KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

MATEMATINĖS SISTEMOTYROS KATEDRA

Tomas Lazauskas

Aptarnavimo sistemų modeliavimo Markovo grandinėmis programinių priemonių sukūrimas ir tyrimas

Magistro darbas

Vadovas
doc. dr. E. Valakevičius

KAUNAS, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

MATEMATINĖS SISTEMOTYROS KATEDRA

TVIRTINU
Katedros vedėjas
prof. habil. dr. V. Pekarskas
2010 06 05

Aptarnavimo sistemų modeliavimo Markovo grandinėmis programinių priemonių sukūrimas ir tyrimas

Taikomosios matematikos magistro baigiamasis darbas

Vadovas
doc. dr. E. Valakevičius
2010 06 03

Recenzentas
dr. R. Alzbutas
2010 06 01

Atliko
FMMM-8 gr. stud.
T. Lazauskas
2010 05 25

KAUNAS, 2010

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., vyriausiasis analitikas (DnB NORD Bankas)

Zenonas Navickas, profesorius (KTU)

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

Lazauskas T. Creation and analysis of software for modeling Markovian queuing systems: Master's work in applied mathematics / supervisor dr. E. Valakevičius; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2010. – 60 p.

SUMMARY

When modeling stochastic systems with large set of states using Markovian processes we encounter several problems: 1) generating all possible states of the system, 2) finding the matrix of transition intensities among states, 3) constructing and solving the system of equations for calculating the steady state probabilities of states, 4) estimating system's stochastic characteristics.

This paper analyses the underlying multi-server and multi-class queuing system which is receiving applications of corresponding intensities distributed by Poisson process and can be spread among multi-class servers with fixed waiting positions, where service time $\boldsymbol{S}$ of a application has a general probability distribution function $G(t)$.

The object of this paper is to create the software which enables to compute stationary probabilities and to estimate stochastic characteristics of the system which is defined in schematic structure.

Turinys

LENTELIŲ SĄRAŠAS

PAVEIKSLĖLIŲ SĄRAŠAS

Įvadas

Modeliuojant stochastines sistemas Markovo procesais su didele būsenų aibe, susiduriama su keletu problemų:

1) visų galimų sistemos būsenų generavimas,
2) perėjimų intensyvumų tarp būsenų matricos suradimas,
3) lygčių sistemos stacionariosioms būsenų tikimybėms suskaičiuoti sudarymas ir sprendimas,
4) aptarnavimo sistemos tikimybinių charakteristikų apskaičiavimas.

Darbe bus nagrinėjama mišraus tipo masinė aptarnavimo sistema, kurią gali sudaryti n paraiškų šaltinių, su atitinkamais paprasčiausių srautų intensyvumais, o paraiškos gali būti paskirstomos į m aptarnavimo įrenginių su fiksuotomis prioritetinėmis paraiškų eilėmis, kur paraiškos aptarnavimo laikas S yra pasiskirstęs pagal apibendrintą tikimybinio skirstinio funkciją $G(t)$.

Darbo tikslas – sukurti programinę įrangą, kuri pagal pateiktą sistemos funkcionavimo aprašymą schematine struktūra apskaičiuoja sistemos stacionariąsias tikimybes ir tikimybines charakteristikas.

1 TEORINĖ DALIS

1.1 Sistemos aprašymas

Darbe nagrinėjame sistemą S, kurios evoliuciją aprašo atsitiktinis Markovo procesas ξ(t). Atsitiktiniu procesu vadiname atsitiktinių dydžių sistemą $x = x(t, w)$, kur $t \epsilon T$ ir $w \epsilon \Omega$, apibrėžtą vienoje tikimybinėje erdvėje (Ω, F, P) (Ω - elementariųjų įvykių aibė, F – įvykių algebra ir P – tikimybė, apibrėžta įvykių algebroje F) ir priklausančią nuo parametro t, įgyjančio reikšmes iš tam tikros aibės T. Atsitiktinis procesas yra vadinamas Markovo, jeigu visos proceso tikimybinės charakteristikos ateityje priklauso tik nuo to, kokioje būsenoje x_{i_0} šis procesas yra dabartiniu momentu t_0 ir nepriklauso nuo to kaip šis procesas kito iki momento t_0, t.y. bet kuriam natūraliam skaičiui n ir bet kuriems laiko momentams $t_n, t_{n-1}, \dots, t_1, t_0 < t$ iš T, jei Δτ > 0, teisinga (1.1.) lygybė:

$$p\left(x(t_0 - \tau) = \frac{x_j}{x(t_0)} = x_{i_0}, x(t_1) = x_{i_1}, \dots, , x(t_n) = x_{i_n}\right) = \frac{x_j}{x(t_0)} = x_{i_0}. \qquad \mathbf{(1.1.)}$$

Norint aprašyti Markovo procesą, reikia:

1. Nurodyti visas galimas būsenas, kuriose gali būti sistema.

2. Sudaryti sistemos būsenų grafą, t.y. nurodyti visus galimus sistemos perėjimus iš būsenos į būseną.

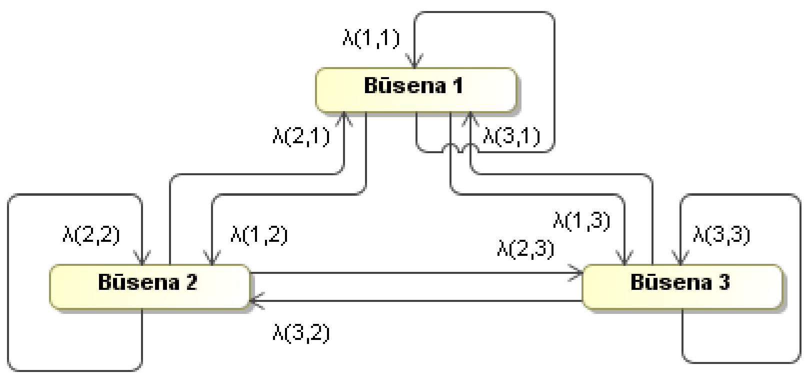
3. Kiekvienam galimam perėjimui nurodyti atitinkamą įvykių srauto intensyvumą $\lambda_{i,j}(t)$, kuris procesą perveda iš būsenos S_i į būseną S_j.

4. Nurodyti kokioje būsenoje sistema yra pradiniu laiko momentu (t = 0).

Sutarta, kad jei iš būsenos S_i į būseną S_j pereiti negalima, tai būsenų grafe rodyklės nežymimos ir atitinkamo srauto intensyvumas laikomas lygiu nuliui, t.y. $\lambda_{i,j} = 0$. Bendru atveju sistemos būsenų perėjimo matrica įgauna pavidalą:

$$\begin{pmatrix} \lambda_{1,1} & \lambda_{1,2} & \dots & \lambda_{1\,j} & \dots & \lambda_{1,n} \\ \lambda_{2,1} & \lambda_{2,2} & \dots & \lambda_{2,j} & \dots & \lambda_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \lambda_{i,j} & \lambda_{i,j} & \dots & \lambda_{i,j} & \dots & \lambda_{i,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \lambda_{n,1} & \lambda_{n,2} & \dots & \lambda_{n,j} & \dots & \lambda_{n,n} \end{pmatrix} \qquad (\mathbf{1.2.})$$

O būsenų grafikas, esant trim būsenom, pavaizduotas 1.1. pav.



1.1. pav. Trijų būsenų perėjimų grafas

Darbe nagrinėjame atvejį kuomet galimų reikšmių ξ(t) erdvė (sistemos būsenų erdvė) yra diskreti ir baigtinė. Tegul sistemos būsenos yra S_1, S_2, …, S_n, tada $\xi(t) \epsilon\ S = (S_1, S_2, \dots, S_n)$. Jei sistemoje su baigtiniu būsenų skaičiumi visi įvykių srautai yra paprasčiausi, tai tam, kad egzistuotų sistemos stacionarus darbo režimas pakanka, kad iš kiekvienos būsenos būtų galima pereiti į bet kurią būseną. Stacionariosios sistemos būsenų tikimybes gali būti apskaičiuotos iš algebrinių lygčių sistemos:

$$\sum_{j=1}^{N} \lambda_{ji}\, q_j = q_i \sum_{j=1}^{N} \lambda_{ij}\ , i = \overline{1, N} \qquad (\mathbf{1.3.})$$

Nežinomas Markovo proceso tikimybes $q_1, q_2, \dots , q_n$ galima apskaičiuoti tik laisvai parinktos konstantos tikslumu. Tam, kad gautume tikslias tikimybių reikšmes, reikia prie lygčių sistemos **(1.3.)** pridėti normavimo sąlygą, kad bendra tikimybių suma yra lygi 1, t.y.:

$$\sum_{i=1}^{N} q_i = 1. \qquad (\mathbf{1.4.})$$

Kuomet sistema turi daug (šimtus ar tūkstančius) būsenų, tai be kompiuterio sudaryti būsenų grafą ir atitinkamą algebrinių lygčių sistemą praktiškai neįmanoma. Jei lygčių sistemą ir turėtume, tai gauti analitinį sprendinį taip pat retai pavyksta. Paprastai tiesinių lygčių sistemos koeficientų matrica yra labai „reta“, t.y. turi daug nulinių elementų.

1.2 Metodų apžvalga

Markovo procesai suteikia galimybę pakankamai paprastai, bet efektyviai aprašyti ir analizuoti dinaminės sistemos savybes. Dažnai yra naudojami tiesioginiai ir/arba iteraciniai metodai Markovo grandinių būsenų skaitinių sprendimų analizei. Tiesioginiai metodai pasižymi tuo, kad atlieka skaičiavimus ir pakeitimus parametrų matricoje ir naudoja fiksuotą skaičiavimo laiką, nepriklausomą nuo parametrų reikšmių. Tačiau šie metodai turi savybę kaupti apvalinimo paklaidas ir sukelia problemų susijusių su retų matricų saugojimu.

Iteraciniai metodai yra paremti konvergavimu į norimą sprendinį. Šio metodo evoliuciją galima nutraukti bet kuriuo metu, kai tik paskutinės iteracijos metu gautas rezultatas yra pakankamai arti tikslaus atsakymo. Vienas iš didžiausių šio tipo metodų privalumų lyginant su tiesioginio skaičiavimo metodais yra tas, kad jie išsaugo parametrų matricos retumą, kadangi gali būti panaudotos efektyvios matricų saugojimo schemos, o taip pat ir matricų retumu besiremiantys algoritmai. O vienas iš pagrindinių šio metodų grupės trukumų yra tai, kad iteracinio metodo konvergavimas į norimą rezultatą yra negarantuotas ir dažniausiai priklauso nuo pasirinkto metodo savybių. O taip pat konvergavimo greitis priklauso nuo parametrų matricos reikšmių.

Šiame skyriuje supažindinsime su kai kuriais tiesioginiais ir iteraciniais metodais, jų veikimo principais ir idėjomis. Nagrinėsime šiuos tiesioginius metodus: Gauso eliminuojantis metodas, Grassmann‘o algoritmas, Įdėtųjų Markovo grandinių metodas ir tokius iteracinius metodus: laipsnių, Jokobio bei Gauso ir Zeidelio.

1.2.1 Gauso metodas

Vienas iš pagrindinių metodų skirtų tiesinių lygčių sprendimui yra Gauso, kurio pagalba galima išspręsti bet kurią tiesinę lygčių sistemą. Gauso eliminuojantis metodas – tai nuoseklaus nežinomųjų eliminavimo metodas. Nežinomieji eliminuojami per keletą kartų, elementariai pertvarkant sistemos lygtis.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} \qquad (\mathbf{1.5.})$$

Pirmuoju žingsniu iš visų lygčių, išskyrus pirmąją, eliminuojame x_1 ir toliau pirmosios lygties nebenaudojame eliminuodami kitus nežinomuosius. Antruoju žingsniu iš visų kitu lygčių, išskyrus antrąją, eliminuojame x_2 ir t.t. Elementariaisiais lygčių eilučių pertvarkymais lygčių sistema **(1.5.)** yra pakeičiama ekvivalenčia sistema **(1.6.):**

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1, \\ \qquad a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ \qquad\qquad a_{33}^{(2)}x_3 + \cdots + a_{3n}^{(2)}x_n = b_3^{(2)}, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \qquad\qquad\qquad a_{nn}^{(n-1)} \cdot x_n = b_n^{(n-1)}. \end{cases} \qquad (\mathbf{1.6.})$$

Sistema virsta trikampe sistema, kuri turi tiek pat lygčių, kiek ir nežinomųjų. Koeficientai a_{11}, $a_{22}^{(1)}$, $a_{33}^{(2)}$, …, $a_{nn}^{(n-1)}$ nelygūs nuliui. Iš **(1.6.)** sistemos paskutinės lygties randame b_n, paskui iš priešpaskutinės lygties - b_{n-1} ir t.t., kol pagaliau iš pirmos lygties gauname b_1. Šiuo atveju sistema yra suderinta ir turi vienintelį sprendinį.

Tačiau Gauso metodas spręsti darbe nagrinėjamai lygčių sistemai yra netinkamas, nes jis daugelį nulinių elementų paverčia nenuliniais.

1.2.2 Grassmann‘o algoritmas

Grassmann‘o algoritmas yra skaitiškai stabilus Gauso eliminuojančio algoritmo variantas. Šis algoritmas visiškai nenaudoja atimties veiksmo, todėl yra mažiau jautrus paklaidoms, atsiradusioms dėl atimties tarp labai artimų skaičių. Šis algoritmas buvo sukurtas ergodinėm, diskretaus laiko Markovo grandinėm tirti ir pagrįstas regeneracijos procesų teorija.

Naujos Markovo grandinės, kurios turi viena būsena mažiau negu pradinė grandinė, perėjimų intensyvumų reikšmės yra apskaičiuojamos. Šis šalinamasis žingsnis, t.y. $\bar{q}_{j,i}$ elemento apskaičiavimas, pasiekiamas paprasčiausiai pridedant neneigiamus dydžius prie pradinių neneigiamų reikšmių $q_{j,i}, j \neq i$. Tik tai diagonalieji $q_{i,i}$ ir $\bar{q}_{i,i}$elementai yra neigiami.

Eliminuojanti procedūra yra iteraciniu būdu pritaikoma generuojančiai matricai, su reikšmėmis $q_{j,i}^{(k)}$, pažingsniui tol kol galiausiai gaunama viršutinė trikampė matrica, kur $q_{j,i}^{(k)}$ nusako matricos elementus po k -tosios eliminacijos iteracijos, kur $1 \leq k \leq n-1$. Galiausiai, kiekvienas $q_{j,i}^{(n-1)}$diagonalusis elementas yra lygus -1.

Po eliminacijos proceso seka pakeitimo procesas, kuris nusako tarpusavio ryšius tarp būsenų tikimybių. Norint rasti stacionarias sistemos būsenų tikimybes, reikia įvesti normalizavimo sąlygą.

Šis algoritmas gali būti išskaidytas į keturis žingsnius.

Pirmame žingsnyje nusakome matricą A:

$$A = P - I, \text{ kur P – perėjimų intensyvumų matrica} \qquad \mathbf{(1.7.)}$$

Antras (eliminacijos) žingsnis:

$$l = n-1, n-2, \dots, 1:$$

$$a_{j,i}^{(n-l)} = \begin{cases} \dfrac{a_{j,i}^{(n-l-1)}}{\sum_{m=0}^{l-1} a_{l,m}^{(n-l-1)}}, j < l, i = l \\ a_{j,i}^{(n-l-1)} + \dfrac{a_{j,l}^{(n-l-1)} a_{l,i}^{(n-l-1)}}{\sum_{m=0}^{l-1} a_{l,m}^{(n-l-1)}}, j \neq i, 1 \leq j, i \leq l-1 \\ -1, j = i = l \\ 0, j = l. i < l \end{cases} \qquad \mathbf{(1.8.)}$$

Trečias žingsnis:

$$l = 1,2, \dots, n-1$$
$$x_l = \sum_{i=0}^{l-1} x_i a_{il}^{(n-l)} \qquad \mathbf{(1.9.)}$$

Ketvirtas žingsnis:

$$l = 1,2, \dots, n-1$$
$$\left.\begin{matrix} \pi_i \\ \nu_i \end{matrix}\right\} = \frac{x_i}{\sum_{j=0}^{n-1} x_j} \qquad \mathbf{(1.10.)}$$

Matrica A yra išskaidoma į dvi matricas, viršutinę trikampę matricą U ir apatinę matricą L, taip, kad galiotų (**1.11.**) lygybė.

$$0 = xA = xUL \tag{1.11.}$$

Akivaizdu, kad $0 = xU$ sprendinys galios ir $0 = xA$ lygčių sistemai, taigi, nėra tikslo apskaičiuoti matricą L. Grassmann‘o algoritmas išvengia suprastinimo paklaidų, tačiau gali pasitaikyti ir kauptis apvalinimo paklaidos, todėl šio metodo pritaikomumas yra ribotas iki vidutinio dydžio (apie 500 būsenų) Markovo sistemoms.

1.2.3 Įdėtųjų Markovo grandinių metodas

Darbe aptarnavimo sistemos funkcionavimas yra nagrinėjamas tam tikrais laiko momentais. Jeigu šiais laiko momentais sistemos darbas aprašomas Markovo grandine, tai ji vadinama įdėtąją Markovo grandine į Markovo procesą, Tokie momentai mūsų nagrinėjamu atveju yra momentai kai sistema keičia savo būseną. Įdėtos Markovo grandinės perėjimo tikimybės iš būsenos S_i į būseną S_j yra apskaičiuojamos pagal formulę:

$$p_{ij} = \frac{\lambda_{ij}}{\sum_{j=1}^{N} \lambda_{ij}}, i, j = \overline{1, N} \tag{1.12.}$$

čia N – sistemą sudarančių būsenų skaičius. O įdėtos Markovo grandinės stacionarios tikimybės apskaičiuojamos iš tiesinių lygčių sistemos:

$$p_i = \sum_{j=1}^{N} p_j p_{ji}, i = \overline{1, N} \tag{1.13.}$$

Markovo proceso stacionarios tikimybės, gaunamos iš **(1.3.)** lygčių sistemos, nesutampa su analogiškomis įdėtų Markovo grandinių stacionariomis tikimybėmis, tačiau egzistuoja ryšys tarp šių tikimybių:

$$q_i = \frac{p_i / \sum_{j=1}^{N} \lambda_{ij}}{\sum_{j=1}^{N} (p_j / \sum_{i=1}^{N} \lambda_{ji})}, i = \overline{1, N} \tag{1.14.}$$

Darbe įdėtąją Markovo grandine yra vadinama Markovo grandinės, aprašomos būsenų aibe $(S_1, S_2, \dots, S_n)$, evoliucija $(S_m^{(n)}, m \geq 0)$ nagrinėjama būsenų aibėje $(S_1, S_2, \dots, S_p)$, čia $p < n$. Markovo grandinė, kuri aprašyta būsenų poaibyje $(S_1, S_2, \dots, S_p)$, vadinama įdėta Markovo grandine $(S_m^{(p)}, m \geq 0)$ į grandinę $(S_m^{(n)}, m \geq 0)$.

Įdėtos Markovo grandinės tikimybės apskaičiuojamos taip:

$$p_{ij}^{(p)} = p_{ij}^{(n)} + f_{ij}, \quad i, j = \overline{1, p} \tag{1.15.}$$

čia $p_{ij}^{(p)}$, $p_{ij}^{(n)}$ – atitinkamai Markovo grandinių $(S_m^{(p)}, m \geq 0)$ ir $(S_m^{(n)}, m \geq 0)$ perėjimo tikimybės per vieną žingsnį; f_{ij} - tikimybė, kad Markovo grandinė iš būsenos S_i per vieną žingsnį pateks į vieną iš poaibio $(S_r: p < r \leq n)$ būseną ir iš jo po pirmo sugrįžimo pateks į būseną S_j, t.y.:

$$f_{ij} = \sum_{m=2}^{\infty} P(S_m^{(n)} = S_j, S_k^{(n)} \in (S_r: p < r \leq n), k = \overline{1, m-1} | S_0^{(n)} = S_i) \qquad (\mathbf{1.16.})$$

Yra įrodyta, kad :

$$f_{ij} = \frac{p_{in}^{(n)} \cdot p_{nj}^{(n)}}{1 - p_{nn}^{(n)}}, \quad i, j = \overline{1, p} \qquad (\mathbf{1.17.})$$

jei $p = n - 1$.

Norėdami išspręsti **(1.6.)** lygčių sistemą, įdėkime Markovo grandinę į Markovo procesą.

Tegu $P_N = ||p_{in}^{(N)}||_{NxN}$ - Markovo grandinės $(S_m^{(N)}, m \geq 0)$, aprašytos būsenų aibėje $X_N = (S_1, S_2, \dots, S_n)$, perėjimo tikimybių matrica. Stacionarių tikimybių $p_i = p_i^{(N)}, \ i = \overline{1, N}$ skaičiavimas susideda iš dviejų etapų: sistemos lygčių skaičiaus sumažinimo (Markovo grandinių įdėjimo etapas) ir stacionarių tikimybių apskaičiavimo etapas.

Tegu $X_N, X_{N-1}, X_{N-2}, \dots, X_1$ – Markovo grandinių būsenų aibių seka, gaunama nuosekliai „išmetant" po vieną būseną, t.y.:

$$X_k = (S_1, S_2, \dots, S_k), k = \overline{1, N},$$
$$X_k = X_{k+1} / S_{k+1}, k = \overline{1, N-1}.$$

Sutarta, kad Markovo grandinė su n būsenų ir perėjimų matrica $P_n = ||p_{ij}^{(n)}||_{nxn}$ yra žymima $(S_m^{(n)}, m \geq 0)$. Tuomet sudaryta Markovo grandinių seka įgyja tokį pavidalą:

$$\left(S_m^{(N)}, m \geq 0\right), \left(S_m^{(N-1)}, m \geq 0\right), \dots, \left(S_m^{(1)}, m \geq 0\right) \qquad (\mathbf{1.18.})$$

kur bet kuri grandinė $\left(S_m^{(k)}, m \geq 0\right)$ įdedama į grandinę $\left(S_m^{(k+1)}, m \geq 0\right)$, $k = \overline{1, N-1}$. Šiuo atveju $\left(S_m^{(k)}, m \geq 0\right)$ taip pat yra įdėta į Markovo grandinę $\left(S_m^{(N)}, m \geq 0\right)$. Kadangi Markovo grandinė $\left(S_m^{(n)}, m \geq 0\right)$ pilnai apibrėžiama perėjimų matrica P_n ir būsenų aibe $X_n = (S_1, S_2, \dots, S_n)$, tai tam, kad galėtume aprašyti Markovo grandinių seką $(\mathbf{1.18.})$, būtina sudaryti perėjimų matricų $P_N, P_{N-1}, \dots, P_1$ seką.

Remiantis $(\mathbf{1.17.})$ išraiška, matricų $P_N, P_{N-1}, \dots, P_1$ elementai gali būti apskaičiuojami pagal formulę:

$$p_{ij}^{(k)} = p_{ij}^{(k+1)} + \frac{p_{i,k+1}^{(k+1)} \cdot p_{k+1,j}^{(k+1)}}{1 - p_{k+1,k+1}^{(k+1)}}, i,j = \overline{1,k}, k = N-1, \dots, 1 \qquad (\mathbf{1.19.})$$

Naudojant įdėtų Markovo grandinių seką $\left(S_m^{(N-1)}, m \geq 0\right), \left(S_m^{(N-2)}, m \geq 0\right), \dots, \left(S_m^{(1)}, m \geq 0\right)$, galima apskaičiuoti Markovo grandinės $\left(S_m^{(N)}, m \geq 0\right)$ stacionarias tikimybes $p_i^{(N)}$ laisvai pasirinktos konstantos tikslumu, skaičiuojant pagal formules:

$$\bar{p}_i^{(1)} = 1$$

$$\bar{p}_i^{(k+1)} = \begin{cases} \bar{p}_i^{(k)}, i = \overline{1,k}; \\ \dfrac{\sum_{j=1}^{k} \bar{p}_j^{(k)} \cdot p_{ij}^{(k+1)}}{1 - p_{ii}^{(k+1)}}, i = k+1; k = \overline{1, N-1}; \end{cases} \qquad (\mathbf{1.20.})$$

Sunormuotos tikimybės $p_j^{(N)}$ yra tokios

$$p_j = p_j^{(N)} = \frac{\bar{p}_j^{(N)}}{\sum_{j=1}^{N} \bar{p}_j^{(N)}}, j = \overline{1,N} \qquad (\mathbf{1.21.})$$

Markovo grandinės stacionariosios tikimybės skaičiuojamos iš formulių **(1.19.)**, **(1.20.)** ir **(1.21.)**.

Sprendžiant praktinius uždavinius, formulės **(1.19.)**, **(1.20.)** ir **(1.21.)** nėra patogios, nes sudarant realių sistemų matematinius modelius, mes žinome perėjimo intensyvumus λ_{ij} tarp būsenų, S_i ir S_j, o ne perėjimo tikimybes. Tačiau yra išvestos formulės, pagal kurias galima skaičiuoti Markovo proceso stacionarias tikimybes, naudojant perėjimo intensyvumus tarp būsenų:

$$r_1^{(1)} = 1;$$

$$r_i^{(k+1)} = \begin{cases} r_i^{(k)}, i = \overline{1,k}; \\ \dfrac{\sum_{j=1}^{k} r_j^{(k)} \cdot \lambda_{ji}^{(k+1)}}{\bar{S}_i}, i = k+1; k = \overline{1, N-1}; \end{cases} \qquad (\mathbf{1.22.})$$

$$\text{čia } \bar{S}_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_{ij}$$

Suskaičiavus Markovo proceso stacionariąsias tikimybes galime apskaičiuoti įvairias sistemos tikimybines charakteristikas, apskaičiuojamas pagal formules, kurios kiekvienai sistemai parašomos atskirai.

Darbe aptarnavimo sistemos stacionariąsias tikimybes apskaičiuosime remdamiesi įdėtųjų Markovo grandinių metodu, nes kaip jau minėjome, dažniausiai perėjimų tarp būsenų matrica turi daug nulinių elementų, o Gauso metodas spręsti tokiai lygčių sistemai yra netinkamas, nes jis daugelį nulinių elementų paverčia nenuliniais, o Grassmann‘o algoritmas netinka didelių (virš 500 būsenų) Markovo sistemų analizei. Iteraciniai metodai, kurių algoritmai pateikti skyriuose 1.2.4., 1.2.5. ir 1.2.6. nebuvo pritaikyti, kadangi jų skaičiavimo rezultatai ne visada yra tikslūs ir pasiekiamas konvergavimas.

1.2.4 Laipsnių metodas

Laipsnių metodas yra patikimas iteracinis metodas skaičiuoti stacionariasas baigtinių ergodinių Markovo grandinių sistemos būsenų tikimybes. Kartais jo konvergavimo greitis yra pakankamai lėtas ir išimtinė sąlyga reikalinga konvergavimui - perėjimų matrica P nebūtų periodinė. Laipsnių metodas remiasi diskretaus laiko Markovo grandinės nepastovumu ir iteracijos yra kartojamos tol, kol pasiekiamas tam tikras, nebūtinai pastovių būsenų, stacionarumas ir konvergencija. Dėl šių priežasčių, šis metodas gali būti panaudotas diskretaus laiko Markovo grandinių būsenų tikimybinio vektoriaus $v(n)$ apskaičiavimui.

Lygybė $v = vP$ sufleruoja, kad iteracinį procesą reikia pradėti, nuo intuityvaus pradinio tikimybinio vektoriaus $v^{(0)}$ parinkimo, kuris pakartotinai bus dauginamas iš perėjimų matricos P, kol bus pasiektas konvergavimas į ieškomą vektorių v, t.y. $\lim_{i\to\infty} v^{(i)} = v$. Kadangi laikome, kad Markovo grandinė yra ergodinė, arba bent jau neperiodinė, tai procedūros konvergavimas į stacionariųjų būsenų tikimybinį vektorių fiksuotu tikslumu yra garantuotas. Atskirą šio metodo iteraciją galime išreikšti **(1.23.)** formule:

$$v^{(i+1)} = v^{(i)}P, i \geq 0 \qquad (\mathbf{1.23.})$$

Ryšys tarp pradinio ir iteracinio stacionarių sistemos būsenų tikimybinių vektorių gali būti išreikštas **(1.24.)** išraiška:

$$v^{(i)} = v^{(0)}P^{i}, i \geq 0 \qquad (\mathbf{1.24.})$$

Norint apskaičiuoti galutinį tikimybinio vektoriaus v rezultatą, belieka iš naujo atlikti normavimo procedūrą. Šio metodo konvergavimo greitis priklauso nuo gretimų tikrinių reikšmių santykio, kuo daugiau nedominuojančios tikrinės reikšmės yra arčiau 1, tuo lėčiau metodas konverguoja.

Šio metodo algoritmas:

Pirmuoju žingsniu yra sudaroma matrica A:

$$A = \begin{cases} P, \\ Q/q + I; \end{cases}$$

$$v^{(0)} = \left(v_0^{(0)}, v_1^{(0)}, \dots, v_{n-1}^{(0)}\right)$$

Nustatomas konvergavimo kriterijus ε ir prisikiriame $n = 0$. Sudaroma atinkama vektoriaus normos funkcija $f\left(\left\|v^{(n)}, v^{(l)}\right\|\right), n \geq 1$. Tariame, kad metodo pabaigos sąlyga baigti = ne;

Antrąjį žingsnį kartojame tol, kol pasiekiame konvergavimą:

$$v^{(n+1)} = v^{(n)}A;$$

Jeigu $f\left(\left\|v^{(n)}, v^{(l)}\right\|\right) < \varepsilon$ ir $l \leq n$ tuomet baigti = taip;

Kitu atveju $n = n + 1; l = l + 1$.

Paskutiniu trečiuoju žingsniu randame ieškomas reikšmes:

$$\left.\begin{matrix}\pi \\ v\end{matrix}\right\} \approx v^{(n)}.$$

1.2.5 Jakobio metodas

Tiesinę lygčių sistemą (**1.5.**) sutrumpintai galime užrašyti (**1.25.**) formule:

$$b = xA \qquad (\mathbf{1.25.})$$

Kaip jau buvo minėta anksčiau, matrica *A* nusako diskretaus laiko Markovo grandinės parametrų rinkinį. Sprendžiant (**1.25.**) lygčių sistemą galima taikyti normalizavimo sąlygą, t.y. parenkant vektorių b lygų $b = [0,0, \dots, 0,1]$, priešingu atveju $b = 0$. Iš (**1.5.**) lygčių sistemos galime išreikšti j-tąją lygtį tokiu būdu:

$$b_j = \sum_{i \in S} a_{ij} x_i \qquad (\mathbf{1.25.})$$

Sprendžiant šią lygtį nežinomojo x_j atžvilgiu, gauname tokią išraišką:

$$x_j = \frac{b_j - \sum_{i, i \neq j} a_{ij} x_i}{a_{jj}} \qquad (\mathbf{1.26.})$$

Norint išspręsti šią lygčių sistemą, turi būti parinktas pradinis sprendinys $\hat{x} = [\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{n-1}]$. Kuo jis bus artimesnis ieškomam sprendiniui, tuo greičiau Jakobio metodas konverguos į sprendinį. Pradinio arba tarpinio sprendinio reikšmės yra panaudojamos kaip $x_i, i \neq j$ kintamųjų reikšmės (**1.26.**) lygties dešinėje pusėje. Šis iteracinis metodas reikalauja, kad tokia procedūra būtų pritaikyta visoms lygtims iš (**1.25.**) lygčių sistemos. x^k - k-tosios iteracijos sprendinys yra gaunamas pasinaudojant (k-1)-tosios lygties sprendiniais kiekvienai lygčiai atskirai, kaip parodyta (**1.27.**) formulėje.

$$x^{(k)}{}_j = \frac{b_j - \sum_{i, i \neq j} a_{ij} x^{(k-1)}{}_i}{a_{jj}}, \forall j \in S \qquad (\mathbf{1.27.})$$

(**1.27.**) formule išreikštą lygtį galime užrašyti matricinėje formoje:

$$x^{(k)} = \left(b + x^{(k-1)}(U + L)\right) D^{-1} \qquad (\mathbf{1.28.})$$

Kur $A = D - L - U$, t.y. matrica A yra išskaidoma į diagonaliąją matricą D bei viršutinę ir apatinę matricas U ir L.

Dažniausiai iteracinio metodo pabaiga nusakoma tam tikros vektorių normos reikšmės, tarp gretimų sprendinių, palyginimu su konvergavimo kriterijumi ε (**1.29.**).

$$f\left(\left\|x^{(k)}, x^{(k-1)}\right\|\right) < \varepsilon, k \geq 1 \qquad (\mathbf{1.29.})$$

Reikia pažymėti, kad dėl savo paprastumo, šis metodas lėtai konverguoja į ieškomą sprendinį, todėl jis retai taikomais be papildomų pakeitimų. Tačiau remiantis šiuo metodu buvo sukurtas daug greičiau konverguojantis ir gerai žinomas Gauso ir Zeidelio metodas.

1.2.6 Gauso ir Zeidelio metodas

Tiesinę lygčių sistemą (**1.5.**) perrašykime išreikšdami x_i iš i −tosios lygties, t.y.

$$x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j\right), i = \overline{1, n} \qquad (\mathbf{1.30.})$$

Tarkime, kad $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$ yra pradinis duotosios lygčių sistemos sprendinys. Kadangi mūsų atveju apie jį nieko nežinoma, galima daryti prielaidą, kad $x_i^{(0)} = 0$ $(i = \overline{1, n})$. Tada šį iteracinį metodą bus galima nusakyti (**1.31.**) formule:

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)}\right), i = \overline{1, n} \qquad (\mathbf{1.31.})$$

o Gauso ir Zeidelio iteracinį procesą (**1.32.**) formule:

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}\right), i = \overline{1, n} \qquad (\mathbf{1.32.})$$

Gauso ir Zeidelio iteracinio proceso k-tojoje iteracijoje patikslintos x_i reikšmės naudojamos toje pačioje iteracijoje kitoms x_j $(j = \overline{i + 1, n})$ reikšmėms tikslinti, dėl šios priežasties šis metodas konverguoja greičiau už Jakobio metodą.

Pakankama šio metodo konvergavimo sąlyga:

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \ \forall i = \overline{1,n} \ ir \ \exists i |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \qquad (\mathbf{1.33.})$$

Skaičiavimas paprastai baigiamas, kai $\max_{1 \leq i \leq n} \left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^{(k)}} \right| < \varepsilon$, kur ε – norimas sprendinio tikslumas.

1.3 Ne Markovo grandine aprašytos aptarnavimo sistemos modeliavimas

Dažnai modeliuojant aptarnavimo sistemas susiduriama su ne Markovo grandinėmis, t.y. paraiškų srautas pasiskirstęs pagal Puasono skirstinį, tačiau paraiškų aptarnavimo skirstinys nebūtinai yra eksponentinis. Šiai problemai spręsti gali būti naudojama paprasta trijų momentų aproksimacija panaudojant sudėtingesnę aproksimaciją, kuri atsižvelgia į pradinio paraiškų aptarnavimo skirstinio formą.

