KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

MATEMATINĖS SISTEMOTYROS KATEDRA

Mindaugas Šnipas

STOCHASTINIŲ SISTEMŲ APROKSIMAVIMAS MARKOVO MODELIAIS

Magistro darbas

Vadovas
doc. dr. E. Valakevičius

KAUNAS, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

MATEMATINĖS SISTEMOTYROS KATEDRA

TVIRTINU
Katedros vedėjas

prof. habil.dr. V.Pekarskas
2008 06 06

STOCHASTINIŲ SISTEMŲ APROKSIMAVIMAS MARKOVO MODELIAIS

Taikomosios matematikos magistro baigiamasis darbas

Vadovas
doc. dr. E. Valakevičius
2008 06 03

Recenzentas
doc.dr. D.Makackas
2008 06 01

Atliko
FMMM-6 gr. stud.
M. Šnipas
2008 05 25

KAUNAS, 2008

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, habil. dr., valdybos pirmininko pavaduotojas (DnB NORD Bankas)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, dr., vice-prezidentas projektams (UAB „Baltic Amadeus")

Šnipas M. Approximation of stochastic systems by Markovian models: Master's work in applied mathematics / supervisor dr. assoc. prof. E. Valakevičius; Department of mathematical research in systems, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2008. – 97 p.

SUMMARY

Application of numerical methods with approximation allows to extend a class of systems represented by Markovian processes under investigation compared with analytical methods. In this paper we used approximation of positive distribution functions, using phase-type distributions: mixtures of Erlang distributions and Coxian distribution – both 2 and 3 moments-matching algorithms was used.

Analysis of M/G/1 and G/M/1 queueing systems showed, that moment-based queueing approximation gives high accuracy. In purpose to compute characteristics of M/G/1 and G/M/1 systems described in an event-based language, algorithms and software was created. Comparison to simulation results shows, that event-based language enables to get more precise results.

Analysis of G/G/1 systems showed, that moment-based approximation can be used to analyse difficult queueing systems.

SANTRAUKA

Dažnai realių stochastinių sistemų negalime aprašyti Markovo procesais, nes operacijų trukmės nėra pasiskirstę pagal eksponentinį dėsnį. Šiame darbe nagrinėjome sistemų aproksimavimo galimybes, taikant eksponentinių skirstinių mišinius ir sąsūkas.

Skirstinių aproksimavimui taikėme Erlango mišinius ir Kokso skirstinį. Skirstinių aproksimavimą pritaikėme aptarnavimo sistemų M/G/1 ir G/M/1 tyrimui. Atlikti teoriniai skaičiavimai parodė, kad gaunamas aukštas aproksimavimo tikslumas.

Aptarnavimo sistemų modeliavimui naudojome skaitmeninio Markovo procesų modeliavimo sistemą naudojant įvykių kalbą. Darbe sukurti metodai leidžia tiksliai apskaičiuoti sistemų charakteristikas, naudojant aproksimavimą eksponentiniais mišiniais ir sąsūkomis. Sukurta programinė įranga leidžia automatizuoti sistemų M/G/1 ir G/M/1 modeliavimą, naudojant aproksimavimą eksponentiniais mišiniais.

Sistemos G/G/1 (neištiriamos analiziniais metodais) aproksimavimo rezultatai leidžia tikėtis, kad šiame darbe nagrinėjamas metodas gali būti naudojamas ir sudėtingų sistemų modeliavime.

Darbe gauti rezultatai pristatyti dvejose mokslinėse konferencijose.

TURINYS

ĮVADAS....9
1. TEORINĖ DALIS....11
1.1 Skirstinių aproksimavimas eksponentiniais mišiniais....11
1.1.1 Erlango mišiniai....11
1.1.1.1 Pirmojo tipo Erlango mišinys....11
1.1.1.2 Antrojo tipo Erlango mišinys....12
1.1.2 Kokso skirstinys....13
1.1.2.1 Kokso skirstinio parinkimas sulyginant 2 pradinius momentus....13
1.1.2.2 Kokso skirstinio parinkimas sulyginant 3 pradinius momentus....14
1.2 Aptarnavimo sistemų teorijos elementai....15
1.2.1 Bendroji teorija....15
1.2.1.1 Laplaso-Stiltjeso transformacija....15
1.2.1.2 Kendalo klasifikacija....16
1.2.1.3 Sistemos užimtumo koeficientas....17
1.2.1.4 Pagrindinės aptarnavimo sistemų charakteristikos....17
1.2.1.5 Little‘o formulė....18
1.2.1.6 PASTA savybė....18
1.2.2 Aptarnavimo sistemų modeliai....19
1.2.2.1 M/G/1 sistema su eile....19
1.2.2.2 G/M/1 sistema su eile....20
2. TIRIAMOJI DALIS....22
2.1 Analizinis aptarnavimo sistemų charakteristikų apskaičiavimas....22
2.1.1 M/G/1 sistemos tyrimas....22
2.1.1.1 Lognormalusis skirstinys....23
2.1.1.2 Tolygusis skirstinys....26
2.1.1.3 Veibulo skirstinys....28
2.1.1.4 Gama skirstinys....29
2.1.2 G/M/1 sistemos tyrimas....30
2.1.2.1 Lognormalusis skirstinys....31
2.1.2.2 Tolygusis skirstinys....34
2.1.2.3 Veibulo skirstinys....35
2.1.2.4 Gama skirstinys....36
2.2 Aptarnavimo sistemų skaitmeninis modeliavimas....37
2.2.1 M/G/1 sistemų skaitmeninis modeliavimas....38
2.2.1.1 M/G/1 sistemos modeliavimas, kai G – Erlango mišinys....38
2.2.1.2 M/G/1 sistemos modeliavimas, kai G – Kokso skirstinys....40
2.2.2 G/M/1 sistemų modeliavimas....42
2.2.2.1 G/M/1 sistemos modeliavimas, kai G – Erlango mišinys....42
2.2.2.2 G/M/1 sistemos modeliavimas, kai G – Kokso skirstinys....44
2.3 Aptarnavimo sistemų imitacinis modeliavimas....46
2.3.1 Imitacinio modeliavimo programa ARENA....46
2.3.2 Imitacinio ir skaitmeninio modeliavimo rezultatų palyginimas....48
2.3.2.1 M/G/1 sistemų skaitmeninio ir imitacinio modeliavimo palyginimas....48
2.3.2.2 G/M/1 sistemų skaitmeninio ir imitacinio modeliavimo palyginimas....49
2.4 Sistemos G/G/1 tyrimas....51
3. PROGRAMINĖ REALIZACIJA....54
IŠVADOS....57
PADĖKOS....58
LITERATŪROS SĄRAŠAS....59
PRIEDAI....60

LENTELIŲ SĄRAŠAS

2.1 lentelė. Sistemos M/G/1 aproksimavimas, kai G – lognormalusis skirstinys	26
2.2 lentelė. Sistemos M/G/1 aproksimavimas, kai G – tolygusis skirstinys	28
2.3 lentelė. Sistemos M/G/1 aproksimavimas, kai G – Veibulo skirstinys	29
2.4 lentelė. Sistemos M/G/1 aproksimavimas, kai G – gama skirstinys	30
2.5 lentelė. Sistemos G/M/1 aproksimavimas, kai G – lognormalusis skirstinys	33
2.6 lentelė. Sistemos G/M/1 aproksimavimas, kai G – tolygusis skirstinys	35
2.7 lentelė. Sistemos G/M/1 aproksimavimas, kai G – Veibulo skirstinys	36
2.8 lentelė. Sistemos G/M/1 aproksimavimas, kai G – gama skirstinys	37
2.9 lentelė. Sistemų M/G/1 skaitmeninio ir imitacinio modeliavimo palyginimas	48
2.10 lentelė. Sistemų G/M/1 skaitmeninio ir imitacinio modeliavimo palyginimas	50
2.11 lentelė. Sistemos G/G/1 skaitmeninio ir imitacinio modeliavimo palyginimas	53

PAVEIKSLŲ SĄRAŠAS

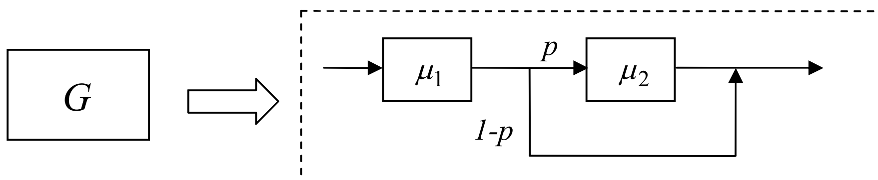
1.1 pav. Pirmojo tipo Erlango mišinio fazinė schema	11
1.2 pav. Antrojo tipo Erlango mišinio fazinė schema	12
1.3 pav. Kokso skirstinio fazinė schema	13
2.1 pav. Grafinis lygties sprendimo patikrinimas	34
2.2 pav. Sistemos M/G/1 aproksimavimo schema (G – Erlango mišinys)	38
2.3 pav. Sistemos M/G/1 aproksimavimo schema (G – Kokso skirstinys)	40
2.4 pav. Sistemos G/M/1 aproksimavimo schema (G – Erlango mišinys)	42
2.5 pav. Sistemos G/M/1 aproksimavimo schema (G – Kokso skirstinys)	44
2.6 pav. Aptarnavimo sistemos imitacinio modelio schema	46
2.7 pav. Programos ARENA paraiškų srauto komponento “Create” aprašymo langas	46
2.8 pav. Programos ARENA komponento “Process” aprašymo langas	47
2.9 pav. Programos ARENA komponento “Decide” aprašymo langas	47
2.10 pav. Aptarnavimo sistemos imitacinio modelio schema su būsenos tikimybių skaičiavimu	48
2.11 Sistemos G/G/1 aproksimavimo schema	51
3.1 Programos langas	54
3.2 pav. Sistemos aprašymas *Switch-Case* sakiniu	56

ĮVADAS

Markovo procesai yra vieni nuodugniausiai ištyrinėtų ir plačiausiai praktikoje taikomų atsitiktinių procesų. Jų taikymo sritys – masinio aptarnavimo teorija, ekonomika, branduolinė sauga, medicina, biologijos mokslai ir daugybė kitų. Markovo procesus patogu taikyti įvairių sistemų modeliavimui ne tik dėl jų natūralaus panašumo į realias sistemas (kai sistema aprašoma tam tikra būsenų aibe, su perėjimais tarp jų), bet ir dėl nuodugniai išplėtotos matematinės teorijos.

Tolydaus laiko Markovo procesai glaudžiai susiję su eksponentiniu skirstiniu – t.y., perėjimo trukmė iš vienos būsenos į kitą pasiskirsčiusi pagal eksponentinį dėsnį. Realiose sistemose ši sąlyga dažnai nebūna išpildyta, ir tai apsunkina jų tyrimą ir modeliavimą.

Šiame darbe buvo tiriamas stochastinių sistemų aproksimavimas Markovo procesais. Aproksimavimui naudojome fiktyvių fazių metodą – t.y., būsena, kurios skirstinys neeksponentinis (žymim G), pakeičiama eksponentinių fazių mišiniu.



pav. Sistemos aproksimavimo Markovo modeliu schema

Pirmoji problema – parinkti tinkamą fazių mišinį ir jo parametrus. Šiame darbe naudojome Erlango mišinius (galimi atskiri atvejai – hipereksponentinis skirstinys, Erlango skirstinys) ir Kokso skirstinį. Aproksimuojamų skirstinių parametrams apskaičiuoti naudojome momentų metodą – t.y., juos parinkome taip, kad sutaptų tiriamo skirstinio G ir aproksimuojamo mišinio pirmi pradiniai (o tuo pačiu ir centriniai) momentai. Naudojome dviejų ir trijų momentų metodus (praktiniuose skaičiavimuose trijų ar netgi dvejų momentų sulyginimas duoda pakankamai gerus rezultatus. Keturių ar netgi daugiau momentų skaičiavimas leistų išplisti aproksimuojamų skirstinių klasę, tačiau apsunkintų parametrų apskaičiavimą ir netgi susiaurintų tiriamų skirstinių G klasę). Pvz., norint apskaičiuoti antros eilės Kokso skirstinio (žr. pav. aukščiau) parametrus 3 momentų metodu, reikia išspręsti lygčių sistemą:

$$\begin{cases} \dfrac{1}{\mu_1} + \dfrac{p}{\mu_2} = m_1 \\ \dfrac{1}{\mu_1^2} + \dfrac{p}{\mu_1\mu_2} + \dfrac{p}{\mu_2^2} = \dfrac{m_2}{2} \\ \dfrac{1}{\mu_1^3} + \dfrac{p}{\mu_1^2\mu_2} + \dfrac{p}{\mu_1\mu_2^2} + \dfrac{p}{\mu_2^3} = \dfrac{m_3}{6} \end{cases}$$

čia: $m_1, m_2,\ m_3$ - tiriamo mišinio pradiniai momentai; p, μ_1, μ_2 – Kokso skirstinio parametrai.

Skirstinių aproksimavimo sulyginant 3 pradinius momentus naudojant Erlango mišinius nuodugniai išnagrinėtas [1] straipsnyje. Kokso skirstinio parametrų nustatymas pateikiamas [2] straipsnyje, o skirstinio galimo aproksimavimo 3 momentais sąlygos gerai ištirtos [3].

Didžiausią dėmesį skyrėme aptarnavimo sistemų aproksimavimui. Aptarnavimo sistemoms M/G/1 ar G/M/1 yra sukurta teorija, leidžianti gauti analizinius rezultatus (pvz., [4]) , o tai leidžia patikrinti praktines mūsų tiriamo metodo galimybes. Praktiniam aptarnavimo sistemų modeliavimui naudojome prof. H. Pranevičiaus 1996 pateiktą kompiuterinę sistemą, skirtą modeliuoti Markovo procesams (plačiau apie minėtą sistemą galite rasti [5]). Minėtoji sistema labai patogi aprašant ir apskaičiuojant įvairaus sudėtingumo aptarnavimo sistemų charakteristikas. Šiame darbe sukurti metodai leidžia ją taikyti ir naudojant aproksimavimą.

Tiriant sudėtingas sistemas, ar netgi kai kurias paprastesnių sistemų charakteristikas, analitiniai metodai neveikia, todėl vienetelis patikimesnis tyrimo metodas yra imitacinis modeliavimas. Šiame darbe naudojome imitacinio modeliavimo programą ARENA 9.0, kad palygintume gautus skaitmeninio modeliavimo rezultatus. Gauti rezultatai rodo, kad mūsų nagrinėjamas metodas gali būti papildomas įrankis analizuojant aptarnavimo sistemas ar kitas stochastines sistemas.

1. TEORINĖ DALIS

1.1 Skirstinių aproksimavimas eksponentiniais mišiniais

1.1.1 Erlango mišiniai

1.1.1.1 Pirmojo tipo Erlango mišinys

Nagrinėsime dviejų tipų Erlango mišinius. Pirmojo tipo Erlango mišinys naudojamas lyginant du pradinius momentus. Jo paskirstymo funkcija:

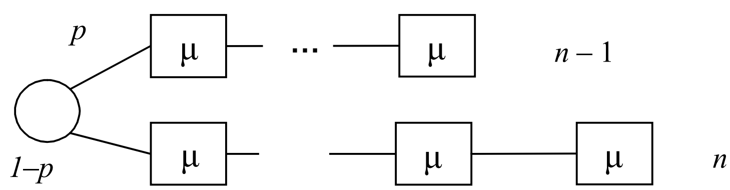
$$f(t) = p\mu \frac{(\mu t)^{n-2}}{(n-2)!} e^{-\mu t} + (1-p)\mu \frac{(\mu t)^{n-1}}{(n-1)!} e^{-\mu t} \quad , \ t > 0.$$

čia: $0 \le p \le 1$; $\mu > 0$; $n \in N$.

Laplaso-Stiltjeso transformacija:

$$X(s) = p\left(\frac{\mu}{\mu + s}\right)^{n-1} + (1-p)\left(\frac{\mu}{\mu + s}\right)^{n}$$

Skirstinio fazinė schema:



1.1 pav. Pirmojo tipo Erlango mišinio fazinė schema

Atskiri atvejai: a) kai n = 1, gaunamas ekponentinis skirstinys ; b) kai p = 0 arba p = 1 gaunamas paprastas Erlango skirstinys (atskiru atveju, tiesiog eksponentinis).

Šis Erlango mišinys gali būti naudojams, kai variacijos koeficientas c tenkina sąlygą $0 \le c \le 1$. Konkrečiau, parametras n parenkamas taip, kad $\frac{1}{n} \le c^2 \le \frac{1}{n-1}$. Parinkus n, parametrai p ir μ randami iš formulių:

$$p = \frac{1}{1 + c^2}\left(nc^2 - \sqrt{n(1 + c^2) - n^2 c^2}\right) \qquad \mu = \frac{n - p}{E(X)}$$

Taip parinkus mišinio parametrus, jo pirmi du momentai sutaps su aproksimuojamo skirtinio G(x) momentais.

1.1.1.2 Antrojo tipo Erlango mišinys

Nagrinėsime antros eilės Erlango mišinį. Jo pasiskirstymo funkcija:

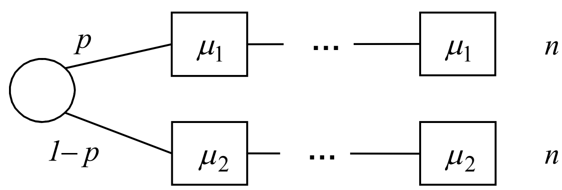
$$f(t) = p\mu_1 \frac{(\mu_1 t)^{n-1}}{(n-1)!} \exp\{-\mu_1 t\} + (1-p)\mu_2 \frac{(\mu_2 t)^{n-1}}{(n-1)!} \exp\{-\mu_2 t\},\ \mathrm{t} > 0.$$

čia: $0 \le p \le 1$; $\mu_1 > 0$; $\mu_2 > 0$; $n \in N$.

Laplaso-Stiltjeso transformacija:

$$X(s) = p\left(\frac{\mu_1}{\mu_1 + s}\right)^n + (1-p)\left(\frac{\mu_2}{\mu_2 + s}\right)^n$$

Antrojo tipo Erlango mišinio fazinė schema:



1.2 pav. Antrojo tipo Erlango mišinio fazinė schema

Šis Erlango mišinys turi keturis parametrus, todėl juo galima aproksimuoti skirstinius sulyginant tris pradinius momentus. Straipsnyje [1] įrodoma, kad tokio tipo 2-os eilės Erlango mišiniu galima aproksimuoti bet kokį teigiamą skirstinį taip, kad sutaptų trys pirmieji momentai (jei aproksimuojamo skirstinio trys pirmieji momentai egzistuoja). Pateikiame antrojo tipo Erlango mišinio parametrų radimo algoritmą:

1) Parametras n yra mažiausias natūrinis skaičius, tenkinantis nelygybes:

$$n > \frac{1}{c^2}$$

$$n > \frac{-\gamma + \frac{1}{c^2} + \frac{1}{c} + 2c}{\gamma - c + \frac{1}{c}}.$$

2.) Suradus n, likusieji parametrai randami iš formulių:

a) $$x = \mu_1 \mu_3 - \frac{n+2}{n+1}\mu_2^2$$

$$y = \mu_2 - \frac{n+1}{n}\mu_1^2$$

b) $A = n(n+2)\mu_1 y$

$$B = -\left(nx + \frac{n(n+2)}{n+1} y^2 + (n+2)\mu_1^2 y \right)$$

$$C = \mu_1 x$$

c) $\lambda_1 = \dfrac{2A}{-B + \sqrt{B^2 - 4AC}}$

$$\lambda_2 = \frac{2A}{-B + \sqrt{B^2 - 4AC}}$$

$$p = \left(\frac{\mu_1}{n} - \frac{1}{\lambda_2} \right) \left(\frac{1}{\lambda_1} - \frac{1}{\lambda_2} \right)$$

Galimi atskiri Erlango mišinio atvejai: a) kai p = 0 arba p = 1, gaunamas paprastas Erlango skirstinys ; b) kai n = 1 gaunamas eksponentinių skirstinių mišinys – hipereksponentinis skirstinys.

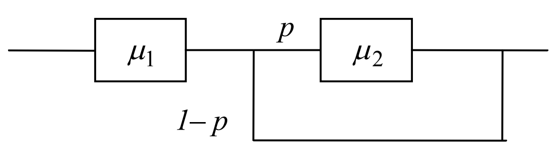
1.1.2 Kokso skirstinys

1.1.2.1 Kokso skirstinio parinkimas sulyginant 2 pradinius momentus

Nagrinėsime antros eilės Kokso skirstinį. Jo pasiskirstymo funkcija:

$$f(x) = \mu_1 e^{-\mu_1 x} + \frac{p_1 \mu_1}{p_1 \mu_2 - \mu_1} \left(\mu_1 e^{-\mu_1 x} - p_2 \mu_2 e^{-p_2 \mu_2 x} \right)$$

Skirstinio fazinė schema:



1.3 pav. Kokso skirstinio fazinė schema

Laplaso-Stiltjeso transformacija:

$$X(s) = p\left(\frac{\mu_1}{\mu_1 + s} \right)\left(\frac{\mu_2}{\mu_2 + s} \right) + (1-p)\left(\frac{\mu_1}{\mu_1 + s} \right)$$

Nesunku įsitikint, kad antros eilės Kokso skirstinio variacijos koeficientas tenkina sąlygą $c^2 \geq 0.5$. Taigi, norint aproksimuoti konkretų skirstinį, būtina patikrinti šią sąlygą. Reikia pastebėti, kad dalis praktikoje paplitusių skirstinių šios sąlygos netenkina.

Kokso skirstinys priklauso nuo trijų parametrų, todėl lyginant 2 pradinius momentus reikia įvesti papildomas sąlygas. Pvz., praktikoje dažnai naudojama papildoma sąlyga $\frac{p}{\mu_1} = \frac{1-p}{\mu_2}$ (taip vadinamoji subalansuotų vidurkių sąlyga). Tokiu atveju, aproksimuojant G(x) Kokso skirstiniu, parametrai randami iš lygybių:

$$\mu_1 = \frac{2}{E(X)} \qquad p = \frac{0.5}{c^2} \qquad \mu_2 = p\mu_1 .$$

1.1.2.2 Kokso skirstinio parinkimas sulyginant 3 pradinius momentus

Skirtingai nuo Erlango mišinio, Kokso skirstinio parinkimas sulyginant 3 pirmuosius momentus ne visada įmanomas. Straipsnyje [3] apibrėžtos sąlygos, kada skirstinys G(x) gali būtųi tinkamai aproksimuojamas Kokso skirstiniu – t.y., kad sutaptų trys pirmieji momentai. Šios sąlygos išreikštos naudojant taip vadinamus normalizuotus momentus m_2 ir m_3, kurie apibrėžiami taip:

$$m_2 = \frac{E\left(X^2\right)}{\left(E(X)\right)^2} \quad m_3 = \frac{E\left(X^3\right)}{E\left(X\right)E(X^2)} .$$

Nesunku nustatyti sąryšį tarp normalizuotų momentų m_2 ir m_3 bei variacijos koeficiento c ir asimetrijos koeficiento γ:

$$m_2 = c^2 + 1 \quad \text{ir} \quad m_1 = \gamma\sqrt{c^2 + 1} .$$

Teorema: Skirstinio funkciją G galima aproksimuoti antros eilės Kokso skirstiniu taip, kad sutaptų trys pirmieji momentai, jei G normalizuoti momentai tenkina sąlygas:

$$\left\{\frac{4}{3} m_2 \leq m_3 \leq \frac{6\left(m_2 - 1\right)}{m_2} \cap \frac{3}{2} \leq m_2 \leq 2\right\} \cup \left\{\frac{4}{3} m_2 \leq m_3 \cap 2 < m_2\right\}.$$

Pateiksime parametrų apskaičiavimo algoritmą (žr. [2]), aproksimuojant skirstinį G antros eilės Kokso skirstiniu:

a) Pažymim $f_i = \frac{g_i}{i!}$. (čia g_i - skirstinio G i-tasis pradinis momentas).

b) $D = \left(f_1 f_2 - f_3\right)^2 - 4\left(f_2^2 - f_1 f_3\right)\left(f_1^2 - f_2\right)$

c) Skirstinio parametrai:

$$\mu_2 = \frac{f_1 f_2 - f_3 \pm \sqrt{D}}{2\left(f_2^2 - f_1 f_3\right)}$$

$$\mu_1 = \frac{\mu_2 f_1 - 1}{\mu_2 f_2 - f_1}$$

$$p = \frac{\mu_2 \left(f_1 \mu_1 - 1\right)}{\mu_1}.$$

1.2 Aptarnavimo sistemų teorijos elementai

1.2.1 Bendroji teorija

1.2.1.1 Laplaso-Stiltjeso transformacija

Neneigiamo atsitiktinio dydžio X Laplaso-Stiltjeso transformacija vadinama funkcija X(s), apibrėžiama taip:

$$\widetilde{X}(s) = E\left(e^{-sX}\right) = \int_0^\infty e^{-sX}\, dF(x), \quad s \ge 0.$$

Pvz., dažnai naudojama eksponentinio skirstinio Laplaso-Stiltjeso transformacija $\widetilde{X}(s) = \frac{\mu}{\mu + s}$.

