KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

Ingrida Svilainytė

REGENERACINIS METODAS EILIŲ TEORIJOJE

Magistro darbas

Vadovas
prof. dr. A. J. Aksomaitis

KAUNAS, 2005

KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU
Katedros vedėjas
prof. dr. J.Rimas
2005 06 03

REGENERACINIS METODAS EILIŲ TEORIJOJE

Taikomosios matematikos magistro baigiamasis darbas

Kalbos konsultantas
dr. J. Džežulskienė

2005 05 31

Recenzentas
dr. R. Jankūnienė
2005 06 01

Vadovas
prof. dr. A. J. Aksomaitis

2005 06 03

Atliko
FMMM 3 gr. stud.
I. Svilainytė
2005 05 26

KAUNAS, 2005

KVALIFIKACINĖ KOMISIJA

Pirmininkas:	Leonas Saulis, profesorius (VGTU)
Sekretorius:	Eimutis Valakevičius, docentas (KTU)
Nariai:	Algimantas Jonas Aksomaitis, profesorius (KTU)
	Vytautas Janilionis, docentas (KTU)
	Vidmantas Povilas Pekarskas, profesorius (KTU)
	Rimantas Rudzkis, banko NORD/LB vyriausias analitikas
	Zenonas Navickas, profesorius (KTU)
	Arūnas Barauskas, UAB „Elsis“ generalinio direktoriaus pavaduotojas

Svilainytė I. Regenerative method in queue theory: Master‘s work in applied mathematics / supervisor dr. Prof. A. J. Aksomaitis; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2005. – 77 p.

SUMMARY

While modeling stochastic systems it is very important results to be examined using reliable statistics analysis. Estimation methods that can allow user to make statistic conclusions about model from simulation results are needed. These methods are also used while determining relation between simulation time and precision of estimations.

To complete the task regeneration method was chosen. This method is successfully used in various practical problems solving. The usage of regeneration method is based on the fact that many stochastic systems renovate in a probability sense. Besides regeneration method easily solves such “tactical” problems like:

- How to start simulation?
- Which moment to start to gather data at?
- What to do with strongly correlated data?

Received results are confidential intervals of the queue forming in two and three – channel system characteristics (medium waiting time in queue and maximal length of queue). From these results we can judge about effectiveness of system work and relation between results’ precision and simulation time.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

ĮVADAS

Dauguma užduočių, iškylančių tyrinėjant operacijas ir analizuojant sistemas, per daug sudėtingos, kad jas būtų galima išspręsti, konstruojant ir tiriant matematinius modelius. Dažnai vienintelis praktinis panašaus tipo uždavinių sprendimas yra kompiuterinis modeliavimas. Tai, be abejo, susiję su stochastinėmis sistemomis, apimančiomis įvairias aptarnavimo sistemas, aptarnavimo tinklus, atsargų valdymo sistemas, techninės kontrolės, profilaktinius ir remonto darbus, taip pat ir kitas daugiamates sistemas.

Stochastinių sistemų modeliavimą reikėtų traktuoti kaip statistinį eksperimentą. Iš pradžių darome sistemos modelį, leidžiantį tyrinėti problemos esmę vaizduojant sistemos struktūrą ir jo priežastinį – pasekminį mechanizmą. Po to su šiuo modeliu atliekami imitaciniai eksperimentai ir analizuojami gauti duomenys, kuriais remiantis daromos išvados apie sistemos veiklą. Taigi imitaciniai eksperimentai visiškai analogiški paprastiems statistiniams bandymams ir tam, kad būtų gauta reikšmingų rezultatų, reikia remtis patikimomis statistinėmis procedūromis.

Vis dėlto dauguma svarbių statistinių modelių žymiai sudėtingesni nei tie eksperimentai, kurie analizuojami klasikiniais statistiniais metodais. Todėl statistinių procedūrų rinkinys, naudojamas modeliavimo rezultatams analizuoti, yra labai ribotas. Esant šioms aplinkybės, adekvati gaunamų rezultatų analizė dažnai yra sudėtinga arba tiesiog neįmanoma.

Problemoms spręsti pradėti vystyti statistiniai gaunamų duomenų analizės metodai, taikomi būtent regeneracinių modelių klasei, kuri yra gana didelė ir įdomi. Regeneracinio metodo taikymas yra sąlygotas to fakto, kad dauguma stochastinių sistemų kartojasi pagal tikimybę. Vartotojui tai suteikia galimybę vykstant modeliavimui stebėti nepriklausomas ir vienodai pasiskirsčiusias duomenų grupes, kartu ir palengvina statistinę analizę. Dar daugiau, regeneracinis metodas pateikia tokių sunkių taktinių problemų sprendimo būdą:

- kaip pradėti modeliavimą;
- nuo kurio momento pradėti rinkti ir fiksuoti duomenis;
- kaip elgtis su stipriai koreliuotais duomenimis.

Šiame darbe nagrinėjamas eilių aprašymas regeneraciniu metodu, apskaičiuojamas vidutinis paraiškų laukimo eilėje laikas ir vidutinis maksimalus eilės ilgis, bei jų pasikliautinieji intervalai dvikanalėje ir trikanalėje sistemoje. Regeneraciniu momentu imamas momentas, kai bent vienas aptarnavimo įrenginys yra laisvas.

2004m. konferencijoje „V studentų konferencija“ buvo pristatytas pranešimas tema „Maksimalios eilės dvikanalėje sistemoje tyrimas regeneraciniu metodu“, konferencijos pranešimo medžiaga pateikta ketvirtame priede.

1. BENDROJI DALIS

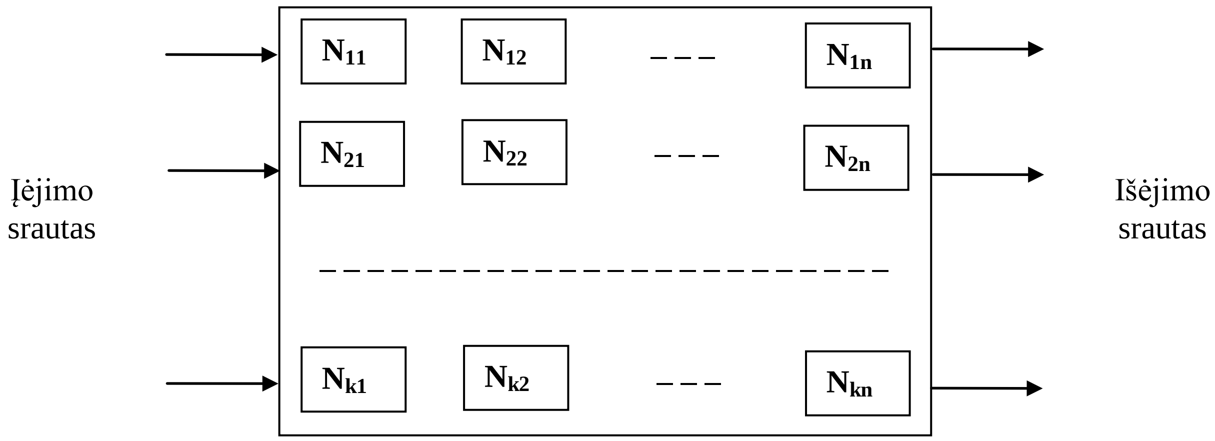
1.1 REGENERACINIAI PROCESAI IR EILIŲ TEORIJA

1.1.1 PAGRINDINĖS EILIŲ TEORIJOS SĄVOKOS

Eilių teorija nagrinėja stochastinių sistemų, kurios vadinamos eilių aptarnavimo sistemomis, funkcionavimo efektyvumą. Eilių aptarnavimo sistemos aptarnauja į jas patenkančių paraiškų srautus. Paraiškos suprantame kaip tam tikro užsakymo pasirodymą. Paraiškos į sistemą pateka atsitiktiniais laiko momentais, viena paskui kitą. Paraiška aptarnaujama per tam tikrą laiko tarpą, po kurio sistema tampa laisva ir gali aptarnauti kitą atėjusią paraišką. Kiekviena tokia sistema gali būti sudaryta iš kelių tarpusavyje nepriklausomų elementų, kurie dažnai vadinami aptarnavimo kanalais arba aparatais. Tokie pavyzdžiai gali būti bilietų kasos, aerouostai, informacinės sistemos ir kt.

Sistemos aptarnavimo efektyvumą, priklausantį nuo paraiškų srauto charakteristikų, sistemos kanalų skaičiaus ir kanalų našumo, galima įvertinti tam tikru eilių teorijos matematiniu modeliu. Sistemos efektyvumo kriterijumi gali būti įvairūs dydžiai ir funkcijos: paraiškos aptarnavimo tikimybė, vidutinis aptarnautų paraiškų skaičius per laiko vienetą, vidutinis laukimo laikas iki aptarnavimo pradžios, vidutinis maksimalus eilės ilgis, vidutinis kanalų ir sistemos prastovų laikas, eilės ilgio pasiskirstymo dėsnis, sistemos pralaidumo koeficientas ir kt. Šių kriterijų skaitinės charakteristikos tam tikru mastu apibūdina sistemos sugebėjimą įvykdyti jai keliamus uždavinius.

Visos eilių aptarnavimo sistemos turi tam tikrą struktūrą, kurios schema pavaizduota 1.1.1.1 pav.



1.1.1.1 pav. Aptarnavimo sistemos struktūra

Kiekvieną aptarnavimo sistemą sudaro šie pagrindiniai elementai: įėjimo srautas, išėjimo srautas ir aptarnavimo sistema.

Paraiškų srautas, kuris patenka į aptarnavimo sistemą, vadinamas įėjimo srautu, o išeinantis iš aptarnavimo sistemos srautas – išėjimo srautu. Visuma aptarnavimo aparatų ir taisyklės, pagal kurias aptarnaujamos paraiškos, sudaro aptarnavimo sistemą [4].

1.1.2 APTARNAVIMO SISTEMŲ KLASIFIKACIJA IR PAGRINDINĖS JŲ CHARAKTERISTIKOS

Klasifikuojant eilių aptarnavimo sistemas, svarbus požymis yra sistemos veikimas, atėjus naujai paraiškai, kai visi aptarnavimo kanalai užimti. Galimi du atvejai:

- paraiškos palieka aptarnavimo sistemą ir tolimesniame procese ši paraiška nebedalyvauja. Tokios sistemos vadinamos sistemomis be eilės;
- paraiška laukia eilėje tol, kol vienas aptarnavimo kanalas tampa laisvas. Kai tik tai įvyksta, aptarnaujama paraiška esanti eilėje. Tokios sistemos vadinamos sistemomis su eile.

Sistemose su eile paraiškos, kurios yra eilėje, gali būti aptarnaujamos skirtingai. Jos gali būti aptarnaujamos:

- tokia tvarka, kokia jos patenka į sistemą;
- atsitiktinai imant paraiškas iš eilės;
- pagal paraiškų prioritetus.

Sistemos su eile skirstomos į sistemas:

- su apribojimais,
- be apribojimų.

Apribojimai gali būti eilei, paraiškų laukimo laikui eilėje ir paraiškos buvimo laikui sistemoje. Sistemose su neribota eile kiekviena paraiška, jei nėra laisvų aptarnavimo kanalų, stoja į eilę ir „kantriai“ laukia kol atsilaisvins kanalas; kiekviena paraiška anksčiau ar vėliau bus aptarnauta.

Sistemose su ribotu laukimo laiku eilėje paraiškos laukimo laikas iki aptarnavimo yra ribojamas. Jei laukimo laikas viršijamas, neaptarnauta paraiška palieka sistemą.

Mišraus tipo aptarnavimo sistemos sujungia sistemų su eile ir be eilės savybes. Tokios sistemos pavyzdys yra sistema su apribojamu eilės dydžiu. Jei atėjusi paraiška randa visus kanalus užimtus, tai ji stoja į eilę tik tuo atveju, jei eilė nedidelė, (kai paraiškų eilės ilgis neviršija tam tikro skaičiaus). Jei eilėje laukia daug paraiškų, tai atėjusi paraiška palieka sistemą.

Mišraus tipo sistemose dar gali būti apribojamas paraiškos buvimo laikas eilėje. Realiose sistemose šie apribojimai gali būti vienu metu [4].

1.1.3 DVIKANALĖ IR TRIKANALĖ APTARNAVIMO SISTEMOS

Šiame darbe, nagrinėjant regeneracinio metodo taikymą eilių teorijoje, modeliuojama dvikanalė ir trikanalė aptarnavimo sistemos su eile be apribojimų.

Regeneraciniu momentu galima imti įvairius momentus (bent vienas kanalas laisvas, visi kanalai laisvi, bent du kanalai laisvi ir pan.). Šiame darbe regeneraciniu momentu imamas momentas, kai bent vienas aptarnavimo kanalas yra laisvas.

Standartinės dvikanalė ir trikanalė aptarnavimo sistemos pavaizduotos 1.1.3.1 pav. ir 1.1.3.2 pav. bei pasižymi tokiomis savybėmis:

- modeliuojama sistema su eile;
- sistemos eilė be apribojimų;
- eilėje esančios paraiškos aptarnaujamos tokiu principu: pirma atėjo – pirma aptarnaujama.

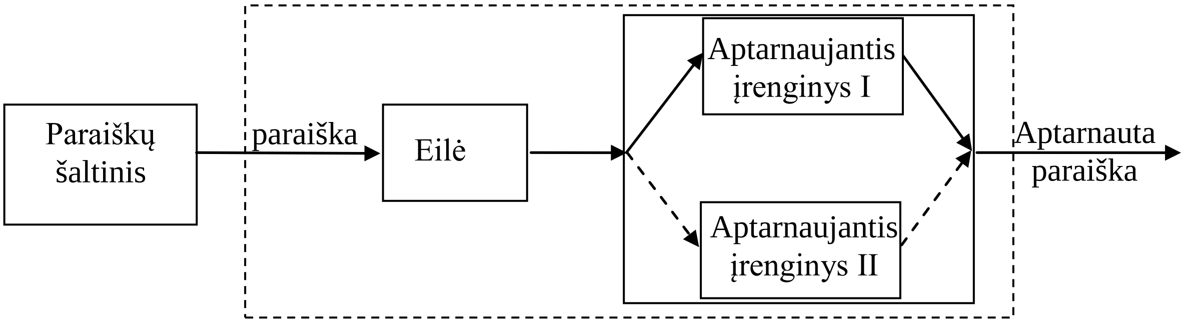
Paraiškos sistemoje pasirodo po vieną. Jeigu atėjusi paraiška randa bent vieną aptarnavimo įrenginį laisvą, tai tuojau pat pradedamas jos aptarnavimas, jį baigus paraiška palieka sistemą. Tačiau jeigu įrenginys randamas užimtas, tai paraiška stoja į eilę ir laukia.

Sistemos efektyvumo kriterijai – vidutinio paraiškos laukimo laiko eilėje iki aptarnavimo pradžios pasikliautinasis intervalas bei vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas. Jie buvo skaičiuojami, esant trims skirtingiems paraiškų atėjimo srauto ir laukimo eilėje tipams:

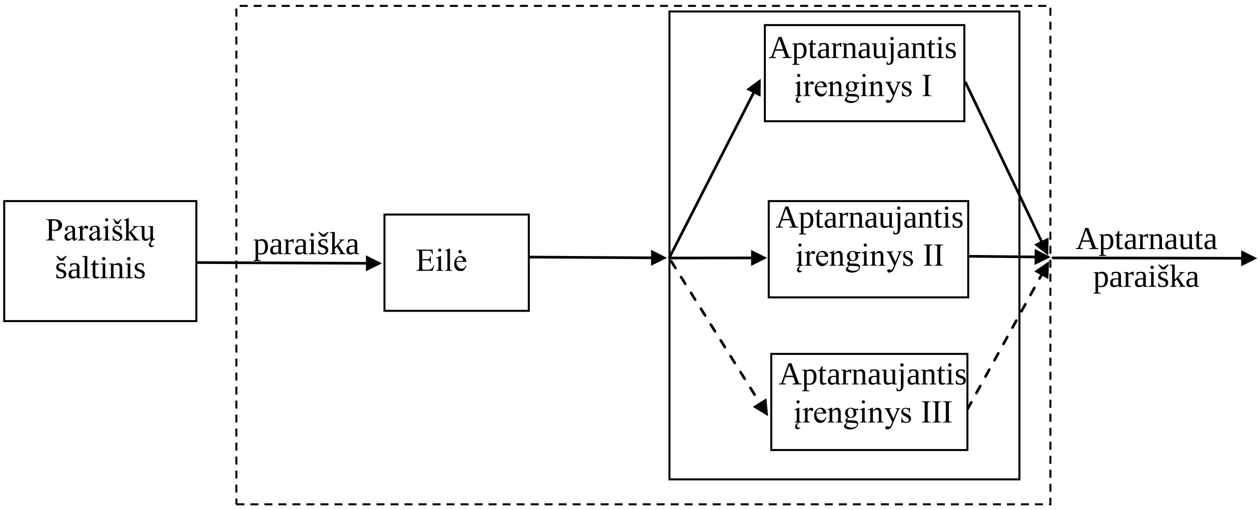
- paraiškos sistemoje pasirodo determinuotai; paraiškų laukimo laiko trukmės pasiskirstę pagal tolygųjį dėsnį;
- paraiškų atėjimo ir laukimo laiko trukmės pasiskirstę pagal tolygųjį skirstinį;
- paraiškų atėjimo ir laukimo laiko trukmės pasiskirstę pagal eksponentinį dėsnį.

Išnagrinėsime pavyzdį, paaiškinantį regeneracinio metodo taikymą modeliuojamos sistemos efektyvumo kriterijaus apskaičiavimui.

Tarkime, šioje sistemoje šaltinis generuoja naujas paraiškas kas 60 sekundžių. Be to, skirtingų paraiškų laukimo eilėje laiko ilgiai yra nepriklausomi atsitiktiniai dydžiai, vienodai pasiskirstę pagal tolygųjį skirstinį intervale nuo 10 iki 90 sekundžių. Tarkime, kad sistemos funkcionavimo kokybės kriterijus $E\{W\}$ – vidutinis paraiškos laukimo eilėje laikas (neįskaitant pačio aptarnavimo laiko). Kadangi skaičiavimo požiūriu paprasti analitiniai rezultatai tiksliai $E\{W\}$ reikšmei nustatyti duotoje sistemoje neegzistuoja, akivaizdžiu tampa modeliavimo reikalingumas.



1.1.3.1 pav. Dvikanalė aptarnavimo sistema



1.1.3.2 pav. Trikanalė aptarnavimo sistema

Taigi užduotis susideda iš sistemos modeliavimo ir gautų rezultatų analizės, siekiant įvertinti dydį E{W}. Ir reikia gauti tam tikrą įverčio E{W} „patikimumo“ matą, pavyzdžiui 90% - tinį pasikliautinąjį intervalą.

Tegu W_1 reiškia pirmosios paraiškos laukimo eilėje laiką, W_2 – antrosios paraiškos laukimo eilėje laiką ir t.t. Tada, jei bendras modeliuojant aptarnautų paraiškų skaičius N, empirinis vidurkis

$$\frac{W_1 + W_2 + \cdots + W_N}{N} \qquad (1.1.2.1)$$

yra suderintasis E{W} įvertis, kadangi akivaizdu, kad empirinis vidurkis su tikimybe 1 artėja prie tikrosios E{W} reikšmės, kai $N \to \infty$. Tačiau empirinis vidurkis (1.1.3.1) priklausys nuo pradinių sąlygų, kitaip tariant bus tikrosios reikšmės paslinktasis įvertis. Pavyzdžiui, jei kaip paprastai, $W_1 = 0$, tai keletas pirmųjų laukimo eilėje laiko reikšmių turi tendenciją būti mažos. Poslinkis išnyks, jei pradinė W_1 reikšmę pasirinksime atsitiktiniu būdu iš pačio W pasiskirstymo. Nelaimei, neaiškus net dydžio W vidurkis, jau nekalbant apie jo skirstinį. Taigi toks „sprendimas“ nėra visiškai priimtinas.

Problemą galime išspręsti tęsdami modeliavimą, nerenkant duomenų tol, kol įsitikinsime, jog modeliuojama sistema pasiekė stacionarų rėžimą, ir tada pradedant tuo momentu, rinksime duomenis. Pavyzdžiui, galima būtų sumodeliuoti sistemą 2000 paraiškų aptarnavimui, atmesti pirmųjų 1000 laukimo eilėje laikų ir naudoti vidurkį

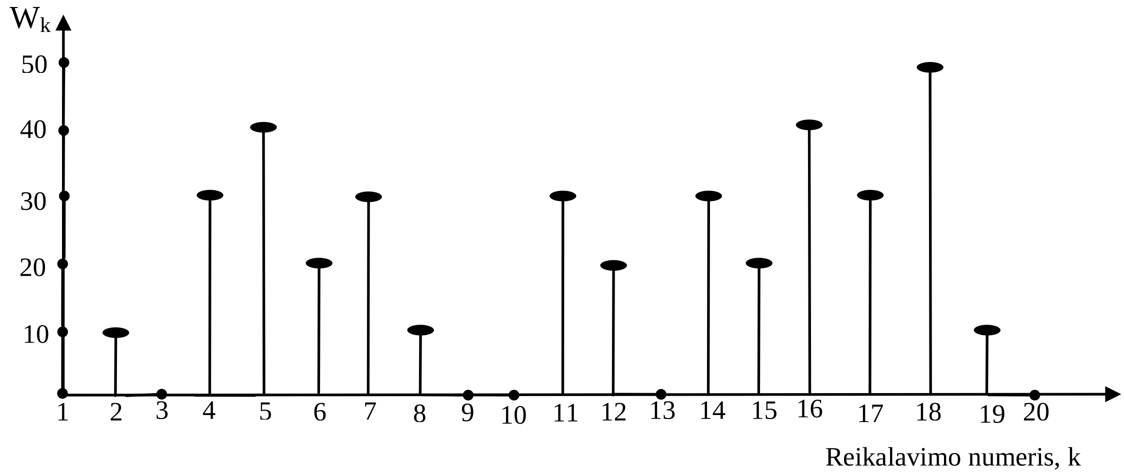
$$\frac{W_{1001}+W_{1002}+\cdots+W_{2000}}{1000} \tag{1.1.2.2}$$

kaip E{W} įvertį. Bet anaiptol neaišku, kokio būtent ilgumo turi būti „stabilizacijos“ periodas. Todėl modeliuojant tokiu būdu nenašiai sunaudojama daug „mašininio“ laiko.

Egzistuoja ir kitas sunkumas, kylantis naudojant empirinio vidurkio (1.1.2.1) įvertinimo procedūrą. Pagal modeliavimo rezultatus norėtųsi gauti tam tikrą dydžio E{W} įverčio patikimumo matą, nusakantį tikimybę, kad pakartojus modeliavimą gauti įverčiai būtų analogiški. Pavyzdžiui, tarkime, kad norime sudaryti 90% - tinį pasikliautinąjį intervalą *I* tikrajai E{W}reikšmei tokį, jog esant bet kokiam nepriklausomam imitacinio eksperimento pakartojimui tikimybė, kad tikroji E{W} reikšmė priklauso intervalui *I*, būtų 0,90. Tačiau tokiems intervalams konstruoti, naudojant klasikinę statistiką, eksperimento duomenys turi būti nepriklausomi ir vienodai pasiskirstę, pagal tam tikrą tikimybinį dėsnį. Aptarnavimo sistemos modeliavimo rezultatai nusako laukimo eilėje laiko ilgių seką W_1, W_2,..., W_N. Akivaizdu, kad jei W_k – didelis, tai kita (k+1)-moji paraiška, irgi lauks ilgai, tai nepriklauso nuo to, ar modeliavimas prasideda rinkimu iš stacionaraus W pasiskirstymo ar ne. O kadangi laukimų W_1, W_2,..., W_N ilgiai nėra nepriklausomi dydžiai, klasikinė statistika mažai naudinga įverčio E{W} patikimumo nustatymui.

Taigi poslinkiai, nulemti pradinių sąlygų, ir stipriai koreliuoti pradiniai duomenys sudaro rimtas kliūtis pasiūlytos įvertinimo procedūros, kuri remiasi empiriniu vidurkiu (1.1.2.1), naudojimui. Tačiau egzistuoja paprastas šių kliūčių pašalinimo metodas.

Išnagrinėsime tokį pavyzdį. Tarkime, kad modeliavimas pradedamas, nustačius W_1 = 0, jog galima būtų imituoti tam tikrą neilgą laiko tarpą, ir, kad paraiškų laukimo eilėje laiko stebėjimas atitinka pavaizduotą 1.1.3.2 pav.



1.1.3.3 pav. Paraiškų laukimo eilėje laikai

Matome, jog paraiškos 1, 3, 9, 10, 13 ir 20 – atėjimo momentu randa aptarnavimo įrenginį laisvą, ir joms nereikia laukti eilėje, kai tuo tarpu paraiškos 2, 4, 5, 6, 7, 8, 11, 12, 14, 15, 16, 17, 18, 19 turi laukti , kol jas aptarnaus. Be to, nuo 1-osios paraiškos atėjimo iki 2-osios išėjimo aptarnaujantis įrenginys nuolat užimtas. Po to, kol nepasirodo 3-oji paraiška, jis nuolat laisvas. Ir vėl užimtas, kol vyksta 3, 4, 5, 6, 7, 8 paraiškų aptarnavimas, Jis laisvas nuo to momento, kol neaptarnaujama 1000 paraiškų. Ir toliau modeliuojamoji aptarnavimo sistema veiks tokiu pat principu – aptarnaujantis įrenginys užimtas, po to laisvas, vėl užimtas, vėl laisvas ir t.t.