Tarkime, kad turime tipo *M*/*G*/c aptarnavimo sistemą, t.y. ateinančių paraiškų srautas yra paprasčiausias (Puasono) su intensyvumu λ, o paraiškos aptarnavimo laikas S yra pasiskirstęs pagal apibendrintą tikimybinio skirstinio funkciją $G(t)$. Yra priimama sąlyga: $\rho = \lambda E(S)/c < 1$. Sistemų, kurios yra aprašomos tokiomis grandinėmis, analizei nėra jokių paprastų analizinių sprendinių. Vienas iš galimų būdų šiai problemai spręsti yra aproksimacijos, kuri gaunama taikant eksponentinių skirstinių sąsukas ir apjungimus. Tokiu būdu Markovo grandinė su baigtiniu būsenų skaičiumi gali nusakyti sistemos evoliuciją laike.

Tarkime, kad $m_k, k = \overline{1,3}$ nusako k-tąjį ne centrinį momentą (pvz. $E[S^k]$, kur S yra atsitiktinis aptarnavimo laiko kintamasis). Tuomet apibrėžiame naują kintamąjį Y, tokiu būdu:

$$Y = \begin{cases} Y_1, & su\ tikimybe\ p_2 \\ Y_1 + Y_2, & su\ tikimybe\ p_1 p_2 \\ \dots & \\ Y_1 + Y_2 + \dots + Y_2, & su\ tikimybe\ {p^{n-1}}_1 p_2 \\ \dots & \end{cases} \qquad (\mathbf{1.34.})$$

Kur $Y_i, i = 1{,}2$ yra atsitiktiniai dydžiai pasiskirstę pagal eksponentinį skirstinį, kurių vidurkiai yra atitinkamai lygūs $1/\mu_1$ ir $1/\mu_2$, $p_1 + p_2 = 1$. Atsitiktinis dydis Y yra lygus atsitiktinių dydžių sumai sudarytai iš N sumuojamų narių. Kur N - yra neneigiamas, sveikas skaičius, pasiskirstęs pagal geometrinį skirstinį su $E(N) < \infty$. Šio skaičiaus tikimybinė tankio funkcija pateikta (**1.35.**) išraiškoje.

$$f(y) = p_2\mu_1\left(\frac{\mu_2-\mu_1}{\mu_2p_2-\mu_1}e^{-\mu_1 y} - \frac{\mu_2p_1}{\mu_2p_2-\mu_1}e^{-\mu_2p_2y}\right) \qquad (\mathbf{1.35.})$$

Kuomet $\mu_1 = \mu_2$, Y tikimybinio tankio funkcija bus $f(y) = p_2\mu_2e^{-\mu_2p_2y}$.

Apibrėžkime dar vieną kintamąjį X, tokiu būdu:

$$X = \begin{cases} X_1 & su\ tikimybe\ p_2 \\ X_1 + X_2 & su\ tikimybe\ p_1 \end{cases} \qquad (\mathbf{1.36.})$$

Kur X_1 ir X_2 yra nepriklausomi atsitiktiniai dydžiai pasiskirstę pagal eksponentinį skirstinį su parametrais μ_1 ir μ_2p_2, $p_1 + p_2 = 1$.

Įrodysime, kad kintamieji Y ir X, aprašyti (**1.34.**) ir (**1.36.**) lygtimis yra ekvivalentūs parodydami, kad šių dydžių tikimybinių tankių funkcijos yra lygios.

Nesunku patikrinti, kad X tankio funkcija yra:

$$f(x) = \mu_1e^{-\mu_1x} + \frac{p_1\mu_1}{p_1\mu_2-\mu_1}(\mu_1e^{-\mu_1x} - p_2\mu_2e^{-\mu_2p_2x}) \qquad (\mathbf{1.37.})$$

arba

$$f(x) = p_2\mu_1\left(\frac{\mu_2-\mu_1}{\mu_2p_2-\mu_1}e^{-\mu_1 x} - \frac{\mu_2p_1}{\mu_2p_2-\mu_1}e^{-\mu_2p_2x}\right) \qquad (\mathbf{1.38.})$$

Akivaizdu, kad (**1.38.**) ir (**1.35.**) yra lygios.

Reikia pastebėti, kad paraiškos aptarnavimo laikas, išreikštas atsitiktiniu kintamuoju (**1.36.**), leidžia pritaikyti automatinį Markovo grandinėmis aprašytų aptarnavimo sistemų skaitiniais modeliais konstravimo metodą. O tą padaryti su (**1.34.**) yra neįmanoma.

Momentų palyginimo metodas skirstinių aproksimacijai yra plačiai taikomas eilių aproksimacijai. Pirmi trys momentai bet kurio neišsigimusio skirstinio, kurio apibrėžimo sritis apima ir $[0, \infty)$, gali būti prilyginti (**1.38.**) išraiška aprašytam skirstiniui.

Norint apskaičiuoti aproksimacijos parametrų μ_1, μ_2, p_1 ir p_2 reikšmes, reikia išspręsti sudėtingą netiesinių lygčių sistemą (**1.39.**):

$$\begin{cases} \dfrac{1! \cdot p_2\mu_1}{\mu_2p_2-\mu_1}\left(\dfrac{\mu_2-\mu_1}{\mu_1^2} - \dfrac{\mu_2p_1}{\mu_2^2p_2^2}\right) = m_1; \\ \dfrac{2! \cdot p_2\mu_1}{\mu_2p_2-\mu_1}\left(\dfrac{\mu_2-\mu_1}{\mu_1^3} - \dfrac{\mu_2p_1}{\mu_2^3p_2^3}\right) = m_2; \\ \dfrac{3! \cdot p_2\mu_1}{\mu_2p_2-\mu_1}\left(\dfrac{\mu_2-\mu_1}{\mu_1^4} - \dfrac{\mu_2p_1}{\mu_2^4p_2^4}\right) = m_3; \\ p_1 + p_2 = 1. \end{cases} \qquad (\mathbf{1.39.})$$

Kurios sprendiniai yra:

$$\mu_2 = \frac{g_2 - g_1^2}{g_1^3 - 2g_1g_2 + g_3};\ \mu_1 = \frac{1 - \mu_2 g_1 \pm \sqrt{(1-\mu_2 g_1)^2 + 4\mu_2^2(g_2 - g_1^2)}}{2g_1 - 2\mu_2(g_2 - g_1^2)}$$
$$p_1 = \frac{\mu_2(\mu_1 g_1 - 1)}{\mu_2(\mu_1 g_1 - 1) + \mu_1};\ p_2 = \frac{\mu_1}{\mu_2(\mu_1 g_1 - 1) + \mu_1} \qquad (\mathbf{1.39.})$$

Kur $g_k = \frac{m_k}{k!}, k = \overline{1,3}$.

1.4 Matricos kompaktiška forma

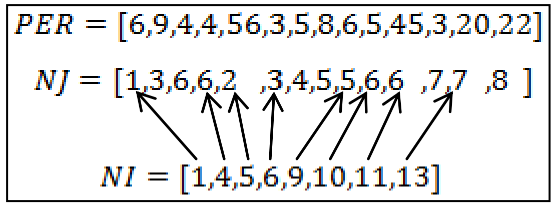
Aptarnavimo sistemų būsenų perėjimo iš vienos į kitą intensyvumų matricos dažniausiai būna „retos“, t.y. turi daug nulinių elementų. Kadangi nėra tikslo saugoti didelius kiekius nereikalingos informacijos ir operuoti ja, dažnai yra taikomos matricų kompaktiškos formos ir modifikuojami algoritmai darbui su jomis. Tokiu būdu „retos“ matricos gali būti „suspaustos“, ko pasėkoje galima sutaupyti pakankamai didelius kiekius kompiuterio atminties.

Viena iš populiariausių matricos kompaktiškų formų, kuri ir bus pritaikyta programinėje įrangoje saugoti perėjimų intensyvumų matricų reikšmes, yra suspaustos eilutės saugojimo algoritmas.

Šio algoritmo esmė yra matricos reikšmes saugoti trijų vektorių pagalba: eilučių indeksų vektorius NI, stulpelių indeksų vektorius NJ ir matricos reikšmių vektorius PER. Vienmatis masyvas PER yra sudaromas taip, kad visos matricos nenulinės reikšmės yra paeiliui (iš kairės į dešinę stulpelių atžvilgiu ir eilučių didėjimo tvarka) išsaugojamos šiame masyve (žr. 1.2. pav.), atitinkamai išsaugojant šių reikšmių stulpelių indeksus vienmačiame masyve NJ, kurio ilgis sutampa su masyvu PER. Kiekvienas masyvo NI elementas yra nuoroda į pirmą nenulinį kiekvieno matricos eilutės elementą vektoriuose NJ ir PER. 1.2. pav. pateiktos matricos kompaktiška forma yra atvaizduota 1.3. pav.

6	0	9	0	0	4	0	0
0	0	0	0	0	4	0	0
0	56	0	0	0	0	0	0
0	0	3	5	8	0	0	0
0	0	0	0	6	0	0	0
0	0	0	0	0	5	0	0
0	0	0	0	0	45	3	0
0	0	0	0	0	0	20	22

1.2. pav. Matricos elementų nuskaitymo tvarka



1.3. pav. Matricos kompaktiška forma

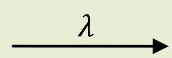
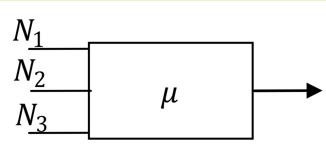
2 PROGRAMINĖ REALIZACIJA IR VARTOTOJO INSTRUKCIJA

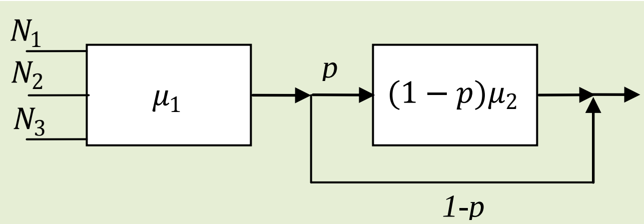
2.1 Aptarnavimo sistemos aprašymas programinėje įrangoje

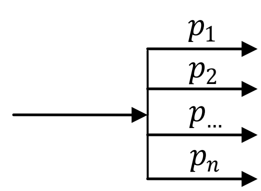
2.1.1 Aptarnavimo sistemos komponentai

Programinėje įrangoje buvo sukurti specialūs komponentai, kurių pagalba būtų galima aprašyti daugumą aptarnavimo sistemų. Detali informacija apie juos yra pateikta toliau esančioje lentelėje (2.1. lent.).

Lentelė 2.1. Sistemos komponentai

Komponentas	**Komp. programoje (*Type*)**	**Žymėjimas**
Paraiškų šaltinis	Stream	
Parametrai:	Paraiškų generavimo intensyvumas λ (*Intensivity*); Paraiškų generavimo skirstinys (*Distribution*); Komponento, į kurį siunčiamos sugeneruotos paraiškos, numeris (*To ID*); Komponento, į kurį siunčiamos sugeneruotos paraiškos, prioritetinės eilės numeris (*To row*);	
Veikimas:	Komponentas, pagal nurodytą paraiškų generavimo skirstinį su intensyvumu λ, generuoja paraiškas į komponento, kurio numeris nurodytas (*To ID*) parametru, prioritetinę eilę, kurios numeris nurodomas parametru (*To row*).	
Aptarnavimo įrenginys	Service	
Parametrai:	Paraiškų aptarnavimo intensyvumas μ (*Intensivity*); Paraiškų aptarnavimo skirstinys (*Distribution*); Aukščiausio prioriteto paraiškų eilės ilgis (*L1*);	

Komponentas	Komp. programoje (*Type*)	Žymėjimas
		Vidutinio prioriteto paraiškų eilės ilgis (*L2*); Žemo prioriteto paraiškų eilės ilgis (*L3*); Komponento, iš kurio ateina paraiškos, numeris (*From ID*); Komponento, į kurį siunčiamos aptarnautos paraiškos, numeris (*To ID*); Komponento, į kurį siunčiamos aptarnautos paraiškos, prioritetinės eilės numeris (*To row*); Komponento gedimo intensyvumas (*Parameters-> break_int*) Komponento pataisymo po gedimo intensyvumas (*Parameters->fix_int*) Paraiškų, kurios palieka aptarnavimo įrenginį, intensyvumas (*Parameters-> leave_int*), kuomet aukščiausio prioriteto eilė yra ilgesnė arba lygi (*Parameters-> leave_n1*) arba vidutinio prioriteto eilė yra ilgesnė arba lygi (*Parameters-> leave_n2*) arba žemo prioriteto eilė yra ilgesnė arba lygi (*Parameters-> leave_n3*) nurodytoms parametrų reikšmėms.
Veikimas:		Komponentas aptarnauja paraiškas, kurios yra prioritetinėse eilėse. Pirmiausia yra aptarnaujamos paraiškos iš aukščiausio prioriteto eilės, paskui iš vidutinio, o galiausiai iš žemo. Jeigu yra aptarnaujama vidutinio prioriteto paraiška ir ateina aukščiausio prioriteto paraiška, aptarnavimas yra nutraukiamas, o vidutinio prioriteto paraiška gražinama į savo eilę ir pradedama aptarnauti aukščiausio prioriteto paraiška. Jeigu yra aptarnaujama žemo prioriteto paraiška ir ateina vidutinio prioriteto paraiška, tuomet yra baigiama aptarnauti žemo prioriteto paraišką, o tik tada pradeda aptarnauti vidutinio prioriteto paraišką. Paraiškos aptarnaujamos pagal nurodytą paraiškų aptarnavimo skirstinį su intensyvumu μ ir išsiunčiamos į komponento, kurio numeris nurodytas (*To ID*) parametru prioritetinę eilę, kurios numeris nurodomas parametru (*To row*). Taip pat šis komponentas gali sugesti su intensyvumu (*Parameters-> break_int*) ir gali būti sutaisytas su atitinkamai intensyvumu (*Parameters-> fix_int*). Kuomet komponentas sugenda ir jame yra aptarnaujama paraiška, ji yra grąžinama į savo prioritetinę eilę, jeigu ji nėra užpildyti, priešingu atveju paraiška palieka sistemą. Be to, aptarnavimo įrenginį gali palikti paraiškos iš prioritetinių eilių, jeigu eilėse paraiškų kiekis pasiekia parametrais (*Parameters-> leave_n1, leave_n2, leave_n3*) nurodytas reikšmes. Paraiškos palieka prioritetinę eilę su intensyvumu, kuris nurodomas parametru (*Parameters-> leave_int*).
Aproksimuojantis aptarnavimo įrenginys	Service-Dual	

Komponentas	**Komp. programoje (*Type*)**	**Žymėjimas**
Parametrai:		Galioja tie patys parametrai, kurie yra aprašyti komponentui „Aptarnavimo įrenginys“ (išskyrus gedimo ir pataisymo, bei paraiškų palikimo prioritetines eiles parametrus) bei du papildomi: Paraiškų aptarnavimo intensyvumas 2 μ_2 (*Parameters->intens_2*); Tikimybė, kad aptarnauta pirmajame komponento aptarnavimo įrenginyje, bus aptarnaujama ir antrajame įrenginyje (*Parameters->probability*)
Veikimas:		Veikimo principas išlieka toks pats kaip ir aptarnavimo įrenginio (išskyrus įrenginio gedimą, jo pataisymą ir paraiškų pasišalinimo iš eilės, kuomet ji pasiekia atitinkamą ilgį), tik vietoje to, kad aptarnauta paraiška yra išsiunčiama į kitą komponentą, ji gali būti dar pakartotinai aptarnaujama su tikimybe p sekančiame aptarnavimo įrenginyje, kuris paraiškas aptarnauja su intensyvumu $(1-p\,)\mu_2$, o taip pat su tikimybe $1-p$ gali palikti komponentą pakartotinai neaptarnauta.
Tikimybinis skaidymas	Splitter	
Parametrai:		Komponento, iš kurio ateina paraiškos, numeris (*From ID*); Paraiškų paskirstymo į komponentus parametras (*Parameters->splits*)
Veikimas:		Paraiškos ateina iš komponento, kurio ID nurodytas parametru (*From ID)* ir pagal nurodyta tikimybinį išskaidymą yra paskirstomos i kitus komponentus, pagal parametrą (*Parameters->splits*). Pastarasis parametras yra sudaromas taip: pirmiausiai yra nurodoma tikimybė su kuria paraiška yra pasiunčiama į atitinkamą komponentą, nurodomas komponento, į kurį siunčiama paraiška, ID numeris, o galiausiai jo prioritetinės eilės numeris. Visos šios reikšmės yra atskiriamos kableliu, o kiekvienas papildomas skaidymas kabliataškiu. Pavyzdžiui, norėdami aprašyti tokį tikimybinio skaidymo komponentą, kuris atėjusią paraišką su tikimybė 0.4 pasiunčia į antro komponento pirmąją prioritetinę eilę, o su tikimybe 0.6 ją pasiunčia į trečio komponento antrą prioritetinę eilę, parametras būtų toks: *splits = [0.4, 2, 1; 0.6, 3, 2]*.

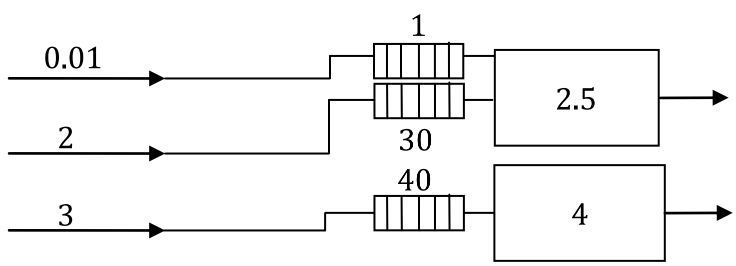
Pastaba: visi parametrai programinėje įrangoje gali būti nurodomi dvejopai, t.y. jeigu paaiškinime parametras nurodomas be „Parameters->“, tuomet, to parametro reikšmė turi būti įvedama į atitinkamą to parametro stulpelį, priešingu atveju, parametras yra vedamas į „Parameters“ stulpelį, parašant parametro pavadinimą, tarpą, lygybės ženklą, tarpą ir laužtiniuose skliaustuose nurodant to parametro reikšmę. Pvz. „break_int = [0.01] fix_int = [0.7]“.

Panaudojant šiuos komponentus ir nusakant ryšius tarp jų, programinėje įrangoje galima sukurti daugelio aptaranavimo sistemų schematines struktūras, kuriomis remiantis yra atliekamas modeliavimo procesas.

2.1.2 Aptarnavimo sistemos aprašymas panaudojant komponentus

Tarkime, kad programinėje įrangoje norime aprašyti tokią sistemą:

Pasienio kontrolės postas turi du terminalus, kuriuose yra tikrinami automobiliai. Pirmas terminalas yra skirtas tik lengviesiems automobiliams bei spec. (išskirtiniam) transportui, o antrasis krovininiams automobiliams. Lengvieji automobiliai į pasienį atvyksta laiko momentais, kurie pasiskirstę pagal Puasono skirstinį su intensyvumu 2, krovininiai automobiliai su intensyvumu 3, o spec. transportas su intensyvumu 0.01. Lengvųjų automobilių eilė gali būti iki 30 automobilių, spec. Transporto tik 1, ir jie bus aptarnaujami viename pasienio punkte pagal eksponentinį skirstinį su intensyvumu 2.5. Krovininių automobilių eilė gali siekti iki 40, kurie bus aptarnaujami pagal eksponentinį skirstinį su intensyvumu 4. Šiuo atveju aptarnavimo sistemos schema gali būti tokia (2.1. pav.):



2.1. pav. Aptarnavimo sistemos schema

Programinėje įrangoje aprašysime nagrinėjamą aptarnavimo sistemą, pagal pateiktą apibūdinimą ir schematinę struktūrą, panaudojant programinius komponentus.

Aprašinėjant aptarnavimo sistemas programinėje įrangoje, patogiausia yra pradėti nuo sistemą sudarančių komponentų sukūrimo, o tik tada nusakyti ryšius tarp jų bei nurodyti parametrus. Kaip galima pastebėti iš 2.1. pav. aptarnavimo sistemą galima išskaidyti į tris Puasoninius paraiškų šaltinius ir du aptarnavimo įrenginius.

Taigi, pirmuoju žingsniu sukuriame tris paraiškų šaltinius (*stream*) ir du aptaravimo įrenginius (*service*) pasinaudojant „System“ polangyje esančiu mygtuku „New“ ir nurodant kiekvienam komponentui tipą (*Type* stulpelis) sistemos komponentų lentelėje. Po šių veiksmų pagrindinis programos langas atrodo taip (2.2. pav.):

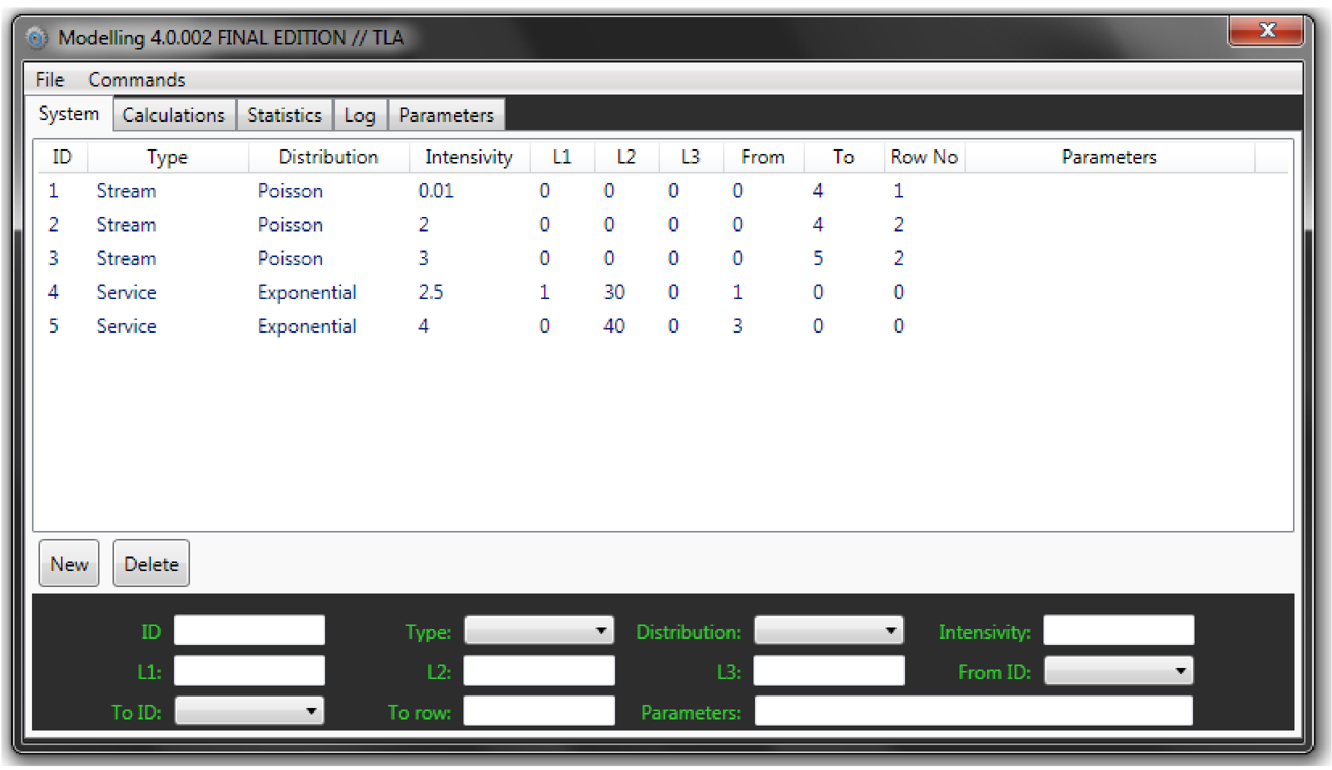
2.2. pav. Pagrindinis programos langas po komponentų sukūrimo

Sekančiu žingsniu reikia nurodyti kiekvieno komponento parametrus ir savybes. Tarkime, kad komponentas, kurio ID yra 1, atitiks išskirtinių automobilių srautą, lengvųjų automobilių srautą atitiks komponentas, kurio ID – 2, o krovininių – 3. Kadangi visų šių automobilių srautai yra pasiskirstę pagal Puasono skirstinį, tai visiems trims komponentams reikia parametrą *Distribution* parinkti „Poisson“ ir atitinkamai nurodyti intensyvumus 0.01, 2 ir 3. Tarkime, kad komponentas, kurio ID yra 4 atitiks pasienio punktą, aptarnaus lengvuosius automobilius ir spec. automobilius, tad jo aukščiausios prioritetinės eilės ilgis turi būti lygus 1 (parametras *L1*), o vidutinės – 30 (parametras *L2*), o komponentas, kurio ID yra 5, atitiks pasienio punktą skirtą krovininiams automobiliams su vidutinio prioriteto eile, kurios ilgis 40 (parametras *L2*). Abiejų pastarųjų komponentų skirstiniai yra eksponentiniai (parametro *Distribution* reikšmė „Exponential“) su atitinkamais intensyvumais 2.5 ir 4.

Paskutinis žingsnis atliekant sistemos aprašymą programinėje įrangoje – ryšių tarp sistemos objektų nusakymas. Kaip jau buvo minėta pavyzdžio formuluotėje, specialiųjų ir lengvųjų automobilių srautai yra nukreipiami į pirmąjį aptarnavimo įrenginį, kurio ID – 4, tik į skirtingas prioritetines eiles.

Tokiu atveju pirmam (ID = 1) ir antram komponentui (ID = 2) parametrą *To* parenkame lygų 4, tik pirmajam parametrą *To row* suteikiame lygų 1, o antrajam - 2. Atitinkamai ketvirto komponento (ID = 4) parametrą *From* parenkame lygų 1 arba 2. Krovininių automobilių srautui (ID = 3) parametrą *To* parenkame lygų 5, o *To row* – 2, kadangi kroviniai automobiliai yra aptarnaujami vidutinio prioriteto eilėje, atskirame pasienio punkte negu lengvieji ir spec. automobiliai. Tai pat, reikia yra krovininių automobilių aptarnavimo įrenginiui (ID = 5) nurodyti parametrą *From* lygų 3, kuris nurodo, kad į jį paraiškos ateina iš krovininių automobilių srauto.

Nagrinėjamos aptarnavimo sistemos galutinis aprašymas programinėje įrangoje pateiktas toliau esančiame paveiksle (2.3. pav.).



2.3. pav. Aptarnavimo sistemos aprašymas

2.1.3 Sistemos būsenos sudarymas

Sistemos būsena yra sudaroma priklausomai pagal sistemą sudarančių aptarnavimo ir aproksimuojančių aptarnavimo įrenginių skaičiaus. Kiekvienas paprastas aptarnavimo įrenginys sistemos būseną papildo iki penkių komponentų: trys komponentai yra skirti paraiškų prioritetinėse eilėse kiekiui saugoti, vienas komponentas - nusakyti, kurio prioriteto paraiška yra aptarnaujama aptarnavimo įrenginyje, o penktasis nusakyti ar komponentas yra sugedęs ar ne. Aproksimuojančio aptarnavimo įrenginio atveju, sistemos būsena gali būti papildyta iki šešių komponentų, kurių pirmi penki sutampa su aptarnavimo įrenginio komponentais, tik priešingai nuo paprasto aptarnavimo įrenginio, išplėtotame įrenginyje aptarnaujamos paraiškos būsenai nusakyti yra naudojami du komponentai, kadangi patį išplėtotą aptarnavimo įrenginį sudaro du paprasti aptarnavimo įrenginiai, tad antrojo užimtumas nusakomas šeštajame komponente.

Pasinaudokime anksčiau apibrėžtu aptarnavimo sistemos pavyzdžiu (žr. 2.1.2. skyrių). Tokios sistemos būsena yra sudaryta iš penkių komponentų ir yra išreiškiama (**2.1.**) išraiška.

$$(0;0;0;0;0) \qquad (\mathbf{2.1.})$$

Sudarant šią būseną, programinė įranga pirmiausiai nuskaito visus sistemos komponentus, kurie yra aprašomi *System* polangyje, ir analizuoja juos didėjimo tvarka pagal identifikacinį numerį *ID*. Kaip jau buvo minėta anksčiau, sistemos būseną sudaro tik komponentai, kurie nusako aptarnavimo įrenginius, taigi, nagrinėjamo pavyzdžio atveju, pirmas aptarnavimo įrenginys, kuris bus įtraukiamas į sistemos būseną, turi *ID* lygų 4. Kaip galime pastebėti iš sistemos aprašymo 2.1. pav., šis aptarnavimo įrenginys nėra išplėtotas ir yra naudojamos tik dvi prioritetinės eilės, todėl, sistemos būseną papildys trimis komponentais, kur pirmi du nusakys aptarnavimo įrenginio prioritetinių eilių ilgius, o trečias nurodys, kurio prioriteto paraiška yra aptarnaujama įrenginyje. Programinė įranga yra sudaryta taip, kad pirmiausia įtraukia įrenginio prioritetinių eilių komponentus, o tik paskui įrenginio aptarnaujamos būsenos statuso komponentą. Taigi, po pirmojo žingsnio sistemos būsena yra sudaryta iš trijų komponentų: $(0;0;0)$, kur pirmasis nurodo paraiškų skaičių aukščiausio prioriteto eilėje, antrasis – vidutinio, o trečiasis, kokio prioriteto paraiška yra aptarnaujama įrenginyje.

Tokiu pačiu principu yra prijungiami ir kito aptarnavimo įrenginio sistemos būsenos komponentai, kurie šiuo atveju yra tik du, t.y. vidutinio prioriteto eilės ilgis ir aptarnavimo įrenginyje aptarnaujamos paraiškos prioriteto numeris. Papildžius šiais komponentais, galiausiai yra suformuojama sistemos būsena, kuri nusako nagrinėjamą aptarnavimo sistemą, bet kuriuo laiko momentu (**2.1.**).

2.1.4 Sistemos įvykių aibės sudarymas

Visi sistemos komponentai, kurie yra naudojami aptarnavimo sistemoms modeliuoti, turi aprašytus (suprogramuotus) galimus įvykius, kuriuos jie gali įvykdyti. Toliau esančioje lentelėje yra pateiktas kiekvieno komponento įvykių sąrašas (lent. 2.2.).

Lentelė 2.2. Sistemos komponentų įvykiai

Komponentas	Įvykiai
Paraiškų šaltinis (Stream)	e_1 – Paraiškų šaltinis sugeneruoja paraišką į nurodyta objektą;
Aptarnavimo įrenginys (Service)	e_1 – Aptarnavimo įrenginys aptarnauja paraišką;
	e_2 – Aptarnavimo įrenginys sugenda;
	e_3 – Aptarnavimo įrenginys pataisomas;
	e_4 – Paraiška palieka aukščiausio prioriteto eilę ir sistemą, jeigu paraiškų eilė didesnė už n1;
	e_5 – Paraiška palieka vidutinio prioriteto eilę ir sistemą, jeigu paraiškų eilė didesnė už n2;
	e_6 – Paraiška palieka vidutinio prioriteto eilę ir sistemą, jeigu paraiškų eilė didesnė už n3;
Aproksimuojantis aptarnavimo įrenginys (Service-Dual)	e_1 – Aptarnavimo įrenginys aptarnauja paraišką pirmajame aptarnavimo įrenginyje ir ji palieka įrenginį;
	e_2 – Aptarnavimo įrenginys aptarnauja paraišką pirmajame aptarnavimo įrenginyje ir ji pasiunčiama į antrąjį aptarnavimo įrenginį;
	e_3 – Aptarnavimo įrenginys aptarnauja paraišką iš antrojo aptarnavimo įrenginio ir ji palieka įrenginį;
Tikimybinis skaidymas (splitter)	Komponentas skirtas tik nukreipi paraiškas iš vieno komponento į kelis kitus pagal tikimybinį skaidymą. Aprašytų įvykių neturi.

Anksčiau pateiktam pavyzdžiui (2.1. pav.), remiantis 2.1. ir 2.2. lentelėmis, sudarysime galimų įvykių aibę, kurią vėliau panaudosime visų sistemos būsenų generavimui.

Programinė įranga visų pirma iš programos *System* polangio (2.3. pav.) nuskaito visus komponentus ir analizuoja kiekviena paeiliui. Priklausomai nuo kiekvieno komponento tipo ir jo parametrų, visų galimų įvykių aibė yra papildoma įvykiais, kuriuos gali atlikti nagrinėjamas aptarnavimo sistemos struktūrinėje schemoje aprašytas komponentas.

2.3. pav. pateikto pavyzdžio atveju, programinė įranga pirmiausia sistemos įvykių aibę E papildo trimis įvykiais, t.y. kiekvienas paraiškų šaltinis gali sugeneruoti paraišką. Tokiu atveju sistemos įvykių aibė bus:

$E = (e_1, e_2, e_3)$,

čia:

e_1 – pirmas paraiškų šaltinis sugeneruoja paraišką į pirmo aptarnavimo įrenginio aukščiausio prioriteto eilę, e_2 – antras paraiškų šaltinis sugeneruoja paraišką į pirmo aptarnavimo įrenginio vidutinio prioriteto eilę, e_3 – trečias paraiškų šaltinis sugeneruoja paraišką į antro aptarnavimo įrenginio vidutinio prioriteto eilę.

Toliau programinė įranga analizuoja aptarnavimo įrenginius ir priklausomai nuo jų parametrų įvykių aibė papildoma įvykiais. Pirmasis aptarnavimo įrenginys turi dvi naudojamas prioritetines eiles, bet jis negali sugesti, o taip pat eilėse nėra „nekantrių paraiškų“. Tad toks aptarnavimo įrenginys pridės tik vieną naują įvykį e_4 – pirmas aptarnavimo įrenginys aptarnauja paraišką.

Atitinkamai antrasis aptarnavimo įrenginys papildo aibę irgi tik vienu įvykiu: e_5 – antras aptarnavimo įrenginys aptarnauja paraišką.

Galiausiai sistemos visų galimų įvykių aibė yra tokia: $E = (e_1, e_2, e_3, e_4, e_5)$, kurią programinė įranga naudoja generuojant sistemos būsenas ir perėjimus tarp jų.

2.1.5 Sistemos būsenų ir perėjimų intensyvumų tarp jų generavimas

Sistemos būsenų generavimas yra atliekamas po to, kai programinė įranga sugeneruoja sistemos pradinę būsena ir visų galimų įvykių aibę. Toliau pratęskime aptarnavimo sistemos, aprašytos 2.1. pav. pavaizduota schema, analizę. Kaip jau buvo minėta ankstesniame skyriuje (žr. 2.1.3. skyrių), šios sistemos pradinė būsena yra sudaryta iš penkių komponentų ir yra išreikšta tokia forma: $(0; 0; 0; 0; 0)$, o taip yra žinoma ir visų galimų įvykių aibė $E = (e_1, e_2, e_3, e_4, e_5)$ (žr. 2.1.4. skyrių). Remdamiesi šiais duomenimis galime sugeneruoti visas galimas sistemos būsenas, o taip pat ir nustatyti perėjimų intensyvumus tarp jų.

Pirmuoju žingsniu programinė įranga pradinei būsenai bando įvykdyti visus įvykius iš galimų įvykių aibės sistemos būsenai $(0; 0; 0; 0; 0)$.