Laplaso-Stiltjeso transformacijos savybės:

a) $\left|\widetilde{X}(s)\right| \le 1$, kai $s \ge 0$. Be to $\widetilde{X}(0) = 1$.

b) $\widetilde{X}^{(k)}(0) = (-1)^k E\left(X^k\right)$

c) Jei X ir Y yra nepriklausomi neneigiami atsitiktiniai dydžiai jų sumos Z = X + Y Laplaso-Stiltjeso transformacija bus:

$$\widetilde{Z}(s) = \widetilde{X}(s) \cdot \widetilde{Y}(s)$$

d) Jeigu atsitiktinis dydis su tikimybe p įgyja reikšmę X, o su tikimybe q – reikšmę Y, tai jo Laplaso-Stiltjeso transformacija:

$$\widetilde{Z}(s) = p \cdot \widetilde{X}(s) + q \cdot \widetilde{Y}(s).$$

1.2.1.2 Kendalo klasifikacija

Pagrindiniai aptarnavimo sistemas apibūdinantys komponentai:

a) **Atvykstančių paraiškų srautas.** Paprastai laikoma, kad visų paraiškų atvykimo laikai yra nepriklausomi ir vienodai pasiskirstę. Praktikoje dažniausiai naudojamas ir geriausiai ištirtas Puasoninis srautas. Puasoninio srauto laiko tarpai tarp paraiškų atvykimo pasiskirstę pagal eksponentinį dėsnį.

b) **Aptarnavimo laikas.** Paprastai laikoma, kad kiekvienos paraiškos aptarnavimo laikas yra nepriklausomi vienodai pasiskirstę atsitiktiniai dydžiai.

c) **Aptarnavimo pajėgumai.** T.y., aptarnavimo įrenginių skaičius.

d) **Eilės dydis.** Gali būti baigtinis natūralusis skaičius arba begalybė – t.y., eilės ilgis neribotas. (Nors teoriniuose modeliuose daugelis formulių išvesta begalinėms eilėms, praktiniame sistemų modeliavime įvedami apribojimai.)

e) **Aptarnavimo tvarka.** Paraiškos gali būti aptarnaujamos įvairia tvarka, pvz. FIFO (first in first out) – pirmiausia aptarnaujama seniausiai atėjusi paraiška. Taip pat gali egzistuoti prioritetai (vieno tipo paraiškos aptarnaujimos pirmiau negu kito tipo), arba įvairiai nustatoma tvarka esant daugiau nei vienam aptarnavimo įrenginiui.

Anglų statistiko M. Kendalo įvesta klasifikacija leidžia sutrumpintai aprašyti aptarnavimo sistemą. Pagal šią notaciją, sistema apibūdinama simbolių trejetu a/b/c. Pirma raidė žymi ateinančio srauto skirstinį, antra – aptarnavimo laiko skirstinį. Pvz., raidė M žymi Puasoninį srautą, t.y. eksponentinį skirstinį (M – nuo angliško žodžio *memoryless* – be atminties, pagal unikalią eksponentinio skirstinio savybę,). G žymi bet kokį skirstinį (angl. *general* – bendrasis), D – determinuotą (t.y., be jokio atsitiktinumo, apibrėžtas).Trečia raidė nurodo aptarnavimo įrenginių skaičių. Taigi, M/G/1 žymi aptarnavimo sistemą, kurios paraiškų atėjimo srautas Puasoninis, aptarnavimo laikas pasiskirstęs pagal skirstinį G, o sistema turi vieną aptarnavimo įrenginį. Toks žymėjimas gali būti išplėstas pridedant daugiau raidžių. Pvz., jei sistemos M/G/1 maksimalus leidžiamas eilės ilgis N, tai rašome M/G/1/N.

Paprastai laikoma, kad paraiškos ateina po vieną, eilės ilgis neribojamas, o paraiškos aptarnaujamos atėjimo tvarka. Jei specialiai nenurodysime kitaip, toliau laikysimės tokių pačių prielaidų.

1.2.1.3 Sistemos užimtumo koeficientas

Tarkim, aptarnavimo sistemos G/G/1 paraiškų srauto dažnis yra λ, o vidutinė paraiškos aptarnavimo trukmė E(B). Be to, aptarnavimo įrenginys vienu metu gali aptarnauti tik vieną paraišką. Tokiu atveju, kad eilės ilgis nenutoltų į begalybę, reikia pareikalauti, kad būtų λE(B)<1. Paprastai tariant, paraiškų aptarnavimo trukmė turi būti trumpesnė nei atvykimo trukmė. Plačiau nesigilinant, galima sakyti kad sistemos G/G/1, kurios λE(B)=1 eilės ilgis taip pat neaprėžtai didėja, išskyrus sistemas D/D/1. Pažymėkime:

$$\rho = \lambda E(B).$$

Jeigu ρ < 1, dydis ρ vadinamas užimtumo koeficientu, nes jis parodo kokią laiko dalį aptarnavimo įrenginys vidutiniškai yra užimtas.

Daugiakanalėje aptarnavimo sistemoje G/G/c (kurioje yra c lygiagrečiai veikiančių aptarnavimo įrenginių), turime reikalauti, kad λE(B) < c. Tuomet užimtumo koeficientas kiekvienam įrenginiui yra ρ=λE(B)/c.

1.2.1.4 Pagrindinės aptarnavimo sistemų charakteristikos

Tirant aptarnavimo sistemas svarbu įvertinti tokias charakteristikas:

a) Vidutinis paraiškos laukimo ir buvimo sistemoje laikas. Buvimo sistemoje laikas, tai laukimo laikas plius aptarnavimo laikas.
b) Sistemoje esančių paraiškų skaičius (t.y., esančių eilėje ir aptarnaujamų). Idealiu atveju nustatomas sistemoje esančių paraiškų skirstinys. Sudėtingesnėms sistemoms pavyksta įvertinti tik vidurkį ar keletą pradinių momentų.

Nagrinėkime sistemą G/G/c. Tarkim, atsitiktinis dydis L(t) parodo paraiškų skaičių sistemoje momentu t, o dydis S(n) parodo n-tosios paraiškos buvimo sistemoje laiką. Jeigu sistemos užimtumo koeficientas ρ mažesnis už vienetą, galima įrodyti, kad šie atsitiktiniai dydžiai turi ribinį skirstinį, kai t → ∞, ir kai n → ∞. Be to, jų skirstiniai nepriklauso nuo pradinės sistemos būsenos.
Tegu atsitiktiniai dydžiai L ir S turi ribinius L(t) ir S(n) skirstinius. Tada:

$$p_k = P(L = k) = \lim_{t \to \infty} P(L(t) = k)$$

$$F_S(x) = P(S \le x) = \lim_{n \to \infty} P(S(n) \le x).$$

Per ilgą laiko tarpą nusistovėjusios vidutinės reikšmės gali būti išreiškiamos taip:

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t L(x)dx = E(L)$$

$$\lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}S_k = E(S).$$

Taigi, E(L) – vidutinis sistemoje esančių paraiškų skaičius, E(S) – vidutinis paraiškos buvimo sistemoje laikas.

1.2.1.5 Little‘o formulė

Little’o formulė nusako labai svarbų sąryšį tarp svarbiausių aptarnavimo sistemos charakteristikų: E(L) – vidutinis sistemoje esančių paraiškų skaičius, E(S) – vidutinis paraiškos buvimo sistemoje laikas ir λ – vidutinis paraiškų skaičius ateinantis per laiko vienetą, kitaip saknat – paraiškųsrauto intensyvumas. Šiuos dydžius siejanti Little’o formulė:

$$E(L) = \lambda E(S). \qquad \textbf{(1.1)}$$

Be to, laikomasi prielaidos kad aptarnavimo sistema sugeba aptarnauti visas paraiškas.

Panašų sąryšį galima nustatyti tarp eilėje esančių paraiškų skaičiaus L^q ir vidutinio laukimo laiko E(W):

$$E(L^q) = \lambda E(W). \qquad \textbf{(1.2)}$$

Anksčiau aprašytas sistemos užimtumo koeficientas gali būti interpretuojamas kaip Little’o formulė taikoma tik aptarnavimo įrenginiui.

1.2.1.6 PASTA savybė

Pirmiausia paminėsime labai svarbią sistemų su Puasoniniu srautu savybę. Ji skamba taip: į sistemą M/…/… atvykstanti paraiška ras vidutiniškai tokią pačią būseną kaip ir išorinis sistemos stebėtojas tam tikru laiko momentu. Kitaip sakant, tikimybė, kad į sistemą atvykstanti paraiška randa sistemą tam tikroje būsenoje A, sutampa su tikrąja sistemos būsenos A tikimybe. Ši savybė galioja tik Puasoniniam srautui. Intuityviai šią savybę galime paaiškinti Puasoninio srauto visišku atsitiktinumu (t.y., eksponentinis skirstinys neturi atminties). Ši savybė dar vadinama PASTA (angl. Poisson Arivals See Time Average) savybe.

Kad bendru atveju PASTA savybė negalioja, galima įsitikinti iš paprasto pavyzdžio. Tarkim, nagrinėkime sistemą D/D/1, kuri tuščia 0 laiko momentu, paraiškos atvyksta momentais 1,3,5…, o aptarnavimo trukmė 1.Akivaizdu, kad kieviena atvykstanti paraiška ras tuščią sistemą. Tuo tarpu išorinis stebėtojas matys, kad sistema būna užimta pusę stebimo laiko.

1.2.2 Aptarnavimo sistemų modeliai

1.2.2.1 M/G/1 sistema su eile

Nagrinėkime sistemą M/G/1, kurios paraiškų atvykimo srautas λ, o paraiškos aptarnaujamos atvykimo tvarka. Paraiškų aptarnavimo trukmė B kiekvienai paraiškai yra atsitiktiniai vienodai pasiskirstę dydžiai, kurių pasiskirstymo funkcija F(t) ir tankio funkcija f(t). Pareikalausime, kad $\rho = \lambda E(B)$ būtų mažesnis už 1. Pažymėkime L_k^d - skaičius paraiškų, esančių sistemoje kai išvyksta k-toji paraiška. Apibrėžkime ribinį skirstinį:

$$d_n = \lim_{k\to\infty} P(L_k^d = n)\,.$$

Galime sakyti, kad d_n yra tikimybė, kad iš sistemos išeinanti paraiška palieka sistemoje n paraiškų. Dar svarbesnis dydis p_n, kuris apibrėžiamas taip:

$$p_n = \lim_{t\to\infty} P(L(t) = n)$$

čia L(t) anksčiau minėtas dydis, parodantis sistemoje esančiu paraiškų skaičių laiko momentu t. Taigi, p_n galim interpretuoti kaip tikimybę, kad sistemoje yra n paraiškų. Trečias svarbus ribinis skirstinys įvertiną sistemos būseną, kurią randa atvykstanti paraiška. Apibrėžkime:

$$a_n = \lim_{t\to\infty} P(L_k^a = n)\,.$$

čia L_k^a - paraiškų skaičius sistemoje, kurį randa k-toji į sistemą atvykstanti paraiška. Tuomet a_n galime laikyti tikimybe, kad į sistemą atvykstanti paraiška ras sitemoje n paraiškų.

Iš PASTA savybės galime spręsti, kad sistemos M/G/1 $a_n = p_n$ kiekvienam n. Be to, be įrodymo pasakysime, kad galioja ir $a_n = d_n$. Reikia pažymėti, kad ši savybė galioja visoms sistemoms, kuriose paraiškos atvyksta ir išvyksta po vieną. Taip pat ir G/G/1 sistemoms. Apibendrinant, sistemoms M/G/1 kiekvienam n galioja

$$a_n = d_n = p_n\,.$$

Sistemos M/G/1 charakteristikoms apskaičiuoti patogu naudoti Polačeko-Chinčino formulę. Pažymėkime: L – paraiškų skaičius sistemoje; $P_L(z)$ - dydžio L momentus generuojanti funkcija; ρ – sistemos užimtumo koeficientas ($\rho = \lambda E(B)$); $\widetilde{B}(s)$ - aptarnavimo laiko Laplaso-Stiltjeso transformacija. Polačeko-Chinčino formulė:

$$P_L(z) = \frac{(1-\rho)\cdot \widetilde{B}(\lambda - \lambda z)\cdot(1-z)}{\widetilde{B}(\lambda - \lambda z) - z}\,. \qquad \textbf{(1.3)}$$

Taigi, Polačeko-Chinčino formulė leidžia rasti dydžio L charakteristinę funkciją. Kaip žinome iš tikimybių teorijos, diferencijuodami galime apskaičiuoti dydžio L momentus. Tiesa, norėdami rasti L skirstinį, turime apversti formulę, o tai bendru atveju gali būti labai sudėtingas uždavinys. Paprastesniais atvejais galima rasti ir analizines išraiškas, arba pasinaudoti matematinių programų paketais (pvz., MathCad komanda Series).

Polačeko-Chinčino formulė turi ir kitą formą. Pažymėkime: S – paraiškos buvimo sistemoje laikas. $\widetilde{S}(s)$ - a.d. S Laplaso-Stiltjeso transformacija. Tuomet, Polačeko-Chinčino formulė paraiškos buvimo sistemoje laikui:

$$\widetilde{S}(s) = \frac{(1-\rho)\cdot\widetilde{B}(s)\cdot s}{\lambda\widetilde{B}(s)+s-\lambda}. \qquad \textbf{(1.4)}$$

Galima išvesti ir trečią Polačeko-Chinčino formulės variantą, skirtą vidutiniam paraiškos laukimo eilėje laikui. Jeigu W – paraiškos laukimo eilėje laikas, B – paraiškos aptarnavimo trukmė, be to, W ir B yra nepriklausomi, tai bendras paraiškos buvimo sistemoje laikas S bus W ir B suma. Iš Laplaso-Stiltjeso transformacijos savybių žinome:

$$\widetilde{S}(s) = \widetilde{W}(s)\cdot\widetilde{B}(s). \qquad \textbf{(1.5)}$$

Čia $\widetilde{W}(s)$ - a.d. W Laplaso-Stiltjeso transformacija.

Taigi, iš (1.3) ir (1.4) gauname:

$$\widetilde{W}(s) = \frac{(1-\rho)\cdot s}{\lambda\widetilde{B}(s)+s-\lambda} \qquad \textbf{(1.6)}$$

Tai vadinamoji Polačeko-Chinčino formulės trečioji forma.

Suradę laukimo laiko W Laplaso-Stiltjeso transformaciją, galime rasti E(W). Iš čia, pasinaudodami Little‘o (1.2) formule galime rasti ir vidutinį sistemos eilės ilgį E(L^q):

$$E\left(L^q\right) = \lambda E(W).$$

1.2.2.2 G/M/1 sistema su eile

Analogiškai anksčiau nagrinėtai M/G/1 sistemai, laikysimės prielaidų, kad paraiškos atvyksta ir yra aptarnaujamos po vieną. Atvykimo laikai yra nepriklausomi ir vienodai pasiskirstę atsitiktiniai dydžiai, su pasiskirstymo funkcija $F_A(t)$ ir tankio funkcija $f_A(t)$. Tegu paraiškų atvykimo vidurkis 1/λ, o paraiškų aptarnavimo laikas pasiskirstęs pagal eksponentinį dėsnį su parametru μ. Dėl stabilumo reikalausime, kad galiotų sąlyga $\rho = \frac{\lambda}{\mu} < 1$.

Nagrinėdami sistemas M/G/1 apibrėžėme ribinius skirstinius a_n, d_n ir p_n. Nagrinėjant sistemas G/M/1 analizinės išraiškos apskaičiuojamos naudojant a_n (a_n yra tikimybė, kad į sistemą ateinanti paraiška sistemoje ras n kitų paraiškų).

Aptarnavimo sistemų teorijoje (pvz. [1]) įrodoma, kad G/M/1 sistemoms tikimybės a_n priklauso tik nuo tam tikro dydžio σ. Dydis σ priklauso intervalui (0 ; 1) ir yra lygties

$$\sigma = \widetilde{A}(\mu - \mu\sigma) \tag{1.7}$$

sprendinys.

Akivaizdu, kad lygtis turi sprendinį σ = 1, kadangi $\widetilde{A}(0) = 1$, tačiau šis sprendinys mums netinka. Galima įrodyti, kad galiojant sąlygai $\rho < 1$, lygtis 1) turi vienintelį sprendinį intervale (0;1), kuris ir yra mūsų ieškomasis. Tuomet, a_n turi pavidalą:

$$a_n = (1 - \sigma)\sigma^n. \quad n = 0,1,2,... \tag{1.8}$$

Taigi, galime daryti išvadą, kad nepriklausomai nuo atvykimo srauto, sistemoj G/M/1 esančių paraiškų skaičius (kurį randa atvykstanti paraiška) turi geometrinį skirstinį.

Paraiškos laukimo eilėje laiko W skirstinys:

$$F_W(t) = 1 - \sigma \cdot e^{-\mu\cdot(1-\sigma)\cdot t} \qquad t \geq 0.$$

Vidutinį paraiškos buvimo sistemoje laiką galima rasti iš Laplaso-Stiltjeso transformacijos, bet taip pat ir skaičiuojant tiesiogiai. Pažymim L_a - paraiškų skaičius kurį randa į sistemą atvykusi paraiška. Tuomet:

$$E(S) = E(L^a)\frac{1}{\mu} + \frac{1}{\mu}.$$

Pagal Little‘o taisyklę:

$$E(L) = \lambda E(S) = \lambda\left(E(L^a)\frac{1}{\mu} + \frac{1}{\mu}\right) \tag{1.9}$$

Kadangi nagrinėjame sistemą G/M/1, tai atvykimo srautas nėra Puasoninis. Taigi, $E(L^a) \neq E(L)$. $E(L^a)$ galime apskaičiuoti tiesiogiai, pagal 1) formulę.

$$E(L^a) = \sum_{n=0}^{\infty} na_n = \sum_{n=0}^{\infty} n\cdot(1-\sigma)\sigma^n = \frac{\sigma}{1-\sigma}. \tag{1.10}$$

Iš (1.9) ir (1.10) gauname, kad:

$$E(L) = \frac{\lambda}{(1-\sigma)\mu} = \frac{\rho}{(1-\sigma)}. \tag{1.11}$$

2. TIRIAMOJI DALIS

2.1 Analizinis aptarnavimo sistemų charakteristikų apskaičiavimas

2.1.1 M/G/1 sistemos tyrimas

Ištirsime metodo galimybes aptarnavimo sistemų G/M/1 aproksimavime. Tikslas – patikrinti aproksimavimo tikslumą, bei palyginti su imitacinio modeliavimo rezultatais bei analiziniais skaičiavimais.

Nagrinėsime keturis skirstinius:

1) Lognormalųjį (tankio funkcija $p(t) = \frac{1}{t\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\ln(t)-m)^2}{2\sigma^2}\right\}$) su parametrais m = – 0.5 ir σ = 0.9;
2) Tolygųjį (tankio funkcija $p(t) = \frac{1}{b-a}$, $a \le t \le b$) su parametrais a = 0.2 ir b =1.2;
3) Veibulo (pasiskirstymo funkcija $p(t) = 1 - \exp\left\{-\frac{t}{\lambda}\right\}^k$, $t \ge 0$, $k > 0$, $\lambda > 0$) su parametrais k = 0.7 ir λ = 0.7;
4) Gama (tankio funkcija $p(t) = \frac{\lambda^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\lambda \cdot t}$) su parametrais α = 1.5 ir λ = 1.6.

Laikysime, kad paraiškų atvykimo srautas λ = 1. Įvertinsime kai kurias svarbesnes sistemos charakteristikas: vidutinį sistemoje esančių paraiškų skaičių E(L); vidutinį eilės ilgį $E(L^q)$; tikimybes a_n ir p_n (n = 0,1,2,3,4,5).

Naudodami 1 skyriaus formules, apskaičiuosime analizines sistemos charakteristikų išraiškas tikrajam skirstiniui, ir naudojant įvairias aproksimacijas. Kadangi atskirais atvejais analizinių išraiškų radimas gali būti labai sudėtingas, naudosimės matematiniais programų paketais (pvz. MathCad). Kompiuteriniai skaičiavimai atlikti 15 skaitmenų po kablelio tikslumu, tačiau dėl patogumo pateiksime suapvalintus rezultatus (4 skaitmenų po kablelio tikslumu)

2.1.1.1 Lognormalusis skirstinys

Tiriamo lognormaliojo skirstinio 3 pradiniai momentai: $m_1 = 0.9094$, $m_2 = 1.8589$, $m_3 = 8.542$.

Pirmiausia apskaičiuosime sistemos charakteristikas tiesiogiai iš lognormaliojo skirstinio. Lognormaliojo skirstinio Laplaso-Stiltjeso transformacija B(s):

$$\widetilde{B}(s) = \int_0^\infty \frac{1}{t\sigma\sqrt{2\pi}} \exp\left\{ -st - \frac{(\ln(t) - m)^2}{2\sigma^2} \right\} dt$$

Deja, lognormaliojo skirstinio Laplaso-Stiltjeso transformacija neturi paprastos analizinės išraiškos, todėl kompiuteriniuose skaičiavimuose naudosime integralinę išraišką.

Lognormaliojo skirstinio vidurkis (arba aptarnavimo laiko vidurkis E(B)): $E(B) = 0.9094$.

Sistemos užimtumo parametras ρ sutampa su E(B): $\rho = \lambda \cdot E(B) = E(B) = 0.9094$.

