = P(T^{(k)} < t) = P(t > T^{(k)})$, iš kurios galima gauti $f_k(t)$ priklausomybės nuo t , k ir λ , kur λ – Puasono srauto vidutinis intensyvumas, formulę.

Iš nelygybės $t > T^{(k)}$ išplaukia, jog per laiko intervalą t turėjo įvykti nemažiau negu k puasoninių įvykių ir galėjo įvykti daugiau negu k puasoninių įvykių.

Todėl $P(t > T^{(k)}) = p_k(t) + p_{k+1}(t) + p_{k+2}(t) + \dots = 1 - p_0(t) - p_1(t) - \dots - p_{k-1}(t) = 1 - e^{-\lambda t} - \sum_{i=1}^{k-1} \frac{(\lambda t)^i}{i!} e^{-\lambda t}$. Diferencijuodami dešiniąją ir kairiąją lygybės $\int_0^t f_k(z) dz =$

$1 - e^{-\lambda t} - \sum_{i=1}^{k-1} \frac{(\lambda t)^i}{i!} e^{-\lambda t}$ puses bei naudodamiesi integralo diferencijavimo formule

$\frac{d}{dy} \int_{\alpha(y)}^{\beta(y)} \varphi(x, y) dx = \int_{\alpha(y)}^{\beta(y)} \frac{\partial \varphi(x, y)}{\partial y} dx + \varphi(\beta(y), y) \frac{d\beta(y)}{dy} - \varphi(\alpha(y), y) \frac{d\alpha(y)}{dy}$, gauname, jog

$$f_k(t) = \lambda e^{-\lambda t} + \lambda \sum_{i=1}^{k-1} \frac{(\lambda t)^i}{i!} e^{-\lambda t} - \lambda \sum_{i=1}^{k-1} \frac{(\lambda t)^{i-1}}{(i-1)!} e^{-\lambda t} = \lambda \left[\sum_{i=0}^{k-1} \frac{(\lambda t)^i}{i!} e^{-\lambda t} - \sum_{j=0}^{k-2} \frac{(\lambda t)^j}{j!} e^{-\lambda t} \right] =$$

$$= \lambda \frac{(\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t} = \left(k \Lambda_k \right)^k \frac{t^{k-1}}{(k-1)!} e^{-k \Lambda_k t}, \text{ kur } \Lambda_k = \frac{1}{T^{(k)}} = \frac{1}{k \bar{\tau}} = \frac{\lambda}{k} - \text{Erlango srauto } E_k$$

vidutinis intensyvumas.

Irodysiu, jog laiko intervalo $T^{(k)}$ tarp dviejų gretimų Erlango srauto įvykių vidurkis $\bar{T}^{(k)}$ tikrai yra lygus $\frac{k}{\lambda}$, ir apskaičiuosiu atsitiktinio dydžio $T^{(k)}$ dispersiją

$$D_{T^{(k)}} = M[(T^{(k)})^2] - (M[T^{(k)}])^2:$$

$$1) \quad \bar{T}^{(k)} = M[T^{(k)}] = \int_0^\infty t f_k(t) dt = \frac{\lambda^k}{(k-1)!} \int_0^\infty t^k e^{-\lambda t} dt. \text{ Kadangi } \int_0^\infty t^k e^{-\lambda t} dt = \frac{k!}{\lambda^{k+1}}, \text{ tai}$$

$$\bar{T}^{(k)} = \frac{\lambda^k}{(k-1)!} \cdot \frac{k!}{\lambda^{k+1}} = \frac{k}{\lambda}.$$

$$2) \quad D_{T^{(k)}} = \int_0^\infty t^2 f_k(t) dt - \left[\int_0^\infty t f_k(t) dt \right]^2; \quad \int_0^\infty t^2 f_k(t) dt = \frac{\lambda^k}{(k-1)!} \int_0^\infty t^{k+1} e^{-\lambda t} dt =$$

$$= \frac{\lambda^k}{(k-1)!} \cdot \frac{(k+1)!}{\lambda^{k+2}} = \frac{k(k+1)}{\lambda^2}; \quad D_{T^{(k)}} = \frac{k(k+1)}{\lambda^2} - \left(\frac{k}{\lambda} \right)^2 = \frac{k}{\lambda^2}.$$

Iš gautų rezultatų išplaukia, jog laiko intervalų $T^{(k)}$ tarp gretimų Erlango srauto

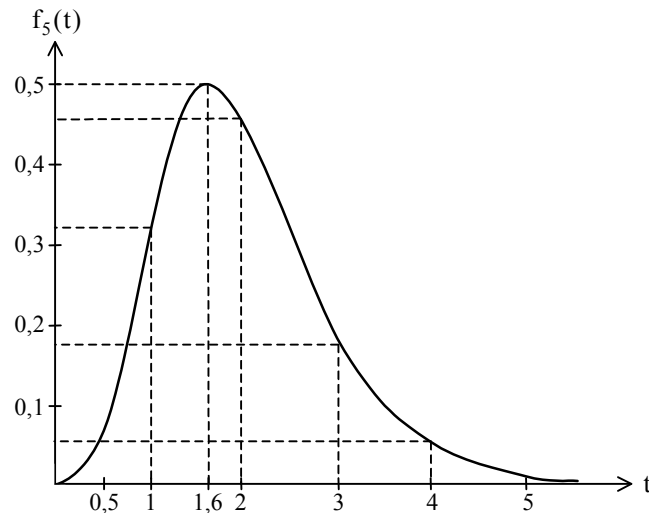
$$E_k \text{ įvykių variacija } V_{T^{(k)}} = \frac{\sigma_{T^{(k)}}}{T^{(k)}} = \frac{\sqrt{D_{T^{(k)}}}}{\bar{T}^{(k)}} = \frac{\sqrt{k\lambda}}{\lambda k} = \frac{1}{\sqrt{k}}.$$

Tai reiškia, jog, keisdami k , mes galime gauti visą eilę atsitiktinių įvykių srautų: nuo labiausiai atsitiktinai išsibarsčiusių įvykių Puasono sraute, kur, kai $k=1$, gauname, jog variacija $v=1$, t. y. laiko intervalo $\tau = T^{(1)}$ vidurkis $\bar{\tau}$ lygus jo išsibarstymo kvadratiniam nuokrypiui σ_τ , iki beveik reguliarių įvykių srautų,

kuriuose laiko intervalų tarp gretimų įvykių variacija, esant pakankamai didelei k reikšmei, yra artima 0.

Iš to ir išplaukia Erlango srautų panaudojimo įvairių empirinių srautų imitavimui galimybė: **parenkant atitinkamą k reikšmę, galima gauti Erlango srautą, pakankamai gerai aproksimuojantį duotąjį empirinį srautą.**

Pavyzdžiui, kai, apdorodami statistinių stebėjimų rezultatus, gauname, jog vidutinė laiko intervalo trukmė tarp atsitiktinių įvykių srauto gretimų įvykių yra $\bar{T}=2$ min. ir tos trukmės T vidutinis kvadratinis nuokrypis $\sigma_T=0,9$ min., tai nagrinėjamąjį empirinį skirstinį galima aproksimuoti penktosios eilės Erlango skirstiniu E_5 , turinčiu tankio funkciją $f_5(t) \approx 4,07t^4 e^{-2,5t}$, kurios grafikas pavaizduotas 303 pav.



303 pav.

Šis grafikas gaunamas taip:

$$1) \text{ kadangi } v_T = \frac{\sigma_T}{\bar{T}} = \frac{1}{\sqrt{k}}, \text{ tai } k = \frac{(\bar{T})^2}{(\sigma_T)^2} = \frac{4}{0,81} \approx 4,95 \approx 5;$$

$$2) \text{ kadangi } \Lambda_k = \frac{1}{\bar{T}}, \text{ tai } \Lambda_k = \frac{1}{2} \text{ ir } k\Lambda_k \approx 2,5;$$

$$3) \text{ kadangi } f_k(t) = (k\Lambda_k)^k \frac{t^{k-1}}{(k-1)!} e^{-k\Lambda_k t}, \text{ tai } f_5(t) \approx (2,5)^5 \frac{t^4}{4!} e^{-2,5t} \approx 4,07t^4 e^{-2,5t};$$

$$4) \text{ kadangi } f'_5(t) \approx 4,07t^3 e^{-2,5t} \cdot (4 - 2,5t) \text{ ir } f'_5(t) > 0, \text{ tai } f_5(t) \text{ turi maksimumą taške } t = \frac{4}{2,5} = 1,6;$$

$$5) f_5(0)=0; f_5(0,5) \approx 0,06; f_5(1) \approx 0,34; f_5(1,6) \approx 0,49; f_5(2) \approx 0,44; f_5(3) \approx 0,19; f_5(4) \approx 0,05.$$

Žinant funkcijos $f_5(t)$ reikšmes ir turint empirinius duomenis, naudojantis, pavyzdžiui, χ^2 ar Stjudento kriterijumi, nesunku įsitikinti, kad šis Erlango skirstinys neblogai aproksimuoja duotąjį empirinį skirstinį.

Erlango skirstinys yra dalinis gama skirstinio, turinčio tikimybinį tankį $f_{\lambda,\eta}(x) = \frac{\lambda^\eta}{\Gamma(\eta)} x^{\eta-1} e^{-\lambda x}$, atvejis, nes **Erlango skirstinį galima gauti iš gama skirstinio, kai**

parametras $\eta > 0$ yra sveikas skaičius $k=1, 2, 3, \dots$. Tuomet $f_{\lambda,k}(x) = \frac{\lambda^k}{\Gamma(k)} x^{k-1} e^{-\lambda x} =$

$= \frac{\lambda^k}{(k-1)!} x^{k-1} e^{-\lambda}$. Pakeitę kintamąjį x kintamuoju t , gauname Erlango skirstinio tankio

funkciją $f_{\lambda,k}(t) = \frac{\lambda^k t^{k-1}}{(k-1)!} e^{-\lambda t}$.

Ir ką tik minėtasis χ^2 („chi kvadrato“) skirstinys taip pat yra dalinis gama skirstinio atvejis. Jis gaunamas, kai gama skirstinio mastelio parametras λ tampa lygus $\frac{1}{2}$, o formos parametras η tampa lygus $\frac{n}{2}$, kur n – teigiamas sveikas skaičius.

Vadinasi, $\chi_n^2 \sim G\left(\frac{1}{2}, \frac{n}{2}\right)$.

Naudodamiesi gama skirstinio formule ir pasirinkdami atitinkamas parametrų λ ir η reikšmes, mes galime imituoti gana įvairius empirinius skirstinius.

1.14.10 Statistinis imitacinis modeliavimas ir jo taikymai

Mokėdami generuoti atsitiktinių skaičių bei pseudoatsitiktinių dydžių ir įvykių sekas, galime imituoti stochastinius procesus, t. y. pasitelkti statistinį modeliavimą stochastinių objektų ir sistemų funkcionavimo imitavimui ir tyrimai, eksperimentuojant su imitaciniais modeliais.

Statistinis modeliavimas žymiai išplečia imitacinio modeliavimo galimybes ir padeda išspręsti daug sudėtingų uždavinių, neišsprendžiamų kitais būdais. Pavyzdžiui, naudojantis analiziniais metodais, apskaičiuoti masinio aptarnavimo sistemų charakteristikas pavyksta tik tais atvejais, kai masinio aptarnavimo sistemų struktūra gana paprasta ir atsitiktinių dydžių srautai gali būti aprašyti paprastais dėsniais (eksponentiniu, Puasono, Erlango ir pan.). Kai srautuose įvykiai pasiskirstę pagal sudėtingesnius dėsnius ar/ir masinio aptarnavimo struktūra sudėtingesnė (hierarchinė, daugiafazinė, tinklinė, su sąveikaujančiais tarp savęs kanalais, grįžtamaisiais ryšiais ir valdymo kontūrais), tai apskaičiuoti tokios sistemos statistines charakteristikas (sistemos praleidžiamąją galią, kanalų vidutinį apkrovimą, aptarnaujamų klientų laukimo eilėse laiką bei vidutinį tų eilių ilgį ir pan.) analiziniais metodais jau nebepavyksta. Tačiau tų charakteristikų artutines reikšmes nesunku apskaičiuoti, pasitelkus statistinį imitacinį modeliavimą ir to modeliavimo rezultatų statistinį apdorojimą matematinės statistikos metodais.