Ciklu vadinsime laiko tarpą, sudarytą iš užimtumo periodo ir po jo einančio prastovos periodo. Tada gauta trajektorija susideda iš penkių pilnų ciklų, kuriems vykstant sistema aptarnauja tokius ateinančius paraiškų rinkinius:

{1, 2}, {3, 4, 5, 6, 7, 8}, {9}, {10, 11, 12}, {13, 14, 15, 16, 17, 18, 19}

Šeštasis ciklas prasideda 20 -tosios paraiškos atėjimu. Tokiu būdu, kiekvieno naujo ciklo pradžią užduoda paraiška, randanti aptarnavimo įrenginį laisvą.

Pastebėkime, kad sistema „atsistato“ (šis terminas nereiškia, sakysim, prietaiso atsistatymo po sugedimo, o tik apibrėžia sistemos grįžimą į tam tikras fiksuotas būsenas) kiekvieno ciklo pradžioje iki būsenos, kurioje buvo, kai atėjo 1-oji paraiška, t.y. ciklo pradžios momentu tolimesnis modeliuojamos sistemos elgesys nepriklauso nuo jos elgesio praeityje ir pasiskirsto pagal tą patį tikimybinį dėsnį, kaip ir tuo momentu, kada pirmoji paraiška patenka į laisvą aptarnaujantį įrenginį, t.y. kiekvieno ciklo pradžia neatsiejama nuo momento, kada pati pirmoji paraiška ateina į sistemą. Taigi, natūralu grupuoti modeliavimo rezultatus: pirmoji grupė sudaryta iš laukimo laiko reikšmių pirmajame cikle, antroji –iš laukimo laiko eilėje reikšmių antrajame cikle ir t.t. Tos grupės tokios:

$\{W_1, W_2\}$,

$\{W_3, W_4, W_5, W_6, W_7, W_8\}$,

$\{W_9\}$,

$\{W_{10}, W_{11}, W_{12}\}$,

$\{W_{13}, W_{14}, W_{15}, W_{16}, W_{17}, W_{18}, W_{19}\}$.

Kadangi kiekvienas ciklas prasideda esant toms pačioms sąlygoms ir šiais momentais sistema regeneruoja, o ciklų duomenų grupės statistiškai nepriklausomos ir turi vienodus skirstinius. Taigi, jeigu sakysime, kad Y_k reiškia laukimo laikų sumą k-tajame cikle, o α_k – k-tajame cikle aptarnautų paraiškų skaičius, tai poros

$$(Y_1, \alpha_1), (Y_2, \alpha_2), (Y_3, \alpha_3), (Y_4, \alpha_4) \text{ ir } (Y_5, \alpha_5)$$

yra nepriklausomos ir vienodai pasiskirsčiusios. Tačiau Y_k ir α_k priklausomi.

Iš 1.1.3.2 pav. gauname

$$\begin{aligned}(Y_1,\alpha_1)&=(10,\,2),\\(Y_2,\alpha_2)&=(130,\,6),\\(Y_3,\alpha_3)&=(0,1),\\(Y_4,\alpha_4)&=(50,\,3),\\(Y_5,\alpha_5)&=(180,\,4).\end{aligned}$$

Tokiu būdu stipriai koreliuotus duomenis { W_1, W_2,..., W_{19} } suskirstome į grupes, kurios yra nepriklausomos ir vienodai pasiskirsčiusios.

Visa tai įtakoja sudėtingų problemų sprendimą, su kuriomis susiduriama analizuojant modeliavimo rezultatus, t.y. gaunant pagrįstą dydžio E{W} įvertį. Tarkime, atnaujiname modeliavimą nuo 20-osios paraiškos ir tęsiame, kol negausime *n* pilnų ciklų. Jeigu N – bendras per *n* ciklų aptarnautų paraiškų skaičius, tai

$$\frac{W_1+W_2+\cdots+W_N}{N}=\frac{W_1+W_2+\cdots+W_n}{a_1+\alpha_2+\cdots+\alpha_n} \qquad (1.1.3.2)$$

Dešinę (2) pusę galima užrašyti ir tokiu pavidalu

$$\frac{(Y_1+Y_2+\cdots+Y_n)/n}{(\alpha_1+\alpha_2+\cdots+\alpha_n)/n} \qquad (1.1.3.3)$$

Kadangi kiekvienas rinkinys $\{Y_1,Y_2,\ldots,Y_n\}$ ir $\{\alpha_1,\alpha_2,\ldots,\alpha_n\}$susideda iš vienodai pasiskirsčiusių atsitiktinių dydžių, tai pagal stiprųjį didžiųjų skaičių dėsnį, kai $n\to\infty$, (1.1.3.3) skaitiklis artėja į E{Y_1}, o vardiklis – į E{α_1} su tikimybe lygia 1. Pastebėkime, kad $N\geq n$ ir $N\to\infty$, kai $n\to\infty$. Be to, kairioji (1.1.3.2) pusė su tikimybe 1 artėja į E{W}, kai $N\to\infty$.

Taigi

$$E\{W\}=\frac{E\{Y\}}{E\{\alpha\}}$$

Tokiu būdu, E{W} įvertinimo užduotis tampa ekvivalenti santykio $\frac{E\{Y\}}{E\{\alpha\}}$ įvertinimui. Ir kadangi šis santykis gali būti įvertintas pagal nepriklausomas ir vienodai pasiskirsčiusias poras $(Y_1,\alpha_1),\ldots,(Y_n,\alpha_n)$, todėl galima pasinaudoti klasikine statistika ir apskaičiuoti santykio $\frac{E\{Y\}}{E\{\alpha\}}$ pasikliautinąjį intervalą.

Taip gavome paprastą sprendimą, kaip elgtis su stipriai koreliuotais aptarnavimo sistemos modeliavimo rezultatais. Be to, tarę, kad W_1=0, modeliavimą pradedame nuo naujo ciklo, todėl nereikalingas joks stabilizacijos periodas. Tam, kad gauti statistiškai reikšmingą E{W}įvertį tinkama kiekviena trajektorijos atkarpa.

Taigi, matome, kad šis imitacinis eksperimentas padėjo suprasti kaip spręsti sunkias „taktines“ problemas, su kuriomis susidūrėme pradžioje, o būtent, kaip pradėti modeliavimą, nuo kurio momento rinkti duomenis, ką daryti su stipriai koreliuotais duomenimis.

Sistemos efektyvumo kriterijai – vidutinio paraiškos laukimo eilėje laiko iki aptarnavimo pradžios pasikliautinasis intervalas ir vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas. Jie buvo skaičiuojami, esant trims skirtingiems paraiškų atėjimo srautams ir laukimo eilėje tipams:

- paraiškos sistemoje pasirodo determinuotai; paraiškų laukimo laiko trukmės pasiskirstę pagal tolygųjį dėsnį;
- paraiškų atėjimo ir laukimo laiko trukmės pasiskirstę pagal tolygųjį skirstinį;
- paraiškų atėjimo ir laukimo laiko trukmės pasiskirstę pagal eksponentinį dėsnį.

1.2 REGENERACINIS METODAS

Regeneracinis metodas tai – stochastinių sistemų, kurios pasižymi savybe laikas nuo laiko regeneruotis, modeliavimo rezultatų analizės būdas. Jei modeliavimo rezultatai traktuojami kaip stochastinis procesas, tai regeneraciniai procesai turi savybę nuolat grįžti į tam tikrą regeneracijos tašką, kuriuo pradedant tolesnis proceso vystymasis nepriklauso nuo jo elgesio praeityje ir apibrėžiamas tuo pačiu tikimybiniu skirstiniu. Po to jeigu modeliavimo rezultatai grupuojami atitinkamai pagal nuoseklius sugrįžimus į regeneracijos tašką, tai tos grupės yra nepriklausomos ir vienodai pasiskirsčiusios, kas labai palengvina jų statistinę analizę. Šis metodas taikomas bet kokiai diskretaus modeliavimo schemai, kurią galima traktuoti kaip regeneracinį procesą. Diskrečiu vadiname tokį modeliavimo procesą, kuriame imituojami sistemos būklės pokyčiai tik diskrečiais laiko momentais.

1. 2. 1 REGENERACINIAI PROCESAI DISKREČIAIS LAIKO MOMENTAIS

Atsitiktinių K išmatavimo vektorių seka $\{X_{n,}\ n \geq 1\}$ yra regeneracinis procesas, jei egzistuoja didėjanti atsitiktinių diskrečių laiko momentų seka $1 \leq \beta_1 < \beta_2 < \ldots$, tokia, kad proceso vystymasis, pradedant nuo kiekvienu šių momentų, apibrėžiamas tuo pačiu tikimybiniu skirstiniu, kaip ir momentu β_1. Tai reiškia, kad tarp bet kurių dviejų gretimų sekos regeneracijos momentų, β_j ir β_{j+1}, proceso dalis

$$\{\mathbf{X}_n, \beta_j \leq n < \beta_{j+1}\}$$

yra nepriklausoma „tikimybinė kopija“ kitos proceso dalies tarp bet kurių kitų dviejų gretimų regeneracijos momentų. Tačiau proceso daliai, esančiai tarp momento 1 ir momento β_1, nors ir nepriklausomai nuo kitų dalių, leidžiamas skirtingas pasiskirstymas. Proceso dalį $\{\mathbf{X}_n, \beta_j \leq n < \beta_{j+1}\}$ vadinsime j-tuoju ciklu.

Modeliuojamoje dvikanalėje ir trikanalėje aptarnavimo sistemoje $\mathbf{X}_n = \mathbf{W}_n$ regeneracijos momentais $\{\beta_j, j \geq 1\}$ laikysime tų paraiškų eilės numerius, kurios atėjimo momentu randa aptarnavimo įrenginį laisvą. Tarkime, kad

$$\alpha_j = \beta_{j+1} - \beta_j, \quad j \geq 1.$$

Pastebėsime, kad *regeneracijos periodai* $\{\alpha_j, j \geq 1\}$ tarp sekos regeneracijos momentų nepriklausomi ir vienodai pasiskirstę.

Toliau tarsime, kad $E\{\alpha_j\} < \infty$. Tai nėra pernelyg apribojanti sąlyga. Ji teisinga beveik kiekvienoje eilių teorijos sistemoje.

Regeneracijos savybė – galingas aukščiausio lygio instrumentas analitiniams proceso $\{X_{n,}\ n \ge 1\}$ rezultatams gauti. Svarbiausia tai, kad iš tikrųjų bet kuris regeneracinis procesas diskrečiais laiko momentais, tam tikra prasme pasiskirstęs stacionariai ir dažniausiai tokia įprastine prasme: egzistuoja k-matis atsitiktinis vektorius **X** toks, kad seka $\mathbf{X_n}$ konverguoja į **X**, kai $n \to \infty$, t.y.

$$\lim_{n\to\infty} P\{X_n \le x\} = P\{X \le x\}. \tag{1.2.1.1}$$

Kompaktiškai tai pateiksime tokiu būdu:

$$\mathbf{X}_n \Rightarrow \mathbf{X}, \text{ kai } n \to \infty.$$

Kadangi dabar žinome, kad regeneracinių modelių atveju skirstinys stacionarus, galime įvertinti jų charakteristikas. Tarkime, kad modeliavimo tikslas yra statistikos $r \equiv E\{f(X)\}$ įvertinimas. (Čia *f* išmatuojama funkcija).

Tam pailiustruoti, tarkime, kad **X** – tikrinis dydis, todėl keičiame **X** į X. Jeigu *f(x) = x,* $\forall\, x$, tai

$$r \equiv E\{f(X)\} = E\{X\};$$

tokiu būdu, r įvertinimas ekvivalentus $E\{X\}$ įvertinimui.

Peržiūrėsime tuos regeneracijos tyrimus, kurie bus naudojami gaunant dydžiui *r* pasikliautinąjį intervalą. Sakykime, kad

$$Y_j = \sum_{i=\beta_j}^{\beta_{j+1}-1} f(X_i)$$

t.y. Y_j – tai dydžių $f(X_j)$ suma *j*-ajame cikle, $\alpha_j = \beta_{j+1} - \beta_j$ – j-tojo ciklo ilgis. Tada fundamentali regeneracinio ciklo savybė, kurią naudosime, apibrėžiama teiginiais (1.2.1.2) ir (1.2.1.3).

<u>Teiginys</u>. Tarkime, kad seka $\{(\mathbf{Y}_j, \alpha_j), j \ge 1\}$ susideda iš nepriklausomų ir vienodai pasiskirsčiusių atsitiktinių vektorių.

Jeigu

$$E\{|f(X)|\} < \infty \tag{1.2.1.2}$$

tai

$$r \equiv E\{f(X)\} = \frac{E\{Y_1\}}{E\{\alpha_1\}} \tag{1.2.1.3}$$

Žinome, kad (1.2.1.2) teisingas, kadangi vektoriai $\{(\mathbf{Y}_j, \alpha_j), j \ge 1\}$ charakterizuoja regeneracijos sekos ciklus, o tie ciklai nepriklausomi ir vienodai elgiasi tikimybine prasme. Tiriant (1.2.1.3), aišku, kad, esant pakankamai bendroms sąlygoms, su tikimybe lygia 1

$$\sum_{j=1}^{N} Y_j = \frac{f(X_1) + \cdots + f(X_N)}{N} \to E\{f(X)\}, \text{ kai } N \to \infty. \tag{1.2.1.4}$$

Tarkime, $\beta_1 = 1$, taigi pirmasis regeneracijos ciklas prasideda nuo pačios modeliavimo pradžios. Jeigu *n*-tasis ciklas baigiasi *N* momentu, tada santykį (1.2.1.4) galime užrašyti taip:

$$\frac{(Y_1 + \ldots Y_n)/n}{(\alpha_1 + \ldots + \alpha_n)/n}, \tag{1.2.1.5}$$

kadangi Y_j reikšmių $f(X_i)$ suma j-tajame cikle, o α_j j-tojo ciklo ilgis. Bet santykis (1.2.1.5) su tikimybe lygia 1 konverguoja į $\frac{E\{Y_1\}}{E\{\alpha_1\}}$, kai $n \to \infty$ (dėl (1.2.1.2) ir didžiųjų skaičių dėsnio), o kadangi $n \to \infty$, kai $N \to \infty$, gauname esminį santykį

$$E\{f(X)\} = \frac{E\{Y_1\}}{E\{\alpha_1\}}.$$

Tuo atveju, kai $\beta_1 > 1$ (tai yra ciklas neprasideda iš karto), santykį (1.2.1.5) būtina pakeisti, taip:

$$\frac{(Y_0 + Y_1 + \ldots Y_n)/n}{(\alpha_0 + \alpha_1 + \ldots + \alpha_n)/n}, \tag{1.2.1.5`}$$

čia α_0 ir Y_0 atitinka pradinį „pereinamąjį periodą“ modeliuojant ir yra tokie

$$\alpha_0 = \beta_1 - 1 \text{ ir } Y_0 = f(X_1) + \ldots + f(X_{\beta_1 - 1}).$$

Atsitiktinis dydis α_0 pasiskirstęs kitaip nei α_j, $j \geq 1$, o Y_0 kitaip nei Y_j, $j \geq 1$. Tačiau esant dideliems *n*, *Y₀* bus mažas lyginant su $Y_1 + Y_2 + \ldots + Y_n$, o α_0 bus mažas lyginant su $\alpha_1 + \alpha_2 + \ldots + \alpha_n$, todėl santykis (1.2.1.5`) vis dėlto konverguoja į $\frac{E\{Y_1\}}{E\{\alpha_1\}}$, ir (1.2.1.3) teisingas.

Prielaida $E\{|f(X)|\} < \infty$ nėra labai apribojanti ir nesudaro kliūčių regeneracinio metodo naudojimui.

Taigi, dėl nepriklausomų ir vienodai pasiskirsčiusių stebėjimų santykio $\frac{E\{Y_1\}}{E\{\alpha_1\}}$ įvertinimui galima naudoti klasikinės statistikos rezultatus. Šiam dydžiui galima gauti pasikliautinąjį intervalą.

1. 2. 2 PASIKLIAUTINIEJI (TIKIMYBINIAI) VIDUTINIO BUVIMO LAIKO EILĖJE INTERVALAI

Užduotis: turint nepriklausomas ir vienodai pasiskirsčiusias poras $(Y_{1,}\alpha_1),(Y_{2,}\alpha_2),\ldots,(Y_{n,}\alpha_n)$ apskaičiuoti $100(1-\delta)\%$-tinį pasikliautinąjį intervalą dydžiui $\dfrac{E\{Y_1\}}{E\{\alpha_1\}}$, kai n pakankamai didelis. Vienas iš tokių metodų šiam intervalui gauti remiasi centrine ribine teorema.

Žymime:

$$V_j = Y_j - r\alpha_j .$$

Dydžiai V_j – nepriklausomi ir vienodai pasiskirstę ir $E\{V_j\} = E\{Y_j\} - rE\{\alpha_j\} = 0$. Dydžiai $\overline{Y}, \overline{\alpha}, \overline{V}$ reiškia empirinius vidurkius:

$$\overline{Y} = \frac{1}{n}\sum_{j=1}^{n} Y_j ,$$

$$\overline{\alpha} = \frac{1}{n}\sum_{j=1}^{n} \alpha_j ,$$

$$\overline{V} = \frac{1}{n}\sum_{j=1}^{n} V_j .$$

Pastebėsime, kad $\overline{V} = \overline{Y} - r\overline{\alpha}$.

Žymime $\sigma^2 = E\{V_j^2\}$.

Tada, tarus, kad $0 < \sigma^2 < \infty$, iš centrinės ribinės teoremos išplaukia, kad kiekvienam baigtiniam x

$$\lim_{n\to\infty} P\left\{\frac{\sqrt{n}\overline{V}}{\sigma} \le x\right\} = \Phi(x); \tag{1.2.2.1}$$

čia Φ – standartinė normalioji skirstinio funkcija.

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-y^2}{2}} dy .$$

Prielaida, kad $0 < \sigma^2 < \infty$, neapriboja modeliavimo taikymo.

Toliau iš (1.2.2.1) turime

$$\lim_{n\to\infty} P\left\{\frac{\sqrt{n}\left|\overline{Y} - r\overline{\alpha}\right|}{\sigma} \le x\right\} = \Phi(x),$$

arba

$$\lim_{n\to\infty} P\left\{\frac{\sqrt{n}|\hat{r}-r|}{\sigma/\overline{\alpha}} \le x\right\} = \Phi(x); \tag{1.2.2.2}$$

čia $\hat{r} = \dfrac{\overline{Y}}{\overline{\alpha}}$.

Tačiau betarpiškai iš (1.2.2.2) pasikliautinojo intervalo gauti negalime, lygiai kaip ir negalime sužinoti σ reikšmės. Tačiau σ galime įvertinti tam tikru būdu:

$\sigma^2 = E\left\{(Y_1 - r\alpha_1)^2\right\} = DY_1 - 2r\operatorname{cov}(Y_{1,}\alpha_1) + r^2 D\alpha_1.$

Tarkime, kad s_{11}, s_{22}, s_{12} yra atitinkamai Y_j empirinė dispersija, α_j empirinė dispersija ir (Y_j, α_j) koreliacinis momentas, t.y.

$$s_{11} = \frac{1}{n-1}\sum_{j=1}^{n}\left(Y_j - \overline{Y}\right)^2 = \frac{1}{n-1}\sum_{j=1}^{n} Y_j^2 - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} Y_j\right)^2,$$

$$s_{22} = \frac{1}{n-1}\sum_{j=1}^{n}\left(\alpha_j - \overline{\alpha}\right)^2 = \frac{1}{n-1}\sum_{j=1}^{n} \alpha_j^2 - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} \alpha_j\right)^2,$$

$$s_{12} = \frac{1}{n-1}\sum_{j=1}^{n}\left(Y_j - \overline{Y}\right)^2\left(\alpha_j - \overline{\alpha}\right) = \frac{1}{n-1}\sum_{j=1}^{n} Y_j\alpha_j - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} Y_j\right)\left(\sum_{j=1}^{n} \alpha_j\right).$$

Tegu $s^2 = s_{11} - 2\hat{r}s_{12} + \hat{r}^2 s_{22}$. Tada sus tikimybe 1 $s^2 \to \sigma^2$, kai $n \to \infty$.

Todėl (1.2.2.2) pakeičiame σ į s, t.y.

$$\lim_{n\to\infty} P\left\{\frac{\sqrt{n}|\hat{r}-r|}{s/\overline{\alpha}} \le x\right\} = \Phi(x), \tag{1.2.2.1}$$

Žymime $z_\delta^* = \Phi^{-1}\left(1 - \dfrac{\delta}{2}\right)$, t.y. $\Phi\left(z_\delta^*\right) = 1 - \dfrac{\delta}{2}$.

Iš (2.2.2.3) dideliems n turime:

$$P\left\{-z_\delta^* \le \frac{\sqrt{n}|\hat{r}-r|}{s/\overline{\alpha}} \le z_\delta^*\right\} \cong 1 - \delta.$$

Tai galima perrašyti taip:

$$P\left\{\hat{r} - \frac{z_\delta^* s}{\overline{\alpha}\sqrt{n}} \le r \le \hat{r} + \frac{z_\delta^* s}{\overline{\alpha}\sqrt{n}}\right\} \cong 1 - \delta. \tag{1.2.2.3`}$$

Tokiu būdu gaunamas $100(1-\delta)\%$ -tinis pasikliautinasis intervalas dydžiui $r \equiv E\{f(X)\}$:

$$\hat{I} = \left[\hat{r} - \frac{z_\delta^* s}{\overline{\alpha}\sqrt{n}}, \hat{r} + \frac{z_\delta^* s}{\overline{\alpha}\sqrt{n}}\right]. \tag{1.2.2.4}$$

Jeigu dideliems n per $\hat{J}$ išreikštume intervalo $\hat{I}$ ilgį, tai su didele tikimybe

$$\hat{J} \cong \frac{2z_\delta^*\sigma}{E\{\alpha_1\}\sqrt{n}}.$$

Todėl tam, kad būtų sumažintas pasikliautinojo intervalo $\hat{I}$ ilgis du kartus (esant tokiam pačiam patikimumo lygiui), būtina keturis kartus padidinti modeliuojamų ciklų skaičių.

1. 3 MAKSIMALUS STACIONARIOS EILĖS ILGIS

Apžvelgsime aptarnavimo vienkanalės sistemos paprasčiausią modelį.

Tegu momentu $t_0 = 0$ į sistemą ateina pirmoji paraiška. Likusios paraiškos ateina momentais t_n (n $\geq$ 0). Tarkime, kad dydžiai $t_n - t_{n-1}$ bus nepriklausomi ir turės tą pačią pasiskirstymo funkciją $U(x)$ tokią, kad

$$U(0+) = 0 \text{ ir } 1 < a = \int_0^\infty x dU(x) < \infty .$$

Jeigu momentu t_n sistemoje paraiškų nėra, tai į sistema ateina (n +1)-oji paraiška. Priešingu atveju ši paraiška stoja į eilę. Tariame, kad paraiškos aptarnavimo laikai T_n (n $\geq$ 0) yra nepriklausomi standartiniai eksponentiniai atsitiktiniai dydžiai ir nepriklauso nuo $t_n - t_{n-1}$.

Matome, kad esant tokioms prielaidoms, vienos paraiškos aptarnavimo trukmės tikimybė lygi vienetui. Iš čia išplaukia, kad ribojimas $a > 1$ reiškia, kad vidutinis aptarnavimo laikas yra mažesnis už vidutinį intervalo tarp gretimų paraiškų atėjimo į sistemą momentų ilgį. Iš didžiųjų skaičių dėsnio (DSD) iš karto išplaukia egzistavimas su tikimybe 1 tokio baigtinio momento t^*, kai aptarnavimo įrenginys taps laisvas. Periodas (0, t^*) vadinamas užimtumo periodu.

Kai ateina pirmoji paraiška po pirmojo t^* momento procesas pasikartoja, todėl galime kalbėti apie antrąjį ir sekančius užimtumo periodus. Mūsų atveju užimtumo periodai yra nepriklausomi ir vienodai pasiskirstę (n. v. p.) atsitiktiniai dydžiai.

Mus domina maksimalus eilės ilgis, t.y. jei Y_k reiškia paraiškų skaičių sistemoje prieš ateinant k-tajai paraiškai pagal eiliškumą, tai mums yra įdomus dydis

$$Q_n = \max_{1 \leq k \leq n} Y_k .$$

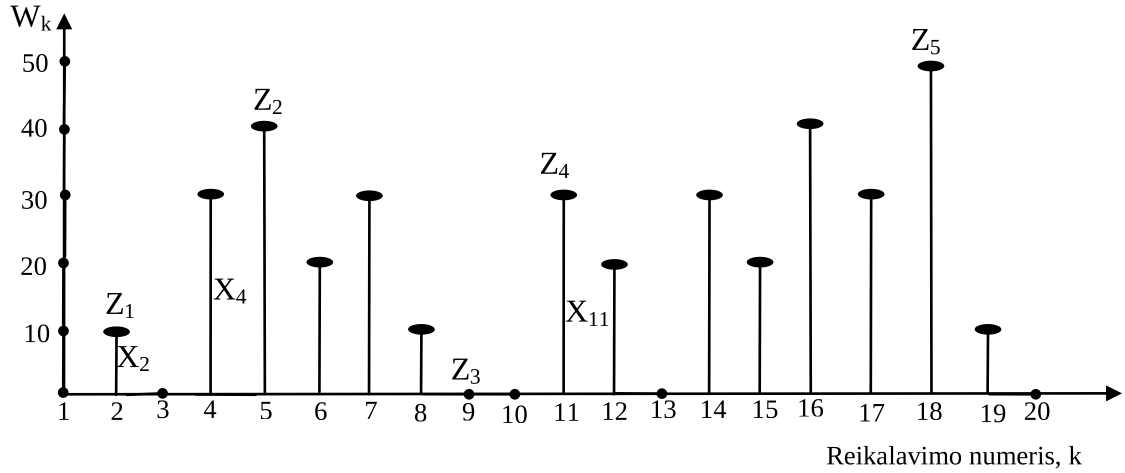
Kadangi dydžiai Y_k yra priklausomi, tektų spręsti sudėtinga uždavinį. Tačiau šią problemą galima išspręsti paprastesniu būdu.