Tikimybes p_n teoriškai galima gauti iš Polačeko-Chinčino formulės (1.3), tačiau lognormaliojo skirstinio atveju tai labai sudėtingas uždavinys, todėl dydžius p_n apskaičiuosime iš generuojančios funkcijos P_L, pasinaudodami savybe:

$$p_n = \frac{P_L^{(n)}(0)}{n!}.$$

Pateikiame p_n (priminsime, kad sistemoms M/G/1 $p_n = a_n$) reikšmes:

$p_0 = 0.0906$; $p_1 = 0.0851$ $p_2 = 0.0731$ $p_3 = 0.0640$ $p_4 = 0.0571$ $p_5 = 0.0515$

Vidutinį paraiškų skaičių sistemoje teoriškai galima apskaičiuoti naudojant pirmą arba antrą Polačeko-Chinčino formulės variantą. Naudojant (1.3) formulę, vidurkis skaičiuojamas iš generuojančios funkcijos, naudojant (1.6) – iš Laplaso-Stiltjeso transformacijos. Kadangi lognormaliojo skirstinio Laplaso-Stiltjeso transformacija turi sudėtingą išraišką, tiesioginis E(L) skaičiavimas per daug sudėtingas, netgi naudojant kompiuterį (pvz., šiuo atveju programa MathCad skaičiuojant tiesiogiai rodo klaidą). Naudosime paprastesnę analizinę formulę vidutiniam eilės ilgiui $E(L^q)$ apskaičiuoti:

$$E(L^q) = \frac{\lambda \cdot \rho \cdot E(B^2)}{2 \cdot (1-\rho) E(B)} = 10.2559. \qquad \textbf{(2.1)}$$

čia $E(B^2)$ – nagrinėjamo lognormaliojo skirstinio antrasis pradinis momentas m_2.

Vidutinis paraiškų skaičius sistemoje randamas iš lygybės:

$E(L) = E(L^q) + \lambda E(B) = 11.1653$.

Little‘o formulė $E(L) = \lambda E(S)$. Mūsų atveju λ = 1, taigi E(L) = E(S).

Dabar bandysime aproksimuoti lognormalųjį skirstinį pirmojo tipo Erlango mišiniu. Aproksimuojant lognormalųjį skirstinį Erlango mišiniu, gauti tokie mišinio parametrai:

k = 1; p = 0.1103; μ = 0.9784.

Kadangi k = 1, tai viena mišinio dedamoji lygi nuliui, mišinys išsigimsta. Tokio Erlango mišinio Laplaso-Stiltjeso transformacija B(s):

$$\widetilde{B}(s) = p + (1 - p)\frac{\mu}{\mu + s}. \qquad \textbf{(2.2)}$$

Generuojančią dydžio L funkciją gauname įstatę (2.2) į (1.3) išraišką.Kadangi L generuojanti funkcija paprastesnė negu lognormaliojo skirstinio atveju, galima programa MathCad išskleisti generuojančią funkciją (komanda „series“) ir gauti tikimybes p_n :

$p_0 = 0.0906$; $p_1 = 0.0741$ $p_2 = 0.0680$ $p_3 = 0.0625$ $p_4 = 0.0574$ $p_5 = 0.0527$

Galime pastebėti skirtumą tarp skaičiavimų tiesiogiai iš lognormaliojo skirstinio ir naudojant šią aproksimaciją.

Tuo tarpu apskaičiuojant vidutinį eilės ilgį rezultatai sutampa – tiek skaičiuojant tiesiogiai iš Polačeko-Chinčino formulės paraiškos laukimo laikui, tiek ir iš 1) formulės.

$$E(L^q) = 10.2559.$$

Tokio rezultato ir reikėjo tikėtis, kadangi, kaip matome iš 1), vidutinis paraiškos laukimo laikas priklauso tik nuo dviejų pradinių skirstinio momentų (mūsų aproksimacija parinkta taip, kad šie du momentai sutaptų).

Aproksimuokime nagrinėjamą lognormalųjį skirstinį antrojo tipo Erlango (t.y., sulyginsim tris pradinius momentus) mišiniu. Pagal anksčiau pateiktą algoritmą gauname tokias antrojo tipo Erlango mišinio parametrų reikšmes:

n = 1; p = 0.0069; $\mu_1 = 0.2101$; $\mu_2 = 1.1328$.

Kaip matome, n = 1, taigi Erlango mišinys virsta paprastu eksponentiniu mišiniu, arba tiesiog hipergeometriniu skirstiniu. Nagrinėkamo Erlango mišinio Laplaso-Stiltjeso transformacija:

$$\widetilde{B}(s) = p\frac{\mu_1}{\mu_1 + s} + (1 - p)\frac{\mu_2}{\mu_2 + s}.$$

Laplaso-Sitltjeso transformacijos išraiška nesudėtinga, todėl įstačius ją į Polačeko-Chinčino formulę (1.3), galima gauti ir analizinį generuojančios funkcijos skleidinį. Kompiuteriu apskaičiuotos p_n reikšmes:

$p_0 = 0.0906$; $p_1 = 0.0808$; $p_2 = 0.0723$; $p_3 = 0.0649$; $p_4 = 0.0585$; $p_5 = 0.0529$.

Galime pastebėti, kad sulyginus 3 pirmuosius momentus, skirtumas tarp tiesiogiai apskaičiuotų (t.y., iš lognormaliojo skirstinio) ir aproksimacijos mažesnis negu naudojant 2 momentų aproksimaciją.

Vidutinis eilės ilgis, kaip ir sulyginus 2 momentus, sutampa su aproksimuojamo skirstinio atveju (tiek skaičiuojant iš Laplaso-Stiltjeso transformacijos, tiek ir tiesiogiai iš (2.1) formulės)

$$E(L^q) = 10.2559\,.$$

Aproksimuosime Kokso skirstiniu, sulygindami 2 pradinius momentus. Kokso skirstinio parametrai:

$$p = 0.4007\,; \quad \mu_1 = 2.1993; \quad \mu_2 = 0.8812.$$

Kokso skirstinio Laplaso-Stiltjeso transformacija:

$$\widetilde{B}(s) = p\,\frac{\mu_1}{\mu_1 + s}\cdot\frac{\mu_2}{\mu_2 + s} + (1-p)\frac{\mu_1}{\mu_1 + s}\,.$$

Analogiškai Erlango mišiniams, įstatę Laplaso-Stiltjeso transformaciją į Polačeko-Chinčino formulę, ir gautąją generuojančią funkciją išskleidę eilute (MathCad komanda „series"), gaunam p_n reikšmes:

$$p_0 = 0.0906; \quad p_1 = 0.0769; \quad p_2 = 0.0685; \quad p_3 = 0.0623; \quad p_4 = 0.0570; \quad p_5 = 0.0523.$$

Taigi, taikant Kokso aproksimaciją dviem momentam, gaunami panašūs rezultatai, kaip ir Erlango mišinio dviem momentam atveju.

Vidutinis eilės ilgis sutampa su tikrąja reikšme:

$$E(L^q) = 10.2559\,.$$

Aproksimuosime Kokso skirstiniu, sulygindami 3 pradinius momentus. Sulyginant 3 pirmuosius momentus, gaunamos tokios Kokso skirstinio parametrų reikšmės:

$$p = 0.0056; \quad \mu_1 = 1.1328; \quad \mu_2 = 0.2101.$$

(Įdomu pastebėti, kad μ_1 ir μ_2 reikšmės sutapo su Erlango mišinio 3 momentų aproksimacija).

MathCad‘u apskaičiuotos p_n reikšmės:

$$p_0 = 0.0906; \quad p_1 = 0.0808; \quad p_2 = 0.0723; \quad p_3 = 0.0649; \quad p_4 = 0.0585; \quad p_5 = 0.0529.$$

Taigi, p_n reikšmės sutapo su Erlango mišinio 3 momentams atveju (bent jau 4 skaičių po kablelio tikslumu).

Sistemos M/G/1 aproksimavimo rezultatus (kai G lognormalusis skirstinys) pateikiame lentelėje.

2.1 lentelė

Sistemos M/G/1 aproksimavimas, kai G – lognormalusis skirstinys

	p_0	p_1	p_2	p_3	p_4	p_5	$E(L)$	$E(L^q)$
Lognormalusis	0.0906	0.0851	0.0731	0.064	0.0571	0.0515	11.1653	10.2559
Erlango (2 mom.)	0.0906	0.0741	0.068	0.0625	0.0574	0.0527	11.1653	10.2559
Santykinė pakl.	0.00%	-12.93%	-6.98%	-2.34%	0.53%	2.33%	0.00%	0.00%
Erlango (3 mom.)	0.0906	0.0808	0.0723	0.0649	0.0585	0.0529	11.1653	10.2559
Santykinė pakl.	0.00%	-5.05%	-1.09%	1.41%	2.45%	2.72%	0.00%	0.00%
Kokso (2 mom.)	0.0906	0.0769	0.0685	0.0623	0.057	0.0523	11.1653	10.2559
Santykinė pakl.	0.00%	-9.64%	-6.29%	-2.66%	-0.18%	1.55%	0.00%	0.00%
Kokso (3 mom.)	0.0906	0.0808	0.0723	0.0649	0.0585	0.0529	11.1653	10.8155
Santykinė pakl.	0.00%	-5.05%	-1.09%	1.41%	2.45%	2.72%	0.00%	0.00%

2.1.1.2 Tolygusis skirstinys

Nagrinėjamo tolygiojo skirstinio 3 pirmieji pradiniai momentai: $m_1 = 0.7$, $m_2 = 0.5733$, $m_3 = 0.518$.

Šio skirstinio analizinė išraiška paprastesnė negu lognormaliojo, todėl ir sistemos M/G/1 analizė šiek tiek paprastesnė.

Tolygaus skirstinio Laplaso-Stiltjeso transformacija $\widetilde{B}(s)$:

$$\widetilde{B}(s) = \int_a^b \frac{1}{b-a} \cdot e^{-s \cdot t} dt = \frac{e^{-a \cdot s}}{s(b-a)} - \frac{e^{-b \cdot s}}{s(b-a)}.$$

.

Laplaso-Stiltjeso transformacijos išraiška nesudėtinga, todėl charakteristikų reikšmes galima skaičiuoti tiesiogiai iš Polačeko-Chinčino formulių (1.3) ir (1.6). Apskaičiuojam p_n reikšmes:

$p_0 = 0.3$; $p_1 = 0.2797$; $p_2 = 0.1821$; $p_3 = 0.1057$; $p_4 = 0.0591$; $p_5 = 0.0327$.

Vidutinis eilės ilgis $E(L^q)$:

$$E(L^q) = \frac{\lambda \cdot \rho \cdot E(B^2)}{2 \cdot (1-\rho) E(B)} = 0.9556 .$$

Skaičiuojam vidutinį paraiškų skaičių sistemoje E(L) iš generuojančios funkcijos:

$$E(L)=\frac{d\widetilde{B}(s)}{ds}\bigg|_{s=0}=1.6556$$

Aproksimuoje tolygųjį skirstinį pirmojo tipo Erlango mišiniu (t.y., sulyginam 2 pradinius momentus). Erlango mišinio parametrai:

n = 6; p = 0.0622; μ = 8.4825.

Erlango mišinio Laplaso-Stiltjeso transformacija $\widetilde{B}(s)$:

$$\widetilde{B}(s)=p\left(\frac{\mu}{\mu+s}\right)^{n-1}+(1-p)\left(\frac{\mu}{\mu+s}\right)^{n}.$$

Įstatę į Polačeko-Chinčino (1.3) formulę gaunam paraiškų skaičiaus sitemoje L generuojančią funkciją $P_L(z)$, iš kurios randame p_n:

p_0 = 0.3; p_1= 0.2812; p_2 = 0.1813; p_3 = 0.1048; p_4 = 0.0588; p_5 = 0.0328..

Skaičiuojant iš Polačeko-Chinčino formulės (1) E(L) reikšmė sutapo su tikrąja (t.y., gauta skaičiuojant iš nagrinėjamo tolygaus skirstinio):

$$E(L)=\frac{dP_L(z)}{dz}\bigg|_{z=1}=1.6556$$

Vidutinis paraiškų skaičius taip pat sutampa su tikrąja reikšme:

$$E(L^q)=0.9556\,.$$

Aproksimuojame tolygųjį skirstinį antrojo tipo Erlango mišiniu (t.y., sulyginam 3 pradinius momentus). Gauti Erlango mišinio parametrai:

n = 9; p = 0.9438; μ_1 = 12.1731; μ_2 = 230.84.

Kaip ir ankstesniais atvejais, taikant aproksimavimą sistemos charakteristikas galima apskaičiuoti išskleidžiant Polačeko-Chinčino formules. Tikimybės p_n:

p_0 = 0.3; p_1= 0.2794; p_2 = 0.1826; p_3 = 0.1056; p_4 = 0.0590; p_5 = 0.0327.

Vidutinis eilės ilgis:

$$E(L)=0.9556$$

Vidutinis paraiškų skaičius sistemoje:

$$E(L)=1.6556$$

Parodysime, kad Kokso skirstiniu negalima aproksimuoti tolygaus skirstinio, kurio parametrai a ir b yra teigiami. Kaip minėjome anksčiau, aproksimuojant Kokso skirstiniu aproksimuojamo skirstinio variacijos koeficientas c turi tenkinti sąlygą:

$$c^2 \geq 0.5 .$$

Deja, tolygusis skirstinys, kurio parametrai a>0 ir b>0 (o aptarnavimo sistemų teorijoje būtent tokie skirstiniai ir naudojami), šios sąlygos netenkina. Tuo nesunku įsitikinti:

$$c^2 = \frac{Var(X)}{E^2(X)} = \frac{(b-a)^2/12}{((a+b)/2)^2} = \frac{1}{3}\left(\frac{b-a}{b+a}\right)^2 \leq \frac{1}{3}, \text{ kai a >0 b>a.}$$

Skaičiavimų rezultatus pateikiame lentelėje:

2.2 lentelė

Sistemos M/G/1 aproksimavimas, kai G – tolygusis skirstinys

	p_0	p_1	p_2	p_3	p_4	p_5	$E(L)$	$E(L^q)$
Tolygusis	0.3000	0.2797	0.1821	0.1057	0.0591	0.0327	1.6556	0.9556
Erlango (2 mom.)	0.3000	0.2812	0.1813	0.1048	0.0588	0.0328	1.6556	0.9556
Santykinė pakl.	0.00%	0.54%	-0.44%	-0.85%	-0.51%	0.31%	0.00%	0.00%
Erlango (3 mom.)	0.3000	0.2794	0.1826	0.1056	0.0590	0.0327	1.6556	0.9556
Santykinė pakl.	0.00%	-0.11%	0.27%	-0.09%	-0.17%	0.00%	0.00%	0.00%

2.1.1.3 Veibulo skirstinys

Tiriamo Veibulo skirstinio 3 pradiniai momentai: $m_1 = 0.8861$, $m_2 = 2.4643$, $m_3 = 12.7713$.

Veibulo skirstinio Laplaso-Stiltjeso transformacijos išraiška sudėtinga, todėl sistemos charakteristikų apskaičiavimui naudojome formulę:

$$p_n = \frac{P_L^{(n)}(0)}{n!} \qquad \textbf{(2.3)}$$

Vidutinių charakteristikų apskaičiavimas tiesiogiai iš Polačeko-Chinčino formulių nedavė rezultatų, todėl naudojome formules $E(L^q) = \dfrac{\lambda \cdot \rho \cdot E(B^2)}{2 \cdot (1-\rho)E(B)}$ ir $E(L) = E(L^q) + \lambda E(B)$.

Naudojant aproksimacijas charakteristikų reikšmės apskaičiuotos tiesiogiai iš Polačeko-Chinčino formulių. Pateikiame aproksimuojamų skirstinių parametrus. Pirmojo tipo Erlango mišinio parametrai:

$$n = 1; \quad p = 0.3628; \quad \mu = 0.7191$$

Antrojo tipo Erlango mišinio parametrai:

$$n = 1; \quad p = 0.3106; \quad \mu_1 = 0.5313; \quad \mu_2 = 2.2869$$

Kokso skirstinio parametrų reikšmės, sulyginant 2 pradinius momentus:

p = 0.2338; $\mu_1 =$ 2.2571; $\mu_2 = 0.5277$.

Kokso skirstinio parametrų reikšmės, sulyginant 3 pradinius momentus:

p = 0.2385; $\mu_1 =$ 2.2869; $\mu_2 = 0.5313$.

Skaičiavimų rezultatus pateikiame lentelėje:

2.3 lentelė

Sistemos M/G/1 aproksimavimas, kai G – Veibulo skirstinys

	p_0	p_1	p_2	p_3	p_4	p_5	$E(L)$	$E(L^q)$
Veibulo	0.1139	0.0771	0.0649	0.0576	0.0522	0.0477	11.7016	10.8155
Erlango (2 mom.)	0.1139	0.0671	0.0620	0.0573	0.053	0.049	11.7016	10.8155
Santykinė pakl.	0.00%	-12.97%	-4.47%	-0.52%	1.53%	2.73%	0.00%	0.00%
Erlango (3 mom.)	0.1139	0.0800	0.0648	0.0566	0.0512	0.0470	11.7016	10.8155
Santykinė pakl.	0.00%	3.76%	-0.15%	-1.74%	-1.92%	-1.47%	0.00%	0.00%
Kokso (2 mom.)	0.1139	0.0802	0.0649	0.0566	0.0512	0.0469	11.7016	10.8155
Santykinė pakl.	0.00%	4.02%	0.00%	-1.74%	-1.92%	-1.68%	0.00%	0.00%
Kokso (3 mom.)	0.1139	0.0800	0.0648	0.0566	0.0512	0.047	11.7016	10.8155
Santykinė pakl.	0.00%	3.76%	-0.15%	-1.74%	-1.92%	-1.47%	0.00%	0.00%

2.1.1.4 Gama skirstinys

Tiriamo gama skirstinio 3 pirmieji pradiniai momentai: $m_1 = 0.9375$, $m_2 = 1.4648$, $m_3 = 3.2043$.

Gama skirstinio Laplaso-Stiltjeso transformacijos išraiška nesudėtinga, todėl sistemos charakteristikas galėjome apskaičiuoti tiesiogiai iš Polačeko-Chinčino formulių (programa MathCad pateikia tikslų sprendinį).

Vidutinių charakteristikų apskaičiavimas tiek tiesiogiai iš Polačeko-Chinčino (1.3) ir (1.6) formulių, tiek ir naudojant paprastesnes formules $E(L^q) = \dfrac{\lambda \cdot \rho \cdot E(B^2)}{2 \cdot (1-\rho)E(B)}$ ir $E(L) = E(L^q) + \lambda E(B)$ davė tuos pačius rezultatus.

Naudojant aproksimacijas charakteristikų reikšmės apskaičiuotos tiesiogiai iš Polačeko-Chinčino formulių. Pateikiame aproksimuojamų skirstinių parametrus. Pirmojo tipo Erlango mišinio parametrai:

n = 2; p = 0.3101; $\mu = 1.8026$

Antrojo tipo Erlango mišinio parametrai:

n = 1; p = 0.3106; $\mu_1 =$ 0.5313; $\mu_2 =$ 2.2869

Kokso skirstinio parametrų reikšmės, sulyginant 2 pradinius momentus:

$$p = 0.75; \quad \mu_1 = 2.1333; \quad \mu_2 = 1.6$$

Kokso skirstinio parametrų reikšmės, sulyginant 3 pradinius momentus:

$$p = 0.7075; \quad \mu_1 = 1.4587; \quad \mu_2 = 2.808.$$

Skaičiavimų rezultatus pateikiame lentelėje:

2.4 lentelė

Sistemos M/G/1 aproksimavimas, kai G – gama skirstinys

	p_0	p_1	p_2	p_3	p_4	p_5	$E(L)$	$E(L^q)$
Gama	0.0625	0.067	0.064	0.0598	0.0555	0.0514	12.6563	11.7188
Erlango (2 mom.)	0.0625	0.0664	0.0639	0.0598	0.0556	0.0515	12.6563	11.7188
Santykinė pakl.	0.00%	-0.90%	-0.16%	0.00%	0.18%	0.19%	0.00%	0.00%
Erlango (3 mom.)	0.0625	0.0671	0.064	0.0597	0.0555	0.0514	12.6563	11.7188
Santykinė pakl.	0.00%	0.15%	0.00%	-0.17%	0.00%	0.00%	0.00%	0.00%
Kokso (2 mom.)	0.0625	0.0665	0.0639	0.0598	0.0555	0.0515	12.6563	11.7188
Santykinė pakl.	0.00%	-0.75%	-0.16%	0.00%	0.00%	0.19%	0.00%	0.00%
Kokso (3 mom.)	0.0625	0.0669	0.0641	0.0598	0.0555	0.0514	12.6563	11.7188
Santykinė pakl.	0.00%	-0.15%	0.16%	0.00%	0.00%	0.00%	0.00%	0.00%

Apibendrinant, galima teigti, kad sistemų G/M/1 aprokismavimas duoda gerus rezultatus. Tikimybių p_n paklaidų moduliai dažniausiai neviršija 5 %, o vidutinės sistemų charakteristikos $E(L)$ ir $E(L^q)$ apskaičiuojamos tiksliai.

2.1.2 G/M/1 sistemos tyrimas

Kaip rašėme teorinėje dalyje, G/M/1 sistemoms neišvedamos analitinės formulės tikimybėms p_n (tikimybė, kad sistemoje yra n paraiškų) apskaičiuoti, o tiktai tikimybėms a_n (tikimybė, kad į sistemą atvykusis paraiška randa n paraiškų). Priminsime, kad sistemoms G/M/1, skirtingai nuo sistemų M/G/1, šios tikimybės nesutampa. Akivaizdu, kad sistemų G/M/1 vidutinės charakteristikos $E(L^a)$ (vidutinis paraiškų skaičius, kurį randa atvykusi paraiška) ir $E(L)$ (vidutinis paraiškų skaičius sitemoje, kurį mato išorinis sistemos stebėtojas) taip pat skirsis. Taigi, sistemoms G/M/1 apskaičiuosime ir šias papildomas charakteristikas.

Kaip bendruosius skirstinius G parinksime tuos pačius anksčiau nagrinėtus skirstinius: lognormalųjį (su parametrais m = – 0.5 ir σ = 0.9) , tolygųjį (su parametrais a = 0.2 ir b = 1.2) , Veibulo (su parametrais k = 0.7 ir λ = 0.7) ir gama (su parametrais α = 1.5 ir λ = 1.6). Dėl stabilumo, t.y., kad būtų tenkinama sąlyga $\rho = \frac{\lambda}{\mu} < 1$ (priminsime, kad 1/λ yra paraiškų srauto vidurkis), parinksime μ = 1.5 visiems keturiems atvejams.

2.1.2.1 Lognormalusis skirstinys

Nagrinėjant G/M/1 sistemą, svarbiausia nustatyti parametrą σ, kuris yra lygties (1.7)

$$\sigma = \widetilde{A}(\mu - \mu\sigma)$$

sprendinys , priklausantis intervalui (0 ; 1).

Lognormaliojo skirstinio atveju, Laplaso-Stiltjeso transformacija $\widetilde{A}(s)$ neturi paprastos analizinės išraiškos, todėl lygties 1) sprendinys gaunamas skaitiniais metodais. Mūsų nagrinėjamu atveju, programa MathCad (sprendimo blokas „Given-Find“) pateikia tokį apytikslį lygties sprendinį:

$$\sigma = 0.5438 .$$

Užimtumo koeficientas ρ:

$$\rho = \frac{E(B)}{m_1} = \frac{1/1.5}{0.9094} = 0.7331 .$$

Apskaičiuosime sistemos charakteristikas. Tikimybės a_n turi pavidalą (1.8):

$$a_n = (1 - \sigma)\sigma^n \quad n = 0,1,2...$$

Pagal MathCad apskaičiuotą σ reikšmę:

$a_0 = 0.4562$; $a_1 = 0.2481$; $a_2 = 0.1349$; $a_3 = 0.0734$; $a_4 = 0.0399$; $a_5 = 0.0217$.

