Г.В. ОВЕЧКИН, П.В. ОВЕЧКИН

КОМПЬЮТЕРЛІК МОДЕЛЬДЕУ

«СТАНКИН Мәскеу мемлекеттік технологиялық университеті» жоғары кәсіптік білім беретін федералды мемлекеттік бюджеттік білім беру мекемесі «Байланыс желілері және коммутация жүйелері», «Тхнологиялық процестерді және өндірістерді автоматтандыру», «Компьютер желісі» мамандықтарында оқитын орта кәсіптік білім беру мекемелерінің студенттеріне арналған оқулық ретінде ұсынған.

Пікірдің тіркеу нөмірі 435, 20 қазан 2014 ж. «БДФИ» ФММ

ACADEMA

Мәскеу
«Академия» баспа орталығы
2015

ӘОЖ 004.94(075.32)
КБЖ 32.972ші723
О-314

Бұл кітап Қазақстан Республикасының Білім және ғылым министрлігі және «Кәсіпқор» холдингі» КЕАҚ арасында жасалған шартқа сәйкес ««ТжКБ жүйесі үшін шетел әдебиетін сатып алуды және аударуды ұйымдастыру жөніндегі қызметтер» мемлекеттік тапсырмасын орындау аясында қазақ тіліне аударылды.
Аталған кітаптың орыс тіліндегі нұсқасы Ресей Федерациясының білім беру үдерісіне қойылатын талаптардың ескерілуімен жасалды.
Қазақстан Республикасының техникалық және кәсіптік білім беру жүйесіндегі білім беру ұйымдарының осы жағдайды ескеруі және оқу үдерісінде мазмұнды бөлімді (технология, материалдар және қажетті ақпарат) қолдануы қажет.
Аударманы «Delta Consulting Group» ЖШС жүзеге асырды, заңды мекенжайы: Астана қ., Иманов көш., 19,
«Алма-Ата» БО, 809С , телефоны: 8 (7172) 78 79 29, эл. поштасы: info@dcg.kz

Пікір беруші –
А. А. Соломашкин, МҒМ МемҒЗТИ Республикалық ауылшаруашылық академияның аға ғылыми қызметкері

Овечкин Г.В.

О-314 Компьютерлік модельдеу: Орташа кәсіби білім беретін мекеменің студ. арналған оқулық / Г. В. Овечкин, П. В. Овечкин. – М. : «Академия» баспа орталығы, 2015. – 224 с.
ISBN 978-601-333-233-8 (каз.)
ISBN 978-5-4468-1492-3 (рус.)

Оқулық 230111 «Компьютерлік желілер» (ОП.11), 230401 «Ақпараттық жүйелер (салалар бойынша)» (ОП.09), 220703 «Технологиялық процестер мен өндірісті (салалар бойынша) автоматтандыру» (ЕН.02) мамандықтары бойынша (барлығында тереңдетілген дайындық үшін) орташа кәсіби білім берудің Федералды мемлекеттік білім беру стандарттарына сай жасалынған.

Компьютерлік модельдеуді құру теориялары мен парактикасы қарастырылған. Белгіленген тарату заңы бар кездейсоқ мән генераторлары егжей-тегжейлі сипатталған, олардың сапасын тексеру ұсыныстары мен әдістемелері келтірілген. Нақты мысалдармен статистикалық сынау әдісінің көмегімен модельдеудің негізгі тәсілдері көрсетілген. Жаппай қызмет көрсету жүйесін модельдеу, модельдеудің Pilgrim құралдық зат көмегімен модель құру мәселелері егжей-тегжейлі сипатталған. Машиналық тәжірибені стратегиялық және тактикалық жоспарлау негізі қарастырылған.

Орташа кәсіби білім беру мекемелерінің студенттеріне арналған.

ӘОЖ 004.94(075.32)
КБЖ 32.972ші723

ISBN 978-601-333-233-8 (каз)
ISBN 978-5-4468-1492-3 (рус.)

Құрметті оқырман

Осы оқулық 230111 «Компьютерлік желілер» (ОП.11), 230401 «Ақпараттық жүйелер (салалар бойынша)» (ОП.09), 220703 «Технологиялық үрдістер мен өндірісті (салалар бойынша) автоматтандыру» (ЕН.02) мамандықтары үшін (барлығында тереңдетілген дайындық үшін) оқу-әдістемелік жиынтықтың бөлігі болып табылады.

Оқулық 230111, 230401 мамандықтары үшін «Компьютерлік модельдеудің» жалпы кәсіби пәнін және 220703 мамандығының сол пәннің жалпы табиғи-ғылыми циклына арналған.

Жаңа буынның оқу-әдістемелік жиынтығы жалпы білім беру және жалпы кәсіби мамандықтар мен кәсіби модульдерді меңгеруге мүмкіндік беретін дәстүрлі және инновациялық оқу материалдарынан тұрады. Әр жиынтық жұмыс берушінің талабын ескере отырып, жалпы және кәсіби құзыретті игеру үшін қажетті білім беру және бақылау құралдарынан, оқулықтар мен оқу құралдарынан тұрады.

Оқу баспалары электронды білім беру қорларымен толығып отырады. Электронды қорларда интерактивті жаттығулар мен тапсырмалар, мультимедиялық нысандар, ғаламтордағы қосымша материалдар мен қорларға сілтемелері бар теориялық және тәжірибелік модульдерден тұрады. Оларға оқу процесінің негізгі көрсеткіштері: жұмыс уақыты, бақылау және практикалық тапсырмаларды орындау нәтижелері белгіленетін терминологиялық сөздік пен электронды журнал қосылған. Электронды қор оқу процесіне оңай енгізіледі және әр түрлі оқу бағдарламаларына бейімдеуге болады.

Кіріспе

Модельдеу адам қызметінің барлық салаларында дерлік қолданылады. Оны қандай да бір нысанның жұмыс істеу қағидасын түсінуге, оны басқаруды үйренуге, ол жаңа жағдайларда қалай жұмыс істейтіндігін түсінуге, нысанның жұмысын оңтайландыруға және т.с.с. қолдануға болады. Сонымен қатар модельдеу бірқатар себептер (қымбат, денсаулық үшін қауіпті, өте үлкен немесе кіші нысандар, жылдам немесе баяу процестер) бойынша шынайы тәжірибе қою мүмкін болмайтын нысандарға зерттеу жүргізуге мүмкіндік береді. Сонымен бірге, модельдеу тіпті жоқ нысандарды зерттеуге мүмкіндік береді. Мысалы, модельдеу жаңа ұшу аппараттарын құрастыру кезінде кең қолданылады.