Tegu $N(n)$ – užimtumo periodų, įvykusių iki n-tos paraiškos atėjimo, skaičius, o X_j + 1, maksimalaus ilgio eilė, užfiksuota j-ame užimtumo periode. Kaip jau minėjome anksčiau, dydžiai X_j nepriklausomi ir vienodai pasiskirstę, ir yra akivaizdžios nelygybės

$$Z_{N(n)} \leq Q_n \leq Z_{N(n)+1},$$

čia $Z_N = \max(X_1, X_2, \ldots, X_N)$. Tokiu būdu susidūrėme su nauja problema – dydžio $Z_{N(n)}$ asimptotinio elgesio tyrimo, kai *N(n)* – atsitiktinis dydis, įgyjantis sveikas reikšmes [6].

1. 3. 1 PASIKLIAUTINIEJI (TIKIMYBINIAI) VIDUTINIO MAKSIMALAUS EILĖS ILGIO INTERVALAI



1.3.1.1 pav. Paraiškų laukimo eilėje laikai

<u>Užduotis:</u> turint nepriklausomus ir vienodai pasiskirsčiusius dydžius $Z_k = \max\left(X_1, X_2, \ldots, X_{n_k}\right)$ apskaičiuoti $100(1-\delta)\%$-tinį pasikliautinąjį intervalą dydžiui $E\{Z\}$, kur $Z = \dfrac{Z_1 + Z_2 + \cdots + Z_n}{n}$, kai n pakankamai didelis. Vienas iš tokių metodų tokiam tikimybiniam intervalui gauti remiasi centrine ribine teorema.

Tarus, kad $0 < \sigma^2 < \infty$, iš centrinės ribinės teoremos išplaukia, kad kiekvienam baigtiniam x

$$\lim_{n\to\infty} P\left\{ \frac{\sqrt{n}\left|\hat{\bar{Z}} - MZ\right|}{\sigma} \le x \right\} = \Phi(x); \tag{1.3.1.1}$$

čia Φ – standartinė normalioji skirstinio funkcija.

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-y^2}{2}} dy .$$

Prielaida, kad $0 < \sigma^2 < \infty$, neapriboja modeliavimo taikymo.

Tada

$$\lim_{n\to\infty} P\left\{ \frac{\sqrt{n}\left|\hat{r}_1 - r_1\right|}{\sigma} \le x \right\} = \Phi(x); \tag{1.3.1.2}$$

čia $\hat{r}_1 = \hat{\bar{Z}}$, o $r_1 = MZ$.

Tačiau betarpiškai iš (1.3.1.2) pasikliautinojo intervalo gauti negalime, lygiai kaip ir negalime sužinoti σ reikšmės. Tačiau σ galime įvertinti tam tikru būdu:

$$\sigma^2 = DZ_1 .$$

Tegu $s^{`2}$ reiškia Z_j empirinę dispersiją

$$s^{`2} = \frac{1}{n-1}\sum_{j=1}^{n}\left(Z_j - \overline{Z}\right)^2 = \frac{1}{n-1}\sum_{j=1}^{n} Z_j^2 - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} Z_j\right)^2,$$

Tada sus tikimybe 1 $s^{`2} \to \sigma^2$, kai $n \to \infty$.

Todėl (1.3.1.2) pakeičiame σ į s`, t.y.

$$\lim_{n\to\infty} P\left\{\frac{\sqrt{n}\left|\hat{r}_1 - r_1\right|}{s`} \le x\right\} = \Phi(x), \tag{1.3.1.3}$$

Žymime $z_\delta^* = \Phi^{-1}\left(1 - \frac{\delta}{2}\right)$, t.y. $\Phi\left(z_\delta^*\right) = 1 - \frac{\delta}{2}$.

Iš (1.3.1.3) dideliems n turime:

$$P\left\{- z_\delta^* \le \frac{\sqrt{n}\left|\hat{r}_1 - r_1\right|}{s`} \le z_\delta^*\right\} \cong 1 - \delta .$$

Tai galima perrašyti taip:

$$P\left\{- z_\delta^* \le \frac{\sqrt{n}\left|\hat{r}_1 - r_1\right|}{s`} \le z_\delta^*\right\} \cong 1 - \delta . \tag{1.3.1.3`}$$

Tokiu būdu gaunamas $100(1-\delta)\%$ -tinis pasikliautinasis intervalas dydžiui $r_1 \equiv E\{f_1(X)\}$, kur $f_2(X) = \frac{Y_1 + \cdots + Y_n}{n}$:

$$\hat{I} = \left[\hat{r}_1 - \frac{z_\delta^* s`}{\sqrt{n}}, \hat{r}_1 + \frac{z_\delta^* s`}{\sqrt{n}}\right]. \tag{1.3.1.4}$$

Pastebėkime, kad jeigu dideliems n per $\hat{J}$ išreikštume intervalo $\hat{I}$ ilgį, tai su didele tikimybe

$$\hat{J} \cong \frac{2 z_\delta^* \sigma}{E\{\alpha_1\}\sqrt{n}} . \tag{1.3.1.5}$$

Todėl tam, kad sumažinti pasikliautinojo intervalo $\hat{I}$ ilgį du kartus (esant tokiam pačiam patikimumo lygiui), būtina keturis kartus padidinti modeliuojamų ciklų skaičių.

1.4 KITI GALIMI DYDŽIŲ ĮVERČIAI

Kaip mes jau matėme, regeneracinis metodas stacionarių charakteristikų vertinimui modeliavimo būdu reikalauja vidurkio vertinimo. Vienas iš būdų gauti pasikliautinąjį intervalą dviems vidurkiams buvo parodytas 1.2.2 skyrelyje. Čia mes peržiūrėsime keletą galimų taškinių įverčių ir pasikliautinuosius dydžių intervalus, kuriuos galima naudoti kartu su regeneraciniu metodu [7].

Tarkim, mums duotos stebėjimų poros

$$(Y_1, \alpha_1), (Y_2, \alpha_2), \ldots, (Y_n, \alpha_n),$$

kurios yra nepriklausomos ir vienodai pasiskirsčiusios, ir užduotis yra įvertinti dydį

$$r \equiv \frac{E\{Y_1\}}{E\{\alpha_1\}}.$$

Taip, kaip ir 1.2.2 skyrelyje, tegu

$$\bar{Y} = \frac{1}{n}\sum_{j=1}^{n} Y_j,$$

$$\bar{\alpha} = \frac{1}{n}\sum_{j=1}^{n} \alpha_j,$$

$$s_{11} = \frac{1}{n-1}\sum_{j=1}^{n}\left(Y_j - \bar{Y}\right)^2,$$

$$s_{22} = \frac{1}{n-1}\sum_{j=1}^{n}\left(\alpha_j - \bar{\alpha}\right)^2,$$

$$s_{12} = \frac{1}{n-1}\sum_{j=1}^{n}\left(Y_j - \bar{Y}\right)^2\left(\alpha_j - \bar{\alpha}\right).$$

Tarkim, kad mus domina dydžio r $100(1-\delta)\%$-tinis pasikliautinasis intervalas.

Pirmiausia peržiūrėkime kitokius dydžio r taškinius įverčius [7].

Klasikinis įvertis

$$\hat{r}_c(n) = \frac{\bar{Y}}{\bar{\alpha}};$$

(šis įvertis buvo nagrinėtas skyrelyje 1.2.2);

Bilio įvertis

$$\hat{r}_b(n) = \frac{\bar{Y}}{\bar{\alpha}} \frac{1 + \dfrac{s_{12}}{n\bar{Y}\bar{\alpha}}}{1 + \dfrac{s_{22}}{n\bar{\alpha}^2}};$$

Filerio įvertis

$$\hat{r}_f(n) = \frac{\overline{Y}\overline{\alpha} - k_\delta s_{12}}{\overline{\alpha}^2 - k_\delta s_{22}},$$

čia

$$k_\delta = \frac{\left[\Phi^{-1}(1-\delta/2)\right]^2}{n};$$

Tino įvertis

$$\hat{r}_t(n) = \frac{\overline{Y}}{\overline{\alpha}}\left[1 + \frac{1}{n}\left(\frac{s_{12}}{\overline{Y}\overline{\alpha}} - \frac{s_{22}}{\overline{\alpha}^2}\right)\right].$$

„kišeninio peiliuko" įvertis

$$\hat{r}_j(n) = \frac{1}{n}\sum_{i=1}^{n}\theta_i,$$

čia

$$\theta_i = n\left(\frac{\overline{Y}}{\overline{\alpha}}\right) - (n-1)\left(\frac{\sum\limits_{k\neq i} Y_k}{\sum\limits_{k\neq i}\alpha_k}\right);$$

Visi šie įverčiai yra pagrįsti (artėja prie r su tikimybe 1, kai $n \to \infty$).

Prie dydžio r pasikliautinojo intervalo konstravimo uždavinio galima prieiti keliais būdais.

Tegu, kaip ir 1.2.2 skyrelyje,

$$V_j = Y_j - r\alpha_j,$$

Dydžiai V_j – nepriklausomi ir vienodai pasiskirstę ir $E\{V_j\} = 0$. Tarę, kad $\sigma^2 = E\{V_j^2\}$ ir laikydami, kad $0 < \sigma^2 < \infty$, iš centrinės ribinės teoremos išplaukia, kad

$$\frac{\sum_{i=1}^{n} V_i}{\sigma\sqrt{n}} \Rightarrow N(0,1), \text{ kai } n \to \infty, \qquad (1.4.1)$$

čia $N(0, 1)$ atsitiktinis dydis pasiskirstęs pagal normalųjį skirstinį su nuliniu vidurkiu ir dispersija 1; dviguba rodyklė reiškia konvergavimą pagal tikimybę.

Perrašykime (1.4.1) kitaip

$$\frac{\sqrt{n}\left[\hat{r}_c(n) - r\right]}{\sigma/\overline{\alpha}} \Rightarrow N(0,1), \text{ kai } n \to \infty. \qquad (1.4.2)$$

Kadangi su tikimybe 1

$$\sqrt{n}\left[\hat{r}_c(n) - \hat{r}_b(n)\right] \to 0$$

ir

$$\sqrt{n}\left[\hat{r}_c(n)-\hat{r}_t(n)\right]\to 0,$$

kai $n\to\infty$, galima formulėje (1.4.2) $\hat{r}_c(n)$ pakeisti $\hat{r}_b(n)$ arba $\hat{r}_t(n)$ nekeičiant rezultato. Tačiau visi šie tvirtinimai apie konvergavimą į normalųjį skirstinį priklauso nuo konstantos σ. Daugumai modelių konstanta σ negali būti išskaičiuojama. O tiems atvejams, kai tai įmanoma, išskaičiavimas yra gana sudėtingas. Todėl pagrindiniu uždaviniu konstruojant pasikliautinąjį intervalą yra dydžio σ įvertinimas. Klasikiniame pasikliautinojo intervalo konstravimo metode σ yra pakeistas dydžiu

$$\sqrt{s_{11}-2\hat{r}_c s_{12}+\hat{r}_c^2 s_{22}}\equiv \hat{s}_c(n),$$

Tuomet dydžio r 100 (1 – δ)% - tinis pasikliautinasis intervalas yra

$$\hat{I}_c(n)=\left[\hat{r}_c-\frac{z_\delta^*\hat{s}_c}{\overline{\alpha}\sqrt{n}};\hat{r}_c+\frac{z_\delta^*\hat{s}_c}{\overline{\alpha}\sqrt{n}}\right], \tag{1.4.3}$$

čia

$$z_\delta^*=\Phi^{-1}\left(1-\frac{\delta}{2}\right).$$

Toliau kaip ir (1.4.2) vietoje $\hat{r}_c(n)$ galima įrašyti arba *Bilio įvertį* $\hat{r}_b(n)$, arba *Tino* $\hat{r}_t(n)$ ir taip gauti sekančias dvi pasikliautinojo intervalo išraiškas:

$$\hat{I}_{bc}(n)=\left[\hat{r}_b-\frac{z_\delta^*\hat{s}_c}{\overline{\alpha}\sqrt{n}};\hat{r}_b+\frac{z_\delta^*\hat{s}_c}{\overline{\alpha}\sqrt{n}}\right], \tag{1.4.4}$$

$$\hat{I}_{tc}(n)=\left[\hat{r}_t-\frac{z_\delta^*\hat{s}_c}{\overline{\alpha}\sqrt{n}};\hat{r}_t+\frac{z_\delta^*\hat{s}_c}{\overline{\alpha}\sqrt{n}}\right]. \tag{1.4.5}$$

Filerio metode dydis σ keičiamas dydžiu

$$\sqrt{s_{11}-2rs_{12}+r^2 s_{22}}\equiv \hat{s}_f(n),$$

Todėl (1.4.2) perrašomas taip

$$\frac{\overline{\alpha}\sqrt{n}\left[\dfrac{\overline{Y}}{\overline{\alpha}}-r\right]}{\sqrt{s_{11}-2rs_{12}+r^2 s_{22}}}\Rightarrow N(0,1),\ \text{kai}\ n\to\infty.$$

Tuomet dydžio r 100(1 – δ)% - tinis pasikliautinasis intervalas yra:

$$\hat{I}_f(n)=\left[\hat{r}_f-\frac{\sqrt{D}}{\overline{\alpha}^2-k_\delta s_{22}};\hat{r}_f+\frac{\sqrt{D}}{\overline{\alpha}^2-k_\delta s_{22}}\right], \tag{1.4.6}$$

čia

$$D=\left[\overline{Y}\overline{\alpha}-k_\delta s_{12}\right]^2-\left[\overline{\alpha}^2-k_\delta s_{22}\right]\left[\overline{Y}^2-k_\delta s_{11}\right]$$

ir

$$k_\delta = \frac{\left[\Phi^{-1}\left(1-\delta/2\right)\right]^2}{n}.$$

Paskutinis pasikliautinojo intervalo tipas gaunamas iš „kišeninio peiliuko“ metodo. Pažymėkime

$$\hat{s}_j(n) = \left\{ \frac{\sum_{i=1}^{n}\left[\theta_i - \hat{r}_j(n)\right]^2}{n-1} \right\}^{1/2},$$

$$\frac{\hat{r}_j(n) - r}{\sqrt{n}\hat{s}_j(n)} \Rightarrow N(0,1), \text{ kai } n \to \infty.$$

Tada „kišeninio peiliuko“ metodas duoda tokią pasikliautinojo intervalo skaičiavimo išraišką:

$$\hat{I}_j(n) = \left[\hat{r}_j - \frac{z_\delta^* \hat{s}_j}{\sqrt{n}}; \hat{r}_j + \frac{z_\delta^* \hat{s}_j}{\sqrt{n}} \right].$$

„Kišeninio peiliuko“ metodas duoda gana tikslų dydžio r pasikliautinąjį intervalą, tačiau reikalauja daug kompiuterinės atminties ir apsunkina programas. Todėl taškiniams įverčiams labiau rekomenduojami yra *Bilio* ir *Tino* metodai, o intervaliniams įverčiams – klasikinis metodas. Ir taškiniams ir intervaliniams įverčiams suprogramuoti yra lengviausiai klasikinį metodą. Filerio metodas nerekomenduojama nei taškiniams nei intervaliniams įverčiams, kadangi yra žymiais sudėtingesnis už klasikinį [7].

2. TIRIAMOJI DALIS

2.1 PASIKLIAUTINOJO INTERVALO APSKAIČIAVIMO ALGORITMAS

Reziumuojame apytikslio 100(1-δ)%-tinio pasikliautinojo intervalo dydžiams $r \equiv E\{f(X)\}$ ir $r_1 \equiv E\{f_1(X)\}$ gavimo procedūrą.

1. Sumodeliuojame *n* regeneracinių ciklų.
2. Apskaičiuojame Z_j, Y_j ir α_j kiekvienam *j*-tajam ciklui, kur Z_j – maksimalus eilės ilgis *j*-tajame cikle, Y_j – dydžių $f(\mathbf{X}_i)$ suma *j*-tame cikle ir α_j – *j*-tojo ciklo ilgis.
3. Apskaičiuojame statistikas:

$$\overline{Y} = \frac{1}{n}\sum_{j=1}^{n} Y_j\,, \qquad \overline{\alpha} = \frac{1}{n}\sum_{j=1}^{n}\alpha_j\,, \qquad \hat{r} = \frac{\overline{Y}}{\overline{\alpha}}, \qquad \hat{\overline{Z}} = \frac{1}{n}\sum_{j=1}^{n} Z_j\,,$$

$$s_{11} = \frac{1}{n-1}\sum_{j=1}^{n} Y_j^2 - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} Y_j\right)^2,$$

$$s_{22} = \frac{1}{n-1}\sum_{j=1}^{n} \alpha_j^2 - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} \alpha_j\right)^2,$$

$$s_{12} = \frac{1}{n-1}\sum_{j=1}^{n} Y_j\alpha_j - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} Y_j\right)\left(\sum_{j=1}^{n} \alpha_j\right),$$

$$s^{`2} = \frac{1}{n-1}\sum_{j=1}^{n} Z_j^2 - \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} Z_j\right)^2$$

4. Suformuojame pasikliautinuosius intervalus.

$$\hat{r} \pm \frac{z_\delta^* s}{\overline{\alpha}\sqrt{n}} \text{ ir } \hat{r}_1 \pm \frac{z_\delta^* s^`}{\sqrt{n}}$$

čia $z_\delta^* = \Phi^{-1}\left(1 - \frac{\delta}{2}\right)$ ir Φ – standartinė normalioji skirstinio funkcija.

Pastebėsime, kad tuo atveju, kai pirmasis ciklas neprasideda kartu su modeliavimu, aukščiau pateikta procedūra nurodo, kad duomenis iš pirmo ciklo būtina atmesti.

2.2 VIDUTINIO BUVIMO LAIKO PASIKLIAUTINOJO INTERVALO APSKAIČIAVIMO ALGORITMO TAIKYMO PAVYZDYS

Tarkime, kad būtina gauti 90%-tinį pasikliautinąjį intervalą E{W} – vidutiniam paraiškos laukimo laikui stacionariame rėžime modeliuojant aptarnavimo sistemą.

Modelyje buvo gautos tokios paraiškų laukimo laiko reikšmės:

$W_1 = 0$,	$W_{19} = 1$,	$W_{37} = 1$,
$W_2 = 6$,	$W_{20} = 11$,	$W_{38} = 5$,
$W_3 = 0$,	$W_{21} = 8$,	$W_{39} = 4$,
$W_4 = 0$,	$W_{22} = 4$,	$W_{40} = 0$,
$W_5 = 25$,	$W_{23} = 0$,	$W_{41} = 9$,
$W_6 = 7$,	$W_{24} = 10$,	$W_{42} = 11$,
$W_7 = 19$,	$W_{25} = 2$,	$W_{43} = 5$,
$W_8 = 5$,	$W_{26} = 13$,	$W_{44} = 0$,
$W_9 = 7$,	$W_{27} = 3$,	$W_{45} = 3$,
$W_{10} = 11$,	$W_{28} = 6$,	$W_{46} = 37$,
$W_{11} = 0$,	$W_{29} = 20$,	$W_{47} = 8$,
$W_{12} = 16$,	$W_{30} = 1$,	$W_{48} = 7$,
$W_{13} = 4$,	$W_{31} = 8$,	$W_{49} = 2$,
$W_{14} = 10$,	$W_{32} = 1$,	$W_{50} = 5$,
$W_{15} = 1$,	$W_{33} = 11$,	$W_{51} = 4$,
$W_{16} = 1$,	$W_{34} = 5$,	$W_{52} = 0$,
$W_{17} = 4$,	$W_{35} = 7$,	
$W_{18} = 7$,	$W_{36} = 7$,	

1. Šioje realizacijoje 8 ciklai, kurių pradžia paraiškos 1, 3, 4, 11, 23, 40, 44, 52.
2. Toliau skaičiuojame buvimo laiko eilėje sumas kiekviename cikle, bei paraiškų skaičių kiekviename cikle. Gauname tokius rezultatus:

$$\alpha_1 = 2, \qquad Y_1 = \sum_{i=1}^{2} W_i = 6;$$

$$\alpha_2 = 1, \qquad Y_2 = \sum_{i=3}^{3} W_i = 0;$$

$$\alpha_3 = 7, \qquad Y_3 = \sum_{i=4}^{10} W_i = 74;$$

$$\alpha_4 = 12, \qquad Y_4 = \sum_{i=11}^{22} W_i = 67;$$

$$\alpha_5 = 17, \qquad Y_5 = \sum_{i=23}^{39} W_i = 104;$$

$$\alpha_6 = 4, \qquad Y_6 = \sum_{i=40}^{43} W_i = 25;$$

$$\alpha_7 = 8, \qquad Y_7 = \sum_{i=44}^{51} W_i = 66;$$

$$\alpha_8 = 1. \qquad Y_8 = \sum_{i=51}^{52} W_i = 0.$$

3. Apskaičiuojame empyrinius vidurkius, bei vidutinio buvimo laiko eilėje įverčius:

$$\overline{Y} = \frac{1}{8}\sum_{j=1}^{8} Y_j = 42{,}75, \quad \overline{\alpha} = \frac{1}{8}\sum_{j=1}^{8} \alpha_j = 6{,}5, \qquad \hat{r} = \frac{\overline{Y}}{\overline{\alpha}} = 6{,}58,$$

Apskaičiuojame Y_j empirinė dispersija:

$$s_{11} = \frac{1}{7}\sum_{j=1}^{8} Y_j^2 - \frac{1}{56}\left(\sum_{j=1}^{8} Y_j\right)^2 = 1596{,}79.$$

Apskaičiuojame α_j empirinė dispersija:

$$s_{22} = \frac{1}{7}\sum_{j=1}^{8} \alpha_j^2 - \frac{1}{56}\left(\sum_{j=1}^{8} \alpha_j\right)^2 = 32{,}85.$$

Apskaičiuojame (Y_j, α_j) koreliacinį momentą:

$$s_{12} = \frac{1}{7}\sum_{j=1}^{8} Y_j\alpha_j - \frac{1}{56}\left(\sum_{j=1}^{8} Y_j\right)\left(\sum_{j=1}^{8} \alpha_j\right) = 215{,}29.$$

Tuomet empirinės vidutinio buvimo laiko eilėje dispersijos kvadratas bus:

$$s^2 = s_{11} - 2\hat{r}s_{12} + \hat{r}^2 s_{22} = 185{,}83.$$

4. Toliau apskaičiuojame pasikliautinąjį vidutinio buvimo laiko eilėje intervalą:

$$\hat{I} = \hat{r} \pm \frac{z_\delta^* s}{\overline{\alpha}\sqrt{n}} = 6{,}58 \pm \frac{1{,}645 \cdot \sqrt{185{,}83}}{6{,}5 \cdot \sqrt{8}} = 6{,}58 \pm 1{,}22.$$

Taigi pasikliautinasis vidutinio buvimo laiko eilėje intervalas yra:

$$\hat{I} = [5{,}36;\ 7{,}80].$$

2.3 MAKSIMALAUS EILĖS ILGIO PASIKLIAUTINOJO INTERVALO APSKAIČIAVIMO ALGORITMO TAIKYMO PAVYZDYS

Tarkime, kad būtina gauti 90%-tinį pasikliautinąjį intervalą E{Z} – vidutiniam maksimaliam eilės ilgiui stacionariame rėžime, modeliuojant aptarnavimo sistemą.

Modelyje buvo gautos tokios paraiškų laukimo laiko reikšmės:

$W_1 = 0$,	$W_{19} = 1$,	$W_{37} = 1$,
$W_2 = 6$,	$W_{20} = 11$,	$W_{38} = 5$,
$W_3 = 0$,	$W_{21} = 8$,	$W_{39} = 4$,
$W_4 = 0$,	$W_{22} = 4$,	$W_{40} = 0$,
$W_5 = 25$,	$W_{23} = 0$,	$W_{41} = 9$,
$W_6 = 7$,	$W_{24} = 10$,	$W_{42} = 11$,
$W_7 = 19$,	$W_{25} = 2$,	$W_{43} = 5$,
$W_8 = 5$,	$W_{26} = 13$,	$W_{44} = 0$,
$W_9 = 7$,	$W_{27} = 3$,	$W_{45} = 3$,
$W_{10} = 11$,	$W_{28} = 6$,	$W_{46} = 37$,
$W_{11} = 0$,	$W_{29} = 20$,	$W_{47} = 8$,
$W_{12} = 16$,	$W_{30} = 1$,	$W_{48} = 7$,
$W_{13} = 4$,	$W_{31} = 8$,	$W_{49} = 2$,
$W_{14} = 10$,	$W_{32} = 1$,	$W_{50} = 5$,
$W_{15} = 1$,	$W_{33} = 11$,	$W_{51} = 4$,
$W_{16} = 1$,	$W_{34} = 5$,	$W_{52} = 0$,
$W_{17} = 4$,	$W_{35} = 7$,	
$W_{18} = 7$,	$W_{36} = 7$,	

3. Šioje realizacijoje 8 ciklai, kurių pradžia paraiškos 1, 3, 4, 11, 23, 40, 44, 52.
4. Apskaičiuojame maksimalius eilės ilgius kiekviename cikle:

$$Z_1 = \max(X_1, X_2) = 1$$

$$Z_2 = \max(X_3) = 1$$

$$Z_3 = \max(X_4, \ldots, X_{10}) = 3$$

$$Z_4 = \max(X_{11}, \ldots, X_{22}) = 3$$

$$Z_5 = \max(X_{23}, \ldots, X_{39}) = 2$$

$$Z_6 = \max(X_{40}, \ldots, X_{43}) = 2$$

$$Z_7 = \max(X_{44}, X_{51}) = 4$$

$$Z_8 = \max(X_{51}, X_{52}) = 1$$

3. Apskaičiuojame vidutinio maksimalaus eilės ilgio taškinį įvertį:

$$\hat{r}_1 = \frac{1}{8}\sum_{j=1}^{8} Z_j = 2{,}13 .$$

Randame vidutinio maksimalaus eilės ilgio empirinės dispersijos kvadratą:

$$s^{`2} = \frac{1}{7}\sum_{j=1}^{8} Z_j^2 - \frac{1}{56}\left(\sum_{j=1}^{8} Z_j\right)^2 = 1{,}26 .$$

4. Toliau apskaičiuojame pasikliautinąjį vidutinio buvimo laiko eilėje intervalą:

$$\hat{I}_1 = \hat{r}_1 \pm \frac{z_\delta^* s^`}{\sqrt{n}} = 2{,}13 \pm \frac{1{,}645 \cdot \sqrt{1{,}26}}{\sqrt{8}} = 2{,}13 \pm 0{,}65 .$$

Taigi pasikliautinasis vidutinio buvimo laiko eilėje intervalas yra:

$$\hat{I}_1 = [1{,}48;\ 2{,}78] .$$

2.4 KLASIKINIO ĮVERČIO PALYGINIMAS SU KITAIS GALIMAIS DYDŽIŲ ĮVERČIAIS

Atlikdami tyrimą vidutinį buvimo laiką eilėje dvikanalėje trikanalėje aptarnavimo sistemose vertinome pagal klasikinį metodą, t.y. kaip laukimo laikų sumų regeneraciniuose cikluose vidurkio santykį su paraiškų skaičiaus regeneraciniuose cikluose vidurkiu. Kaip jau žinome vidutinis buvimo laikas eilėje gali būti įvertintas ir kitais metodais, tokiais kaip Tino, Bilio, Filerio ir kitais. Šiais metodais gautus įverčius ir pasikliautinuosius intervalus pateikiame 2.3.1 lentelėje.