Vidutines sistemos charakteristikas randam iš formulių (1.10) ir (1.11):

$$E(L^a) = \frac{\sigma}{1-\sigma} = 1.1923$$

$$E(L) = \frac{\rho}{1-\sigma} = 1.6072 .$$

Aproksimuojame tiriamą lognormalųjį skirstinį Erlango pirmojo tipo mišiniu. (Kadangi tiriame tuos pačius skirstinius kaip ir sistemoms M/G/1, naudojame anksčiau apskaičiuotas Erlango mišinio parametrų reikšmes). Erlango mišinio Laplaso-Stiltjeso transformacija:

$$\widetilde{B}(s) = p + (1-p)\frac{\mu}{\mu + s} .$$

Kadangi Laplaso-Stiltjeso transformacija paprastesnė negu lognormaliojo skirstinio, galima rasti tikslų lygties sprendinį. Pateikiame MathCad'u apskaičiuotą lygties sprendinį (4 skaičių po kablelio tikslumu):

$$\sigma = (0.7625\,;1.0000)$$

Kadangi sprendinys $\sigma = 1$ mums netinka, galutinis atsakymas $\sigma = 0.7625$. Kaip matome, sprendinys ženkliai skiriasi nuo tiesiogiai iš lognormaliojo skirstinio apskaičiuotos reikšmės ($\sigma = 0.5438$). Santykinė paklaida viršija net 40 %. Galimi du variantai: arba aproksimavimas sistemoms G/M/1 netinka, arba tiesioginis skaičiavimas nėra tikslus (prisiminkime, kad sprendinys gautas naudojant skaitinius metodus). Aproksimavus lognormalųjį skirstinį pirmojo tipo Erlango mišiniu gautos tikimybės a_n:

$a_0 = 0.2375$; $a_1 = 0.1811$; $a_2 = 0.1318$; $a_3 = 0.1053$; $a_4 = 0.0803$; $a_5 = 0.0612$.

Vidutinės sistemos charakteristikos:

$$E\left(L^a\right) = \frac{\sigma}{1-\sigma} = 3.2112$$

$$E(L) = \frac{\rho}{1-\sigma} = 3.0873\,.$$

Taigi, atlikus aproksimaciją, gautos sistemos charakteristikų reikšmės ženkliai (daugiau nei 100 %) skiriasi nuo tiesiogiai apskaičiuotų reikšmių.

Aproksimuojame antrojo tipo Erlango mišiniu - t.y., sulyginam 3 pradinius skirstinių momentus. Lygties 1) sprendinys (MathCad'as randa tikslų sprendinį):

$$\sigma = (0.7445\,;1.0000\,;1.1507)$$

Parenkame sprendinį priklausantį intervalui (0 ; 1). Taigi, $\sigma = 0.7445$.

Apskaičiuojame sistemos charakteristikas. Tikimybės a_n:

$a_0 = 0.2555$; $a_1 = 0.1902$; $a_2 = 0{,}1416$; $a_3 = 0.1054$; $a_4 = 0.0785$; $a_5 = 0.0584$.

Vidutinės charakteristikos $E\left(L^a\right)$ ir $E(L)$:

$$E\left(L^a\right) = \frac{\sigma}{1-\sigma} = 2.9144\,.$$

$$E(L) = \frac{\rho}{1-\sigma} = 2.8697\,.$$

Kaip matome, aproksimacijos rezutai smarkiai skiriasi nuo tiesiogiai lognormaliąjam skirstiniui apskaičiuotų reikšmių. Kadangi lognmormaliojo skirstinio Laplaso-Stiltjeso transformacijos pavidalas sudėtingas, gali būti kad tiesioginis skaičiavimas naudojant apytikslį lygties (1.7) sprendinį nėra tikslus.

Aproksimuojam Kokso skirstiniu, sulygindami 2 pradinius momentus.

Lygties 1) sprendinys (MathCad'as randa tikslų sprendinį):

$$\sigma = (0.7580\,; 1.0000\,; 2.2957).$$

Parenkame sprendinį priklausantį intervalui (0 ; 1). Taigi, $\sigma = 0.7580$ .Tikimybės a_n:

$a_0 = 0.2420$; $a_1 = 0.1834$; $a_2 = 0{,}1390$; $a_3 = 0.1054$; $a_4 = 0.0799$; $a_5 = 0.0606$.

Vidutinės charakteristikos $E(L^a)$ ir $E(L)$:

$$E(L^a) = \frac{\sigma}{1-\sigma} = 3.1319\,.$$

$$E(L) = \frac{\rho}{1-\sigma} = 3.0219\,.$$

Aproksimuojam Kokso skirstiniu, sulygindami 3 pradinius momentus.

Lygties 1) sprendinys (MathCad'as randa tikslų sprendinį):

$$\sigma = (0.7445\,; 1.0000\,; 1.1507)$$

Parenkame sprendinį priklausantį intervalui (0 ; 1). Taigi, $\sigma = 0.7445$. Įdomu pastebėti, kad σ reikšmė praktiškai sutapo su antrojo tipo Erlango mišinio atveju (MathCad apskaičiuotos reikšmės sutapo 12 skaitmenų po kablelio tikslumu).

Kadangi σ reikšmė sutapo, sistemos charakteristikos bus tokios pačios kaip ir antrojo tipo Erlango mišinio atveju. Tikimybės a_n:

$a_0 = 0.2555$; $a_1 = 0.1902$; $a_2 = 0{,}1416$; $a_3 = 0.1054$; $a_4 = 0.0785$; $a_5 = 0.0584$.

Vidutinės charakteristikos $E(L^a)$ ir $E(L)$:

$$E(L^a) = \frac{\sigma}{1-\sigma} = 2.9144\,.$$

$$E(L) = \frac{\rho}{1-\sigma} = 2.8697\,.$$

Skaičiavimų rezultatus pateikiame lentelėje:

2.5 lentelė

Sistemos G/M/1 aproksimavimas, kai G – lognormalusis skirstinys

	a_0	a_1	a_2	a_3	a_4	a_5	$E(L^a)$	$E(L)$
Lognormalusis	0.4562	0.2481	0.1349	0.0734	0.0399	0.0217	1.1923	1.6072
Erlango (2 mom.)	0.2375	0.1811	0.1318	0.1053	0.0803	0.0612	3.2112	3.0873
Santykinė pakl.	-47.94%	-27.01%	-2.30%	43.46%	101.2%	182.03%	169.33%	92.09%

Erlango (3 mom.)	0.2555	0.1902	0.1416	0.1054	0.0785	0.0584	2.9144	2.8697
Santykinė pakl.	-43.99%	-23.34%	4.97%	43.60%	96.74%	169.12%	144.44%	78.55%
Kokso (2 mom.)	0.2420	0.1834	0.139	0.1054	0.0799	0.0606	3.1319	3.0219
Santykinė pakl.	-46.95%	-26.08%	3.04%	43.60%	100.25%	179.26%	162.68%	88.02%
Kokso (3 mom.)	0.2555	0.1902	0.1416	0.1054	0.0785	0.0584	2.9144	2.8697
Santykinė pakl.	-43.99%	-23.34%	4.97%	43.60%	96.74%	169.12%	144.44%	78.55%

2.1.2.2 Tolygusis skirstinys

Tolygiojo skirstinio Laplaso-Stiltjeso transformacijos išraiška žymiai paprastesnė negu lognormaliojo, todėl galima būtų tikėtis, kad sistemos G/M/1 analizė bus tikslesnė bei patikimesnė.

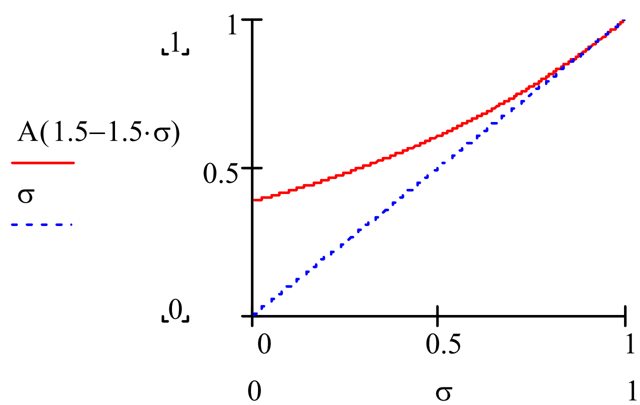
Sistemos parametro σ reikšmę galima apskaičiuoti tiksliai. Martchcad'o apskaičiuotas tikslus sprendinys:

$$\sigma = 0.0804\,;$$

Tuo tarpu skaitiniais metodais gautas sprendinys labai smarkiai skiriasi nuo tikslaus:

$$\sigma = 0.9196\,.$$

Kuris sprendinys teisingas, patikrinsime grafiškai. T.y., patikrinsime kur kertasi kreivės $\widetilde{A}(1.5 - 1.5 \cdot \sigma)$ ir σ (šiuo atveju $\widetilde{A}(s)$ – tolygaus skirstinio Laplaso-Stiltjeso transformacija).



2.1 pav. Grafinis lygties sprendimo patikrinimas

Iš 2.1 pav. akivaizdu, kad teisingas yra sprendinys $\sigma = 0.9196$.

Apskaičiuojam sistemos charakteristikas (naudojam sprendinį $\sigma = 0.9196$):

a_0 = 0.0804 ; a_1 = 0.0739; a_2 = 0,0680; a_3 = 0.0625; a_4 = 0.0575; a_5 = 0.0529.

$$E\left(L^a\right) = \frac{\sigma}{1-\sigma} = 11.4451\,.$$

$$E(L) = \frac{\rho}{1-\sigma} = 11.8525 .$$

Palyginsime tiesiogiai iš tolygaus skirstinio gautas sistemos charakteristikų reikšmes su aproksimuojamos sistemos reikšmėmis. metu Apruoksimuojamų sistemų parametrų σ reikšmės (priminsime, kad tolygaus skirstinio negalime aproksimuoti Kokso skirstiniu):

a) pirmojo tipo Erlango mišinys (sulyginant 2 momentus) – σ = 0.9195;

b) antrojo tipo Erlango mišinys (sulyginant 3 momentus) – σ = 0.9197;

Priminsime, kad skaičiuojant tiesiogiai iš tolygaus skirstinio gauta $\sigma = 0.9196$. Taigi, apriksimacija labai tiksli. Rezultatus surašome į lentelę:

2.6 lentelė

Sistemos G/M/1 aproksimavimas, kai G – tolygusis skirstinys

	a_0	a_1	a_2	a_3	a_4	a_5	$E(L)$	$E(L^q)$
Tolygusis	0.0804	0.0739	0.0680	0.0625	0.0575	0.0529	11.4451	11.8525
Erlango (2 mom.)	0.0805	0.0740	0.0680	0.0626	0.0575	0.0529	11.4284	11.8366
Santykinė pakl.	0.12%	0.14%	0.00%	0.16%	0.00%	0.00%	-0.15%	-0.13%
Erlango (3 mom.)	0.0800	0.0739	0.0679	0.0625	0.0575	0.0529	11.4277	11.8549
Santykinė pakl.	-0.50%	0.00%	-0.15%	0.00%	0.00%	0.00%	-0.15%	0.02%

2.1.2.3 Veibulo skirstinys

Kadangi Veibulo skirstinio Laplaso-Stiltjeso transformacijos išraiška sudėtinga, σ reikšmę apskaičiuosime skaitiniais metodais. Martchcad'o apskaičiuotas sprendinys:

$$\sigma = 0.8320$$

Sistemos charakteristikos:

a_0 = 0.1680; a_1= 0.1398; a_2= 0,1163; a_3 = 0.0968; a_4= 0.0805; a_5 = 0.0670.

$$E(L^a) = \frac{\sigma}{1-\sigma} = 4.9518 .$$

$$E(L) = \frac{\rho}{1-\sigma} = 2.6453 .$$

Aproksimavimo metu gautos sistemos parametro σ reikšmės:

1) pirmojo tipo Erlango mišinys – σ = 0.8422

2) antrojo tipo Erlango mišinys – σ = 0.8305

3) Kokso skirstinys sulyginant 2 pradinius momentus – σ = 0.8302

4) Kokso skirstinys sulyginant 3 pradinius momentus – σ = 0.8305

Aproksimavimo rezultatus surašome į lentelę:

2.7 lentelė

Sistemos G/M/1 aproksimavimas, kai G – Veibulo skirstinys

	a_0	a_1	a_2	a_3	a_4	a_5	$E(L^a)$	$E(L)$
Veibulo	0.1680	0.1398	0.1163	0.0968	0.0805	0.0670	4.9518	4.4780
Erlango (2 mom.)	0.1578	0.1329	0.1119	0.0943	0.0794	0.0669	5.3377	4.7684
Santykinė pakl.	-6.07%	-4.94%	-3.78%	-2.58%	-1.37%	-0.15%	7.79%	6.49%
Erlango (3 mom.)	0.1695	0.1408	0.1169	0.0971	0.0806	0.0670	4.8984	4.4378
Santykinė pakl.	0.89%	0.72%	0.52%	0.31%	0.12%	0.00%	-1.08%	-0.90%
Kokso (2 mom.)	0.1698	0.1409	0.1170	0.0971	0.0807	0.0670	4.8908	4.4321
Santykinė pakl.	1.07%	0.79%	0.60%	0.31%	0.25%	0.00%	-1.23%	-1.03%
Kokso (3 mom.)	0.1695	0.1408	0.1169	0.0971	0.0806	0.0670	4.8984	4.4378
Santykinė pakl.	0.89%	0.72%	0.52%	0.31%	0.12%	0.00%	-1.08%	-0.90%

2.1.2.4 Gama skirstinys

Kadangi Laplaso-Stiltjeso transformacijos išraiška nesudėtinga, galima rasti tikslią σ reikšmę. Martchcad'o apskaičiuotas tikslus sprendinys (jis sutapo ir su apytiksliu sprendiniu):

$$\sigma = (0.6611\,;1.0000)$$

Mūsų ieškomas sprendinys $\sigma = 0.6611$.

Sistemos charakteristikos:

a_0 = 0.3389; a_1= 0.2240; a_2 = 0,1481; a_3 = 0.0979; a_4 = 0.0647; a_5 = 0.0428.

$$E\left(L^a\right) = \frac{\sigma}{1-\sigma} = 1.9507\,.$$

$$E(L) = \frac{\rho}{1-\sigma} = 2.0983\,.$$

Aproksimavimo metu gautos sistemos parametro σ reikšmės:

1) pirmojo tipo Erlango mišinys – σ = 0.6629

2) antrojo tipo Erlango mišinys – σ = 0.6608

3) Kokso skirstinys sulyginant 2 pradinius momentus – σ = 0.6626

4) Kokso skirstinys sulyginant 3 pradinius momentus – 0.6613

Aproksimavimo rezultatus surašome į lentelę:

2.8 lentelė

Sistemos G/M/1 aproksimavimas, kai G – gama skirstinys

	a_0	a_1	a_2	a_3	a_4	a_5	$E(L^a)$	$E(L)$
Gama	0.3389	0.224	0.1481	0.0979	0.0647	0.0428	1.9507	2.0983
Erlango (2 mom.)	0.3371	0.2235	0.1481	0.0982	0.0651	0.0432	1.9667	2.1097
Santykinė pakl.	-0.53%	-0.22%	0.00%	0.31%	0.62%	0.93%	0.82%	0.54%
Erlango (3 mom.)	0.3392	0.2241	0.1481	0.0979	0.0647	0.0427	1.9485	2.0967
Santykinė pakl.	0.09%	0.04%	0.00%	0.00%	0.00%	-0.23%	-0.11%	-0.08%
Kokso (2 mom.)	0.3374	0.2236	0.1481	0.0981	0.065	0.0431	1.9634	2.1073
Santykinė pakl.	-0.44%	-0.18%	0.00%	0.20%	0.46%	0.70%	0.65%	0.43%
Kokso (3 mom.)	0.3387	0.224	0.1481	0.0979	0.0648	0.0428	1.952	2.0992
Santykinė pakl.	-0.06%	0.00%	0.00%	0.00%	0.15%	0.00%	0.07%	0.04%

Apibendrinant, galima teigti, kad sistemų G/M/1 aprokismavimas duoda gerus rezultatus. Tikimybių a_n ir vidutinių sistemos charakteristikų $E(L)$ ir $E(L^q)$ paklaidų moduliai dažniausiai neviršija 5 %.

2.2 Aptarnavimo sistemų skaitmeninis modeliavimas

Sistemoms modeliuoti naudosime prof. H. Pranevičiaus 1996 m. pasiūlytą kompiuterinę sistemą (žr. [5]), kuri leidžia analizuoti Markovo procesus, ir yra ypač patogi modeliuojant aptarnavimo sistemas.

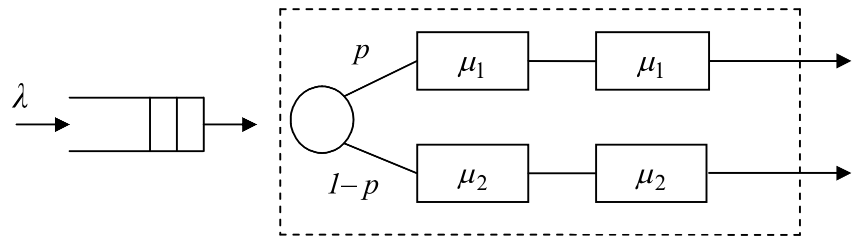
Panagrinėkime, pvz., sistemą M/M/1/N, tai yra, aptarnavimo sistemą su vienu aptarnavimo įrenginiu, ir ribotu paraišku skaičiumi N. Jeigu nagrinėjamas sistemos parametras x – paraiškų skaičius sistemoje, tai sistemą M/M/1/N galim interpretuoti kaip tolydaus laiko markovo procesą su diskrečia būsenų aibe. Akivaizdu, kad tokio proceso būsenų skaičius yra N + 1. Tokio proceso stacionarių tikimybių apskaičiavimas, net ir esant nelabai dideliems N (pvz., 20) yra komplikuotas uždavinys.

Minėta kompiuterinė sistema naudoja tam tikrą įvykių kalbą – t.y., detaliai aprašomas kiekvienas įvykis pakeičiantis sistemos būseną, taip pat būsenų apribojimai. Sistema apskaičiuoja kiekvienos būsenos stacionariąsias tikimybes, o iš jų galima rasti ir kitas charakteristikas, pvz., vidutinį eilės ilgį ar kt. Knygoje [5] pateikti rezultatai rodo, kad kompiuterinė sistema gerai apskaičiuoja įvairaus sudėtingumo Markovo procesų charakteristikas, pvz., M/M/c sistemų, ar M/M/1 sistemas su kokybės kontrole. Mūsų tikslas – ją pritaikyti sistemų M/G/1 ir G/M/1 modeliavimui, atliekant G aproksimavimą Erlango mišiniais ir Kokso skirstiniu.

2.2.1 M/G/1 sistemų skaitmeninis modeliavimas

2.2.1.1 M/G/1 sistemos modeliavimas, kai G – Erlango mišinys

Nagrinėkime sistemą M/G/1, kai skirstinio funkcija G aproksimuojama Erlango mišiniu. Šiuo atveju kaip pavyzdį pasirinksime antrojo tipo antros eilės Erlango mišinį .Aproksimuojamos sistemos fazinė schema:



2.2 pav. Sistemos M/G/1 aproksimavimo schema (G – Erlango mišinys)

Pirmiausiai atliksime sistemos formalizavimą – t.y., aprašysime tiriamą sistemą įvykių kalba.

Sistemos įvykių aibė $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

e_1 – paraiška atėjo į aptarnavimo bloką, pirmąją (fiktyvią) fazę su intensyvumu λ ;

e_2 – paraiška perėjo į aptarnavimo bloko antrą fazę su tikimybe p ;

e_3 – paraiška perėjo į aptarnavimo bloko trečią fazę su tikimybe $1 - p$;

e_4 – paraiška perėjo į aptarnavimo bloko ketvirtą fazę su intensyvumu μ_1 ;

e_5 – paraiška perėjo į aptarnavimo bloko penktą būseną su intensyvumu μ_2 ;

e_6 – paraiška baigta aptarnauti ketvirtoje fazėje ir palieka sistemą su intensyvumu μ_1 .

e_7 – paraiška baigta aptarnauti penktoje fazėje ir palieka sistemą su intensyvumu μ_2 .

Sistemos būsenų aibė:

$N = \{(n_1, n_2)\}, n_1 = \overline{0, L}; n_2 = \overline{0,5}$.

n_1 – paraiškų skaičius sistemoje (0 – jei sistema tuščia) ;

n_2 – parodo kurioje aptarnavimo bloko fazėje aptarnaujama paraiška (0 – jei aptarnavimo blokas tuščias);

Paraiškų skaičius sistemoje n_1 apribotas tam tikru natūraliuoju skaičiumi L. T.y., nors modeliuojame sistemą su begaline eile, praktikoje būtina įvesti apribojimą. Kitaip sakant, sistema M/G/1/∞ aproksimuojama M/G/1/L. Reikiamas tikslumas pasiekiamas parinkus pakankamai didelį L.

Pateikiame sistemos aprašymo įvykių kalba pseudokodą:

e1: if $n_2 = 0$ and $n_1 < L$ then $n_1 \leftarrow n_1 + 1$; $n_2 \leftarrow 1$; else if $n_1 < L$ then $n_2 \leftarrow n_2 + 1$; end if end if Return Intens $\leftarrow \lambda$	e2: if $n_2 = 1$ then $n_2 \leftarrow 2$ end if Return Intens $\leftarrow p \cdot C$
e3: if $n_2 = 1$ then $n_2 \leftarrow 3$ end if Return Intens $\leftarrow (1 - p) \cdot C$	e4: if $n_2 = 2$ then $n_2 \leftarrow 4$ end if Return Intens $\leftarrow \mu_1$
e5: if $n_1 = 3$ then $n_1 \leftarrow 5$ end if Return Intens $\leftarrow \mu_2$	e6: if $n_1 = 4$ if $n_2 > 1$ then $n_1 \leftarrow n_1 - 1$; $n_2 \leftarrow 1$; else $n_2 \leftarrow n_2 + 1$; $n_2 \leftarrow 0$ end if end if Return Intens $\leftarrow \mu_1$
e7: if $n_1 = 5$ if $n_2 > 1$ then $n_1 \leftarrow n_1 - 1$; $n_2 \leftarrow 1$; else $n_2 \leftarrow n_2 + 1$; $n_2 \leftarrow 0$ end if end if Return Intens $\leftarrow \mu_2$	

Įvykiuose e2 ir e3 naudojamas simbolis C – pakankamai didelis teigiamas skaičius. Skaičius C įvedamas tam, kad paraiškų buvimo laikas fiktyvioje fazėje būtų daug kartų mažesnis negu fazėse su eksponentiniais skirstiniais. Praktinis sistemų modeliavimas parodė, kad toks modeliavimas (t.y., įvedant fiktyvią fazę) duoda teisingus rezultatus. Konstantos C parinkimas priklauso nuo kitų sistemos būsenų intensyvumų. Praktikoje galima naudoti tokį parinkimo metodą – konstanta C 100000 kartų didesnė už didžiausią iš likusių sistemos intensyvumų. Metodo kalibravimas parodė, kad taip aprašyta sistema teisingai (bent jau 4 skaitmenų po kablelio tikslumu) apskaičiuoja sistemos M/G/1 charakteristikas, kai G – Erlango mišinys.