Esant būsenai $(0; 0; 0; 0; 0)$ ir įvykus įvykiui e_1 (pirmas paraiškų šaltinis sugeneruoja paraišką į pirmo aptarnavimo įrenginio aukščiausio prioriteto eilę), sistemos būsena pasikeičia į $(0; 0; 1; 0; 0)$, kadangi pirmojo aptarnavimo įrenginio eilė yra tuščia (būsenos pirmasis elementas), o taip pat jame nėra aptarnaujama jokia paraiška (būsenos trečias elementas), paraiška yra iškarto paduoda į aptarnavimo sistemą aptarnavimui ir į trečią būsenos vektoriaus komponentę yra įrašomas aptarnaujamos paraiškos prioritetinis numeris. Taip pat yra žinoma, kad pirmasis paraiškų šaltinis paraiškas generuoja su intensyvumu 0.01, kuris ir nusako perėjimo intensyvumą iš būsenos $(0; 0; 0; 0; 0)$ į būseną $(0; 0; 1; 0; 0)$.

Visos sistemos būsenos yra saugojamos būsenų vektoriuje S, kuris yra vis papildomas nesikartojančiomis būsenomis, būsenų generavimo metu. Pradinis vektoriaus S elementas visada būna pradinė aptarnavimo sistemos būsena, kuri nagrinėjamo pavyzdžio atveju yra $(0; 0; 0; 0; 0)$, taigi ir $S = \{(0; 0; 0; 0; 0)\}$.

Kadangi sugeneruota nauja būsena $(0; 0; 1; 0; 0)$ dar nėra įtraukta į vektorių S, tad papildome jį šia būsena. Perėjimų intensyvumus tarp būsenų saugojame matricoje PER, kurios eilė yra $n\ x\ n$, kur $n -$ būsenų vektoriaus S elementų skaičius. Taigi, papildžius būsenų vektorių S, atitinkamai padidėja ir perėjimų matricos eilė.

Šio atveju, kai vektorius S yra papildomas būsena $(0; 0; 1; 0; 0)$, ko pasėkoje perėjimų matrica iš $\mathrm{PER} = (0)$ praplečiama iki $\mathrm{PER} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. Kiekvienas perėjimų matricos elementas saugoja tokia informaciją savyje: matricos eilutės numeris nusako būsenos numerį iš kurios pereinama, o stulpelio numeris nusako į kurią būsena pereinama su intensyvumu išsaugotu matricos elemente. Taigi, papildžius sistemos būsenų vektorių S ir praplėtus perėjimų matricą, išsaugojame perėjimo intensyvumą iš būsenos $(0; 0; 0; 0; 0)$ į būseną $(0; 0; 1; 0; 0)$.

Perėjimų matricos eilutės ir stulpelio numeris atitinka vieną vektoriaus S būseną. Taigi, norint išsaugoti perėjimo intensyvumą iš būsenos $(0; 0; 0; 0; 0)$ į būseną $(0; 0; 1; 0; 0)$, visų pirma nustatome jų eilės numerius būsenų vektoriuje, kurie atitinkamai yra lygūs 1 ir 2. Tuomet šį intensyvumą išsaugosime elemente $PER_{12} = 0.01$ ir $\mathrm{PER} = \begin{pmatrix} 0 & 0.01 \\ 0 & 0 \end{pmatrix}$.

Visi ankščiau aprašyti veiksmai yra atliekami ir su kitais įvykiais. Sistemai esant būsenoje $(0; 0; 0; 0; 0)$ ir įvykus įvykiui e_2 (antras paraiškų šaltinis sugeneruoja paraišką į pirmo aptarnavimo įrenginio vidutinio prioriteto eilę), yra sugeneruojama nauja sistemos būsena $(0; 0; 2; 0; 0)$. Kadangi pirmas aptarnavimo įrenginys bei jo eilė yra tuščia, paraiška iškarto yra aptarnaujama įrenginyje. Atitinkamai yra papildomas vektorius S nauja būsena ir praplečiama matrica PER:

$$S = \{(0; 0; 0; 0; 0), (0; 0; 1; 0; 0), (0; 0; 2; 0; 0)\}\ , \quad \mathrm{PER} = \begin{pmatrix} 0 & 0.01 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

$PER_{13} = 2$, kadangi antrasis paraiškų šaltinis generuoja paraiškas su intensyvumu 2.

Sistemai esant būsenoje $(0; 0; 0; 0; 0)$ ir įvykus įvykiui e_3 yra sugeneruojama nauja būsena $(0; 0; 0; 0; 2)$ į kurią pereinama su intensyvumu lygiu 3. Tuomet būsenų vektorius ir perėjimų matrica bus:

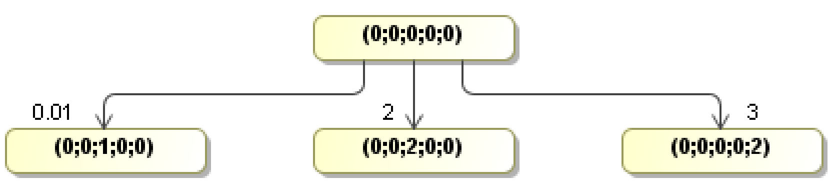
$$S = \{(0; 0; 0; 0; 0), (0; 0; 1; 0; 0), (0; 0; 2; 0; 0), (0; 0; 0; 0; 2)\}; \qquad \mathrm{PER} = \begin{pmatrix} 0 & 0.01 & 2 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Įvykiai e_4 ir e_5 sistemai esant būsenoje $(0; 0; 0; 0; 0)$ įvykti negali, nes aptarnavimo įrenginiuose nėra aptarnaujamų paraiškų. Glaustai visą informaciją, apie naujai sugeneruotas būsenas galime pateikti 2.3. lentelėje.

Lentelė 2.3. Sugeneruotos būsenos pirmoje iteracijoje

Būsena	Įvykis	Rezultatas	Int.	Paaiškinimas
$(\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$	e_1	$(0; 0; 1; 0; 0)$	0.01	Atėjo aukščiausio prioriteto paraiška į pirmąjį aptarnavimo įrenginį.
$(\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$	e_2	$(0; 0; 2; 0; 0)$	2	Atėjo vidutinio prioriteto paraiška į pirmąjį aptarnavimo įrenginį.
$(\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$	e_3	$(0; 0; 0; 0; 2)$	3	Atėjo vidutinio prioriteto paraiška į antrąjį aptarnavimo įrenginį.
$(\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$	e_4	$(0; 0; 0; 0; 0)$	0	Įvykis negali įvykti.
$(\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$	e_5	$(0; 0; 0; 0; 0)$	0	Įvykis negali įvykti.

Būsenų grafas po pirmosios generavimo iteracijos pavaizduotas toliau esančiame paveiksliuke (2.4. pav.).



2.4. pav. Sistemos būsenų grafas

Antroji būsenų generavimo iteracija, kaip ir visos tolimesnės, visoms prieš tai buvusios iteracijos naujoms būsenoms bando įvykdyti visus galimus įvykius iš įvykių aibės. Generavimo procedūra nutraukiama, tuomet, kai paskutinės iteracijos metu nėra sugeneruojama nė viena nauja būsena.

Antrosios iteracijos metu, priešingai nuo pirmosios, bus panaudotos trys pradinės būsenos, kurios buvo sugeneruotos pirmosios iteracijos metu, t.y. $(0; 0; 0; 1; 0)$, $(0; 0; 0; 2; 0)$, $(0; 0; 0; 0; 2)$. 2.4.

lentelėje glaustai pateiksime informaciją apie šios iteracijos metu naujai sugeneruotas būsenas ir perėjimus tarp jų (naujos būsenos yra pabrauktos).

Lentelė 2.4. Sugeneruotos būsenos antroje iteracijoje

Būsena	Įvykis	Rezultatas	Int.	Paaiškinimas
$(\mathbf{0};\mathbf{0};\mathbf{1};\mathbf{0};\mathbf{0})$	e_1	$\underline{(1;0;1;0;0)}$	0.01	Atėjo aukščiausio prioriteto paraiška į pirmąjį aptarnavimo įrenginį. Paraiška stoja į savo eilę, kadangi aptarnavimo įrenginys yra užimtas.
$(\mathbf{0};\mathbf{0};\mathbf{1};\mathbf{0};\mathbf{0})$	e_2	$\underline{(0;1;1;0;0)}$	2	Atėjo vidutinio prioriteto paraiška į pirmąjį aptarnavimo įrenginį. Paraiška stoja į savo eilę, kadangi aptarnavimo įrenginys yra užimtas.
$(\mathbf{0};\mathbf{0};\mathbf{1};\mathbf{0};\mathbf{0})$	e_3	$\underline{(0;0;1;0;2)}$	3	Atėjo vidutinio prioriteto paraiška į antrąjį aptarnavimo įrenginį.
$(\mathbf{0};\mathbf{0};\mathbf{1};\mathbf{0};\mathbf{0})$	e_4	$(0;0;0;0;0)$	2.5	Aptarnavimo įrenginys aptarnauja paraišką ir ji palieka sistemą.
$(\mathbf{0};\mathbf{0};\mathbf{1};\mathbf{0};\mathbf{0})$	e_5	–	0	Įvykis negali įvykti.
*****				
$(\mathbf{0};\mathbf{0};\mathbf{2};\mathbf{0};\mathbf{0})$	e_1	$(0;1;1;0;0)$	0.01	Atėjo aukščiausio prioriteto paraiška į pirmąjį aptarnavimo įrenginį. Kadangi aptarnavimo sistemoje yra vidutinio prioriteto paraiška, tai ji yra grąžinama į savo eilę ir pradedama aptarnauti atėjusi aukščiausio prioriteto paraiška.
$(\mathbf{0};\mathbf{0};\mathbf{2};\mathbf{0};\mathbf{0})$	e_2	$\underline{(0;1;2;0;0)}$	2	Atėjo vidutinio prioriteto paraiška į pirmąjį aptarnavimo įrenginį. Paraiška stoja į eilę, kadangi aptarnavimo įrenginys yra užimtas.
$(\mathbf{0};\mathbf{0};\mathbf{2};\mathbf{0};\mathbf{0})$	e_3	$\underline{(0;0;2;0;2)}$	3	Atėjo vidutinio prioriteto paraiška į antrąjį aptarnavimo įrenginį.
$(\mathbf{0};\mathbf{0};\mathbf{2};\mathbf{0};\mathbf{0})$	e_4	$(0;0;0;0;0)$	2.5	Aptarnavimo įrenginys aptarnauja paraišką ir ji palieka sistemą.
$(\mathbf{0};\mathbf{0};\mathbf{2};\mathbf{0};\mathbf{0})$	e_5	–	0	Įvykis negali įvykti.
*****				
$(\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{2})$	e_1	$(0;0;1;0;2)$	0.01	Atėjo aukščiausio prioriteto paraiška į pirmąjį aptarnavimo įrenginį.
$(\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{2})$	e_2	$(0;0;2;0;2)$	2	Atėjo vidutinio prioriteto paraiška į pirmąjį aptarnavimo įrenginį.
$(\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{2})$	e_3	$\underline{(0;0;0;1;2)}$	3	Atėjo vidutinio prioriteto paraiška į antrąjį aptarnavimo įrenginį. Paraiška stoja į eilę, kadangi aptarnavimo įrenginys yra užimtas.
$(\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{2})$	e_4	–	0	Įvykis negali įvykti.
$(\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{2})$	e_5	$(0;0;0;0;0)$	4	Aptarnavimo įrenginys aptarnauja paraišką ir ji palieka sistemą.

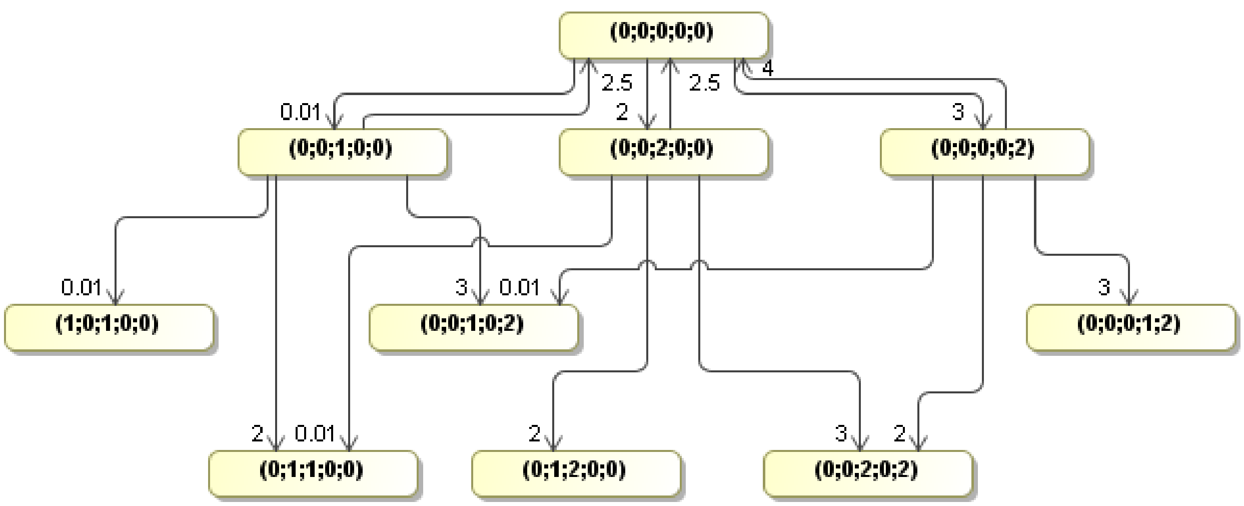
Po šios iteracijos sistemos būsenų vektorius bus sudarytas iš 10 komponentų:

$$S = \{(0;0;0;0;0),(0;0;1;0;0),(0;0;2;0;0),(0;0;0;0;2),(1;0;1;0;0),(0;1;1;0;0),$$
$$(0;0;1;0;2),(0;1;2;0;0),(0;0;2;0;2),(0;0;0;1;2)\}$$

Perėjimų matrica:

$$PER = \begin{pmatrix} 0 & 0.01 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.5 & 0 & 0 & 0 & 0.01 & 2 & 3 & 0 & 0 & 0 \\ 2.5 & 0 & 0 & 0 & 0 & 0.01 & 0 & 2 & 3 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Būsenų grafas atrodys taip, kaip pavaizduota 2.5. pav.



2.5. pav. Sistemos būsenų grafas

Tęsiant tokio tipo iteracinį būsenų generavimą yra sugeneruojamos visos sistemos galimos būsenos ir perėjimų intensyvumai tarp jų.

2.2 Tikimybinių charakteristikų skaičiavimas

Paskutinis aptarnavimo sistemos modeliavimo žingsnis programinėje įrangoje yra tikimybinių charakteristikų apskaičiavimas. Darbe bus įvertintos tokios aptarnavimo įrenginių tikimybinės charakteristikos:

- Tikimybė, kad atėjusi paraiška į aptarnavimo įrenginį bus iš karto aptarnaujama;
- Tikimybė, kad atėjusi paraiška į aukščiausio prioriteto eilę nebus aptarnaujama, o stos į eilę;
- Tikimybė, kad atėjusi paraiška į vidutinio prioriteto eilę nebus aptarnaujama, o stos į eilę;
- Tikimybė, kad atėjusi paraiška į žemo prioriteto eilę nebus aptarnaujama, o stos į eilę;
- Tikimybė, kad atėjusi paraiška į aukščiausio prioriteto eilę nebus aptarnaujama, nes eilė bus pasiekusi savo limitą ir ji paliks sistemą;
- Tikimybė, kad atėjusi paraiška į vidutinio prioriteto eilę nebus aptarnaujama, nes eilė bus pasiekusi savo limitą ir ji paliks sistemą;
- Tikimybė, kad atėjusi paraiška į žemo prioriteto eilę nebus aptarnaujama, nes eilė bus pasiekusi savo limitą ir ji paliks sistemą;
- Tikimybė, kad aptarnavimo įrenginys suges;
- Kiekvienos prioritetinės eilės vidutinę paraiškų eilę.

Kadangi programoje tarp sistemos būsenų aibės ir stacionariųjų tikimybių egzistuoja abipusė vienareikšmė atitiktis $P(S_i) = q_i$, t.y., kad sistema bus būsenoje S_i atitinka tikimybė q_i, apskaičiuojant charakteristikas mums reikia surasti tik sąlygas tenkinančias būsenas.

Pirmu atveju reikia surasti būsenas, kurių visos komponento koordinatės yra lygios 0 (nebūtinai visos sistemos) ir pagal šių būsenų indeksus bus galima nustatyti visas būsenų stacionariąsias tikimybes, nes tik šiais atvejais atėjusi bet kurio prioriteto paraiška iš karto bus aptarnaujama. Suradus šias būsenas belieka tik susumuoti jų stacionariąsias tikimybes.

Nuo antro ir ketvirto atvejais imtinai, reikia surasti visas sistemos būsenas, kuomet nagrinėjama aptarnavimo įrenginio prioritetinė eilė yra didesnė arba lygi 0, tačiau neviršija maksimalios eilės ilgio ir nagrinėjamas aptarnavimo įrenginys yra užimtas. Suradus šias būsenas belieka tik susumuoti jų stacionariąsias tikimybes.

Nuo penkto iki septinto atvejais imtinai, reikia surasti visas sistemos būsenas, kuriose nagrinėjamos aptarnavimo įrenginio prioritetinės eilės ilgis yra lygus maksimaliai galimai reikšmei ir susumuoti šių būsenų stacionariąsias tikimybes.

Aštuntu atveju reikia surasti visas sistemos būsenas, kurių komponentė, žyminti, kad nagrinėjamas aptarnavimo įrenginys yra sugedęs nėra tuščia, ir susumuoti šių būsenų stacionariąsias tikimybes.

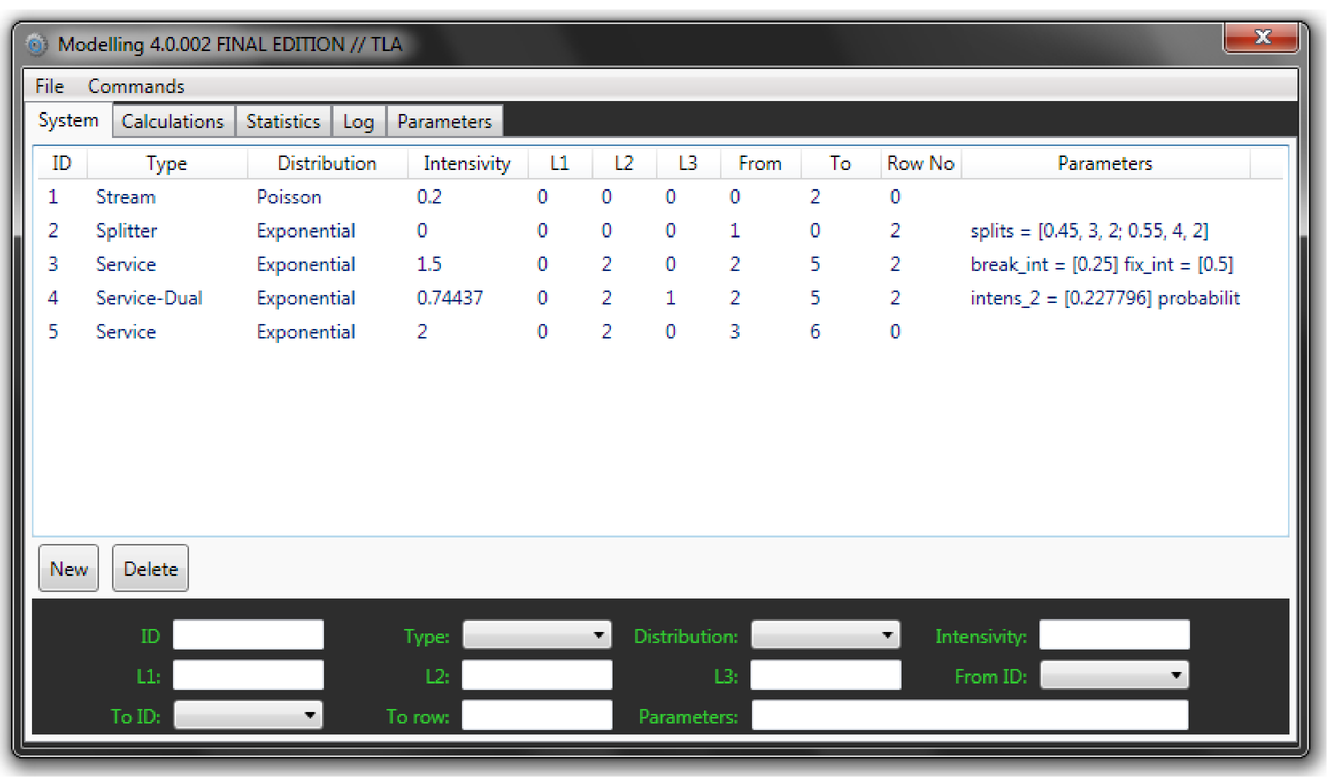
Paskutiniu atveju tikimybinė charakteristika gali būti apskaičiuojama pagal tokią formulę:

$$P_j = \sum_{i=1}^{N} q_i \cdot r_j$$

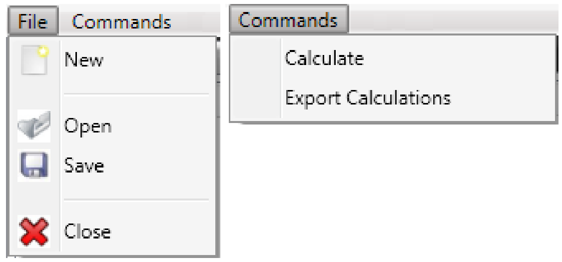
kur N – sistemos būsenų skaičius, q_i – i-tosios būsenos stacionarioji tikimybė, o r_j - i-tosios būsenos vektoriaus j-tosios koordinatės reikšmė.

2.3 Programinės priemonės ir jos taikymo aprašymas

Sukurtos programinės įrangos pagrindinis programos langas yra pavaizduotas 2.6. pav., o vartotojo veiksmai yra atliekami pasinaudojant programos meniu, kurio struktūra yra atvaizduota 2.7. pav.



2.6. pav. Pagrindinis programos langas



2.7. pav. Komandų „File“ ir „Commands“ skleistinės

Pradiniai programos duomenys į programinę įrangą gali būti suvesti dvejopai. Vienas iš būdų yra nuskaityti duomenis iš failo, kuris parenkamas meniu punkto „File->Open“ pagalba. Šis failas yra specialios XML struktūros ir gali būti sukurtas tik išsaugojant jau esamą programinėje įrangoje aptarnavimo sistemos schematinę struktūrą meniu punkto „File->Save“ pagalba. Kitas būdas yra sudaryti aptarnavimo sistemos schematinę struktūrą panaudojant mygtukus „New“ ir „Delete“. Pirmuoju mygtuku yra sukuriami programinės įrangos komponentai, kurių parametrus vartotojas įveda pasinaudodamas po mygtukais esančiais įvedimo laukais. Parametrų galimos reikšmės yra pateiktos 2.1. lentelėje.

Kuomet vartotojas nuskaito duomenis iš failo arba pats sudaro aptarnavimo sistemos schematinę struktūrą, norint apskaičiuoti sistemos tikimybines charakteriskas ir stacionariąsias tikimybes reikia pasirinkti meniu punktą „Commands->Calculate“. Po šio veiksmo programa suranda sistemos būsenų stacionariąsias tikimybes ir jas atvaizduoja programos polangyje „Calculations“, apskaičiuoja tikimybines charakteristikas ir jas pateikia „Statistics“ polangyje bei suformuoja informacinius pranešimus „Log“ polangyje. Vartotojas taip pat turi galimybę pasirinkti skaičiavimų nustatymus, t.y. ar naudoti matricos kompaktišką formą ir fizinius failus tarpiniams duomenimis saugoti, polangyje „Parameters“.

Programos polangiai „Calculations“, Statistics“, „Log“ ir „Parameters“ yra pateikti 6.3. priede.

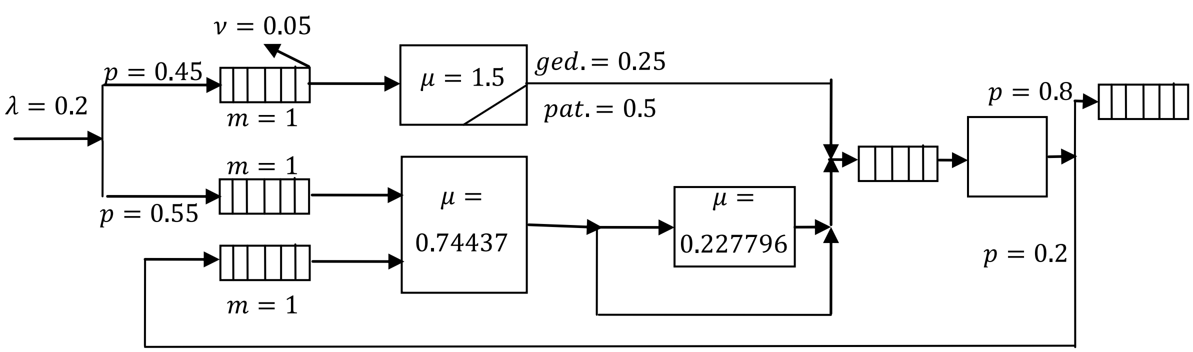
Taip pat “File->Close” meniu pagalba yra baigiamas programos darbas, „File->New“ išvalo visus programos langus ir paruošia programą naujos sistemos modeliavimui, o „Commands->Export Calculations“ išeksportuoja apskaičiuotas stacionariąsias būsenų tikimybes ir būsenų reikšmes į failą.

Optimaliam programinės įrangos veikimui keliami tokie kompiuterio reikalavimai: nežemesnė nei *Windows XP* operacinė sistema su įdiegtu *Service Pack 2* ir *.Net Framework 3.5* komponentu, 1 GB operatyviosios atminties.

3 TIRIAMOJI DALIS IR REZULTATAI

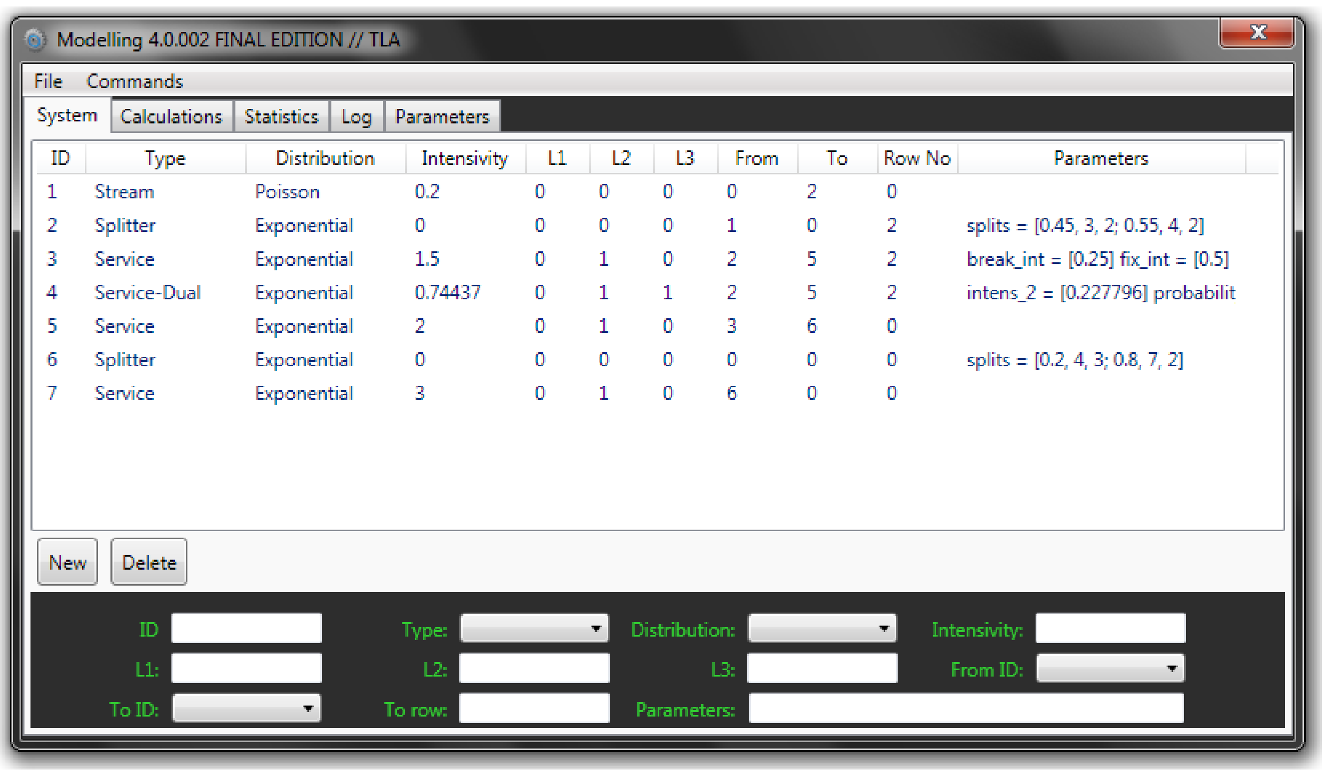
3.1 Sprendžiami uždaviniai

Norėdami pademonstruoti programinės priemonės galimybes aprašyti ir modeliuoti aptarnavimo sistemas, pateiksime pavyzdį panaudodami visus aptarnavimo sistemos komponentus ir jų galimybes. Tarkime, kad norime aprašyti gamybos procesą, kuris yra pateiktas 3.1. pav.



3.1. pav. Pavyzdinė aptarnavimo sistemos schema

Šio proceso sudarymas schematine struktūra programinėje įrangoje yra pateikta 3.2. pav.



ID	Type	Distribution	Intensivity	L1	L2	L3	From	To	Row No	Parameters
1	Stream	Poisson	0.2	0	0	0	0	2	0	
2	Splitter	Exponential	0	0	0	0	1	0	2	splits = [0.45, 3, 2; 0.55, 4, 2]
3	Service	Exponential	1.5	0	1	0	2	5	2	break_int = [0.25] fix_int = [0.5]
4	Service-Dual	Exponential	0.74437	0	1	1	2	5	2	intens_2 = [0.227796] probabilit
5	Service	Exponential	2	0	1	0	3	6	0	
6	Splitter	Exponential	0	0	0	0	0	0	0	splits = [0.2, 4, 3; 0.8, 7, 2]
7	Service	Exponential	3	0	1	0	6	0	0	

3.2. pav. Aptarnavimo proceso schematinė struktūra

Šios sistemos apskaičiuotos sistemos stacionariųjų tikimybių reikšmės yra pateiktos 6.1 priede, o apskaičiuotos aptarnavimo sistemos charakteristikos 3.1. lentelėje, kuri buvo sudaryta iš programinės įrangos rezultatų, kurie pateikti 6.2. priede

Lentelė 3.1. Aptarnavimo sistemos charakteristikos

Aptarnavimo įrenginys	Stochastinės charakteristikos
3	Tikimybė, kad atėjusi bet kurio prioriteto paraiška bus iš karto aptarnauta: 0.960438967849912; Tikimybė, kad atėjusi vidutinio prioriteto paraiška stos į eilę: 0.978284413632912 Tikimybė, kad atėjusi vidutinio prioriteto paraiška bus neaptarnauta: 0.0217155863670856 Vidutinio prioriteto eilės vidutinės eilės ilgis: 0.0217155863670856 Tikimybė, kad aptarnavimo įrenginys suges: 0.333335402715041
4	Tikimybė, kad atėjusi bet kurio prioriteto paraiška bus iš karto aptarnauta: 0.737292512249025; Tikimybė, kad atėjusi vidutinio prioriteto paraiška stos į eilę: 0.961708644095299 Tikimybė, kad atėjusi vidutinio žemo paraiška stos į eilę: 0.967684163689258 Tikimybė, kad atėjusi vidutinio prioriteto paraiška bus neaptarnauta: 0.0382913559046989 Tikimybė, kad atėjusi vidutinio žemo paraiška bus neaptarnauta: 0.0323158363107401 Vidutinio prioriteto eilės vidutinės eilės ilgis: 0.0382913559046989 Žemo prioriteto eilės vidutinės eilės ilgis: 0.0323158363107401
5	Tikimybė, kad atėjusi bet kurio prioriteto paraiška bus iš karto aptarnauta: 0.799229542443682; Tikimybė, kad atėjusi vidutinio prioriteto paraiška stos į eilę: 0.970357852200268 Tikimybė, kad atėjusi vidutinio prioriteto paraiška bus neaptarnauta: 0.0296421477997315 Vidutinio prioriteto eilės vidutinės eilės ilgis: 0.0296421477997315
7	Tikimybė, kad atėjusi bet kurio prioriteto paraiška bus iš karto aptarnauta: 0.893344280187885; Tikimybė, kad atėjusi vidutinio prioriteto paraiška stos į eilę: 0.991608153792756 Tikimybė, kad atėjusi vidutinio prioriteto paraiška bus neaptarnauta: 0.00839184620724279 Vidutinio prioriteto eilės vidutinės eilės ilgis: 0.00839184620724279

3.2 Modelio adekvatumo tikrinimas

3.2.1 Skaičiavimų rezultatų palyginimas su analizinėmis išraiškomis

Tarkime, kad į n-kanalės aptarnavimo sistemos įėjimą ateina paprasčiausias paraiškų srautas su intensyvumu λ. Vienos paraiškos aptarnavimo laikas – eksponentinis su parametru μ. Stacionarios tikimybės egzistuoja tuomet, kai $\rho/n = \chi < 1$, čia $\rho = \lambda/\mu$. Tokios aptarnavimo sistemos būsenos:

S_0 − aptarnavimo sistema laisva;

S_1 − užimtas vienas kanalas; ... ;

S_k − užimta k kanalų $(1 \le k \le n)$; ... ;

S_{n+1} − užimti visi n kanalai, viena paraiška stovi eilėje; ... ;

S_{n+r} − užimti visi n kanalai, r paraiškų stovi eilėje $(1 \leq \mathrm{r} \leq \mathrm{m})$; ... ;

Tokios aptarnavimo sistemos stacionarios tikimybės yra išreiškiamos formulėmis:

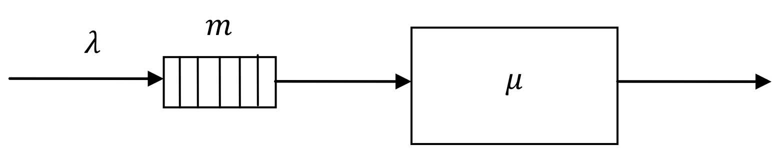
$$p_0 = \left\{1 + \frac{\rho}{1!} + \cdots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n \cdot n!} \cdot \frac{1 - \chi^m}{1 - \chi}\right\}^{-1} \qquad (\mathbf{3.1.})$$

$$p_k = \frac{\rho^k}{k!} p_0 \; (1 \leq k \leq n); \;\; p_{n+r} = \frac{\rho^{n+r}}{n^r \cdot n!} p_0 \; (1 \leq r \leq m) \qquad (\mathbf{3.2.})$$

čia $\chi = \rho/n = 1/(n\mu)$.