2.3.1 lentelė

Pasikliautinųjų intervalų palyginimas

Modeliuojamų paraiškų skaičius n = 10		
	Įvertis	**Pasikliautinasis intervalas**
Eksponentinis modelis		
Klasikinis	6,27	(0,00; 12,93)
Tino	6,27	(0,00; 12,93)
Bilio	6,31	(0,00; 12,95)
Filerio	6,16	(5,04; 7,28)
Tolygusis modelis		
Klasikinis	4,45	(0,70; 8,21)
Tino	4,45	(0,70; 8,21)
Bilio	4,46	(0,70; 8,22)
Filerio	4,44	(4,02; 4,86)
Tolygusis-Determinuotas modelis		
Klasikinis	11,45	(0,00; 23,92)
Tino	11,45	(0,00; 23,92)
Bilio	11,63	(0,00; 24,03)
Filerio	11,05	(7,10; 15,01)
Modeliuojamų paraiškų skaičius n = 1000		
	Įvertis	**Pasikliautinasis intervalas**
Eksponentinis modelis		
Klasikinis	23,2	(11,62; 34,8)
Tino	23,2	(11,62; 34,8)
Bilio	23,2	(11,62; 34,8)
Filerio	23,2	(23,17; 23,23)
Tolygusis modelis		
Klasikinis	17,64	(11,33; 23,95)
Tino	17,64	(11,33; 23,95)
Bilio	17,64	(11,33; 23,95)

Filerio	17,64	(17,63; 17,65)
Tolygusis-Determinuotas modelis		
Klasikinis	19,47	(1,29; 24,66)
Tino	19,47	(1,29; 24,66)
Bilio	19,47	(1,29; 24,66)
Filerio	19,47	(19,46; 19,48)

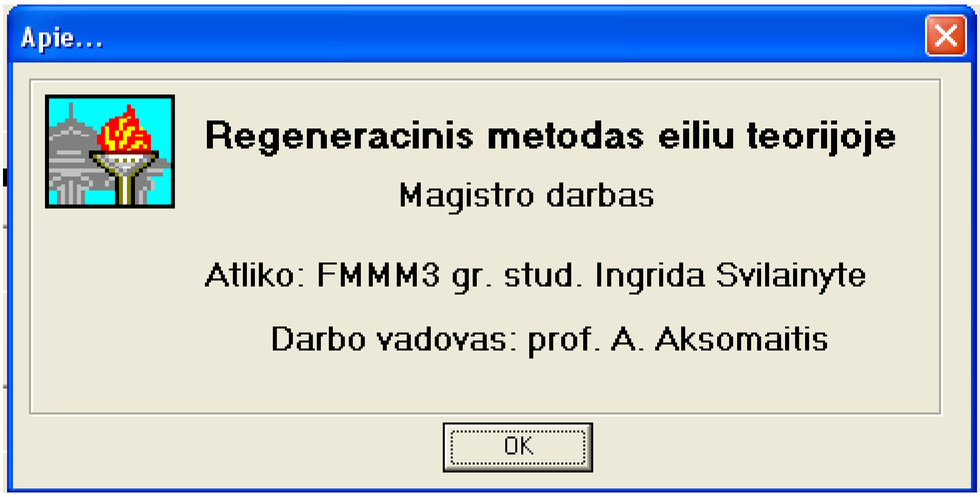
Matome, kad gauti rezultatai yra labai panašūs. Tačiau klasikinį metodą pasitrinkome todėl, kad jis užima pakankamai mažai kompiuterinės atminties ir yra labiausiai rekomenduojamas skaičiuojant pasikliautinuosius intervalus regeneraciniu metodu [7].

3. PROGRAMINĖ REALIZACIJA IR INSTRUKCIJA VARTOTOJUI

Dvikanalė ir trikanalė aptarnavimo sistemos modeliuojamos ir visi skaičiavimai atliekami C++ Builder 5.0 programiniu paketu. Pasirinkome būtent šį programinį paketą, kadangi su juo patogu atlikti modeliavimą ir nesudėtinga sukurti sąsają su vartotoju.

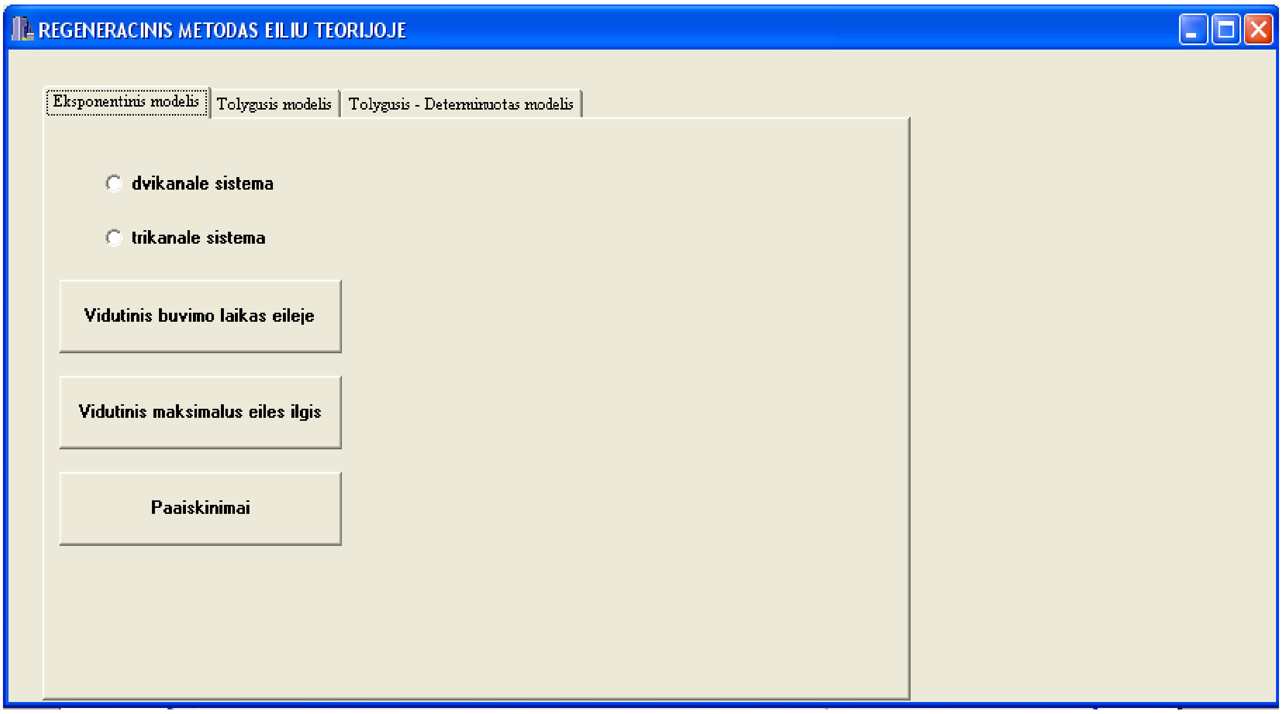
Rezultatai išvedami į bylas „ee.txt“, „tt.txt“, „td.txt“ ir lange.

Programa saugom byloje „Svilainyte.exe“. Atidarius failą pirmiausia ekrane pasirodys pranešimas:



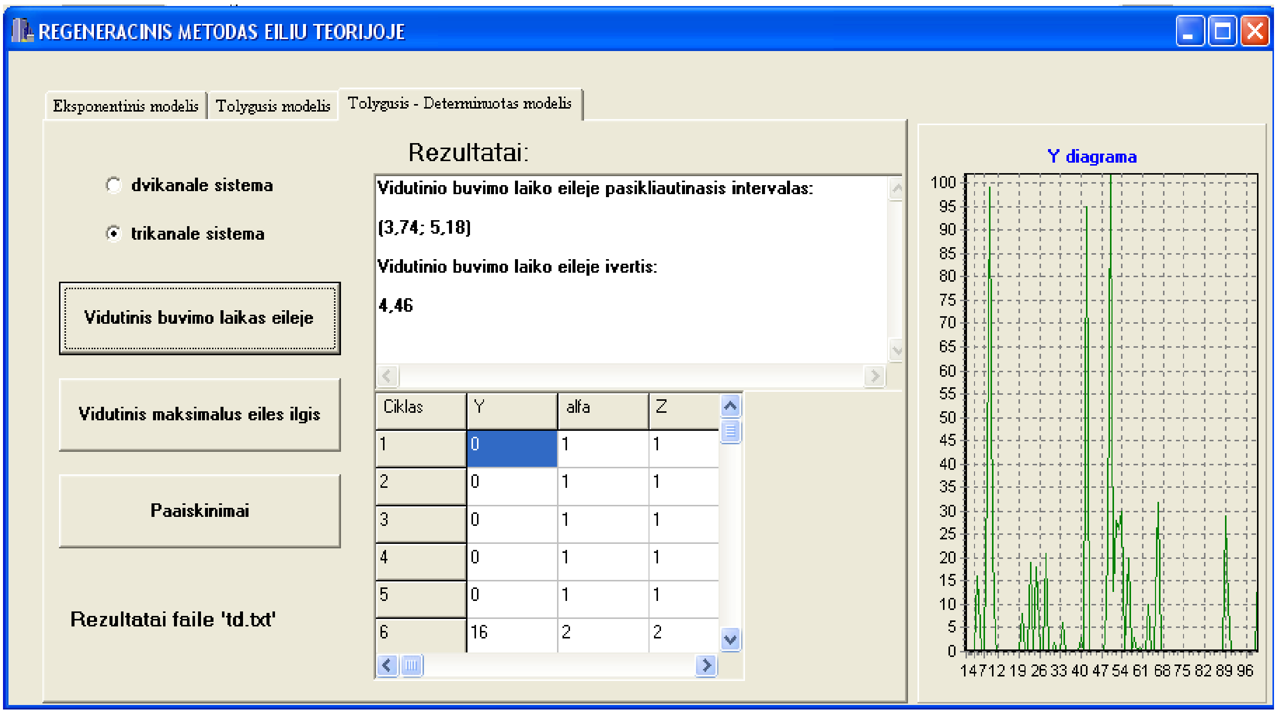
3.1 pav. Pradinis programos pranešimas

Norint, kad būtų atliekami skaičiavimai vartotojas privalo paspausti mygtuką „OK“. Tada ekrane pasirodys pagrindinis programos meniu langas, kuriuo vartotojas valdys programą:



3.2 pav. Programos pagrindinis meniu langas

Pasirinkus atitinkamą puslapį bus atliekami tam tikro modelio skaičiavimai (pvz.: „Tolygus determinuotas modelis" reiškia, kad bus atliekami skaičiavimai su modeliu, kur paraiškos ateina determinuotais laiko momentais, o laukimo laikų eilėje skirstinys bus tolygusis ir pan.). Vartotojas taip pat būtinai turi pasirinkti sistemą kurioje bus atliekami skaičiavimai (pvz.: „trikanalė sistema“ reiškia kad pasikliautinieji intervalai bus apskaičiuojami modelyje su trimis aptarnavimo kanalais).



3.3 pav. Programos rezultatų langas

Lange REZULTATAI rodomas gautojo pasikliautinojo intervalo ilgis ir parametro įvertis (pvz.: pasirinkus mygtuką „Vidutinis buvimo laikas eilėje“ ekrane pasirodys vidutinio buvimo laiko eilėje pasikliautinasis intervalas ir jo įvertis).

Lentelėje pateikiami modeliavimo duomenys: ciklo numeris, paraiškų laukimo laikų ilgių suma atitinkamame cikle, paraiškų skaičius atitinkamame cikle ir maksimalus eiles ilgis atitinkamame cikle.

Grafike vaizduojamas paraiškų laukimo laikų ilgių sumų atitinkamame cikle grafikas.

Paspaudus mygtuką ”✕“ išeinama iš programos.

4. IŠVADOS

1. Regeneracinis metodas pateikia atsakymus į klausimus, kaip pradėti modeliavimą, nuo kurio momento pradėti fiksuoti duomenis, kaip elgtis su stipriai koreliuotais duomenimis;
2. Regeneracinį procesą interpretuojame, kaip sudarytą iš nepriklausomų vienodai pasiskirsčiusių ciklų, kas suteikia galimybę analizuoti taikant gerai ištirtus klasikinės statistikos metodus;
3. Skaičiuojamo pasikliautinojo intervalo tikslumas priklauso nuo modelyje gaunamų ciklų skaičiaus, pasikliovimo lygmens ir laiko trukmės tarp paraiškų atėjimo ir laukimo eilėse skirstinių;

5. LITERATŪROS SĄRAŠAS

1. Aksomaitis A. “Tikimybių teorija ir statistika”. Kaunas: Technologija, 2000.- 450p.;
2. C ++ Builder programinis paketas;
3. Kruopis J. Matematinė statistika. – Vilnius: Mokslo ir enciklopedijų leidykla, 1993. – 411 p.;
4. Pranevičienė I., Pranevičius H. Masinio aptarnavimo teorijos elementai. – Kaunas: Raidė, 1979. – 68p.;
5. Галамбош Я. Асимптотическая теория экстремальных порядковых статистик. М.: Наука. 1984.- 293 p.;
6. Крейн М., Лемуан О. Введение в регенеративный метод анализа моделей. – Москва: Наука, 1982. – 140 с.;

1 PRIEDAS. MODELIAVIMO REZULTATŲ DVIKANALĖJE APTARNAVIMO SISTEMOJE LENTELĖS IR GRAFIKAI

1 lentelė

Modeliavimo rezultatai dvikanalėje aptarnavimo sistemoje

Laukimo laiko eilėje ir paraiškų atėjimo momentai pasiskirstę pagal eksponentinį dėsnį.	
Laukimo laiko momentai pirmame kanale: exp(0,01) Laukimo laiko momentai antrame kanale: exp(0,02) Paraiškų atėjimo momentai: exp(0,02)	Laukimo laiko momentai pirmame kanale: exp(0,2) Laukimo laiko momentai antrame kanale: exp(0,4) Paraiškų atėjimo momentai: exp(0,9)

Ciklas	Y	α	Z	Ciklas	Y	alfa	Z
1	0	1	1	1	0	1	1
2	0	1	1	2	0	1	1
3	0	1	1	3	0	1	1
4	0	1	1	4	0	1	1
5	114	7	2	5	0	1	1
6	0	1	1	6	30	18	9
7	10	2	1	7	0	1	1
8	60	6	2	8	0	1	1
9	0	1	1	9	0	1	1
10	0	1	1	10	0	1	1
11	0	1	1	11	0	1	1
12	0	1	1	12	0	1	1
13	0	1	1	13	0	1	1
14	0	1	1	14	0	1	1
15	0	1	1	15	0	1	1
16	0	1	1	16	0	1	1
17	24	5	2	17	0	1	1
18	0	1	1	18	0	1	1
19	665	15	4	19	0	1	1
20	0	1	1	20	18	10	5
21	59	4	3	21	0	1	1
22	0	1	1	22	0	1	1
23	0	1	1	23	0	1	1
24	0	1	1	24	0	1	1
25	0	1	1	25	0	1	1
26	15	2	1	26	0	1	1
27	0	1	1	27	0	1	1
28	0	1	1	28	0	1	1
29	156	7	3	29	0	1	1
30	0	1	1	30	0	1	1
31	0	1	1	31	0	1	1
32	0	1	1	32	1	2	1
33	0	1	1	33	0	1	1
34	0	1	1	34	0	1	1
35	1364	22	4	35	0	1	1
36	0	1	1	36	23	10	9
37	0	1	1	37	342	64	14
38	105	6	4	38	0	1	1
39	0	1	1	39	0	1	1
40	0	1	1	40	2	3	2
41	0	1	1	41	0	1	1
42	0	1	1	42	0	1	1
43	0	1	1	43	0	1	1
44	0	1	1	44	0	1	1
45	0	1	1	45	0	1	1
46	9	2	1	46	0	1	1
47	0	1	1	47	0	1	1
48	0	1	1	48	0	1	1
49	0	1	1	49	0	1	1
50	1134	19	6	50	0	1	1
51	0	1	1	51	0	1	1
52	0	1	1	52	0	1	1
53	0	1	1	53	5	5	3
54	0	1	1	54	7	5	4
55	1	2	1	55	9	8	5
56	0	1	1	56	0	1	1
57	0	1	1	57	0	1	1
58	0	1	1	58	0	1	1
59	0	1	1	59	0	1	1
60	0	1	1	60	0	1	1

61	0	1	1
62	0	1	1
63	0	1	1
64	0	1	1
65	0	1	1
66	0	1	1
67	0	1	1
68	99	5	2
69	0	1	1
70	62	4	2
71	0	1	1
72	0	1	1
73	19	3	2
74	0	1	1
75	0	1	1
76	0	1	1
77	0	1	1
78	0	1	1
79	0	1	1
80	0	1	1
81	0	1	1
82	0	1	1
83	0	1	1
84	0	1	1
85	0	1	1
86	0	1	1
87	17	2	1
88	0	1	1
89	0	1	1
90	0	1	1
91	0	1	1
92	0	1	1
93	0	1	1
94	36	4	2
95	0	1	1
96	0	1	1
97	0	1	1
98	0	1	1
99	0	1	1
100	0	1	1
101	0	1	1
102	105	5	2
103	0	1	1
104	0	1	1
105	0	1	1
106	0	1	1
107	0	1	1
108	0	1	1
109	0	1	1
110	0	1	1
111	0	1	1
112	0	1	1
113	93	7	2
114	0	1	1
115	0	1	1
116	17	4	2
117	0	1	1
118	0	1	1
119	0	1	1
120	0	1	1
121	0	1	1
122	0	1	1
123	0	1	1
124	0	1	1
125	0	1	1
126	0	1	1
127	0	1	1
128	1	2	1
129	0	1	1
130	0	1	1
131	0	1	1
132	0	1	1
133	0	1	1
134	141	7	3
135	0	1	1
136	168	7	5
137	0	1	1
138	113	8	3
139	0	1	1
140	0	1	1

61	1	2	1
62	0	1	1
63	0	1	1
64	0	1	1
65	1	2	1
66	0	1	1
67	0	1	1
68	0	1	1
69	0	1	1
70	0	1	1
71	0	1	1
72	0	1	1
73	0	1	1
74	0	1	1
75	0	1	1
76	0	1	1
77	0	1	1
78	0	1	1
79	0	1	1
80	0	1	1
81	0	1	1
82	0	1	1
83	4	5	4
84	0	1	1
85	0	1	1
86	0	1	1
87	0	1	1
88	0	1	1
89	0	1	1
90	0	1	1
91	0	1	1
92	0	1	1
93	0	1	1
94	0	1	1
95	0	1	1
96	0	1	1
97	23	13	6
98	0	1	1
99	0	1	1
100	0	1	1
101	0	1	1
102	0	1	1
103	0	1	1
104	3	4	3
105	0	1	1
106	0	1	1
107	0	1	1
108	0	1	1
109	0	1	1
110	0	1	1
111	0	1	1
112	0	1	1
113	25	16	7
114	2	3	2
115	0	1	1
116	0	1	1
117	0	1	1
118	0	1	1
119	0	1	1
120	0	1	1
121	0	1	1
122	0	1	1
123	0	1	1
124	6	4	3
125	0	1	1
126	123	31	10
127	0	1	1
128	0	1	1
129	0	1	1
130	0	1	1
131	0	1	1
132	0	1	1
133	0	1	1
134	0	1	1
135	0	1	1
136	0	1	1
137	0	1	1
138	0	1	1
139	0	1	1
140	1	2	1

141	13	2	1
142	0	1	1
143	0	1	1
144	0	1	1
145	79	4	3
146	0	1	1
147	0	1	1
148	2033	28	3
149	23	3	2
150	0	1	1
151	0	1	1
152	0	1	1
153	0	1	1
154	0	1	1
155	0	1	1
156	0	1	1
157	0	1	1
158	0	1	1
159	0	1	1
160	0	1	1
161	0	1	1
162	19	2	1
163	2	2	1
164	0	1	1
165	0	1	1
166	0	1	1
167	500	15	4
168	0	1	1
169	0	1	1
170	0	1	1
171	0	1	1
172	0	1	1
173	0	1	1
174	480	11	3
175	65	3	2
176	0	1	1
177	7	2	1
178	0	1	1
179	11	2	1
180	0	1	1
181	24	2	1
182	0	1	1
183	0	1	1
184	0	1	1
185	0	1	1
186	0	1	1
187	13	3	2
188	0	1	1
189	0	1	1
190	0	1	1
191	0	1	1
192	17	2	1
193	0	1	1
194	51	4	1
195	0	1	1
196	0	1	1
197	0	1	1
198	0	1	1
199	734	13	6
200	0	1	1
201	0	1	1
202	5	2	1
203	0	1	1
204	0	1	1
205	0	1	1
206	0	1	1
207	0	1	1
208	0	1	1
209	0	1	1
210	0	1	1
211	0	1	1
212	0	1	1
213	0	1	1
214	0	1	1
215	0	1	1
216	0	1	1
217	0	1	1
218	0	1	1
219	0	1	1
220	0	1	1

141	0	1	1
142	0	1	1
143	0	1	1
144	0	1	1
145	0	1	1
146	0	1	1
147	12	9	3
148	0	1	1
149	1	2	1
150	0	1	1
151	0	1	1
152	1	2	1
153	0	1	1
154	0	1	1
155	0	1	1
156	0	1	1
157	0	1	1
158	0	1	1
159	11	9	7
160	6	7	6
161	1	2	1
162	0	1	1
163	0	1	1
164	0	1	1
165	0	1	1
166	0	1	1
167	0	1	1
168	0	1	1
169	0	1	1
170	0	1	1
171	0	1	1
172	0	1	1
173	0	1	1
174	0	1	1
175	0	1	1
176	0	1	1
177	0	1	1
178	0	1	1
179	1	2	1
180	0	1	1
181	0	1	1
182	0	1	1
183	0	1	1
184	0	1	1
185	0	1	1
186	0	1	1
187	3	4	3
188	53	17	9
189	0	1	1
190	0	1	1
191	0	1	1
192	0	1	1
193	0	1	1
194	0	1	1
195	0	1	1
196	0	1	1
197	0	1	1
198	0	1	1
199	0	1	1
200	0	1	1
201	0	1	1
202	0	1	1
203	0	1	1
204	0	1	1
205	0	1	1
206	0	1	1
207	0	1	1
208	0	1	1
209	0	1	1
210	0	1	1
211	0	1	1
212	7	7	4
213	0	1	1
214	0	1	1
215	0	1	1
216	0	1	1
217	1	2	1
218	0	1	1
219	0	1	1
220	0	1	1

221	0	1	1
222	0	1	1
223	0	1	1
224	0	1	1
225	0	1	1
226	0	1	1
227	0	1	1
228	0	1	1
229	0	1	1
230	0	1	1
231	0	1	1
232	0	1	1
233	0	1	1
234	0	1	1
235	17	3	1
236	3	2	1
237	133	7	4
238	0	1	1
239	698	24	5
240	0	1	1
241	0	1	1
242	410	8	6
243	0	1	1
244	36	2	1
245	0	1	2

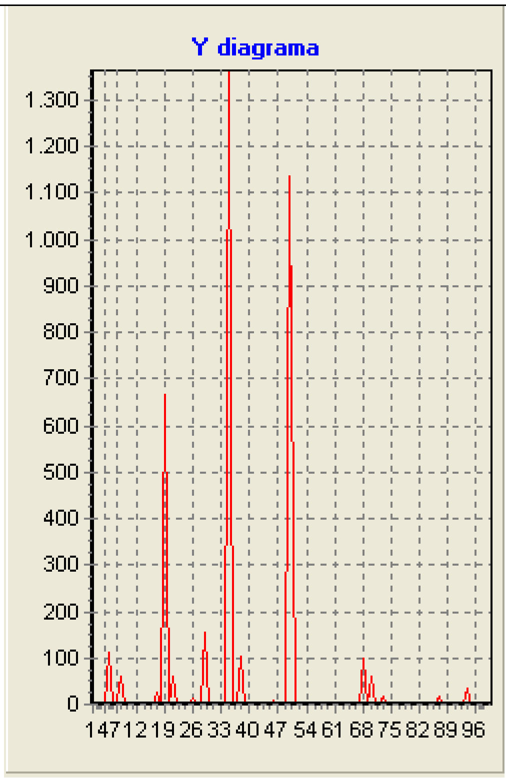
Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (13,05; 26,71);
Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (1,69; 2,41);

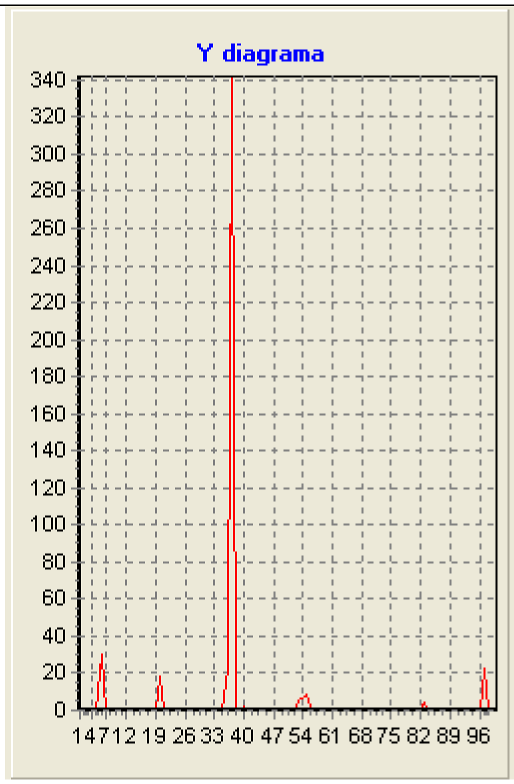
221	0	1	1
222	0	1	1
223	105	37	7
224	0	1	6

Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (0,82; 2,49);
Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (1,62; 2,87);

2 lentelė

Paraiškų laukimo laikų ilgių sumų cikluose grafikai





3 lentelė

Modeliavimo rezultatai dvikanalėje aptarnavimo sistemoje

Laukimo laiko eilėje ir paraiškų atėjimo momentai pasiskirstę pagal tolygųjį dėsnį.