Naudojant įvykių kalbą, apskaičiuojamos sistemos būsenų stacionariosios tikimybės $\pi(n_1, n_2)$. Iš jų galima apskaičiuoti sistemos charakteristikas. Pavyzdžiui, vidutinis sistemoje esančių paraiškų skaičius E(L) randamas iš formulės:

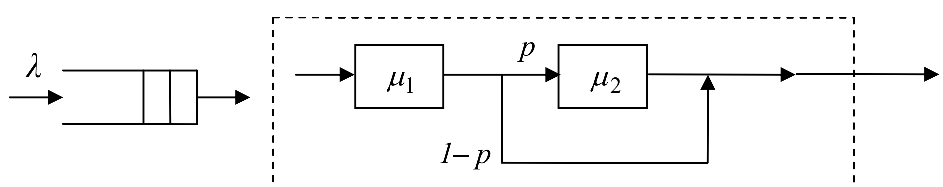
$$E(L) = \sum_{n_1, n_2} n_1 \pi(n_1, n_2) \tag{2.4}$$

Tikimybes p_n galima rasti iš formulės:

$$p_n = \sum_{n_1 = n} \pi(n_1, n_2). \tag{2.5}$$

2.2.1.2 M/G/1 sistemos modeliavimas, kai G – Kokso skirstinys

Nagrinėkime sistemą M/G/1, kai skirstinio funkcija G aproksimuojama Kokso skirstiniu. Sistemos fazinė schema (reikia pažymėti, kad sistemos schema nepriklauso nuo to,ar naudojam 2 ar trijų momentų sulyginimo metodą):



2.3 pav. Sistemos M/G/1 aproksimavimo schema (G – Kokso skirstinys)

Sistemos formalizavimas:

Sistemos įvykių aibė $E = \{e_1, e_2, e_3, e_4\}$

e_1 – paraiška atėjo į aptarnavimo bloką, pirmąją fazę su intensyvumu λ ;

e_2 – paraiška perėjo į aptarnavimo bloko antrą fazę su tikimybe p ir intensyvumu μ_1 ;

e_3 – paraiška baigta aptarnauti pirmoje fazėje ir palieka sistemą su tikimybe $1-p$ ir intensyvumu μ_1;

e_4 – paraiška baigta aptarnauti antroje fazėje ir palieka sistemą su intensyvumu μ_2.

Sistemos būsenų aibė:

$N = \{(n_1, n_2)\}, n_1 = \overline{0, L}; n_2 = \overline{0,2}$.

n_1 – paraiškų skaičius sistemoje (0 – jei sistema tuščia) ;

n_2 – parodo kurioje aptarnavimo bloko fazėje aptarnaujama paraiška (0 – jei aptarnavimo blokas tuščias);

Sistemos aprašymo įvykių kalba pseudokodas:

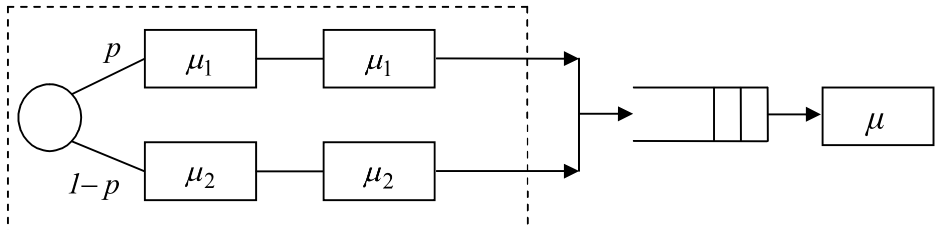
e1: if $n_2 = 0$ and $n_1 < L$ then $n_1 \leftarrow n_1 + 1$; $n_2 \leftarrow 1$; else if $n_1 < L$ then $n_2 \leftarrow n_2 + 1$; end if end if Return Intens $\leftarrow \lambda$	e2: if $n_2 = 1$ then $n_2 \leftarrow 2$ end if Return Intens $\leftarrow p \cdot \mu_1$
e3: if $n_2 = 1$ if $n_1 > 1$ then $n_1 \leftarrow n_1 - 1$; $n_2 \leftarrow 1$; else $n_1 \leftarrow 0$; $n_2 \leftarrow 0$ end if end if Return Intens $\leftarrow (1-p) \cdot \mu_1$	e4: if $n_2 = 2$ if $n_2 > 1$ then $n_1 \leftarrow n_1 - 1$; $n_2 \leftarrow 1$; else $n_1 \leftarrow 0$; $n_2 \leftarrow 0$ end if end if Return Intens $\leftarrow \mu_2$

Sistemos charakteristikos randamos iš (2.4) ir (2.5) formulių.

2.2.2 G/M/1 sistemų modeliavimas

2.2.2.1 G/M/1 sistemos modeliavimas, kai G – Erlango mišinys

Nagrinėkime sistemą G/M/1, kai skirstinio funkcija G aproksimuojama antrojo tipo antros eilės Erlango mišiniu . Sistemos fazinė schema:



2.4 pav. Sistemos G/M/1 aproksimavimo schema (G – Erlango mišinys)

Sistemos formalizavimas.

Sistemos įvykių aibė $E = \{e_1, e_2, e_3, e_4, e_5\}$

e_1 – paraiška atėjo į srauto bloką, pirmąją būseną;

e_2 – paraiška perėjo į srauto bloko antrą būseną;

e_3 – paraiška perėjo į srauto bloko trečią būseną;

e_4 – paraiška perėjo į srauto bloko ketvirtą būseną;

e_5 – paraiška perėjo į srauto bloko penktą būseną;

e_6 – paraiška atvyko į aptarnavimo įrenginį iš srauto bloko ketvirtos būsenos.

e_7 – paraiška atvyko į aptarnavimo įrenginį iš srauto bloko penktos būsenos.

e_8 – paraiška baigta aptarnauti aptarnavimo įrenginyje.

Sistemos būsenų aibė:

$N = \{(n_1, n_2, n_3)\}, n_1 = \overline{0,5}; n_2 = \overline{0, L}; n_3 = 0,1;$.

n_1 – parodo kurioje srauto bloko fazėje yra paraiška (0 – jei srauto blokas tuščias) ;

n_2 – paraiškų skaičius aptarnavimo sistemoje (eilėje ir aptarnavimo įrenginyje);

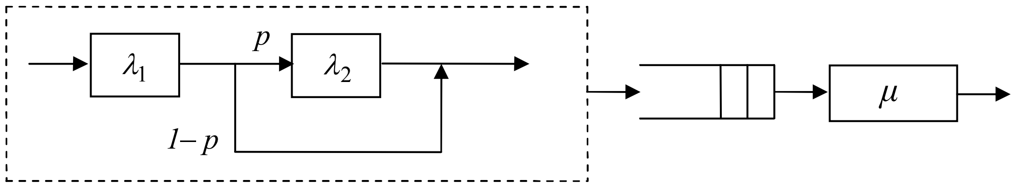
n_3 – paraiškų skaičius aptarnavimo įrenginyje (0 – jei aptarnavimo įrenginys tuščias, ir 1 – jei užimtas);

Sistemos aprašymo įvykių kalba pseudokodas:

e1: if $n_1 = 0$ then $n_1 \leftarrow 1$ end if Return Intens $\leftarrow C$	e2: if $n_1 = 1$ then $n_1 \leftarrow 2$ end if Return Intens $\leftarrow p \cdot C$
e3: if $n_1 = 1$ then $n_1 \leftarrow 3$ end if Return Intens $\leftarrow (1 - p) \cdot C$	e4: if $n_1 = 2$ then $n_1 \leftarrow 4$ end if Return Intens $\leftarrow \lambda_1$
e5: if $n_1 = 3$ then $n_1 \leftarrow 5$ end if Return Intens $\leftarrow \lambda_2$	e6: if $n_1 = 4$ if $n_3 = 0$ and $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$; $n_3 \leftarrow 1$ else if $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$ end if Return Intens $\leftarrow \lambda_1$
e7: if $n_1 = 5$ if $n_3 = 0$ and $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$; $n_3 \leftarrow 1$ else if $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$ end if Return Intens $\leftarrow \lambda_2$	e8: if $n_3 = 1$ if $n_2 > 1$ then $n_2 \leftarrow n_2 - 1$; $n_3 \leftarrow 1$ else $n_2 \leftarrow 0$; $n_3 \leftarrow 0$ end if end if Return Intens $\leftarrow \mu$

2.2.2.2 G/M/1 sistemos modeliavimas, kai G – Kokso skirstinys

Nagrinėkime sistemą G/M/1, kai skirstinio funkcija G aproksimuojama Kokso skirstiniu. Sistemos fazinė schema:



2.5 pav. Sistemos G/M/1 aproksimavimo schema (G – Kokso skirstinys)

Skaitmeninio modeliavimo algoritmas nepriklauso nuo aproksimavimo – t.y., sistemos aprašymas sutampa tiek taikant 2, tiek ir 3 momentų sulyginimą.

Sistemos formalizavimas:

Sistemos įvykių aibė $E = \{e_1, e_2, e_3, e_4\}$

e_1 – paraiška atėjo į srauto bloką, pirmąją fazę ;

e_2 – paraiška perėjo į srauto bloko antrą fazę su tikimybe p ir intensyvumu λ_1 ;

e_3 – paraiška baigta aptarnauti pirmoje srauto bloko fazėje ir ateina į aptarnavimo įrenginį su tikimybe $1-p$ ir intensyvumu λ_1 ;

e_4 – paraiška baigta aptarnauti srauto bloko antroje fazėje ir ateina į aptarnavimo įrenginį su intensyvumu λ_2 .

e_5 – paraiška baigta aptarnauti ir palieka sistemą su intensyvumu μ .

Sistemos būsenų aibė:

$N = \{(n_1, n_2, n_3)\}, n_1 = \overline{0,2}; n_2 = \overline{0,L}; n_3 = 0,1.$

n_1 – parodo kurioje srauto bloko fazėje yra atvykstanti paraiška (0 – jei srauto blokas tuščias);

n_2 – paraiškų skaičius sistemoje (0 – jei sistema tuščia) ;

n_3 – paraiškų skaičius aptarnavimo įrenginyje (0 – jei aptarnavimo blokas tuščias).

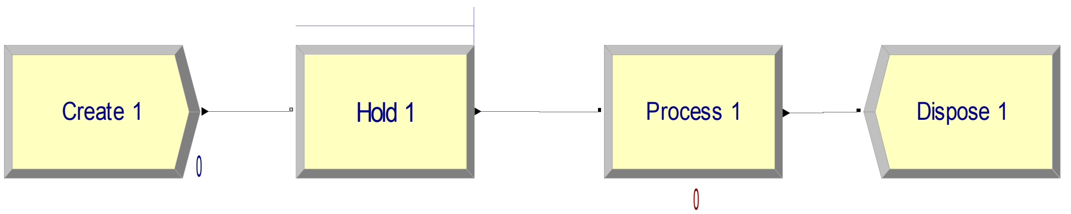
Sistemos aprašymo įvykių kalba pseudokodas:

e1: if $n_1 = 0$ then $n_1 \leftarrow 1$ end if Return Intens $\leftarrow C$	e2: if $n_1 = 1$ then $n_1 \leftarrow 2$ end if Return Intens $\leftarrow p \cdot \lambda_1$
e3: if $n_1 = 1$ if $n_3 = 0$ and $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$; $n_3 \leftarrow 1$ else if $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$ end if Return Intens $\leftarrow (1 - p) \cdot \lambda_1$	e4: if $n_1 = 2$ if $n_3 = 0$ and $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$; $n_3 \leftarrow 1$ else if $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$ end if Return Intens $\leftarrow \lambda_2$
e5: if $n_3 = 1$ if $n_2 > 1$ then $n_2 \leftarrow n_2 - 1$; $n_3 \leftarrow 1$ else $n_2 \leftarrow 0$; $n_3 \leftarrow 0$ end if end if Return Intens $\leftarrow \mu$	

2.3 Aptarnavimo sistemų imitacinis modeliavimas

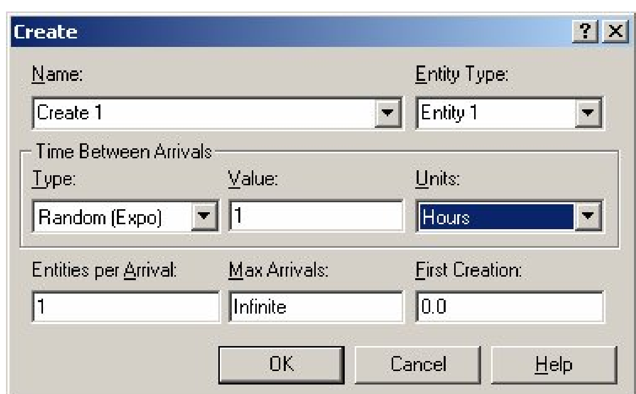
2.3.1 Imitacinio modeliavimo programa ARENA

Imitaciniam modeliavimui naudojome kompanijos „Rockwell“ sukurtą imitacinio modeliavimo sistemą ARENA 9.0. Programa ARENA 9.0 galima atlikti sudėtingų realaus pasaulio sistemų modeliavimą. Paparastų, analiziškai ištiriamų, aptarnavimo sistemų modeliavimas yra nesudėtingas. Pateikiame paprastos sistemos su eile modelio schemą:



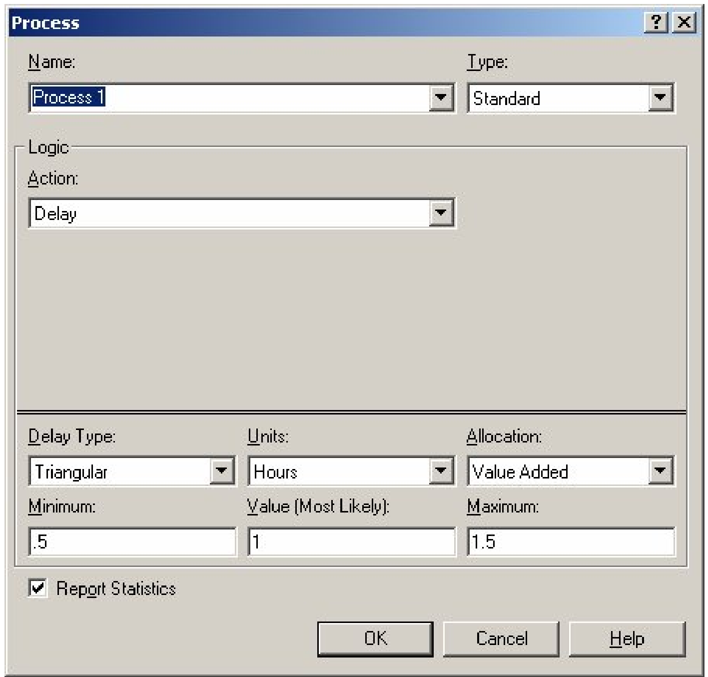
2.6 pav. Aptarnavimo sistemos imitacinio modelio schema

Programos ARENA komponentas „Create“ generuoja paraiškų srautą. Vartotojas gali pasirinkti įvairius srauto parametrus: srauto skirstinį, laiko vienetus, maksimalų paraiškų sukūrimo skaičių ir pan.



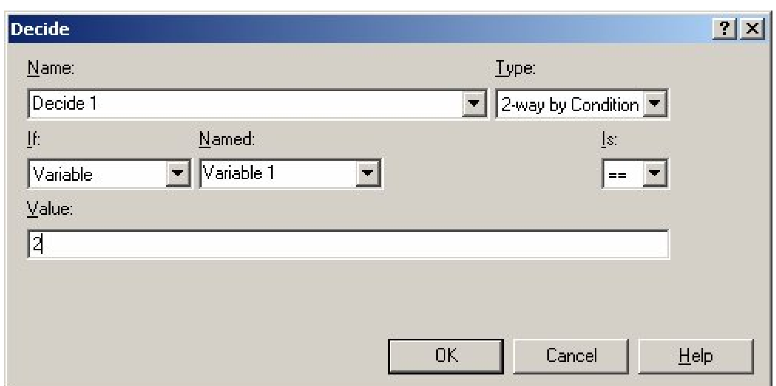
2.7 pav. Programos ARENA paraiškų srauto komponento “Create” aprašymo langas

Blokas „Hold“ suformuoja paraiškų eilę. Eilės charakteristikas (vidutinį paraiškų skaičių eilėje ir kt.) vartotojas randa modeliavimo ataskaitoje („Reports“) pasibaigus modeliavimo sesijai. Blokas „Process“ imituoja aptarnavimo įrenginio veiklą. Vartotojas gali nurodyti įrenginio režimą, aptarnavimo skirstinį, laiko vienetus ir pan.



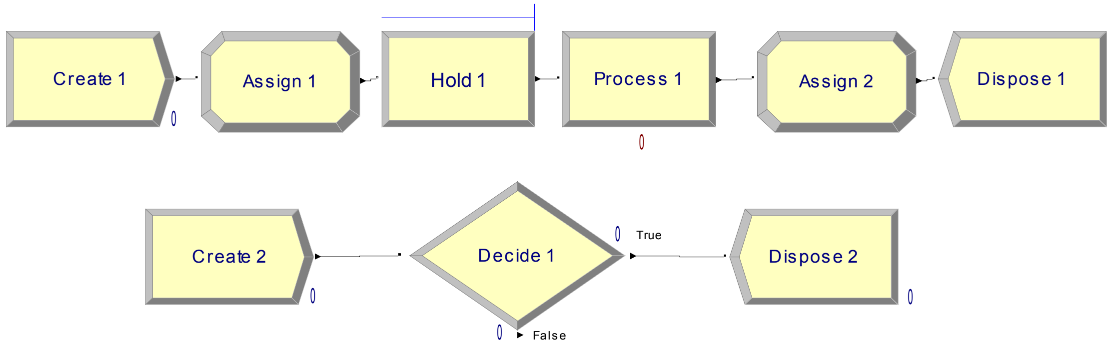
2.8 pav. Programos ARENA komponento “Process” aprašymo langas

Sistemos būsenos tikimybėms apskaičiuoti naudojome „Decide“ bloką. Vartotojas gali nurodyti sąlygas, kuria kryptimi judės paraiška. Prie „Decide“ bloko esantis skaitiklis fiksuoja rezultatus. Pvz., bloke „Asign“ sukūrus kintamąjį „Variable 1“, fiksuojantį paraiškų skaičių sistemoje, taip aprašytas „Decide“ blokas padės apskaičiuoti tikimybę a_2.



2.9 pav. Programos ARENA komponento “Decide” aprašymo langas

Jei „Decide“ blokas sujungtas su aptarnavimo įrenginiu, modelis apskaičiuos tikimybes a_n. Norint apskaičiuoti tikimybes p_n, reikia tame pačiame modelyje sukurti atskirą srautą (žr. 2.10 pav.)



2.10 pav. Aptarnavimo sistemos imitacinio modelio schema su būsenos tikimybių skaičiavimu

2.3.2 Imitacinio ir skaitmeninio modeliavimo rezultatų palyginimas

Palyginsime skaitmeninio (t.y., naudojant įvykių kalbą) ir imitacinio modeliavimo rezultatus. Skaitmeniniam modeliavimui naudojome anksčiau aprašytus sitemų modeliavimo algoritmus. Imitacinis modeliavimas atliktas programa ARENA. Kiekvienam atskiram atvejui buvo sugeneruota 1000000 paraiškų. Palyginimui pateikiame geriausią kiekvieno skirstinio skaitmeninio modeliavimo aproksimaciją (t.y., geriausia sutampančią su teorine aproksimacija). Priminsime, kad skaitmeninio modeliavimo rezulatai sutampa su teoriškai apskaičiuotomis aproksimuojamų sistemų charakteristikomis.

2.3.2.1 M/G/1 sistemų skaitmeninio ir imitacinio modeliavimo palyginimas

Modeliuojant M/G/1 sistemas gauti rezultatai panašūs į teoriškai apskaičiuotas reikšmes. Taigi, darome prielaidą, kad teoriškai apskaičiuotos charakteristikų reikšmės yra teisingos, todėl skaitmeninio ir imitacinio modeliavimo rezultatai turi paklaidas. Rezultatus pateikiame 2.9 lentelėje.

2.9 lentelė

Sistemų M/G/1 skaitmeninio ir imitacinio modeliavimo palyginimas

	p_0	p_1	p_2	p_3	p_4	p_5	$E(L)$	$E(L^q)$
Lognormalusis	**0.0906**	**0.0851**	**0.0731**	**0.064**	**0.0571**	**0.0515**	**11.1653**	**10.2559**
Erlango (3 mom.) Santykinė pakl.	0.0906 0.00%	0.0808 -5.05%	0.0723 -1.09%	0.0649 1.41%	0.0585 2.45%	0.0529 2.72%	11.1653 0.00%	10.2559 0.00%
Imitacinis model. Santykinė pakl.	0.0874 -3.53%	0.0820 -3.64%	0.0707 -3.24%	0.0616 -3.76%	0.0553 -3.19%	0.0499 -3.19%	11.6424 4.27%	10.7297 4.63%

Tolygusis	**0.3000**	**0.2797**	**0.1821**	**0.1057**	**0.0591**	**0.0327**	**1.6556**	**0.9556**
Erlango (2 mom.) Santykinė pakl.	0.3000 0.00%	0.2794 -0.11%	0.1826 0.27%	0.1056 -0.09%	0.0590 -0.17%	0.0327 0.00%	1.6556 0.00%	0.9556 0.00%
Imitacinis model. Santykinė pakl.	0.3004 0.13%	0.2794 -0.11%	0.18256 0.25%	0.1055 -0.19%	0.05908 -0.03%	0.03314 1.33%	1.6523 -0.20%	0.9528 -0.29%
Gama	**0.0625**	**0.0670**	**0.0640**	**0.0598**	**0.0555**	**0.0514**	**12.6563**	**11.7188**
Kokso (3 mom.) Santykinė pakl.	0.0625 0.00%	0.0669 -0.15%	0.0641 0.16%	0.0598 0.00%	0.0555 0.00%	0.0514 0.00%	12.6563 0.00%	11.7188 0.00%
Imitacinis model. Santykinė pakl.	0.0626 0.16%	0.0674 0.54%	0.0643 0.53%	0.0607 1.45%	0.0566 1.95%	0.0526 2.39%	12.2780 -2.99%	11.3401 -3.23%
Veibulo	**0.1139**	**0.0771**	**0.0649**	**0.0576**	**0.0522**	**0.0477**	**11.7016**	**10.8155**
Erlango (3 mom.) Santykinė pakl.	0.1139 0.00%	0.0800 3.76%	0.0648 -0.15%	0.0566 -1.74%	0.0512 -1.92%	0.0470 -1.47%	11.7016 0.00%	10.8155 0.00%
Imitacinis model. Santykinė pakl.	0.1123 -1.37%	0.0764 -0.94%	0.0648 -0.13%	0.0579 0.58%	0.0520 -0.47%	0.0474 -0.61%	11.6109 -0.78%	10.7230 -0.86%

Galime pastebėti, kad įvertinant tikimybes p_n nėra labai ryškių kokybinių skirtumų. Abiem atvejais paklaidos praktiškai neviršija 5 %, o pvz. Tolygaus skirstinio atveju netgi 0.5 %. Įvertinant vidutines sistemos charakteristikas $E(L)$ ir $E(L^q)$ išryškėja skaitmeninio modeliavimo pranašumas. Visais atvejais skaitmeninio modeliavimo rezultatai tiksliai sutampa su teorinėm reikšmėm, tuo tarpu naudojant imitacinį modeliavimą nepavyksta išvengti nedidelių paklaidų.