Statistinį modeliavimą galima taikyti ir sudėtingų mišrių procesų imitavimui, kuriuose dalis procesų – stochastiniai, dalis – determinuoti. Tokio modeliavimo pavyzdžiu gali būti jau anksčiau mūsų minėtas darbo centrų sąveikos modelis (DCSM). Kaip kitą statistinio modeliavimo panaudojimo imitaciniuose modeliuose pavyzdį panagrinėsime vieną iš užduočių, kurią įvykdė Vilniaus universiteto trečiojo kurso studentai, studijuodami „Matematinės sistemų teorijos pradmenų“ kursą.

1.14.11. Masinio aptarnavimo sistemos (kirpyklos) funkcionavimo imitavimas ir jo rezultatų statistinė analizė

Studentai gavo tokią užduotį:

„Kirpykloje dirba $k=4$ kirpėjai. Klientai ateina į kirpyklą atsitiktiniais laiko momentais. Stebėjimais buvo nustatyta, kad laiko intervalai tarp tokių momentų pasiskirstę pagal eksponentinį dėsnį su vidutiniu intervalo ilgiu $\bar{\tau}=5$ min. Tais pačiais stebėjimais buvo nustatyta, kad vienu ir tuo pačiu momentu gali ateiti n klientų, kur $n=1,2,3,4$, su tikimybėmis, atitinkamai, $p(1)=0,71$, $p(2)=0,23$, $p(3)=0,05$, $p(4)=0,01$.

Atėję į kirpyklą klientai tuoj pat pradedami aptarnauti, jeigu tuo metu yra neužimtų kirpėjų, arba užima eilę. Eilės ilgis m negali būti didesnis negu $\hat{m}=8$ žmonės, todėl klientai, kurių numeris m laukiančiųjų aptarnavimo eilėje būtų didesnis, negu $\hat{m}=8$, eilės neužima, o iškart išeina iš kirpyklos.

Kiekvienas klientas su tikimybėmis, atitinkamai, $p_1=0,5$, $p_2=0,4$ ir $p_3=0,1$ gali pareikalaus vienos iš trijų aptarnavimo rūšių r .

Aptarnavimo trukmė – atsitiktinė, pasiskirsčiusi pagal eksponentinį dėsnį su vidurkiais: pirmajai aptarnavimo rūšiai $\bar{\tau}_1=12$ min., antrajai – $\bar{\tau}_2=15$ min., trečiajai – $\bar{\tau}_3=25$ min.

Reikia imituoti šiek tiek idealizuotą (supaprastintą) kirpyklos funkcionavimą, neatsižvelgiant į eilę niuansų:

- kad skirtingi kirpėjai gali dirbti nevienodai sparčiai ir nevienodai kokybiškai;
 - kad net vienas ir tas pats kirpėjas pamainos bėgyje gali dirbti nevienodai sparčiai;
 - kad kirpėjai pavargsta ir gali ilsėtis;
 - kad į kirpyklą gali skambinti telefonu, o taip pat gali atsirasti ir kitų reikalų, trumpam atitraukiančių kirpėjus nuo darbo;
 - kad apie pamainos vidurį kirpėjai gali pietauti, tačiau nebūtinai visi kartu ;
 - kad kliento aptarnavimo trukmės pasiskirstymas artimesnis normaliajam, o ne eksponentiniam pasiskirstymui;
 - kad į pamainos pabaigą kirpėjai gali paprašyti klientų nebeužiminėti eilės;
 - kad klientai gali palikti eilę, nesulaukę aptarnavimo pradžios;
 - kad gali ateiti sutartu laiku anksčiau užsirašę klientai, kuriuos kirpėjai aptarnaus be eilės;
 - kad kirpėjai gali susirgti, kad juos gali skubiai kur nors iškvieisti ir t.t.
- Imituojant kirpyklos funkcionavimą, reikia:

- generuoti atsitiktinius intervalų τ dydžius ir nustatyti atsitiktinius klientų atėjimo momentus $t^{(a)}$;
- kiekvienam tokiam momentui $t^{(a)}$ generuoti atsitiktinį klientų skaičių n ;
- nustatyti, kas atsitiks su kiekvienu atėjusiu klientu: ar bus pradėta iškart jį aptarnauti, ar jis stos į eilę, ar iškart išeis iš kirpyklos;
- kiekvienam kirpėjo j pradėtam aptarnauti klientui i generuoti aptarnavimo rūšį r bei aptarnavimo trukmę τ_{ij} ir nustatyti aptarnavimo pradžią ir pabaigą;
- nustatyti, kada kuris nors klientas stos į eilę, kada buvo pradėtas aptarnauti ir kiek jis pralaukė eilėje;
- nustatyti koks buvo eilės ilgis m po to ar kito įvykio, keičiančio kirpyklos būseną;
- nustatyti kirpėjų atidirbto laiko ir prastovų dydžius, prastovų pradžios ir baigimo momentus, apskaičiuoti kiekvieno kirpėjo atidirbtą su klientais laiką;
- nustatyti, kada ir kiek klientų paliko kirpyklą iškart (be aptarnavimo) ir kiek klientų buvo aptarnauta.

Atsižvelgiant į šiuos reikalavimus, reikia sudaryti kirpyklos funkcionavimo imitavimo algoritmo blokinę schemą, po to – imitavimo programą kompiuteriui, o taip pat imitavimo rezultatų statistinio apdorojimo ir analizės programą.

Šios programos privalo:

- imituoti kirpyklos funkcionavimą ir chronologine tvarka registruoti įvykius bei jų charakteristikas modeliavimo rezultatų protokole;
- reikalui esant, spausdinti bet kurį minėto protokolo fragmentą;
- kaupiti statistinę informaciją apie imituojamos kirpyklos veiklą, kartojant modeliavimo seansus, pavyzdžiui, modeliuojant kelių pamainų darbą;
- apskaičiuoti ir išspausdinti kirpyklos funkcionavimo statistines charakteristikas:
 - vidutinį kiekvieno kirpėjo apkrovimą $\bar{q}_j$ (procentais), t.y. tą vidutinę visos pamainos laiko dalį, kurią kirpėjas dirbo, aptarnaudamas klientus;
 - vidutinę kirpyklos praleidžiamą galią $\bar{A}$, t.y. vidutinį per vieną valandą aptarnautų klientų skaičių;
 - vidutinę vieno kliento aptarnavimo trukmę $\bar{\tau}^{(a)}$;
 - vidutinį kliento laukimą eilėje $\bar{\tau}^{(e)}$;
 - vidutinį laukiančiųjų aptarnavimo eilės ilgį $\bar{m}$;
 - vidutinį aptarnautų klientų procentą $\bar{\alpha}$, t.y., tą vidutinę visų klientų dalį, kurią sudaro aptarnauti klientai, lyginant su visų, užėjusių į kirpyklą, klientų skaičiumi.“

Papildoma užduotis:

„1. Nustatykite, kaip pasikeistų kirpyklos funkcionavimo statistinės charakteristikos (žiūr. 4a, b, c, d, e, f), jei:

- 1) kirpėjų skaičius k būtų padidintas iki 5;
- 2) kirpėjų skaičius k būtų sumažintas iki 3;
- 3) pasikeistų pasiskirstymų vidurkiai $\bar{\tau}$, $\bar{\tau}_1$, $\bar{\tau}_2$, $\bar{\tau}_3$;

4) klientų aptarnavimo trukmės vidurkiai liktų tie patys, tačiau pasikeistų pasiskirstymas: pavyzdžiui, vietoj nelabai tinkamo eksponentinio būtų panaudotas labiau atitinkantis tikrovę normalusis pasiskirstymas su vidurkiais $\bar{\tau}_1=12$ min., $\bar{\tau}_2=15$ min., $\bar{\tau}_3=25$ min. vidutiniais kvadratiniais nuokrypiais $\sigma_1=0,8$ min., $\sigma_2=1$ min., $\sigma_3=3$ min.;

5) pasikeistų maksimalus vienu metu ateinančių klientų skaičius n bei tikimybės $p(1)$, $p(2)$, ..., $p(n)$;

- 6) pasikeistų aptarnavimo rūšių skaičius $\hat{r}$ bei tikimybės $p_1, p_2, \dots, p_r, \dots, p_{\hat{r}}$.

2. Įvertinę kirpėjo darbo ir prastovų laiko vieneto kainas, kliento laukimo eilėje laiko vieneto kainą ir pelną, kurį duoda kirpyklai vieno kliento aptarnavimas, nustatykite (raskite) optimalų kirpėjų skaičių šioje kirpykloje, atsižvelgdami į tai, jog klientas, palikęs kirpyklą dėl to, kad eilė buvo per ilga, yra prarastas kirpyklos pelnas.

3. Pasiūlykite naują imitavimo algoritmą, artimesnį realioms kirpyklos funkcionavimo sąlygoms.

Visų pirma, toks algoritmas turėtų atsižvelgti į tai, jog kirpėjai negali dirbti visą pamainą be pertraukų. Patobulintas imitavimo algoritmas turėtų imituoti galimas atsitiktines ir reglamentuotas pertraukas (pavyzdžiui, pietų pertrauką).

Antra, patobulintas algoritmas turėtų atsižvelgti į tai, jog klientai, užėmę eilę, gali su tam tikra tikimybe ją palikti, jei jiems tenka laukti ilgiau, negu tikėjosi.

Trečia, reikėtų atsižvelgti į tai, jog prieš pamainos pabaigą kirpėjai gali paprašyti klientų nebeužimti eilės, kad netektų jų aptarnauti pamainai pasibaigus, o taip pat gali nebepradėti aptarnauti klientų, atėjusių prieš pačią pamainos pabaigą.

Ketvirta, būtų tikslinga, jog algoritmas numatytų tokią klientų priskyrimo kirpėjams tvarką, kad kirpėjų apkrovimas būtų tolygesnis.

Penkta, būtų naudinga atsižvelgti ir į kai kuriuos kitus niuansus, paminėtus šios užduoties pradžioje (žiūr.: a, b, c, d, e, f, g, h, i, j).

4. Nustatykite kokiais kirpyklos statistines charakteristikas galima rasti ne tik imitacinio modeliavimo metodu, bet ir analiziškai? Palyginkite charakteristikų reikšmes, rastas įvairiais metodais.

Kaip galima nustatyti mašininį eksperimentų skaičių, užtikrinantį, jog tų eksperimentų rezultatų statistinis apdorojimas leis gauti pakankamai tiksliai statistinių charakteristikų reikšmes?

Kuo skiriasi imitacinio modeliavimo metodas nuo analizinių metodų? Ką bendro turi imitacinio modeliavimo programų sudarymas su sudėtingų technologinių procesų konstravimu, kalendoriniu planavimu ir tinklinių grafikų sudarymu?“

Pagrindinę šios užduoties dalį beveik visi studentai pajėgė įvykdyti iškart. Papildomą užduoties dalį, kuri nebuvo būtina, įvykdė tik keletas studentų, tačiau kai kurie studentai, vykdydami pagrindinę užduotį, atsižvelgė ir į kai kuriuos niuansus, paminėtus papildomoje užduotyje.

Su studentų sudarytomis programomis, modeliavimo protokolais ir modeliavimo metu sukauptos informacijos statistinio apdorojimo rezultatais bei iš jų išplaukiančiomis išvadomis galima susipažinti Vilniaus universiteto Matematikos fakulteto Matematinės statistikos katedroje, todėl čia apsiribosiu tik kai kuriais studentų darbų fragmentais bei kai kuriomis pastabomis, kurios būtų naudingos skaitytojui, vykdančiam panašias užduotis. Pavaizduosiu ir kelis imitavimo ir informacijos apdorojimo algoritmų blokinių schemų variantus.

Studentų sudarytos algoritmų blokinių schemos ir programos, būdamos panašios savo sudėtimi ir paskirtimi, skyrėsi savo struktūra ir tą skirtumą sąlygojo pasirinkta modeliavimo protokolo forma bei informacijos kaupimo ir jos saugojimo kompiuteryje būdas.