Laukimo laiko momentai pirmame kanale: t(30; 90) Laukimo laiko momentai antrame kanale: t(31; 90) Paraiškų atėjimo momentai: t(30; 35)				Laukimo laiko momentai pirmame kanale: t(9; 15) Laukimo laiko momentai antrame kanale: t(10; 15) Paraiškų atėjimo momentai: exp(5; 8)			
Ciklas	Y	alfa	Z	Ciklas	Y	alfa	Z
1	6	2	1	1	2	2	1
2	0	1	1	2	1	2	1
3	0	1	1	3	3	2	1
4	24	2	1	4	4	2	1
5	12	2	1	5	9	4	2
6	5	2	1	6	3	2	1
7	0	1	1	7	1	2	1
8	0	1	1	8	0	1	1
9	0	1	1	9	0	1	1
10	0	1	1	10	0	1	1
11	0	1	1	11	0	1	1
12	0	1	1	12	0	1	1
13	0	1	1	13	0	1	1
14	3	2	1	14	38	15	3
15	6	2	1	15	24	11	5
16	0	1	1	16	2	2	1
17	308	14	3	17	1	2	1
18	33	4	1	18	0	1	1
19	39	5	2	19	0	1	1
20	85	5	2	20	0	1	1
21	42	3	2	21	2	3	2
22	0	1	1	22	2	2	1
23	0	1	1	23	11	4	2
24	55	5	3	24	3	2	1
25	25	2	1	25	4	2	1
26	19	2	1	26	5	2	1
27	38	4	1	27	16	6	3
28	21	2	1	28	0	1	1
29	56	5	3	29	0	1	1
30	0	1	1	30	0	1	1
31	0	1	1	31	5	3	1
32	0	1	1	32	1	2	1
33	0	1	1	33	5	2	1
34	0	1	1	34	4	2	1
35	4	2	1	35	17	6	2
36	0	1	1	36	27	10	2
37	0	1	1	37	1171	102	7
38	0	1	1	38	1	2	1
39	0	1	1	39	0	1	1
40	2	2	1	40	1	2	1
41	81	6	2	41	2	2	1
42	37	4	2	42	6	4	3
43	25	2	1	43	0	1	1
44	4	2	1	44	0	1	1
45	3	2	1	45	1	2	1
46	75	7	2	46	0	1	1
47	19	2	1	47	14	6	2
48	123	9	3	48	6	2	1
49	2594	47	4	49	28	9	3
50	0	1	1	50	14	6	3
51	0	1	1	51	0	1	1
52	0	1	1	52	0	1	1
53	0	1	1	53	0	1	1
54	0	1	1	54	0	1	1
55	36	3	1	55	0	1	1
56	0	1	1	56	0	1	1
57	0	1	1	57	0	1	1
58	0	1	1	58	0	1	1
59	0	1	1	59	0	1	1
60	0	1	1	60	0	1	1
61	0	1	1	61	8	4	2
62	0	1	1	62	0	1	1
63	13	2	1	63	0	1	1
64	30	3	2	64	43	11	3
65	0	1	1	65	1	2	1
66	0	1	1	66	5	4	3

67	0	1	1
68	17	2	1
69	3	2	1
70	69	5	2
71	34	3	2
72	0	1	1
73	0	1	1
74	0	1	1
75	0	1	1
76	0	1	1
77	0	1	1
78	0	1	1
79	6	3	2
80	38	4	3
81	0	1	1
82	4	2	1
83	0	1	1
84	493	22	4
85	0	1	1
86	0	1	1
87	0	1	1
88	0	1	1
89	0	1	1
90	0	1	1
91	17	2	1
92	0	1	1
93	0	1	1
94	0	1	1
95	0	1	1
96	37	3	1
97	0	1	1
98	0	1	1
99	282	12	4
100	0	1	1
101	0	1	1
102	0	1	1
103	0	1	1
104	0	1	1
105	0	1	1
106	0	1	1
107	0	1	1
108	0	1	1
109	0	1	1
110	0	1	1
111	0	1	1
112	29	3	1
113	22	3	2
114	0	1	1
115	0	1	1
116	0	1	1
117	0	1	1
118	0	1	1
119	10	2	1
120	16	2	1
121	0	1	1
122	32	5	3
123	4	2	1
124	6	2	1
125	15	2	1
126	66	4	2
127	8	3	2
128	35	5	2
129	0	1	1
130	0	1	1
131	14	2	1
132	62	6	2
133	0	1	1
134	134	11	4
135	0	1	1
136	0	1	1
137	0	1	1
138	0	1	1
139	0	1	1
140	364	15	2
141	0	1	1
142	8	2	1
143	251	15	2
144	13	2	1
145	83	5	2
146	38	4	3

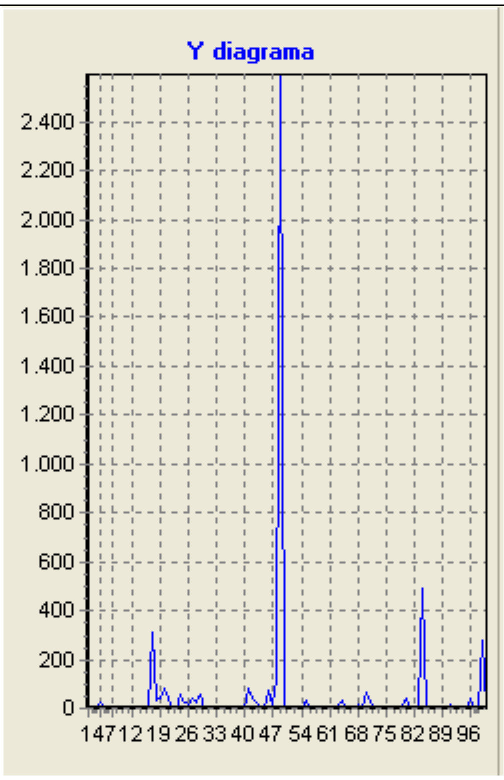
67	4	4	3
68	0	1	1
69	0	1	1
70	0	1	1
71	16	7	2
72	1	2	1
73	0	1	1
74	0	1	1
75	1	2	1
76	3	3	2
77	3	4	3
78	1	2	1
79	0	1	1
80	0	1	1
81	0	1	1
82	22	8	2
83	20	8	2
84	2	2	1
85	3	2	1
86	4	2	1
87	3	3	2
88	20	12	5
89	3	2	1
90	1	2	1
91	69	16	5
92	36	10	2
93	4	2	1
94	10	4	2
95	0	1	1
96	0	1	1
97	0	1	1
98	6	5	2
99	7	5	4
100	3	2	1
101	1	2	1
102	270	48	4
103	1	2	1
104	0	1	1
105	2	2	1
106	0	1	1
107	0	1	1
108	5	4	2
109	3	2	1
110	86	18	2
111	18	8	2
112	36	12	3
113	0	1	1

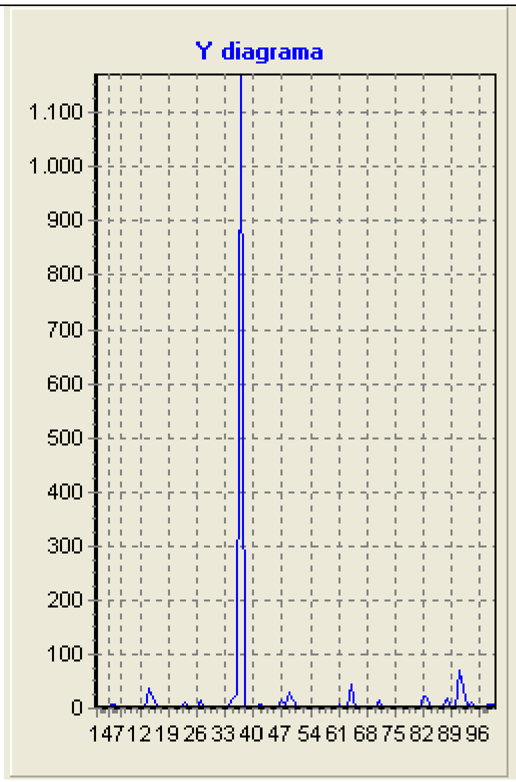
147	0	1	1
148	17	2	1
149	0	1	1
150	0	1	1
151	0	1	1
152	0	1	1
153	181	13	2
154	0	1	1
155	163	10	3
156	0	1	1
157	163	9	3
158	53	4	2
159	0	1	1
160	0	1	1
161	0	1	1
162	0	1	1
163	0	1	1
164	0	1	1
165	0	1	1
166	0	1	1
167	0	1	1
168	0	1	1
169	0	1	1
170	92	8	3
171	512	20	4
172	0	1	1
173	0	1	1
174	58	4	2
175	0	1	1
176	0	1	1
177	375	12	3
178	0	1	3

Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (4,60; 12,32); Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (2,24; 3,41);	Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (1,50; 4,45); Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (2,79; 6,11);

4 lentelė

Paraiškų laukimo laikų ilgių sumų cikluose grafikai





5 lentelė

Modeliavimo rezultatai dvikanalėje aptarnavimo sistemoje

Laukimo laiko eilėje momentai pasiskirstę pagal tolygųjį dėsnį.							
Laukimo laiko momentai pirmame kanale: t(30; 90) Laukimo laiko momentai antrame kanale: t(31; 90) Paraiškų generuojamos kas 35 laiko vienetus				Laukimo laiko momentai pirmame kanale: t(9; 20) Laukimo laiko momentai antrame kanale: t(12; 20) Paraiškų generuojamos kas 8 laiko vienetus			
Ciklas	Y	alfa	Z	Ciklas	Y	alfa	Z
1	0	1	1	1	0	1	1
2	0	1	1	2	0	1	1
3	16	2	1	3	3	2	1
4	0	1	1	4	0	1	1
5	0	1	1	5	1	2	1
6	0	1	1	6	0	1	1
7	0	1	1	7	3	2	1
8	16	2	1	8	0	1	1
9	0	1	1	9	0	1	1
10	0	1	1	10	0	1	1
11	0	1	1	11	1	2	1
12	0	1	1	12	0	1	1
13	0	1	1	13	0	1	1
14	0	1	1	14	0	1	1
15	0	1	1	15	0	1	1
16	0	1	1	16	3	2	1
17	0	1	1	17	0	1	1
18	18	2	1	18	1	2	1
19	0	1	1	19	0	1	1
20	0	1	1	20	0	1	1
21	0	1	1	21	0	1	1
22	0	1	1	22	0	1	1
23	0	1	1	23	0	1	1
24	0	1	1	24	0	1	1
25	0	1	1	25	1	2	1
26	0	1	1	26	2	2	1
27	0	1	1	27	15	5	3
28	0	1	1	28	4	2	1
29	0	1	1	29	1	2	1
30	6	2	1	30	0	1	1
31	64	5	3	31	0	1	1
32	9	2	1	32	11	5	3
33	0	1	1	33	4	2	1
34	0	1	1	34	6	2	1
35	0	1	1	35	3	2	1
36	0	1	1	36	1	2	1
37	59	5	2	37	0	1	1
38	0	1	1	38	1	2	1
39	0	1	1	39	0	1	1
40	10	2	1	40	0	1	1
41	0	1	1	41	0	1	1
42	0	1	1	42	0	1	1
43	0	1	1	43	0	1	1
44	0	1	1	44	0	1	1
45	0	1	1	45	2	3	2
46	4	2	1	46	0	1	1
47	0	1	1	47	4	3	1
48	0	1	1	48	5	2	1
49	0	1	1	49	0	1	1
50	3	2	1	50	0	1	1
51	1	2	1	51	0	1	1
52	0	1	1	52	0	1	1
53	0	1	1	53	6	3	2
54	0	1	1	54	0	1	1
55	0	1	1	55	0	1	1
56	14	2	1	56	0	1	1
57	25	2	1	57	0	1	1
58	1	2	1	58	2	2	1
59	0	1	1	59	0	1	1
60	0	1	1	60	0	1	1
61	34	3	2	61	2	2	1
62	0	1	1	62	7	4	3
63	0	1	1	63	0	1	1
64	0	1	1	64	0	1	1
65	0	1	1	65	3	2	1
66	12	2	1	66	1	2	1

67	0	1	1	67	3	3	1
68	0	1	1	68	0	1	1
69	7	2	1	69	14	6	2
70	20	3	2	70	5	2	1
71	0	1	1	71	1	2	1
72	0	1	1	72	0	1	1
73	0	1	1	73	0	1	1
74	16	2	1	74	0	1	1
75	0	1	1	75	0	1	1
76	0	1	1	76	0	1	1
77	0	1	1	77	0	1	1
78	0	1	1	78	4	3	2
79	0	1	1	79	3	2	1
80	0	1	1	80	5	2	1
81	0	1	1	81	2	2	1
82	57	5	2	82	2	2	1
83	7	2	1	83	0	1	1
84	0	1	1	84	0	1	1
85	0	1	1	85	3	2	1
86	0	1	1	86	0	1	1
87	0	1	1	87	0	1	1
88	0	1	1	88	0	1	1
89	0	1	1	89	1	2	1
90	3	2	1	90	0	1	1
91	18	3	2	91	6	4	2
92	4	2	1	92	3	2	1
93	9	2	1	93	0	1	1
94	0	1	1	94	0	1	1
95	0	1	1	95	0	1	1
96	0	1	1	96	8	4	1
97	0	1	1	97	0	1	1
98	0	1	1	98	0	1	1
99	0	1	1	99	3	2	1
100	13	2	1	100	5	2	1
101	0	1	1	101	4	2	1
102	0	1	1	102	0	1	1
103	0	1	1	103	0	1	1
104	0	1	1	104	0	1	1
105	0	1	1	105	0	1	1
106	0	1	1	106	0	1	1
107	35	6	4	107	1	2	1
108	0	1	1	108	0	1	1
109	0	1	1	109	3	2	1
110	0	1	1	110	3	2	1
111	35	4	1	111	21	7	3
112	0	1	1	112	3	2	1
113	0	1	1	113	0	1	1
114	19	2	1	114	0	1	1
115	60	5	2	115	0	1	1
116	0	1	1	116	0	1	1
117	0	1	1	117	5	4	2
118	0	1	1	118	1	2	1
119	0	1	1	119	0	1	1
120	4	2	1	120	0	1	1
121	0	1	1	121	0	1	1
122	1	2	1	122	0	1	1
123	0	1	1	123	0	1	1
124	0	1	1	124	2	2	1
125	29	3	2	125	7	4	3
126	49	3	2	126	0	1	1
127	0	1	1	127	2	2	1
128	10	2	1	128	2	2	1
129	0	1	1	129	4	2	1
130	0	1	1	130	8	3	2
131	0	1	1	131	4	2	1
132	0	1	1	132	0	1	1
133	6	2	1	133	3	2	1
134	7	2	1	134	5	3	2
135	0	1	1	135	2	2	1
136	0	1	1	136	0	1	1
137	0	1	1	137	1	2	1
138	0	1	1	138	11	6	2
139	0	1	1	139	5	3	2
140	0	1	1	140	19	7	3
141	0	1	1	141	4	2	1
142	8	2	1	142	3	2	1
143	32	3	2	143	3	2	1
144	0	1	1	144	0	1	1
145	0	1	1	145	1	2	1
146	14	2	1	146	0	1	1

147	13	2	1
148	7	2	1
149	28	3	2
150	15	2	1
151	0	1	1
152	18	2	1
153	27	3	2
154	4	2	1
155	0	1	1
156	0	1	1
157	0	1	1
158	0	1	1
159	44	5	2
160	18	2	1
161	0	1	1
162	14	3	1
163	59	4	2
164	8	2	1
165	1	2	1
166	0	1	1
167	0	1	1
168	0	1	1
169	2	2	1
170	0	1	1
171	0	1	1
172	2	2	1
173	0	1	1
174	0	1	1
175	0	1	1
176	0	1	1
177	0	1	1
178	0	1	1
179	0	1	1
180	0	1	1
181	0	1	1
182	5	2	1
183	0	1	1
184	0	1	1
185	80	7	3
186	130	7	3
187	1	2	1
188	0	1	1
189	0	1	1
190	0	1	1
191	0	1	1
192	0	1	1
193	17	2	1
194	0	1	1
195	0	1	1
196	10	3	2
197	0	1	1
198	0	1	1
199	0	1	1
200	0	1	1
201	5	2	1
202	0	1	1
203	0	1	1
204	35	4	2
205	15	2	1
206	0	1	1
207	0	1	1
208	29	3	2
209	28	3	2
210	0	1	1
211	0	1	1
212	0	1	1
213	0	1	1
214	0	1	1
215	0	1	1
216	17	3	2
217	0	1	1
218	0	1	1
219	15	2	1
220	0	1	1
221	0	1	1
222	0	1	1
223	0	1	1
224	16	3	2
225	0	1	1
226	33	3	1

147	0	1	1
148	2	3	2
149	0	1	1
150	0	1	1
151	0	1	1
152	0	1	1
153	0	1	1
154	0	1	1
155	0	1	1
156	0	1	1
157	1	2	1
158	0	1	1
159	0	1	1
160	10	5	1
161	6	2	1
162	24	7	2
163	4	2	1
164	1	2	1
165	0	1	1
166	0	1	1
167	0	1	1
168	3	2	1
169	1	2	1
170	8	5	2
171	2	2	1
172	3	2	1
173	0	1	1
174	0	1	1
175	7	4	2
176	4	3	2
177	8	4	1
178	9	4	2
179	0	1	1
180	0	1	1
181	0	1	1
182	0	1	1
183	0	1	1
184	3	3	2
185	0	1	1
186	0	1	1
187	3	2	1
188	0	1	1
189	0	1	1
190	1	2	1
191	6	4	3
192	7	3	2
193	3	2	1
194	4	2	1
195	1	2	1
196	0	1	1
197	0	1	1
198	0	1	1
199	3	2	1
200	2	2	1
201	0	1	1
202	6	4	3
203	0	1	1
204	0	1	1
205	0	1	1
206	2	3	2
207	0	1	1
208	0	1	1
209	0	1	1
210	0	1	1
211	6	4	2
212	2	2	1
213	0	1	1
214	9	4	2
215	0	1	1
216	0	1	1
217	0	1	1
218	3	3	2
219	0	1	1
220	2	2	1
221	0	1	1
222	0	1	1
223	0	1	1
224	0	1	1
225	0	1	1
226	0	1	1

227	26	4	2
228	0	1	1
229	0	1	1
230	0	1	1
231	0	1	1
232	0	1	1
233	12	2	1
234	1	2	1
235	0	1	1
236	31	4	3
237	0	1	1
238	0	1	1
239	0	1	1
240	7	3	2
241	0	1	1
242	0	1	1
243	0	1	1
244	0	1	1
245	22	4	3
246	0	1	1
247	0	1	1
248	0	1	1
249	17	2	1
250	20	2	1
251	0	1	1
252	0	1	1
253	0	1	1
254	12	3	2
255	0	1	1
256	6	2	1
257	0	1	1
258	0	1	1
259	0	1	1
260	0	1	1
261	0	1	1
262	0	1	1
263	0	1	1
264	0	1	1
265	0	1	1
266	0	1	1
267	0	1	1
268	0	1	1
269	13	2	1
270	23	2	1
271	0	1	1
272	0	1	1
273	0	1	1
274	0	1	1
275	0	1	1
276	20	4	1
277	0	1	1
278	7	2	1
279	15	3	1
280	0	1	1
281	0	1	1
282	0	1	1
283	21	3	2
284	0	1	1
285	0	1	1
286	0	1	1
287	0	1	1
288	0	1	1
289	7	2	1
290	38	3	2
291	10	2	1
292	8	2	1
293	140	8	4
294	3	2	1
295	0	1	1
296	48	7	2
297	40	4	2
298	0	1	1
299	0	1	1
300	0	1	1
301	0	1	1
302	0	1	1
303	0	1	1
304	0	1	1
305	0	1	1
306	0	1	1

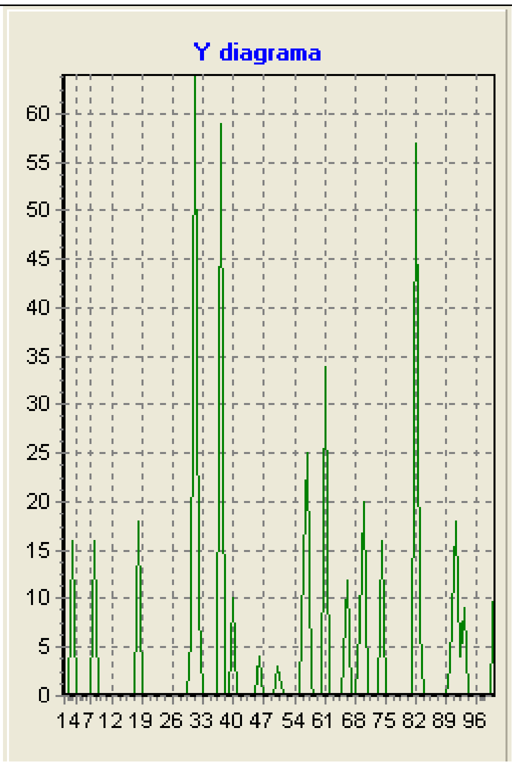
227	0	1	1
228	0	1	1
229	0	1	1
230	0	1	1
231	0	1	1
232	0	1	1
233	3	2	1
234	5	2	1
235	1	2	1
236	0	1	1
237	0	1	1
238	0	1	1
239	5	4	2
240	0	1	1
241	1	2	1
242	0	1	1
243	2	2	1
244	1	2	1
245	7	4	2
246	0	1	1
247	0	1	1
248	0	1	1
249	3	2	1
250	4	2	1
251	14	5	3
252	4	2	1
253	45	10	2
254	19	7	2
255	10	5	2
256	1	2	1
257	0	1	1
258	0	1	1
259	0	1	1
260	0	1	1
261	0	1	1
262	0	1	1
263	0	1	1
264	0	1	1
265	0	1	1
266	2	2	1
267	0	1	1
268	0	1	1
269	0	1	1
270	1	2	1
271	1	2	1
272	0	1	1
273	0	1	1
274	2	2	1
275	7	4	2
276	0	1	1
277	0	1	1
278	0	1	1
279	0	1	1
280	0	1	1

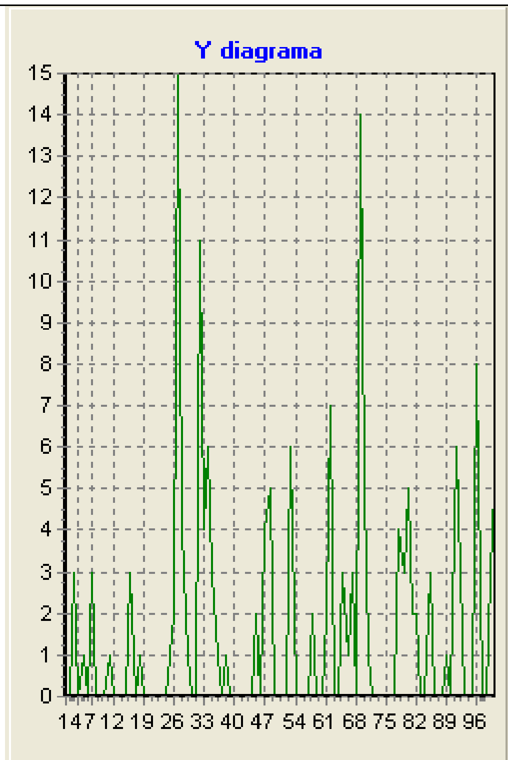
```
| 307 |    0    |    1    |    1    |
| 308 |    0    |    1    |    1    |
| 309 |   10    |    2    |    1    |
| 310 |    0    |    1    |    1    |
| 311 |    0    |    1    |    1    |
| 312 |    0    |    1    |    1    |
| 313 |    0    |    1    |    1    |
| 314 |    0    |    1    |    1    |
| 315 |    0    |    1    |    1    |
| 316 |    0    |    1    |    1    |
| 317 |    0    |    1    |    1    |
| 318 |    0    |    1    |    1    |
| 319 |   11    |    2    |    1    |
| 320 |   31    |    5    |    4    |
| 321 |    0    |    1    |    1    |
| 322 |    0    |    1    |    1    |
| 323 |    0    |    1    |    1    |
| 324 |    0    |    1    |    1    |
------------------------------------
```

Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (4,36; 9,65); Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (1,45; 1,65);	Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (1,31; 3,43); Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (1,67; 1,92);

6 lentelė

Paraiškų laukimo laikų ilgių sumų cikluose grafikai





2 PRIEDAS. MODELIAVIMO REZULTATŲ TRIKANALĖJE APTARNAVIMO SISTEMOJE LENTELĖS IR GRAFIKAI

7 lentelė

Modeliavimo rezultatai trikanalėje aptarnavimo sistemoje

Laukimo laiko eilėje ir paraiškų atėjimo momentai pasiskirstę pagal eksponentinį dėsnį.							
Laukimo laiko momentai pirmame kanale: exp(0,01) Laukimo laiko momentai antrame kanale: exp(0,02) Laukimo laiko momentai antrame kanale: exp(0,01) Paraiškų atėjimo momentai: exp(0,02)				Laukimo laiko momentai pirmame kanale: exp(0,2) Laukimo laiko momentai antrame kanale: exp(0,4) Laukimo laiko momentai antrame kanale: exp(0,2) Paraiškų atėjimo momentai: exp(0,9)			
Ciklas	S	alfa	M	Ciklas	S	alfa	M
1	0	1	1	1	0	1	1
2	0	1	1	2	0	1	1
3	0	1	1	3	0	1	1
4	0	1	1	4	0	1	1
5	23	2	2	5	1	2	2
6	0	1	1	6	0	1	1
7	6	2	1	7	1	2	2
8	0	1	1	8	0	1	1
9	0	1	1	9	0	1	1
10	0	1	1	10	0	1	1
11	0	1	1	11	0	1	1
12	0	1	1	12	0	1	1
13	0	1	1	13	0	1	1
14	0	1	1	14	0	1	1
15	22	3	2	15	0	1	1
16	0	1	1	16	0	1	1
17	0	1	1	17	0	1	1
18	0	1	1	18	0	1	1
19	0	1	1	19	0	1	1
20	0	1	1	20	0	1	1
21	0	1	1	21	0	1	1
22	0	1	1	22	0	1	1
23	0	1	1	23	0	1	1
24	0	1	1	24	0	1	1
25	0	1	1	25	0	1	1
26	0	1	1	26	0	1	1
27	0	1	1	27	0	1	1
28	0	1	1	28	0	1	1
29	0	1	1	29	0	1	1
30	0	1	1	30	0	1	1
31	75	7	2	31	0	1	1
32	82	5	3	32	0	1	1
33	0	1	1	33	2	3	3
34	0	1	1	34	0	1	1
35	0	1	1	35	0	1	1
36	0	1	1	36	0	1	1
37	0	1	1	37	0	1	1
38	0	1	1	38	2	3	3
39	0	1	1	39	0	1	1
40	0	1	1	40	0	1	1
41	0	1	1	41	0	1	1
42	0	1	1	42	0	1	1
43	0	1	1	43	0	1	1
44	0	1	1	44	0	1	1
45	0	1	1	45	0	1	1
46	0	1	1	46	0	1	1
47	0	1	1	47	0	1	1
48	0	1	1	48	0	1	1
49	0	1	1	49	0	1	1
50	0	1	1	50	0	1	1
51	0	1	1	51	0	1	1
52	8	2	1	52	0	1	1
53	0	1	1	53	0	1	1
54	0	1	1	54	0	1	1
55	0	1	1	55	0	1	1
56	0	1	1	56	0	1	1
57	0	1	1	57	0	1	1
58	0	1	1	58	0	1	1
59	0	1	1	59	1	2	2

60	0	1	1	60	0	1	1
61	0	1	1	61	0	1	1
62	269	8	3	62	0	1	1
63	0	1	1	63	0	1	1
64	0	1	1	64	0	1	1
65	10	2	1	65	0	1	1
66	0	1	1	66	1	2	2
67	0	1	1	67	0	1	1
68	0	1	1	68	0	1	1
69	0	1	1	69	0	1	1
70	24	3	2	70	20	8	7
71	0	1	1	71	0	1	1
72	0	1	1	72	0	1	1
73	0	1	1	73	0	1	1
74	0	1	1	74	0	1	1
75	0	1	1	75	8	7	7
76	0	1	1	76	2	2	2
77	0	1	1	77	0	1	1
78	0	1	1	78	0	1	1
79	0	1	1	79	0	1	1
80	0	1	1	80	0	1	1
81	0	1	1	81	0	1	1
82	0	1	1	82	0	1	1
83	0	1	1	83	0	1	1
84	0	1	1	84	0	1	1
85	0	1	1	85	0	1	1
86	0	1	1	86	0	1	1
87	0	1	1	87	0	1	1
88	0	1	1	88	0	1	1
89	0	1	1	89	0	1	1
90	0	1	1	90	0	1	1
91	0	1	1	91	0	1	1
92	15	2	2	92	0	1	1
93	356	9	3	93	0	1	1
94	0	1	1	94	0	1	1
95	0	1	1	95	0	1	1
96	0	1	1	96	0	1	1
97	0	1	1	97	0	1	1
98	0	1	1	98	7	4	4
99	0	1	1	99	0	1	1
100	0	1	1	100	0	1	1
101	0	1	1	101	0	1	1
102	0	1	1	102	0	1	1
103	0	1	1	103	0	1	1
104	0	1	1	104	0	1	1
105	0	1	1	105	0	1	1
106	0	1	1	106	0	1	1
107	0	1	1	107	0	1	1
108	0	1	1	108	0	1	1
109	0	1	1	109	0	1	1
110	0	1	1	110	0	1	1
111	0	1	1	111	0	1	1
112	0	1	1	112	0	1	1
113	0	1	1	113	0	1	1
114	0	1	1	114	0	1	1
115	0	1	1	115	1	2	2
116	0	1	1	116	0	1	1
117	0	1	1	117	0	1	1
118	0	1	1	118	0	1	1
119	0	1	1	119	0	1	1
120	0	1	1	120	0	1	1
121	26	3	2	121	0	1	1
122	0	1	1	122	0	1	1
123	0	1	1	123	0	1	1
124	0	1	1	124	0	1	1
125	0	1	1	125	0	1	1
126	0	1	1	126	0	1	1
127	0	1	1	127	0	1	1
128	0	1	1	128	0	1	1
129	0	1	1	129	0	1	1
130	82	3	3	130	0	1	1
131	0	1	1	131	0	1	1
132	0	1	1	132	0	1	1
133	0	1	1	133	0	1	1
134	0	1	1	134	2	3	3
135	0	1	1	135	0	1	1
136	0	1	1	136	0	1	1
137	0	1	1	137	0	1	1
138	0	1	1	138	0	1	1
139	0	1	1	139	0	1	1

140	0	1	1
141	0	1	1
142	0	1	1
143	0	1	1
144	0	1	1
145	0	1	1
146	0	1	1
147	0	1	1
148	0	1	1
149	0	1	1
150	0	1	1
151	0	1	1
152	0	1	1
153	19	3	2
154	0	1	1
155	0	1	1
156	0	1	1
157	0	1	1
158	0	1	1
159	0	1	1
160	0	1	1
161	0	1	1
162	8	2	2
163	4	2	2
164	0	1	1
165	0	1	1
166	0	1	1
167	0	1	1
168	0	1	1
169	0	1	1
170	0	1	1
171	0	1	1
172	0	1	1
173	0	1	1
174	0	1	1
175	2	2	2
176	0	1	1
177	0	1	1
178	0	1	1
179	0	1	1
180	0	1	1
181	0	1	1
182	0	1	1
183	4	2	2
184	0	1	1
185	0	1	1
186	0	1	1
187	0	1	1
188	0	1	1
189	0	1	1
190	0	1	1
191	0	1	1
192	0	1	1
193	0	1	1
194	0	1	1
195	0	1	1
196	0	1	1
197	0	1	1
198	0	1	1
199	0	1	1
200	0	1	1
201	0	1	1
202	0	1	1
203	0	1	1
204	0	1	1
205	8	2	2
206	8	2	1
207	0	1	1
208	0	1	1
209	29	2	2
210	7	3	1
211	0	1	1
212	0	1	1
213	0	1	1
214	13	2	2
215	122	7	2
216	0	1	1
217	0	1	1
218	0	1	1
219	0	1	1

140	0	1	1
141	0	1	1
142	0	1	1
143	0	1	1
144	0	1	1
145	0	1	1
146	0	1	1
147	0	1	1
148	0	1	1
149	0	1	1
150	0	1	1
151	0	1	1
152	0	1	1
153	0	1	1
154	0	1	1
155	0	1	1
156	0	1	1
157	0	1	1
158	0	1	1
159	0	1	1
160	0	1	1
161	0	1	1
162	0	1	1
163	0	1	1
164	0	1	1
165	0	1	1
166	0	1	1
167	0	1	1
168	0	1	1
169	0	1	1
170	0	1	1
171	0	1	1
172	0	1	1
173	0	1	1
174	0	1	1
175	0	1	1
176	0	1	1
177	0	1	1
178	0	1	1
179	0	1	1
180	0	1	1
181	0	1	1
182	0	1	1
183	0	1	1
184	0	1	1
185	0	1	1
186	0	1	1
187	0	1	1
188	0	1	1
189	0	1	1
190	0	1	1
191	0	1	1
192	1	2	2
193	0	1	1
194	0	1	1
195	0	1	1
196	0	1	1
197	0	1	1
198	0	1	1
199	0	1	1
200	0	1	1
201	0	1	1
202	0	1	1
203	0	1	1
204	0	1	1
205	0	1	1
206	0	1	1
207	0	1	1
208	0	1	1
209	0	1	1
210	0	1	1
211	0	1	1
212	0	1	1
213	0	1	1
214	0	1	1
215	0	1	1
216	0	1	1
217	0	1	1
218	0	1	1
219	0	1	1

220	0	1	1
221	0	1	1
222	0	1	1
223	0	1	1
224	0	1	1
225	0	1	1
226	0	1	1
227	0	1	1
228	0	1	1
229	362	11	2
230	0	1	1
231	0	1	1
232	0	1	1
233	121	8	3
234	0	1	1
235	0	1	1
236	0	1	1
237	0	1	1
238	0	1	1
239	0	1	1
240	0	1	1
241	0	1	1
242	0	1	1
243	0	1	1
244	0	1	1
245	0	1	1
246	0	1	1
247	0	1	1
248	0	1	1
249	0	1	1
250	0	1	1
251	0	1	1
252	0	1	1
253	0	1	1
254	0	1	1
255	0	1	1
256	2	2	2
257	0	1	1
258	0	1	1
259	0	1	1
260	50	4	2
261	0	1	1
262	166	5	2
263	0	1	1
264	0	1	1
265	0	1	1
266	0	1	1
267	0	1	1
268	0	1	1
269	0	1	1
270	0	1	1
271	0	1	1
272	0	1	1
273	0	1	1
274	0	1	1
275	15	3	2
276	146	7	2
277	0	1	1
278	0	1	1
279	0	1	1
280	0	1	1
281	0	1	1
282	0	1	1
283	0	1	1
284	0	1	1
285	0	1	1
286	0	1	1
287	0	1	1
288	0	1	1
289	0	1	1
290	0	1	1
291	0	1	1
292	0	1	1
293	0	1	1
294	0	1	1
295	0	1	1
296	0	1	1
297	0	1	1
298	0	1	1

220	0	1	1
221	0	1	1
222	0	1	1
223	0	1	1
224	0	1	1
225	0	1	1
226	0	1	1
227	2	3	3
228	0	1	1
229	0	1	1
230	0	1	1
231	0	1	1
232	0	1	1
233	0	1	1
234	0	1	1
235	0	1	1
236	0	1	1
237	0	1	1
238	0	1	1
239	0	1	1
240	0	1	1
241	0	1	1
242	0	1	1
243	0	1	1
244	0	1	1
245	0	1	1
246	0	1	1
247	0	1	1
248	0	1	1
249	0	1	1
250	0	1	1
251	0	1	1
252	1	2	2
253	0	1	1
254	0	1	1
255	20	10	9
256	1	2	2
257	2	3	3
258	0	1	1
259	1	2	2
260	0	1	1
261	0	1	1
262	0	1	1
263	0	1	1
264	0	1	1
265	0	1	1
266	0	1	1
267	0	1	1
268	0	1	1
269	0	1	1
270	0	1	1
271	0	1	1
272	0	1	1
273	0	1	1
274	0	1	1
275	0	1	1
276	0	1	1
277	0	1	1
278	0	1	1
279	0	1	1
280	0	1	1
281	0	1	1
282	0	1	1
283	0	1	1
284	0	1	1
285	0	1	1
286	0	1	1
287	0	1	1
288	0	1	1
289	0	1	1
290	0	1	1
291	0	1	1
292	0	1	1
293	0	1	1
294	0	1	1
295	3	2	2
296	0	1	1
297	0	1	1
298	0	1	1
299	0	1	1

299	0	1	1	300	0	1	1
300	0	1	1	301	0	1	1
301	0	1	1	302	0	1	1
302	0	1	1	303	0	1	1
303	24	3	2	304	0	1	1
304	4	2	1	305	0	1	1
305	0	1	1	306	0	1	1
306	0	1	1	307	0	1	1
307	0	1	1	308	0	1	1
308	0	1	1	309	0	1	1
309	0	1	1	310	0	1	1
310	3	2	1	311	0	1	1
311	34	2	2	312	0	1	1
312	305	8	3	313	2	3	3
313	0	1	1	314	0	1	1
314	0	1	1	315	0	1	1
315	0	1	1	316	0	1	1
316	0	1	1	317	0	1	1
317	0	1	1	318	0	1	1
318	0	1	1	319	0	1	1
319	0	1	1	320	0	1	1
320	0	1	1	321	0	1	1
321	0	1	1	322	0	1	1
322	0	1	1	323	0	1	1
323	0	1	1	324	0	1	1
324	0	1	1	325	0	1	1
325	0	1	1	326	0	1	1
326	0	1	1	327	0	1	1
327	0	1	1	328	0	1	1
328	0	1	1	329	0	1	1
329	0	1	1	330	0	1	1
330	0	1	1	331	0	1	1
331	0	1	1	332	0	1	1
332	0	1	1	333	0	1	1
333	0	1	1	334	0	1	1
334	0	1	1	335	0	1	1
335	0	1	1	336	0	1	1
336	0	1	1	337	0	1	1
337	0	1	1	338	0	1	1
338	0	1	1	339	0	1	1
339	0	1	1	340	0	1	1
340	0	1	1	341	0	1	1
341	0	1	1	342	0	1	1
342	0	1	1	343	0	1	1
343	0	1	1	344	0	1	1
344	0	1	1	345	0	1	1
345	0	1	1	346	0	1	1
346	0	1	1	347	0	1	1
347	0	1	1	348	0	1	1
348	0	1	1	349	0	1	1
349	17	3	2	350	0	1	1
350	3	2	2	351	0	1	1
351	0	1	1	352	0	1	1
352	10	3	2	353	0	1	1
353	0	1	1	354	1	2	2
354	0	1	1	355	0	1	1
355	0	1	1	356	0	1	1
356	0	1	1	357	16	8	8
357	0	1	1	358	0	1	1
358	0	1	1	359	0	1	1
359	78	4	2	360	0	1	1
360	0	1	1	361	0	1	1
361	33	4	2	362	0	1	1
362	0	1	1	363	0	1	1
363	9	3	2	364	0	1	1
364	0	1	1	365	0	1	1
365	0	1	1	366	0	1	1
366	0	1	1	367	0	1	1
367	0	1	1	368	0	1	1
368	34	2	2	369	0	1	1
369	0	1	1	370	0	1	1
370	0	1	1	371	0	1	1
371	0	1	1	372	0	1	1
372	0	1	1	373	0	1	1
373	0	1	1	374	0	1	1
374	0	1	1	375	0	1	1
375	202	7	2	376	0	1	1
376	0	1	1	377	0	1	1
377	0	1	1	378	0	1	1
378	0	1	1	379	0	1	1

379	1	2	2	
380	1	2	2	
381	0	1	1	
382	1	2	2	
383	22	2	2	
384	1	2	1	
385	0	1	1	
386	0	1	1	
387	0	1	1	
388	0	1	1	
389	0	1	1	
390	0	1	1	
391	0	1	1	
392	0	1	1	
393	0	1	1	
394	0	1	1	
395	0	1	1	
396	0	1	1	
397	0	1	1	
398	0	1	1	
399	48	3	2	
400	0	1	1	
401	0	1	1	
402	0	1	1	
403	0	1	1	
404	0	1	1	
405	17	4	2	
406	0	1	1	
407	0	1	1	
408	0	1	1	
409	0	1	1	
410	0	1	1	
411	44	5	2	
412	0	1	1	
413	0	1	1	
414	0	1	1	
415	0	1	1	
416	0	1	1	
417	0	1	1	
418	0	1	1	
419	0	1	1	
420	0	1	1	
421	0	1	1	
422	0	1	1	
423	4	2	2	
424	1	2	2	
425	13	4	2	
426	0	1	1	
427	0	1	1	
428	0	1	1	
429	0	1	1	
430	0	1	1	
431	0	1	1	
432	0	1	1	
433	0	1	1	
434	0	1	1	
435	0	1	1	
436	0	1	1	
437	0	1	1	
438	0	1	1	
439	0	1	1	
440	0	1	1	
441	12	3	2	
442	0	1	1	
443	0	1	1	
444	0	1	1	
445	0	1	1	
446	0	1	1	
447	0	1	1	
448	0	1	1	
449	0	1	1	
450	0	1	1	
451	0	1	1	
452	0	1	1	
453	0	1	1	
454	0	1	1	
455	0	1	1	
456	0	1	1	
457	0	1	1	
458	0	1	1	

380	0	1	1	
381	0	1	1	
382	0	1	1	
383	0	1	1	
384	0	1	1	
385	0	1	1	
386	0	1	1	
387	0	1	1	
388	0	1	1	
389	0	1	1	
390	0	1	1	
391	0	1	1	
392	0	1	1	
393	0	1	1	
394	0	1	1	
395	0	1	1	
396	0	1	1	
397	0	1	1	
398	0	1	1	
399	0	1	1	
400	0	1	1	
401	0	1	1	
402	0	1	1	
403	0	1	1	
404	0	1	1	
405	0	1	1	
406	0	1	1	
407	0	1	1	
408	0	1	1	
409	0	1	1	
410	0	1	1	
411	0	1	1	
412	0	1	1	
413	0	1	1	
414	0	1	1	
415	2	2	2	
416	0	1	1	
417	0	1	1	
418	0	1	1	
419	0	1	1	
420	0	1	1	
421	0	1	1	
422	0	1	1	
423	0	1	1	
424	0	1	1	
425	0	1	1	
426	0	1	1	
427	0	1	1	
428	0	1	1	
429	0	1	1	
430	0	1	1	
431	0	1	1	
432	0	1	1	
433	3	4	4	
434	0	1	1	
435	0	1	1	
436	0	1	1	
437	0	1	1	
438	2	3	3	
439	0	1	1	
440	0	1	1	
441	0	1	1	
442	0	1	1	
443	0	1	1	
444	0	1	1	
445	0	1	1	
446	0	1	1	
447	0	1	1	
448	0	1	1	
449	0	1	1	
450	0	1	1	
451	0	1	1	
452	0	1	1	
453	0	1	1	
454	0	1	1	
455	0	1	1	
456	0	1	1	
457	0	1	1	
458	0	1	1	
459	0	1	1	

459	0	1	1	460	0	1	1
460	0	1	1	461	0	1	1
461	0	1	1	462	0	1	1
462	0	1	1	463	0	1	1
463	0	1	1	464	1	2	2
464	0	1	1	465	0	1	1
465	0	1	1	466	0	1	1
466	7	2	1	467	0	1	1
467	0	1	1	468	1	2	2
468	0	1	1	469	0	1	1
469	0	1	1	470	0	1	1
470	22	4	2	471	0	1	1
471	18	3	1	472	0	1	1
472	1392	32	6	473	0	1	1
473	0	1	1	474	0	1	1
474	0	1	1	475	0	1	1
475	0	1	1	476	0	1	1
476	0	1	1	477	0	1	1
477	0	1	1	478	0	1	1
478	0	1	1	479	0	1	1
479	0	1	1	480	5	3	3
480	96	3	2	481	0	1	1
481	74	6	3	482	0	1	1
482	0	1	1	483	0	1	1
483	0	1	1	484	0	1	1
484	0	1	1	485	0	1	1
485	0	1	1	486	0	1	1
486	0	1	1	487	0	1	1
487	0	1	1	488	0	1	1
488	0	1	1	489	0	1	1
489	0	1	1	490	0	1	1
490	0	1	1	491	0	1	1
491	2	2	2	492	0	1	1
492	0	1	1	493	1	2	2
493	8	2	1	494	0	1	1
494	0	1	1	495	0	1	1
495	0	1	1	496	0	1	1
496	0	1	1	497	0	1	1
497	0	1	1	498	0	1	1
498	0	1	1	499	0	1	1
499	11	3	2	500	0	1	1
500	0	1	1	501	0	1	1
501	0	1	1	502	0	1	1
502	0	1	1	503	0	1	1
503	0	1	1	504	0	1	1
504	5	2	1	505	0	1	1
505	0	1	1	506	0	1	1
506	0	1	1	507	0	1	1
507	0	1	1	508	0	1	1
508	0	1	1	509	0	1	1
509	0	1	1	510	0	1	1
510	0	1	1	511	0	1	1
511	0	1	1	512	0	1	1
512	0	1	1	513	0	1	1
513	0	1	1	514	0	1	1
514	0	1	1	515	0	1	1
515	0	1	1	516	0	1	1
516	0	1	1	517	0	1	1
517	0	1	1	518	0	1	1
518	0	1	1	519	0	1	1
519	0	1	1	520	0	1	1
520	0	1	1	521	0	1	1
521	0	1	1	522	0	1	1
522	0	1	1	523	0	1	1
523	0	1	1	524	0	1	1
524	0	1	1	525	0	1	1
525	0	1	1	526	0	1	1
526	0	1	1	527	0	1	1
527	0	1	1	528	0	1	1
528	0	1	1	529	0	1	1
529	0	1	1	530	1	2	2
530	0	1	1	531	0	1	1
531	0	1	1	532	0	1	1
532	0	1	1	533	0	1	1
533	0	1	1	534	0	1	1
534	0	1	1	535	0	1	1
535	0	1	1	536	0	1	1
536	0	1	1	537	0	1	1
537	0	1	1	538	0	1	1
538	0	1	1	539	0	1	1

539	0	1	1
540	0	1	1
541	0	1	1
542	0	1	1
543	0	1	1
544	0	1	1
545	60	4	2
546	0	1	1
547	0	1	1
548	17	2	2
549	0	1	1
550	0	1	1
551	0	1	1
552	0	1	1
553	0	1	1
554	0	1	1
555	0	1	1
556	0	1	1
557	0	1	1
558	0	1	1
559	0	1	1
560	0	1	1
561	0	1	1
562	0	1	1
563	3	2	2
564	0	1	1
565	0	1	1
566	0	1	1
567	0	1	1
568	106	3	3
569	17	2	1
570	0	1	1
571	41	3	2
572	0	1	1
573	0	1	1
574	0	1	1
575	0	1	1
576	0	1	1
577	0	1	1
578	0	1	1
579	0	1	1
580	0	1	1
581	0	1	1
582	0	1	1
583	0	1	1
584	0	1	1
585	0	1	1
586	0	1	1
587	0	1	1
588	170	6	2
589	0	1	1
590	0	1	1
591	0	1	1
592	0	1	1
593	0	1	1
594	0	1	1
595	1	2	2
596	0	1	1
597	0	1	1
598	2	2	1
599	0	1	1
600	0	1	1
601	0	1	1
602	0	1	1
603	0	1	1
604	0	1	1
605	0	1	1
606	41	4	2
607	0	1	1
608	0	1	1
609	4	2	2
610	0	1	1
611	1	2	2
612	0	1	1
613	0	1	1
614	0	1	1
615	0	1	1
616	0	1	1
617	0	1	1
618	0	1	1

540	0	1	1
541	0	1	1
542	0	1	1
543	0	1	1
544	0	1	1
545	0	1	1
546	0	1	1
547	0	1	1
548	1	2	2
549	0	1	1
550	0	1	1
551	0	1	1
552	0	1	1
553	0	1	1
554	0	1	1
555	1	2	2
556	0	1	1
557	3	2	2
558	0	1	1
559	0	1	1
560	58	21	14
561	0	1	1
562	0	1	1
563	0	1	1
564	0	1	1
565	0	1	1
566	0	1	1
567	0	1	1
568	0	1	1
569	6	3	3
570	9	8	8
571	0	1	1
572	0	1	1
573	0	1	1
574	0	1	1
575	0	1	1
576	0	1	1
577	0	1	1
578	0	1	1
579	0	1	1
580	0	1	1
581	0	1	1
582	0	1	1
583	0	1	1
584	0	1	1
585	0	1	1
586	0	1	1
587	0	1	1
588	0	1	1
589	0	1	1
590	0	1	1
591	0	1	1
592	0	1	1
593	0	1	1
594	0	1	1
595	0	1	1
596	0	1	1
597	0	1	1
598	2	2	2
599	0	1	1
600	0	1	1
601	0	1	1
602	0	1	1
603	0	1	1
604	0	1	1
605	0	1	1
606	0	1	1
607	0	1	1
608	0	1	1
609	0	1	1
610	0	1	1
611	0	1	1
612	0	1	1
613	0	1	1
614	0	1	1
615	0	1	1
616	0	1	1
617	0	1	1
618	0	1	1
619	0	1	1