2.3.2.2 G/M/1 sistemų skaitmeninio ir imitacinio modeliavimo palyginimas

Atlikdami teorinę G/M/1 sistemų analizę, pastebėjome, kad lognormaliojo skirstinio atveju charakteristikų reikšmės smarkiai skiriasi skaičiuojant tiesiogiai ir atliekant aproksimaciją. Atlikus imitacinį sistemos G/M/1 modeliavimą, gauti rezultatai panašūs į aproksimuotoms sistemoms apskaičiuotas reikšmes. Turint omenyje, kad pakankamai didelio paraiškų skaičiaus (mūsų atveju 1000000)modeliavimo sesija duoda rezutatus, artimus tikriesiems, galime daryti išvadą, kad teoriškai apskaičiuotos sistemos G/M/1 reikšmės lognormaliajam skirstiniui nėra tikslios. Rezultatus pateikiame lentelėje:

2.10 lentelė

Sistemų G/M/1 skaitmeninio ir imitacinio modeliavimo palyginimas

	a_0	a_1	a_2	a_3	a_4	a_5	$E(L)$
Lognormalusis	0.4562	0.2481	0.1349	0.0734	0.0399	0.0217	1.6072
Erlango (2 mom.)	0.2555	0.1902	0.1416	0.1054	0.0785	0.0584	2.8697
Santykinis skirt.	-43.99%	-23.34%	4.97%	43.60%	96.74%	169.12%	78.55%
Imitacinis model.	0.2618	0.1925	0.1430	0.1053	0.0784	0.0578	2.7973
Santykinis skirt.	-42.61%	-22.41%	6.04%	43.52%	96.61%	166.48%	74.05%
Tolygusis	0.0804	0.0739	0.0680	0.0625	0.0575	0.0529	11.8525
Erlango (3 mom.)	0.0800	0.0739	0.0679	0.0625	0.0575	0.0529	11.8549
Santykinė pakl.	-0.50%	0.00%	-0.15%	0.00%	0.00%	0.00%	0.02%
Imitacinis model.	0.0800	0.0731	0.0674	0.0625	0.0573	0.0531	11.6692
Santykinė pakl.	-0.55%	-1.14%	-0.88%	0.04%	-0.32%	0.45%	-1.55%
Gama	0.3389	0.2240	0.1481	0.0979	0.0647	0.0428	2.0983
Kokso (3 mom.)	0.3387	0.2240	0.1481	0.0979	0.0648	0.0428	2.0992
Santykinė pakl.	-0.06%	0.00%	0.00%	0.00%	0.15%	0.00%	0.04%
Imitacinis model.	0.3401	0.2244	0.1481	0.0974	0.0646	0.0426	2.0929
Santykinė pakl.	0.36%	0.17%	-0.02%	-0.54%	-0.11%	-0.48%	-0.26%
Veibulo	0.1680	0.1398	0.1163	0.0968	0.0805	0.0670	4.4780
Erlango (3 mom.)	0.1695	0.1408	0.1169	0.0971	0.0806	0.0670	4.4378
Santykinė pakl.	0.89%	0.72%	0.52%	0.31%	0.12%	0.00%	-0.90%
Imitacinis model.	0.1675	0.1391	0.1161	0.0969	0.0801	0.0664	4.4835
Santykinė pakl.	-0.32%	-0.53%	-0.14%	0.12%	-0.54%	-0.92%	0.12%

Atmetus lognormaliojo skirstinio duomenis, galime pastebėti panšią tendenciją kaip ir M/G/1 sistemoms. Sistemos būsenų a_n tikimybių įvertinimas gana tikslus tiek atliekant skaitmeninį, tiek ir imitacinį modeliavimą. Šiuo atveju paklaidos dažnai neviršija netgi 1%. Tuo tarpu vidutinių sistemos charakteristikų nei vienu atveju nepavyksta apskaičiuoti absoliučiai tiksliai, tačiau paklaidos taip pat nedidelės.

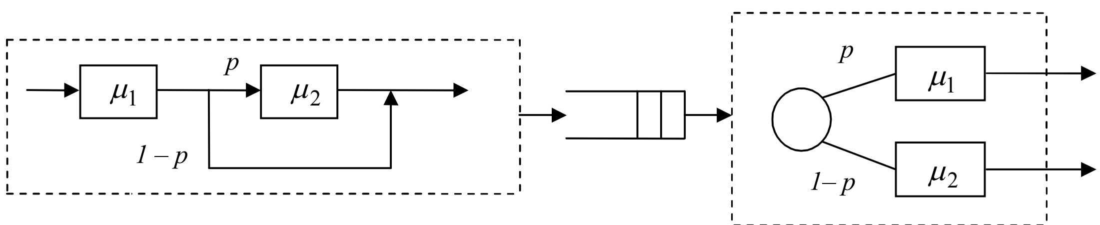
2.4 Sistemos G/G/1 tyrimas

Sistemos G/G/1, nors iš pirmo žvilgsnio ir paprastos, gali būti laikomos sudėtingomis, nes bendru atveju jos neturi analitinių sprendimo metodų. Pabandysime mūsų nagrinėjamą aproksimavimo metodą pritaikyti tokiai sistemai. Kaip bendruosius skirstinius G imsime mūsų anksčiau nagrinėtus skirstinius. Paraiškų atėjimo laikas pasiskirstęs pagal gama skirstinį su parametrais $\alpha = 1.5$ ir $\lambda = 1.6$. Paraiškų aptarnavimo laikas pasiskirstęs pagal lognormalųjį dėsnį su parametrais $m = -0.9$ ir $\sigma = 0.9$.

Gama skirstinį aproksimuojam Kokso skirstiniu, sulygindami 3 pradinius momentus. Gauti Kokso skirstinio parametai: p = 0.7075; $\mu_1 = 1.4587$; $\mu_2 = 2.808$.

Lognormalųjį skirstinį aproksimuojam antrojo tipo Erlango mišiniu (t.y., sulyginam 3 pradinius momentus). Erlango mišinio parametrai: n = 1; p = 0.0069; $\mu_1 = 0.3134$; $\mu_2 = 1.69$.

Nagrinėjamos sistemos fazinė schema:



2.11 Sistemos G/G/1 aproksimavimo schema

Atliksime sistemos formalizavimą:

Sistemos būsenų aibė:

$N = \{(n_1, n_2, n_3)\}, n_1 = \overline{0,2}; n_2 = \overline{0,L}; n_3 = \overline{0,3}.$

n_1 – parodo kurioje srauto bloko fazėje yra atvykstanti paraiška (0 – jei srauto blokas tuščias);

n_2 – paraiškų skaičius sistemoje (0 – jei sistema tuščia) ;

n_3 – paraiškų skaičius aptarnavimo įrenginyje (0 – jei aptarnavimo blokas tuščias).

Sistemos įvykių aibė $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$

e_1 – paraiška atėjo į srauto bloką, pirmąją fazę ;

e_2 – paraiška perėjo į srauto bloko antrą fazę su tikimybe p ir intensyvumu λ_1 ;

e_3 – paraiška baigta aptarnauti pirmoje srauto bloko fazėje ir ateina į aptarnavimo bloko pirmą fazę su tikimybe $1-p$ ir intensyvumu λ_1 ;

e_4 – paraiška baigta aptarnauti srauto bloko antroje fazėje ir ateina į aptarnavimo bloko pirmą fazę su intensyvumu λ_2.

e_5 – paraiška perėjo į aptarnavimo bloko antrą fazę su tikimybe p;

e_6 – paraiška perėjo į aptarnavimo bloko trečią fazę su tikimybe $1-p$;

e_7 – paraiška baigta aptarnauti antroje aptarnavimo bloko fazėje ir palieka sistemą su intensyvumu μ_1.

e_8 – paraiška baigta aptarnauti trečioje aptarnavimo bloko fazėje ir palieka sistemą su intensyvumu μ_2.

Sisemos aprašymo įvykių kalba pseudokodas:

e1: if $n_1 = 0$ then $n_1 \leftarrow 1$ end if Return Intens $\leftarrow C$	e2: if $n_1 = 1$ then $n_1 \leftarrow 2$ end if Return Intens $\leftarrow p \cdot \lambda_1$
e3: if $n_1 = 1$ if $n_3 = 0$ and $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$; $n_3 \leftarrow 1$ else if $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$ end if Return Intens $\leftarrow (1-p) \cdot \lambda_1$	e4: if $n_1 = 2$ if $n_3 = 0$ and $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$; $n_3 \leftarrow 1$ else if $n_2 < L$ then $n_2 \leftarrow n_2 + 1$; $n_1 \leftarrow 0$ end if Return Intens $\leftarrow \lambda_2$
e4: if $n_2 = 1$ then $n_2 \leftarrow 2$ end if Return Intens $\leftarrow p \cdot C$	e5: if $n_2 = 1$ then $n_2 \leftarrow 3$ end if Return Intens $\leftarrow (1-p) \cdot C$

e7:	e8:

if $n_1 = 2$
    if $n_2 > 1$
        then $n_1 \leftarrow n_1 - 1;\ n_2 \leftarrow 1;$
        else $n_2 \leftarrow n_2 + 1;\ n_2 \leftarrow 0$
    end if
end if
Return Intens $\leftarrow \mu_1$
	if $n_1 = 3$
	    if $n_2 > 1$
	        then $n_1 \leftarrow n_1 - 1;\ n_2 \leftarrow 1;$
	        else $n_2 \leftarrow n_2 + 1;\ n_2 \leftarrow 0$
	    end if
	end if
	Return Intens $\leftarrow \mu_2$

Atlikdami skaitmeninį modeliavimą įvertinsime vidutinį eilės ilgį $E(L^q)$ ir ir tikimybes p_n ($n = \overline{0,5};$).

Kadangi analizinių metodu sistemos G/G/1 charakteristikoms apskaičiuoti nėra, patikrinimui naudosime imitacinį modeliavimą (modeliavimo sesijos metu sugeneruosim 5000000 atvykstančių paraiškų). Kompiuterinio ir imitacinio modeliavimo rezultatus surašome į lentelę:

2.11 lentelė

Sistemos G/G/1 skaitmeninio ir imitacinio modeliavimo palyginimas

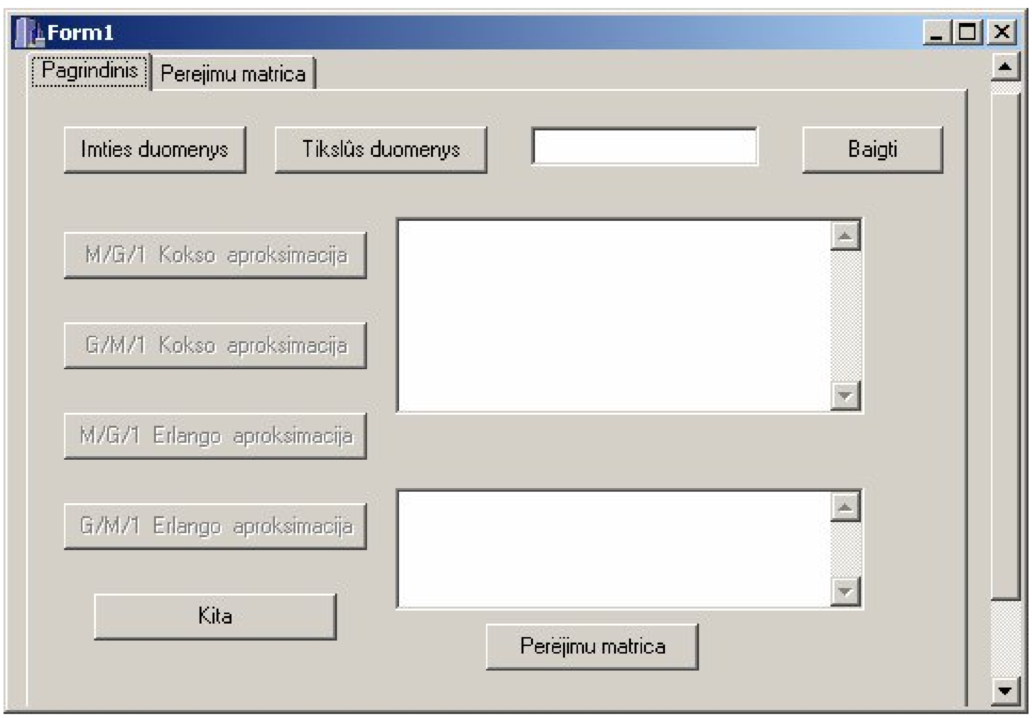
	p_0	p_1	p_2	p_3	p_4	p_5	$E(L^q)$
Skaitmeninis modeliavimas	0.3498	0.2597	0.1515	0.0895	0.0536	0.0328	1.0941
Imitacinis modeliavimas	0.3499	0.2692	0.1452	0.0839	0.0517	0.0328	1.0948
Santykinis skirtumas (proc.)	0.03%	3.66%	-4.16%	-6.26%	-3.54%	0.00%	0.06%

Kaip matome, tarp imitacinio ir įvykių kalba aprašyto kompiuterinio modeliavimo rezultatų skirtumas nedidelis. Turint omenyje, kad esant dideliam modeliuojamų paraiškų skaičiui imitacinio modeliavimo rezultatai gana artimi realiems duomenims, galim sakyti, kad mūsų nagrinėjamas metodas taip pat tinka sistemų G/G/1 modeliavimui.

3. PROGRAMINĖ REALIZACIJA

Magistro darbe buvo sukurta universali programinė įranga, skirta modeliuoti aptarnavimo sistemas. Programa koduota „C++ Builder“ programavimo kalba. Paprastų aptarnavimo sistemų M/G/1 ir G/M/1 modeliavimas automatizuotas – t.y., vartotojas gali perduoti programai imties duomenis (gautus stebint atsitiktinį dydį, turintį skirstinį G) ar skirstinio G pradinius momentus, ir programa pateikia aptarnavimo sistemos charakteristikas. Sudėtingesnių sistemų aprašymą vartotojas gali pateikti pats. Sistema aprašoma įvykių kalba, kurios pavyzdžių galima rasti [1] ar šiame magistriniame darbe. Sistemos veikimo principai aprašyti [1]. Nuskaičiusi parametrus programa modeliuoja būsenų aibę kartu su perėjimų matrica. Sugeneravusi sistemos modelio būsenas, programa suskaičiuoja stacionarias tikimybes. Statistinių charakteristikų skaičiavimo metodai gali skirtis įvairioms sistemoms. Sistemų M/G/1 ir G/M/1 charakteristikų skaičiavimas pateiktas šiame darbe. Bendru atveju, kaip ir sistemos aprašymu įvykių kalba, jų sukūrimu turi rūpintis vartotojas.

Paleidus programą vartotojas mato tokį programos langą:



3.1 Programos langas

Mygtukų paaiškinimai:

„Imties duomenys“ – nuskaitomi duomenys, gauti stebint tam tikrą atsitiktinį dydį. Duomenų failas „Imties duomenys.txt“ saugomas programos kataloge „Duomenys“. Duomenys – realūs teigiami skaičiai. Sėkmingai nuskaičiusi duomenis programa apskaičiuoja empirinius pradinius momentus ir įvertiną galimą skirstinių aproksimavimą.

„Tikslus duomenys“ – nuskaitomi aproksimuojamo skirstinio 3 pradiniai momentai . Duomenų failas „Tikslus duomenys“ saugomas programos kataloge „Tikslus duomenys“. Sėkmingai nuskaičiusi duomenis programa įvertiną galimą skirstinių aproksimavimą.

„M/G/1 Kokso aproksimacija“ – jei nuskaityti duomenys tenkina aproksimavimo sąlygas (1.), programa apskaičiuoja ir išveda į ekrana sistemos M/G/1 charakteristikas, kai skirstinys G aproksimuotas Kokso skirstiniu. Sistemos

„G/M/1 Kokso aproksimacija“ – jei nuskaityti duomenys tenkina aproksimavimo sąlygas (1.), programa apskaičiuoja ir išveda į ekrana sistemos G/M/1 charakteristikas, kai skirstinys G aproksimuotas Kokso skirstiniu.

M/G/1 Erlango aproksimacija“ – programa apskaičiuoja ir išveda į ekrana sistemos M/G/1 charakteristikas, kai skirstinys G aproksimuotas Erlango mišiniu.

„G/M/1 Erlango aproksimacija“ – programa apskaičiuoja ir išveda į ekrana sistemos G/M/1 charakteristikas, kai skirstinys G aproksimuotas Erlango mišiniu.

„Kita“ – apskaičiuoja vartotojo aprašytos sistemos charakteristikas. Šiuo atveju, vartotojas pats aprašo reikiamą sistemą, sistemos charakteristikų apskaičiavimą, nurodo duomenų failą.

„Perėjimų matrica“ – sudaro aprašytos sistemos būsenų perėjimo matricą ir išvedą į ekraną (programos lango puslapyje „Perėjimų matrica “).

„Baigti“ – baigia programos darbą ir uždaro programos langą.

Imties duomenų faile „Imties duomenys.txt“ duomenys – realūs teigiami skaičiai, surašyti po vieną eilutėje, be jokių papildomų skyriklių.

Duomenų faile „Tikslus duomenys“ turi būti trys teigiami skaičiai – konkretaus skirstinio pradiniai momentai. Skaičiai surašyti po vieną eilutėje, be jokių papildomų skyriklių.

Sistemų M/G/1 ir G/M/1 duomenų failai saugomi programos kataloge “Duomenys” ir vartotojas jų neturėtų koreguoti. Tuo tarpu modeliuojant kitokio tipo sistemas vartotojas privalo pats sukurti reikiamus duomenų failus. Pateikiame šių failų struktūrą:

1) Pirmoje duomenų failo eilutėje turi būti įvesta pradinė sistemos būsena. Ji turi būti rašoma po „=“ ir „ “. gale turi būti „;“ simbolis. Pvz.

pradine busena= 0 0 0;

2) Antroje duomenų failo eilutėje turi būti aptarnaujančių įrenginių intensyvumai. Pradžioje eilutės padedami „=“ ir „ “, o gale – „;“. Pvz.

intensyvumai= 0.75 1 2

3) Trečioje eilutėje įvedamas paraiškų srautų skaičius. Pradžioje eilutės padedami „=“ ir „ “, o gale – „;“. Pvz.

srautu skaicius= 1;

4) Ketvirtoje eilutėje įvedamas sistemos įvykių skaičius. Pradžioje eilutės padedami „=“ ir „ “, o gale – „;“. Pvz.

ivykiu skaicius= 4;

Jei vartotojas nori modeliuoti savo pasirinktą (ne M/G/1 ar G/M/1) aptarnavimo sistemą, jos aprašymą reikia pateikti formoje *TForm1* esančiame metode *Busenos(int *, int)*. Sistema modeliavimui, turi būti pateikiama *C++ Builder* kalboje naudojamu *switch-case* sakiniu 3.2 pav .

```
switch (poz)
 {
  case 0:           //tikrinam ivyki e{1}
  {
   if(xx[0]+xx[1]<L)
   {
    xx[1]++;
    radom=true;
   }
   else
   {
    xx[0]++;
    radom=true;
   }
   return Intens0;
  }    //ivykio e{1} aprasymo pabaiga
...
...
  case n:           //tikrinam ivyki e{n+1}
  {
   if(...)
   {
    ...;
    radom=true;
   }
   else
   {
    ...;
    radom=true;
   }
   return ...;
  }
 }   //ivykio e{n+1} aprasymo pabaiga
}   //metodo pabaiga
```

3.2 pav. Sistemos aprašymas *Switch-Case* sakiniu

Čia `xx[i]` yra tiriamos būsenos vektorius kur `xx[i]` yra (i +1) -oji vektoriaus koordinatė. `Intens0` yra perėjimo intensyvumas iš atitinkamos būsenos. Modeliuojant aptarnavimo sistemas, intensyvumai gali būti įvedami pradinių duomenų byloje, arba nurodomi tiesiogiai aprašant sistemą. Atlikus būsenos koordinatės keitimą, reikalinga pažymėti, kad nauja būsena rasta, sakiniu „`radom=true;`”. *Case* sakinio pabaigoje reikia gražinti reikiamą perėjimo į naujai rastą būseną intensyvumą, sakiniu „`return Intens;`”

IŠVADOS

- M/G/1 sistemų aproksimavimui vidutinėms charakteristikoms įvertinti pakanka sulyginti 2 pradinius momentus. Vidutinės sistemų charakteristikos naudojant aproksimavimą sutampa su analizinėm reikšmėm. Norint tiksliau įvertinti sistemos būsenų tikimybes p_n rekomenduotume naudoti 3 momentų sulyginimą.
- G/M/1 sistemų aproksimavimas sunkiau įvertinamas, tačiau norint tiksliau apskaičiuoti charakteristikas tikslinga naudoti 3 momentų sulyginimą. Skirtingai nuo M/G/1 sistemų, vidutinės charakteristikos neapskaičiuojamos tiksliai, o kartais jų tikslumas netgi mažesnis negu sistemos būsenų tikimybių a_n.
- Tiek tiriant M/G/1, tiek ir G/M/1 sistemas įvairiems skirstiniams geriausios aproksimacijos didžiausios paklaidos neviršijo 5 %, o kai kuriais atvejais ir 0.5 %.
- Tolygaus skirstinio aproksimavimui netinka Kokso skirstinys.
- Modeliuojant sistemas kompiuterine įvykių kalbą, sistemų charakteristikos apskaičiuojamos tiksliai (naudojant ekponentinius mišinius). Sistemų aproksimavimui taikant Erlango mišinius reikia įvesti fiktyvią fazę su labai dideliu (palyginus su kitomis fazėmis) intensyvumu.
- Sistemoms G/M/1 naudojant šiame darbe sukurtus sistemų modeliavimo algoritmus galima apskaičiuoti ne tik tikimybes a_n (netiesiogiai), bet ir p_n (šių tikimybių apskaičiavimas nepateikiamas klasikinėje aptarnavimo sistemų teorijos literatūroje).
- Sistemos G/G/1 tyrimas parodo, kad kompiuterinė įvykių kalba tinka naudoti net ir aptarnavimo sistemoms , kurios neaprašomos analiziniais metodais.

PADĖKOS

Norėčiau padėkoti darbo vadovui doc. dr. E. Valakevičiui už pasiūlytą įdomią magistrinio darbo temą bei pagalbą ir konsultacijas atliekant darbe iškeltas užduotis.

Taip pat noriu padėkoti visiems draugams, davusiems naudingų patarimų programavimo „C++ Builder“ klausimais .