Buvo studentų, kurių sudarytos programos veikė stambiais etapais, išskirtais pagal technologinį principą: pradžioje generavo ir įsiminė kirpyklos įėjimo trajektoriją

visai pamainai, po to generavo klientų aptarnavimą, t.y. generavo kiekvieno kliento aptarnavimo rūšį ir trukmę, apskaičiavo jo stovėjimo eilėje laiką, aptarnavimo pradžią ir pabaigą ir t.t., trečiame etape apdorojo sukaupią statistinę informaciją, o ketvirtajame etape spausdino modeliavimo ir statistinio apdorojimo rezultatus.

Tačiau dauguma studentų sudarė programas, kurios generavo ir protokolavo įvykius bei kaupė informaciją chronologine tvarka (žiūr. 103-ąją lentelę, kur parodytas **modeliavimo protokolo fragmentas**).

103. lentelė

Įvykio laikas	Įvykio ir sistemos būsenos charakteristika
7 ⁰⁰	Pamaina prasidėjo. Visi kirpėjai laisvi. Klientų nėra.
7 ⁰²	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)} = 1$.
7 ⁰²	Klientą Nr.1 pradėjo aptarnauti kirpėjas Nr.1. Aptarnavimo rūšis $r=2$. Aptarnavimo trukmė $\tau_{1,1}=7$ min. Aptarnavimas baigsis $t_{1,1}^{(b)}=7^{09}$. Pirmojo kirpėjo bendras atidirbtas laikas taps $T_1=7$ min.
7 ⁰³	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)} = 2$.
7 ⁰³	Klientą Nr.2 pradėjo aptarnauti kirpėjas Nr.2. Aptarnavimo rūšis $r=1$. Aptarnavimo trukmė $\tau_{2,2}=14$ min. Aptarnavimas baigsis $t_{2,2}^{(b)}=7^{17}$. Antrojo kirpėjo bendras atidirbtas laikas taps $T_2=14$ min.
7 ⁰⁹	Klientas Nr.1 po aptarnavimo palieka kirpyklą. Kirpėjas Nr.1 tampa laisvas. Bendras aptarnautų klientų skaičius $N^{(apt)} = 1$.
7 ¹⁰	Atvyko du klientai. Bendras atvykusių klientų skaičius $N^{(a)} = 4$.
7 ¹⁰	Klientą Nr.3 pradėjo aptarnauti kirpėjas Nr.3. Aptarnavimo rūšis $r=2$. Aptarnavimo trukmė $\tau_{3,3}=2$ min. Aptarnavimas baigsis $t_{3,3}^{(b)}=7^{12}$. Trečiojo kirpėjo bendras atidirbtas laikas taps $T_3=2$ min.
7 ¹⁰	Klientą Nr.4 pradėjo aptarnauti kirpėjas Nr.4. Aptarnavimo rūšis $r=3$. Aptarnavimo trukmė $\tau_{4,4}=20$ min. Aptarnavimas baigsis $t_{4,4}^{(b)}=7^{30}$. Ketvirtojo kirpėjo bendras atidirbtas laikas taps $T_4=20$ min.
7 ¹⁰	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)} = 5$.
7 ¹⁰	Klientą Nr.5 pradėjo aptarnauti kirpėjas Nr.1. Aptarnavimo rūšis $r=2$. Aptarnavimo trukmė $\tau_{5,1}=37$ min. Aptarnavimas baigsis $t_{5,1}^{(b)}=7^{47}$. Pirmojo kirpėjo bendras atidirbtas laikas taps $T_1=44$ min.
7 ¹²	Klientas Nr.3 po aptarnavimo palieka kirpyklą. Kirpėjas Nr.3 tampa laisvas. Bendras aptarnautų klientų skaičius $N^{(apt)} = 2$.
7 ¹³	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)} = 6$.
7 ¹³	Klientą Nr.6 pradėjo aptarnauti kirpėjas Nr.3. Aptarnavimo rūšis $r=2$. Aptarnavimo trukmė $\tau_{6,3}=2$ min. Aptarnavimas baigsis $t_{6,3}^{(b)}=7^{15}$. Trečiojo kirpėjo bendras atidirbtas laikas taps $T_3=4$ min.
7 ¹⁵	Klientas Nr.6 po aptarnavimo palieka kirpyklą. Kirpėjas Nr.3 tampa laisvas. Bendras aptarnautų klientų skaičius $N^{(apt)} = 3$.
7 ¹⁷	Klientas Nr.2 po aptarnavimo palieka kirpyklą. Kirpėjas Nr.2 tampa laisvas. Bendras aptarnautų klientų skaičius $N^{(apt)} = 4$.
7 ¹⁹	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)} = 7$.
7 ¹⁹	Klientą Nr.7 pradėjo aptarnauti kirpėjas Nr.3. Aptarnavimo rūšis $r=1$. Aptarnavimo trukmė $\tau_{7,3}=10$ min. Aptarnavimas baigsis $t_{7,3}^{(b)}=7^{29}$. Trečiojo kirpėjo bendras atidirbtas laikas taps $T_3=14$ min.

7 ²⁹	Klientas Nr.7 po aptarnavimo palieka kirpyklą. Kirpėjas Nr.3 tampa laisvas. Bendras aptarnautų klientų skaičius $N^{(apt)}=5$.
7 ³⁰	Klientas Nr.4 po aptarnavimo palieka kirpyklą. Kirpėjas Nr.4 tampa laisvas. Bendras aptarnautų klientų skaičius $N^{(apt)}=6$.
7 ³³	Atvyko du klientai. Bendras atvykusių klientų skaičius $N^{(a)}=9$.
7 ³³	Klientą Nr.8 pradėjo aptarnauti kirpėjas Nr.2. Aptarnavimo rūšis $r=1$. Aptarnavimo trukmė $\tau_{8,2}=3$ min. Aptarnavimas baigsis $t_{8,2}^{(b)}=7^{36}$. Antrojo kirpėjo bendras atidirbtas laikas taps $T_2=17$ min.
7 ³³	Klientą Nr.9 pradėjo aptarnauti kirpėjas Nr.3. Aptarnavimo rūšis $r=1$. Aptarnavimo trukmė $\tau_{9,3}=13$ min. Aptarnavimas baigsis $t_{9,3}^{(b)}=7^{46}$. Trečiojo kirpėjo bendras atidirbtas laikas taps $T_3=27$ min.
7 ³⁵	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)}=10$.
7 ³⁵	Klientas Nr.10 pradėjo aptarnauti ketvirtasis kirpėjas. Aptarnavimo rūšis $r=3$. Aptarnavimo trukmė $\tau_{10,4}=7$ min. Aptarnavimas baigsis $t_{10,4}^{(b)}=7^{42}$. Ketvirtojo kirpėjo bendras atidirbtas laikas taps $T_4=27$ min.
7 ³⁶	Klientas Nr.8 po aptarnavimo palieka kirpyklą. Kirpėjas Nr.2 tampa laisvas. Bendras aptarnautų klientų skaičius $N^{(apt)}=7$.
7 ⁴¹	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)}=11$.
7 ⁴¹	Klientą Nr.11 pradėjo aptarnauti kirpėjas Nr.2. Aptarnavimo rūšis $r=2$. Aptarnavimo trukmė $\tau_{11,2}=14$ min. Aptarnavimas baigsis $t_{11,2}^{(b)}=7^{55}$. Antrojo kirpėjo bendras atidirbtas laikas taps $T_2=31$ min.
7 ⁴¹	Atvyko vienas klientas. Bendras atvykusių klientų skaičius $N^{(a)}=12$.
7 ⁴¹	Klientas Nr.12 stoja į laukiančiųjų aptarnavimo eilę. Eilės ilgis tampa $m=1$.

Ši protokolo forma yra gana patogi skaitytojui, tačiau turi trūkumą, jog ja naudojantis, reikia spausdinti ir po to skaityti daug žodžių. Todėl kai kurie studentai pabandė šią formą struktūrizuoti.

Buvo išskirti 9 įvykių tipai:

1. Pamainos pradžia.
2. Klientų atėjimas į kirpyklą.
3. Kliento aptarnavimo pradžia.
4. Kliento stojimas į eilę.
5. Kliento aptarnavimo pabaiga ir jo išėjimas iš kirpyklos.
6. Kliento išėjimas iš eilės.
7. Neaptarnauto kliento išėjimas iš kirpyklos.
8. Oficialioji pamainos pabaiga.
9. Faktiškoji pamainos pabaiga.

Kiekvienam įvykių tipui buvo sudarytas specialus **protokolo eilutės formatas**.

Pirmojo tipo formatas:

t	1
---	---

 *, kur $t=t^{(p,p)}$ – laikas, atitinkantis pamainos pradžią, pavyzdžiui, 7 val. 00 min., „1“ – įvykio tipo numeris, * – eilutės pabaigos ženklas.

Kai protokole spausdinami žodžiai, tai vietoj tipo numerio „1“ turi būti spausdinama frazė: „Prasidėjo pamaina“.

Antrojo tipo formatas:

t	2	n	$N^{(a)}$
---	---	---	-----------

 *, kur $t=t^{(a)}$ – klientų atvykimo laikas, „2“ – įvykio tipo numeris, n – atėjusių momentu $t^{(a)}$ klientų skaičius, $N^{(a)}$ – visų atėjusių į kirpyklą iki momento $t^{(a)}$ imtinai klientų skaičius.

Kai protokole spausdinami ne tik skaičiai, bet ir frazės, sudarytos iš žodžių, tai vietoj skaičių 2, n, N, spausdinama frazė „Atvyko n klientų. Bendras atvykusių klientų skaičius $N^{(a)}$ “.

Trečiojo tipo formatas:

t	3	N_i	j	r	τ_{ij}	$t_{ij}^{(b)}$	T_j
---	---	-------	---	---	-------------	----------------	-------

 *, kur $t=t^{(apt)}$ – aptarnavimo pradžios laikas, „3“ – įvykio tipo numeris. N_i – aptarnaujamo kliento numeris, j – aptarnaujančio kirpėjo numeris, r – aptarnavimo rūšies numeris, τ_{ij} – aptarnavimo trukmė (minutėmis), $t_{ij}^{(b)}$ – aptarnavimo pabaigos laikas, T_j – kirpėjo j nuo pamainos pradžios atidirbtas laikas.

„Žodiniame“ protokole vietoj skaičių 3, N_i , j, r, τ_{ij} , $t_{ij}^{(b)}$, T_j spausdinamos frazės: „Klientą numeris N_i pradėjo aptarnauti kirpėjas numeris j. Aptarnavimo rūšis r. Aptarnavimo trukmė τ_{ij} min. Aptarnavimas baigsis $t_{ij}^{(b)}$ ir kirpėjo j nuo pamainos pradžios atidirbtas laikas taps T_j “.

Ketvirtojo tipo formatas:

t	4	N_i	m
---	---	-------	---

 *, kur $t=t^{(s)}$ – kliento stojimo į eilę laikas, „4“ – įvykio tipo numeris, N_i – kliento numeris, m – eilės ilgis po to, kai klientas stojo į eilę.

„Žodiniame“ protokole vietoj skaičių 4, N_i , m spausdinama: „Klientas numeris N_i stojo į eilę. Eilės ilgis tampa m.“.

Penktojo tipo formatas:

t	5	N_i	j	$N^{(apt)}$
---	---	-------	---	-------------

 *, kur $t=t^{(b)}$ – kliento N_i aptarnavimo baigimo laikas, „5“ – įvykio tipo numeris, j – kirpėjo, baigusio aptarnavimą, numeris, $N^{(apt)}$ – bendras aptarnautų klientų skaičius.

„Žodiniame“ protokole vietoj skaičių 5, N_i , j, $N^{(apt)}$ spausdinama: „Klientas numeris N_i , kurį aptarnavo kirpėjas j, po aptarnavimo palieka kirpyklą. Kirpėjas j tampa laisvas. Bendras aptarnautų klientų skaičius yra $N^{(apt)}$ “.