Kadangi šios formulės tinka tik n-kanalių aptarnavimo sistemų su viena bendra paraiškų eile visiems kanalams modeliuoti, o programinėje įrangoje keli aptarnavimo įrenginiai negali turėti bendros paraiškų eilės, skaičiavimo ir modeliavimo rezultatus galime palyginti tik vienkanalei aptarnavimo sistemai su fiksuotu eilės ilgiu.

Tokios aptarnavimo sistemos schema yra pavaizduota 3.2. pav.



3.3. pav. Vienkanalė aptarnavimo sistema

Atliksime keletą aptarnavimo sistemos modeliavimo bandymų su skirtingomis parametrų reikšmėmis ir palyginsime su rezultatais gautais panaudojus analitines išraiškas.

Tarkime, kad paraiškos į aptarnavimo sistema ateina su intensyvumu $\lambda = 2$, o jos yra aptarnaujamos intensyvumu $\mu = 3$, o galima paraiškų eilė $m = 50$.

Pateiksime keleto būsenų stacionarių tikimybių skaičiavimo analizinėmis formulėmis pavyzdžių:

Būsena (0;0) atitinka sistemos būseną S_0, kurios stacionarioji tikimybė yra apskaičiuojama tokiu būdu:

$$p_0 = \left\{1 + \frac{\rho}{1!} + \cdots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n \cdot n!} \cdot \frac{1-\chi^m}{1-\chi}\right\}^{-1} = \left\{1 + \rho + \rho^{1+1} \cdot \frac{1-\chi^{50}}{1-\chi}\right\}^{-1}$$

$$= \left\{1 + \frac{2}{3} + \left(\frac{2}{3}\right)^{1+1} \cdot \frac{1-\left(\frac{2}{3}\right)^{50}}{1-\frac{2}{3}}\right\}^{-1} = \left\{1 + \frac{2}{3} + \frac{4}{9} \cdot \frac{1-\left(\frac{2}{3}\right)^{50}}{\frac{1}{3}}\right\}^{-1}$$

$$= 3.33333333565678\mathrm{E}-01$$

$$p_1 = \frac{\rho^k}{k!} p_0 = \frac{\left(\frac{2}{3}\right)^1}{1} p_0 = 2.22222222377119\mathrm{E}-01$$

$$p_8 = \frac{\left(\frac{2}{3}\right)^{1+7}}{1^7 \cdot 1!} p_0 = 1.30061474459402\mathrm{E}-02$$

Lentelė 3.2. Vienkanalės aptarnavimo sistemos modeliavimo rezultatai 1

$\boldsymbol{\lambda = 2;\ \mu = 3; m = 50}$.					
Būsena	**Nr.**	**Apskaičiuota reikšmė**	**Sumodeliuota reikšmė**	**Paklaida**	**Santykinė paklaida**
(0;0)	1	3.33333333565678E-01	3.33333333565678E-01	0.00E+00	0.00E+00
(0;2)	2	2.22222222377119E-01	2.22222222377119E-01	0.00E+00	0.00E+00
(7; 2)	9	1.30061474459402E-02	1.30061474459402E-02	0.00E+00	0.00E+00
(12;2)	14	1.71274369658472E-03	1.71274369658472E-03	0.00E+00	0.00E+00
(13;2)	15	1.14182913105648E-03	.14182913105648E-03	0.00E+00	0.00E+00
(25;2)	27	8.80047290443781E-06	8.80047290443782E-06	0.00E+00	0.00E+00
(26;2)	28	5.86698193629187E-06	5.86698193629188E-06	-1.02E-20	1.73E-15
(38;2)	40	4.52188634506022E-08	4.52188634506023E-08	-9.93E-23	2.20E-15
(39;2)	41	3.01459089670681E-08	3.01459089670682E-08	-9.93E-23	3.29E-15
(48;2)	50	7.84164273288567E-10	7.84164273288569E-10	-2.07E-24	2.64E-15
(49;2)	51	5.22776182192378E-10	5.2277618219238E-10	-2.07E-24	3.96E-15
(50;2)	52	3.48517454794919E-10	3.4851745479492E-10	-9.82E-25	2.82E-15

Kaip galime pastebėti iš lentelėje 3.1. pateikto skaičiavimo rezultatų palyginimo fragmento, skaičiavimo paklaidos atsiranda prie tų būsenų, kurių eilės yra didesnės, o šių aptarnavimo sistemų atveju, tai sąlygoja ir mažesnes būsenų stacionariąsias tikimybes ($\rho/n = \chi < 1$ - ši sąlyga reiškia, kad paraiškos turi būti aptarnaujamos su didesniu intensyvumu, nei jos patenka į sistemą). Atlikime bandymą su vienkanalei aptarnavimo sistemos, kurios paraiškų eilės ilgis būtų pakankamai didelis, t.y. 1500.

Lentelė 3.3. Vienkanalės aptarnavimo sistemos modeliavimo rezultatai 2

$\boldsymbol{\lambda = 2;\ \mu = 3; m = 1500}$.					
Būsena	**Nr.**	**Apskaičiuota reikšmė**	**Sumodeliuota reikšmė**	**Paklaida**	**Santykinė paklaida**
(0;0)	1	3.33333333333333E-01	3.33333333333333E-01	0.00E+00	0.00E+00
(100;0)	102	5.46589872573292E-19	5.46589872573296E-19	-3.95E-33	7.22E-15
(200;0)	202	1.34442219959861E-36	1.34442219959860E-36	1.00E-50	7.46E-15
(400;0)	402	8.13361972848106E-72	8.13361972848087E-72	1.90E-85	2.34E-14
(600;0)	602	4.92075851672842E-107	4.92075851672825E-107	1.70E-120	3.46E-14
(700;0)	702	1.21033654678015E-124	1.21033654678010E-124	5.00E-138	4.13E-14
(900;0)	902	7.32241495114538E-160	7.32241495114501E-160	3.70E-173	5.06E-14
(1000;0)	1002	1.80106103478391E-177	1.80106103478380E-177	1.10E-190	6.11E-14
(1200;0)	1202	1.08962389709799E-212	1.08962389709791E-212	7.99E-226	7.34E-14
(1300;0)	1302	2.68009824180423E-230	2.68009824180403E-230	2.00E-243	7.46E-14
(1400;0)	1402	6.59211550412259E-248	6.59211550412207E-248	5.20E-261	7.89E-14
(1500;0)	1502	1.62143260802407E-265	1.62143260802393E-265	1.40E-278	8.63E-14

Lentelėje 3.3., priešingai negu lentelėje 3.2., tikslią reikšmę apskaičiuosime pasinaudodami ne n-kanalei aptarnavimo sistemai skirtomis analizinėmis formulėmis, o formulėmis, skirtomis paprasčiausių vienkanalių aptarnavimo sistemų su apribojimu eilės dydžiu stacionarioms tikimybėms rasti (**3.3.**) ir (**3.4.**). Šias formules galima nesunkiai išsivesti iš (**3.1.**) ir (**3.2.**), laikant, kad $n = 1$.

$$p_0 = \frac{1-\rho}{1-\rho^{m+2}} \qquad (\mathbf{3.3.})$$

$$p_k = \rho^k p_0 \ (1 \le k \le m+1) \qquad (\mathbf{3.4.})$$

Remiantis 3.2. ir 3.2. lentelių rezultatais, bei tyrimais (kurių rezultatai susiję su santykinės paklaidos didėjimu esant dideliam būsenų kiekiui ir yra panašūs lentelėse pateiktiems rezultatams), atliktais keičiant vienkanalės aptarnavimo sistemos parametrus, galime daryti prielaidą, kad esant pakankamai dideliam būsenų skaičiui, skaičiuojant stacionariąsias tikimybes kaupiasi paklaidos. Viena iš priežasčių yra ta, kad skaičiavimo tarpiniams ir galutiniams rezultatams saugoti yra naudojami *double* tipo kintamieji, kurie saugoja skaičių reikšmes 15 skaičių tikslumu.

3.2.2 Skaičiavimų rezultatų palyginimas su žinomais sprendiniais

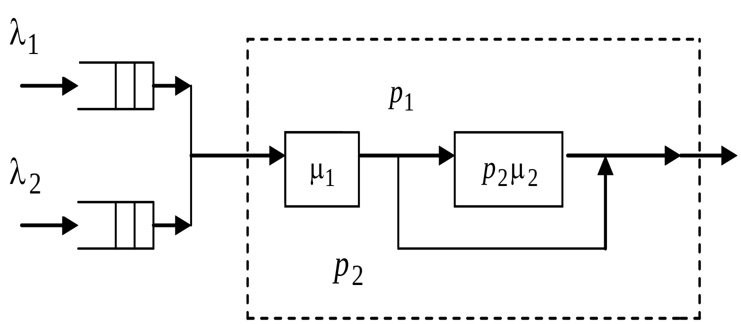
Norint įsitikinti, kad sukurtoji programinė įranga atlieka skaičiavimus gerai ir galutinės suskaičiuotos stacionariosios tikimybės yra pakankamai tikslios nepakanka skaičiavimo rezultatus palyginti tik su žinomomis analizinėmis formulėmis. Kitas būdas yra palyginti gautus rezultatus modeliuojant programine įranga su literatūroje paskelbtais aptarnavimo sistemų modeliavimo rezultatais.

Pasirenkame [3] straipsnyje publikuotą antrą pavyzdį. Vienas iš pagrindinių šio pasirinkimo kriterijų yra, kad jame nagrinėjama ne Markovo aptarnavimo sistema (paraiškos ateina pagal Puasono dėsnį, tačiau aptarnaujamos ne pagal Eksponentinį dėsnį). Šiuo pavyzdžiu iliustruosime, kaip programinėje įrangoje yra panaudojamos aproksimuojantis aptarnavimo įrenginio komponentas bei kaip yra taikoma ne Markovo grandinių modeliavimo teorija.

Tarkime, kad paraiškų aptarnavimo laikai dvikanalėje aptarnavimo sistemoje yra pasiskirstę pagal lognormalųjį skirstinį (**3.5.**), su parametrais $\alpha = 0.9, \lambda = -0.05$.

$$g(x) = \frac{1}{\alpha x\sqrt{2\pi}} e^{\frac{-(ln x - \lambda)^2}{2\alpha^2}}, x > 0 \qquad (\mathbf{3.5.})$$

Tokios aptarnavimo sistemos aproksimuojanti sistema pavaizduota 3.2. pav.



3.4. pav. Aproksimuojanti aptarnavimo sistema

Remdamiesi 1.3. skyriuje išdėstyta teorija, apskaičiuosime aproksimacijos parametrus μ_1, μ_2, p_1 ir p_2.

Yra žinoma, kad lognormaliojo skirstinio k-tąjį ne centrinį momentą, galima rasti pagal (**3.6.**) formulę.

$$E[X^k] = e^{k\lambda + \frac{1}{2}k^2\alpha^2} \qquad (\mathbf{3.6.})$$

Taigi, pirmieji trys ne centriniai momentai: $m_1 = 1.42618, m_2 = 4.57225, m_3 = 28.3606$. Įstatydami šias reikšmes į (**1.39.**), randame aproksimacijos parametrus, t.y.

$$\lambda_1 = 0.2; \lambda_2 = 0.9;\ \mu_1 = 0.74437;\ \mu_2 = 0.227796;\ p_1 = 0.018503.$$

Parinkę prioritetinių eilių ilgius $l_1 = 25;\ l_2 = 25$ ir sudarius atitinkamą sistemos modelį programinėje įrangoje (žr. priedą 9.4.), apskaičiuojamos šios aptarnavimo sistemos stacionariosios būsenų tikimybės, bei remiantis jomis, vidutinės aptarnavimo sistemos paraiškų eilės. Atlikus šiuos skaičiavimus (žr. priedą 9.4.) randamos vidutinės paraiškų eilės:

$$L_1 = 1.85506424629069\text{E-}01$$

$$L_2 = 1.42354348835089\text{E-}01$$

Kurios sutampa su straipsnyje pateiktais skaičiavimo rezultatais: $L_q^{(1)} = 0.1855, L_q^{(2)} = 0.1424$.

3.3 Programinės įrangos tyrimai

Norėdami ištirti sukurtą programinę įrangą buvo sudaryti testai, kurių pagalba nustatysime programinės įrangos skaičiavimo trukmės ir naudojamos operatyviosios atminties priklausomybes nuo naudojamo metodo, būsenų skaičiaus ir perėjimų tarp būsenų skaičiaus. Atliktų testų rezultatai yra pateikti toliau esančioje 3.4. lentelėje. Testai buvo atliekami su kompiuteriu, kurio parametrai: Pentium Dual-Core E5200@2,5 GHz, 4GB operatyviosios atminties ir Windows 7 operacine sistema su 90 GB laisvos vietos kietajame diske.

Detaliau paaiškinsime kiekvieno 3.4. lentelės stulpelio reikšmę:

0 – Testo numeris;

1 – Nurodo kokia matricos forma buvo naudojama skaičiavimų metu (norm. – paprasta, komp . - kompaktiška);

2 – Nurodo ar tarpiniams rezultatams saugoti buvo naudojami failai („-“ – ne, „+“ - taip);

3 – Nurodo programos sugaištą trukmę sekundėmis (milisekundžių tikslumu), aptaranavimo sistemos aprašymo nuskaitymui;

4 – Nurodo programos sugaištą trukmę sekundėmis (milisekundžių tikslumu), aptaranavimo sistemos pradinės būsenos sudarymui;

5 – Nurodo programos sugaištą trukmę sekundėmis (milisekundžių tikslumu), aptaranavimo sistemos visų galimų būsenų aibės sudarymui ir perėjimų intensyvumui tarp būsenų nustatymui;

6 – Nurodo programos sugaištą trukmę sekundėmis (milisekundžių tikslumu), aptaranavimo sistemos lygties sprendimui;

7 – Nurodo bendrą programos sugaištą trukmę (milisekundžių tikslumu);

8 – Nurodo programinės įrangos naudojamą operatyviosios atminties kiekį megabaitais;

9 – Nurodo aptarnavimo sistemos būsenų skaičių;

10 – Nurodo perėjimų tarp būsenų bendrą skaičių.

Lentelė 3.4. Programinės įrangos testai

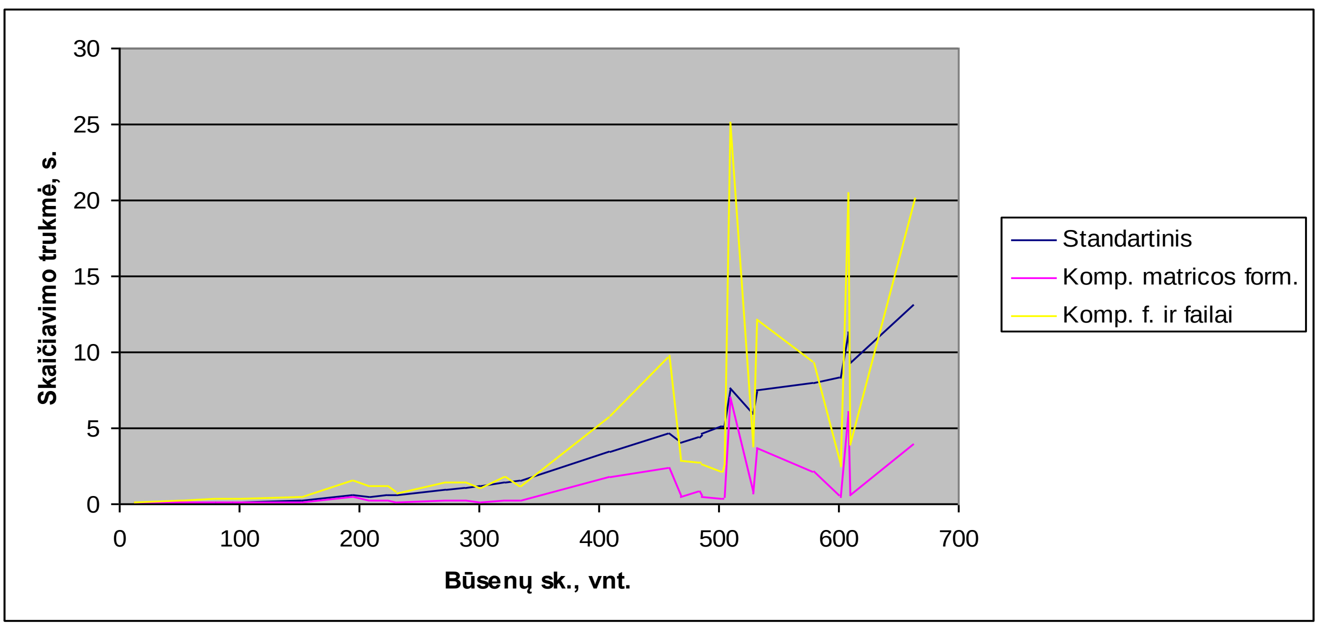
0	1	2	3	4	5	6	7	8	9	10
Nr.	Mat. f.	Failai	Schem. nusk., s.	Pr. būs., s.	Visos būs., s.	Lygt. spr., s.	Bendra trukmė, s.	RAM, MB	Būs. sk., vnt.	Per. Sk., vnt.
1	norm.	-	0.02	0.01	0.04	5.180008	5.260008	733.4141	505	1448
2	norm.	-	0.02	0.02	0.05	9.226032	9.326032	1245.625	609	1756
3	norm.	-	0.03	0	0.03	0	0.07	56.9375	13	28
4	norm.	-	0.0156	0	0.0468	4.992009	5.070009	734.418	503	1005
5	norm.	-	0.0156	0.0156	0.0156	1.092002	1.154402	202.8438	302	602
6	norm.	-	0.0156	0.0312	0.0312	8.299215	8.377215	1206.414	602	1202
7	norm.	-	0.0312	0	0.0312	0.0156	0.078	57.125	52	102
8	norm.	-	0.0156	0.0156	0.0312	0.982802	1.045202	191.7539	289	1088
9	norm.	-	0.0312	0	0.0468	4.258808	4.352408	654.6602	484	1848
10	norm.	-	0.0312	0	0.0468	0.452401	0.546001	106.9805	196	903
11	norm.	-	0.0312	0	0.0936	7.363213	7.503613	854.3359	532	2619
12	norm.	-	0.0156	0	0.0624	5.89681	5.990411	841.2188	529	2001
13	norm.	-	0.0156	0.0156	0.0468	4.508408	4.602008	667.293	485	1608
14	norm.	-	0.0156	0	0.0312	0.0468	0.1092	61.01172	102	202
15	norm.	-	0.0312	0	0.0312	0.514801	0.577201	125.6758	232	462
16	norm.	-	0.0312	0	0.0312	0.436801	0.514801	118.2813	210	826
17	norm.	-	0.0156	0.0156	0.0468	4.009207	4.102807	611.6992	469	1683
18	norm.	-	0.0156	0	0.078	7.909214	8.018414	1088.734	580	2850
19	norm.	-	0.0312	0	0.1404	12.99482	13.18202	1579.656	663	3329
20	norm.	-	0.0156	0.0156	0.078	4.508408	4.633208	569.5898	459	2289
21	norm.	-	0.0312	0.0156	0.0936	7.456813	7.597213	772.5703	510	2753
22	norm.	-	0.0312	0	0.0312	0.530401	0.608401	126.1914	224	960
23	norm.	-	0.0312	0.0156	0.0156	1.497603	1.575603	257.75	335	668
24	norm.	-	0.0312	0.0156	0.0312	1.326002	1.419603	244.1445	322	1202
25	norm.	-	0.0312	0.0156	0.0468	3.322806	3.432006	425.7734	410	1778
26	norm.	-	0.0312	0	0.0936	11.20082	11.34122	1247.922	608	2928
27	norm.	-	0.0312	0	0.0312	0.842402	0.920402	164.0781	273	999
28	norm.	-	0.0156	0	0.0468	0.0312	0.0936	57.92578	81	279
29	norm.	-	0.0156	0.0156	0.0312	0.156	0.2184	75.35547	152	302
30	norm.	-	0.0156	0.0312	0.0156	0.0468	0.1092	67.89844	96	314
1	komp.	-	0.0312	0	0.0468	0.358801	0.436801	76.30469	505	1448
2	komp.	-	0.0312	0	0.0468	0.514801	0.608401	81.66797	609	1756
3	komp.	-	0.0312	0.0156	0.0156	0.0156	0.0936	56.74609	13	28
4	komp.	-	0.0156	0	0.0468	0.2496	0.327601	67.29297	503	1005
5	komp.	-	0.0156	0	0.0468	0.0936	0.156	62.43359	302	602

0	1	2	3	4	5	6	7	8	9	10
6	**komp.**	-	0.0312	0	0.0312	0.358801	0.436801	75.04688	602	1202
7	**komp.**	-	0.0312	0	0.0312	0.0156	0.078	56.41797	52	102
8	**komp.**	-	0.0312	0	0.0312	0.2028	0.280801	73.96094	289	1088
9	**komp.**	-	0.0156	0.0156	0.0468	0.717601	0.811201	86.47266	484	1848
10	**komp.**	-	0.0156	0.0156	0.0312	0.374401	0.452401	74.81641	196	903
11	**komp.**	-	0.0312	0	0.0936	3.603606	3.744007	137.7422	532	2619
12	**komp.**	-	0	0.0156	0.0468	0.686401	0.764401	158.4492	529	2001
13	**komp.**	-	0.0156	0	0.0624	0.374401	0.468001	85.38281	485	1608
14	**komp.**	-	0.0156	0.0156	0.0312	0.0312	0.0936	64.60547	102	202
15	**komp.**	-	0.0156	0	0.0468	0.0624	0.1248	59.95313	232	462
16	**komp.**	-	0.0312	0	0.0312	0.2028	0.2652	64.5	210	826
17	**komp.**	-	0.0156	0.0156	0.0468	0.405601	0.499201	78.19141	469	1683
18	**komp.**	-	0.0312	0	0.078	2.043604	2.168404	115.543	580	2850
19	**komp.**	-	0.0156	0.0156	0.1248	3.931207	4.102807	156.5078	663	3329
20	**komp.**	-	0.0156	0	0.0936	2.308804	2.418004	121.918	459	2289
21	**komp.**	-	0.0312	0	0.1092	6.895212	7.051212	213.1758	510	2753
22	**komp.**	-	0.0312	0	0.0312	0.2028	0.2652	64.75781	224	960
23	**komp.**	-	0.0156	0.0156	0.0312	0.1248	0.1872	63.37109	335	668
24	**komp.**	-	0.0156	0.0156	0.0312	0.2184	0.296401	67.47266	322	1202
25	**komp.**	-	0.0312	0	0.0468	1.700403	1.794003	95.28516	410	1778
26	**komp.**	-	0.0156	0.0156	0.0936	5.959211	6.084011	201.2852	608	2928
27	**komp.**	-	0.0156	0.0156	0.0312	0.1716	0.249601	74.17969	273	999
28	**komp.**	-	0.0156	0.0156	0.0312	0.0312	0.1092	64.86328	81	279
29	**komp.**	-	0.0312	0	0.0312	0.0468	0.1092	57.21875	152	302
30	**komp.**	-	0.0156	0.0156	0.0156	0.0312	0.0936	56.86328	96	314
31	**komp.**	-	0.0312	0	0.234	33.24366	33.52446	883.8398	918	5037
32	**komp.**	-	0.0156	0	0.2184	4.617608	4.882809	263.4922	1574	5888
1	**komp.**	+	0.0156	0	0.0468	2.620805	2.683205	56.87109	505	1448
2	**komp.**	+	0.0312	0	0.0468	3.790807	3.884407	58.59766	609	1756
3	**komp.**	+	0.0312	0	0.0312	0.0468	0.1092	56.90625	13	28
4	**komp.**	+	0.0312	0	0.0312	2.028004	2.106004	56.8125	503	1005
5	**komp.**	+	0.0156	0	0.0312	0.951602	1.014002	57.0625	302	602
6	**komp.**	+	0.0156	0	0.0468	2.480404	2.558405	61.06641	602	1202
7	**komp.**	+	0.0156	0	0.0312	0.1404	0.2028	57.16406	52	102
8	**komp.**	+	0.0156	0.0156	0.0312	1.326002	1.404003	56.95313	289	1088
9	**komp.**	+	0.0312	0	0.0468	2.683205	2.776805	64.83594	484	1848
10	**komp.**	+	0.0312	0	0.0468	1.528803	1.606803	56.9375	196	903
11	**komp.**	+	0.0156	0.0156	0.0936	12.01202	12.15242	70.05078	532	2619
12	**komp.**	+	0.0156	0.0156	0.0624	3.681606	3.790807	65.16797	529	2001
13	**komp.**	+	0.0156	0	0.0468	2.605205	2.667605	57.08984	485	1608
14	**komp.**	+	0.0156	0	0.0312	0.265201	0.327601	57.11719	102	202
15	**komp.**	+	0.0156	0	0.0312	0.624001	0.686401	61.20313	232	462
16	**komp.**	+	0.0156	0	0.0468	1.060802	1.170002	56.875	210	826
17	**komp.**	+	0.0312	0	0.0468	2.745605	2.886005	64.79688	469	1683
18	**komp.**	+	0.0312	0	0.078	9.219616	9.344416	59.69141	580	2850

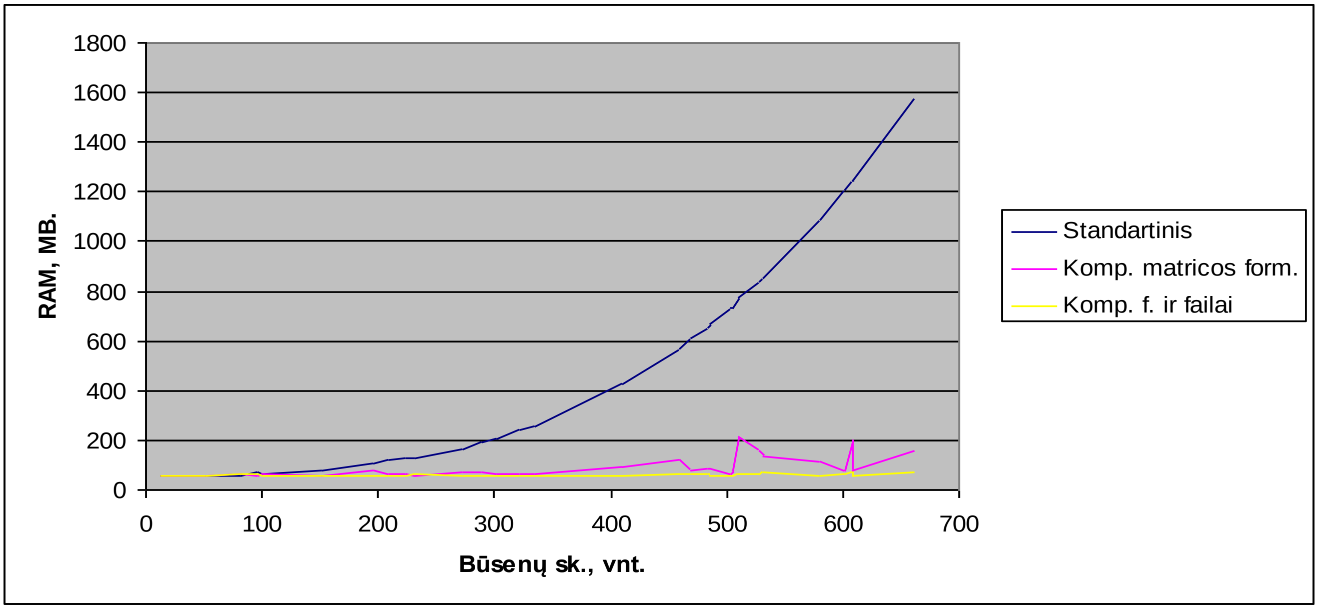
0	1	2	3	4	5	6	7	8	9	10
19	**komp.**	+	0.0312	0.0156	0.1248	20.06603	20.25324	69.30469	663	3329
20	**komp.**	+	0.0312	0	0.078	9.640817	9.765617	62.3125	459	2289
21	**komp.**	+	0.0312	0	0.1092	25.00684	25.16284	61.41406	510	2753
22	**komp.**	+	0.0156	0.0156	0.0312	1.123202	1.201202	57.01172	224	960
23	**komp.**	+	0.0312	0	0.0312	1.154402	1.232402	57.17969	335	668
24	**komp.**	+	0.0156	0.0156	0.0468	1.684803	1.762803	57.14063	322	1202
25	**komp.**	+	0.0312	0	0.0624	5.70961	5.81881	60.17578	410	1778
26	**komp.**	+	0.0312	0	0.0936	20.29564	20.42044	70.48047	608	2928
27	**komp.**	+	0.0312	0	0.0312	1.294802	1.372802	57.03125	273	999
28	**komp.**	+	0.0156	0.0156	0.0156	0.2652	0.327601	61.40625	81	279
29	**komp.**	+	0.0156	0	0.0312	0.483601	0.530401	56.84375	152	302
30	**komp.**	+	0.0312	0	0.0312	0.280801	0.358801	64.80469	96	314
31	**komp.**	+	0.0312	0	0.234	110.4902	110.771	61.77344	918	5037
32	**komp.**	+	0.0156	0	0.2184	33.25926	33.50886	68.17578	1574	5888

Šių testų rezultatus panaudosime skaičiavimo trukmės priklausomybių ir naudojamos operatyviosios atminties priklausomybių tyrimams.

Visų pirma grafiškai atvaizduosime skaičiavimo trukmės ir naudojamos RAM priklausomybes nuo būsenų skaičiaus 3.5. pav. ir 3.6. pav.

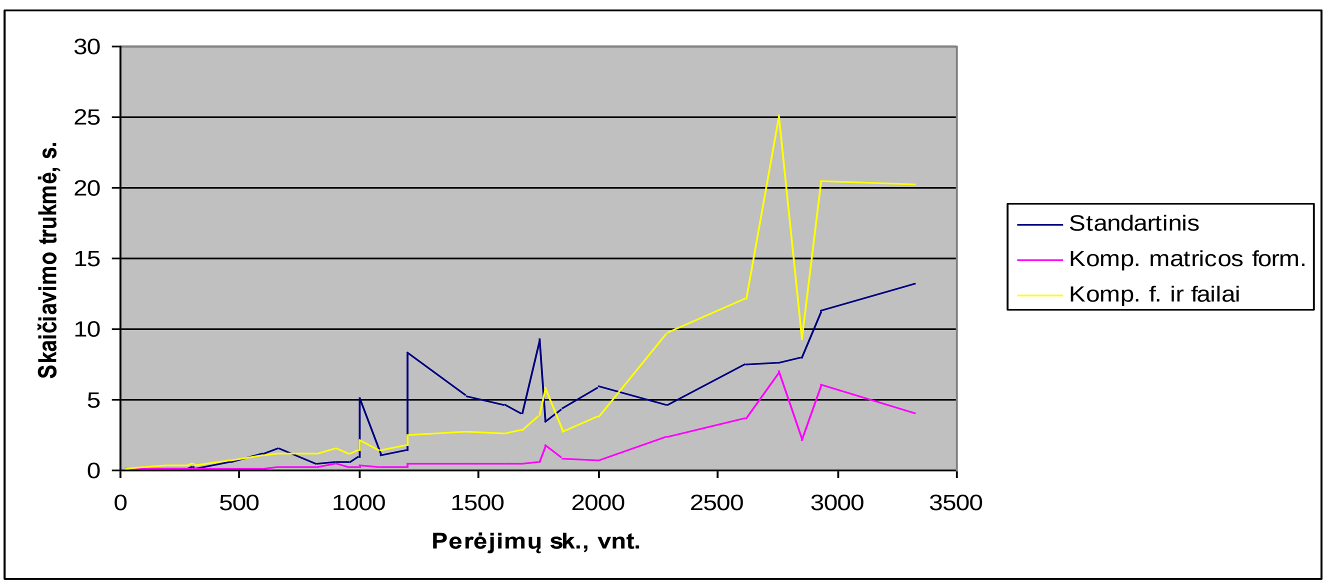


3.5. pav. Skaičiavimo trukmės priklausomybės nuo būsenų skaičiaus tyrimas

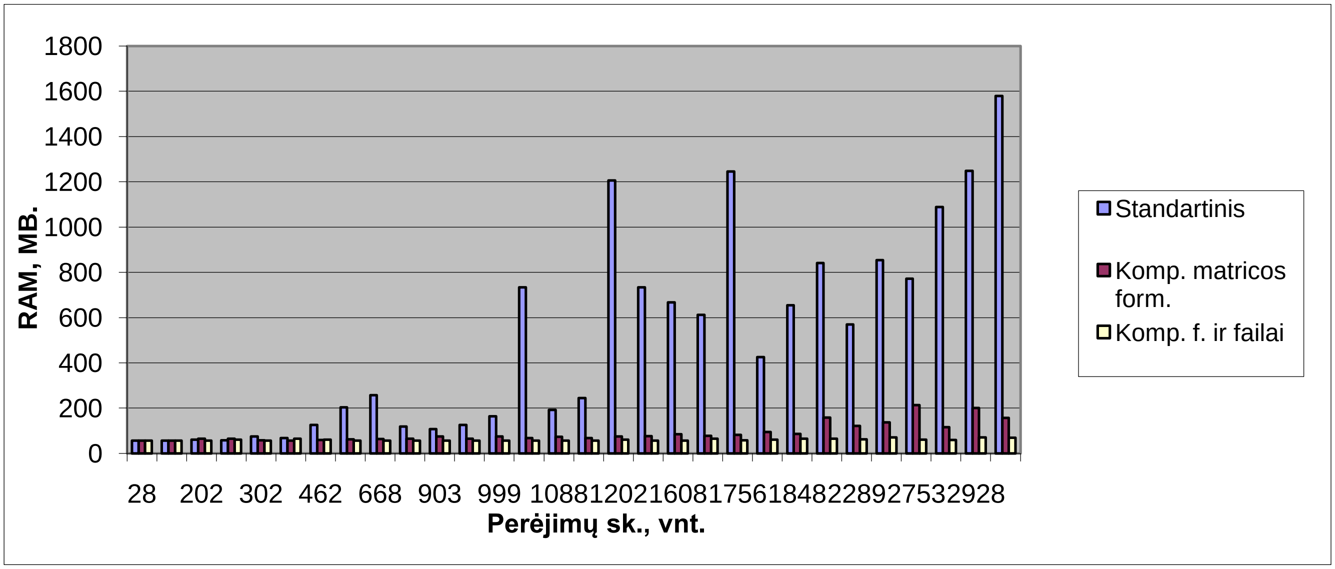


3.6. pav. Naudojamos operatyviosios atminties (RAM) priklausomybės nuo būsenų skaičiaus tyrimas

Kaip galime pastebėti iš 3.5. ir 3.6. grafikų, skaičiavimo laikas ir naudojamas operatyviosios atminties kiekis priklauso ne tik nuo būsenų skaičiaus. Palyginus 21 ir 12 testų rezultatus, galime pastebėti, kad nors sistemų būsenų skaičiai yra beveik vienodi, tačiau skaičiavimo trukmė ir naudojamas RAM kiekis stipriai skiriasi. Vienas iš greičiausiai pastebimų skirtumų tarp šių dviejų skaičiavimų yra tas, kad gan ženkliai skiriasi perėjimų tarp būsenų skaičius. Atlikime tuos pačius tyrimus tik tirkime priklausomybes ne nuo būsenų skaičiaus, o pagal perėjimų tarp būsenų kiekį. Šių tyrimų rezultatai yra pateikti 3.7. ir 3.8. pav.