Модельдеудің барлық түрлерінің ішінен біз айтқан міндеттерді шешу үшін пайдаланылатын ең әйгілі тәсіл болып табылатын имитациялық модельдеу қызықтыратын болады. Имитациялық модельдеу кезінде жүйені зерттеу үшін зерттелетін жүйенің қызмет ету динамикасының үлгілі уақытында қайта жасайтын бағдарламалар пайдаланылады. Және осындай бағдарламаны жазу жасап шығарушыдан белгілі бір білім мен ептілікті талап етеді.

Оқырманға ұсынылып отырған оқулық 230111, 230401 және 220703 мамандығының студенттеріне «Компьютерлік модельдеу» пәнін меңгеруге арналған, орташа кәсіби білім берудің оқу стандарты осы пәнді оқу нәтижесінде оқушы:

- *білу:* математикалық модельдерді құрудың негізгі қағидалары; күрделі жүйелерді сипаттау кезінде және шешім қабылдау кезінде қолданылатын математикалық модельдеудің негізгі түрлері; модельдер, жүйелер, міндеттер мен әдістерді жіктеу; компьютерлік тәжірибе жүргізу тәсілдері; әр түрлі типті математикалық модельдерді зерттеу әдістері;
- *істеу:* практикалық міндеттерді шешу үшін математикалық әдістер

мен есептеу алгоритмдерін қолдану; математикалық модельдерді құру мен зерттеуге арналған құралдық амалдарды пайдалану;

- *игеру:* математикалық модельдеу дағдыларын игеру керектігін қарастырады.

Пән стандартының көрсетілген талаптарына байланысты оқулықтың келесі құрылымы ұсынылады.

Бірінші тарауда математикалық модельдеудің негізгі түсінігі сипатталып, математикалық модельдеудің жіктелуі ұсынылған, компьютерлік және имитациялық модельдеудің негізі қарастырылған, жүйенің компьютерлік модельдеуі кезінде орындалатын типтік сатылары аталған.

Екінші тарау компьютерлік модельдеу кезінде пайдаланылатын жалған, кездейсоқ санның базалық генераторын құрумен байланысты мәселелерді қарастыруға арналған. Негізгі назар модельдеу кезінде физикалық және кестелік генератормен салыстырғанда бірқатар артықшылықтары бар арифметикалық генераторға аударылған.

Үшінші тарау кездейсоқ сан генераторының сапасын тексеру кезінде қолданылатын әдістерді сипаттаудан тұрады. Біртектілікті тексеруде қалыптасқан сан тәуелсіздігінің критериі қарастырылып, қалыптасатын жалған кездейсоқ сандардың біртектілігі мен тәуелсіздігін тексеру үшін графикалық та, статистикалық та тесттер сипатталған.

Төртінші тарауда белгіленген тарату заңы бар үздіксіз кездейсоқ мән генераторының негізгі әдістері сипатталған. Негізгі назар әмбебап генерация әдістеріне (сонымен бірге дәл және жуық әдістер қарастырылған) де, тарату заңдылығы ең жиі қолданылатын кездейсоқ мән генерациясының жеке әдістеріне де аударылған, сонымен қатар бірлеспеген оқиғалардың мен күрделі тәуелсіз және тәуелді оқиғалардың толық тобын модельдеу мәселелері сипатталған.

Бесінші тарауда туынды және жеке таралу заңдары бар дискретті кездейсоқ мәнге арналған генераторларды құру әдістері қарастырылған. Қарапайым және күрделі оқиға модельдеуіне де назар аударылған.

Алтыншы тарауда статистикалық сынау әдісі (Монте-Карло әдісі) көмегімен жүйені модельдеу талқыланады, сонымен қатар оны пайдаланудың бірқатар мысалдары келтірілген. Бір өлшемді және екі өлшемді кездейсоқ кезудің классикалық модельдеуі қарастырылған.

Жетінші тарау жаппай қызмет көрсету жүйелерін модельдеуге арналған. Онда осындай жүйелерді жіктеу ұсынылған, олардың негізгі сипаттамалары қарастырылған және де аналитикалық есептеу, компьютерлік модельдеу көмегімен сипаттамалар алу әдісі берілген.

Сегізінші тарауда компьютерлік тәжірибені жобалау мәселелері қарастырылған. Белгіленген дәлділік пен модельдеу нәтижелерінің нақтылығын қамтамасыз ету мәселелері шешілетін тәжірибені тактикалық жоспарлауға негізгі назар аударылған.

Тоғызыншы тарауда Pilgrim имитациялық үлгілеудің құралдық тәсілі көмегімен жаппай қызмет көрсету жүйелерін модельдеудің негізі ұсынылған. Жүйенің негізгі нысандары қарастырылып, оларды пайдалану мысалдары берілген.

Оныншы тарау «Компьютерлік модельдеу» пәнін оқитын студенттердің орындауына ұсынылған жеке тапсырма нұсқаларымен жабдықталған тоғыз лабораториялық жұмыстан тұрады.

Материалды игеруді жеңілдету үшін оқулықта қарастырылған әдістер мен алгоритмдердің көп бөлігі C# заманауи бағдарламалау тілінде жүзеге асырылатын мысалдармен қамтылғанын айта кеткен жөн. Сонымен қатар, әр тараудың соңында материалды меңгеру деңгейін бағалау үшін пайдалануға болатын бақылау сұрақтарының тізімі ұсынылған.

1 Тарау

ЖАЛПЫ ТҮСІНІК КОМПЬЮТЕРЛІК МОДЕЛЬДЕУ

1.1. МОДЕЛЬ МЕН МОДЕЛЬДЕУ ТҮСІНІГІ

Адамзат алдында үнемі оны қоршаған әлемдегі нысандар мен жүйелерді пайдалану қажеттілігі туындайды. Сонымен қатар әр түрлі мақсаттар болуы мүмкін: жүйенің жұмыс істеу қағидаларын жақсырақ түсіну, оларды жетілдіру, жүйені басқарып үйрену, белгіленген жағдайда олардың жосығын болжау және т.с.с. Зерттеу процесінде ең жиі қолданылатын әдістердің бірі – модельдеу болып табылады.

Оқулықта пайдаланылатын негізгі түсінікке анықтама берейік.

Жүйе – белгілі бір мақсатқа жету үшін қызмет ететін және бір-бірімен әрекеттесетін нысандардың жиынтығы болып табылады [5]. Жүйе жоғарырақ қатардағы басқа жүйенің элементі (жүйе үсті) болуы және төменірек қатардағы жүйеден (жүйе асты) тұруы мүмкін.

Модель деп – түпнұсқа нысан жайлы жаңа мағлұмат алу мақсатында зерттеуші құратын және жасап шығарушы тұрғысынан түпнұсқаның өзіне тән ғана қасиетін көрсететін кез-келген табиғи нысанды айтады [2].