Šeštojo tipo formatas:

t	6	N_i	$\tau_i^{(e)}$	m	$T^{(e)}$	$N^{(s)}$
---	---	-------	----------------	---	-----------	-----------

 *, kur $t=t^{(e)}$ – laiko momentas, kuriuo klientas N_i išeina iš eilės, „6“ – įvykio tipo numeris, $\tau_i^{(e)}$ – laiko intervalas, kurį klientas praleido eilėje (minutėmis), m – eilės ilgis tuoj pat po to, kai klientas N_i paliko eilę, $T^{(e)}$ – bendras laikas, kurį visi klientai prastovėjo eilėje nuo pamainos pradžios iki momento t, kada klientas N_i paliko eilę (minutėmis), $N^{(s)}$ – bendras visų jau stovėjusių eilėje klientų skaičius.

„Žodiniame“ protokole vietoj skaičių 6, N_i , $\tau_i^{(e)}$, m, $T^{(e)}$, $N^{(s)}$ spausdinama: „Klientas N_i išeina iš eilės. Jis eilėje prastovėjo $\tau^{(e)}$ min. Po išėjimo eilės ilgis tapo m. Bendras visų klientų iki šio momento prastovėtas eilėje laikas yra $T^{(e)}$. Bendras stovėjusių eilėje klientų skaičius tapo $N^{(s)}$ “.

Septintojo tipo formatas:

t	7	N_i	$N^{(is)}$
---	---	-------	------------

 *, kur $t=t^{(is)}$ – laiko momentas, kuriuo neaptarnautas klientas N_i palieka kirpyklą, „7“ – įvykio tipo numeris, $N^{(is)}$ – bendras išėjusių be aptarnavimo klientų skaičius (nuo pamainos pradžios iki momento t imtinai).

„Žodiniame“ protokole vietoj skaičių 7, N_i , $N^{(is)}$ spausdinama: „Neaptarnautas klientas N_i palieka kirpyklą. Bendras išėjusių iš kirpyklos be aptarnavimo klientų skaičius tampa $N^{(is)}$ “.

Aštuntojo tipo formatas:

t	8
---	---

 *, kur $t=t^{(o.p)}$ – pamainos oficialiosios pabaigos laikas, „8“ – įvykio tipo numeris.

„Žodiniame“ protokole vietoj skaičiaus „8“ spausdinama: „Pamaina formaliai pasibaigė“.

Devintojo tipo formatas:

t	9
---	---

 *, kur $t=t^{(f.p)}$ – faktiškasis pamainos pabaigos laikas, „9“ – įvykio tipo numeris.

„Žodiniame“ protokole vietoj skaičiaus „9“ spausdinama: „Pamaina faktiškai pasibaigė“.

Naudojantis įvestais formatais ir žymėjimais, imitavimo protokolą, pavaizduotą 103-oje lentelėje, galima padaryti kompaktiškesnį (žiūr. 104 lentelę).

104. lentelė

Įvykio laikas	Įvykio tipas	Įvykio ir sistemos būsenos charakteristika
7 ⁰⁰	1*	
7 ⁰²	2	1; 1*
7 ⁰²	3	1; 1; 2; 7; 7 ⁰⁹ ; 7*
7 ⁰³	2	1; 2*
7 ⁰³	3	2; 2; 1; 14; 7 ¹⁷ ; 14*
7 ⁰⁹	5	1; 1; 1*
7 ¹⁰	2	2; 4*
7 ¹⁰	3	3; 3; 2; 2; 7 ¹² ; 2*
7 ¹⁰	3	4; 4; 3; 20; 7 ³⁰ ; 20*
7 ¹⁰	2	1; 5*
7 ¹⁰	3	5; 1; 2; 37; 7 ⁴⁷ ; 44*
7 ¹²	5	3; 3; 2*
7 ¹³	2	1; 6*
7 ¹³	3	6; 3; 2; 2; 7 ¹⁵ ; 4*
7 ¹⁵	5	6; 3; 3*
7 ¹⁷	5	2; 2; 4*
7 ¹⁹	2	1; 7*
7 ¹⁹	3	7; 3; 1; 10; 7 ²⁹ ; 14*
7 ²⁹	5	7; 3; 5*
7 ³⁰	5	4; 4; 6*
7 ³³	2	2; 9*
7 ³³	3	8; 2; 1; 3; 7 ³⁶ ; 17*
7 ³³	3	9; 3; 1; 13; 7 ⁴⁶ ; 27*
7 ³⁵	2	1; 10*
7 ³⁵	3	10; 4; 3; 7; 7 ⁴² ; 27*
7 ³⁶	5	8; 2; 7*
7 ⁴¹	2	1; 11*
7 ⁴¹	3	11; 2; 2; 14; 7 ⁵⁵ ; 31*
7 ⁴¹	2	1; 12*
7 ⁴¹	4	12; 1*

Protokolas tapo kompaktiškas, tačiau iškilo kitas nepatogumas: norint jį iššifruoti, reikia įsiminti formatus arba turėti juos prieš akis. Šį trūkumą galima sušvelninti, spausdinant protokolo eilutėse ne tik skaičius (charakteristikų reikšmes), bet ir charakteristikų identifikatorius (žiūr. 105 lentelę).

105. lentelė

Įvykio laikas	Įvykio tipas	Įvykio ir sistemos būsenos charakteristika
7 ⁰⁰	1*	
7 ⁰²	2	n=1; N ^(a) =1 *
7 ⁰²	3	N _i =1; j=1; r=2; τ _{1,1} =7; t _{1,1} ^(b) =7 ⁰⁹ ; T ₁ =7 *

7^{03}	2	$n=1; N^{(a)}=2 *$
7^{03}	3	$N_i=2; j=2; r=1; \tau_{2,2}=14; t_{2,2}^{(b)}=7^{17}; T_2=14 *$
7^{09}	5	$N_i=1; j=1; N^{(apt)}=1 *$
7^{10}	2	$n=2; N^{(a)}=4 *$
7^{10}	3	$N_i=3; j=3; r=2; \tau_{3,3}=2; t_{3,3}^{(b)}=7^{12}; T_3=2 *$
7^{10}	3	$N_i=4; j=4; r=3; \tau_{4,4}=20; t_{4,4}^{(b)}=7^{30}; T_4=20 *$
7^{10}	2	$n=1; N^{(a)}=5 *$
7^{10}	3	$N_i=5; j=1; r=2; \tau_{5,1}=37; t_{5,1}^{(b)}=7^{47}; T_1=44 *$
7^{12}	5	$N_i=3; j=3; N^{(apt)}=2 *$
7^{13}	2	$n=1; N^{(a)}=6 *$
7^{13}	3	$N_i=6; j=3; r=2; \tau_{6,3}=2; t_{6,3}^{(b)}=7^{15}; T_3=4 *$
7^{15}	5	$N_i=6; j=3; N^{(apt)}=3 *$
7^{17}	5	$N_i=2; j=2; N^{(apt)}=4 *$
7^{19}	2	$n=1; N^{(a)}=7 *$
7^{19}	3	$N_i=7; j=3; r=1; \tau_{7,3}=10; t_{7,3}^{(b)}=7^{29}; T_3=14 *$
7^{29}	5	$N_i=7; j=3; N^{(apt)}=5 *$
7^{30}	5	$N_i=4; j=4; N^{(apt)}=6 *$
7^{33}	2	$n=2; N^{(a)}=9 *$
7^{33}	3	$N_i=8; j=2; r=1; \tau_{8,2}=3; t_{8,2}^{(b)}=7^{36}; T_2=17 *$
7^{33}	3	$N_i=9; j=3; r=1; \tau_{9,3}=13; t_{9,3}^{(b)}=7^{46}; T_3=27 *$
7^{35}	2	$n=1; N^{(a)}=10 *$
7^{35}	3	$N_i=10; j=4; r=3; \tau_{10,4}=7; t_{10,4}^{(b)}=7^{42}; T_4=27 *$
7^{36}	5	$N_i=8; j=2; N^{(apt)}=7 *$
7^{41}	2	$n=1; N^{(a)}=11 *$
7^{41}	3	$N_i=11; j=2; r=2; \tau_{11,2}=14; t_{11,2}^{(b)}=7^{55}; T_2=31 *$
7^{41}	2	$n=1; N^{(a)}=12 *$
7^{41}	4	$N_i=12; m=1 *$

Šitoki protokolą skaityti jau gana paprasta.

Kaip jau buvo minėta anksčiau, studentų sudarytų algoritmų ir programų blokinės schemos skyrėsi savo struktūra, būdamos panašios savo sudėtimi.

Panagrinėsime vieną iš studentų sudarytų blokinių schemų tokio imitavimo algoritmo, kurio organizacija įdomi tuo, jog ji gana artima realaus imituojamo proceso organizacijai: **kompiuterio programa, imituodama kirpyklos funkcionavimą, modeliavo ir kirpyklos elementus – kirpyklos įėjimą, klientus, kirpėjų darbo vietas, klientų eilę, kirpyklos išėjimą, išskirdama tiems elementams kompiuterio atmintyje atitinkamus „darbo laukus“, kuriuose buvo užrašomos besikeičiančios elementų charakteristikos** (žiūr. 304 pav.).

Nagrinėjamoje programoje įvykių registravimui ir kai kurių charakteristikų skaičiavimui buvo organizuoti specialūs **skaitliukai**:

1) bendro atvykusių klientų skaičiaus $N^{(a)}$ skaitliukas, kurio veikimo algoritmą išreiškia formulė $N^{(a)} := N^{(a)} + n$, reiškianti, jog nauja to skaitliuko reikšmė gaunama pridėjus prie senosios jo reikšmės ką tik atvykusių klientų skaičių n ;

2) ką tik atvykusių, bet dar neregistruotų ir dar nepradėtų aptarnauti klientų skaitliukas $n:=n-1$; į šį skaitliuką kiekvieną kartą įrašomas atvykusių klientų skaičius n , o po to, registruojant (pradedant aptarnauti) kiekvieną klientą, iš to skaitliuko turinio atimama po 1;

3) klientų registravimo ir numerių N_i jiems priskyrimo skaitliukas $N_i:=N_i+1$;

4) bendro aptarnautų klientų skaičiaus $N^{(apt)}$ skaitliukas $N^{(apt)} := N^{(apt)} + 1$;

5) kiekvieno kirpėjo j ($j=1, 2, \dots, k$) nuo pamainos pradžios atidirbto laiko T_j skaitliukas $T_j := T_j + \tau_{ij}$;

6) eilės ilgio m skaitliukas: jo reikšmė apskaičiuojama naudojantis formule $m := m+1$, kai klientas stoja į eilę, ir $m := m-1$, kai klientas išeina iš eilės;

7) klientų, stovėjusių eilėje, bendro skaičiaus $N^{(s)}$ skaitliukas $N^{(s)} := N^{(s)} + 1$;

8) visų klientų prastovėto eilėje bendro laiko $T^{(e)}$ skaitliukas $T^{(e)} := T^{(e)} + \tau_i^{(e)}$;

9) išvykusių neaparnautų klientų bendro skaičiaus $N^{(is)}$ skaitliukas $N^{(is)} := N^{(is)} + 1$;

10) imituojamų pamainų p skaitliukas $p := p+1$, skaičiuojantis nuo 0 iki $\hat{p}$, kur $\hat{p} \geq 1$ yra pamainų skaičius, kurias reikia imituoti.

11) Skaitliukų skaičių galima buvo sumažinti, nes, pavyzdžiui, turint skaitliuką N_i , buvo galima neturėti skaitliuko $N^{(a)}$, o žinant $N^{(a)}$ ir $N^{(apt)}$, galima buvo neskaičiuoti $N^{(is)}$, o surasti šį skaičių pamainos gale, naudojantis lygybe $N^{(is)} := N^{(a)} - N^{(apt)}$, tačiau programos autorius perteklinių skaitliukų organizavimą paaikškino taip: juos organizuoti labai paprasta, o jų buvimas padeda patikrinti, ar imitavimo metu nebuvo trikdžių, kas yra aktualu, kai imitavimas trunka ilgai.

Be šių skaitliukų programoje buvo sudarytas dar vienas specialus skaitliukas – registras, skirtas imituojamo laiko skaičiavimui ir registravimui, pavadintas **taimeriu** („laikrodžiu“).