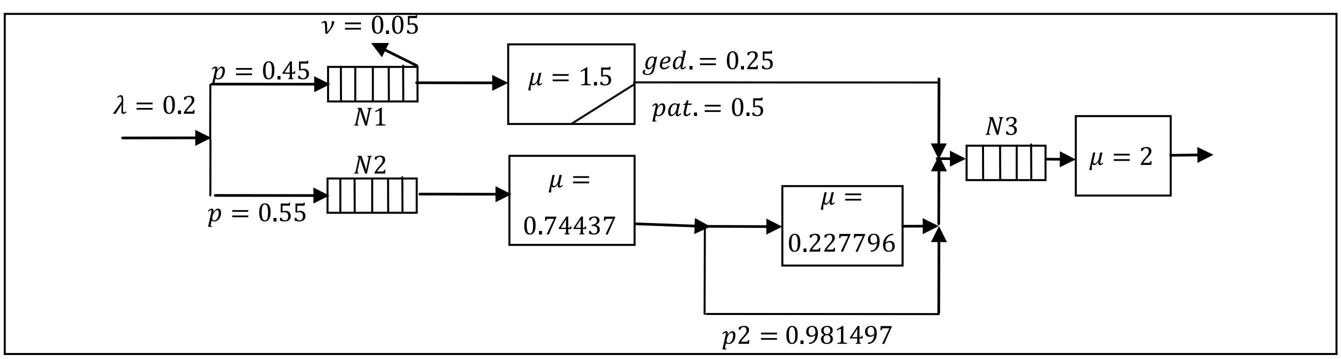


3.7. pav. Skaičiavimo trukmės priklausomybės nuo perėjimų skaičiaus tyrimas

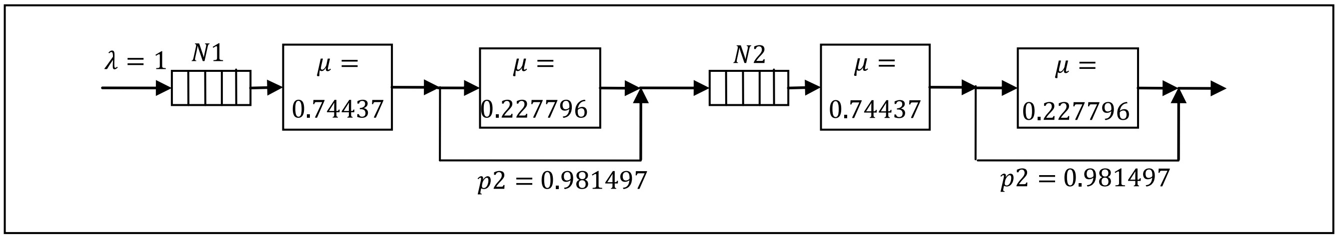


3.8. pav. Naudojamos operatyviosios atminties (RAM) priklausomybės nuo perėjimų skaičiaus tyrimas

Panaši situacija gaunama ir tyrimuose, kuomet buvo tiriamos priklausomybės nuo perėjimų tarp būsenų skaičiaus. Akivaizdu, kad šių tyrimų rezultatai priklauso nuo abiejų parametrų bei pačios aptarnavimo sistemos struktūros. Pasinaudodami 3.4. lentelės duomenimis ir 3.5. ir 3.7. grafikus išskirkime du testus, kuomet būsenų skaičius ir perėjimų skaičius yra panašūs, tačiau skaičiavimo trukmės labai skiriasi, t.y. 3.4. lentelėje pateikti testai, kurių numeriai yra 11 ir 12. Vienas iš faktorių įtakojantis skaičiavimo trukmę – šių aptarnavimo sistemų schematinių struktūrų (3.9. pav. ir 3.10. pav.) skirtumas.

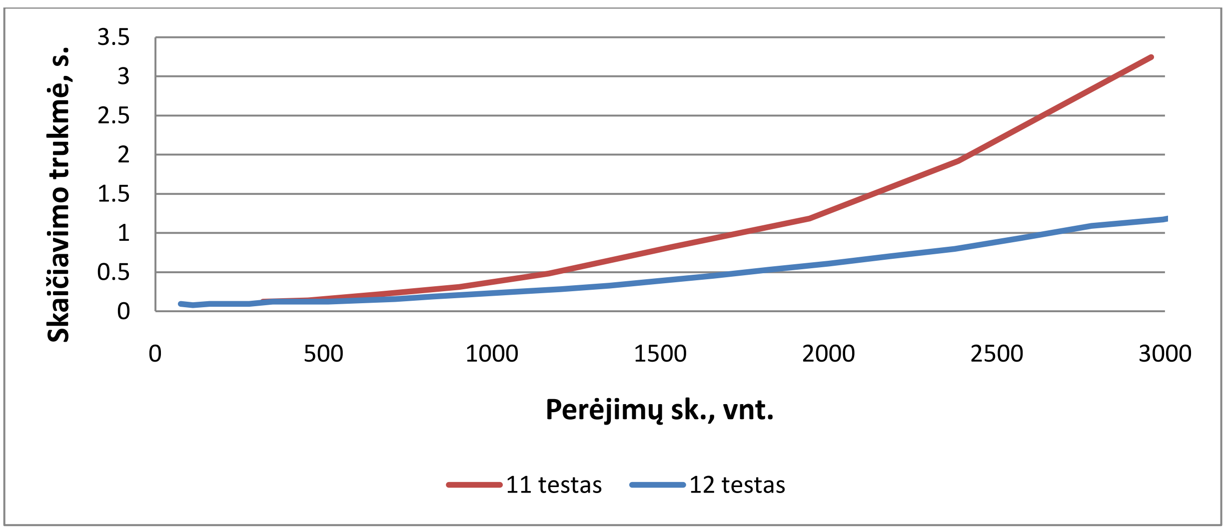


3.9. pav. 11 testo aptarnavimo sistemos struktūra

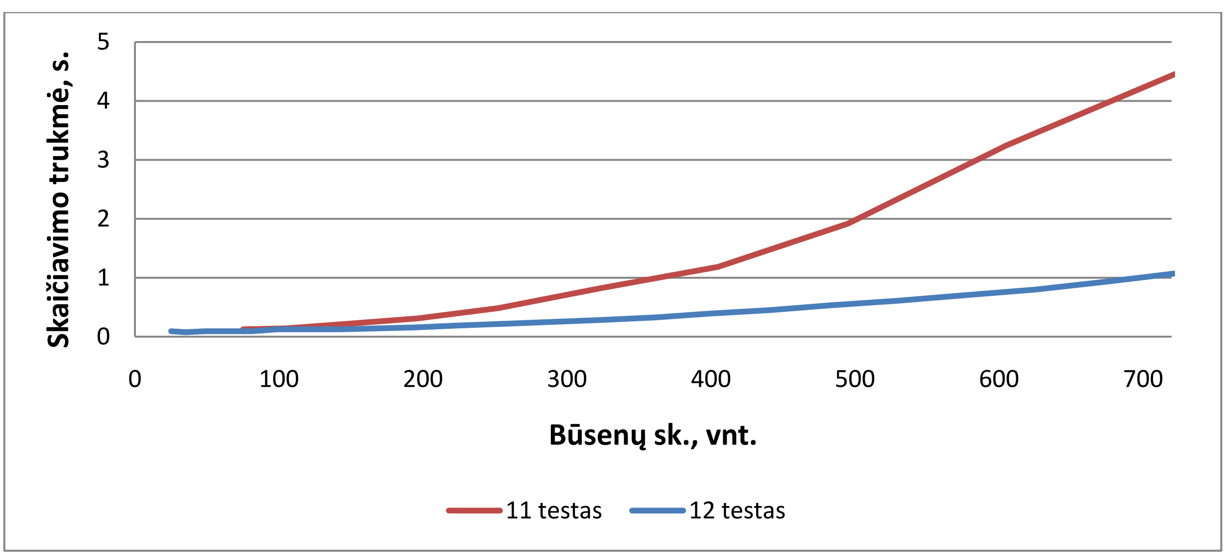


3.10. pav. 12 testo aptarnavimo sistemos struktūra

Šių dviejų aptarnavimo sistemų modeliavimo rezultatai, keičiant aptarnavimo įrenginių ilgius pateikti prieduose 6.5. ir 6.6. Remiantis minėtuose prieduose pateiktais modeliavimo rezultatais, galime palyginti dviejų skirtingų aptarnavimo sistemų modeliavimo laiko priklausomybes nuo būsenų ir perėjimų skaičiaus. Rezultatai yra pateikti 3.11. pav. ir 3.12. pav. esančiuose grafikuose.



3.11. pav. Skaičiavimo trukmės priklausomybė nuo perėjimų skaičiaus



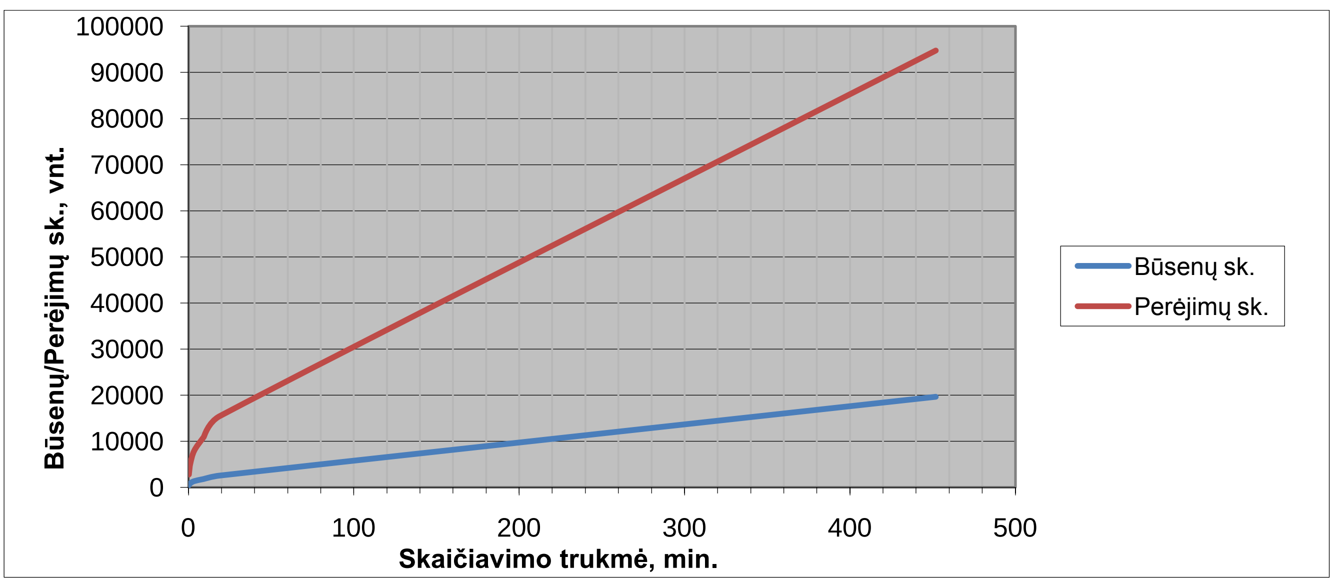
3.12. pav. Skaičiavimo trukmės priklausomybė nuo būsenų skaičiaus

Remiantis 3.11. pav. ir 3.12. pav. galime, teigti, kad aptarnavimo sistemos modeliavimo, sukurta programine įranga, laikas priklauso ne tik nuo būsenų skaičiaus ir perėjimų skaičiaus tarp jų, tačiau ir nuo aptarnavimo sistemą nusakančios schematinės struktūros.

Taip pat, atliksime tyrimus, kokias sistemas programinė įranga gali maksimaliai modeliuoti. Kadangi bus pakankamai didelis sistemos būsenų ir perėjimų tarp jų skaičius, naudosime kompaktišką matricos formą ir tarpinius rezultatus išsaugosime į failus kompiuterio kietajame diske.

Lentelė 3.5. Programinės įrangos testai su dideliu kiekiu būsenų ir perėjimų

0	1	2	3	4	5	6	7	8	9	10
Nr.	**Mat. f.**	**Failai**	**Schem. nusk., s.**	**Pr. būs., s.**	**Visos būs., s.**	**Lygt. spr., s.**	**Bendra trukmė, min.**	**RAM, MB**	**Būs. sk., vnt.**	**Per. Sk., vnt.**
1	komp.	+	0.0156	0	0.0936	12.69842	0.2	69.98828	510	2753
2	komp.	+	0.0312	0.0156	0.1092	22.68244	0.4	69.74219	630	3429
3	komp.	+	0.0312	0	0.2028	66.39372	1.1	70.05078	930	5209
4	komp.	+	0.0312	0.0156	0.390001	206.638	3.45	71.88672	1380	7954
5	komp.	+	0.0312	0	0.702001	548.9514	9.2	73.87109	1840	10912
6	komp.	+	0.0312	0	1.201202	1125.104	18.8	76.96875	2576	15424
7	komp.	+	0.312001	0	28.48565	27086.4	451.9	90.82422	19683	94770



3.13. pav. Skaičiavimo trukmės priklausomybės nuo perėjimų skaičiaus tyrimas

Kaip galime pastebėti iš 3.5. lentelės duomenų, kuomet sistemos būsenų ir perėjimų tarp jų skaičiai yra pakankamai dideli, programa pradeda naudoti didelį kiekį kompiuterio resursų, taip žymiai prailgindama skaičiavimo trukmę. Šį programos trūkumą galima išspręsti pritaikant iteracinius metodus sistemos stacionariosios tikimybėms skaičiuoti, tačiau ne visada rezultatus bus tikslus ir/arba surastas per priimtiną laiką.

Taip pat esant nurodytiems kompiuterio parametrams, maksimalus aptarnavimo sistemos dydis, kurią galima modeliuoti yra 19863 būsenų su 94770 perėjimais tarp jų.

4 IŠVADOS

- Pasiūlyta nauja aptarnavimo sistemų, aprašomų Markovo grandinėmis, modelių kūrimo metodika.
- Iš sukurtų aptarnavimo sistemų komponentų galima sudaryti įvairių aptarnavimo sistemų ir jų tinklų schematinę struktūrą iš kurios automatizuotai gaunamas matematinis modelis.
- Eksperimentiškai buvo nustatyta, kad programinė priemonė leidžia skaičiuoti didelės apimties Markovo grandines, kurių dydis priklauso nuo kompiuterio resursų.
- Modeliavimo eksperimentai parodė, kad didelės apimties Markovo grandinėms skaičiuoti patartina naudoti kompaktišką matricos formą. O labai didelės apimties Markovo grandinėms skaičiuoti reikalingi papildomi išoriniai atminties saugojimo įrenginiai tarpiniams rezultatams saugoti, o tai žymiai prailgina skaičiavimo trukmę.
- Sistemų modeliavimo trukmė priklauso ne tik nuo būsenų ir perėjimų tarp jų skaičiaus, bet ir nuo schematinės aptarnavimo sistemos struktūros.
- Modeliavimo eksperimentai patvirtino modelių adekvatumą analiziniams modeliams.

5 ŠALTINIAI IR LITERATŪRA

1. Dimitar Radev, Vladimir Denchev, Elena Rashkova / *Steady-state solutions of markov chains* / The 7th Balkan Conference on Operational Research „BACOR 05“, Romania, 2005.
2. Fethulah Smailbegovic, Georgi N. Gaydadjiev, Stamatis Vassiliadis / *Sparse matrix storage format*, 2008
3. G. Mickevičius, E. Valakevičius / *Modellig of non-Markovian queuing systems*, Technologiacal and economic development of Economy, 2006, Vol. XII, No. 4, 295-300 p.
4. H. Pranevičius, I. Pranevičienė, E. Valakevičius / *Aptarnavimo sistemų analiziniai ir skaitmeniniai modeliai*, Kaunas, Technologija, 1995, 134 p.
5. Kostas Plukas / *Skaitiniai metodai ir algoritmai*, Kaunas, Naujasis lankas, 2001, 548 p.
6. William J. Srewart / *Introduction to the Numerical Solution of Markov Chains,* Prinston University Press, 1994, 539 p.

6 PRIEDAI

6.1 Aptarnavimo sistemos stacionarios tikimybės

(0;0;0;0;0;0;0;0;0;0;0)	2.93432479002879E-01	(0;1;0;0;0;3;0;0;2;0;0)	2.34879282851302E-03	(0;0;2;0;0;0;0;1;2;0;2)	3.48850886334549E-05
(0;0;2;0;0;0;0;0;0;0;0)	1.76308270915410E-02	(0;1;0;0;0;0;0;0;2;0;2)	1.96503893142929E-03	(0;0;2;0;0;2;0;0;0;1;2)	3.62470578556016E-05
(0;0;0;0;0;2;0;0;0;0;0)	4.32702769823939E-02	(0;0;0;1;0;3;0;0;2;0;0)	1.06733881736612E-03	(1;0;2;1;0;3;0;0;2;0;0)	1.87272502695608E-06
(0;1;0;0;0;0;0;0;0;0;0)	1.91586310064355E-01	(0;0;0;0;0;0;3;0;2;0;0)	2.72250678178147E-04	(1;0;2;0;0;0;3;0;2;0;0)	4.49221887895963E-07
(1;0;2;0;0;0;0;0;0;0;0)	1.04527383013988E-03	(0;0;0;0;1;3;0;0;0;0;0)	4.11398767183588E-03	(1;0;2;0;1;3;0;0;0;0;0)	1.25293698504599E-05
(0;0;2;0;0;2;0;0;0;0;0)	2.59993748550207E-03	(0;0;0;0;0;3;0;0;0;0;2)	4.75012438277440E-03	(1;0;2;0;0;3;0;0;0;0;2)	1.22141624629840E-05
(0;0;0;0;0;0;0;0;2;0;0)	8.64999003386208E-02	(0;0;0;0;0;0;0;0;0;1;2)	4.48382244739890E-03	(1;0;2;0;0;0;0;0;0;1;2)	1.13162943398174E-05
(1;1;0;0;0;0;0;0;0;0;0)	1.21297119095272E-02	(1;1;0;1;0;3;0;0;0;0;0)	1.91385226769650E-04	(0;1;0;0;1;0;2;0;2;0;0)	1.37847230020888E-05
(0;0;0;1;0;2;0;0;0;0;0)	6.55651182769004E-03	(1;1;0;0;0;0;3;0;0;0;0)	5.31679512180917E-05	(0;1;0;0;0;0;2;0;2;0;2)	2.99201319254587E-06
(0;1;0;0;0;2;0;0;0;0;0)	2.83639253676249E-02	(0;0;2;1;0;0;3;0;0;0;0)	5.21201503829134E-05	(0;1;0;0;0;3;0;1;2;0;0)	2.49519056355494E-04
(0;0;0;0;0;0;2;0;0;0;0)	1.74105635934679E-03	(0;1;0;1;0;0;2;0;2;0;0)	2.37912576821399E-05	(0;1;0;0;1;2;0;0;0;0;2)	2.81248031220657E-04
(1;0;2;0;0;2;0;0;0;0;0)	1.54222753407114E-04	(0;1;0;1;1;2;0;0;0;0;0)	8.30406840010607E-04	(0;1;0;0;0;0;0;1;2;0;2)	8.01930580286475E-05
(0;0;2;0;0;0;0;0;2;0;0)	5.25274977614937E-03	(0;1;0;1;0;2;0;0;0;0;2)	2.05746450215943E-04	(0;1;0;0;0;2;0;0;0;1;2)	1.54799952453209E-04
(0;0;2;1;0;2;0;0;0;0;0)	3.93985096914817E-04	(0;0;0;1;1;0;2;0;0;0;0)	6.96646080404756E-04	(0;0;0;0;1;0;2;0;0;0;2)	5.54894447997096E-05
(0;0;0;0;0;2;0;0;2;0;0)	1.27660918177667E-02	(0;0;0;1;0;0;2;0;0;0;2)	1.07294788894617E-04	(0;0;0;0;0;0;2;0;0;1;2)	1.19254472380796E-05
(1;1;0;0;0;2;0;0;0;0;0)	1.79593668955413E-03	(0;1;0;0;1;0;2;0;0;0;0)	3.27034413036973E-04	(0;0;0;1;0;3;0;1;2;0;0)	1.26830636230921E-04
(0;0;2;0;0;0;2;0;0;0;0)	1.04623358010551E-04	(0;1;0;0;0;0;2;0;0;0;2)	4.44541813096060E-05	(0;0;0;0;0;0;3;1;2;0;0)	2.53349678217893E-05
(0;1;0;0;0;0;0;0;2;0;0)	2.35661682458239E-02	(0;1;0;1;0;0;3;0;0;0;0)	5.49333142997080E-04	(0;0;0;0;1;3;0;0;2;0;0)	5.61221787926565E-04
(0;0;0;0;0;3;0;0;0;0;0)	4.09005025944821E-02	(0;0;2;1;0;2;0;1;2;0;0)	9.85615308378728E-06	(0;0;0;0;0;3;0;0;2;0;2)	7.39960156995176E-04
(0;0;0;0;0;0;0;0;0;0;2)	4.40148718504319E-02	(1;0;2;0;0;2;0;1;2;0;0)	4.41825575889680E-06	(0;0;0;0;0;0;0;0;2;1;2)	5.10860598868984E-04
(0;1;0;1;0;2;0;0;0;0;0)	4.32057916697533E-03	(1;0;2;1;0;0;2;0;2;0;0)	3.25510837458634E-07	(1;1;0;1;0;3;0;0;2;0;0)	1.05780431249413E-05
(0;0;0;1;0;0;2;0;0;0;0)	1.07119263900132E-03	(1;0;2;1;1;2;0;0;0;0;0)	5.88508260964123E-06	(1;1;0;0;0;0;3;0;2;0;0)	2.31315464072612E-06
(0;1;0;0;0;0;2;0;0;0;0)	1.15080008911399E-03	(1;0;2;1;0;2;0;0;0;0;2)	1.95301928436960E-06	(1;1;0;0;1;3;0;0;0;0;0)	9.93169329110974E-05
(1;0;2;1;0;2;0;0;0;0;0)	2.33655069016325E-05	(0;1;0;1;0;2;0;1;2;0;0)	2.83472667021501E-05	(1;1;0;0;0;3;0;0;0;0;2)	8.43576118133184E-05
(0;0;2;0;0;2;0;0;2;0;0)	7.75310312112705E-04	(0;0;0;1;0;0;2;1;2;0;0)	1.58458084007333E-05	(1;1;0;0;0;0;0;0;0;1;2)	7.39484443897798E-05
(1;0;2;0;0;0;0;0;2;0;0)	1.91224962789992E-04	(0;0;0;1;1;2;0;0;2;0;0)	2.98854196312457E-04	(0;0;2;1;0;0;3;0;2;0;0)	9.93495377686454E-06
(1;0;2;0;0;0;2;0;0;0;0)	6.20244811555455E-06	(0;0;0;1;0;2;0;0;2;0;2)	1.01494290241361E-04	(0;0;2;1;1;3;0;0;0;0;0)	6.42240974230045E-05
(0;0;0;0;0;0;0;1;2;0;0)	1.80421031040271E-02	(1;1;0;0;0;2;0;1;2;0;0)	2.47955068077240E-05	(0;0;2;1;0;3;0;0;0;0;2)	3.43934552042841E-05
(1;1;0;0;0;0;0;0;2;0;0)	1.48082611742250E-03	(1;1;0;1;0;0;2;0;2;0;0)	1.47892224102024E-06	(0;0;2;0;1;0;3;0;0;0;0)	2.45969549757435E-05
(0;0;2;0;0;3;0;0;0;0;0)	2.45920259524129E-03	(1;1;0;1;1;2;0;0;0;0;0)	5.20668953098875E-05	(0;0;2;0;0;0;3;0;0;0;2)	8.58260151663795E-06
(0;0;2;0;0;0;0;0;0;0;2)	2.65706323941528E-03	(1;1;0;1;0;2;0;0;0;0;2)	1.28721652821419E-05	(0;1;0;1;0;0;3;0;2;0;0)	1.81562173051567E-05
(0;0;0;1;0;2;0;0;2;0;0)	1.44600686673075E-03	(0;0;2;0;0;0;2;1;2;0;0)	1.96172568313736E-06	(0;1;0;1;1;3;0;0;0;0;0)	4.50822382621341E-04
(1;1;0;1;0;2;0;0;0;0;0)	2.73638212630884E-04	(0;0;2;0;1;2;0;0;2;0;0)	1.14053544663839E-04	(0;1;0;1;0;3;0;0;0;0;2)	1.40077637446038E-04
(0;0;2;1;0;0;2;0;0;0;0)	6.43758530539112E-05	(0;0;2;0;0;2;0;0;2;0;2)	7.26762815748667E-05	(0;1;0;0;1;0;3;0;0;0;0)	1.88779458288265E-04
(0;1;0;0;0;2;0;0;2;0;0)	3.61965243506323E-03	(0;0;2;1;1;0;2;0;0;0;0)	4.18341114286764E-05	(0;1;0;0;0;0;3;0;0;0;2)	3.11111035596230E-05
(0;0;0;0;0;0;2;0;2;0;0)	3.63389518380615E-04	(0;0;2;1;0;0;2;0;0;0;2)	6.50083288293486E-06	(0;0;0;1;1;0;3;0;0;0;0)	4.58512494267190E-04
(0;0;0;0;1;2;0;0;0;0;0)	7.20395210415666E-03	(1;0;2;0;0;3;0;0;2;0;0)	2.07884208240491E-05	(0;0;0;1;0;0;3;0;0;0;2)	8.26953635548983E-05
(0;0;0;0;0;2;0;0;0;0;2)	6.46771600971502E-03	(1;0;2;0;0;0;0;0;2;0;2)	1.87808960279069E-05	(1;0;2;1;0;0;2;1;2;0;0)	2.01505303936295E-08
(1;1;0;0;0;0;2;0;0;0;0)	7.28921577908374E-05	(1;0;2;0;1;0;2;0;0;0;0)	2.23229025201793E-06	(1;0;2;1;1;2;0;0;2;0;0)	5.11797896135973E-07
(0;1;0;0;0;3;0;0;0;0;0)	1.75461727088742E-02	(1;0;2;0;0;0;2;0;0;0;2)	4.61026453035487E-07	(1;0;2;1;0;2;0;0;2;0;2)	1.71777493990250E-07
(0;1;0;0;0;0;0;0;0;0;2)	1.45031764628455E-02	(0;1;0;0;0;0;2;1;2;0;0)	5.39410038163589E-06	(1;1;0;1;0;0;2;1;2;0;0)	9.32191115927284E-08
(0;0;0;1;0;3;0;0;0;0;0)	5.73750887007815E-03	(0;1;0;0;1;2;0;0;2;0;0)	5.35839553831526E-04	(1;1;0;1;1;2;0;0;2;0;0)	2.84074639205376E-06
(0;0;0;0;0;0;3;0;0;0;0)	1.47166179232687E-03	(0;1;0;0;0;2;0;0;2;0;2)	2.32455902921307E-04	(1;1;0;1;0;2;0;0;2;0;2)	9.11419455741355E-07
(0;1;0;1;0;0;2;0;0;0;0)	7.11097801395196E-04	(0;0;0;0;1;0;2;0;2;0;0)	1.07275112991839E-04	(0;0;2;1;1;0;2;0;2;0;0)	6.64129325270024E-06
(0;0;2;1;0;2;0;0;2;0;0)	8.81853767499525E-05	(0;0;0;0;0;0;2;0;2;0;2)	2.11292945102208E-05	(0;0;2;1;0;0;2;0;2;0;2)	4.62096723881788E-07
(1;0;2;0;0;2;0;0;2;0;0)	2.84889579942800E-05	(0;0;0;0;0;3;0;1;2;0;0)	1.39129463006701E-03	(0;0;0;1;1;2;0;1;2;0;0)	2.78951904708459E-05
(1;0;2;1;0;0;2;0;0;0;0)	3.81479335053543E-06	(0;0;0;0;1;2;0;0;0;0;2)	9.10108317143102E-04	(0;0;2;0;1;2;0;1;2;0;0)	1.58243525100456E-05
(0;0;0;0;0;2;0;1;2;0;0)	2.42419214261314E-03	(0;0;0;0;0;0;0;1;2;0;2)	6.42779002910734E-04	(0;0;2;1;1;2;0;0;0;0;2)	9.43733261927506E-06

(1;1;0;0;0;2;0;0;2;0;0)	2.27630451050123E-04	(0;0;0;0;0;2;0;0;0;1;2)	6.20203729231378E-04	(0;0;0;1;0;2;0;1;2;0;2)	4.15403944926126E-06
(0;0;2;0;0;0;0;1;2;0;0)	1.02949187740590E-03	(1;1;0;0;0;3;0;0;2;0;0)	1.44940805055888E-04	(0;0;2;0;0;2;0;1;2;0;2)	3.85550847067746E-06
(0;0;2;0;0;0;2;0;2;0;0)	2.21827707834500E-05	(1;1;0;0;0;0;0;0;2;0;2)	1.19555123732590E-04	(0;0;2;1;0;2;0;0;0;1;2)	3.33613310719794E-06
(0;0;2;0;1;2;0;0;0;0;0)	4.32816725456374E-04	(1;1;0;0;1;0;2;0;0;0;0)	2.05581958631181E-05	(1;0;2;0;1;0;2;0;2;0;0)	1.67401214384763E-07
(0;0;2;0;0;2;0;0;0;0;2)	3.90492551442419E-04	(1;1;0;0;0;0;2;0;0;0;2)	2.77316551699269E-06	(1;0;2;0;0;0;2;0;2;0;2)	3.29368232681430E-08
(1;0;2;0;0;3;0;0;0;0;0)	1.28422883874667E-04	(0;0;2;1;0;3;0;0;2;0;0)	6.43274124152510E-05	(1;0;2;0;0;3;0;1;2;0;0)	2.49681663237456E-06
(1;0;2;0;0;0;0;0;0;0;2)	1.19144961940497E-04	(0;0;2;0;0;0;3;0;2;0;0)	1.65012201719641E-05	(1;0;2;0;1;2;0;0;0;0;2)	2.41395550809109E-06
(0;1;0;0;0;0;0;1;2;0;0)	3.68509431387543E-03	(0;0;2;0;1;3;0;0;0;0;0)	2.45146088959612E-04	(1;0;2;0;0;0;0;1;2;0;2)	8.96141648511723E-07
(0;0;0;0;0;3;0;0;2;0;0)	9.51986276792969E-03	(0;0;2;0;0;3;0;0;0;0;2)	2.81285517525713E-04	(1;0;2;0;0;2;0;0;0;1;2)	1.52584862498962E-06
(0;0;0;0;0;0;0;0;2;0;2)	9.15738155333490E-03	(0;0;2;0;0;0;0;0;0;1;2)	2.62241642242589E-04	(0;1;0;1;1;0;2;0;2;0;0)	1.17906776785308E-05
(1;1;0;0;0;3;0;0;0;0;0)	1.10184783492542E-03	(1;0;2;1;0;0;3;0;0;0;0)	3.08519369365500E-06	(0;1;0;1;0;0;2;0;2;0;2)	1.00668109616610E-06
(1;1;0;0;0;0;0;0;0;0;2)	9.10033011463792E-04	(0;1;0;1;0;3;0;0;2;0;0)	1.71632814592640E-04	(0;1;0;0;1;2;0;1;2;0;0)	3.74335669121267E-05
(0;0;2;1;0;3;0;0;0;0;0)	3.44404419412247E-04	(0;1;0;0;0;0;3;0;2;0;0)	3.74576379587945E-05	(0;1;0;1;1;2;0;0;0;0;2)	3.71899885861321E-05
(0;0;2;0;0;0;3;0;0;0;0)	8.83648445974593E-05	(0;1;0;0;1;3;0;0;0;0;0)	1.60248975175974E-03	(0;1;0;0;0;2;0;1;2;0;2)	7.66845734722340E-06
(0;1;0;1;0;2;0;0;2;0;0)	2.48604860141755E-04	(0;1;0;0;0;3;0;0;0;0;2)	1.37080566158927E-03	(0;1;0;1;0;2;0;0;0;1;2)	1.34944223527196E-05
(0;0;0;1;0;0;2;0;2;0;0)	2.10171335953409E-04	(0;1;0;0;0;0;0;0;0;1;2)	1.21562662498203E-03	(0;0;0;1;1;0;2;0;0;0;2)	5.50778561911420E-05
(0;0;0;1;1;2;0;0;0;0;0)	1.81490802544196E-03	(0;0;0;1;0;0;3;0;2;0;0)	1.62903691345189E-04	(0;0;0;1;0;0;2;0;0;1;2)	4.71410836224852E-06
(0;0;0;1;0;2;0;0;0;0;2)	7.83318087509642E-04	(0;0;0;1;1;3;0;0;0;0;0)	1.07328631922812E-03	(0;0;0;0;1;0;2;1;2;0;0)	9.57740744519121E-06
(1;1;0;1;0;0;2;0;0;0;0)	4.50604542301820E-05	(0;0;0;1;0;3;0;0;0;0;2)	5.74175736017869E-04	(0;0;0;0;1;2;0;0;2;0;2)	1.46469404615997E-04
(0;1;0;0;0;0;2;0;2;0;0)	5.28226191895011E-05	(0;0;0;0;1;0;3;0;0;0;0)	4.10675855468307E-04	(0;0;0;0;0;0;2;1;2;0;0)	6.08007981126197E-07
(0;1;0;0;1;2;0;0;0;0;0)	3.18002385999323E-03	(0;0;0;0;0;0;3;0;0;0;2)	1.42652579371829E-04	(0;0;0;0;0;2;0;0;2;1;2)	6.18229282820648E-05
(0;1;0;0;0;2;0;0;0;0;2)	2.17439541582423E-03	(1;1;0;1;0;0;3;0;0;0;0)	3.47038336042219E-05	(1;1;0;0;1;0;2;0;2;0;0)	8.29284312935341E-07
(0;0;0;0;1;0;2;0;0;0;0)	6.73942288918631E-04	(0;1;0;1;1;0;2;0;0;0;0)	3.65981991961667E-04	(1;1;0;0;0;0;2;0;2;0;2)	1.68480658939230E-07
(0;0;0;0;0;0;2;0;0;0;2)	1.92259245541974E-04	(0;1;0;1;0;0;2;0;0;0;2)	2.02143595884242E-05	(1;1;0;0;0;3;0;1;2;0;0)	1.49756794644743E-05
(0;1;0;1;0;3;0;0;0;0;0)	3.04144999303568E-03	(1;0;2;1;0;2;0;1;2;0;0)	3.07878193086127E-07	(1;1;0;0;1;2;0;0;0;0;2)	1.73756681469636E-05
(0;1;0;0;0;0;3;0;0;0;0)	8.43245574939080E-04	(1;1;0;1;0;2;0;1;2;0;0)	1.63887219348807E-06	(1;1;0;0;0;0;0;1;2;0;2)	4.68897837286223E-06
(0;0;0;1;0;0;3;0;0;0;0)	8.67651618889577E-04	(0;0;2;1;0;0;2;1;2;0;0)	7.66168380751518E-07	(1;1;0;0;0;2;0;0;0;1;2)	9.37503171179162E-06
(1;0;2;1;0;2;0;0;2;0;0)	2.57784778930913E-06	(0;0;2;1;1;2;0;0;2;0;0)	1.77576078177308E-05	(0;0;2;0;1;0;2;0;0;0;2)	3.33562373572368E-06
(0;0;0;1;0;2;0;1;2;0;0)	1.83071944094837E-04	(0;0;2;1;0;2;0;0;2;0;2)	5.66075886837485E-06	(0;0;2;0;0;0;2;0;0;1;2)	6.72451832187334E-07
(1;1;0;1;0;2;0;0;2;0;0)	1.55544571419268E-05	(1;0;2;0;0;0;2;1;2;0;0)	5.85947193513882E-08	(0;0;2;1;0;3;0;1;2;0;0)	6.73960886705624E-06
(0;0;2;0;0;2;0;1;2;0;0)	1.37932001141462E-04	(1;0;2;0;1;2;0;0;2;0;0)	4.48296840508037E-06	(0;0;2;0;0;0;3;1;2;0;0)	1.29534518962089E-06
(0;0;2;1;0;0;2;0;2;0;0)	1.28479819283734E-05	(1;0;2;0;0;2;0;0;2;0;2)	2.43347762364504E-06	(0;0;2;0;1;3;0;0;2;0;0)	3.27316694944293E-05
(0;0;2;1;1;2;0;0;0;0;0)	1.08978495252180E-04	(1;0;2;1;1;0;2;0;0;0;0)	2.35442062830203E-06	(0;0;2;0;0;3;0;0;2;0;2)	4.21970997302298E-05
(0;0;2;1;0;2;0;0;0;0;2)	4.73746923388049E-05	(1;0;2;1;0;0;2;0;0;0;2)	2.43631000670064E-07	(0;0;2;0;0;0;0;0;2;1;2)	2.86740006468379E-05
(1;0;2;0;0;0;0;1;2;0;0)	3.54027955420637E-05	(0;1;0;1;0;0;2;1;2;0;0)	1.82721627793029E-06	(1;0;2;1;0;0;3;0;2;0;0)	2.50305515163024E-07
(1;0;2;0;0;0;2;0;2;0;0)	6.09181115507085E-07	(0;1;0;1;1;2;0;0;2;0;0)	4.69795468228859E-05	(1;0;2;1;1;3;0;0;0;0;0)	3.39792987331730E-06
(1;0;2;0;1;2;0;0;0;0;0)	2.28017303129321E-05	(0;1;0;1;0;2;0;0;2;0;2)	1.56342218528210E-05	(1;0;2;1;0;3;0;0;0;0;2)	1.40059282518241E-06
(1;0;2;0;0;2;0;0;0;0;2)	1.75595390934377E-05	(0;0;0;1;1;0;2;0;2;0;0)	1.10459832580926E-04	(1;0;2;0;1;0;3;0;0;0;0)	1.33574985783244E-06
(0;1;0;0;0;2;0;1;2;0;0)	4.13356361444749E-04	(0;0;0;1;0;0;2;0;2;0;2)	8.92992327180330E-06	(1;0;2;0;0;0;3;0;0;0;2)	3.36161493769636E-07
(0;0;0;0;0;0;2;1;2;0;0)	3.76041394501610E-05	(0;0;0;0;1;2;0;1;2;0;0)	2.79699236847755E-04	(0;1;0;0;1;0;2;0;0;0;2)	1.12274921884571E-05
(0;0;0;0;1;2;0;0;2;0;0)	1.91855649026398E-03	(0;0;0;1;1;2;0;0;0;0;2)	1.59217552957995E-04	(0;1;0;0;0;0;2;0;0;1;2)	2.62986295809146E-06
(0;0;0;0;0;2;0;0;2;0;2)	1.25576672581939E-03	(0;0;0;0;0;2;0;1;2;0;2)	7.34967099936021E-05	(0;1;0;1;0;3;0;1;2;0;0)	1.95877005062423E-05
(1;1;0;0;0;0;0;1;2;0;0)	2.23337522614849E-04	(0;0;0;1;0;2;0;0;0;1;2)	5.81003830813026E-05	(0;1;0;0;0;0;3;1;2;0;0)	3.47862174175530E-06
(1;1;0;0;0;0;2;0;2;0;0)	3.29568896581952E-06	(1;1;0;0;0;0;2;1;2;0;0)	3.00909306320206E-07	(0;1;0;0;1;3;0;0;2;0;0)	7.32264133560568E-05
(1;1;0;0;1;2;0;0;0;0;0)	1.99628645815403E-04	(1;1;0;0;1;2;0;0;2;0;0)	3.32181255743534E-05	(0;1;0;0;0;3;0;0;2;0;2)	1.38433610366395E-04
(1;1;0;0;0;2;0;0;0;0;2)	1.36518134818711E-04	(1;1;0;0;0;2;0;0;2;0;2)	1.40458619157140E-05	(0;1;0;0;0;0;0;0;2;1;2)	7.71213837803346E-05
(0;0;2;0;0;3;0;0;2;0;0)	5.64209436656508E-04	(1;1;0;1;1;0;2;0;0;0;0)	2.30312269474677E-05	(0;0;0;1;0;0;3;1;2;0;0)	1.19217537894912E-05
(0;0;2;0;0;0;0;0;2;0;2)	5.30421067374982E-04	(1;1;0;1;0;0;2;0;0;0;2)	1.25675106026582E-06	(0;0;0;1;1;3;0;0;2;0;0)	1.47331776354005E-04
(0;0;2;0;1;0;2;0;0;0;0)	4.05085743323686E-05	(0;0;2;0;1;0;2;0;2;0;0)	6.38166771558539E-06	(0;0;0;1;0;3;0;0;2;0;2)	6.91837660036910E-05
(0;0;2;0;0;0;2;0;0;0;2)	1.16343793993048E-05	(0;0;2;0;0;0;2;0;2;0;2)	1.14945848932153E-06	(0;0;0;0;1;0;3;0;2;0;0)	5.80433070511428E-05
(1;0;2;1;0;3;0;0;0;0;0)	1.97744093494000E-05	(0;0;2;0;0;3;0;1;2;0;0)	7.83277101072216E-05	(0;0;0;0;0;0;3;0;2;0;2)	1.41987990728273E-05
(1;0;2;0;0;0;3;0;0;0;0)	5.14983503840774E-06	(0;0;2;0;1;2;0;0;0;0;2)	5.40223894943909E-05	(0;0;0;0;1;3;0;0;0;0;2)	2.93729877756041E-04
(0;0;0;0;0;3;0;0;0;1;2)	3.43015385785342E-04	(1;1;0;1;1;0;2;0;0;0;2)	5.88052105701882E-07	(0;0;0;0;0;0;2;1;2;1;2)	3.55078554770657E-08