LITERATŪROS SĄRAŠAS

1. Johnson A. M., Taaffe R. M. Matching Moments to Phase Distributions: Mixture of Erlang Distribution of Common Order, *School of Industrial Engineering, Purdue University*
2. Khonomenko A. D., Bubnov V.P. A Use of Coxian Distribution Law for Iterative Solution of $M/G/n/R \leq \infty$ Queueing systems, *Problems of Control and Information Theory 14 (2), 1985, p. 143-153*
3. Osogami T., Harchol-Balter M. Necessary and Sufficient Conditions for Representing General Distributions by Coxians, *School of Computer Science, Carnegie Mellon University, 2003*
4. Adan I., Resing J. Queueing Theory. 2001, *Department of Mathematicsand Computing Science, Eindhoven University of Technology.*
5. Pranevičius, H., Valakevičius, E. Numerical Models of Systems Specified by Marcovian processes. Kaunas, 1996.
6. Whitt, W. On Approximations for Queues, Part III: Mixtures of Exponential Distributions, *AT&T ell Labs Tech Journal*, 63:1, 1984, p. 163-175.
7. Johnson A. M., Luhman A. J. Behaviour of Queueing Approximations Based on Sample Moments.
8. Pranevičius, H., Pranevičienė, I. Masinio aptarnavimo teorijos elementai. Vilnius, 1980.
9. Yao, D.D.W., Buzacott, J.A. Queueing models for a flexible machining station, Part II, *European journal of operational research,* vol 19, 1985, p.242-251.
10. Mickevičius, G., Valakevičius, E., Modelling of non-Markovian queuing systems, *Technological and economic development of economy*, Vol XII, No 4, 2006, p. 295-300.
11. http://en.wikipedia.org/wiki/Phase-type_distribution

PRIEDAI

Programos tekstas

"Unit1.h"

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

#ifndef Unit1H
#define Unit1H
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <Dialogs.hpp>
#include <Grids.hpp>
struct st
{
float BInt;
int eil;
int st;
};
struct TBusIntens    //uzkoduota perejimu matrica
{
 st BI;
 TBusIntens *kitas;
};
struct TBus
{
 int *sk;
 TBus *kitas;
};
struct TBusA    //~=TBus tik skirtas realiems skaiciams
{
 double *sk;
 TBusA *kitas;
};
```

```
//---------------------------------------------------------------------------
class TForm1 : public TForm
{
__published:	// IDE-managed Components
	TPageControl *PageControl1;
	TTabSheet *TabSheet1;
	TTabSheet *TabSheet2;
	TButton *Button1;
	TButton *Button2;
	TButton *Button3;
	TButton *Button4;
	TButton *Button5;
	TButton *Button6;
	TButton *Button7;
	TEdit *Edit1;
	TMemo *Memo1;
	TMemo *Memo2;
	TOpenDialog *OpenDialog1;
	TStringGrid *StringGrid1;
	TButton *Button8;
	TButton *Button9;
	TButton *Button10;
	void __fastcall FormCreate(TObject *Sender);
	void __fastcall Button7Click(TObject *Sender);
	void __fastcall Button4Click(TObject *Sender);
	void __fastcall Button3Click(TObject *Sender);
	void __fastcall Button8Click(TObject *Sender);
	void __fastcall Button9Click(TObject *Sender);
	void __fastcall Button1Click(TObject *Sender);
	void __fastcall Button2Click(TObject *Sender);
	//void __fastcall PageControl1Change(TObject *Sender);
private:  // User declarations
float *Tintens;
	TBusIntens *TPr; //perejimu tikimybiu koduotas dinaminis sarasas
	TBus *TBPr;
	float *r;      //normuotos stac tikimybes
	float *r_apj;  //apjungtos stac tikimybes, nes vienai busenai dvi fazes
	float m1,m2,m3,c,c2,asim; // momentai
	float kok2_miu1,kok2_miu2,kok2_p;
	float kok3_miu1,kok3_miu2,kok3_p;
	float erl2_miu,erl2_p;
	float erl3_miu1,erl3_miu2,erl3_p;
	int erl2_n,erl3_n;
```

```
float *y;
int n;
int L;        //max eiles ilgis
int M;        //ivykiu skaicius
int R;          //srautu skaicius
int K;       //kasu skaicius
int tipas;
bool radom;     //ar rasta nauja busena;
int IK;         //intensyvumu skaicius
int N;        //intervalu skaicius
TBusA *TBAPr;           //nereik?
void SuskaidytiIntervalais(double);
bool ErgodiskumoPatikrinimas();
void Normavimas();
void Grafika();
bool PatikrinimasVienatiskumoA(double [], int, double);
void SurastiFaziuIntens();
void PerejimuTikimybes();
void IvestiPradSalygas();
void IsvestiMemo();
void IsvestiStrGrid();
void BusenuRadimas(int tipas);
void Isvedimas(int tipas);
void IvestiPradSalygasKoksoMG1();
void IvestiPradSalygasErlangoMG1();
void IvestiPradSalygasKoksoGM1();
void IvestiPradSalygasErlangoGM1();
float Busenos(int*,int);
float BusenosKoksoMG1(int *xx, int poz);
float BusenosErlangoMG1(int *xx, int poz);
float BusenosKoksoGM1(int *xx, int poz);
float BusenosErlangoGM1(int *xx, int poz);
void KasuSk(AnsiString);
bool PatikrinimasVienatiskumo(int [], int, double);
void Apjungimas();
void StacTikRadimas();
void SurastiMax();
void Statistika_S5();
void Statistika_S4();
void Statistika_S3();
void Statistika_S2();
void Statistika_S1();
```

```
        void StatistikaKoksoGM1();
        void Kokso2();
        void Kokso3();
        void Erlango2();
        void Erlango3();
        void Momentai();

public:     // User declarations
        __fastcall TForm1(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm1 *Form1;
//---------------------------------------------------------------------------
#endif
```

"Unit1.cpp"

```
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::FormCreate(TObject *Sender)
{
 TPr=NULL;
 TBPr=NULL;
 Button3->Enabled = false;
 Button4->Enabled = false;
 Button5->Enabled = false;
 Button6->Enabled = false;
 //PageControl1->Pages[0]->Show();
 PageControl1->Pages[0]->Caption="Pagrindinis";
 PageControl1->Pages[1]->Caption="Perejimu matrica";

}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button7Click(TObject *Sender)
{
```

```
 Close();
}
//------------------------------------------------------------------------------
void TForm1::Normavimas()    //sunormuoja TPr sarasa
{
 TBusIntens *D;
 float suma;    //suma visu iseinanciu is tiriamosios busenos
 int i;
 for (i=0;i<N;i++)
 {
  suma=0;
  D=TPr;
  while (D)
  {
   if (D->BI.eil==i)
    suma=suma+D->BI.BInt;
   D=D->kitas;
  }

  D=TPr;
  while (D)
  {
   if (D->BI.eil==i)
    D->BI.BInt=D->BI.BInt/suma;
   D=D->kitas;
  }
 }
}
//------------------------------------------------------------------------
void TForm1::Apjungimas()    //sudeda pasikartojancius intensyvumus pereinant is
vienos busenos i kita
{
 TBusIntens *D0;
 TBusIntens *D1;
 TBusIntens *D;
 TBusIntens *istr;
 D1=TPr;
 D=TPr;
 int i;
 while (D)
 {
  D0=D;
  D1=D->kitas;
```

```
  while (D1)
  {
   if(D->BI.eil==D1->BI.eil)
    if(D->BI.st==D1->BI.st)
     {
      D->BI.BInt=D->BI.BInt+D1->BI.BInt;
      istr=D1;
      D0->kitas=D1->kitas;
      D1=D1->kitas;
      delete istr;
     }
    else {D1=D1->kitas;D0=D0->kitas;}
  else {D1=D1->kitas;D0=D0->kitas;}
  }
 D=D->kitas;
 }
}
//---------------------------------------------------------------------------
bool TForm1::ErgodiskumoPatikrinimas()    //patikrina ar is bet kurios busenos
galima patekti i bet kuria kita
{
 int i=0;
 bool yra=false;
 TBusIntens *X,*Y;
 int *A=new int[N];
 for (i=0;i<N;i++) A[i]=i;      //susirasom visus intervalus
 for (i=0;i<N;i++)
 {
  yra=false;
  X=TPr;
  while ((X)&&(!yra))
  {
   if ((X->BI.eil==A[i])||(X->BI.st==A[i])) yra=true;
   X=X->kitas;
  }
  if (!yra) A[i]=-2;  //tokios busenos/intervalo nera
 }
 yra=false;
 int k=-1;
 while (k!=0)
 {
  if (k==-1) A[N-1]=-1; //pradedam eiti nuo galo
  else A[0]=-1;         //nuo pradziu
```

```
  while (!yra)
  {
   yra=true;
   for (i=N-1;i>=0;i--)
   {
    if (A[i]==-1)
    {
     X=TPr;
     while (X)
     {
      if (X->BI.eil==i)
       if (A[X->BI.st]!=-1) {A[X->BI.st]=-1; yra=yra*false;}
      X=X->kitas;
     }
    } //if (A[i]==-1) pab
   } //for (i=N-1;i>=0;i--) pab
  } //while (!yra) pab
  k=0;
 }
 yra=true;     //patikrinam ar i visas busenas sugebejom nueiti
 i=0;
 while ((i<N)&&(yra))
 {
  if ((A[i]!=-1)&&(A[i]!=-2)) yra=false;
  i++;
 }
if(yra) return true;  //visos busenos yra griztamos
else return false;
}
//----------------------------------------------------------------------
void TForm1::StacTikRadimas()    //kilti lygiu galima tik su liambda wadinas
{                                //vadinasi visada bus tik vienas atejimas ir
keli isejimai.gal ce ir blogai..
 r=new float[N];
 float *S;         //S yra d elementu sarasas
 S=new float[N];
 int i=N-1;
 float a,b,c,d;              //formules nariai
 TBusIntens *P,*X,*ir;
 int b1,b2;                //busena su kuria trinama busena turi santykiu
 float per;                 //pasikeites perejimu intensyvumas suskaiciuotas su
a,b,c,d
 bool yra;
```

```
 int kzk;                //parodo kelinta "iseinama" intensyvuma pasirinkti
 int kzzk;               //neimam reikiamo intensyvumo kol kzzk!=kzk;
TBusIntens *xx=TPr,*Pp,*Pg;
 Pg=new TBusIntens;         //pirmas elem
 Pg->BI.BInt=xx->BI.BInt;
 Pg->BI.eil=xx->BI.eil;
 Pg->BI.st=xx->BI.st;
 Pg->kitas=NULL;
 P=Pg;
 xx=xx->kitas;

 while (xx)
 {
  Pp=new TBusIntens;
  Pp->BI.BInt=xx->BI.BInt;
  Pp->BI.eil=xx->BI.eil;
  Pp->BI.st=xx->BI.st;
  Pp->kitas=NULL;
  Pg->kitas=Pp;
  Pg=Pg->kitas;
  xx=xx->kitas;
 }
 while (i>0)
 {
  b1=0;
  b2=0;
  d=0;
  X=P;
  kzk=0;
  while (X)    //surandam paskutini formules elementa
  {
    if (X->BI.eil==i)
     if (X->BI.st!=i)
      d=d+X->BI.BInt;
    X=X->kitas;
  }
  while ((b1==0)&&(b2==0))
  {
    b1=-1;
    b2=-1;
    a=0;
    b=0;
    c=0;
```

```
X=P;
while ((X)&&(b1==-1))    //busena is kurios galim patekti i trinama busena
{
  if (X->BI.st==i)
   if (X->BI.eil!=i)   //apsauga nuo rinkiu
    {
     b1=X->BI.eil;
     b=X->BI.BInt;
    }
  X=X->kitas;
}
X=P;
kzzk=0;
while ((X)&&(b2==-1))   //busena i kuria galim patekti is trinama busena
{
  if (X->BI.eil==i)
   if (X->BI.st!=i)
   {
    if (kzzk==kzk)
    {
     b2=X->BI.st;
     c=X->BI.BInt;
    }
    kzzk++;
   }
  X=X->kitas;
}
X=P;
while ((X)&&(a==0))    //surandam pirmaji formules elementa
{
  if ((X->BI.eil==b1)&&(X->BI.st==b2))
  {
   a=X->BI.BInt;
  }
  X=X->kitas;
}
if (d!=0) per=a+b*c/d;
else per=0;

if ((per>0)&&(b2!=-1))
 {
 //patikrinam ar tokio dar ner
  X=P;
```

```
   yra=false;
   while ((X)&&(!yra))
   {
     if (X->BI.eil==b1)
      if (X->BI.st==b2)
      {
       X->BI.BInt=per;
       yra=true;
      }
     X=X->kitas;
   }
 //irasom nauja gauta perejimu intensyvuma
  if (!yra)
  {
   ir=new TBusIntens;
   ir->BI.BInt=per;
   ir->BI.eil=b1;
   ir->BI.st=b2;
   ir->kitas=P;
   P=ir;
  }
   X=TPr;
   yra=false;
   while ((X)&&(!yra))
   {
     if (X->BI.eil==b1)
      if (X->BI.st==b2)
      {
       X->BI.BInt=per;
       yra=true;
      }
     X=X->kitas;
   }
 //irasom nauja gauta perejimu intensyvuma
  if (!yra)
  {
   ir=new TBusIntens;
   ir->BI.BInt=per;
   ir->BI.eil=b1;
   ir->BI.st=b2;
   ir->kitas=TPr;
   TPr=ir;
  }
```

```
    b1=0;    //kad dar karta wykdytu cikla
    b2=0;
    kzk++;
   }// if per>0 pabaiga
   //istrinam ta intensyvuma kuriuo atejom i trinama busena
   if ((b2==-1)&&(b1!=-1))
   {
    X=P;
    yra=false;
    TBusIntens *Y=NULL,*Z;
    while ((X)&&(!yra))
    {
      if (X->BI.st==i)
       if (X->BI.eil!=i)
       {
        if (Y)
        {
         Y->kitas=X->kitas;
         Z=X;
         delete Z;
         yra=true;
        }
       else   //jeigu reik trinti pask elementa
       {
        Z=X;
        X=X->kitas;
        P=P->kitas;
        delete Z;
        yra=true;
       }
      }
      Y=X;
      if (X->kitas) X=X->kitas;
    }
   b1=0;
   b2=0;
   kzk=0;
   } //if b2==-1 pab
   //jeigu neradom jokio "ieinancio" inetnsyvumo istrinam visus "iseinancius"
inetnsyvumus
   if(b1==-1)
   {
    X=P;
```

```
  TBusIntens *Y=NULL,*Z;
  while (X)
  {
   yra=false;
    if (X->BI.eil==i)
    {
     if (Y)
     {
      Y->kitas=X->kitas;
      Z=X;
      delete Z;
      yra=true;
     }
     else   //jeigu reik trinti pask elementa
     {
      Z=X;
      X=X->kitas;
      P=P->kitas;
      delete Z;
      yra=true;
     }
    }
    if (!yra)
    {
     Y=X;
     X=X->kitas;
    }
    else if (Y) X=Y->kitas;
  }
 }     //if b1==-1 pab

}      //while (b1==0)&&(b2==0) pabaiga
S[i]=dl;
i--;
}
//r saraso formavimas
i=1;
int j;
float rr;      //busimas r
r[0]=P->BI.BInt;
while (i<N)
{
 rr=0;
```

```
   X=TPr;

    while (X)
    {
      if (X->BI.st==i)
       if (X->BI.eil<i)    //nepriklauso rinkes
       {
         rr=rr+r[X->BI.eil]*X->BI.BInt/S[i];
       }
      X=X->kitas;
    }
   r[i]=rr;
   i++;
  }

  float suma=0;
  for(j=0;j<N;j++)
   suma=suma+r[j];
  for(j=0;j<N;j++)
   r[j]=r[j]/suma;

 }
 //---------------------------------------------------------------------
 void TForm1::IvestiPradSalygas()
 {
  int i=0;
  AnsiString DF;
  OpenDialog1->Filter="Tekstines bylos (*.txt)|*.txt";
  if (OpenDialog1->Execute() && FileExists(OpenDialog1->FileName))
   {
    DF=OpenDialog1->FileName;
    KasuSk(DF);
   }
  else DF="";
  FILE *F;
  char elem;
  TBus *D;
  D=new TBus;
  D->sk=new int[K];
  Tintens=new float[IK];
   if ((F = fopen(DF.c_str(), "r"))==NULL);
   else
    {
```

```
    i=0;
     while (elem!='=') fscanf(F, "%c", &elem);
     fscanf(F, "%c", &elem);
    while (elem!=';')
    {
     fscanf(F, "%d%c", &D->sk[i], &elem);
     i++;
    }
     elem='a';
     i=0;
      while (elem!='=') fscanf(F, "%c", &elem);
      fscanf(F, "%c", &elem);
     while (elem!=';')
     {
      fscanf(F, "%f%c", &Tintens[i], &elem);
      i++;
     }
      while (elem!='=') fscanf(F, "%c", &elem);
      fscanf(F, "%c", &elem);
     fscanf(F, "%d %c", &R,&elem);      //srautu skaicius

      while (elem!='=') fscanf(F, "%c", &elem);
      fscanf(F, "%c", &elem);
     fscanf(F, "%d %c", &M,&elem);      //ivykiu skaicius
  fclose(F);
  D->kitas=TBPr;
  TBPr=D;
  N=1;                  //busenu sk
  //if (CheckBox3->Checked==true) SurastiFaziuIntens();
  }
 L=StrToInt(Edit1->Text); //nuskaitom max parasku skaiciu sistemoje
 if (L<0)
 {
  L=0;
  ShowMessage("Ivedet neigiama maksimalu paraisku kieki sistemoje");
 }
}
//---------------------------------------------------------------------------
void TForm1::IvestiPradSalygasKoksoMG1()
{

 int i=0;
 AnsiString DF;
```

```
/*
OpenDialog1->Filter="Tekstines bylos (*.txt)|*.txt";
if (OpenDialog1->Execute() && FileExists(OpenDialog1->FileName))
 {
  DF=OpenDialog1->FileName;
  KasuSk(DF);
 }
else DF="";
DF = "Duomenys/KoksoMG1.txt";
KasuSk(DF);
FILE *F;
char elem;
TBus *D;
D=new TBus;
D->sk=new int[K];
Tintens=new float[IK];
 if ((F = fopen(DF.c_str(), "r"))==NULL);
 else
  {
   i=0;

    while (elem!='=') fscanf(F, "%c", &elem);
    fscanf(F, "%c", &elem);
   while (elem!=';')
   {
    fscanf(F, "%d%c", &D->sk[i], &elem);
    i++;
   }
    elem='a';
    i=0;
     while (elem!='=') fscanf(F, "%c", &elem);
     fscanf(F, "%c", &elem);

    while (elem!=';')
    {
     fscanf(F, "%f%c", &Tintens[i], &elem);
     i++;
    }
     while (elem!='=') fscanf(F, "%c", &elem);
     fscanf(F, "%c", &elem);
    fscanf(F, "%d %c", &R,&elem);      //srautu skaicius

     while (elem!='=') fscanf(F, "%c", &elem);
```

```
      fscanf(F, "%c", &elem);
     fscanf(F, "%d %c", &M,&elem);      //ivykiu skaicius
  fclose(F);
  D->kitas=TBPr;
  TBPr=D;
  N=1;                  //busenu sk
  //if (CheckBox3->Checked==true) SurastiFaziuIntens();
  }
 M = 4;
 R = 1;
 K = 2;
 //IK = 5;
 L=StrToIntDef(Edit1->Text,25); //nuskaitom max parasku skaiciu sistemoje
 if (L<0)
 {
  L=25;
  ShowMessage("Ivedet neigiama maksimalu paraisku kieki sistemoje.Pagal
nutylejima programa priskirs 25");
 }
}
//----------------------------------------------------------------------
void TForm1::IvestiPradSalygasKoksoGM1()
{
 int i=0;
 AnsiString DF;
 DF = "Duomenys/KoksoGM1.txt";
 KasuSk(DF);
 FILE *F;
 char elem;
 TBus *D;
 D=new TBus;
 D->sk=new int[K];
 Tintens=new float[IK];
  if ((F = fopen(DF.c_str(), "r"))==NULL);
  else
   {
    i=0;
     while (elem!='=') fscanf(F, "%c", &elem);
     fscanf(F, "%c", &elem);
    while (elem!=';')
    {
     fscanf(F, "%d%c", &D->sk[i], &elem);
     i++;
```

```
    }
     elem='a';
     i=0;
      while (elem!='=') fscanf(F, "%c", &elem);
      fscanf(F, "%c", &elem);
     while (elem!=';')
     {
      fscanf(F, "%f%c", &Tintens[i], &elem);
      i++;
     }
      while (elem!='=') fscanf(F, "%c", &elem);
      fscanf(F, "%c", &elem);
     fscanf(F, "%d %c", &R,&elem);      //srautu skaicius
      while (elem!='=') fscanf(F, "%c", &elem);
      fscanf(F, "%c", &elem);
     fscanf(F, "%d %c", &M,&elem);      //ivykiu skaicius
  fclose(F);
  D->kitas=TBPr;
  TBPr=D;
  N=1;                  //busenu sk
  }
 M = 5;
 R = 1;
 K = 3;
 //IK = 5;
 L=StrToIntDef(Edit1->Text,25); //nuskaitom max parasku skaiciu sistemoje
 if (L<0)
 {
  L=25;
  ShowMessage("Ivedet neigiama maksimalu paraisku kieki sistemoje.Pagal
nutylejima programa priskirs 25");
 }
}
//---------------------------------------------------------------------
void TForm1::IsvestiMemo()
{
 AnsiString x;
 TBus *D;
 D=TBPr;
 Memo1->Lines->Add("Busenu aibe su");
 Memo1->Lines->Add("stacionariomis tikimybemis >0.00005:");
 for(int k=0; k<N; k++)
 {
```

```
  x="[";
  x=x+IntToStr(D->sk[0]);
  for(int j=1; j<K; j++)
  {
   x=x+";"+IntToStr(D->sk[j]);
  }
  x=x+"]";
  if (r[k]>=0.00005)
   Memo1->Lines->Add(IntToStr(k)+") "+x+" - "+FloatToStrF(r[k],ffFixed,6,4));
  D=D->kitas;
 }
Memo2->Lines->Add("INTENSYVUMAI:");
 for(int j=0; j<IK; j++)
 {
  Memo2->Lines->Add(IntToStr(j)+"-as intensyvumas:
"+FloatToStrF(Tintens[j],ffFixed,15,13));
 }
}
//---------------------------------------------------------------------
void TForm1::IsvestiStrGrid()
{
 int i,j;
 TBusIntens *D=TPr;
 StringGrid1->ColCount=N+1;
 StringGrid1->RowCount=N+1;
 for(i=0;i<N;i++)
  StringGrid1->Cells[i+1][0]="B"+IntToStr(i);
 for(i=0;i<N;i++)
  StringGrid1->Cells[0][i+1]="B"+IntToStr(i);
 for(i=1;i<=N;i++)
  for(j=1;j<=N;j++)
   StringGrid1->Cells[i][j]=0;
 float x;
 while (D)
  {
   x=StrToFloat(StringGrid1->Cells[D->BI.st+1][D->BI.eil+1]);
   x=x+D->BI.BInt;
   StringGrid1->Cells[D->BI.st+1][D->BI.eil+1]=FloatToStrF(x,ffFixed,3,2);
   D=D->kitas;
  }

}
void TForm1::BusenuRadimas(int tipas)
```

```
{
 int cikl_kint=0;
 bool yra;                  //ziurim ar yra nors wiena laiswa kasa
 int i,j,ii;
 float ints;                //intensyvumas
 int *x;                  //tiriamosios busenos coo
 TBus *D1;                //pagalbine rodykle
 TBus *D2;                //pagalbine rodykle
 TBus *Dck;                //pagalbine rodykle,rodo i ta bsuena kurios numeris
cikl_kint
 TBusIntens *D;             //pagalbine rodykle
 x=new int[K];
 D2=TBPr;
 Dck=TBPr;
 while(cikl_kint<N)
  {
   for (j=0;j<K;j++)
    x[j]=Dck->sk[j];
   yra=false;
//tikrinam srautus
 for (ii=0;ii<R;ii++)
  {
   radom=false;
   if (tipas==1) ints=BusenosKoksoMG1(x,ii);     // BUSENOS !!!
   else ints=BusenosKoksoGM1(x,ii);   // BUSENOS
   //ints=0.1;
   if (radom)
   {
    if(!PatikrinimasVienatiskumo(x,cikl_kint,ints))
     {
      D1=new TBus;
      D1->sk=new int[K];
      for(j=0;j<K;j++)
       D1->sk[j]=x[j];
      D1->kitas=NULL;
      D2->kitas=D1;
      D2=D2->kitas;
      D=new TBusIntens;
      D->BI.BInt=ints;
      D->BI.eil=cikl_kint;
      D->BI.st=N;
      D->kitas=TPr;
      TPr=D;
```

```
      N++;
     }
    for (j=0;j<K;j++)   //atstatom i Dck->sk busena
     x[j]=Dck->sk[j];
   }
  }
  //tikrinam aptarnavimo aparatus
  for (ii=R;ii<M;ii++)
  {
   radom=false;
   if (tipas==1) ints=BusenosKoksoMG1(x,ii);     // BUSENOS !!!
   else ints=BusenosKoksoGM1(x,ii);   // BUSENOS
   //ints=0.1;
   if (radom)
   {
    if(!PatikrinimasVienatiskumo(x,cikl_kint,ints))
     {
      D1=new TBus;
      D1->sk=new int[K];
      for(j=0;j<K;j++)
       D1->sk[j]=x[j];
      D1->kitas=NULL;
      D2->kitas=D1;
      D2=D2->kitas;
      D=new TBusIntens;
      D->BI.BInt=ints;
      D->BI.eil=cikl_kint;
      D->BI.st=N;
      D->kitas=TPr;
      TPr=D;
      N++;
     }
    for (j=0;j<K;j++)   //atstatom i Dck->sk busena
     x[j]=Dck->sk[j];
   }
  }
  cikl_kint++;
  Dck=Dck->kitas;
  } //while(cikl_kint<N) pab
}
//---------------------------------------------------------------------
float TForm1::Busenos(int *xx, int poz)
{
```