Кез-келген модель модельдеуді жүргізу үшін құрылады. *Модельдеуді* нысанның қандай да бір құбылысы, процесі немесе жүйесін құру және олардың моделін зерделеу жолымен зерттеу деп түсінеміз.

Бір нысанға көптеген модель сай келетіндігін айта кеткен жөн, өйткені әр түрлі мақсаттағы зерттеулер үшін түпнұсқаның маңызды қасиеттері ерекшеленуі мүмкін. Мысалы, адамның моделі ретінде оның мүсінін, қаңқасын, киімінің мөлшерін, талдау нәтижелерін және т.с.с. пайдалануға болады. Сонымен қатар бір үлгі әр түрлі бірнеше

нысанға сай келуі мүмкін.

Тағы бір айта кететін жайт, модельдеу кезінде зерттелетін жүйеге барабар модельді пайдаланған маңызды. Бұл дегеніміз, тәжірибе қоюшы көзқарасымен зерттеу мақсатына жету үшін жеткілікті модель мен жүйенің маңызды қасиеті сай келуі керек. Жүйенің өзіне тәжірибе қою кезінде ең барабар нәтижелер алынатындығы анық. Бірақ зерттеу процесінде нақты жүйені емес, ал оның моделін пайдалану керектігін меңзейтін бірнеше орынды себептер бар.

Біріншіден, нақты жүйелер өте күрделі және оның жұмысына әсер ететін факторлардың саны тіпті көп. Тәжірибе жүргізу процесінде осы факторлардың бәрін бақылау және өлшеу мүмкін болмайды. Сондықтан зерттеуші модель көмегімен жағдайды жеңілдету қажет, нәтижесінде осы факторлардың алуандығы қажетті деңгейге дейін азаяды және оларды бақылау және өлшеу біршама жеңілдейді.

Екіншіден, нақты жүйеге тәжірибе жүргізу бірқатар себептер бойынша көптеген жағдайда мүмкін болмайды, мысалы, тәжірибенің аса жоғары құны (елдің экономикасы), жүйенің өзі мен қоршаған ортаға оның қаупі (ядролық реактор), ғылым мен техниканың заманауи деңгейінің шектелгендігі.

Сонымен қатар зерттелетін жүйелер тәжірибе жүргізу үшін өте үлкен (Күн жүйесі) немесе кіші (атом) болуы, зерттелетін процестер өте жылдам (іштен жанатын қозғалтқыш процесі) немесе тым баяу (геологиялық процестер) болуы, зерттелетін нысан жоқ (ескі өркениет) немесе әлі ойлап табылмаған (жаңа ұшу аппараты) болуы мүмкін. Модельді пайдалану осындай жүйелерге жүргізуге мүмкіндік береді.

Модельдеу анықтамасында түпнұсқаның қандай қасиеттерін маңызды деп санау және модельге қою керектігі көрсетілмеген. Бұл сұраққа жалпы және толық жауап жоқ. Түпнұсқаның ең маңызды қасиеттерін таңдау модельді құрайтын зерттеуші тәжірибесі негізінде анықталады және көп жағдай модельдеу мақсатына байланысты белгілі болады. *Модельдеуді қолданудың әдеттегі мақсаты* төмендегідей:

1) *түпнұсқаны зерттеу.* Бұл жағдайда зерттеуші нысан немесе құбылыстың мәнін зерттеумен айналысады;

2) *талдау* («егер былай болса ше...»). Бұл кезде зерттеуші түпнұсқада әр түрлі әсердің салдарын болжауды үйренеді;

3) *синтез* («... қалай жасауға болады»). Бұл кезде зерттеуші түпнұсқаға әсер ету арқылы оны басқаруды үйренеді;

4) *оңтайландыру* («қалай жақсырақ жасаса болады»). Бұл жағдайда белгіленген жағдайда ең жақсы шешімді таңдау қажет.

1.2. МОДЕЛЬДЕУ ТҮРЛЕРІН ЖІКТЕУ

Модельдеудің әр түрлі түрлерінің қарапайым жіктелуі сур. 1.1 ұсынылған.

Физикалық модельдеу кезінде жүйенің өзі немесе оған ұқсас және сол немесе басқа физикалық табиғатқа ие жүйе (мысалы, кеменің кішірейтілген модельдерін гидродинамикалық зерттеу) пайдаланылады. Физикалық модель кішірейтілген немесе ұлғайтылған масштабта (масштабталған модельдер) жүзеге асырылуы мүмкін.

Математикалық модельдеу деп математикалық модельдің дәл жүйесіне сәйкестілікті орнату процесі және нақты жүйенің сипаттамасын алуға мүмкіндік беретін осы үлгіні зерттеу деп түсінеді. Математикалық модельдеуді қолдану шын тәжірибе қою қиын немесе мүмкін емес (қымбат, денсаулық үшін қауіпті, бір реттік процестер, физикалық немесе уақыттың шектеуден мүмкін болмайтын – алыста орналасқан, әлі немесе қазір жоқ және т.с.с) нысандарды зерттеуге мүмкіндік береді. Модель түріне байланысты

Сур. 1.1. Модельдеу түрлерінің жіктелуі

математикалық модельдеуді аналитикалық және компьютерлік деп бөледі.

Аналитикалық модельдеу кезінде жүйе элементтерінің қызмет ету процестеріелестету үшін математикалық салыстыру (алгебралық, интегралды, дифференциалды, логикалық және т.б.) пайдаланылады. Аналитикалық модельді келесі әдістермен зерттеуге болады [12]:

а) *аналитикалық*, ол кезде изделіп отырған сипаттама үшін анық тәуелділік орнатылады;

б) *сандық,* ол кезде берілген кіру мәндері үшін жуық сандық шешім алынады;

в) *сапалық*, анық түрде шешімнің тек кейбір қасиетін ғана табуға болатын, мысалы, маңызды, жалғыздық, тәуелділік сипаты.

Аналитикалық модельдің мысалы математикалық маятниктің тербелу моделі болып табылады (сур. 1.2).

Маятниктің қозғалысы дифференциалды теңдеумен сипатталады

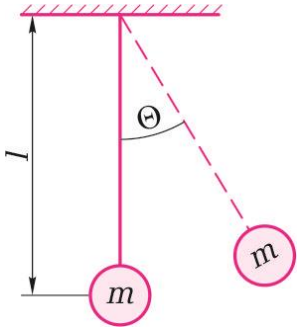
$$ml^2\left[d^2\theta(t)/dt^2\right] = mgl\theta(t) + 0, \qquad (1.1)$$

мұндағы *m, l*–маятник ілгішінің сәйкес салмағы мен ұзындығы; *g*–бос құлаудың үдеуі; 9(f) –уақыт мезетіндегі f маятниктің ауытқу бұрышы.