(1;1;0;1;0;0;3;0;2;0;0)	1.11607996584185E-06	(1;1;0;1;0;0;2;0;0;1;2)	4.82951274445992E-08	(1;0;2;1;1;0;3;0;2;0;0)	1.02081647586495E-07
(1;1;0;1;1;3;0;0;0;0;0)	2.80482958976601E-05	(1;1;0;0;1;0;2;1;2;0;0)	5.79401565316136E-08	(1;0;2;1;0;0;3;0;2;0;2)	8.41860875550119E-09
(1;1;0;1;0;3;0;0;0;0;2)	8.63535520104705E-06	(1;1;0;0;1;2;0;0;2;0;2)	1.32927937187858E-06	(0;0;2;1;1;3;0;1;2;0;0)	3.85099204677018E-07
(1;1;0;0;1;0;3;0;0;0;0)	1.17883341705891E-05	(1;1;0;0;0;0;2;1;2;0;2)	2.02012257203459E-09	(1;0;2;1;1;3;0;0;0;0;2)	1.60277745750352E-07
(1;1;0;0;0;0;3;0;0;0;2)	1.92176418575115E-06	(1;1;0;0;0;2;0;0;2;1;2)	4.82138916692523E-07	(1;0;2;1;0;3;0;0;0;1;2)	8.66821061513048E-08
(0;0;2;1;1;0;3;0;0;0;0)	2.74983794396497E-05	(0;1;0;1;1;0;2;1;2;0;0)	7.59406587195041E-07	(0;0;2;0;1;0;3;1;2;0;0)	1.57120386420470E-07
(0;0;2;1;0;0;3;0;0;0;2)	5.00062306475245E-06	(0;0;0;1;1;0;2;0;2;0;2)	4.09734651441063E-06	(1;0;2;0;1;0;3;0;0;0;2)	6.11873165577232E-08
(0;1;0;1;1;0;3;0;0;0;0)	2.34487747673734E-04	(0;1;0;1;0;0;2;1;2;0;2)	1.04464807907990E-08	(1;0;2;0;0;0;3;0;0;1;2)	1.64961474457802E-08
(0;1;0;1;0;0;3;0;0;0;2)	1.53090094772375E-05	(0;0;0;1;0;0;2;0;2;1;2)	2.49506170831870E-07	(1;0;2;0;1;3;0;1;2;0;0)	2.71866508166477E-08
(1;0;2;1;1;0;2;0;2;0;0)	1.63598772036408E-07	(0;1;0;1;1;2;0;0;2;0;2)	2.00624455679641E-06	(1;1;0;0;1;3;0;1;2;0;0)	1.00067057523343E-07
(1;0;2;1;0;0;2;0;2;0;2)	1.15368377658926E-08	(0;0;0;0;1;2;0;1;2;0;2)	5.54525441588957E-06	(0;1;0;0;1;0;2;0;0;1;2)	5.82004456989500E-07
(0;0;2;1;1;2;0;1;2;0;0)	1.45158097247371E-06	(0;0;0;1;1;2;0;0;0;1;2)	8.35667559354865E-06	(0;1;0;1;0;3;0;1;2;0;2)	1.46499754486771E-06
(1;0;2;0;1;2;0;1;2;0;0)	4.50103728641239E-07	(0;0;2;0;1;0;2;0;2;0;2)	2.85006784830113E-07	(0;1;0;0;0;0;3;1;2;0;2)	2.88564600881916E-08
(1;0;2;1;1;2;0;0;0;0;2)	3.75820109238008E-07	(0;0;2;0;0;3;0;1;2;0;2)	1.84232813850127E-06	(0;1;0;0;1;3;0;0;2;0;2)	1.54904372781861E-06
(0;0;2;1;0;2;0;1;2;0;2)	2.03334592441833E-07	(0;0;2;0;1;2;0;0;0;1;2)	3.60019027642103E-06	(0;1;0;0;0;3;0;0;2;1;2)	4.03149761947641E-06
(1;0;2;0;0;2;0;1;2;0;2)	9.01217164399207E-08	(0;1;0;1;0;2;0;0;2;1;2)	4.42172435325097E-07	(0;0;0;1;0;0;3;1;2;0;2)	1.72745083284939E-07
(1;0;2;1;0;2;0;0;0;1;2)	1.36927497265526E-07	(0;0;0;0;0;2;0;1;2;1;2)	3.20885839712204E-06	(0;0;0;1;1;3;0;0;2;0;2)	5.62391182100287E-06
(1;1;0;1;1;0;2;0;2;0;0)	7.08947810078864E-07	(0;0;2;0;0;0;2;0;2;1;2)	4.20791811920535E-08	(0;0;0;1;0;3;0;0;2;1;2)	4.99533119716354E-06
(1;1;0;1;0;0;2;0;2;0;2)	5.26109730852353E-08	(0;0;2;0;0;0;0;1;2;1;2)	1.47823255126634E-06	(0;0;0;0;1;0;3;0;2;0;2)	1.55328045969670E-06
(1;1;0;0;1;2;0;1;2;0;0)	2.20281297047176E-06	(1;0;2;1;0;0;3;1;2;0;0)	1.46741890768622E-08	(0;0;0;0;0;0;3;0;2;1;2)	4.67035439592870E-07
(1;1;0;1;1;2;0;0;0;0;2)	2.25245186600763E-06	(1;0;2;1;1;3;0;0;2;0;0)	2.18429096129956E-07	(0;0;0;0;1;3;0;0;0;1;2)	1.02002453945011E-05
(1;1;0;0;0;2;0;1;2;0;2)	4.33447154483533E-07	(1;0;2;1;0;3;0;0;2;0;2)	1.13508590460874E-07	(1;1;0;1;1;0;3;0;2;0;0)	2.89847688185073E-07
(1;1;0;1;0;2;0;0;0;1;2)	7.99392665210023E-07	(1;0;2;0;1;0;3;0;2;0;0)	8.39962582484972E-08	(1;1;0;1;0;0;3;0;2;0;2)	3.10934142934051E-08
(0;0;0;1;1;0;2;1;2;0;0)	7.20951857709993E-06	(1;0;2;0;0;0;3;0;2;0;2)	2.10012158770107E-08	(1;1;0;1;1;3;0;0;0;0;2)	8.13674525216886E-07
(0;0;2;1;1;0;2;0;0;0;2)	3.29243715797847E-06	(0;0;2;0;1;3;0;1;2;0;0)	1.30871019280414E-06	(1;1;0;1;0;3;0;0;0;1;2)	4.99641415976758E-07
(0;0;0;1;0;0;2;1;2;0;2)	2.44544270857587E-07	(1;0;2;0;1;3;0;0;0;0;2)	6.34613982734069E-07	(1;1;0;0;1;0;3;0;0;0;2)	3.00957461223008E-07
(0;0;2;1;0;0;2;0;0;1;2)	2.57305100467224E-07	(1;0;2;0;0;3;0;0;0;1;2)	7.87719125239495E-07	(1;1;0;0;0;0;3;0;0;1;2)	9.15455628898037E-08
(0;1;0;1;1;2;0;1;2;0;0)	3.51372993733718E-06	(0;1;0;0;1;0;2;0;2;0;2)	7.42784987802858E-07	(0;0;0;1;1;0;3;1;2;0;0)	4.33425845053184E-06
(0;0;0;1;1;2;0;0;2;0;2)	1.58925622804561E-05	(0;1;0;0;0;3;0;1;2;0;2)	3.48206784671896E-06	(0;0;2;1;1;0;3;0;0;0;2)	2.07684380538215E-06
(0;0;2;0;1;0;2;1;2;0;0)	3.53838045905253E-07	(0;1;0;0;1;2;0;0;0;1;2)	1.26180554974277E-05	(0;0;2;1;0;0;3;0;0;1;2)	1.88253905734342E-07
(0;0;2;0;1;2;0;0;2;0;2)	8.39242650235961E-06	(0;1;0;0;0;0;2;0;2;1;2)	1.13211345552953E-07	(0;1;0;1;1;3;0;1;2;0;0)	7.46558312935050E-07
(0;1;0;1;0;2;0;1;2;0;2)	2.43945295941092E-07	(0;1;0;0;0;0;0;1;2;1;2)	2.98488408093012E-06	(0;1;0;0;1;0;3;1;2;0;0)	3.16574923953164E-07
(0;0;0;1;0;2;0;0;2;1;2)	4.18136690259384E-06	(0;0;0;0;1;0;2;0;0;1;2)	2.97351238598006E-06	(0;1;0;1;1;0;3;0;0;0;2)	5.96708506473860E-06
(0;0;2;0;0;0;2;1;2;0;2)	3.44458505493184E-08	(0;0;0;1;0;3;0;1;2;0;2)	1.19798035637637E-05	(0;1;0;1;0;0;3;0;0;1;2)	6.43009331632983E-07
(0;0;2;0;0;2;0;0;2;1;2)	3.41074502744140E-06	(0;0;0;0;0;0;3;1;2;0;2)	4.69901131003456E-07	(1;0;2;1;1;0;2;0;2;0;2)	5.13177388638045E-09
(1;0;2;0;1;0;2;0;0;0;2)	1.26901041116327E-07	(0;0;0;0;1;3;0;0;2;0;2)	1.86803192488245E-05	(1;1;0;1;1;0;2;0;2;0;2)	1.59970194519780E-08
(1;0;2;0;0;0;2;0;0;1;2)	2.66775082216122E-08	(0;0;0;0;0;3;0;0;2;1;2)	3.15330955853694E-05	(0;0;0;1;1;0;2;1;2;0;2)	1.02920172657347E-07
(1;0;2;1;0;3;0;1;2;0;0)	2.07630288468229E-07	(1;1;0;1;0;0;3;1;2;0;0)	4.24621110025614E-08	(0;0;2;1;1;0;2;0;0;1;2)	1.08334478376112E-07
(1;0;2;0;0;0;3;1;2;0;0)	3.72367091718998E-08	(1;1;0;1;1;3;0;0;2;0;0)	9.99619240607376E-07	(1;0;2;1;0;0;2;0;2;1;2)	2.11521502918414E-10
(1;0;2;0;1;3;0;0;2;0;0)	8.71891683138108E-07	(1;1;0;1;0;3;0;0;2;0;2)	6.08520952188813E-07	(1;1;0;1;0;0;2;0;2;1;2)	8.26092636949692E-10
(1;0;2;0;0;3;0;0;2;0;2)	1.38902478274875E-06	(1;1;0;0;1;0;3;0;2;0;0)	3.68305328586819E-07	(0;0;0;1;0;0;2;1;2;1;2)	6.38566159908932E-09
(1;0;2;0;0;0;0;0;2;1;2)	8.41255653142837E-07	(1;1;0;0;0;0;3;0;2;0;2)	1.06873235098960E-07	(0;0;2;1;1;2;0;1;2;0;2)	1.98080515544241E-08
(0;1;0;1;1;0;2;0;0;0;2)	9.67077031263557E-06	(1;1;0;0;1;3;0;0;0;0;2)	3.45606322097989E-06	(1;0;2;0;1;2;0;1;2;0;2)	5.48507623510853E-09
(0;1;0;1;0;0;2;0;0;1;2)	8.85458577824525E-07	(1;1;0;0;0;3;0;0;0;1;2)	4.74066552397605E-06	(1;0;2;1;1;2;0;0;0;1;2)	1.77095777338950E-08
(0;1;0;0;1;0;2;1;2;0;0)	1.18799261167814E-06	(0;0;2;1;1;0;3;0;2;0;0)	4.21067466378142E-06	(1;1;0;0;1;2;0;1;2;0;2)	3.84498962557302E-08
(0;1;0;0;1;2;0;0;2;0;2)	2.23777415731683E-05	(0;0;2;1;0;0;3;0;2;0;2)	3.43110015128377E-07	(1;1;0;1;1;2;0;0;0;1;2)	8.35894625205216E-08
(0;1;0;0;0;0;2;1;2;0;2)	6.34655087607579E-08	(0;0;0;1;1;3;0;1;2;0;0)	8.34996740131258E-06	(0;1;0;1;1;2;0;1;2;0;2)	1.02783339584417E-07
(0;1;0;0;0;2;0;0;2;1;2)	8.34810202107456E-06	(0;0;2;1;1;3;0;0;0;0;2)	4.40791882246962E-06	(0;0;0;1;1;2;0;0;2;1;2)	4.59371365896784E-07
(0;0;0;0;1;0;2;0;2;0;2)	5.74386613807078E-06	(0;0;2;1;0;3;0;0;0;1;2)	2.17489880176981E-06	(0;0;2;0;1;0;2;1;2;0;2)	8.54601907905746E-08
(0;0;0;0;0;3;0;1;2;0;2)	3.49680435533801E-05	(0;0;0;0;1;0;3;1;2;0;0)	3.43686632705322E-06	(0;0;2;0;1;2;0;0;2;1;2)	2.65623805554984E-07
(0;0;0;0;1;2;0;0;0;1;2)	6.22487285263297E-05	(0;0;2;0;1;0;3;0;0;0;2)	1.71581557332843E-06	(1;0;2;0;1;0;2;0;0;1;2)	9.28530911816714E-09
(0;0;0;0;0;0;2;0;2;1;2)	9.12569904831161E-07	(0;0;2;0;0;0;3;0;0;1;2)	4.29311030678459E-07	(1;0;2;1;0;3;0;1;2;0;2)	1.73175706622313E-08
(0;0;0;0;0;0;0;1;2;1;2)	2.90382369140115E-05	(0;1;0;0;1;3;0;1;2;0;0)	2.17070865465402E-06	(1;0;2;0;0;0;3;1;2;0;2)	3.63067960355442E-10
(1;1;0;0;1;0;2;0;0;0;2)	6.84072313061823E-07	(0;1;0;1;1;0;3;0;2;0;0)	7.26020553136727E-06	(1;0;2;0;1;3;0;0;2;0;2)	1.96867804384951E-08
(1;1;0;0;0;0;2;0;0;1;2)	1.51308714798157E-07	(0;1;0;1;0;0;3;0;2;0;2)	7.46584985326341E-07	(1;0;2;0;0;3;0;0;2;1;2)	4.74092179872217E-08
(1;1;0;1;0;3;0;1;2;0;0)	1.11971807500127E-06	(0;1;0;1;1;3;0;0;0;0;2)	1.36665296743286E-05	(0;0;2;1;0;2;0;1;2;1;2)	5.79733164677772E-09
(1;1;0;0;0;0;3;1;2;0;0)	1.91887880283775E-07	(0;1;0;1;0;3;0;0;0;1;2)	8.53208730644297E-06	(1;0;2;0;0;2;0;1;2;1;2)	3.85141943008976E-09
(1;1;0;0;1;3;0;0;2;0;0)	4.37124509747509E-06	(0;1;0;0;1;0;3;0;0;0;2)	5.06023110742853E-06	(1;1;0;0;0;2;0;1;2;1;2)	1.67603021873157E-08
(1;1;0;0;0;3;0;0;2;0;2)	8.31994091552818E-06	(0;1;0;0;0;0;3;0;0;1;2)	1.59711976217147E-06	(0;1;0;1;0;2;0;1;2;1;2)	1.05189968705115E-08
(1;1;0;0;0;0;0;0;2;1;2)	4.54762633084949E-06	(0;0;0;1;1;0;3;0;0;0;2)	3.51072709440249E-05	(0;0;2;0;0;0;2;1;2;1;2)	1.22918049143502E-09
(0;0;2;1;0;0;3;1;2;0;0)	5.67143636354052E-07	(0;0;0;1;0;0;3;0;0;1;2)	3.48192556709216E-06	(1;1;0;0;1;0;2;0;0;1;2)	3.61274528698180E-08
(0;0;2;1;1;3;0;0;2;0;0)	8.65567676873140E-06	(1;0;2;1;1;0;2;1;2;0;0)	8.59473617315153E-09	(1;1;0;1;0;3;0;1;2;0;2)	8.28817336728645E-08
(0;0;2;1;0;3;0;0;2;0;2)	3.80763355000420E-06	(1;1;0;1;1;0;2;1;2;0;0)	3.50701105326235E-08	(1;1;0;0;0;0;3;1;2;0;2)	1.44106379233267E-09
(0;0;2;0;1;0;3;0;2;0;0)	3.44574552347624E-06	(0;0;2;1;1;0;2;0;2;0;2)	2.04261046797424E-07	(1;1;0;0;1;3;0;0;2;0;2)	7.71948381422084E-08
(0;0;2;0;0;0;3;0;2;0;2)	7.62330212524890E-07	(1;0;2;1;0;0;2;1;2;0;2)	1.42920913299411E-10	(1;1;0;0;0;3;0;0;2;1;2)	2.31646829525923E-07
(0;0;0;0;1;3;0;1;2;0;0)	2.80680590160329E-05	(1;1;0;1;0;0;2;1;2;0;2)	5.45914467625836E-10	(0;1;0;1;1;0;2;0;0;1;2)	3.50704557757924E-07
(0;0;2;0;1;3;0;0;0;0;2)	1.71596449281686E-05	(0;0;2;1;0;0;2;0;2;1;2)	1.20888793870165E-08	(0;1;0;0;1;0;2;1;2;0;2)	1.52115365974506E-07
(0;0;2;0;0;3;0;0;0;1;2)	1.97763641275975E-05	(1;0;2;1;1;2;0;0;2;0;2)	2.32083183847877E-08	(0;1;0;0;1;2;0;0;2;1;2)	5.94682148131775E-07
(1;0;2;1;1;0;3;0;0;0;0)	1.53647549902263E-06	(1;1;0;1;1;2;0;0;2;0;2)	8.77426277210033E-08	(0;0;0;0;1;0;2;0;2;1;2)	1.27717122687466E-06
(1;0;2;1;0;0;3;0;0;0;2)	1.86410020479147E-07	(0;0;0;1;1;2;0;1;2;0;2)	4.03917004630465E-07	(0;0;0;0;0;0;3;0;1;2;1;2)	2.88471300095342E-06
(0;1;0;1;0;0;3;1;2;0;0)	1.35839567892817E-06	(0;0;2;0;1;2;0;1;2;0;2)	2.74035887928733E-07	(0;0;2;1;0;0;3;1;2;0;2)	6.54086343263464E-09
(0;1;0;1;1;3;0;0;2;0;0)	1.68542022546867E-05	(0;0;2;1;1;2;0;0;0;1;2)	4.69656751890092E-07	(0;0;2;1;1;3;0;0;2;0;2)	2.83185134840748E-07
(0;1;0;1;0;3;0;0;2;0;2)	1.05461464006529E-05	(1;0;2;0;1;0;2;0;2;0;2)	6.96349834604765E-09	(0;0;2;1;0;3;0;0;2;1;2)	2.69446910466914E-07