Kai imituojamas laikas keičiasi tolydžiai mažais ir vienodais žingsneliais – kvantais Δt (kaip, pavyzdžiui, Dž. Foresterio modeliuose, su kuriais mes susidursime antrojoje šios mokymo priemonės dalyje), tai taimeriu gali būti paprastas skaitliukas $t := t + \Delta t$. Kirpyklos funkcionavimo atveju irgi buvo galima naudotis tokiu taimeriu, tačiau tai būtų buvę neekonomiška kompiuterio laiko panaudojimo atžvilgiu. Todėl programos autorius naudojo kitokios konstrukcijos **taimeriu, keičiančiu savo laiką stambiais šuoliais nuo vieno įvykio įvykimo iki kito**.

Tokiu taimeriu imituojamas ir vaizduojamas laikas keičiasi šuoliu, išskvietus iš **būsimųjų įvykių grandinės (BĪG)** eilinių būsimųjų įvykių, kuris turi įvykti iš BĪG išskviestame ir tą įvykį aprašančiame įrašė nurodytu laiku. Tas laikas **perrašomas į taimerį ir tampa imituojamu laiko momentu**, kuris nesikeičia, kol neįvykdomos visos tą įvykį imituojančios kompiuterinės operacijos. Tai reiškia, jog **imituojamas laikas ir fizinis kompiuterio darbo laikas**, kuris tolydžiai didėja imitavimo metu, sutapdamas su astronominiu laiku, – **du skirtingi dalykai**, kurių kitimo tempai gali būti visiškai skirtingi: pirmojo kitimas yra valdomas kompiuterio programos ir gali būti kiek norima pagreitintas ar sulėtintas, net gi gali būti pakeista jo kitimo kryptis, o antrojo tėkmė yra natūrali ir objektyvi laiko tėkmė, nepriklausanti nei nuo kompiuterio, nei nuo programuotojo bei eksperimentuotojo.

Būsimųjų įvykių grandinė (BĪG) nagrinėjamoje programoje vadinama specialiai išskirta kompiuterio operatyvinės atminties zona, kurioje išimenuama informacija apie tuos antrojo, penktojo ir aštuntojo tipų įvykius, kurie bus imituojami artimiausiu laiku: apie klientų, kurie turi neužilgo atvykti į kirpyklą, atvykimo laiką $t^{(a)}$, apie kai kurių jau atvykusių ir aptarnaujamų klientų aptarnavimo laukiamą pabaigą $t_{ij}^{(b)}$; apie būsimą oficialią pamainos pabaigą $t^{(o.p.)}$.

Ši atminties zona vadinama **grandine** todėl, kad joje informacija apie įvykius išrikiuota tam tikra tvarka, ir įvykiai sudaro savo rūšies grandinę. Kai būsimieji įvykiai turi įvykti ne vienu metu, tai jie išrikiuojami įvykių įvykimo laiko didėjimo tvarka. Kai keli įvykiai turėtų įvykti tuo pačiu metu, tai pirmenybė (prioritetas) pirmoje eilėje suteikiama penktojo tipo įvykiams. Tai ir suprantama: jei vienu metu atvyksta naujas klientas ir baigiasi kokio nors kliento aptarnavimas, tai, prieš pradedant spręsti, ką daryti su naujai atvykusiu klientu (ar pradėti jį aptarnauti, ar statyti į eilę, ar iš viso nepriimti), tikslinga išleisti iš kirpyklos jau aptarnautą klientą ir tokiu būdu organizuoti laisvą vietą naujų klientų aptarnavimui. Jeigu būtų naudojama atvirkštinė tvarka, tai kartais galėtų atsitikti net taip, kad ką tik atvykęs klientas būtų išsiunčiamas iš kirpyklos be aptarnavimo, kai tuo tarpu, kaip tik tuo metu atsilaivina kirpėjas, kuris galėtų pradėti aptarnauti tą klientą.

Nesunku suprasti, kad tais atvejais, kai vienu metu turėtų įvykti antrojo ir aštuntojo tipų įvykiai, pirmenybė atiduodama aštuntojo tipo įvykiui: pamaina oficialiai pasibaigia ir nauji klientai nebepriimami, nes kitaip jų aptarnavimas turėtų prasidėti jau pamainai pasibaigus.

Kyla klausimas, kaip rikiuojami to paties tipo įvykiai grandinėje, kai keli tokie įvykiai turi įvykti tuo pačiu momentu. Tokia situacija negalima antrojo ir aštuntojo tipų įvykiams, nes kiekvienu momentu grandinėje gali būti ne daugiau kaip po vieną šių tipų įvykių. Todėl belieka apsispręsti, kaip elgtis su penktojo tipo įvykiais, turinčiais įvykti tuo pačiu metu. Tuo atveju, kai mums neaktualu vienodinti visų kirpėjų apkrovimą, vienu metu turinčius įvykti penktojo tipo įvykius galima rikiuoti kirpėjų numerių j didėjimo tvarka. Esant tokiai rikiavimo tvarkai, dažniau būtų apkrautas pirmasis kirpėjas, šiek tiek mažiau – antrasis ir t.t., tačiau tokia tvarka kirpėjams gali būti nepageidautina: kai kirpėjai dirba individualiai, didžiausias uždarbis tenka pirmajam kirpėjui, o mažiausias – k-jam, ir kirpėjai turintys didesnius numerius, gali likti nepatenkinti; kai kirpėjai dirba brigadiniu metodu ir gauna vienodus atlyginimus, nepatenkintais gali tapti kirpėjai, turintys pirmuosius numerius. Todėl kai kurie

studentai, norėdami išlyginti kirpėjų apkrovimą, naudojosi tokia 5-jo tipo vienu metu įvyksiančių įvykių rikiavimo BĮG tvarka: įvykiai buvo rikiuojami T_j didėjimo tvarka, ir tik tuo metu, kai keli 5-jo tipo įvykiai, turėdami įvykti vienu metu, turėjo vienodus T_j , jie buvo rikiuojami j didėjimo tvarka. Tai leido apytiksliai suvienodinti kirpėjų apkrovimą.

Reikia pažymėti, kad tokia **5-jo tipo įvykių rikiavimo būsimųjų įvykių grandinėje tvarka yra ne imitavimo algoritmo, o kirpyklos funkcionavimo (joje įvestos tvarkos) savybė, ir programuotojas, imituodamas konkrečios (o ne įsivaizduojamos) kirpyklos veiklą, turėtų įvykius grandinėje rikiuoti taip, kad tai atitiktų kirpykloje vykstančius procesus.**

Jei, pavyzdžiui, kirpykloje būtų priimta tokia tvarka, kad klientas, atėjęs į kirpyklą, baigiantis paskutiniajai pamainos sekunde, būtų priimamas ir aptarnaujamas, tektų keisti vienu metu įvykstančių antrojo ir aštuntojo tipo įvykių prioritetus grandinėje: pirmenybė tuomet turėtų būti suteikta antrojo tipo įvykiui.

Aprašę įvykių rikiavimo BĮG tvarką, pereisime prie jų **įrašymo į BĮG tvarkos** aprašymo.

Nagrinėjamoje programoje antrojo tipo įvykiai buvo įrašomi į BĮG penktajame bloke (žiūr. 304 pav.) po kiekvieno τ , vaizduojančio laiko intervalą nuo einamojo momento iki artimiausio naujų klientų atvykimo į kirpyklą, reikšmės generavimo. Sugeneravus τ , apskaičiuojamas būsimasis klientų atvykimo laikas $t^{(a)} = t + \tau$ ir į BĮG įrašomas įrašas: 2; $t^{(a)}$ *.

Penktojo tipo įvykiai įrašomi į BĮG kiekvieną kartą po to, kai eiliniam klientui, prieš pradėdam jo aptarnavimą, nustatoma to aptarnavimo rūšis r , generuojama to aptarnavimo trukmė τ_{ij} ir apskaičiuojamas aptarnavimo baigimo laikas $t_{ij}^{(b)} = t + \tau_{ij}$ bei kirpėjo atidirbtas laikas T_j nuo pamainos pradžios iki šio kliento aptarnavimo pabaigos: $T_j = T_j + \tau_{ij}$.

Reikšmė T_j surandama, operuojant su skaitliuku T_j , ir įrašoma į jį, o į BĮG įrašomas įrašas: 5; $t_{ij}^{(b)}$; N_i ; j ; T_j *.

Aštuntojo tipo įvykis – oficialioji pamainos pabaiga ir jos laikas $t^{(o,p)}$, t.y. 8; $t^{(o,p)}$ * – nagrinėjamoje programoje buvo įrašomas į BĮG imitavimo pradžioje, įrašant pradines programos parametrų reikšmes, t.y. ketvirtajame programos bloke, ir prabūdavo toje grandinėje iki momento $t = t^{(o,p)}$, t.y., iki tol, kol tas įvykis nebūdavo iškviečiamas iš BĮG ir įvykdavo oficiali pamainos pabaiga.

Dabar pereisime prie **įvykių iškvietimo iš BĮG** aprašymo.

Įrašyti į BĮG įvykiai nagrinėjamoje programoje buvo iškviečiami iš grandinės specialiaame bloke (tas blokas 304 pav. pažymėtas šeštuoju numeriu) **po vieną eilės tvarka nuo grandinės pradžios**. Iškviestus eilinių įvykių iš BĮG, buvo nustatoma nauja taimerio reikšmė t , įrašant į taimerį iškviesto iš BĮG būsimąjo įvykio įvykimo laiką (retkarčiais naujoji taimerio reikšmė galėjo ir sutapti su senąja), o pats iškviestas iš BĮG įrašas, charakterizuojantis būsimąjį įvykį, nagrinėjamoje programoje buvo perduodamas į specialų analizės ir valdymo bloką (304 pav. tas blokas pažymėtas septintuoju numeriu).

Priklausomai nuo iškviesto iš BĮG įvykio tipo programoje valdymas būdavo perduodamas vienai iš trijų galimų krypčių.

Kai iškviestasis įrašas priklausydavo antrajam tipui, tai valdymas būdavo perduodamas aštuntajam ir t.t. blokams, kuriuose būdavo generuojamas atvykusių klientų skaičius n ir sprendžiama, kaip su tais klientais elgtis: ar pradėti juos aptarnauti, t.y. generuoti r , τ_{ij} ir formuoti trečiojo tipo įrašą į protokolą, o po to – ir penktojo tipo įrašus į BĮG; ar statyti į eilę ir formuoti ketvirtojo tipo įrašą į imitavimo protokolą; ar iš viso išleisti klientą iš kirpyklos be aptarnavimo, suformavus ir įrašius į protokolą septintojo tipo įrašą.