Осы теңдеуді пайдаланып, зерттеушіні қызықтыратын сипаттама бағасын алуға болады.

Аналитикалық шешім үнемі қалаулы екендігін ескерейік, бірақ оны әдетте модельдің барабарлығын біршама нашарлататын бірқатар жеңілдететін болжамнан соң алуға болады.

Компьютерлік модельдеу кезінде математикалық модель ЭЕМ арналған алгоритм немесе бағдарлама түрінде қалыптасады, ол онда есептеу тәжірибелерін жүргізуге мүмкіндік береді. Қолданатын әдістерге байланысты компьютерлік модельдеуді айтарлықтай шартты түрде сандық, статистикалық және имитациялық деп бөлуге болады.



Сур. 1.2. Математикалық маятник

Сандық модельдеу кезінде есептік математика әдістері (модель көрсеткішінің әр түрлі жиынтығындағы модельді беретін теңдеу сандық шешу) пайдаланылады.

Статистикалық модельдеу кезінде жүйенің статистикалық сипаттамасын алу мақсатында жүйе (модель) жайлы ақпаратты өңдеу орындалады.

Имитациялық модельдеу кезінде зерттелетін жүйенің қызмет ету процесі оның логикалық және уақыттық жүру бірізділігін ұстану кезінде ЭЕМ қайта жаңғыртылады, ол белгілі бір уақыт мезетіндегі жүйенің жай-күйі немесе оның жекелеген элементтері жайлы ақпаратты беруге мүмкіндік береді.

Оқулықтың қалған бөлігі дәл сол имитациялық модельдеуге арналады. Сондықтан оны егжей-тегжейлі қарастырайық.

1.3. ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ

Қазіргі кезде имитациялық модельдеу күрделі жүйені зерттеудің ең тиімді жобалау сатысында жүйенің құлқы жайлы ақпарат алудың жалғыз практикалық қол жетімді әдісі болып табылады. Аналитикалық модельдеумен салыстырғанда имитациялық модельдеудің негізгі артықшылығы күрделірек міндеттерді шешу мүмкіндігінде. Имитациялық модельдер дискретті және үздіксз элементтердің болуы, жүйе элементтерінің сызықсыз сипаттамалары, көптеген кездейсоқ әсер мен аналитикалық зерттеу кезінде жиі қиындық тудыратын көптеген басқа да факторлар сияқты факторларды айтарлықтай оңай ескеруге мүмкіндік береді.

Имитациялық модельдеуді белгілі бір жағдай бар кезінде пайдаланған жөн. Бұл жағдайларды Р. Шеннон анықтайды [13].

1. Берілген тапсырманың заңды математикалық қойылымы жоқ, немесе қалыптасқан математикалық модельді шешудің аналитикалық әдістері әлі жасап шығарылған жоқ. Бұл санатқа кезекті қарастырумен байланысты жаппай қызмет көрсететін көптеген модельдер жатады.

2. Аналитикалық әдістер бар, бірақ математикалық процедура барынша күрделі, имитациялық модельдеу ол тапсырманы шешудің оңай тәсілін береді.

3. Белгілі көрсеткіштерді бағалаудан басқа имитациялық модельде белгілі бір уақыт аралығында процестің барысын бақылау дұрысырақ.

Имитациялық модельдеуді сонымен қатар білім беру мен кәсіби дайындық саласында да сәтті қолдануға болады. Имитациялық модельді әзірлеу мен пайдалану оқушыларға модельде шын процестер мен жағдайларды көруге мүмкіндік береді.

Сонымен қатар имитациялық модельдеуді қолдану процесінде бірқатар мәселе туындайды [13], олар зерттелетін нысан жайлы нақты емес ақпарат алуға әкеп соғады, сондықтан зерттеуші бұл мәселелерді білуі және оларды шешуге тырысуы керек.

Бірінші мәселе модельдің дәлдігі мен күрделілігі арасында мәміле табуда. Р. Шеннонның ойынша, модельдеу өнері негізінен жүйенің зерттелетін сипаттамаларына әсер етпейтін немесе мардымсыз әсер ететін факторларды табу мен жоюдан тұрады. Егер модель өте қарапайым және онда кейбір маңызды факторлар ескерілмеген болса, онда осы модель бойынша қате нәтиже алу мүмкіндігі жоғары. Егер модель аса күрделі және оған зерттелетін жүйеге аз әсер ететін факторлар кірсе, онда осындай модельді құруға кететін шығын айтарлықтай артады және модельдің логикалық құрылымындағы қателік қаупі жоғарылайды. Сондықтан модельді құрайтын жүйенің құрылымын және оның элементтері аласындары қатынасты талдау, кірулік әсер жиынтығын зерттеу, зерттелетін жүйе жайлы қолда бар статистикалық мәліметті егжей-тегжейлі өідеу бойынша үлкен жұмыс жасау қажет. Ең маңызды факторларды таңдау кезінде әдетте Парето қағидасын негізге алады, онда ең маңызды факторлардың 20 %жүйенің қасиетінің 80 %анықтайды, ал фактордың қалған 80 % оның қасиетінің тек 20 % анықтайды.

Екінші мәселе қоршаған ортаның кездейсоқ әсерін жасанды қайта жаңғыртуға негізделеді. Бұл мәселе өте маңызды, өйткені модельден алынған дұрыс емес қайта жаңғыратын кездейсоқ әсер нәтижесі үлкен ықтималдылықпен шындыққа жанаспайтын болады.

Кездейсоқтықты қалыптастыру үшін модельде әдетте жалған кездейсоқ сан генераторы аталатындар пайдаланылады, оның сапасы модельді пайдаланбас алдын егжей-тегжейлі тексерілуі керек.

Үшінші мәселе модельдің барлығын және оның көмегімен алынған нәтижелерді бағалау болып табылады. Модельдің барабарлығы өзінің нақтылығын растаған басқа модельдермен салыстырып сарапатамалық бағалау әдісін пайдаланып, адынған нәтижелерді жүйе бойынша шын ақпаратпен салыстырып бағалауға болады.

Айта кететін жайт, әр түрлі жүйелерді модельдеу үшін жарамды имитациялық модельдердің көп түрі бар. Кейбір белгілеріне сай

имитациялық модельдердің дихотомикалық жіктеуін қарастырайық. Имитациялық модельдер келесі түрлерге бөлінеді [12].

1. *Статикалық және динамикалық.* Статикалық имитациялық модель – бұл белгілі бір уақыт мезетіндегі жүйе (жүйенің суреті) немесе уақыт ешқандай рөль атқармайтын жүйе. Статикалық имитациялық модельдің мысалдары Монте-Карло әдісі бойынша құрылған модельдер болып табылады. Динамикалық имитациялық модель уақыт бойынша өзгеретін жүйе болып табылады, мысалы жаппай қызмет көрсету жүйесі (ЖҚЖ).