```
switch(poz)
{
 case 0:
    {
     if( xx[0]==0)
       { xx[0]=1;
         radom = true;
       }
     return 0.5*1000000;
    }  // case 0 pabaiga

 case 1:
    {
     if( xx[0]==0)
       { xx[0]=2;
         radom = true;
       }
     return 0.5*1000000;
    }  // case 0 pabaiga

 case 2:
    {
     if( xx[0]==1)
       {
         if((xx[2]==0)&&(xx[1]<L))
           { xx[0]=0;
             xx[1]++;
             xx[2]=1;
             radom=true;
           }
         else if (xx[1]<L)
           { xx[0]=0;
             xx[1]++;
             radom=true;
           }
       }
     return 1;
    }    //  case 2 pabaiga

   case 3:
    {
     if( xx[0]==2)
       {
```

```
          if((xx[2]==0)&&(xx[1]<L))
            { xx[0]=0;
              xx[1]++;
              xx[2]=1;
              radom=true;
            }
          else  if (xx[1]<L)
            { xx[0]=0;
              xx[1]++;
              radom=true;
            }
        }
      return 0.5;
     }    //  case 2 pabaiga

  case 4:
     {
      if( xx[2]==1)
        {
          if( xx[1]>1)
            { xx[1]--;
              xx[2]=1;
              radom=true;
            }
          else
            { xx[1]--;
              xx[2]=0;
              radom=true;
            }
        }
      return 1;
     }    //  case 2 pabaiga
 }   // switch pabaiga

}  // Busenos pabaiga

//---------------------------------------------------------------------
bool TForm1::PatikrinimasVienatiskumo(int s[], int Bnr, double intens) //Bnr
busenos numeris is kurios atejom i surasta busena
{
 bool v=false;
 TBusIntens *D;
 ///TBusA *D1;
```

```
 ///D1=TBAPr;
 TBus *D1;
 D1=TBPr;
 int i=0,j;
 while((i<N)&&(!v))
 {
     j=0;
     v=true;
     while((j<K)&&(v))
      {
       ///if (int(s[j]*100000)!=int(D1->sk[j]*100000))
       if (s[j]!=D1->sk[j])
        {
          v=false;
        }
       j++;
      }
    //}
  D1=D1->kitas;
  i++;
 }

 if (v)
  {
   D=new TBusIntens;
   D->BI.BInt=intens;
   D->BI.eil=Bnr;
   D->BI.st=i-1;
   D->kitas=TPr;
   TPr=D;
  }
 return v;   //jeigu tokia busena jau buwo grazins true
}
//---------------------------------------------------------------------------
float TForm1::BusenosKoksoMG1(int *xx, int poz)
{
    switch(poz)
  {
  case 0:
     {
      if( (xx[1]==0)&&(xx[0]<L))
        {
          xx[0]++;
```

```
        xx[1]=1;
        radom=true;
      }
        else if (xx[0]<L)
          { xx[0]++;
            radom=true;
          }
    return 1;
   }  // case 0 pabaiga

case 1:
   {
    if( xx[1]==1)
      { xx[1]=2;
        radom = true;
      }
    return 0.005592;
    //return 0.7075*1.4587;
   }  // case 1 pabaiga

case 2:
   {
    if( xx[1]==1)
      {
        if( xx[0]>1)
          { xx[0]--;
            xx[1]=1;
            radom=true;
          }
        else
          { xx[0]--;
            xx[1]=0;
            radom=true;
          }
      }
     return  (1-0.005592)*1.132817;
    //return (1-0.7075)*1.4587;
   }  // case 2 pabaiga

case 3:
   {
    if( xx[1]==2)
      {
```

```
          if( xx[0]>1)
            { xx[0]--;
              xx[1]=1;
              radom=true;
            }
          else
            { xx[0]--;
              xx[1]=0;
              radom=true;
            }
        }
      return 0.210069;
      //return 2.808;
     }    //  case 3 pabaiga

 }   // switch pabaiga


} // BusenosKoksoMG1  pabaiga




//----------------------------------------------------------------------
float TForm1::BusenosKoksoGM1(int *xx, int poz)
{
    switch(poz)
  {
  case 0:
     {
      if( xx[0]==0)
        { xx[0]=1;
          radom = true;
        }
        return 100000;
      //return 100000;
     }  // case 0 pabaiga

  case 1:
     {
      if( xx[0]==1)
        { xx[0]=2;
          radom = true;
        }
```

```
    //  return kok3_p*kok3_miu1;
    return 0.7075*1.4587;
   }  // case 1 pabaiga

case 2:
   {
    if( xx[0]==1)
      {
        if((xx[2]==0)&&(xx[1]<L))
          { xx[0]=0;
            xx[1]++;
            xx[2]=1;
            radom=true;
          }
        else if (xx[1]<L)
          { xx[0]=0;
            xx[1]++;
            radom=true;
          }
      }
    //  return (1-kok3_p)*kok3_miu1;
    return (1-0.7075)*1.4587;
   }  // case 2 pabaiga


case 3:
   {
    if( xx[0]==2)
      {
        if((xx[2]==0)&&(xx[1]<L))
          { xx[0]=0;
            xx[1]++;
            xx[2]=1;
            radom=true;
          }
        else if (xx[1]<L)
          { xx[0]=0;
            xx[1]++;
            radom=true;
          }
      }
    //return kok3_miu2;
    return 2.808;
```

```
     }    //  case 3 pabaiga

   case 4:
     {
      if( xx[2]==1)
        {
          if( xx[1]>1)
            { xx[1]--;
              xx[2]=1;
              radom=true;
            }
          else
            { xx[1]--;
              xx[2]=0;
              radom=true;
            }
        }
      return 1.5;
     }    //  case 4 pabaiga

 }   // switch pabaiga


} // BusenosKoksoMG1  pabaiga
//-------------------------------------------------------------------------
float TForm1::BusenosErlangoGM1(int *xx, int poz)
{
 switch(poz)
 {
   case 0:
   {
    if ( xx[0]==0)
       { xx[0]=1;
         radom = true;
       }
    return 100000;
   } // case 0 pabaiga

   case 1:
   {
    if ( xx[0]==1)
       { xx[0]=2;
         radom = true;
```

```
    }
 return 0.0769*100000;
} // case 1 pabaiga

case 2:
{
 if ( xx[0]==1)
    { xx[0]=5;
      radom = true;
    }
 return (1-0.0769)*100000;
} // case 2 pabaiga

case 3:
{
 if ( xx[0]==2)
    { xx[0]=3;
      radom = true;
    }
 return 2.3077;
} // case 3 pabaiga

case 4:
{
 if ( xx[0]==3)
    { xx[0]=4;
      radom = true;
    }
 return 2.3077;
} // case 4 pabaiga

case 5:
{
 if ( xx[0]==4)
    {
     if ((xx[2]==0)&&(xx[1]<L))
        { xx[0]=0;
          xx[1]++;
          xx[2]=1;
          radom = true;
        }
     else  if (xx[1]<L)
        { xx[0]=0;
```

```
          xx[1]++;
          radom =true;
        }
    }
 return 2.3077;
} // case 5 pabaiga

case 6:
{
 if ( xx[0]==5)
    { xx[0]=6;
      radom = true;
    }
 return 2.3077;
} // case 6 pabaiga

case 7:
{
 if ( xx[0]==6)
    { xx[0]=7;
      radom = true;
    }
 return 2.3077;
} // case 7 pabaiga

case 8:
{
 if ( xx[0]==7)
    { xx[0]=8;
      radom = true;
    }
 return 2.3077;
} // case 8 pabaiga

case 9:
{
 if ( xx[0]==8)
    {
     if ((xx[2]==0)&&(xx[1]<L))
        { xx[0]=0;
          xx[1]++;
          xx[2]=1;
          radom = true;
```

```
          }
        else  if (xx[1]<L)
          { xx[0]=0;
            xx[1]++;
            radom =true;
          }
      }
    return 2.3077;
   } // case 9 pabaiga

   case 10:
   {
    if ( xx[2]==1)
      {
        if ( xx[1]>1)
          { xx[1]--;
            xx[2]=1;
            radom = true;
          }
        else
          { xx[1]--;
            xx[2]=0;
            radom = true;
          }
      }
    return 1;
   }  // case 10 pabaiga

 }  //  switch pabaiga

} // BusenosErlango  pabaiga

//----------------------------------------------------------------------

void TForm1::KasuSk(AnsiString DF)
{
 char elem='a';
 float elem1=0;
 FILE *F;
 F = fopen(DF.c_str(),"r");
 K=0;
 IK=0;
 //nuskaitom kiek bus busenos skaitmenu
```

```
 while (elem!='=') fscanf(F, "%c", &elem);
 fscanf(F, "%c", &elem);
 while (elem!=';')
  {
   fscanf(F, "%f%c", &elem1, &elem);
   K++;
  }
 elem='a';
 elem1=0.;
 //nuskaitom kiek bus intensyvumu
 while (elem!='=') fscanf(F, "%c", &elem);
 fscanf(F, "%c", &elem);
 while (elem!=';')
  {
   fscanf(F, "%f%c", &elem1, &elem);
   IK++;
  }
 fclose(F);
}
//---------------------------------------------------------------------------
void TForm1::SurastiFaziuIntens()       //surasti faziu intensyvumus miu1, miu2,
p1
{
 float g1,g2,g3;
 g1=Tintens[0];
 g2=Tintens[1]/2;
 g3=Tintens[2]/6;

 //tikrinsim ar parametrai patenka metodo A.S.
 if (((g2/pow(g1,2)<1) && (g2/pow(g1,2)-g3/(2*pow(g1,3))<0.5)) ||
((g2/pow(g1,2)>1) && (g2/pow(g1,2)-g3/(2*pow(g1,3))>0.5)))
  {
   Memo1->Lines->Add("  Taikoma dvieju momentu");
   Memo1->Lines->Add("lyginimo aproksimacija, nes pateikti");
   Memo1->Lines->Add("pradiniai momentai nepatenka");
   Memo1->Lines->Add("i 'Triju momemtu lyginimo' metodo");
   Memo1->Lines->Add("apibrezimo sriti");

   Tintens[0]=2/g1;
   Tintens[2]=0.5/((2*g2-pow(g1,2))/pow(g1,2));
   Tintens[1]=Tintens[0]*Tintens[2];
```

```
   Tintens[1]=Tintens[1]/(1-Tintens[2]);        //todel kad dvieju ir triju
momentu lyginimo atvejais nagrinejamos skirtingos sistemos schemos - antro
aprato intens=miu2
  }
  else
  {
   Tintens[1]=(g2-pow(g1,2))/(pow(g1,3)-2*g1*g2+g3);
   float posaknis=pow((1-Tintens[1]*g1),2)+4*pow(Tintens[1],2)*(g2-pow(g1,2));
   if (posaknis<0)
     {ShowMessage("Posaknis neigiamas! Aproksimacija su tokiais pradiniais
momentais negali buti atlikta."); posaknis=-posaknis;}
   Tintens[0]=(1+Tintens[1]*g1+sqrt(posaknis))/(2*g1-2*Tintens[1]*(g2-
pow(g1,2)));
   if (Tintens[0]<0)
     (1+Tintens[1]*g1-sqrt(posaknis))/(2*g1-2*Tintens[1]*(g2-pow(g1,2)));
   Tintens[2]=Tintens[1]*(Tintens[0]*g1-1)/(Tintens[1]*(Tintens[0]*g1-
1)+Tintens[0]);
  }
}
//---------------------------------------------------------------------------
void TForm1::Statistika_S5()       //ivairi statistika sistemai S5- 1 srautas 2
aptranavimo sistemos
{
 //float L1=0; //vidutinis 1 eiles ilgis
 //float L2=0;
 TBus *X=TBPr;
 int i=0;
 float* A=new float[7];
 for(i=0;i<K/2+1;i++) A[i]=0;
 i=0;
  while (X)
  {
   A[0]=A[0]+X->sk[0]*r[i];
   if (X->sk[1]>0) A[1]=A[1]+(X->sk[0]-1)*r[i];
   else A[1]=A[1]+X->sk[0]*r[i];
   if (X->sk[0]==0) A[2]=A[2]+r[i];
   X=X->kitas;
   i++;
  }
 Memo1->Lines->Add("L   "+ FloatToStrF(A[0],ffFixed,5,4));
 Memo1->Lines->Add("Lq- "+ FloatToStrF(A[1],ffFixed,5,4));
 Memo1->Lines->Add("p0- "+ FloatToStrF(A[2],ffFixed,5,4));
}
```

```
//---------------------------------------------------------------------------
void TForm1::StatistikaKoksoGM1()
{
 TBus *X=TBPr;
 int i=0;
 float* A=new float[7];
 for(i=0;i<K/2+1;i++) A[i]=0;
 i=0;
  while (X)
  {
   A[0]=A[0]+X->sk[1]*r[i];
   if (X->sk[2]>0) A[1]=A[1]+(X->sk[1]-1)*r[i];
   else A[1]=A[1]+X->sk[1]*r[i];
   if (X->sk[1]==0) A[2]=A[2]+r[i];
   if (X->sk[1]==1) A[3]=A[3]+r[i];
   X=X->kitas;
   i++;
  }
 Memo1->Lines->Add("L-  "+ FloatToStrF(A[0],ffFixed,5,4));
 Memo1->Lines->Add("Lq- "+ FloatToStrF(A[2],ffFixed,5,4));
 Memo1->Lines->Add("p0- "+ FloatToStrF(A[3],ffFixed,5,4));
 Memo1->Lines->Add("p1- "+ FloatToStrF(A[4],ffFixed,5,4));
}
void TForm1::Erlango2()
{
 m1 = 0.9094;
 m2 = 1.8589;
 c = sqrt(m2-m1*m1)/m1;
 c2 = c*c;
 asim = m3-3*m1*m2+2*m1*m1*m1/((m2-m1*m1)*sqrt(m2-m1*m1));
 erl2_n = ceil(1/c2);
 erl2_p=(1/(1+c2))*(erl2_n*c2-sqrt(erl2_n*(1+c2)-erl2_n*erl2_n*c2));
 erl2_miu = (erl2_n-erl2_p)/m1;
 Memo1->Lines->Add(IntToStr(erl2_n));
 Memo1->Lines->Add(FloatToStrF(erl2_p,ffFixed,6,4));
 Memo1->Lines->Add(FloatToStrF(erl2_miu,ffFixed,6,4));

}

void TForm1::Kokso2()
{
 m1 = 0.9094;
 m2 = 1.8589;
```

```
 c = sqrt(m2-m1*m1)/m1;
 c2 = c*c;
 asim = m3-3*m1*m2+2*m1*m1*m1/((m2-m1*m1)*sqrt(m2-m1*m1));
 Memo1->Lines->Add("Kokso2");
 kok2_miu1=2/m1;
 kok2_p=0.5/c2;
 kok2_miu2=kok2_miu1*kok2_p;
 Memo1->Lines->Add(FloatToStrF(kok2_p,ffFixed,6,4));
 Memo1->Lines->Add(FloatToStrF(kok2_miu1,ffFixed,6,4));
 Memo1->Lines->Add(FloatToStrF(kok2_miu2,ffFixed,6,4));
}

void TForm1::Kokso3()
{
 float f1,f2,f3,D;
 float mc2, mc3;
 mc2 = m2/(m1*m1);
 mc3 = m3/(m1*m2);
 if ( (4*mc2/3<=mc3)&&(mc3<=6*(mc2-1)/mc2)&&(3/2<=mc2)&&(mc2<=2)
        || ((4*mc2/3<=mc3)&&(mc2>2)))
    {
      c = sqrt(m2-m1*m1)/m1;
      c2 = c*c;
      // asim = m3-3*m1*m2+2*m1*m1*m1/((m2-m1*m1)*sqrt(m2-m1*m1));
      f1 = m1;
      f2 = m2/2;
      f3 = m3/6;
      c2 = (m2-m1*m1)/(m1*m1);
      D = (f1*f2-f3)*(f1*f2-f3)-4*(f2*f2-f1*f3)*(f1*f1-f2);
      kok3_miu2 = ((f1*f2-f3)+sqrt(D))/(2*(f2*f2-f1*f3));
      kok3_miu1 = (kok3_miu2*f1-1)/(kok3_miu2*f2-f1);
      kok3_p=(kok3_miu2*(f1*kok3_miu1-1))/kok3_miu1;
      Memo1->Lines->Add("Duomenys tenkina Kokso aproksimacijos salygas");
      Button3->Enabled = true;
      Button4->Enabled = true;
      Memo1->Lines->Add("Kokso skirstinio parametrai:");
      Memo1->Lines->Add("p = "+FloatToStrF(kok3_p,ffFixed,6,4));
      Memo1->Lines->Add("miu1 = "+FloatToStrF(kok3_miu1,ffFixed,6,4));
      Memo1->Lines->Add("miu2 = "+FloatToStrF(kok3_miu2,ffFixed,6,4));
    }
 else
    Memo1->Lines->Add("Kokso aproksimacija siem duomenim netinka");
}
```

```
void TForm1::Erlango3()
{
 float xx,yy,A,B,C;
 Button6->Enabled = true;
 Button7->Enabled = true;
 erl3_n = max( ceil(1/(c*c)),ceil((-asim+1/(c*c*c)+1/c+2*c)/(asim-c+1/c)));
 xx = m1*m3-m2*m2*(erl3_n+2)/(erl3_n+1);
 yy = m2-m1*m1*(erl3_n+1)/erl3_n;
 C = m1*xx;
 B = -(erl3_n*xx+yy*yy*(erl3_n*erl3_n+2*erl3_n)/(erl3_n+1)+(erl3_n+2)*yy*m1*m1);
 A = erl3_n*(erl3_n+2)*m1*yy;
 erl3_miu1 = 2*A/(sqrt(B*B-4*A*C)-B);
 erl3_miu2 = 2*A/(-sqrt(B*B-4*A*C)-B);
 erl3_p = (m1/erl3_n-1/erl3_miu2)/(1/erl3_miu1-1/erl3_miu2);
 Button5->Enabled = true;
 Button6->Enabled = true;
 Memo1->Lines->Add("Erlango misinio  parametrai:");
 Memo1->Lines->Add("n = "+IntToStr(erl3_n));
 Memo1->Lines->Add("p = "+FloatToStrF(erl3_p,ffFixed,6,4));
 Memo1->Lines->Add("miu1 = "+FloatToStrF(erl3_miu1,ffFixed,6,4));
 Memo1->Lines->Add("miu2 = "+FloatToStrF(erl3_miu2,ffFixed,6,4));

}
//---------------------------------------------------------------------------
void TForm1::Isvedimas(int tipas)
{
 Memo2->Lines->Add("Ivykiu skaicius "+IntToStr(M));
 if (M>0)
 {
  BusenuRadimas(tipas);
  Apjungimas();
  //if (CheckBox1->Checked==true)
  IsvestiStrGrid();
  //else TabSheet3->TabVisible=false;
  StacTikRadimas();
  IsvestiMemo();
  SurastiMax();
  Normavimas();         //!!!
}
 TBusIntens *D;
 while (TPr)
  {D=TPr; TPr=TPr->kitas; delete D;}
 TBus *DD;
```

```
 while (TBPr)
  {DD=TBPr; TBPr=TBPr->kitas; delete DD;}
}
//---------------------------------------------------------------------
void __fastcall TForm1::Button4Click(TObject *Sender)
{
 Memo1->Lines->Clear();
 Memo2->Lines->Clear();
 K=0;
 M=0;
 IvestiPradSalygasKoksoGM1();
 Isvedimas(2);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button3Click(TObject *Sender)
{
 Memo1->Lines->Clear();
 Memo2->Lines->Clear();
 K=0;
 M=0;
 IvestiPradSalygasKoksoMG1();
 Isvedimas(1);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button8Click(TObject *Sender)
{
 //PageControl1->Pages[0]->Show();
 //PageControl1->Pages[0]->Caption="Pagrindinis";
 //PageControl1->Pages[0]->Show();
 //PageControl1->Pages[1]->Caption="Perejimu matrica";
 //Close();
 PageControl1->Pages[0]->Show();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button9Click(TObject *Sender)
{
 PageControl1->Pages[1]->Show();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
 FILE *FD;
 char R[80];
```

```
 int i = 1;
 float tikr;
 bool visiteig;
 visiteig = true;
 n = 0;
 FD = fopen("Duomenys/Imties duomenys.txt","r");
 while(!feof(FD))
      { fgets(R,80,FD);
        n++;
      }
 fclose(FD);
 y = new float [n+1];
 *(y+0)=7;
 FD = fopen("Duomenys/Imties duomenys.txt","r");
 for (i=1;i<=n;i++) fscanf(FD,"%f",y+i);
 fclose(FD);
 for ( i=1;i<=n;i++)
     { tikr = *(y+i);
       if (tikr<0) visiteig = false;
     }
 if (visiteig == false) ShowMessage("Duomenu faile yra neigiamu skaitmenu");
 Memo1->Lines->Add("Duomenys nuskaityti");
 m1 = 0;
 m2 = 0;
 m3 = 0;
 for ( int i=1; i<=n; i++)
     { m1 += *(y+i);
       m2 += (*(y+i))*(*(y+i));
       m3 += (*(y+i))*(*(y+i))*(*(y+i));
     }
 m1 = m1/n;
 m2 = m2/n;
 m3 = m3/n;
 c = sqrt(m2-m1*m1)/m1;
 c2 = c*c;
 asim = m3-3*m1*m2+2*m1*m1*m1/((m2-m1*m1)*sqrt(m2-m1*m1));
 Memo1->Lines->Add(FloatToStrF(m1,ffFixed,6,2));
 Memo1->Lines->Add(FloatToStrF(m2,ffFixed,6,2));
 Memo1->Lines->Add(FloatToStrF(m3,ffFixed,6,2));

}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
```

```
{
 FILE *FD;
 char R[80];
 int i = 1;
 float tikr;
 bool visiteig;
 visiteig = true;
 n = 0;
 FD = fopen("Duomenys/Tikslus duomenys.txt","r");
 while(!feof(FD))
       { fgets(R,80,FD);
         n++;
       }
 fclose(FD);
 if (n<3)  ShowMessage("Duomenu faile truksta duomenu");
 else
 {
 FD = fopen("Duomenys/Tikslus duomenys.txt","r");
 fscanf(FD,"%f\n",&m1);
 fscanf(FD,"%f\n",&m2);
 fscanf(FD,"%f/n",&m3);
 fclose(FD);
 c = sqrt(m2-m1*m1)/m1;
 c2 = c*c;
 asim = m3-3*m1*m2+2*m1*m1*m1/((m2-m1*m1)*sqrt(m2-m1*m1));
 Memo1->Lines->Add("Duomenys nuskaityti");
 Memo1->Lines->Add("Pradiniai skirstinio G momentai:");
 Memo1->Lines->Add("m1 = "+FloatToStrF(m1,ffFixed,6,4));
 Memo1->Lines->Add("m2 = "+FloatToStrF(m2,ffFixed,6,4));
 Memo1->Lines->Add("m3 = "+FloatToStrF(m3,ffFixed,6,4));
 Erlango3();
 Kokso3();
 }
}
//----------------------------------------------------------------------------
```