619	15	2	2
620	0	1	1
621	0	1	1
622	0	1	1
623	0	1	1
624	9	3	2
625	0	1	1
626	0	1	1
627	10	2	2
628	0	1	1
629	0	1	1
630	0	1	1
631	0	1	1
632	0	1	1
633	0	1	1
634	0	1	1
635	0	1	1
636	0	1	1
637	24	4	2
638	0	1	1
639	0	1	1
640	27	3	3
641	0	1	1
642	0	1	1
643	0	1	1
644	0	1	1
645	0	1	1
646	0	1	1
647	0	1	1
648	0	1	1
649	0	1	1
650	38	2	2
651	58	4	2
652	0	1	1
653	0	1	1
654	0	1	1
655	0	1	1
656	0	1	1
657	0	1	1
658	0	1	1
659	0	1	1
660	0	1	1
661	0	1	1
662	0	1	1
663	0	1	1
664	0	1	1
665	0	1	1
666	0	1	1
667	0	1	1
668	0	1	1
669	0	1	1
670	0	1	1
671	0	1	1
672	11	3	2
673	0	1	1
674	0	1	1
675	0	1	1
676	0	1	1
677	0	1	1
678	0	1	1
679	0	1	1
680	7	2	2
681	0	1	1
682	68	6	3
683	0	1	1
684	0	1	1
685	72	5	2
686	0	1	1
687	0	1	1
688	0	1	1
689	0	1	1
690	0	1	1
691	0	1	1
692	63	5	2
693	0	1	1
694	0	1	1
695	0	1	1
696	0	1	1
697	1	2	1
698	18	3	2

620	0	1	1
621	0	1	1
622	0	1	1
623	0	1	1
624	0	1	1
625	1	2	2
626	0	1	1
627	0	1	1
628	0	1	1
629	0	1	1
630	0	1	1
631	0	1	1
632	0	1	1
633	0	1	1
634	0	1	1
635	0	1	1
636	0	1	1
637	0	1	1
638	0	1	1
639	0	1	1
640	0	1	1
641	0	1	1
642	0	1	1
643	0	1	1
644	0	1	1
645	0	1	1
646	0	1	1
647	0	1	1
648	0	1	1
649	0	1	1
650	0	1	1
651	0	1	1
652	0	1	1
653	0	1	1
654	0	1	1
655	0	1	1
656	2	2	2
657	0	1	1
658	0	1	1
659	0	1	1
660	0	1	1
661	0	1	1
662	0	1	1
663	2	2	2
664	0	1	1
665	0	1	1
666	1	2	2
667	0	1	1
668	0	1	1
669	0	1	1
670	0	1	1
671	0	1	1
672	0	1	1
673	0	1	1
674	0	1	1
675	0	1	1
676	0	1	1
677	0	1	1
678	0	1	1
679	0	1	1
680	0	1	1
681	0	1	1
682	0	1	1
683	0	1	1
684	4	3	3
685	0	1	1
686	0	1	1
687	0	1	1
688	0	1	1
689	0	1	1
690	0	1	1
691	0	1	1
692	0	1	1
693	0	1	1
694	0	1	1
695	0	1	1
696	0	1	1
697	0	1	1
698	0	1	1
699	0	1	1

699	0	1	1
700	0	1	1
701	0	1	1
702	0	1	1
703	0	1	1
704	0	1	1
705	0	1	1
706	0	1	1
707	0	1	1
708	0	1	1
709	0	1	1
710	0	1	1
711	0	1	1
712	0	1	1
713	0	1	1
714	0	1	1
715	0	1	1
716	0	1	1
717	0	1	1
718	0	1	1
719	0	1	1
720	0	1	1
721	0	1	1
722	9	2	2
723	63	7	2
724	0	1	1
725	0	1	1
726	0	1	1
727	0	1	1
728	0	1	1
729	0	1	1
730	0	1	1
731	0	1	1
732	18	2	2
733	0	1	1
734	0	1	1
735	0	1	1
736	0	1	1
737	0	1	1
738	26	3	2
739	0	1	1
740	0	1	1
741	0	1	1
742	0	1	1
743	0	1	1
744	0	1	1
745	0	1	2

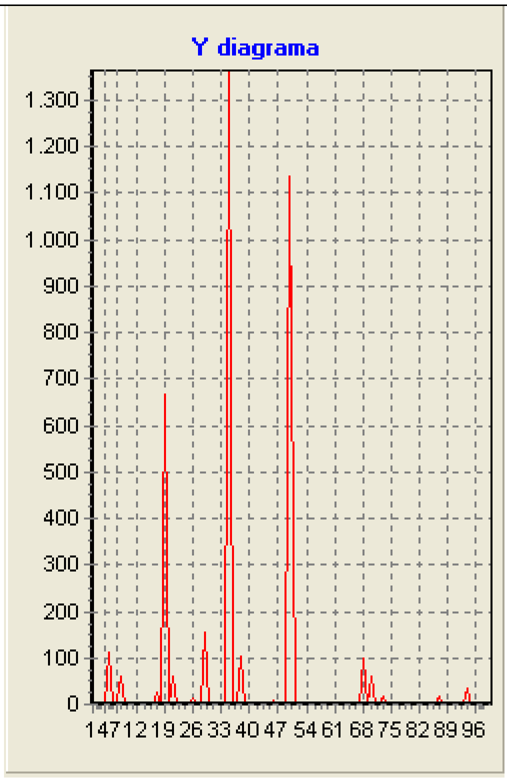
700	0	1	1
701	0	1	1
702	0	1	1
703	0	1	1
704	0	1	1
705	0	1	1
706	0	1	1
707	0	1	1
708	0	1	1
709	0	1	1
710	0	1	1
711	0	1	1
712	0	1	1
713	0	1	1
714	0	1	1
715	0	1	1
716	0	1	1
717	0	1	1
718	0	1	1
719	0	1	1
720	0	1	1
721	0	1	1
722	1	2	2
723	0	1	1
724	0	1	1
725	0	1	1
726	0	1	1
727	0	1	1
728	0	1	1
729	0	1	1
730	0	1	1
731	0	1	1
732	0	1	1
733	0	1	1
734	0	1	1
735	0	1	1
736	0	1	1
737	0	1	1
738	0	1	1
739	0	1	1
740	0	1	1
741	0	1	1
742	0	1	1
743	0	1	1
744	0	1	1
745	0	1	1
746	3	4	4
747	0	1	1
748	6	4	4
749	2	3	2
750	0	1	1
751	0	1	1
752	0	1	1
753	0	1	1
754	0	1	1
755	0	1	1
756	0	1	1
757	0	1	1
758	3	4	4
759	0	1	1
760	0	1	1
761	0	1	1
762	0	1	1
763	0	1	1
764	0	1	1
765	0	1	1
766	0	1	1
767	0	1	1
768	0	1	1
769	1	2	2
770	0	1	1
771	0	1	1
772	1	2	2
773	0	1	1
774	0	1	1
775	0	1	1
776	0	1	1
777	0	1	1
778	0	1	1
779	0	1	1

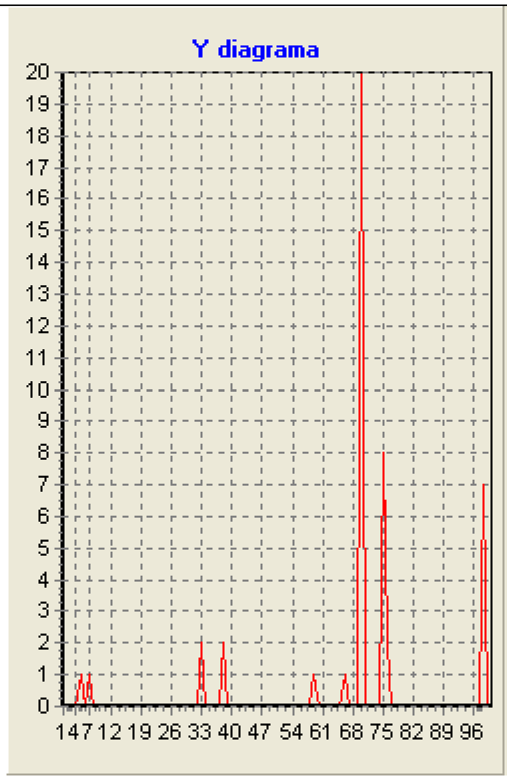
780	0	1	1
781	0	1	1
782	0	1	1
783	1	2	2
784	0	1	1
785	0	1	1
786	0	1	1
787	0	1	1
788	0	1	1
789	0	1	1
790	0	1	1
791	0	1	1
792	0	1	1
793	0	1	1
794	0	1	1
795	0	1	1
796	0	1	1
797	0	1	1
798	0	1	1
799	0	1	1
800	0	1	1
801	0	1	1
802	0	1	1
803	0	1	1
804	0	1	1
805	0	1	1
806	0	1	1
807	0	1	1
808	0	1	1
809	1	2	2
810	0	1	1
811	0	1	1
812	0	1	1
813	0	1	1
814	0	1	1
815	0	1	1
816	0	1	1
817	0	1	1
818	0	1	1
819	0	1	1
820	0	1	1
821	0	1	1
822	0	1	1
823	0	1	1
824	0	1	1
825	0	1	1
826	0	1	1
827	0	1	1
828	0	1	1
829	0	1	1
830	0	1	1
831	0	1	1
832	0	1	1
833	0	1	1
834	0	1	1
835	0	1	1
836	0	1	1
837	0	1	1
838	0	1	1
839	0	1	1
840	0	1	1
841	2	2	2
842	0	1	1
843	0	1	1
844	1	2	2
845	4	4	4
846	0	1	1
847	0	1	1
848	0	1	1
849	0	1	1
850	0	1	1
851	0	1	1
852	0	1	1
853	0	1	1
854	0	1	1
855	0	1	1
856	2	3	3
857	0	1	1
858	0	1	1
859	0	1	1

<table>
<tr>
<td></td>
<td>| 860 | 0 | 1 | 1 |
861	0	1	1
862	0	1	1
863	0	1	1
864	0	1	1
865	0	1	1
866	0	1	1
867	0	1	1
868	0	1	1
869	0	1	1
870	0	1	1 | ---</td>
</tr>
<tr>
<td>Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (3,36; 7,91);
Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (1,18; 1,29);</td>
<td>Vidutinio buvimo laiko eilėje pasikliautinasis intervalas: (0,13; 0,34);
Vidutinio maksimalaus eilės ilgio pasikliautinasis intervalas: (1,08; 1,13);</td>
</tr>
</table>

8 lentelė

Paraiškų laukimo laikų ilgių sumų cikluose grafikai





3 PRIEDAS. PROGRAMOS TEKSTAS

```
//------------Programos tekstas – failas Unit1.cpp----------------------

#include <vcl.h>
#include <stdio.h>
#pragma hdrstop
#include <math.h>
#include "Unit1.h"
#include "Unit2.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma link "Excel_2K_SRVR"
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm1::FormActivate(TObject *Sender)
{
RadioButton1 -> Checked = false;
RadioButton2 -> Checked = false;
RadioButton3 -> Checked = false;
RadioButton4 -> Checked = false;
RadioButton5 -> Checked = false;
RadioButton6 -> Checked = false;
TabSheet1 -> Caption = "Eksponentinis modelis";
TabSheet2 -> Caption = "Tolygusis modelis";
TabSheet3 -> Caption = "Tolygusis - Determinuotas modelis";
Chart1 -> Visible = false;
Chart2 -> Visible = false;
Chart3 -> Visible = false;

AboutBox -> ShowModal();
PageControl1 -> Pages[0] -> Show();
Memo1 -> Clear();
Memo2 -> Clear();
Memo3 -> Clear();
StringGrid1 -> Visible = false;
StringGrid2 -> Visible = false;
StringGrid3 -> Visible = false;
Memo1 -> Visible = false;
Memo2 -> Visible = false;
Memo3 -> Visible = false;
Label4 -> Caption = "";
Label5 -> Caption = "";
Label6 -> Caption = "";
Label1 -> Caption = "";
Label2 -> Caption = "";
Label3 -> Caption = "";
Button1 -> Enabled = true;
Button3 -> Enabled = true;
Button5 -> Enabled = true;
Button2 -> Enabled = true;
Button4 -> Enabled = true;
Button6 -> Enabled = true;
Button7 -> Enabled = true;
Button8 -> Enabled = true;
Button9 -> Enabled = true;
}
//--------Puslapiu pervertimas---------------------------------------------
void __fastcall TForm1::PageControl1Change(TObject *Sender)
{
```

```
RadioButton1 -> Checked = false;
RadioButton2 -> Checked = false;
RadioButton3 -> Checked = false;
RadioButton4 -> Checked = false;
RadioButton5 -> Checked = false;
RadioButton6 -> Checked = false;
Memo1 -> Clear();
Memo2 -> Clear();
Memo3 -> Clear();
StringGrid1 -> Visible = false;
StringGrid2 -> Visible = false;
StringGrid3 -> Visible = false;
Memo1 -> Visible = false;
Memo2 -> Visible = false;
Memo3 -> Visible = false;
Label1 -> Caption = "";
Label2 -> Caption = "";
Label3 -> Caption = "";
Label4 -> Caption = "";
Label5 -> Caption = "";
Label6 -> Caption = "";
Chart1 -> Visible = false;
Chart2 -> Visible = false;
Chart3 -> Visible = false;

Button1 -> Enabled = true;
Button3 -> Enabled = true;
Button5 -> Enabled = true;
Button2 -> Enabled = true;
Button4 -> Enabled = true;
Button6 -> Enabled = true;
Button7 -> Enabled = true;
Button8 -> Enabled = true;
Button9 -> Enabled = true;
}
//--------Mygtukas Vidutinis buvimo laikas eileje----------------------------
//--------modelyje EE--------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Chart1 -> Visible = true;
Label4 -> Caption = "Rezultatai faile 'ee.txt'";
Memo1 -> Visible = true;
StringGrid3 -> Visible = true;
Memo1 -> Clear();
Label1 -> Caption = "Rezultatai:";
if (RadioButton1->Checked)
{
Memo1 -> Clear();
EEmodel();
EM();
SAM();
SIntervalas();
IFaila("ee.txt");
Rodyti1();
Diagrama1();}
else if (RadioButton4 -> Checked)
{
Memo1 -> Clear();
EE3model();
EM();
SAM();
SIntervalas();
IFaila("ee.txt");
Rodyti1();
Diagrama1();

}
else {Memo1 -> Lines -> Add("Pasirinkite sistema");
      Chart1 -> Visible = false;
      StringGrid3 -> Visible = false;
      Label4 -> Caption = "";}
```

```
}
//--------Mygtukas Vidutinis buvimo laikas eileje----------------------------
//--------modelyje TT--------------------------------------------------------
void __fastcall TForm1::Button3Click(TObject *Sender)
{
Chart2 -> Visible = true;
Label5 -> Caption = "Rezultatai faile 'tt.txt'";
Memo2 -> Visible = true;
StringGrid2 -> Visible = true;
Memo2 -> Clear();
Label2 -> Caption = "Rezultatai:";
if (RadioButton2->Checked)
{
Memo2 -> Clear();
TTmodel();
EM();
SAM();
SIntervalas();
IFaila("tt.txt");
Rodyti2();
Diagrama2();}
else if (RadioButton3->Checked)
{
Memo2 -> Clear();
TT3model();
EM();
SAM();
SIntervalas();
IFaila("tt.txt");
Rodyti2();
Diagrama2();}
else {Memo2 -> Lines -> Add("Pasirinkite sistema");
     Chart2 -> Visible = false;
      StringGrid2 -> Visible = false;
      Label5 -> Caption = "";}
}
//--------Mygtukas Vidutinis buvimo laikas eileje ---------------------------
//--------modelyje TD--------------------------------------------------------
void __fastcall TForm1::Button5Click(TObject *Sender)
{
Chart3 -> Visible = true;
Label6 -> Caption = "Rezultatai faile 'td.txt'";
Memo3 -> Visible = true;
StringGrid1 -> Visible = true;
Memo3 -> Clear();
Label3 -> Caption = "Rezultatai:";
if (RadioButton5->Checked)
{
Memo3 -> Clear();
TDmodel();
EM();
SAM();
SIntervalas();
IFaila("td.txt");
Rodyti3();
Diagrama3();}
else if(RadioButton6->Checked)
{
Memo3 -> Clear();
TD3model();
EM();
SAM();
SIntervalas();
IFaila("td.txt");
Rodyti3();
Diagrama3();}
else {Memo3 -> Lines -> Add("Pasirinkite sistema");
     Chart3 -> Visible = false;
     StringGrid1 -> Visible = false;
     Label6 -> Caption = "";}
}
```

```
//--------Mygtukas Maksimalus eiles ilgis modelyje EE------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Chart1 -> Visible = false;
Memo1 -> Visible = true;
Label4 -> Caption = "Rezultatai faile 'ee.txt'";
StringGrid3 -> Visible = true;
Memo1 -> Clear();
Label1 -> Caption = "Rezultatai:";
if (RadioButton1->Checked)
{
Memo1 -> Clear();
EEmodel();
EM();
SAM();
MIntervalas();
IFaila("ee.txt");
Rodyti1();
Diagrama1();}
else if (RadioButton4 -> Checked)
{
Memo1 -> Clear();
EE3model();
EM();
SAM();
MIntervalas();
IFaila("ee.txt");
Rodyti1();
Diagrama1();
}
else {Memo1 -> Lines -> Add("Pasirinkite sistema");
      Chart1 -> Visible = false;
      StringGrid3 -> Visible = false;
      Label4 -> Caption = "";}
}
//--------Mygtukas Maksimalus eiles ilgis modelyje TT------------------------
void __fastcall TForm1::Button4Click(TObject *Sender)
{
Chart2 -> Visible = false;
Memo2 -> Visible = true;
Label5 -> Caption = "Rezultatai faile 'tt.txt'";
StringGrid2 -> Visible = true;
Memo2 -> Clear();
Label2 -> Caption = "Rezultatai:";
if (RadioButton2->Checked)
{
Memo2 -> Clear();
TTmodel();
EM();
SAM();
MIntervalas();
IFaila("tt.txt");
Rodyti2();
Diagrama2();}
else if (RadioButton3->Checked)
{
Memo2 -> Clear();
TT3model();
EM();
SAM();
MIntervalas();
IFaila("tt.txt");
Rodyti2();
Diagrama2();}
else {Memo2 -> Lines -> Add("Pasirinkite sistema");
     Chart2 -> Visible = false;
      StringGrid2 -> Visible = false;
      Label5 -> Caption = "";}
}
//--------Mygtukas Maksimalus eiles ilgis modelyje TD------------------------
void __fastcall TForm1::Button6Click(TObject *Sender)
```

```
{
Chart3 -> Visible = false;
Memo3 -> Visible = true;
Label6 -> Caption = "Rezultatai faile 'td.txt'";
StringGrid1 -> Visible = true;
Memo3 -> Clear();
Label3 -> Caption = "Rezultatai:";
if (RadioButton5->Checked)
{
Memo3 -> Clear();
TDmodel();
EM();
SAM();
MIntervalas();
IFaila("td.txt");
Rodyti3();
Diagrama3();}
else if(RadioButton6->Checked)
{
Memo3 -> Clear();
TD3model();
EM();
SAM();
MIntervalas();
IFaila("td.txt");
Rodyti3();
Diagrama3();}
else {Memo3 -> Lines -> Add("Pasirinkite sistema");
     Chart3 -> Visible = false;
     StringGrid1 -> Visible = false;
     Label6 -> Caption = "";}
}
//--------Mygtukas Paaiskinimai modelyje EE---------------------------------
void __fastcall TForm1::Button7Click(TObject *Sender)
{
Chart1 -> Visible = false;
Memo1 -> Visible = true;
Label4 -> Caption = "";
StringGrid3 -> Visible = false;
Memo1 -> Clear();
AnsiString E;
Label1 -> Caption = "Paaiskinimai:";
E = "Regeneracinis momentas: Bent vienas kanalas laisvas ";
Memo1 -> Lines -> Add(E);
E = "";
Memo1 -> Lines -> Add(E);
E = "Paraisku atejimo ir laukimo eileje laiko momentai ";
E = E + "pasiskirste pagal eksponentini desni.";
Memo1 -> Lines -> Add(E);
}
//--------Mygtukas Paaiskinimai modelyje TT---------------------------------
void __fastcall TForm1::Button8Click(TObject *Sender)
{
Memo2 -> Visible = true;
Label5 -> Caption = "";
StringGrid2 -> Visible = false;
Memo2 -> Clear();
AnsiString E;
Label2 -> Caption = "Paaiskinimai:";
E = "Regeneracinis momentas: Bent vienas kanalas laisvas ";
Memo2 -> Lines -> Add(E);
E = "";
Memo2 -> Lines -> Add(E);
E = "Paraisku atejimo ir laukimo eileje laiko momentai ";
E = E + "pasiskirste pagal tolyguji desni.";
Memo2 -> Lines -> Add(E);
}
//--------Mygtukas Paaiskinimai modelyje TD---------------------------------
void __fastcall TForm1::Button9Click(TObject *Sender)
{
Memo3 -> Visible = true;
```

```
Label6 -> Caption = "";
StringGrid1 -> Visible = false;
AnsiString E;
Memo3 -> Clear();
Label3 -> Caption = "Paaiskinimai:";
E = "Regeneracinis momentas: Bent vienas kanalas laisvas ";
Memo3 -> Lines -> Add(E);
E = "";
Memo3 -> Lines -> Add(E);
E = "Paraiskos generuojamos determinuotais laiko momentais, ";
E = E + "laukimo eileje laiko momentai pasiskirste pagal tolyguji desni.";
Memo3 -> Lines -> Add(E);
}
//--------Spausdinimas i faila -----------------------------------------------
void TForm1::IFaila(char *pav)
{
 int i,j;
   FILE *FR;
   FR=fopen(pav,"w");
   fprintf(FR,"---------------------------------------\n");
   fprintf(FR,"|Ciklas |     S    |   alfa  |   M     |\n");
   fprintf(FR,"---------------------------------------\n");
   for (i=1; i<=c; i++){
      fprintf(FR, "| %4d  |",i);
      fprintf(FR, " %5d    |", S[i]);
      fprintf(FR, " %4d    |", A[i]);
       fprintf(FR, " %4d    |",M[i]);
      fprintf(FR, " \n");
      }
   fprintf(FR,"---------------------------------------\n");
   fclose(FR);

}
//------------Duomenu isvedimo i ekrana procedura----------------------------
void TForm1::Rodyti3(){
  int i,j;
  StringGrid1->ColCount = 4;
  StringGrid1->RowCount = c+1;
  StringGrid1->Cells[0][0]=" Ciklas";
  StringGrid1->Cells[1][0]=" Y";
  StringGrid1->Cells[2][0]=" alfa";
  StringGrid1->Cells[3][0]=" Z";
  for (i=0; i <= c; i++)
     StringGrid1->Cells[0][i+1]=IntToStr(i+1);
  for (i=0; i<=c; i++)
     StringGrid1->Cells[1][i+1]=IntToStr(S[i]);
  for (i=0; i<=c; i++)
     StringGrid1->Cells[2][i+1]=IntToStr(A[i]);
  for (i=0; i<=c; i++)
     StringGrid1->Cells[3][i+1]=IntToStr(M[i]);
}
void TForm1::Rodyti2(){
  int i,j;
  StringGrid2->ColCount = 4;
  StringGrid2->RowCount = c+1;
  StringGrid2->Cells[0][0]=" Ciklas";
  StringGrid2->Cells[1][0]=" Y";
  StringGrid2->Cells[2][0]=" alfa";
  StringGrid2->Cells[3][0]=" Z";
  for (i=0; i <= c; i++)
     StringGrid2->Cells[0][i+1]=IntToStr(i+1);
  for (i=0; i<=c; i++)
     StringGrid2->Cells[1][i+1]=IntToStr(S[i]);
  for (i=0; i<=c; i++)
     StringGrid2->Cells[2][i+1]=IntToStr(A[i]);
  for (i=0; i<=c; i++)
     StringGrid2->Cells[3][i+1]=IntToStr(M[i]);
}
void TForm1::Rodyti1(){
  int i,j;
  StringGrid3->ColCount = 4;
```

```
  StringGrid3->RowCount = c+1;
  StringGrid3->Cells[0][0]=" Ciklas";
  StringGrid3->Cells[1][0]=" Y";
  StringGrid3->Cells[2][0]=" alfa";
  StringGrid3->Cells[3][0]=" Z";
  for (i=0; i <= c; i++)
     StringGrid3->Cells[0][i+1]=IntToStr(i+1);
  for (i=0; i<=c; i++)
     StringGrid3->Cells[1][i+1]=IntToStr(S[i]);
  for (i=0; i<=c; i++)
     StringGrid3->Cells[2][i+1]=IntToStr(A[i]);
  for (i=0; i<=c; i++)
     StringGrid3->Cells[3][i+1]=IntToStr(M[i]);
}

//--------Modeliuojama sistema EE--------------------------------------------
void TForm1::EEmodel()
{
int lauks1, lauks2, kt,kt1;
lauks1 = 0; lauks2 = 0; kt =0;
for(int i = 0; i <= nn; i++)
Y[i]=0;
for(int i = 0; i <= nn; i++)
{
if (lauks1 <= lauks2)
        if (kt >= lauks1)
        {lauks1 = kt + eksponentinis(0.02); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks1 - kt); lauks1 = lauks1 + eksponentinis(0.02);L[i] = kt1;}
else if (kt >= lauks2)
        {lauks2 = kt + eksponentinis(0.02); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks2 - kt); lauks2 = lauks2 + eksponentinis(0.02);L[i] = kt1;}
kt1 = eksponentinis(0.02);
kt = kt + kt1;
}
}
//--------Modeliuojama sistema EE3--------------------------------------------
void TForm1::EE3model()
{
int lauks1, lauks2,lauks3, kt,kt1;
lauks1 = 0; lauks2 = 0;lauks3 =0; kt =0;
for(int i = 0; i <= nn; i++)
Y[i]=0;
for(int i = 0; i <= nn; i++)
{
if (lauks1 <= lauks2)
        if (kt >= lauks1)
        {lauks1 = kt + eksponentinis(0.02); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks1 - kt); lauks1 = lauks1 + eksponentinis(0.02);L[i] = kt1;}
else if (lauks2 <= lauks3)
        if (kt >= lauks2)
        {lauks2 = kt + eksponentinis(0.02); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks2 - kt); lauks2 = lauks2 + eksponentinis(0.02) ;L[i] = kt1;}
else if (kt >= lauks3) {lauks3 = kt + eksponentinis(0.02); Y[i] = Y[i];L[i] = kt1;}
     else {Y[i] = abs(lauks3 - kt); lauks3 = lauks3 + eksponentinis(0.02);L[i] = kt1;}
kt1 = eksponentinis(0.02);
kt = kt + kt1;
}
}

//--------Modeliuojama sistema TT--------------------------------------------
void TForm1::TTmodel()
{
int lauks1, lauks2, kt,kt1;
lauks1 = 0; lauks2 = 0; kt =0;
for(int i = 0; i <= nn; i++)
Y[i]=0;
for(int i = 0; i <= nn; i++)
{
if (lauks1 <= lauks2)
        if (kt >= lauks1)
        {lauks1 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
```

```
        else {Y[i] = abs(lauks1 - kt); lauks1 = lauks1 + tolyg(30,90);L[i] = kt1;}
else if (kt >= lauks2)
        {lauks2 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks2 - kt); lauks2 = lauks2 + tolyg(30,90);L[i] = kt1;}
kt1 =tolyg(30,35);
kt = kt + kt1;
}
}
//--------Modeliuojama sistema TT3-------------------------------------------
void TForm1::TT3model()
{
int lauks1, lauks2,lauks3, kt,kt1;
lauks1 = 0; lauks2 = 0; kt =0;
for(int i = 0; i <= nn; i++)
Y[i]=0;
for(int i = 0; i <= nn; i++)
{
if (lauks1 <= lauks2)
        if (kt >= lauks1)
        {lauks1 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks1 - kt); lauks1 = lauks1 + tolyg(30,90);L[i] = kt1;}
else if (lauks2 <= lauks3)
        if (kt >= lauks2)
        {lauks2 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks2 - kt); lauks2 = lauks2 + tolyg(30,90);L[i] = kt1;}
else if (kt >= lauks3) {lauks3 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
     else {Y[i] = abs(lauks3 - kt); lauks3 = lauks3 + tolyg(30,90);L[i] = kt1;}
kt1 =tolyg(30,35);
kt = kt + kt1;
}
}

//--------Modeliuojama sistema TD--------------------------------------------
void TForm1::TDmodel()
{
int lauks1, lauks2, kt,kt1;
lauks1 = 0; lauks2 = 0; kt =0;
for(int i = 0; i <= nn; i++)
Y[i]=0;
for(int i = 0; i <= nn; i++)
{
if (lauks1 <= lauks2)
        if (kt >= lauks1)
        {lauks1 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks1 - kt); lauks1 = lauks1 + tolyg(30,90);L[i] = kt1;}
else if (kt >= lauks2)
        {lauks2 = kt + tolyg(31,90); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks2 - kt); lauks2 = lauks2 + tolyg(31,90);L[i] = kt1;}
kt1 = tolyg(30,35);
kt = kt + 35;
}
}
//--------Modeliuojama sistema TD3-------------------------------------------
void TForm1::TD3model()
{
int lauks1, lauks2,lauks3,  kt,kt1;
lauks1 = 0; lauks2 = 0; kt =0;
for(int i = 0; i <= nn; i++)
Y[i]=0;
for(int i = 0; i <= nn; i++)
{
if (lauks1 <= lauks2)
        if (kt >= lauks1)
        {lauks1 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks1 - kt); lauks1 = lauks1 + tolyg(30,90);L[i] = kt1;}
else if (lauks2 <= lauks3)
        if (kt >= lauks2)
        {lauks2 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
        else {Y[i] = abs(lauks2 - kt); lauks2 = lauks2 + tolyg(30,90);L[i] = kt1;}
else if (kt >= lauks3)
        {lauks3 = kt + tolyg(30,90); Y[i] = Y[i];L[i] = kt1;}
```

```
     else {Y[i] = abs(lauks3 - kt); lauks3 = lauks3 + tolyg(30,90);L[i] = kt1;}
kt1 = tolyg(30,35);
kt = kt + 35;
}
}
//--------Ieskomas Eiles ilgis ir maksimalus eiles ilgiai--------------------
void TForm1::EM()
{
int  i;
i = 1;
Eil[0] = 1;
Y[0] = 0;
while (i <= nn)
{
if (Y[i] == 0) {Eil[i] = 1; i += 1;}
else if (L[i] > L[i-1]) {Eil[i] = 1; i += 1;}
else {Eil[i] = Eil[i-1] + 1; i += 1;}
}
}
//--------Ieskomos Sumos, alfa ir maksimumai---------------------------------
void TForm1::SAM()
{
int j, i,c_sk;
j = 0;
S[0] = Y[0];
A[0] = 1;
M[0] =1;
i = 1;
while (i <= nn)
{
if (Y[i] == 0) {j += 1; S[j] = Y[i]; A[j] = 1; M[j] = Eil[i]; i += 1;c_sk = j;}
else if (Eil[i] > M[j])
        {M[j] = Eil[i]; S[j] += Y[i]; A[j] += 1; i += 1; c_sk = j;}
     else {S[j] += Y[i]; A[j] += 1; i += 1; c_sk = j;}
}
c = c_sk;
}
//--------Skaiciuojamas vidutinio buvimo laiko eileje------------------------
//--------pasikliautinasis intervalas----------------------------------------
void TForm1::SIntervalas()
{
 AnsiString E;
 for (int i=0; i<=c; i++){
    Y_vid    += S[i];
    alfa_vid += A[i];
    Y_kv += (S[i]*S[i]);
    alfa_kv += (A[i]*A[i]);
    Y_alfa += (S[i]*A[i]);
 }
 r = Y_vid/alfa_vid;
 Y_alfa_sum = Y_vid*alfa_vid;
 s11 = Y_kv/(c-1)-Y_vid*Y_vid/(c*(c-1));
 s22 = alfa_kv/(c-1)-alfa_vid*alfa_vid/(c*(c-1));
 s12 = Y_alfa/(c-1)- Y_alfa_sum/(c*(c-1));
 s = s11 - 2*r*s12 + r*r*s22;
 if (s < 0) {Memo1->Lines->Add("Neracionalus duomenys");
 Memo2->Lines->Add("Neracionalus duomenys");Memo3->Lines->Add("Neracionalus duomenys");}
   else
   s = sqrt(s);

   s_kt = ((1.645)*s)/((alfa_vid/c)*sqrt(c));
   Ik = r - s_kt;
   if (Ik < 0) Ik = 0;
   Id = r + s_kt;
   I=Id-Ik;
 E = "Vidutinio buvimo laiko eileje pasikliautinasis intervalas:";
 Memo1 -> Lines -> Add(E);
 Memo2 -> Lines -> Add(E);
 Memo3 -> Lines -> Add(E);

 E ="";
```

```
 Memo1 -> Lines -> Add(E);
 Memo2 -> Lines -> Add(E);
 Memo3 -> Lines -> Add(E);

 E = "(" + FloatToStrF(Ik, ffFixed, 4,2);
 E = E + "; ";
 E = E + FloatToStrF(Id, ffFixed, 4,2);
 E= E + ")";
 Memo1 -> Lines -> Add(E);
 Memo2 -> Lines -> Add(E);
 Memo3 -> Lines -> Add(E);

 E ="";
 Memo1 -> Lines -> Add(E);
 Memo2 -> Lines -> Add(E);
 Memo3 -> Lines -> Add(E);

 E = "Vidutinio buvimo laiko eileje ivertis:";
 Memo1 -> Lines -> Add(E);
 Memo2 -> Lines -> Add(E);
 Memo3 -> Lines -> Add(E);

 E = "";
 Memo1 -> Lines -> Add(E);
 Memo2 -> Lines -> Add(E);
 Memo3 -> Lines -> Add(E);

 E = FloatToStrF(r,ffFixed, 4,2);
 Memo1 -> Lines -> Add(E);
 Memo2 -> Lines -> Add(E);
 Memo3 -> Lines -> Add(E);

 }
//--------Skaiciuojamas maksimalaus eiles ilgio------------------------------
//--------pasikliautinasis intervalas----------------------------------------
void TForm1::MIntervalas()
{
AnsiString E;
float SZ, SZZ,s1;
for (int i = 0; i <= c; i++)    {SZ += A[i]; SZZ += (A[i]*A[i]);}
r1 = SZ/c;
s2 = ((SZZ)/(c-1))-((SZ*SZ)/(c*(c-1)));
if (s2 < 0) {Memo1 -> Lines -> Add("Neracionalus duomenys!");
Memo2 -> Lines -> Add("Neracionalus duomenys!");
Memo3 -> Lines -> Add("Neracionalus duomenys!");}
else
s1 = sqrt(s2);
Ikk = r1 - ((1.645)*s1)/(sqrt(c));
Idd = r1 + ((1.645)*s1)/(sqrt(c));
II = Idd - Ikk;
E = "Vidutinio maksimalaus eiles ilgio pasikliautinasis intervalas:";
Memo1 -> Lines -> Add(E);
Memo2 -> Lines -> Add(E);
Memo3 -> Lines -> Add(E);

E = "";
Memo1 -> Lines -> Add(E);
Memo2 -> Lines -> Add(E);
Memo3 -> Lines -> Add(E);

E = "(" + FloatToStrF(Ikk, ffFixed, 4,2);
E = E + "; ";
E = E + FloatToStrF(Idd, ffFixed, 4,2);
E= E + ")";
Memo1 -> Lines -> Add(E);
Memo2 -> Lines -> Add(E);
Memo3 -> Lines -> Add(E);

E = "";
Memo1 -> Lines -> Add(E);
Memo2 -> Lines -> Add(E);
```

```
Memo3 -> Lines -> Add(E);

E= "Vidutinio maksimalaus eiles ilgio ivertis:";
Memo1 -> Lines -> Add(E);
Memo2 -> Lines -> Add(E);
Memo3 -> Lines -> Add(E);

E="";
Memo1 -> Lines -> Add(E);
Memo2 -> Lines -> Add(E);
Memo3 -> Lines -> Add(E);

E = FloatToStrF(r1,ffFixed, 4, 2);
Memo1 -> Lines -> Add(E) ;
Memo2 -> Lines -> Add(E);
Memo3 -> Lines -> Add(E);

}

//--------Piesiama sumu diagrama---------------------------------------------
void TForm1::Diagrama1()
{
Series1 -> Clear();
for(int i=1; i<=100;i++)
Series1 -> AddXY(i,S[i],i,clRed);
}
//--------Piesiama sumu diagrama---------------------------------------------
void TForm1::Diagrama2()
{
Series2 -> Clear();
for(int i=1; i<=100;i++)
Series2 -> AddXY(i,S[i],i,clBlue);
}
//--------Piesiama sumu diagrama---------------------------------------------
void TForm1::Diagrama3()
{
Series3 -> Clear();
for(int i=1; i<=100;i++)
Series3 -> AddXY(i,S[i],i,clGreen);
}


//--------Generuojamas eksponentinis atsitiktinis dydis----------------------
int TForm1::eksponentinis(float b){
  float a_sk;
  int max, expo, t;

  max=RAND_MAX+10;
  t=rand();
  a_sk=1*(t-1);
  a_sk=a_sk/(max-1);
  expo=ceil(((-1)*log10(1-a_sk)/b));
 return (expo);
}

//--------Generuojamas tolygusis atsitiktinis dydis--------------------------
int TForm1::tolyg(int a, int b){
  int max;
  float t;
  int tol;
  max=RAND_MAX;
  t=rand();
  t=1*(t-1);
  t=t/(max-1);
  tol=abs(t*(b-a)+a);
return (tol);
}
```