2. *Детерминделген және стохастикалық.* Егер имитациялық модельде мүмкін (кездейсоқ) компоненттер болса, ол детерминделген деп аталады. Әр молекула белгілі бір бағыты мен қозғалыс жылдамдығы бар шарик түрінде ұсынылған газдағы молекулалардың қозғалысын модельдеу мысал бола алады. Екі молекуланың немесе молекулалардың ыдыстың қабырғасымен өзара әсерлесуі абсолютті серпімді соқтығысу заңына сай жүзеге асады және алгоритмдік оңай сипатталады. Стохастикалық имитациялық модельде бірқатар кіру компоненттері кездейсоқ болып табылады. Жаппай қызмет көрсету және қорды басқару жүйелерінің көпшілігі стохастикалық болып табылады. Стохастикалық имитациялық модельдер өздігінен кездейсоқ болып табылатын нәтижені береді және сондықтан ол модельдің шын сипаттамасын бағалау ретінде ғана қарастырылуы мүмкін.

3. *Үздіксіз және дискретті.* Дискретті модельдерде жүйенің жай-күйін сипаттайтын ауыспалылар әп-сәтте өзгереді, ал үздіксізде – уақыт бойынша үздіксіз өзгеріп отырады. Үздіксіз модельдің мысалы ауыспалы жағдайы (координатасы, жылдамдығы, қозғалыс бағыты) әп-сәтте өзгере алмайтын қозғалып келе жатқан автокөліктің моделі бола алады. Ауыспалы жағдай (кезектегі клиент саны, қызмет көрсету құрылғысының жай-күйі) уақыт бойынша әп-сәтте өзгеретін ЖҚЖ моделі дискретті модельдің мысалы бола алады. Сондықтан жақсы имитациялық модельдерге келесі талаптар қойылады:

- модельде сәйкес жүйеге барабар болуы керек;
  - алынатын нәтиженің қажетті дәлдігі қамтамасыз етілуі керек;
  - пайдаланушының модельмен жұмыс істеу қолайлылығы қамтамасыз етілуі керек;
- модельдің жұмыс жылдамдылығы жеткілікті болуы қажет;
- алынған нәтижелердің көрнекілігі қамтамасыз етілуі тиіс;

- модельдеді әзірлеу мен пайдалану құнының қол жетімділігін қамтамасыз ету керек.

1.4. КОМПЬЮТЕРЛІК МОДЕЛЬДЕУДІ ҚҰРЫ МЕН ҚОЛДАНУДЫҢ НЕГІЗГІ САТЫЛАРЫ

Имитациялық модельді әзірлеудің бірізді үрдісі қарапайым модель құрудан басталады, ол кейін мәселені шешуге ұсынылатын талаптарға сай ақырындап күрделенеді.

Модельді құру мен қолданудың әмбебап сызбасы жоқтығына қарамастан осы үрдісітің әдеттегі сатыларын бөлуге болады [5].

1. *Жүйені зерттеу міндеттері мен жоспарлауын құру*. Осы сатыда зерттеуші алдында тұрған мәселе қалыптасады және зерттеудің модельдеу нәтижесінде қол жеткізуге қажетті негізгі мақсаттар анықталады. Сонымен қатар модельдеу нысанының алғашқы зерттелуі орындалады, тәжірибе қоюшы қарауында бар қорлар талқыланады, модельді әзірлеуде қолданылатын тәсілдер таңданады, зерттеудің ірі жоспары құрылады.

2. *Ақпаратты жинақтау және модельді анықтау.* Зерттеуші жүйенің тұжырымдамалық моделін құруды жүзеге асырады. Сонымен қатар жүйені егжей-тегжейлі зерттеу орындалады, оның шекарасы анықталады, қажетті жеңілдету мен жуықтау орнатылады, жүйенің маңызды элементтері мен қасиеттері анықталады, өлшем бірлігі, өлшеу диапазоны, тарату заңы, ауыспалының фукционалды тәуелділігі мен модельдің көрсеткіштері орындалады. Бұл сатыда әзірленген модельдің барабарлығын одан әрі тексеру үшін шын жүйенің жұмы сипаттамалары жайлы ақпаратты мүмкіндігінше жинайды. Тұжырымдамалық модельді белгілі және жақсы әзірленген типтік жүйе ретінде елестеткен жөн: жаппай қызмет көрсету, қорды басқару, автореттеу және т.б.

3. *Тұжырымдамалық модельдің барабарлығын анықтау.* Жетекші, аналитик, сонымен қатар зерттелетін тақырып бойынша сарапшыдан тұратын аудиторияны қарастыру ұсынылатын тұжырымдамалық модельді құрылымдық талдау орындалады. Талдау одан кейін модельдің қайта бағдарламалануын болдырмайтын бағдарламалау басталғанға дейін жүргізіледі. Ол модель үшін

қабылданған жорамал дұрыс және ештеңе жіберілмегендігіне көз жеткізуге көмектеседі. Бұл жағдайда, егер модель жүйеге барабар болмай шықса, зерттеуші 3 сатыға қайтады.

4. *Компьютерлік бағдарлама құру және оны тексеру.* Модельді бағдарламалау мен дұрыстау орындалынады. Сонымен қатар не әмбебап бағдарламалау (C++, C# және с.с.) қолданылады, не модельдеуге арналған бағдарламалық қамсыздандыру (ReThink, Arena, GPSS, Pilgrim және т.б.)пайдаланылады. Бағдарламалау тілін пайдаланудың артықшылығы жүйенің кез-келген ерекшелігін жүзеге асыруға мүмкіндік бере отырып, олар үлкен икемділікке ие болады, және оның көмегімен құрылған модель орындау үшін аз уақытты талап етеді. Модельдеудің бағдарламалық қамсыздандыруын қолдану бағдарламалау уақытын азайтады және модельді іске асыру кезінде қателік ықтималдылығын төмендетуге мүмкіндік береді.

5. *Алдын ала болжам орындау.*Бұл сатыда келесі зерттеу сатысында имитациялық модельдің жүйеге барабарлығын тексеруді жүзеге асыру мақсатында оның алдын ала болжамын орындайды.

6. *Бағдарламалық модельдің сәйкестігін тексеру.*Бағдарламалық модельдің зерттелетін жүйеге барабарлығын тексеру орындалады. Сонымен қатар егер жүйе бар болса, онда бірдей бастапқы мәлімет кезінде модель мен бар жүйенің сипаттамаларын салыстыруға болады. Сонымен бірге, олардың дұрыстығын анықтау мақсатында зерттелетін тақырып бойынша модельдеу нәтижелерін аналитиктер мен сарапшыларға көрсетуге болады. Егер бағдарламалық модель осы тексерістен өтпесе, онда жүйенің тұжырымдамалық моделін айқындау мақсатында 2 сатыға қайтып келу қажет.