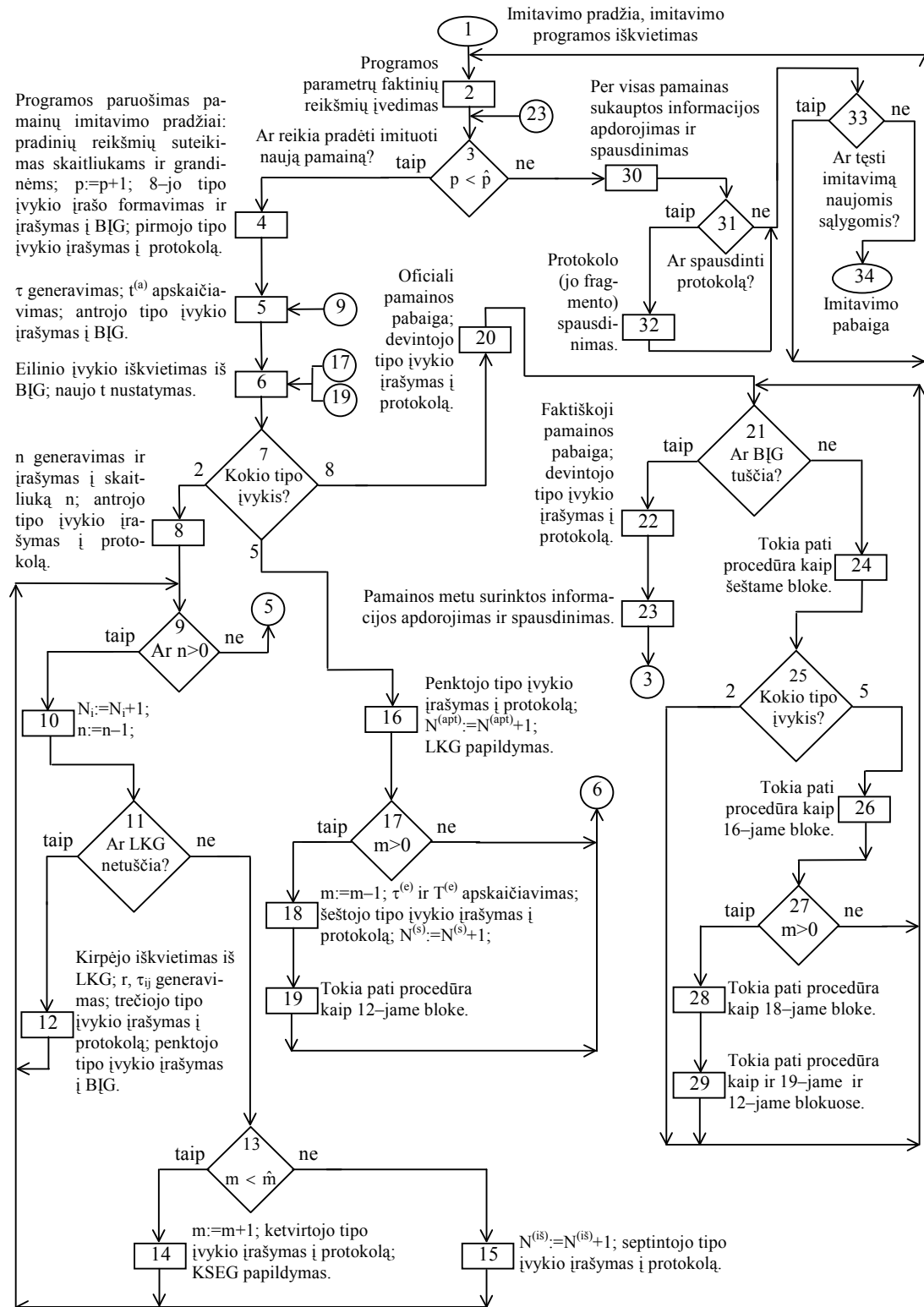
Kai iškviestasis iš BĮG įrašas priklausydavo penktajam tipui, valdymas būdavo perduodamas šešioliktajam (žiūr. 304 pav.) ir t.t. blokams, kur būdavo formuojamas penktojo tipo įrašas į protokolą ir, jei eilės ilgis m būdavo didesnis už 0, klientas iš eilės būdavo iškviečiamas aptarnavimui ir generuojamos r ir τ_{ij} reikšmės bei formuojamas naujas penktojo tipo įrašas į BĮG su po to sekančiu eilinio būsimąjo įvykio iškvietimu iš BĮG, arba iškart (jei $m=0$) buvo grįžtama į šeštąjį bloką ir iškviečiamas iš BĮG eilinis būsimasis įvykis.

Kai iškviestasis iš BĮG įvykis priklausydavo aštuntajam tipui, valdymas būdavo perduodamas į dvidešimtąjį bloką, kur prasidėdavo aštuntojo tipo įrašo į protokolą formavimas, pamainos faktiškosios pabaigos organizavimas ir sukauptos statistinės informacijos apdorojimas, apskaičiuojant $\bar{A}$, $\bar{\tau}^{(apt)}$, $\bar{\tau}^{(e)}$, $\bar{m}$, $\bar{a}$.

Nagrinėjamoje programoje čia aprašytoji BĮG, būdama vienu iš svarbiausių programos komponentų, nėra vienintelė grandinė. Yra dar laisvų (t.y. šiuo metu neaptarnaujančių klientus) kirpėjų grandinė ir klientų, stovinčių eilėje, laukiant aptarnavimo, grandinė.

Laisvų kirpėjų grandinėje (LKG) kirpėjai gali būti rikiuojami jų numerių j didėjimo tvarka arba atidirbtos laiko T_j didėjimo tvarka (kaip tai buvo padaryta nagrinėjamoje programoje) ir tik tuomet, kai T_j vienodi, jų numerių j didėjimo tvarka.

Pamainos imitavimo pradžioje visi kirpėjai būna išrikiuoti LKG numerių j didėjimo tvarka ir išskviečiami iš grandinės, atsiradus neaptarnautiems klientams, formuojant trečiojo tipo įrašą į protokolą. Kirpėjai sugražinami į LKG po kliento aptarnavimo pabaigos, t.y. formuojant penktojo tipo įrašą į protokolą.



304 pav.

Klientų, stovinių eilėje, grandinė (KSEG) formuojasi statant klientus į eilę, t.y. įrašant į KSEG stojusiojo į eilę kliento numerį N_i ir stojimo į eilę laiką bei formuojant ketvirtojo tipo įrašą į

protokolą, ir modeliuoja tą eilę. Klientai KSEG rikiuojami jų stojimo į eilę tvarka ir palieka ją, pereinami į aptarnavimą, t.y. formuojant šeštojo, o po to ir trečiojo tipo įrašus į protokolą.

Dabar jau galima pereiti prie nagrinėjamos programos algoritmo blokinės schemos (žiūr. 304 pav.) nuoseklaus aprašymo.

Pirmasis šios blokinės schemos blokas iškviečia anksčiau sudarytą ir kompiuterio atmintyje (diske arba diskelyje) saugomą kirpyklos imitavimo bendrąją programą. Ši procedūra atliekama interaktyviame režime.

Antrasis blokas interaktyviame režime suteikia iškviestos imitavimo programos formaliems parametrų faktines (konkrečias) reikšmes, pageidautinas tam asmeniui, kuris vykdo matematinius eksperimentus su imitaciniu modeliu. Įrašomi šie dydžiai, kurie kiekvieną kartą (kreipiantis į antrąjį bloką iš pirmojo ar 33-jo bloko) gali būti keičiami:

- 1) imituotinių pamainų p skaičius $\hat{p}$;
- 2) kirpėjų skaičius k ;
- 3) vienu metu galinčių atvykti klientų skaičiaus n maksimumas $\hat{n}$;
- 4) atsitiktinio dydžio n (vienu metu galinčių atvykti klientų skaičiaus) tikimybės $p(1), p(2), \dots, p(n), \dots, p(\hat{n})$;
- 5) intervalų ilgio τ tarp klientų atvykimo į kirpyklą gretimų momentų skirstinio tikimybinius tankius $f(\tau)$ ir jo parametrų reikšmės (pavyzdžiui $\exp, \lambda$ arba $\bar{\tau} = \frac{1}{\lambda}$), kurie reikalingi atsitiktinio dydžio τ generavimo atitinkamos standartinės paprogramės iškvietimui ir priderinimui;
- 6) aptarnavimo rūšių r maksimalus skaičius $\hat{r}$;
- 7) aptarnavimo rūšių numerio r (kaip diskrečiai pasiskirsčiusio atsitiktinio dydžio) tikimybės $p_1, p_2, \dots, p_r, \dots, p_{\hat{r}}$;
- 8) aptarnavimo trukmės τ_{ij} skirstinio tikimybinių tankių $f_i(\tau)$ identifikatoriai (žymėjimai ar pavadinimai) ir tuos tankius charakterizuojančių parametrų reikšmės (pavyzdžiui, $N(0;1); N(\bar{a};\sigma)$, nurodant $\bar{a}$ ir σ reikšmes; $\exp$, nurodant $\bar{\tau}$ ar λ reikšmę ir pan.), kas reikalinga atitinkamų standartinių paprogramių iškvietimui ir priderinimui;
- 9) pradinę pamainų skaitliuko p reikšmę $p=0$;
- 10) pamainų pradžios ir oficialiosios pabaigos normatyvinės reikšmės $t^{(p,p)}$ ir $t^{(o,p)}$;
- 11) specialios nuorodos (pavyzdžiui, spausdinti ar nespausdinti imitavimo eigos protokolą bei jo fragmentus ir pan.).

Trečiasis blokas dirba jau automatiniam režimui ir tikrina ar $p < \hat{p}$, t. y. tikrina, ar jau ne paskutinė pamaina iš tų, kurias reikėjo imituoti. Kai $p < \hat{p}$, tai valdymas perduodamas į ketvirtąjį bloką ir pradedama imituoti nauja pamaina. Kai į klausimą „ $p < \hat{p}$?“ gaunamas neigiamas atsakymas, tai valdymas perduodamas į 30-jį bloką ir pamainų imitavimas nebetešiamas, o pradedama apdoroti per imituotas pamainas sukaupta statistinė informacija.

Ketvirtasis blokas paruošia programą pamainų imitavimo pradžiai – imitavimo modelį perveda į pradinę būseną, įvykdydamas operaciją $p := p+1$ ir suteikdamas pradines reikšmes šiems skaitliukams:

- 1) imituojamo laiko t skaitliukui: $t := t^{(p,p)} = t_0$, kur $t^{(p,p)}$ – normatyvinis pamainos pradžios laikas įrašytas į kompiuterio atmintį antrajame bloke;
- 2) vienu metu atvykusių ir dar nepradėtų aptarnauti klientų skaitliukui: $n := 0$;
- 3) bendro atvykusiųjų į kirpyklą klientų skaičiaus $N^{(a)}$ skaitliukui: $N^{(a)} := 0$;
- 4) bendro aptarnautų klientų skaičiaus $N^{(apt)}$ skaitliukui: $N^{(apt)} := 0$;
- 5) atvykusių klientų numeravimo skaitliukui: $N_i := 0$;
- 6) bendro išvykusių be aptarnavimo klientų skaičiaus $N^{(is)}$ skaitliukui: $N^{(is)} := 0$;
- 7) bendro stovėjusių eilėje klientų skaičiaus $N^{(s)}$ skaitliukui: $N^{(s)} := 0$;
- 8) kiekvieno kirpėjo atidirbto laiko T_j skaitliukams: $T_1 := 0, T_2 := 0, \dots, T_j := 0, \dots, T_k := 0$;
- 9) visų klientų prastovėto eilėje laiko sumos $T^{(e)}$ skaitliukui: $T^{(e)} := 0$;
- 10) eilės ilgio m skaitliukui: $m := 0$.

Be to ketvirtasis blokas paruošia kompiuterio atmintyje vietas $\bar{q}_j, \bar{A}, \bar{\tau}^{(a)}, \bar{\tau}^{(e)}, \bar{m}, \bar{a}$ užrašymui, išvalo BĮG bei KSEG ir surašo visus kirpėjus į LKG, išrikiuodamas juos į didėjimo tvarką. Į išvalytą BĮG įrašomas aštuntojo tipo įvykis.

Ketvirtasis blokas (kaip trečiasis ir visi kiti blokai iki 30-jo imtinai) dirba jau automatiniam režimui, tačiau nagrinėjamoji programa leidžia organizuoti ketvirtojo bloko darbą ir interaktyviame režime.

Penktasis blokas, pasitelkdamas standartinę atsitiktinių (pseudoatsitiktinių) dydžių, priklausančių skirstiniui su antrame bloke įrašytu tikimybinio tankiu $f(\tau)$, generavimo programą, generuoja eilinę τ reikšmę ir apskaičiuoja artimiausią klientų atvykimo laiką $t^{(a)}=t+\tau$, po to įrašydamas į BĮG įrašą „ $t^{(a)}$; 2“ – informacija apie momentu $t=t^{(a)}$ įvyksiantį antrojo tipo įvykį.

Šeštasis blokas iškviečia iš BĮG eilinių (ankstyviausią iš ten esančių) būsimąjį įvykį ir, nustatęs pagal iškviestąjį įrašą to įvykio įvykimo laiką, perrašo jį į taimerį, t. y. tokiu būdu organizuoja imituojamo laiko t postūmį pirmyn, **paversdamas būsimąjį įvykį einamuoju**. Iškviestąjį iš BĮG įrašą šeštasis blokas perduoda septintajam blokui.

Septintasis blokas, analizuodamas tą įrašą, nustato iškviestojo įvykio tipą ir priklausomai nuo to perduoda valdymą į 8-jį arba 16-jį arba 20-jį bloką.