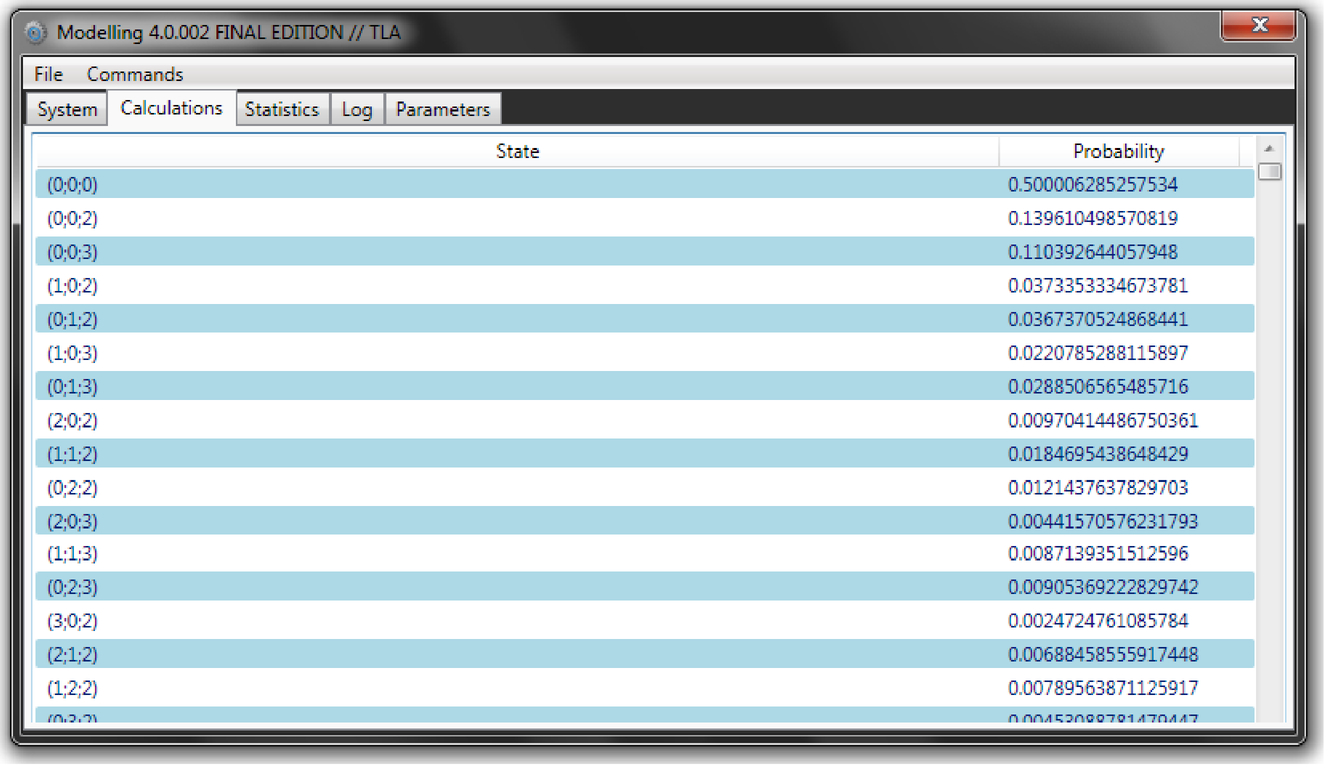
(0;1;0;0;1;0;3;0;2;0;0)	6.14764457683309E-06	(1;0;2;0;0;3;0;1;2;0;2)	4.23471316217029E-08	(0;0;2;0;1;0;3;0;2;0;2)	7.58689273870970E-08
(0;1;0;0;0;0;3;0;2;0;2)	1.90882599439192E-06	(1;0;2;0;1;2;0;0;0;1;2)	1.38980925154313E-07	(0;0;2;0;0;0;3;0;2;1;2)	2.39616977643175E-08
(0;1;0;0;1;3;0;0;0;0;2)	5.76864767947464E-05	(1;0;2;1;0;2;0;0;2;1;2)	5.86938398107188E-09	(0;0;0;0;1;3;0;1;2;0;2)	3.00211714279762E-07
(0;1;0;0;0;3;0;0;0;1;2)	7.91266182561113E-05	(1;1;0;1;0;2;0;0;2;1;2)	2.19552986489703E-08	(0;0;2;0;1;3;0;0;0;1;2)	5.49943866800934E-07
(0;0;0;1;1;0;3;0;2;0;0)	6.99222044938600E-05	(0;0;0;1;0;2;0;1;2;1;2)	1.51617046333152E-07	(0;1;0;0;0;0;2;1;2;1;2)	7.13852495073268E-08
(0;0;0;1;0;0;3;0;2;0;2)	6.70730338316466E-06	(0;0;2;0;0;2;0;1;2;1;2)	1.58916710077578E-07	(0;0;2;1;1;0;3;1;2;0;0)	1.80158595544266E-07
(0;0;0;1;1;3;0;0;0;0;2)	7.51442015304664E-05	(1;0;2;0;0;0;2;0;2;1;2)	8.63788698829401E-10	(1;0;2;1;1;0;3;0;0;0;2)	7.42725128241174E-08
(0;0;0;1;0;3;0;0;0;1;2)	3.82967870094421E-05	(1;0;2;0;0;0;0;1;2;1;2)	3.53679967528147E-08	(1;0;2;1;0;0;3;0;0;1;2)	6.77073805697289E-09
(0;0;0;0;1;0;3;0;0;0;2)	2.92812076582898E-05	(1;1;0;0;1;0;2;0;2;0;2)	3.71738306793750E-08	(1;0;2;1;1;3;0;1;2;0;0)	9.00678283292350E-09
(0;0;0;0;0;0;3;0;0;1;2)	7.67363295539212E-06	(1;1;0;0;0;3;0;1;2;0;2)	1.99557137410540E-07	(1;1;0;1;1;3;0;1;2;0;0)	3.39639797244801E-08
(1;1;0;1;1;0;3;0;0;0;0)	1.48640953834375E-05	(1;1;0;0;1;2;0;0;0;1;2)	7.50648141490355E-07	(1;0;2;0;1;0;3;1;2;0;0)	3.57567742634817E-09
(1;1;0;1;0;0;3;0;0;0;2)	9.41754479076539E-07	(1;1;0;0;0;0;2;0;2;1;2)	4.11237152486301E-09	(1;1;0;0;1;0;3;1;2;0;0)	1.47532065825391E-08
(0;0;2;1;1;0;2;1;2;0;0)	3.32483673709784E-07	(1;1;0;0;0;0;0;1;2;1;2)	1.64888543401615E-07	(0;1;0;1;0;0;3;1;2;0;2)	2.22770055547277E-08
(1;0;2;1;1;0;2;0;0;0;2)	1.19772842932150E-07	(0;1;0;1;1;0;2;0;2;0;2)	4.33295660424022E-07	(0;1;0;1;1;3;0;0;2;0;2)	5.53238361233136E-07
(0;0;2;1;0;0;2;1;2;0;2)	1.03874895545252E-08	(0;0;0;1;1;0;2;0;0;1;2)	2.01033735629749E-06	(0;1;0;1;0;3;0;0;2;1;2)	6.63360249543683E-07
(1;0;2;1;0;0;2;0;0;1;2)	9.40474721123212E-09	(0;1;0;1;0;0;2;0;2;1;2)	2.04470712351896E-08	(0;1;0;0;1;0;3;0;2;0;2)	1.27463628260155E-07
(1;0;2;1;1;2;0;1;2;0;0)	4.03041247830050E-08	(0;1;0;0;1;2;0;1;2;0;2)	4.55044160914607E-07	(0;1;0;0;0;0;3;0;2;1;2)	3.77132475041332E-08
(1;1;0;1;1;2;0;1;2;0;0)	1.90775924459118E-07	(0;1;0;1;1;2;0;0;0;1;2)	1.62764623369544E-06	(0;1;0;0;1;3;0;0;0;1;2)	1.53349050478445E-06
(0;0;2;1;1;2;0;0;2;0;2)	8.60170592772472E-07	(0;0;0;0;1;0;2;1;2;0;2)	1.38801318567295E-06	(0;0;0;1;1;0;3;0;2;0;2)	1.94780584085989E-06
(1;0;2;0;1;0;2;1;2;0;0)	7.93420604472055E-09	(0;0;0;0;1;2;0;0;2;1;2)	5.03495192640067E-06	(0;0;0;1;0;0;3;0;2;1;2)	1.90546520505267E-07
(1;0;2;0;1;2;0;0;2;0;2)	2.59003758339335E-07	(0;0;2;0;1;0;2;0;0;1;2)	2.44853602590109E-07	(0;0;0;1;1;3;0;0;0;1;2)	2.98417418518060E-06
(1;0;2;1;0;2;0;1;2;0;2)	3.55220121366363E-09	(0;0;2;1;0;3;0;1;2;0;2)	6.32213918727211E-07	(0;0;0;0;1;0;3;0;0;1;2)	1.45040103978799E-06
(1;1;0;1;0;2;0;1;2;0;2)	1.61980701062952E-08	(0;0;2;0;0;0;3;1;2;0;2)	2.16860865550002E-08	(1;1;0;1;1;0;3;0;0;0;2)	4.17088474404668E-07
(0;0;2;1;0;2;0;0;2;1;2)	2.22985380631097E-07	(0;0;2;0;1;3;0;0;2;0;2)	9.45839019974891E-07	(1;1;0;1;0;0;3;0;0;1;2)	3.12944905031524E-08
(1;0;2;0;0;0;2;1;2;0;2)	6.22664931146921E-10	(0;0;2;0;0;3;0;0;2;1;2)	1.72349378092718E-06	(0;1;0;1;1;0;3;1;2;0;0)	4.36494334653280E-07
(1;0;2;0;0;2;0;0;2;1;2)	9.42677976356067E-08	(0;1;0;0;0;2;0;1;2;1;2)	2.52547699136616E-07	(0;0;2;1;1;0;2;1;2;0;2)	4.33451723292042E-09
(1;0;2;1;1;0;2;0;0;1;2)	3.93993613210528E-09	(0;0;2;1;1;0;2;0;2;1;2)	6.43578013888384E-09	(0;0;2;1;0;0;3;1;2;1;2)	4.76678766406132E-10
(1;1;0;1;1;0;2;0;0;1;2)	2.28587859303450E-08	(0;1;0;1;1;0;2;0;2;1;2)	3.85061160774704E-08	(0;0;2;1;1;3;0;0;2;1;2)	6.56114183403428E-09
(0;1;0;1;1;0;2;1;2;0;2)	2.89505708422138E-08	(1;0;2;1;0;0;2;1;2;1;2)	5.76840308683894E-11	(0;0;2;0;1;0;3;0;2;1;2)	3.25215451383315E-09
(0;0;0;1;1;0;2;0;2;1;2)	1.65649472788035E-07	(1;1;0;1;0;0;2;1;2;1;2)	1.55574464365070E-11	(0;0;0;0;1;3;0;1;2;1;2)	2.86792524273365E-08
(0;0;2;1;0;0;2;1;2;1;2)	2.25407279760494E-10	(1;0;2;1;1;2;0;0;2;1;2)	6.62250421522822E-10	(0;0;2;1;1;0;3;1;2;0;2)	2.04101219132420E-09
(0;1;0;1;0;0;2;1;2;1;2)	1.67616596309792E-09	(1;1;0;1;1;2;0;0;2;1;2)	3.00351313774229E-09	(1;0;2;1;1;0;3;0;0;1;2)	1.97793685745556E-09
(1;0;2;1;1;2;0;1;2;0;2)	3.29317209274682E-10	(0;0;0;1;1;2;0;1;2;1;2)	3.97959374811992E-08	(1;0;2;1;1;3;0;1;2;0;2)	9.43451516681195E-10
(1;1;0;1;1;2;0;1;2;0;2)	1.22232832378518E-09	(0;0;2;0;1;2;0;1;2;1;2)	1.54543055404017E-08	(1;1;0;1;1;3;0;1;2;0;2)	2.19281231219501E-10
(0;0;2;1;1;2;0;0;2;1;2)	2.28706025959585E-08	(1;0;2;0;1;0;2;0;2;1;2)	1.55723073914305E-09	(1;0;2;0;1;0;3;1;2;0;2)	2.87633604247723E-11
(1;0;2;0;1;0;2;1;2;0;2)	2.34998850038077E-09	(1;0;2;0;0;3;0;1;2;1;2)	4.04084809804016E-09	(1;1;0;0;1;0;3;1;2;0;2)	8.40532789879603E-11
(1;0;2;0;1;2;0;0;2;1;2)	6.92567558000754E-09	(1;1;0;0;1;0;2;0;2;1;2)	8.75244765955854E-09	(1;1;0;1;1;0;3;0;0;1;2)	9.06650554496477E-09
(1;1;0;0;1;0;2;1;2;0;2)	1.44651846145191E-08	(1;1;0;0;0;3;0;1;2;1;2)	1.57281048186098E-08	(0;1;0;1;0;0;3;1;2;1;2)	2.03220802337137E-08
(1;1;0;0;1;2;0;0;2;1;2)	3.52609644619634E-08	(0;1;0;0;1;2;0;1;2;1;2)	3.57862612983442E-08	(0;1;0;1;1;3;0;0;2;1;2)	2.83270209754983E-08
(0;1;0;1;1;2;0;0;2;1;2)	1.10793942003518E-07	(0;0;0;0;1;0;2;1;2;1;2)	4.79645109593322E-08	(0;1;0;0;1;0;3;0;2;1;2)	6.24554433696367E-09
(0;0;0;0;1;2;0;1;2;1;2)	4.42690057598261E-07	(0;0;2;1;0;3;0;1;2;1;2)	8.14675219522944E-09	(0;0;0;1;1;0;3;0;2;1;2)	1.00452883318825E-07
(0;0;2;0;1;0;2;0;2;1;2)	5.17869194225435E-08	(0;0;2;0;0;0;3;1;2;1;2)	5.65898030829897E-10	(0;1;0;1;1;0;3;1;2;0;2)	2.00524852651363E-08
(0;0;2;0;0;3;0;1;2;1;2)	1.46116179325269E-07	(0;0;2;0;1;3;0;0;2;1;2)	2.70545673986163E-08	(0;0;2;1;1;0;2;1;2;1;2)	4.37387402672598E-10
(1;0;2;1;0;0;3;1;2;0;2)	1.03468924678749E-10	(1;0;2;1;1;0;3;0;2;0;2)	2.28602476012899E-09	(0;1;0;1;1;0;2;1;2;1;2)	1.15706684822285E-09
(1;0;2;1;1;3;0;0;2;0;2)	6.70911797403743E-09	(1;0;2;1;0;0;3;0;2;1;2)	1.58583504348532E-10	(1;0;2;1;1;2;0;1;2;1;2)	2.73139959705969E-11
(1;0;2;1;0;3;0;0;2;1;2)	7.66238931344966E-09	(0;0;2;1;1;3;0;1;2;0;2)	4.01220833611444E-09	(1;1;0;1;1;2;0;1;2;1;2)	1.08188284731915E-10
(1;0;2;0;1;0;3;0;2;0;2)	1.51828261941495E-09	(1;0;2;1;1;3;0;0;0;1;2)	5.51045620448453E-09	(1;0;2;0;1;0;2;1;2;1;2)	3.44161912340145E-11
(1;0;2;0;0;0;3;0;2;1;2)	4.81405011483025E-10	(0;0;2;0;1;0;3;1;2;0;2)	1.84152182451409E-09	(1;1;0;0;1;0;2;1;2;1;2)	1.27291727703848E-10
(0;0;2;0;1;3;0;1;2;0;2)	1.10569926398922E-08	(1;0;2;0;1;0;3;0;0;1;2)	1.49496109617126E-09	(1;0;2;1;0;0;3;1;2;1;2)	6.65592603637011E-12
(1;0;2;0;1;3;0;0;0;1;2)	1.79107348614303E-08	(1;0;2;0;1;3;0;1;2;0;2)	2.46449402275849E-09	(1;0;2;1;1;3;0;0;2;1;2)	3.04381027647487E-10
(1;0;2;1;0;2;0;1;2;1;2)	1.36384334111663E-09	(1;1;0;0;1;3;0;1;2;0;2)	5.62574946459283E-10	(1;0;2;0;1;0;3;0;2;1;2)	6.89876013782624E-11
(1;1;0;1;0;2;0;1;2;1;2)	5.81952540861338E-10	(1;1;0;1;1;0;3;0;2;0;2)	9.55517195727765E-09	(0;0;2;0;1;3;0;1;2;1;2)	1.21436938701383E-09
(1;0;2;0;0;0;2;1;2;1;2)	2.31058475621981E-11	(1;1;0;1;0;0;3;0;2;1;2)	6.68960683425598E-10	(1;1;0;1;0;0;3;1;2;1;2)	3.32097919920638E-11
(1;1;0;0;0;0;2;1;2;1;2)	9.32381889398484E-11	(1;1;0;1;1;3;0;0;0;1;2)	2.58491196028305E-08	(1;1;0;1;1;3;0;0;2;1;2)	9.17613414515991E-10
(1;1;0;1;0;0;3;1;2;0;2)	5.34764167009022E-10	(1;1;0;0;1;0;3;0;0;1;2)	9.13741452305314E-09	(1;1;0;0;1;0;3;0;2;1;2)	2.61057500299992E-10
(1;1;0;1;1;3;0;0;2;0;2)	2.74328089404838E-08	(0;1;0;1;0;3;0;1;2;1;2)	6.62332701384122E-08	(0;0;2;1;1;0;3;0;2;1;2)	3.00910570254400E-09
(1;1;0;1;0;3;0;0;2;1;2)	3.76006741309523E-08	(0;1;0;0;0;0;3;1;2;1;2)	9.67474908143681E-10	(0;0;0;1;1;3;0;1;2;1;2)	8.58325348033681E-09
(1;1;0;0;1;0;3;0;2;0;2)	6.10854477068761E-09	(0;1;0;0;1;3;0;0;2;1;2)	9.06367209816002E-08	(0;0;0;0;1;0;3;1;2;1;2)	2.10634549291057E-09
(1;1;0;0;0;0;3;0;2;1;2)	2.00300723705075E-09	(0;0;0;1;0;0;3;1;2;1;2)	3.72927347325249E-08	(0;1;0;0;1;3;0;1;2;1;2)	1.50804693510057E-09
(1;1;0;0;1;3;0;0;0;1;2)	8.25222741280878E-08	(0;0;0;1;1;3;0;0;2;1;2)	1.75132827152317E-07	(1;0;2;1;1;0;3;1;2;0;2)	2.95246746828550E-11
(0;1;0;0;1;0;2;0;2;1;2)	2.00633004509247E-07	(0;0;0;0;1;0;3;0;2;1;2)	7.59135253573279E-08	(1;1;0;1;1;0;3;1;2;0;2)	9.64947706008457E-11
(0;1;0;0;0;3;0;1;2;1;2)	2.82935651568628E-07	(0;0;0;1;1;0;3;1;2;0;2)	6.11877114665976E-08	(0;1;0;1;1;0;3;0;2;1;2)	1.61305754741848E-08
(0;0;0;1;0;3;0;1;2;1;2)	1.42953252373212E-07	(0;0;2;1;1;0;3;0;0;1;2)	6.50133755627202E-08	(1;0;2;1;1;0;2;1;2;1;2)	1.38333576216261E-10
(0;0;0;0;0;0;3;1;2;1;2)	1.31796334146400E-08	(0;1;0;1;1;3;0;1;2;0;2)	3.19471971323341E-08	(1;1;0;1;1;0;2;1;2;1;2)	3.86611580960026E-11
(0;0;0;0;1;3;0;0;2;1;2)	7.04635607666241E-07	(0;1;0;0;1;0;3;1;2;0;2)	2.64226555257014E-09	(1;0;2;1;1;0;3;0;2;1;2)	6.20271744248685E-11
(0;0;2;1;1;0;3;0;2;0;2)	9.44967950629129E-08	(0;1;0;1;1;0;3;0;0;1;2)	2.31124033533787E-07	(0;0;2;1;1;3;0;1;2;1;2)	3.89230275553442E-10
(0;0;2;1;0;0;3;0;2;1;2)	8.56895475863568E-09	(1;0;2;1;1;0;2;0;2;1;2)	7.11856428023035E-10	(0;0;2;0;1;0;3;1;2;1;2)	7.98126580254147E-11
(0;0;0;1;1;3;0;1;2;0;2)	1.01786072633126E-07	(1;1;0;1;1;0;2;0;2;1;2)	1.00434026067076E-09	(1;0;2;0;1;3;0;1;2;1;2)	1.82022834754096E-11
(0;0;2;1;1;3;0;0;0;1;2)	1.59833416961973E-07	(0;0;0;1;1;0;2;1;2;1;2)	1.16719700444347E-08	(1;1;0;0;1;3;0;1;2;1;2)	6.48125656232710E-11
(0;0;0;0;1;0;3;1;2;0;2)	5.52495461695015E-08	(0;0;2;1;1;2;0;1;2;1;2)	1.31539415514571E-09	(1;1;0;1;1;0;3;0;2;1;2)	2.37608885502141E-10
(0;0;2;0;1;0;3;0;0;1;2)	5.22628974803823E-08	(1;0;2;0;1;2;0;1;2;1;2)	3.87393858782666E-10	(0;0;0;1;1;0;3;1;2;1;2)	4.55663750017603E-09
(0;1;0;0;1;3;0;1;2;0;2)	8.81150786345243E-08	(1;1;0;0;1;2;0;1;2;1;2)	1.51626714710262E-09	(0;1;0;1;1;3;0;1;2;1;2)	4.97742112050777E-10
(1;0;2;1;1;0;3;1;2;0;0)	4.23301371731106E-09	(0;1;0;1;1;2;0;1;2;1;2)	2.98974308961553E-09	(0;1;0;0;1;0;3;1;2;1;2)	1.29289320257051E-10

(1;1;0;1;1;0;3;1;2;0;0)	1.77993701248875E-08	(0;0;2;0;1;0;2;1;2;1;2)	2.03764855094867E-09	(0;0;2;1;1;0;3;1;2;1;2)	1.17658570897309E-10
(0;1;0;1;1;0;3;0;2;0;2)	2.22292074234596E-07	(1;0;2;1;0;3;0;1;2;1;2)	1.57144761781853E-10	(1;0;2;1;1;3;0;1;2;1;2)	6.28277084439980E-12
(0;1;0;1;0;0;3;0;2;1;2)	2.25523383178816E-08	(1;0;2;0;0;0;3;1;2;1;2)	1.22743990235636E-11	(1;1;0;1;1;3;0;1;2;1;2)	2.28249781602285E-11
(0;1;0;1;1;3;0;0;0;1;2)	4.89256070512733E-07	(1;0;2;0;1;3;0;0;2;1;2)	1.05225306697271E-09	(1;0;2;0;1;0;3;1;2;1;2)	1.35565437467523E-12
(0;1;0;0;1;0;3;0;0;1;2)	2.11078950071222E-07	(1;1;0;1;0;3;0;1;2;1;2)	4.63954465848046E-09	(1;1;0;0;1;0;3;1;2;1;2)	5.00397799769057E-12
(0;0;0;1;1;0;3;0;0;1;2)	1.83804539990962E-06	(1;1;0;0;0;0;3;1;2;1;2)	4.85859398384004E-11	(0;1;0;1;1;0;3;1;2;1;2)	2.81366272607482E-10
(1;0;2;1;1;0;2;1;2;0;2)	9.33491992947508E-10	(1;1;0;0;1;3;0;0;2;1;2)	2.99597885407012E-09	(1;0;2;1;1;0;3;1;2;1;2)	2.01447159222573E-12
(1;1;0;1;1;0;2;1;2;0;2)	4.15232831343925E-10	(0;1;0;0;1;0;2;1;2;1;2)	1.64735318154090E-08	(1;1;0;1;1;0;3;1;2;1;2)	7.46512987928675E-12
(1;0;2;1;1;0;2;0;0;1;2)	3.93993613210528E-09	(0;0;2;1;1;0;2;0;2;1;2)	6.43578013888384E-09	(0;0;2;1;0;0;3;1;2;1;2)	4.76678766406132E-10
(1;1;0;1;1;0;2;0;0;1;2)	2.28587859303450E-08	(0;1;0;1;1;0;2;0;2;1;2)	3.85061160774704E-08	(0;0;2;1;1;3;0;0;2;1;2)	6.56114183403428E-09

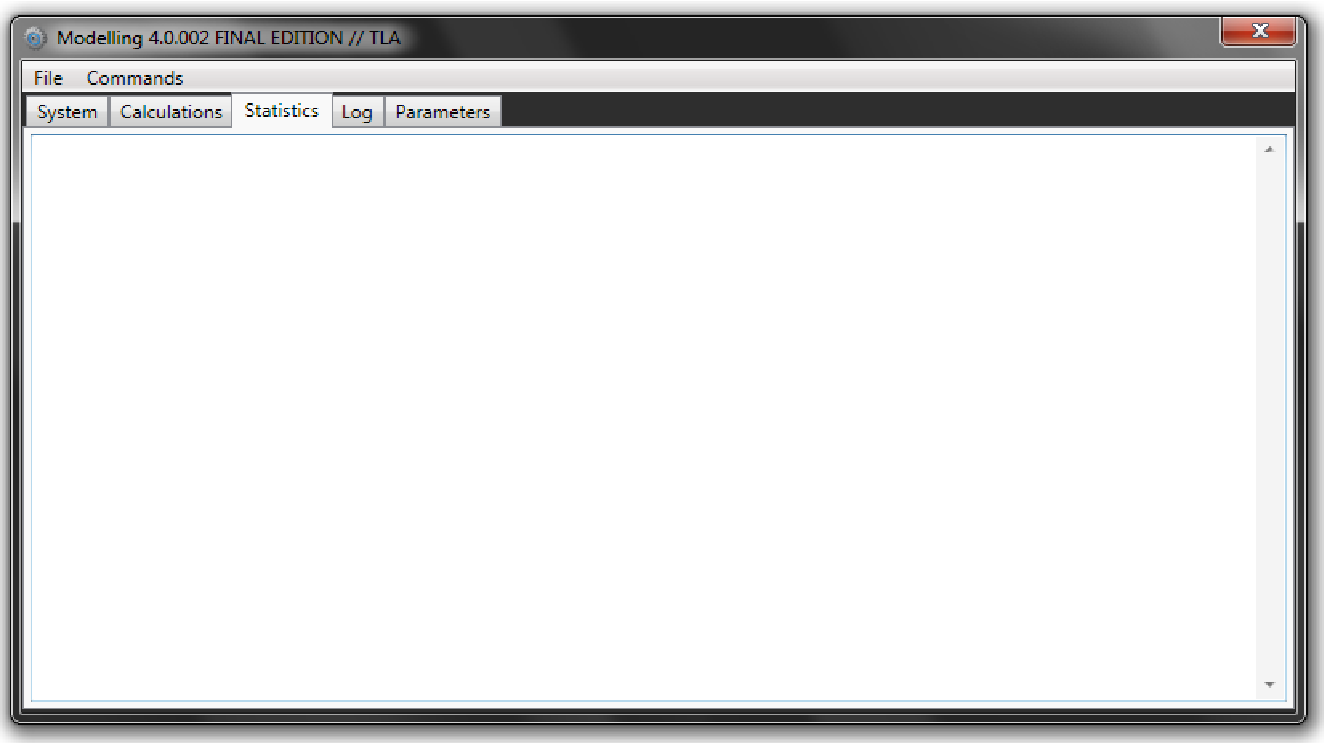
6.2 Aptarnavimo sistemos stochastinės charakteristikos

```
(3) service's  stochastic characteristics:
Probability that application will be serviced instantly: 0.960438967849912
//---
Probability that service will break: 0.333335402715041
//---
Probability that application will be put in medium priority row: 0.978284413632912
//---
Probability that application will not be serviced in medium priority row: 0.0217155863670856
//---
Average medium priority row length: 0.0217155863670856
//--------------------------------------------------------------------------------
(4) extended service's  stochastic characteristics:
Probability that application will be serviced instantly: 0.737292512249025
//---
Probability that application will be put in medium priority row: 0.961708644095299
Probability that application will be put in low priority row: 0.967684163689258
//---
Probability that application will not be serviced in medium priority row: 0.0382913559046989
Probability that application will not be serviced in low priority row: 0.0323158363107401
//---
Average medium priority row length: 0.0382913559046989
Average low priority row length: 0.0323158363107401
//--------------------------------------------------------------------------------
(5) service's  stochastic characteristics:
Probability that application will be serviced instantly: 0.799229542443682
//---
Probability that application will be put in medium priority row: 0.970357852200268
//---
Probability that application will not be serviced in medium priority row: 0.0296421477997315
//---
Average medium priority row length: 0.0296421477997315
//--------------------------------------------------------------------------------
(7) service's  stochastic characteristics:
Probability that application will be serviced instantly: 0.893344280187885
//---
Probability that application will be put in medium priority row: 0.991608153792756
//---
Probability that application will not be serviced in medium priority row: 0.00839184620724279
//---
Average medium priority row length: 0.00839184620724279
//--------------------------------------------------------------------------------
```

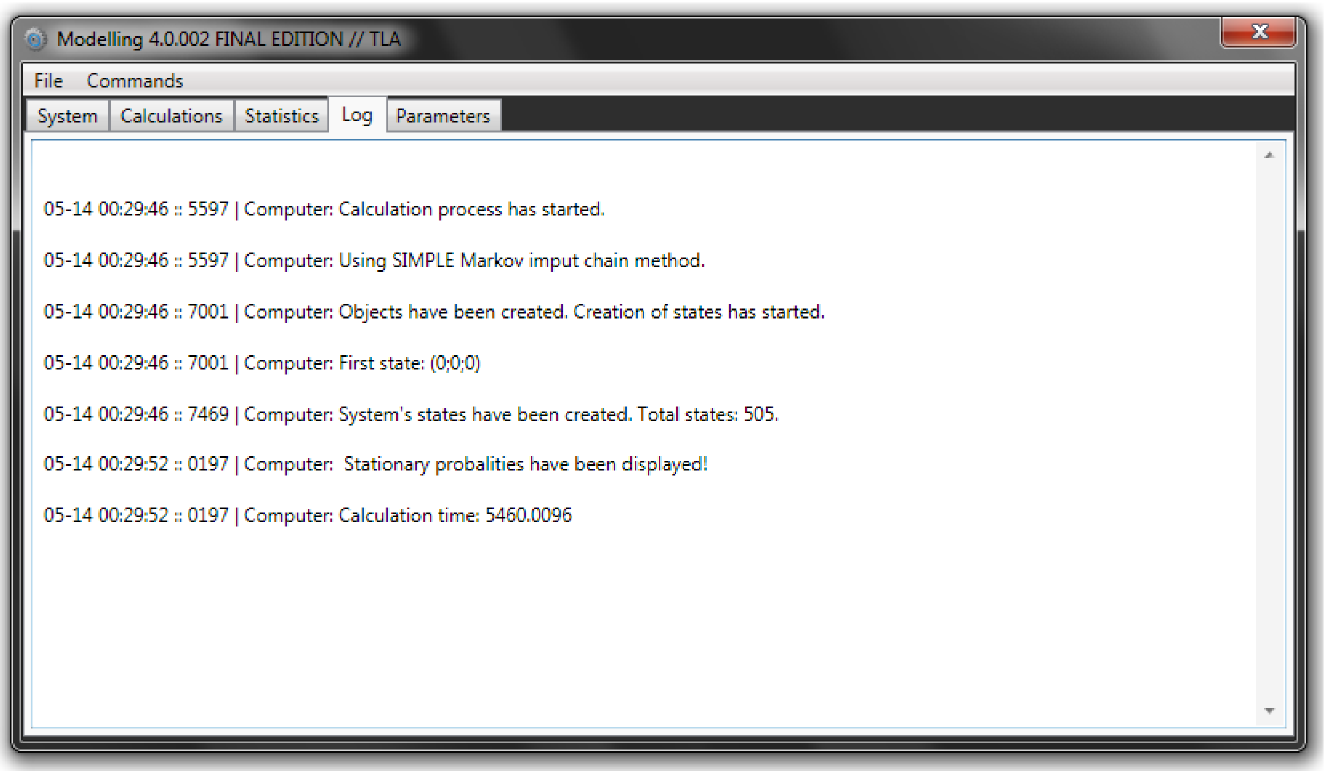
6.3 Programinės įrangos vartotojo sąsajos langai



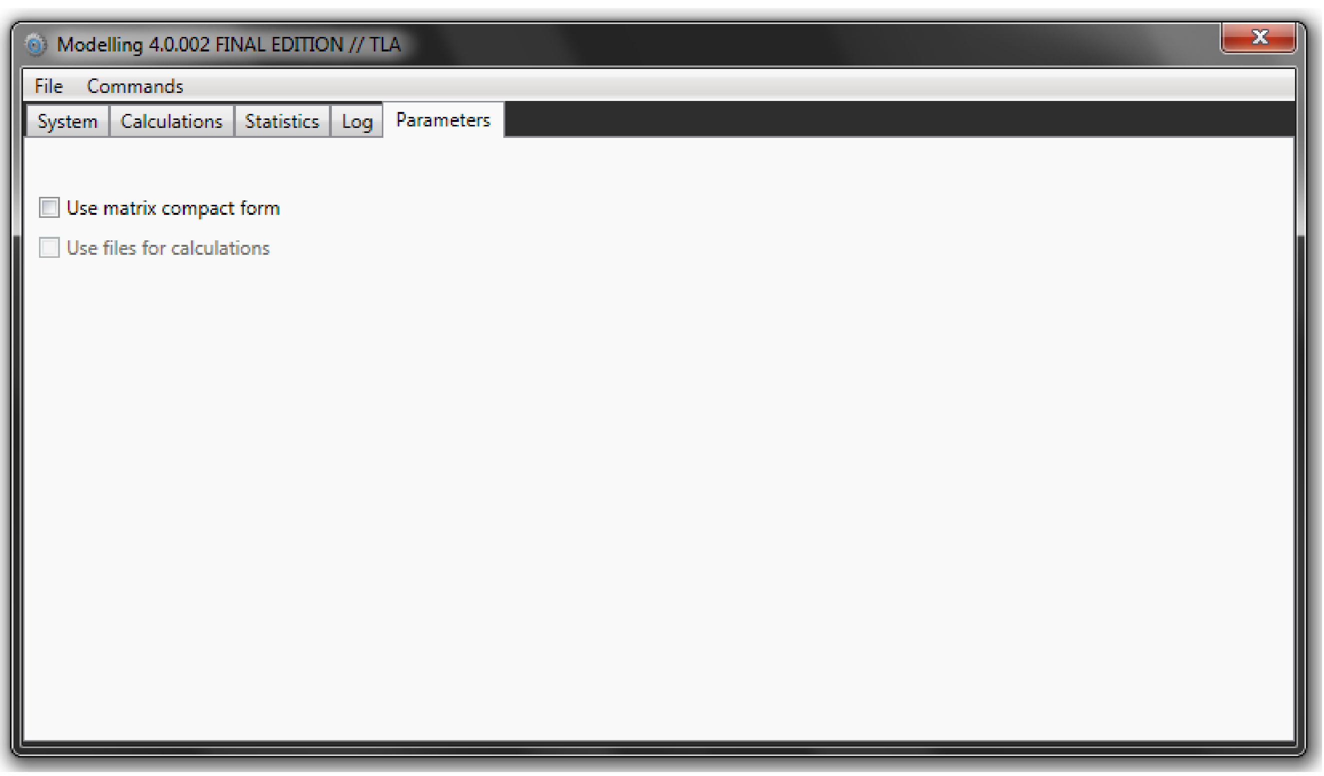
6.1. pav. Programos „Calculations“ polangis



6.2. pav. Programos „Statistics“ polangis



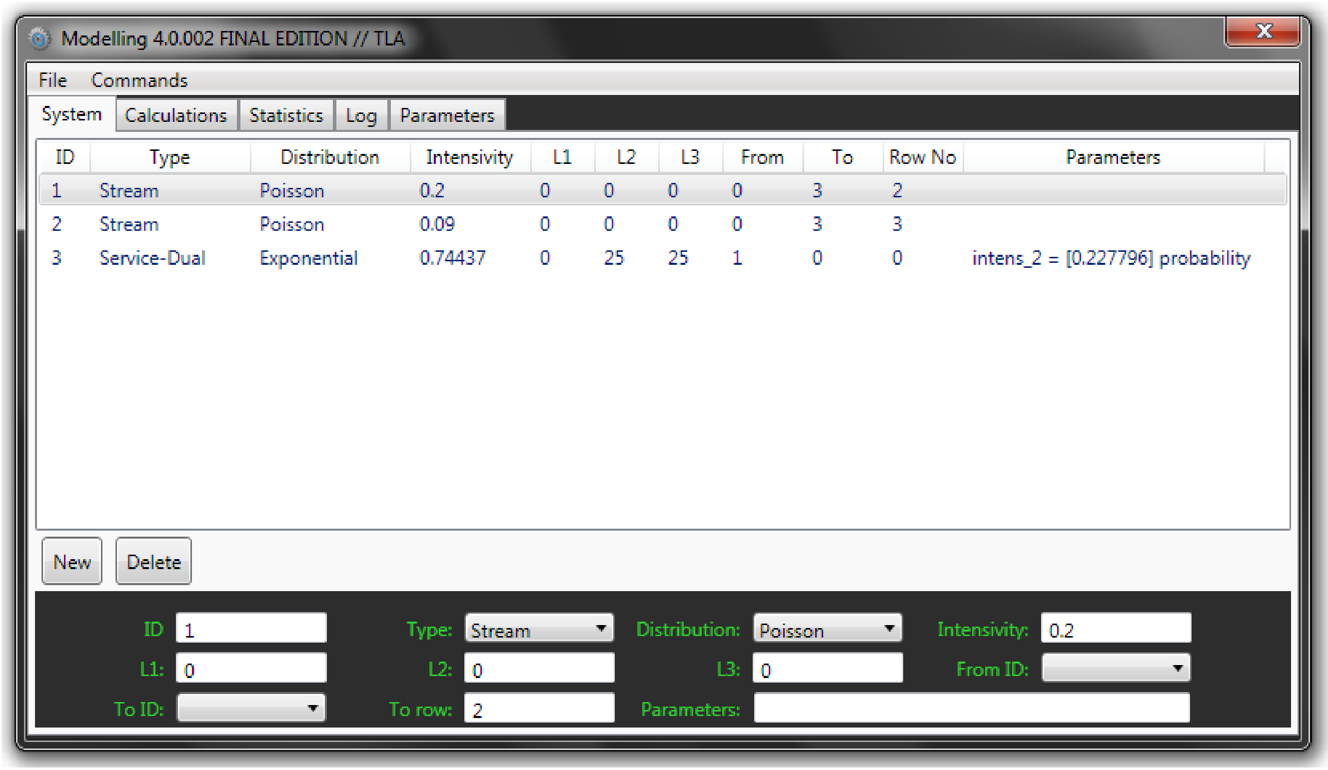
6.3. pav. Programos „Log“ polangis



6.4. pav. Programos „Parameters“ polangis

6.4 Aptarnavimo sistemos schema ir charakteristikos

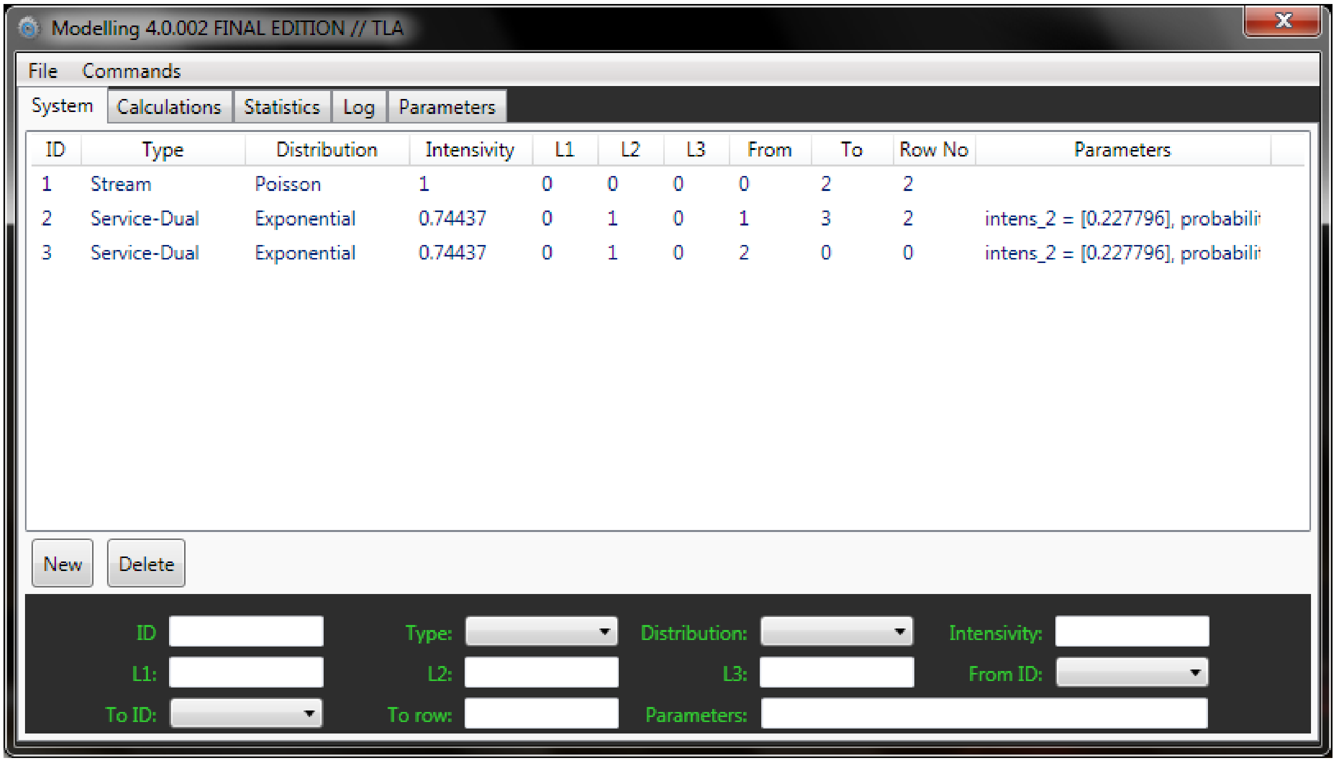
3.2.2. skyriuje pateikto pavyzdžio aptarnavimo sistemos schematinę struktūra programinėje įrangoje pateikta žemiau esančiame paveiksliuke.