7. *Тәжірибені жоспарлау.* Компьютерлік тәжірибенің стратегиялық және тактикалық жоспарлауы орындалады. Бұл сатының негізгі мақсаты қысқа уақыт аралығында барлық қойылған мәселеге талап етілген нақтылығы бар жауап алуға мүмкіндік беретін сондай жоспарлы тәжірибені алу болып табылады. Нәтижесінде модельдеу нәтижесін алу керек бастапқы мәліметтің жиынтығы анықталады, сонымен бірге нәтижелердің белгіленген дәлділігі мен нақтылығын қамтамасыз ететін бастапқы мәліметтің айқын жиынтығына арналған модельдеу көрсеткіштері анықталады.

8. *Жұмыс аралығын орындау.*Алдыңғы сатыда құрыстырылған тәжірибе жоспарына сай модельдеу нәтижелерін алу. Осы нәтижелер келесі сатыда талданады.

9. *Шығатын мәліметті талдау.* Жүйенің белгілі конфигурациясының барабар сипаттамаларын анықтау немесе жүйенің барабар конфигурациясының салыстырмалы теңестіруін жүргізу мақсатында

модельдеу нәтижесін талдау орындалады. Сонымен қатар регрессионды, дисперсті талдау тәсілдері, оңтайландырудың градиентті және басқа да әдістері пайдаланылуы мүмкін. Алынған нәтиже негізінде зерттеу басында қойылған мәселелер бойынша шешім қабылдау үшін жеткілікті қорытынды жасалады.

10. *Құжаттық ұсыну мен нәтижелерді пайдалану.* Ағымдағы және болашақ жобаларда пайдалану үшін компьютерлік бағдарламаның құжаттық жорамалын және зерттеу нәтижелерін рәсімдеу. Зерттеуді тапсырушымен зерттеу нәтижесін талқылау үшін оны ұсыну, сонымен бірге егер олар нақты және анық болса, нәтижелерді шешім қабылдау процесінде пайдалануды орындау.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. Модель және модельдеу дегеніміз не? Олар қандай жағдайларда қолданылады?
2. Модельдеудің қандай түрлерін білесіз?
3. Имитациялық модельдеуді пайдаланудың негізгі артықшылықтары қандай?
4. Имитациялық модельдеуді қолдану процесінде зерттеуші қандай қиындықтарға соқтығады?
5. Жақсы имитациялық модельдерге қандай талаптар қойылады?
6. Имитациялық модельдің қандай түрлерін білесіз?
7. Имитациялық модельдеу көмегімен жүйені зерттеудің әдеттегі сатылары қандай?

ЖАЛҒАН КЕЗДЕЙСОҚ САННЫҢ БАЗАЛЫҚ ГЕНЕРАТОРЛАРЫН БАҒДАРЛАМАЛАУ

2.1. ЖАЛПЫ МАҒҰЛМАТ

Стохастикалық жүйелерді имитациялық модельдеу кезінде уақыт бойынша жүйенің қызмет ету үрдісін сипаттайтын алгоритмдерді іске асыру орындалады. Сонымен бірге қарапайым құбылыстың логикалық және уақыттық бірізділігін ескере отырып, жүйеде жүретін қарапайым құбылыс қайта жаңғыртылады. Әдетте кездейсоқ сипатқа ие осындай құбылыстарды модельдеу үшін кездейсоқ сандар қолданылады. Сәйкесінше, талап етілген көрсеткіштері мен сипаттамалары бар кездейсоқ мәнді түрлендіре алу қажет.

Жалпылама қолданылатын жүйелік бағдарламалық құралдардың көпшілігі кездейсоқ мәнді түрлендіру үшін өз құрамында арнайы стандартты бағдарламашалары болады. Бірақ көп жағдайда пайдаланушыға қажетті стандартты бағдарламашалар жоқ немесе қойылған талапқа жауап бермейтін болып шығады, сондықтан кезедейсоқ мән генераторларын өздігінен әзірлеп және тексеріп білу керек.

Базалық генераторларға кез-келген берілген таратуы бар кездейсоқ мән бірізділігін алатын интервалдағы [0; 1) біркелкі таралған кездейсоқ мәннің генераторы болып табылады. Бұндай үздіксіз кездейсоқ мәннің *X* таралу функциясы

$$Fx(x)\begin{cases} 0, \text{кезінде } \mathbf{x} < 0; \\ \mathbf{x}, \text{кезінде } 0 \le \mathbf{x} < 1; \\ 1, \text{кезінде } \mathbf{x} \ge \mathbf{1} \end{cases}$$

және ықтималдылықтың таралу тығыздығының функциясы бар

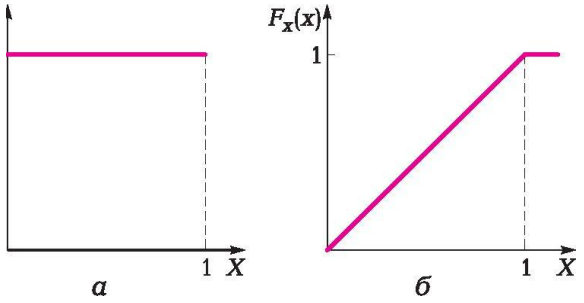
$$f_x(x) = \begin{cases} 1, \text{кезінде } 0 \le x < 1 \\ 0, \text{басқалары кезінде } x \end{cases}$$

Осындай біркелкі тараған кездейсоқ мәннің *X*математикалық болжалы (матболжал) *M(X)* = 1/2, дисперсия *D(X)* = 1/12 құрайды. Ықтималдылық тығыздығының функциясы $f_x(x)$ мен таралу функциясының $F_x(x)$ графигі сур. 2.1 бейнеленген. Одан әрі интервалда[0; 1) біркелкі таралған кездейсоқ мәнді *кездейсоқ сан* деп атаймыз.

Интервалда [0; 1) біркелкі таралу өте маңызды, өйткені барлық басқа таралудың кездейоқ мәні керекті таралумен анықталынатын кейбір әдіспен тәуелсіз кездейсоқ санның түрленуі жолымен алынуы мүмкін. осындай түрленудің кейбіреулері 4 тарауда қарастырылатын болады.

Кездейсоқ сандарды түрлендіру үшін келесі әдістердің бірі қолданылады.