Aštuntasis blokas, pasinaudodamas antrojo bloko įrašytais tikimybėmis $p(1), p(2), \dots, p(n), \dots, p(\hat{n})$, generuoja šiuo momentu atvykstančių klientų skaičių n , įrašydamas tą skaičių į ką tik atvykusių klientų skaitliuką n , apskaičiuoja naują skaitliuko $N^{(a)}:=N^{(a)}+n$ reikšmę ir formuoja antrojo tipo įvykio įvykimą charakterizuojantį standartinį įrašą „ $t(a); 2; n; N^{(a)} *$ “ į imitavimo eigos protokolą.

Devintasis blokas tikrina skaitliuko n reikšmę ir, jei $n>0$, perduoda programos valdymą į 10-jį bloką, registruojantį eilinį atvykusį klientą, o kai n tampa lygus 0, grąžina programos valdymą į penktąjį bloką, formuojantį naujos klientų partijos atvykimo laiką.

Dešimtas blokas registruoja naujai atvykusį eilinį klientą, suteikdamas jam numerį $N_i:=N_i+1$, kurį gauna pridėdamas vienetą prie senos klientų skaitliuko reikšmės N_i . Be to šis blokas mažina vienetu skaitliuko n reikšmę, kuri tampa $n:=n-1$.

Vienioliktasis blokas tikrina LKG ir, jei yra laisvų kirpėjų, perduoda programos valdymą 12-jam blokui, iškviečiančiam eilinį laisvą kirpėją ir organizuojančiam jo darbo pradžią, o jei LKG tuščia – 13-jam blokui, tikrinančiam eilės ilgį m .

Dvyliktasis blokas iškviečia iš LKG eilinių laisvą kirpėją j , keisdamas LKG turinį; pasitelkdamas antrojo bloko įrašytas tikimybes $p_1, p_2, \dots, p_r, \dots, p_{\hat{r}}$, generuoja aptarnavimo rūšį r , ir, pasinaudodamas antrojo bloko įrašytą tikimybinių tankių $f_r(\tau_{ij})$ atitinkančia standartine programa, generuoja aptarnavimo trukmės atsitiktinę reikšmę τ_{ij} . Po to šis blokas apskaičiuoja aptarnavimo baigimo momentą $t_{ij}^{(b)}=t+\tau_{ij}$, naują kirpėjo j atidirbto laiko T_j skaitliuko reikšmę $T_j:=T_j+\tau_{ij}$ ir suformuoja trečiojo tipo įvykį aprašantį standartinį įrašą į protokolą ir penktojo tipo įvykį atitinkantį įrašą į BĮG, papildydamas jį nauju būsimu įvykiu – aptarnaujamo eilinio kliento aptarnavimo baigimu. Baigęs savo darbą, 12-asis blokas perduoda programos eigos valdymą į 9-jį bloką, tikrinantį laukiančiųjų registracijos ką tik atvykusių klientų skaitliuko turinį n .

Tryliktasis blokas tikrina klientų, laukiančių aptarnavimo, eilės ilgį m ir, jei $m<\hat{m}$, perduoda valdymą į 14-jį bloką, įstatantį eilinį atvykusį klientą į laukiančiųjų aptarnavimo eilę, o jei $m=\hat{m}$, – į 15-jį bloką, išvedantį eilinį klientą N_i be aptarnavimo iš kirpyklos.

Keturioliktasis blokas papildoma KSEG nauju klientu įrašydamas į KSEG įrašą: „ $N_i; t^{(s)} *$ “, kur $t^{(s)}$ – stojimo į eilę laikas ir didina stovinčiųjų eilėje skaitliuko reikšmę vienetu: $m:=m+1$. Po to šis blokas formuoja ketvirtojo tipo standartinį įrašą „ $t^{(s)}; 4; N_i; m *$ “ į protokolą ir perduoda programos valdymą devintajam blokui.

Penkioliktasis blokas didina išvykusių iš kirpyklos be aptarnavimo klientų skaitliuko reikšmę vienetu ($N^{(is)}:=N^{(is)}+1$) ir, suformavęs standartinį septintojo tipo įrašą „ $t; 7; N_i; N^{(is)} *$ “ į protokolą, grąžina programos valdymą 9-jam blokui, kuris, pasibaigus visų atvykusių klientų registravimui, grąžins valdymą į 5-jį bloką.

Šešioliktasis blokas gauna programos valdymą iš 7-jo bloko, kai 6-sis blokas iš BĮG iškviečia penktojo tipo įvykį ir įrašo į taimerį eilinio kliento aptarnavimo baigimo laiką. Gavęs programos valdymą ir atitinkamą informaciją, 16-sis blokas didina aptarnautų klientų skaitliuko reikšmę $N^{(apt)}:=N^{(apt)}+1$, formuoja standartinį įrašą apie penktojo tipo įvykį į protokolą ir papildoma KSEG laisvu kirpėju.

Septynioliktasis blokas tikrina eilės ilgį m ir, kai eilės nėra ($m=0$), grąžina programos valdymą į 6-jį bloką, iškviečiantį naują eilinį įrašą iš BĮG ir keičiantį taimerio reikšmę. Kai $m>0$, septynioliktasis blokas perduoda valdymą į 18-jį bloką.

Aštuonioliktasis blokas išveda eilinių (ilgiausiai laukiantį aptarnavimo) klientą iš KSEG, mažina klientų eilės ilgio skaitliuko reikšmę m vienetu ($m:=m-1$), apskaičiuoja $\tau^{(e)}=t-t_i^{(s)}$ ir $T^{(e)}:=T^{(e)}+\tau^{(e)}$, padidindamas bendro visų klientų prastovėto eilėje laiko $T^{(e)}$ skaitliuko reikšmę dydžiu $\tau^{(e)}$, ir padidina vienetu stovėjusių eilėje klientų skaitliuko reikšmę $N^{(s)}:=N^{(s)}+1$. Po to aštuonioliktasis blokas suformuoja standartinį įrašą apie šeštojo tipo įvykį į imitavimo eigos protokolą.

Devynioliktasis blokas veikia lygiai taip pat, kaip ir 12-sis blokas, t. y. organizuoja eilinio kliento (tačiau dabar paimto ne iš ką tik atvykusiųjų tarpo, o iš eilės) aptarnavimo pradžią, formuoja

trečiojo tipo įrašą į protokolą ir penktojo tipo įrašą į BĮG, papildydamas jos turinį ir mažindamas LKG turinį, tačiau valdymą perduoda ne 9-jam, o 6-jam blokui, išskviečiančiam iš BĮG eilinį būsimąjį įvykį.

Dvidešimtas blokas perima programos valdymą iš 7-ojo bloko tik tada, kai 6-sis blokas išskviečia iš BĮG aštuntojo tipo įvykį ir įrašo į taimerį oficialų eilinės pamainos baigimo laiką $t^{(o,p)}$, t. y. tik tada, kai pamainos darbo imitavimas pasiekia oficialiąją jos pabaigą. Blokas, perėmęs valdymą, suformuoja aštuntojo tipo įvykio įvykimą skelbiantį standartinį įrašą į protokolą ir perduoda valdymą į 21-jį bloką.

Dvidešimt pirmasis blokas tikrina, ar, išvedus iš BĮG aštuntojo tipo įrašą, BĮG liko tuščia, nes galėjo būti, kad, oficialiai pamainai pasibaigus, kirpykloje liko klientų, kuriuos dar aptarnauja kirpėjai, o taip pat klientų, vis dar laukiančių savo aptarnavimo pradžios eilėje, kurią jie užėmė dar iki pamainos pabaigos ir todėl dabar manantys, jog turi teisę reikalauti aptarnavimo.

Jei BĮG tuščia, tai dvidešimt pirmasis blokas perduoda programos valdymą 22-jam blokui, kuris formuoja faktišką pamainos pabaigą, o jei būsimųjų įvykių grandinėje šiuo metu, kai $t=t^{(o,p)}$, dar yra įrašytų įvykių, – 24-jam blokui, kuris, veikdamas lygiai taip pat kaip ir 6-sis blokas, išskviečia iš BĮG eilinį būsimąjį įvykį.

Dvidešimt antrasis blokas, perėmęs programos valdymą iš 21-ojo bloko, formuoja devintojo tipo standartinį įrašą į protokolą apie faktišką pamainos pabaigą ir perduoda valdymą 23-jam blokui.

Dvidešimt trečiasis blokas išskviečia statistinės informacijos, sukauptos ką tik pasibaigusios pamainos darbo imitavimo metu, apdorojimo ir statistinių charakteristikų apskaičiavimo programą.

Apskaičiuojamos dydžių $\bar{q}_j$, $\bar{A}$, $\bar{\tau}_i^{(a)}$, $\bar{\tau}^{(e)}$, $\bar{m}$, $\bar{a}$ reikšmės, naudojantis formulėmis:

$$1) \bar{q}_j^{(1)} = \frac{T_j \cdot 100\%}{t^{(f,p)} - t^{(p,p)}} \text{ arba } \bar{q}_j^{(2)} = \frac{T_j \cdot 100\%}{t^{(o,p)} - t^{(p,p)}}, \text{ kur antruoju atveju, lyginant kirpėjo atidirbtą}$$

per pamainą laiką T_j su oficialia pamainos trukme $t^{(o,p)} - t^{(p,p)}$ (kur $t^{(o,p)}$ – oficiali pamainos pabaiga ir $t^{(p,p)}$ – normatyvinė pamainos pradžia), gali atsitikti ir taip, kad $\bar{q}_j^{(2)} > 100\%$, kai tuo tarpu $\bar{q}_j^{(1)}$ visada bus ne didesnis negu 100%;

$$2) \bar{A} = \frac{N^{(apt)}}{t^{(f,p)} - t^{(p,p)}}, \text{ kur } \bar{A} - \text{kirpyklos vidutinė praleidžiamoji galia, rodanti, kiek vidutiniškai}$$

klientų aptarnaujama per laiko vienetą;

$$3) \bar{\tau}^{(a)} = \frac{\sum_{j=1}^k T_j}{N^{(apt)}}, \text{ kur } \bar{\tau}^{(a)} - \text{vidutinė vieno kliento aptarnavimo trukmė;}$$

$$4) \bar{\tau}_0^{(e)} = \frac{T^{(e)}}{N^{(a)}}, \bar{\tau}_1^{(e)} = \frac{T^{(e)}}{N^{(apt)}}, \bar{\tau}_2^{(e)} = \frac{T^{(e)}}{N^{(s)}}, \text{ kur } \bar{\tau}_0^{(e)} - \text{kliento stovėjimo eilėje vidutinė}$$

trukmė, apskaičiuota visų atvykusių į kirpyklą klientų atžvilgiu; $\bar{\tau}_1^{(e)}$ – visų aptarnautų klientų atžvilgiu, o $\bar{\tau}_2^{(e)}$ – tikrai stovėjusių eilėje klientų atžvilgiu;

$$5) \bar{m} = \frac{T^{(e)}}{t^{(f,p)} - t^{(p,p)}}, \text{ kur } \bar{m} - \text{vidutinis eilės ilgis, rodantis, kiek klientų pastoviai (visą laiką)}$$

turėtų stovėti eilėje, kad per pamainą jie visi kartu „pristovėtų“ lygiai tiek laiko $T^{(e)}$, kiek pristovėjo per pamainą visi klientai;

$$6) \bar{a} = \frac{N^{(apt)}}{N^{(a)}} \cdot 100\%, \text{ kur } \bar{a} - \text{klientų, aptarnautų per pamainą, dalis procentais visų, atvykusių}$$

į kirpyklą, klientų atžvilgiu; šis dydis rodo, kokia dalimi kirpyklos dabartinė praleidžiamoji galia patenkina esančią paklausą.

Pabaigęs apskaičiavimus, 23-sis blokas įsimena bei spausdina skaičiavimų rezultatus ir grąžina programos valdymą į 3-jį bloką, tikrinantį, ar praėjusioji pamaina nėra paskutinė: jei ši pamaina nėra paskutinė, trečiasis blokas perduoda valdymą į ketvirtąjį bloką ir prasideda naujos pamainos imitavimas, o kai praėjusi pamaina yra paskutinė, tai valdymas perduodamas į 30-jį bloką, kuriame apdorojama statistinė informacija, sukaupta per visas eksperimentatoriaus numatytas pamainas, kurių skaičius $\hat{p}$ buvo užduotas antrajame bloke.