3.2.2. skyriuje pateikto pavyzdžio apskaičiuotos stochastinės charakteristikos:

(3) extended service's stochastic characteristics:
Probability that application will be serviced instantly: 0.586409142939731
//---
Probability that application will be put in medium priority row: 0.999999999470633
Probability that application will be put in low priority row: 0.999999999908296
//---
Probability that application will not be serviced in medium priority row: 5.29363994190609E-10
Probability that application will not be serviced in low priority row: 9.17009408892797E-11
//---
Average medium priority row length: 0.185506424629069
Average low priority row length: 0.142354348835089
//---

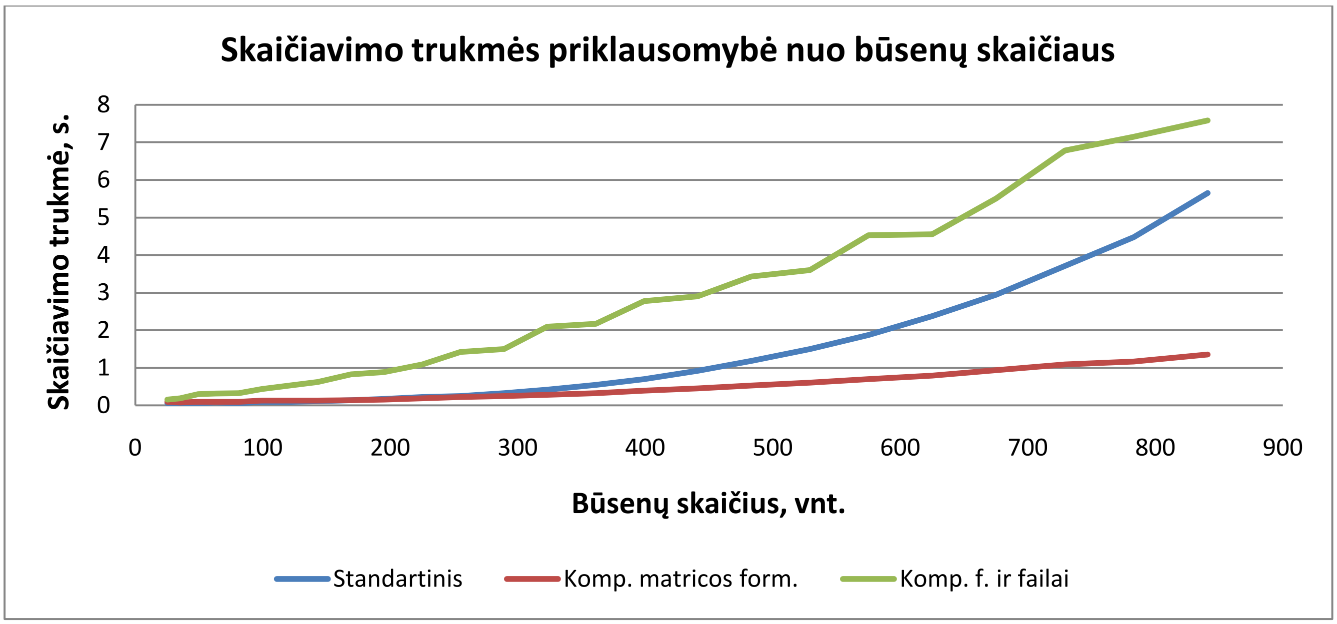
6.5 12 testo modeliavimo rezultatai

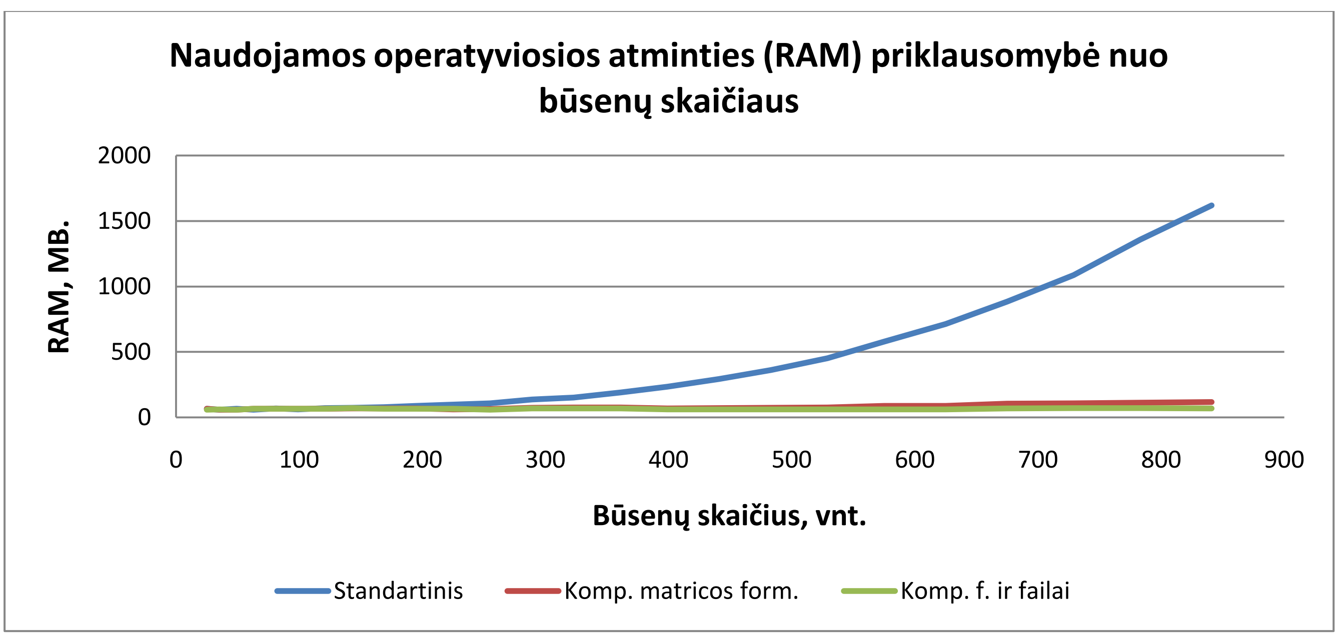


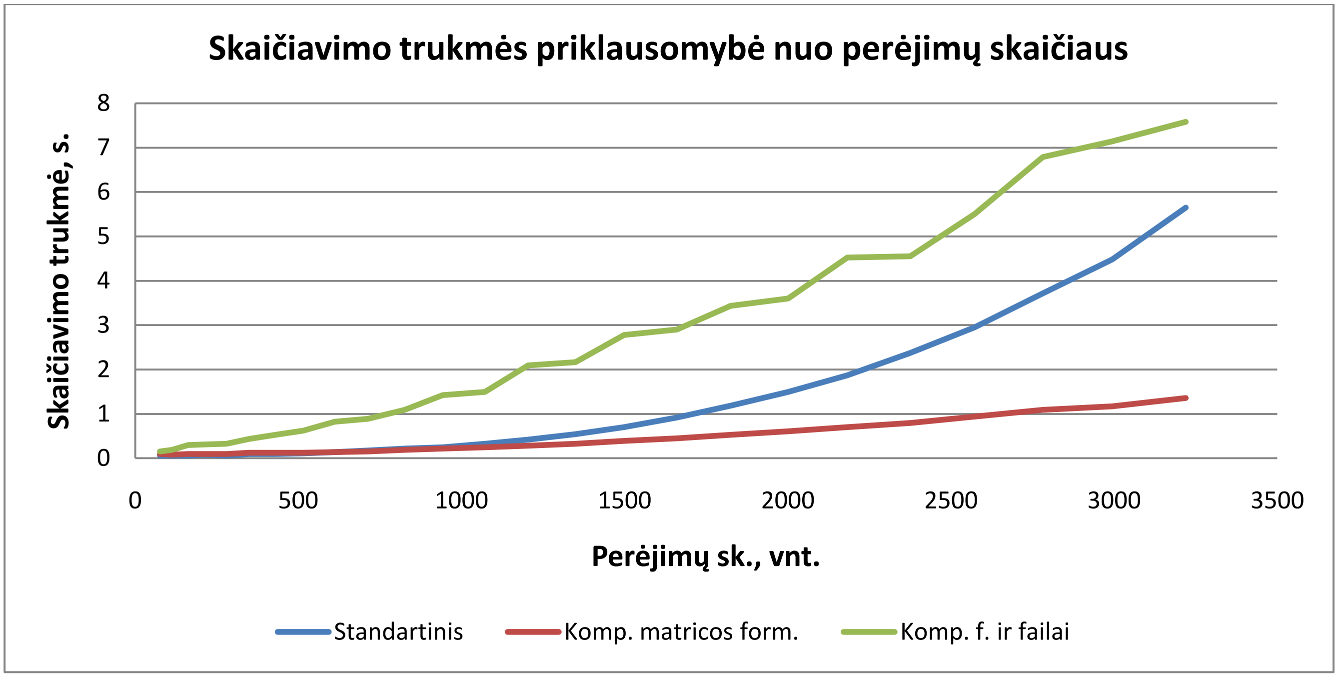
1	2	3	4	5	6	7	8	9	10
Mat. f.	**Failai**	**Schem. nusk., s.**	**Pr. būs., s.**	**Visos būs., s.**	**Lygt. spr., s.**	**Bendra trukmė, s.**	**RAM, MB**	**Būs. sk., vnt.**	**Per. Sk., vnt.**
norm.	-	0.0156	0.0156	0.0156	0.0156	0.0624	67.125	25	75
norm.	-	0.0156	0	0.0312	0	0.0624	57.91016	35	112
norm.	-	0.0156	0.0156	0.0156	0	0.0624	66.94141	49	161
norm.	-	0.0312	0.0156	0.0156	0.0156	0.078	58.94141	63	214
norm.	-	0.0156	0.0156	0.0156	0.0156	0.0624	67.63672	81	279
norm.	-	0.0312	0.0156	0.0156	0.0156	0.0936	60.99609	99	348
norm.	-	0.024001	0.006	0.027002	0.029002	0.096006	71.26953	121	429
norm.	-	0.0312	0.0156	0.0156	0.0468	0.1092	73.96875	143	514
norm.	-	0.0312	0.0156	0.0312	0.0468	0.1404	78.89453	169	611
norm.	-	0.0312	0.0156	0.0312	0.078	0.1716	88.12109	195	712
norm.	-	0.0312	0	0.0468	0.1248	0.2184	98.25391	225	825
norm.	-	0.0312	0.0156	0.0312	0.1716	0.249601	108.3242	255	942
norm.	-	0.0312	0.0156	0.0312	0.234	0.327601	136.9688	289	1071
norm.	-	0.0312	0	0.0468	0.327601	0.421201	150.6836	323	1204
norm.	-	0.0312	0	0.0468	0.452401	0.546001	191.6953	361	1349
norm.	-	0.0312	0	0.0468	0.608401	0.702001	235.3438	399	1498
norm.	-	0.0312	0	0.0624	0.826802	0.920402	292.293	441	1659
norm.	-	0.0312	0	0.0624	1.076402	1.185602	361.1016	483	1824
norm.	-	0.0312	0	0.0624	1.404003	1.497603	451.5391	529	2001

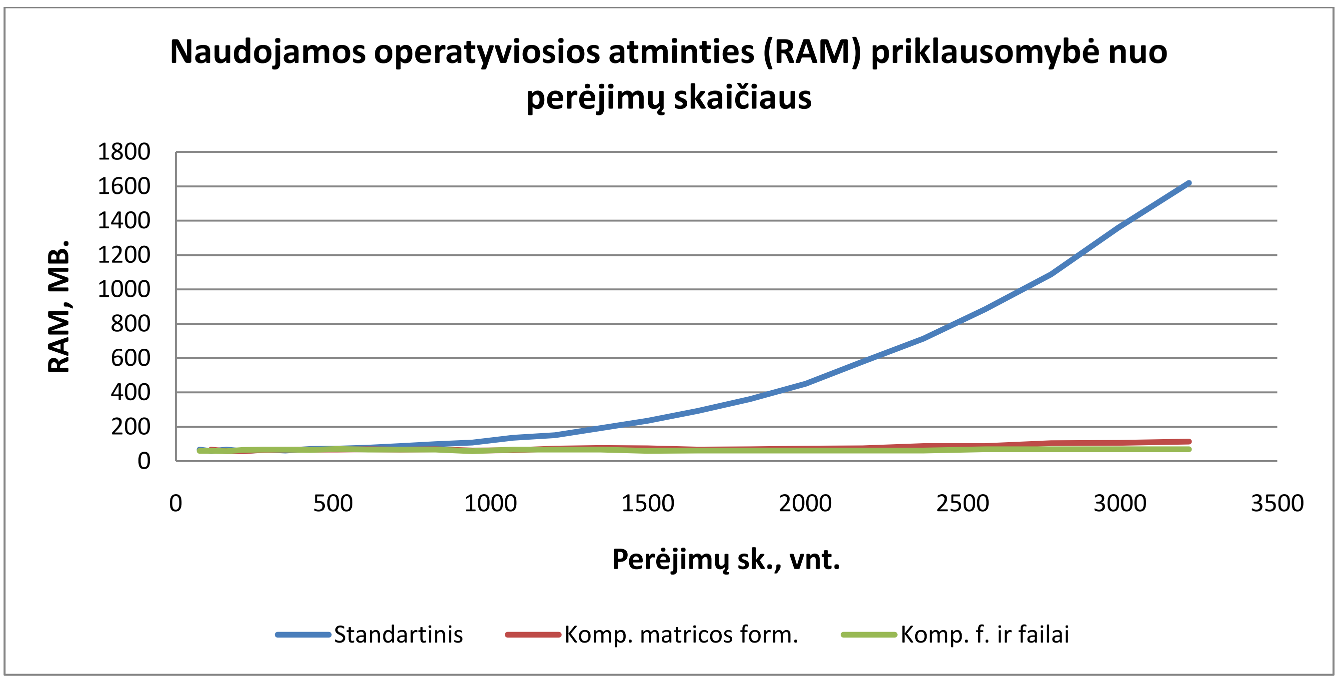
1	2	3	4	5	6	7	8	9	10
norm.	-	0.0312	0	0.0624	1.762803	1.872003	580.3242	575	2182
norm.	-	0.0312	0	0.078	2.246404	2.371204	713.3594	625	2375
norm.	-	0.0156	0.0156	0.0624	2.839205	2.948405	884.2969	675	2572
norm.	-	0.0156	0.0156	0.078	3.588006	3.712807	1088.359	729	2781
norm.	-	0.0312	0	0.0936	4.336808	4.477208	1359.16	783	2994
norm.	-	0.0156	0.0156	0.0936	5.50681	5.64721	1620.598	841	3219
komp.	-	0.0468	0	0.0312	0.0156	0.0936	66.57813	25	75
komp.	-	0.0156	0.0156	0.0156	0.0156	0.078	58.66797	35	112
komp.	-	0.0312	0.0156	0.0156	0.0312	0.0936	58.55469	49	161
komp.	-	0.0312	0	0.0312	0.0156	0.0936	66.52734	63	214
komp.	-	0.0312	0	0.0312	0.0156	0.0936	66.67188	81	279
komp.	-	0.0468	0	0.0312	0.0312	0.1248	66.86328	99	348
komp.	-	0.0468	0	0.0312	0.0312	0.1248	67.03906	121	429
komp.	-	0.0156	0.0156	0.0312	0.0468	0.1248	68.60938	143	514
komp.	-	0.0312	0	0.0312	0.0624	0.1404	67.77734	169	611
komp.	-	0.0156	0.0156	0.0312	0.078	0.156	69.29688	195	712
komp.	-	0.0312	0	0.0468	0.1092	0.1872	62.06641	225	825
komp.	-	0.0312	0	0.0312	0.1404	0.2184	62.86328	255	942
komp.	-	0.0312	0.0156	0.0312	0.1716	0.2496	72.86719	289	1071
komp.	-	0.0156	0.0156	0.0312	0.2184	0.280801	75.99609	323	1204
komp.	-	0.0156	0.0312	0.0312	0.2496	0.327601	75.44531	361	1349
komp.	-	0.0156	0.0156	0.0468	0.296401	0.390001	68.24219	399	1498
komp.	-	0.0312	0.0156	0.0468	0.343201	0.452401	69.88672	441	1659
komp.	-	0.0312	0	0.0624	0.421201	0.530401	73.52734	483	1824
komp.	-	0.0312	0	0.078	0.499201	0.608401	75.69141	529	2001
komp.	-	0.0312	0.0156	0.0624	0.592801	0.702001	87.46484	575	2182
komp.	-	0.0156	0.0156	0.0624	0.686401	0.795601	87.94531	625	2375
komp.	-	0.0312	0	0.078	0.811202	0.936002	104.4453	675	2572
komp.	-	0.0312	0.0156	0.078	0.951602	1.092002	107.043	729	2781
komp.	-	0.0156	0.0156	0.0936	1.029602	1.170002	113.5938	783	2994
komp.	-	0.0312	0.0156	0.0936	1.201202	1.357202	117.6055	841	3219
komp.	+	0.0312	0	0.0156	0.0936	0.156	59.80078	25	75
komp.	+	0.0312	0	0.0156	0.1404	0.1872	60.46094	35	112
komp.	+	0.0312	0.0156	0.0156	0.2184	0.296401	58.78516	49	161
komp.	+	0.0312	0	0.0312	0.234	0.312001	66.51172	63	214
komp.	+	0.0312	0.0156	0.0156	0.249601	0.327601	66.74609	81	279
komp.	+	0.0312	0	0.0312	0.358801	0.436801	67.01953	99	348
komp.	+	0.0156	0.0312	0.0156	0.452401	0.530401	66.375	121	429
komp.	+	0.0156	0.0156	0.0312	0.546001	0.624001	70.76953	143	514
komp.	+	0.0468	0	0.0312	0.748801	0.826801	66.79297	169	611
komp.	+	0.0312	0.0156	0.0312	0.795601	0.889202	66.74609	195	712
komp.	+	0.0468	0	0.0312	0.998402	1.092002	66.60547	225	825
komp.	+	0.0312	0	0.0468	1.326002	1.419603	59.04297	255	942
komp.	+	0.0312	0	0.0468	1.388402	1.497603	67.79688	289	1071
komp.	+	0.0312	0.0156	0.0312	1.965604	2.090404	68.05469	323	1204

1	2	3	4	5	6	7	8	9	10
komp.	+	0.0312	0	0.0468	2.090404	2.168404	68.35156	361	1349
komp.	+	0.0312	0	0.0468	2.667605	2.776805	60.95313	399	1498
komp.	+	0.0156	0.0156	0.0468	2.792405	2.901605	61.87891	441	1659
komp.	+	0.0312	0	0.0624	3.322806	3.432006	62.125	483	1824
komp.	+	0.0312	0	0.0624	3.494406	3.603606	61.51563	529	2001
komp.	+	0.0156	0.0156	0.0624	4.383608	4.524008	62.26172	575	2182
komp.	+	0.0312	0	0.0936	4.414808	4.555208	61.84375	625	2375
komp.	+	0.0312	0	0.078	5.366409	5.50681	69.70313	675	2572
komp.	+	0.0468	0	0.078	6.645612	6.786012	69.92188	729	2781
komp.	+	0.0312	0	0.0936	7.004412	7.144813	70.01563	783	2994
komp.	+	0.0312	0	0.1092	7.425613	7.581613	69.63281	841	3219

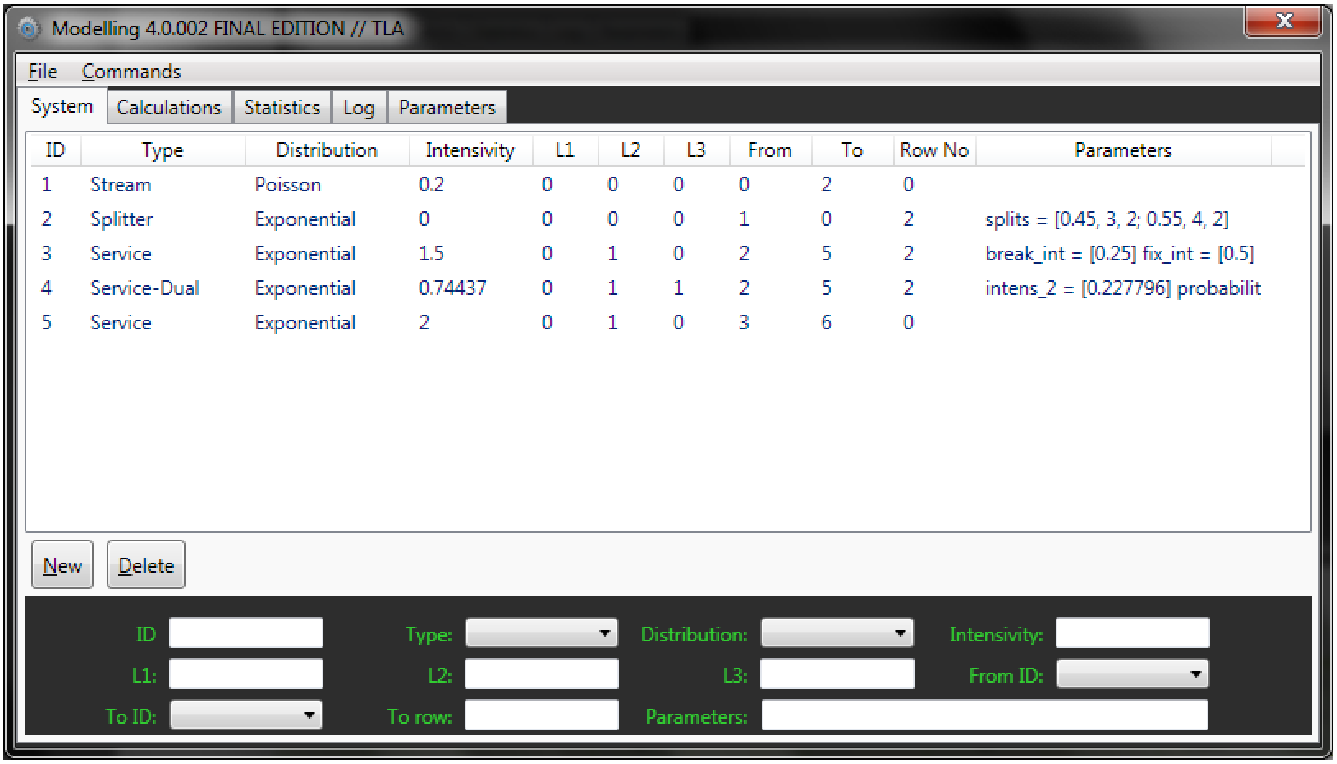





Skaičiavimo trukmės priklausomybė nuo perėjimų skaičiaus
Skaičiavimo trukmė, s.
0
1
2
3
4
5
6
7
8
0
500
1000
1500
2000
2500
3000
3500
Perėjimų sk., vnt.
Standartinis
Komp. matricos form.
Komp. f. ir failai

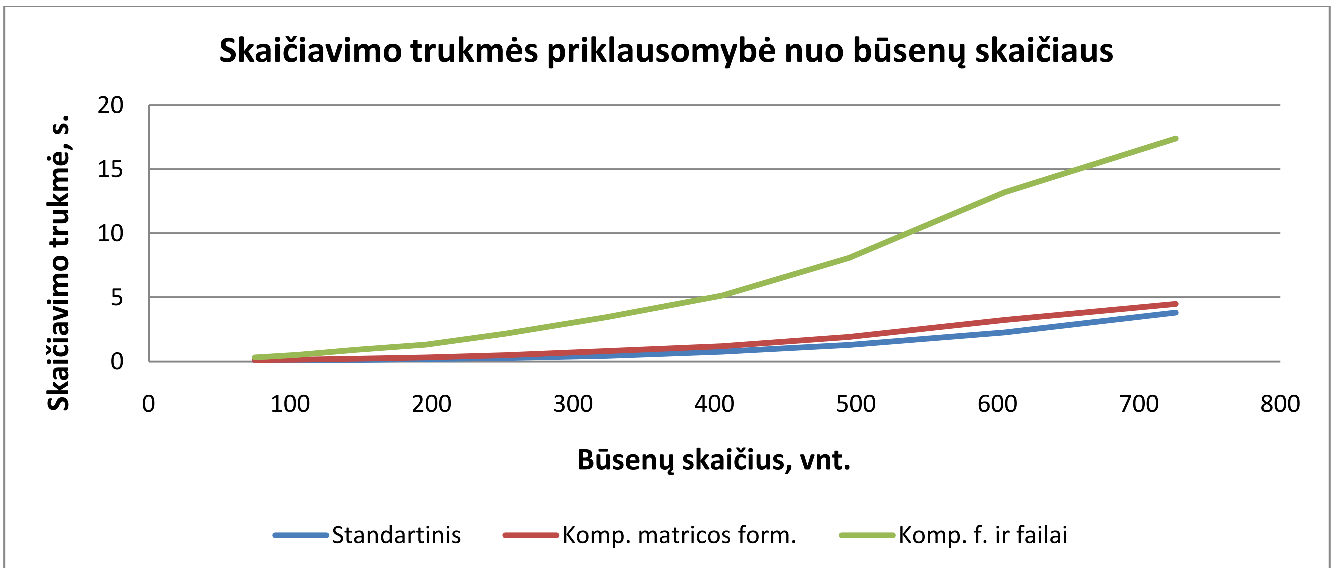

Naudojamos operatyviosios atminties (RAM) priklausomybė nuo perėjimų skaičiaus
RAM, MB.
0
200
400
600
800
1000
1200
1400
1600
1800
0
500
1000
1500
2000
2500
3000
3500
Perėjimų sk., vnt.
Standartinis
Komp. matricos form.
Komp. f. ir failai

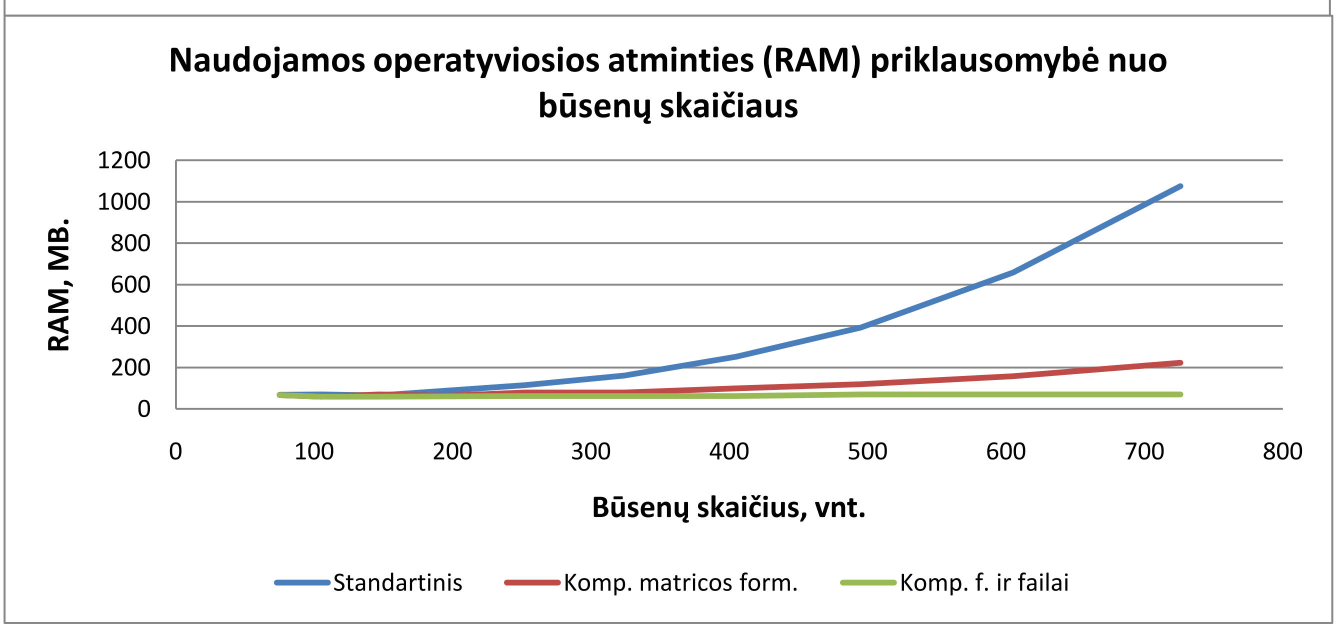
6.6 11 testo modeliavimo rezultatai

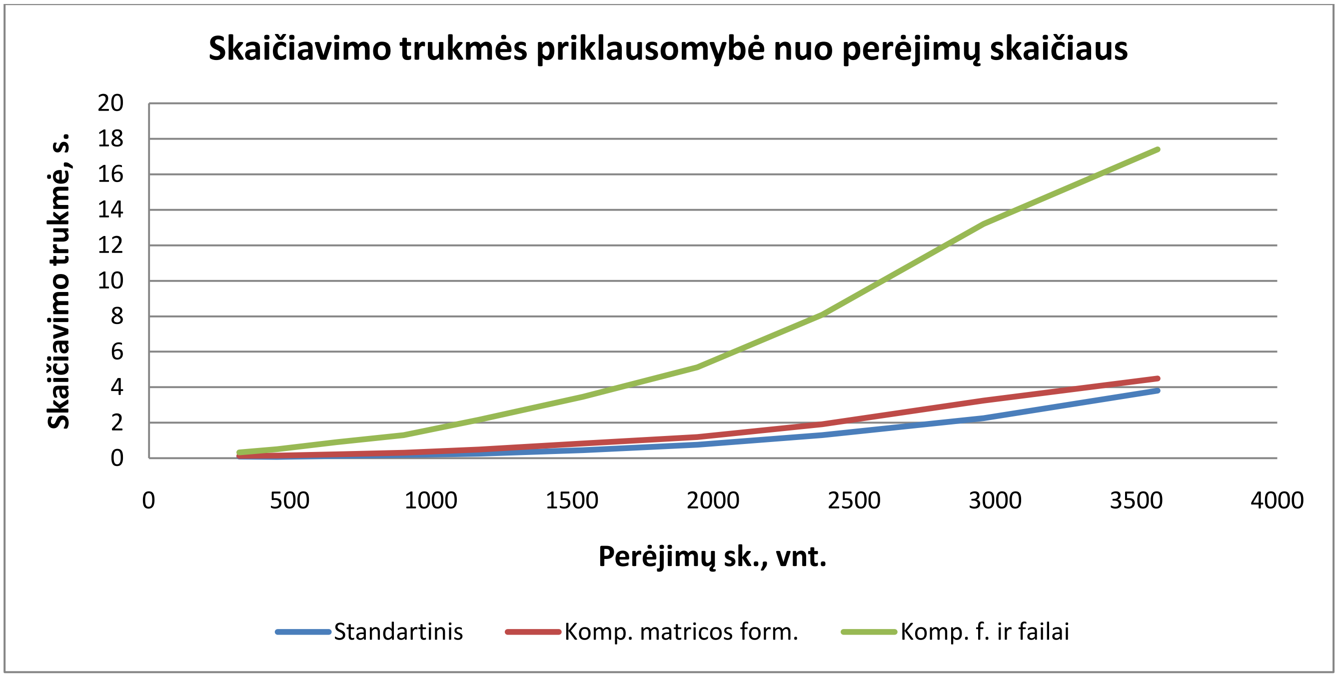


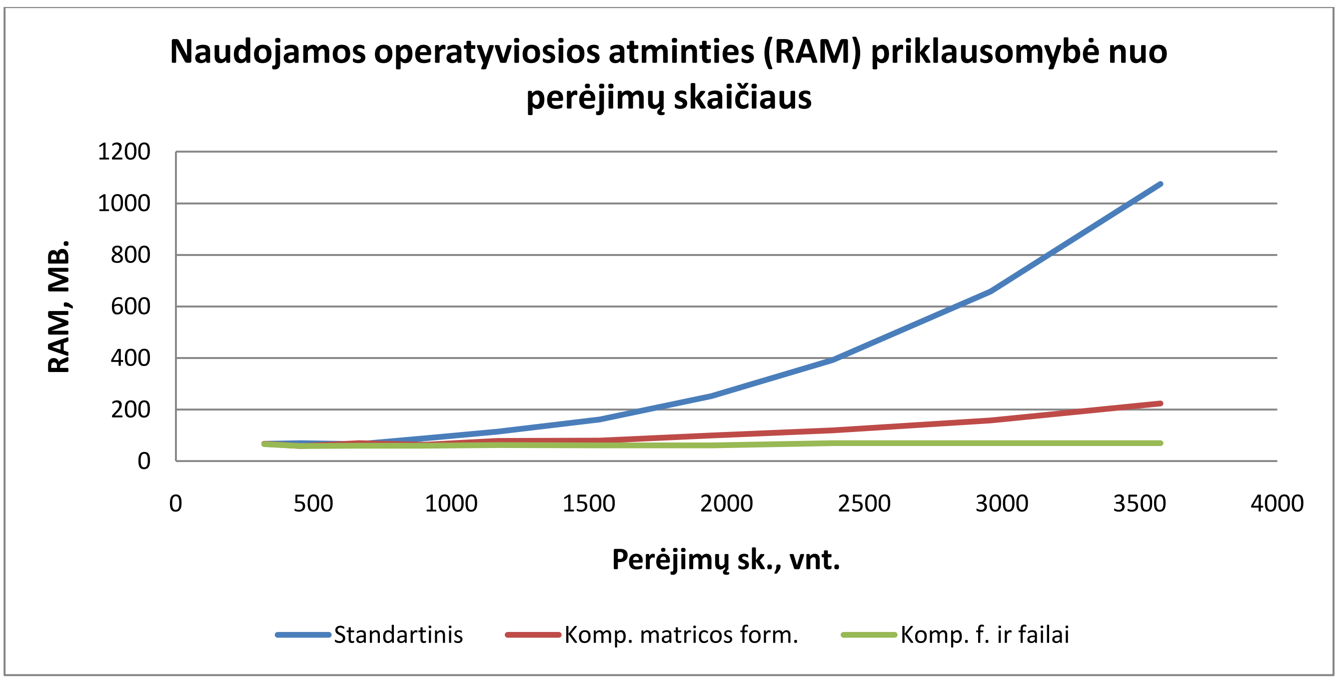
1	2	3	4	5	6	7	8	9	10
Mat. f.	**Failai**	**Schem. nusk., s.**	**Pr. būs., s.**	**Visos būs., s.**	**Lygt. spr., s.**	**Bendra trukmė, s.**	**RAM, MB**	**Būs. sk., vnt.**	**Per. Sk., vnt.**
norm.	-	0.0312	0.0156	0.0312	0.0156	0.0936	67.66797	75	320
norm.	-	0.0156	0.0156	0.0156	0.0156	0.078	69.92969	105	454
norm.	-	0.0312	0	0.0312	0.0468	0.1248	65.91797	147	665
norm.	-	0.0468	0	0.0312	0.0936	0.1872	88.19141	196	903
norm.	-	0.0312	0.0156	0.0312	0.1716	0.2652	114.1406	252	1169
norm.	-	0.0312	0	0.0468	0.358801	0.452401	161.2773	324	1539
norm.	-	0.0312	0.0156	0.0624	0.639601	0.764401	251.8672	405	1944
norm.	-	0.0312	0	0.0936	1.154402	1.294802	392.0508	495	2386
norm.	-	0.0312	0.0156	0.0936	2.106004	2.262004	658.2852	605	2959
norm.	-	0.0312	0.0156	0.1248	3.619206	3.806407	1075.387	726	3575
komp.	-	0.0468	0	0.0312	0.0312	0.1248	66.83203	75	320
komp.	-	0.0312	0	0.0312	0.0624	0.1404	59.26563	105	454
komp.	-	0.0312	0.0156	0.0312	0.1248	0.2184	69.3125	147	665
komp.	-	0.0468	0	0.0468	0.2184	0.312001	64.10938	196	903
komp.	-	0.0312	0.0156	0.0468	0.374401	0.483601	78.41016	252	1169
komp.	-	0.0156	0.0156	0.0624	0.717601	0.826801	79.71094	324	1539
komp.	-	0.0312	0	0.0624	1.076402	1.185602	99.51563	405	1944

1	2	3	4	5	6	7	8	9	10
komp.	-	0.0312	0.0156	0.0624	1.794003	1.918803	119.2422	495	2386
komp.	-	0.0312	0.0156	0.0936	3.104406	3.244806	157.3555	605	2959
komp.	-	0.0312	0.0156	0.1248	4.305608	4.492808	223.2266	726	3575
komp.	+	0.0312	0	0.0312	0.265201	0.327601	66.43359	75	320
komp.	+	0.0312	0	0.0312	0.436801	0.514801	58.55078	105	454
komp.	+	0.0468	0	0.0312	0.811202	0.904802	59.27734	147	665
komp.	+	0.0156	0.0156	0.0312	1.232402	1.310402	60.38281	196	903
komp.	+	0.0312	0.0156	0.0312	2.074804	2.168404	61.80859	252	1169
komp.	+	0.0312	0.0156	0.0624	3.338406	3.463206	61.65234	324	1539
komp.	+	0.0312	0.0156	0.0624	5.007609	5.132409	61.69922	405	1944
komp.	+	0.0156	0.0156	0.078	7.956014	8.080814	70.27344	495	2386
komp.	+	0.0312	0.0156	0.0936	12.94802	13.19762	70.35547	605	2959
komp.	+	0.0312	0	0.1248	17.22243	17.39403	70.40234	726	3575






Skaičiavimo trukmės priklausomybė nuo perėjimų skaičiaus
Skaičiavimo trukmė, s.
20
18
16
14
12
10
8
6
4
2
0
0
500
1000
1500
2000
2500
3000
3500
4000
Perėjimų sk., vnt.
Standartinis
Komp. matricos form.
Komp. f. ir failai


Naudojamos operatyviosios atminties (RAM) priklausomybė nuo perėjimų skaičiaus
RAM, MB.
1200
1000
800
600
400
200
0
0
500
1000
1500
2000
2500
3000
3500
4000
Perėjimų sk., vnt.
Standartinis
Komp. matricos form.
Komp. f. ir failai

6.7 Kompaktinis diskas su programine įranga