1. *Аппараттық.* Аппаратты генератордың негізінде қандай да бір физикалық эффект (мысалы, электронды құрылғылардағы шуыл немесе басқа құбылыстар) жатыр. Осындай генератордың артықшылығы оның көмегімен алынатын кездейсоқ сандар мүлде кездейсоқ болып табылады. Аппараттық генератор санының қоры да шектелген. Аппараттық генератордың кемшілігі ретінде тура модельдеу кезінде кездейсоқ санның алынатын бірізділігінің сапасына кепіл бермейтіндігін атап кету керек. Сонымен қатар,



Сур. 2.1. Функция кестесі:
а–ықтималдылық тығыздығы; *б*–интервалдағы[0; 1) біркелкі тарату үшін таралуы

аппараттық генератор көмегімен модельденетін бағдарламаларды дұрыстау және жүйені ұйымдастырудың ұқсас нұсқаларын салыстыру кезінде пайдалану дұрыс кездейсоқ санның бірдей тізбегін алуға болмайды. Сондықтан модельдеу кезінде аппараттық генераторлар іс жүзінде қолданылмайды.

2. *Кестелік.* Кестелік генераторды қолдану кезінде алдын ала алынған және тексерілген кездейсоқ сандар ЭЕМ жадысында кесте түрінде рәсімделген. Бұл әдіс аппараттық генераторда айтылған кемшіліктен бос. Бірақ кестелік генераторды пайдалану кезінде кездейсоқ сан қоры кесте мөлшерімен шектелген. Сонымен қатар, ЭЕМ қоры тиімсіз пайдаланылады. Сондықтан бұл әдісті модельдеуде салыстырмалы түрде сирек қолданады.

3. *Арифметикалық (сандық).* Арифметикалық генераторды қолдану кезінде әр жаңа сан берілген математикалық формулаға сай бір немесе бірнеше туынды сандармен анықталады. Бұл әдіс бұған дейін қарастырылған генераторлардың кемшілігінен бос. Сондықтан модельдеу кезінде дәл осы арифметикалық генератор жиі қолданылады.

Айта кететін жайт, арифметикалық генераторлар көмегімен ЭЕМ кездейсоқ сан тізбегін алу кезінде алгоритмдер қолданылады. сондықтан осындай тізбектен негізінен детерминделген болып табылады, бірақ сырттан бақылаушығы кездейсоқ сияқты көрінетін, *жалған кездейсоқ* деп аталады. Тағы назар аудартатыны, кез-келген ЭЕМ текn-разрядтық сандарды пайдаланады. Осыған байланысты ЭЕМ интервалдан [0; 1) біркелкі кездейсоқ сандардың үздіксіз жиынтығының орнына сол интервалдан 2^nкездейсоқ саннан артық емес дискретті тізбек ғана алынуы мүмкін. Сондықтан осындай дискретті тізбектің таралуы *квазибіркелкі* деп.

Кездейсоқ сандық мінсіз генераторға келесі талаптар қойылды [5]:

- тізбек квазибіркелкі таралған сандардан тұруы керек;
- сандар статистикалық тәуелсіз болуы қажет;
- кездейсоқ санның тізбегі қайта жаңғыртылатындай болуы тиіс;
- тізбегі периодсыз болуы керек;
- тізбегі есептеу қорларының минимальді шығынымен алынуы қажет;
- генераторлар машина жадысының минимальді көлемін алуы тиіс.

Одан әрі кездейсоқ санның арифметикалық генераторларының

негізгі типтерін қарастырамыз. Кездейсоқ сан генераторлары кейде оқулық мәтінінде датчиктер деп аталатындығын ескерейік.

2.2. КЕЗДЕЙСОҚ САННЫҢ АРИФМЕТИКАЛЫҚ ГЕНЕРАТОРЛАРЫ

Кездейсоқ (жалған кездейсоқ) сандарды шығарудың көптеген алгоритмі, рәсімі және әдісі белгілі. Жалпы алғанда, бағдарламалық генераторлар кейбір рекуррентті формулалар бойынша сандардың реттілігін жасайды

$$x_{n+1} = \varphi(x_n, x_{n-1}, \ldots, x_{n-r}), \quad (2.1)$$

Мұнда ≠ - х п+1 санын алу үшін х п+1....х1 сандарына жасау керек операциялар жиынтығы. Сонымен қатар, х п+1....х1 реттілігінің сапасы алғашқы (бастапқы) х п+1....х1 сандарға қатыстылығы айтарлықтай болады.

Рәсімді бірнеше рет қолдану нәтижесінде (2.1) анықталған сипаттағы сандардың реттілігі алынады, алайда білгілі бір шекараларда ол біркелкі орналастыру қасиеті мен кездейсоқтық қасиетімен қанағаттандырылады.

2.2.1. Конгруэнтті (сызықтық) әдіс

Қазіргі таңда кеңінен қолданылатын сандарды жалпы түрде біркелкі бөлетін бағдарламалық генераторларды келесі формула арқылы көрсетсе болады:

$$y_{n+1} = \left(\sum_{i=0}^{r} a_i y_{n-i} + \mu\right) \bmod m, \quad (2.2)$$

Мұнда а0, а1...а және м, сонымен қатар у0, у1... сандары бүтін сандар болып табылады. Жалған кездейсоқ сандардың /0;1/ интервалында біркелкі орналастыру келесі өрнек арқылы жүзеге асырылады:

$$x_{n+1} = \frac{y_{n+1}}{m}, \quad (2.3)$$

(2.2) және (2.3) формулалары конгруэнтті әдісті анықтайды, бұл көбіне кездейсоқ сандардың сол және басқа да тетігін жасайды

Лемер алгоритмі (мультипликативті датчик). Лемер алгоритмін қолдануға негізделген кездейсоқ сан датчигі сызықты конгруэнтті әдістің жеке жағдайы болып табылады. Ол формуламен ұсынылады (2.2)

$$\mu = a_1 = \ldots = a_j = 0$$

және $a = a_0 > 0$, деп болдай отырып аламыз

$$y_{n+1} = ay_n \bmod m . \qquad (2.4)$$

Лемер алгоритмінде сәйкес тәсілмен таңдалған екі бүтін сан беріледі: көбейткіш *а*және модуль*m,*сонымен қатар бастапқы мән у0. Кездейсоқ санның тізбегі келесідей есептелінеді.