Dvidešimt ketvirtasis blokas perima programos valdymą iš 21-ojo bloko tuo atveju, kai pamainai oficialiai pasibaigus, BĮG nėra tuščia. 24-sis blokas veikia panašiai kai ir jau aprašytas 6-sis

blokas, t. y. iškviečia iš BĮG eilinių būsimųjų įvykių ir perveda taimerį į naują laiką. Po to šis blokas perduoda valdymą 25-jam blokui.

Dvidešimt penktasis blokas veikia panašiai kaip ir 7-sis blokas – jis nustato iškviesto iš BĮG įvykio tipą. Skirtumas tik tame, kad šiuo atveju nebegali būti 8-jo tipo įvykio, todėl 25-sis blokas perduoda valdymą tik dviem kryptimis: kai iškviestasis įvykis priklauso 2-jam tipui, valdymas iškart grąžinamas į 21-jį bloką, nes klientai, atvykę į kirpyklą po pamainos oficialiosios pabaigos, jau nebeaptarnaujami, o jei iškviestasis iš BĮG įvykis priklauso 5-jam tipui, tai valdymas perduodamas 26-jam blokui.

Dvidešimt šeštasis blokas veikia panašiai kaip 16-sis blokas: įrašo informaciją apie 5-jo tipo įvykio įvykimą į protokolą, padidindamas aptarnautų klientų skaitliuko reikšmę $N^{(apt)}$ vienetu ir papildydamas LKG baigusių darbą kirpėju.

Dvidešimt septintasis blokas veikia panašiai kaip ir 17-sis blokas – jis tikrina ar eilėje dar yra neaptarnautų klientų. Kai tokių nėra, valdymas grąžinamas į 21-jį bloką, tikrinantį BĮG turinį, o kai $m > 0$, tai valdymas perduodamas 28-jam blokui.

Dvidešimt aštuntasis blokas veikia panašiai kaip ir 18-sis blokas: išveda iš KSEG eilinių ilgiausiai laukiantį aptarnavimo klientą, mažina vienetu eilės ilgio skaitliuko reikšmę m , apskaičiuoja $\tau^{(e)} = t - t_i^{(s)}$ ir $T^{(e)} = T^{(e)} + \tau^{(e)}$, padidina vienetu stovėjusiųjų eilėje skaitliuko reikšmę $N^{(s)}$ ir įrašo į protokolą standartinį 6-jo tipo įrašą.

Dvidešimt devintasis blokas veikia panašiai kaip ir 19-sis (ir 12-sis) blokas: iš LKG iškviečia laisvą kirpėją, generuoja r ir $\tau_{i,j}$ reikšmes, apskaičiuoja $t_{i,j}^{(b)}$ ir T_j , į protokolą įrašo 3-jo tipo įvykį, o į BĮG – informaciją apie būsimąjį penktojo tipo įvykį. Po to šis blokas grąžina valdymą į 21-jį bloką, tikrinantį BĮG turinį ir, arba grąžinantį valdymą į likusių kirpykloje klientų aptarnavimo ciklą, arba išvedantį iš to ciklo ir perduodantį valdymą per 22-jį ir 23-jį blokus į 3-jį bloką.

Trisdešimtas blokas perima valdymą iš 3-jo bloko tuo atveju, kai baigiasi visų numatytų pamainų funkcionavimo imitavimas, ir apdoroja per visas pamainas sukaupią statistinę informaciją, naudodamasis formulėmis:

$$1) \quad \bar{q}_j = \frac{\sum_{p=1}^{\hat{p}} \bar{q}_j^{(p)}}{\hat{p}}, \text{ kur } \bar{q}_j - \text{vidutinis kirpėjo } j \text{ apkrovimas (procentais) per visas } \hat{p} \text{ pamainų, o}$$

$\bar{q}_j^{(p)}$ – vidutinis to kirpėjo apkrovimas per p -ją pamainą;

$$2) \quad \bar{A} = \frac{\sum_{p=1}^{\hat{p}} \bar{A}^{(p)}}{\hat{p}}, \text{ kur } \bar{A} - \text{vidutinė kirpyklos praleidžiamoji galia, apskaičiuota, atsižvelgiant į}$$

visas $\hat{p}$ pamainas, $\bar{A}^{(p)}$ – vidutinė kirpyklos praleidžiamoji galia per p -ją pamainą;

$$3) \quad \bar{\tau}^{(a)} = \frac{\sum_{p=1}^{\hat{p}} \bar{\tau}_p^{(a)}}{\hat{p}}, \text{ kur } \bar{\tau}^{(a)} - \text{vidutinė vieno kliento aptarnavimo trukmė, apskaičiuota,}$$

atsižvelgiant į visas $\hat{p}$ pamainas, $\bar{\tau}_p^{(a)}$ – vidutinė vieno kliento aptarnavimo trukmė p -je pamainoje;

$$4) \quad \bar{\tau}^{(e)} = \frac{\sum_{p=1}^{\hat{p}} \bar{\tau}_p^{(e)}}{\hat{p}}, \text{ kur } \bar{\tau}^{(e)} - \text{vidutinė vieno kliento stovėjimo eilėje trukmė, apskaičiuota,}$$

atsižvelgiant į visas $\hat{p}$ pamainas, $\bar{\tau}_p^{(e)}$ – vidutinė vieno kliento stovėjimo eilėje trukmė p -joje pamainoje; priklausomai nuo to, kurios klientų grupės atžvilgiu buvo apskaičiuota $\bar{\tau}_p^{(e)}$ (žiūr. 23-jo bloko aprašymą), 30-jame bloke gali būti apskaičiuotos $\bar{\tau}_0^{(e)}$, $\bar{\tau}_1^{(e)}$ ir $\bar{\tau}_2^{(e)}$ reikšmės pavieniui arba ir visos kartu;

$$5) \quad \bar{m} = \frac{\sum_{p=1}^{\hat{p}} \bar{m}_p}{\hat{p}}, \text{ kur } \bar{m} - \text{vidutinis eilės ilgis, apskaičiuotas, atsižvelgiant į visas } \hat{p} \text{ pamainas,}$$

$\bar{m}_p$ – vidutinis eilės ilgis p -oje pamainoje;

$$6) \bar{a} = \frac{\sum_{p=1}^{\hat{p}} \bar{a}^{(p)}}{\hat{p}}, \text{ kur } \bar{a} - \text{vidutinė aptarnautų klientų dalis procentais, visų aptarnautų klientų}$$

atžvilgiu, apskaičiuota visoms $\hat{p}$ pamainoms, $\bar{a}^{(p)}$ – ta pati dalis, apskaičiuota p-jai pamainai;

$$7) \bar{q} = \frac{\sum_{j=1}^k \bar{q}_j}{k}, \text{ kur } \bar{q} - \text{vidutinis kirpėjų apkrovimas per visas } \hat{p} \text{ pamainas, } \bar{q}_j - \text{vidutinis j-jo}$$

kirpėjo apkrovimas apskaičiuotas visoms $\hat{p}$ pamainoms, k – kirpėjų skaičius.

Gauti skaičiavimo rezultatai spausdinami panašiai kaip ir 23-jo bloko rezultatai.

Trisdešimtas blokas, apskaičiuodamas statistines charakteristikas, naudojasi 23-jo bloko kiekvienos pamainos metu gautais rezultatais, kurie kaskart įrašomi į kompiuterio atmintį ir saugojami ten iki tol, kol jų poreikalaus 30-sis blokas.

Taupant kompiuterio atmintį ir domintis tik vidurkiais, apskaičiuotais visų $\hat{p}$ pamainų atžvilgiu, nebuvo būtina saugoti kiekvienos pamainos charakteristikas, gautas 23-jo bloko skaičiavimuose. Buvo galima įvesti specialius **skaitliukus-sumatorius**, sumuojančius dydžius $\bar{q}_j^{(p)}$, $\bar{A}^{(p)}$, $\bar{\tau}_p^{(a)}$, $\bar{\tau}_p^{(e)}$, $\bar{m}_p$, $\bar{a}^{(p)}$ kiekvieną kartą 23-jam blokui baigiant darbą.

Nagrinėjamoje programoje, norint turėti kompiuterio atmintyje ir kiekvienos pamainos skaitines charakteristikas, taip nebuvo daroma: 23-sis blokas nesumuodavo šių charakteristikų, o užrašydavo gautus rezultatus į specialiai išskirtą kompiuterio atminties zoną ir išspausdindavo juos.

Trisdešimt pirmasis blokas ieško atsakymo į klausimą ar spausdinti imitavimo protokolą (ar kokį nors protokolo fragmentą).

Nagrinėjamoje programoje atsakymą į klausimą šis blokas gaudavo iš eksperimentatoriaus interaktyviajame režime: eksperimentatorius, išvydęs minėtąjį klausimą kompiuterio ekrane, turėdavo nuspausti arba klavišą „T“ („taip“) arba „N“ („ne“). Tais atvejais, kai buvo norima spausdinti ne visą protokolą, buvo nurodomas eksperimentatorių dominantis laiko intervalas, įrašant atitinkamą informaciją į displėjaus ekrane pasirodančią eilutę „pamaina... nuo... iki...“, t. y. eksperimentatorius turėdavo užpildyti kompiuterio ekrane atvaizduotoje eilutėje daugtaškiais pažymėtas vietas. Kai eksperimentatorius, jų neužpildydamas, nuspausdavo klavišą „T“, tai printeris pagal nutylėjimo taisyklę spausdindavo visą protokolą, o jei „N“ – nespausdindavo nieko.

Nagrinėjamoje programoje buvo numatytas ir automatinis 31-jo bloko darbo režimas. Tame režime šis blokas gaudavo atsakymą iš specialiai 2-jame bloke įrašytos informacijos, nurodančios, ką reikia spausdinti. Kai tokia informacija 2-jame bloke nebūdavo įrašyta, 31-sis blokas pagal nutylėjimo taisyklę automatiškai pereidavo į interaktyvųjį režimą ir išvedęs minėtąjį klausimą į displėjaus ekraną, sustodavo, laukdamas atsakymo.

Gavęs atsakymą „taip“, 31-sis blokas perduodavo valdymą 32-jam blokui, kuris perduodavo printeriui spausdinimui reikalingą informaciją, o gavęs atsakymą „ne“ – valdymą perduodavo 33-jam blokui, klausiančiam eksperimentatoriaus, ar tęsti imitavimą naujomis sąlygomis, ar tuoj pat baigti darbą.

Trisdešimt antrasis blokas, gavęs valdymą iš 31-jo bloko, inicijuoja spausdintuvo darbą ir spausdina protokolą (ar jo fragmentą), savo eile perduodamas valdymą į 33-jį bloką.

Trisdešimt trečiasis blokas veikia interaktyviame režime ir išveda į displėjaus ekraną klausimą: „Ar tęsime imitavimą naujomis sąlygomis?“.

Gavęs atsakymą „taip“, kai eksperimentatorius nuspaudžia klavišą „T“, šis blokas perduoda valdymą į antrąjį bloką, ir tuomet eksperimentatorius turi įvesti į kompiuterį naujas formalųjų programos parametrų faktines reikšmes (žiūr. antrojo bloko aprašymą). Po to prasideda naujas imitavimo seansas.

Gavęs atsakymą „ne“, kai eksperimentatorius nuspaudžia klavišą „N“, 33-sis blokas perduoda valdymą į 34-jį bloką, kuris užbaigia darbą.

Trisdešimt ketvirtasis blokas – imitavimo pabaiga.

Programos, kurios blokinė schema pavaizduota 304 pav., aprašymas čia atliktas taip detalai todėl, kad studentai, kurdami pirmąją gana sudėtingą imitavimo programą, neretai susiduria su eile sunkumų ir klausimų, ką ir kaip daryti. Mokymo praktika parodė, jog nagrinėdami konkretų tokios programos pavyzdį, studentai žymiai greičiau išmoksta savarankiškai kurti panašias programas ir geriau įsisavina imitacinio modeliavimo esmę.

Atsižvelgiant į šį tikslą, čia ir pateikta viena paprasčiausių ir aiškiausių programų, kurias sudarė trečiojo kurso studentai.