//-----------*Klasės tipas* – ***failas Unit1.***h-----------------------------------

```
//-----------------------------------------------------------------------------

#ifndef Unit1H
#define Unit1H
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <Grids.hpp>
#include <Chart.hpp>
#include <DBChart.hpp>
#include <ExtCtrls.hpp>
#include <Series.hpp>
#include <TeEngine.hpp>
#include <TeeProcs.hpp>
#include "Excel_2K_SRVR.h"
#include <OleServer.hpp>
const nn = 1000;
//---------------------------------------------------------------------------
class TForm1 : public TForm
{
__published:// IDE-managed Components
        TPageControl *PageControl1;
        TTabSheet *TabSheet1;
        TTabSheet *TabSheet2;
        TTabSheet *TabSheet3;
        TButton *Button1;
        TButton *Button2;
        TButton *Button3;
        TButton *Button4;
        TButton *Button5;
        TButton *Button6;
        TButton *Button7;
        TButton *Button8;
        TButton *Button9;
        TMemo *Memo1;
        TMemo *Memo2;
        TMemo *Memo3;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label3;
        TLabel *Label4;
        TLabel *Label5;
        TLabel *Label6;
        TStringGrid *StringGrid1;
        TStringGrid *StringGrid2;
        TStringGrid *StringGrid3;
        TChart *Chart1;
        TLineSeries *Series1;
        TChart *Chart2;
        TLineSeries *Series2;
        TChart *Chart3;
        TLineSeries *Series3;
        TRadioButton *RadioButton1;
        TRadioButton *RadioButton2;
        TRadioButton *RadioButton3;
        TRadioButton *RadioButton4;
        TRadioButton *RadioButton5;
        TRadioButton *RadioButton6;


        void __fastcall FormActivate(TObject *Sender);
        void __fastcall Button5Click(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall Button2Click(TObject *Sender);
        void __fastcall Button3Click(TObject *Sender);
        void __fastcall Button4Click(TObject *Sender);
```

```
        void __fastcall Button6Click(TObject *Sender);
        void __fastcall Button9Click(TObject *Sender);
        void __fastcall Button7Click(TObject *Sender);
        void __fastcall Button8Click(TObject *Sender);
        void __fastcall PageControl1Change(TObject *Sender);



private:// User declarations
        void EEmodel();      //Modeliuoja EE
        void EE3model();
        void TTmodel();      //Modeliuoja TT
        void TT3model();
        void TDmodel();      //Modeliuoja TD
        void TD3model();
        void SAM();          //Ieskomos sumos, alfa, maksimumai
        void SIntervalas();   //E{W} pasikliautinasis intervalas

        void MIntervalas();   //E{Z} pasikliautinasis intervalas
        void IFaila(char *); //Isveda duomenis i faila
        void IFailaS(char *);
        void IFailaA(char *);
        void IFailaM(char *);
        void Diagrama0();
        void Diagrama1();

        void Diagrama2();
        void Diagrama3();
        void Diagrama4();
        int eksponentinis(float);
        int tolyg(int , int );
        void EM();
        void Rodyti1();
        void Rodyti2();
        void Rodyti3();
        float l1, l2, l3,
        a1,a2,a3,b1,b2,b3;
        float r,r1,rb,rt,rf,             //Ivertis
              s11,
              s22,
              s12,
              s,sb,st,sf,
              s2,
              s_kt,s_ktb,s_ktt,s_ktf,
              skv,
              kd,D,
              Ik,Ikk,Ikt,Ikb,Ikf,
              Id,Idd,Idt,Idf,Idb,
              I,II,Ib,It,If;
        float Y_vid,       // Y suma
               alfa_vid,    // alfa suma
               Y_kv,        // Y kvadratu suma
               alfa_kv,     // alfa kvadratu suma
               Y_alfa,      // Y padauginta is alfa suma
               Y_alfa_sum;  //pagalbinis kintamasis
        int Y[nn];           //Laukimo laikai
        int S[nn];           //Laukimo laiku sumos cikluose
        int A[nn];           //Paraisku skaicius cikluose
        int M[nn];           //Maksimalus eiles ilgis cikluose
        int Eil[nn];
        int L[nn];
        int c;               //Ciklu skaicius
public:     // User declarations
        __fastcall TForm1(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm1 *Form1;
//---------------------------------------------------------------------------
#endif
```

4 PRIEDAS. KONFERENCIJOS „V STUDENTŲ KONFERENCIJA“ STRAIPSNIO MEDŽIAGA

MAKSIMALIOS EILĖS DVIKANALĖJE SISTEMOJE TYRIMAS REGENERACINIU METODU

Prof. A. Aksomaitis, I. Svilainytė

Kauno technologijos universitetas

Regeneracinio metodo panaudojimas eilių teorijoje yra pakankamai efektyvus [4]. Jis leidžia priklausomų dydžių charakteristikų skaičiavimą perkelti į nepriklausomų dydžių schemą.

Tiriamas maksimalios eilės dvikanalėje sistemoje ilgis. Regeneraciniu momentu imamas momentas, kai bent vienas kanalas yra laisvas.

Tiriami trys variantai:

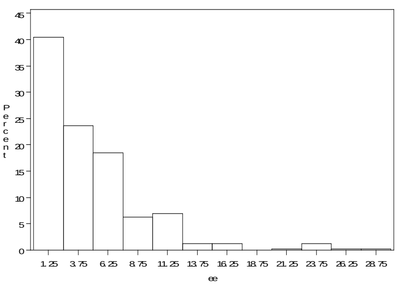
1. Atėjimo srautas pasiskirstęs pagal eksponentinį skirstinį su parametruλ = 0,5, o aptarnavimo intensyvumas pirmame kanale pasiskirstęs pagal eksponentinį skirstinį su parametruλ = 0,1, antrame kanale - λ = 0,2.
2. Atėjimo srautas pasiskirstęs pagal tolygųjį skirstinį su parametrais a = 10, b = 80, o aptarnavimo intensyvumas pirmame kanale pasiskirstęs pagal tolygųjį skirstinį su parametrais a = 70, b = 100, antrame kanale - a = 35, b = 75.
3. Atėjimo srautas pasiskirstęs pagal eksponentinį skirstinį su parametruλ = 0,02, o aptarnavimo intensyvumas pirmame kanale pasiskirstęs pagal tolygųjį skirstinį su parametrais a = 70, b = 100, antrame kanale - a = 35, b = 75.

Rezultatai:

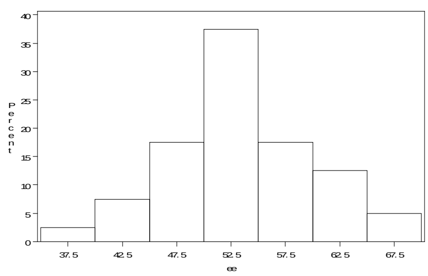
1. Atėjimo srautas ~ E(0,5), pirmos aptarnavimo sistemos intensyvumas ~ E(0,1), antros aptarnavimo sistemos intensyvumas ~ E(0,2).

Gautas išėjimo srautas ,kurio imties histograma pavaizduota 1 pav..
Iš histogramos galima spręsti, kad tai taip pat eksponentinis skirstinys. Ištyrę nustatėme, kad gautos imties vidurkis ME = 4,29 ($\lambda \approx$ 0.23), o dispersija DE = 21.16.

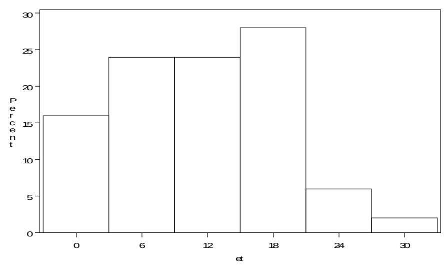
Buvo tiriamas periodų skaičiaus skirstinys. Gautos periodų skaičiaus imties vidurkis MP = 52.93, DP = 42.77. Iš gautos periodų skaičiaus imties nubraižėme histogramą (2. pav.). Iš histogramos sprendžiame, kad periodų skaičius pasiskirstęs pagal Binominį skirstinį su parametrais n = 276, p = 0.19. Braižome gautų maksimumų histogramą(3. pav).



1. pav. Išėjimo srauto intensyvumo histograma



2. pav. Atsitiktinio periodų skaičiaus histograma

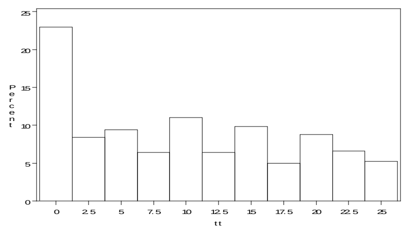


3. pav. Maksimumų histograma

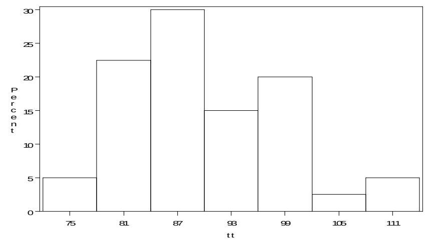
2. Atėjimo srautas ~ T(10, 80), pirmos aptarnavimo sistemos intensyvumas ~ T(70, 100), antros aptarnavimo sistemos intensyvumas ~ T(35, 75).

Gautas išėjimo srautas ,kurio imties histograma pavaizduota 4 pav..
Iš histogramos galima spręsti, kad tai taip pat eksponentinis skirstinys. Ištyrę nustatėme, kad gautos imties vidurkis ME = 10,68 , o dispersija DE = 65,77.

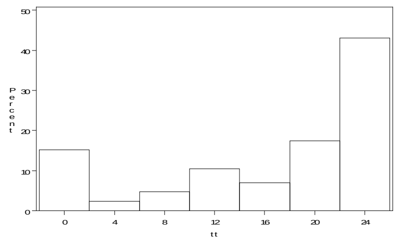
Buvo tiriamas periodų skaičiaus skirstinys. Gautos periodų skaičiaus imties vidurkis MP = 90.05, DP = 79.03. Iš gautos periodų skaičiaus imties nubraižėme histogramą(5. pav.). Iš histogramos sprendžiame, kad periodų skaičius pasiskirstęs pagal Binominį skirstinį su parametrais n = 736, p = 0.12. Braižome gautų maximumų histogramą (6. pav.).



4. pav. Išėjimo srauto intensyvumo histograma



5. pav. Atsitiktinio periodų skaičiaus histograma



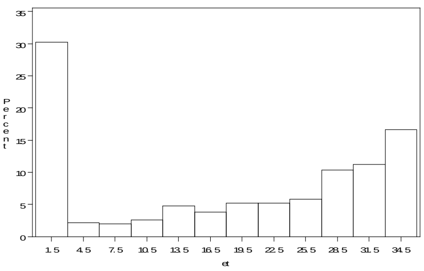
6. pav. Maksimumų histograma

3. Atėjimo srautas ~ E(0,02), pirmos aptarnavimo sistemos intensyvumas ~ T(70, 100), antros aptarnavimo sistemos intensyvumas ~ T(35, 75).

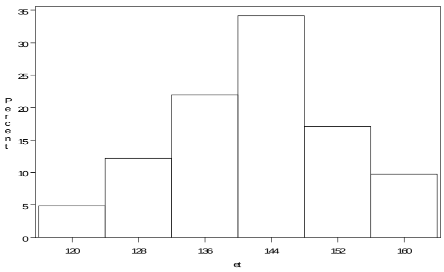
Gautas išėjimo srautas ,kurio imties histograma pavaizduota 7 pav..

Iš histogramos galima spręsti, kad tai taip pat eksponentinis skirstinys. Ištyrę nustatėme, kad gautos imties vidurkis ME = 10,68 , o dispersija DE = 65,77

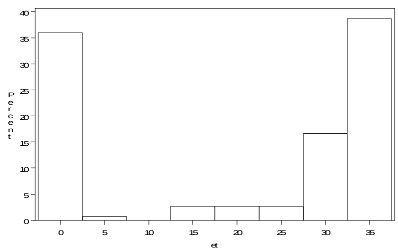
Buvo tiriamas periodų skaičiaus skirstinys. Gautos periodų skaičiaus imties vidurkis MP = 141.2, DP = 100.2. Iš gautos periodų skaičiaus imties nubraižėme histogramą(8. pav.). Iš histogramos sprendžiame, kad periodų skaičius pasiskirstęs pagal Binominį skirstinį su parametrais n = 486, p = 0.29. Braižome gautų maximumų histogramą (9. pav.).



7. pav. Išėjimo srauto intensyvumo histograma



8. pav. Atsitiktinio periodų skaičiaus histograma



9. pav. Maksimumų histograma

Literatūros sąrašas:

7. A. Aksomaitis "Tikimybių teorija ir statistika". Kaunas: Technologija, 2000;

8. Галамбош Я. Асимптотическая теория экстремальных порядковых статистик. М.: Наука. 1984;

9. Kruopis J. Matematinė statistika. – Vilnius: Mokslo ir enciklopedijų leidykla, 1993.

10. Крейн М. Лемуан О. Введение в регенеративный метод анализа моделей. – Москва: Наука, 1982. – 104 с.;

11. Mathcad 2001 Professional programinė įranga;

12. SAS statistinis paketas.