1. у саны алдыңғы қадамнан белгілі. *ayt*туындысы есептелінеді
2. *ау*-ды *m*-гебөлгеннен у+1 қалдығы қалады.
3. у +1интервалынан [0; 1) бүтін сан болып табылатындықтан, интервалдағы [0; 1) санды алу үшін оны тағы да *m* болу керек:

$$x_{i+1} = \frac{y_{i+1}}{m}$$

Айта кететін жайт, формула (2.4) көмегімен *у* әр түрлі мәнінің тек*m*алуға болады.Сондықтан кейбір қадамда *T*алдын алынған уТміндетті түрде алынады. Генератор қалыптастыратын келесі мән тек алдыңғыға ғана байланысты болғандықтан, тізбекті одан әрі қалыптастырғанда қайталанып бастайды. Алынған соң тізбек қайталанып бастайтын Т санының мөлшері *генератор периоды* деп аталады.

y_0, *а*және*т*мәндері көбінесе максималды мүмкін периодты алу талабына сүйеніп таңдалады.

Мысалы, егер*m* = 2^1,мұндағы *1* > 2 – ЭЕМ бүтін тұрақтының максималды мәнін беруге арналған екілік разрядтардың саны (мысалы,*1*15 немесе 31 тең болуы мүмкін). Бұл таңдау осындай *m* кезінде *m* модулі бойынша есептеу жолымен бөлуден қалдық алу оңайлықпен анықталады. Бұл жағдайдағы максимальді период 3 немесе 5 тең y_0 тақ және *a* mod 8 мәнінде қол жеткізіледі, ол *a* = 5^{2p+1}, *p*= 0, 1, 2, ... (немесе*a* = 2^s+ 3, *s*= 3, 4, 5, ...) болғанда орындалады. Осы жағдайдағы максималды қол жетімді мән T_{max}= 2^{i-2} = m/4.

Егер m ретінде 2^1 паз ең қарапайым сан таңдалатын болса, m – 1 периодты генераторды алуға болады. Бұндай генератор қарапайым модульді мультипликативті сызықтық конгруэнтті генератор деп аталады. Мысалы, егер b= 31 болса, онда m= 2^{31} - 1 = 2 1 47 483 647. Бұндайmүшін егер a –m модулі бойынша бастапқықалыпты элемент болса, m – 1 периодты алуға болады, яғни$a1$ - 1 m-ге бөлінетін ең аз l үшін 1 = m – 1 құрайды.

Бұндай генераторды құрастыру кезінде aкөрсеткішін таңдау қиындығын ескереміз. m = = 2^{31} - 1 = 2 1 47 483 647 және $a = a_1$ = 16 807 немесе $a = a_2$ = 630 360 016 генераторлары жақсы қасиетке ие, ал $a = a_2$ генераторы ең жақсысы [5]. Тағы да айта кететін жайт, m – 1 бөлуден ең күрделі қалдық алғандықтан генератор біршама баяу жұмыс істейді.

Мысал ретінде кездейсоқ сан генераторы қызметін іске асыратын және қолданатын C# тіліндегі консольды бағдарламаны келтірейік. Осы қызметі: y_0 = 2 451, m= 4 096, a = 5 көрсеткіштерімен мультипликативті генераторды іске асырады.

```
namespace Mult
RND {
   class Program
   {
      // Кездейсоқ сан мөлшеріprivateconstintN
      = 1500;
      // Мультипликативті генератор
      көрсеткіштеріprivateconstintA= 5;
      // Модуль
      private static int Mm;
      // Кездейсоқ санprivate static int Y;
      /// <summary>Кездейсоқ сан генераторы //
      </summary>
      /// <returns>Кездейсоқ мән</returns> private
      static double Rnd()
      {
         _Y = (A * _Y) % _Mm;
         return (double) Y / Mm;
      }
      /// <summary>Негізгі бағдарлама</summary>
      /// <param name = "args">Дәлелдер</param>
      static void Main(string[] args) {
         Mm = 4096;
```

```
        _Y = 2451;
        // Кездейсоқ санға арналған
        ауқымыdouble[] x = new double[N];
        for (int i = 0; i < N; i++)
        {
          // кездейсоқ санды алух[i] =
          Rnd();
          ... // кездейсоқ санды өңдеу
        }
        ... // кездейсоқ сан ауқымын өңдеу
      }
   }
}
```

Аралас генератор. Егер (2.2) формулаға a = a_2= a_3= ... = a_r= 0 және a = a_0> 0, ц > 0 қойса, онда аралас генераторға арналған көрінісі алынады:

$$Y_{n+1} = (ay_n+\mu) \bmod m. \qquad (2.5)$$

Генератордыңm тең максимальді периодын алу үшін a, mжәнеmбүтін сандары келесі жағдайды қанағаттандыруы керек:

1) $a \bmod 8 = 5$;

2) $m / 100 < a < m-\sqrt{m}$;

3) *a*санының екілік белгісінің анық шаблоны болмауы керек;

4) $\mu / m \approx 1/2 - \sqrt{3}/6$;

5) μ–тақ;

6) *m*–осы ЭЕМ үшін максималды бүтін сан.

Санның 32-разрядты елестетуі жағдайында көрсеткіштердің келесі мәндерін алуға болады [5]:

$$a=16\ 070\ 093;\ \mu = 453\ 816\ 693;\ m=2^{31} = 2147\ 483\ 648.$$

Аралас генераторды іске асыратын функция мысалын жазайық:

```
private double Rnd()
{
   _Y = (A * _Y + _Mu) % _Mm;
   return (double) Y / Mm;
}
```

Бұл функцияны пайдалану үшін негізгі бағдарламада *A, _Mu, _Mm*және бастапқы мән*_Y,*беру керек, мысалы:

```
…
_A = 165;
_Mu = 3463;
_Mm = 4 0 9 6 * 4;
_Y = 3887;
double[] x = new
double[N]; for (int i = 0;
i < N; i++) {
   x[i] = Rnd();
…
```

Аддитивті генераторлар. Егер (2.2) формулаға $a_0 = a_1 = 1$ және $a_2 = ... = a_r = 0$ қойса, онда алынған генератор аддитивті немесе Фибоначчи генераторы деп аталады:

$$Y_{n+1} = (Yn + Y_{n-1}) \bmod m \qquad (2.6)$$

Алдын қарастырылған генераторлармен салыстырғанда Фибоначчи генераторында екі бастапқы у0және*у1*мәндері беріледі.

Аддитивті генератор үшін келесі функцияны қолдануға болады:

```
private double Rnd()
{
   int y =  (_Y0 + _Y1) % _Mm;
   _Y0 = _Y1;
   _Y1 = y;
   return (double)y / Mm;
}
```

Бұл кезде *_Mm* және бастапқы мән_Y0, _Y1 негізгі бағдарламада анықталуы керек, мысалы:

```
_Mm = 4096*4;
_Y0 = 3971;
_Y1 = 1013;

double[] x = new
double[N]; for (int i = 0;
i < N; i++) {
```

```
   x[i] = Rnd();

}
```

Жалпыланған аддитивті генератор. Барлық a_i = 1, i= 0,1,..., r, жалпыланған аддитивті генератор деп аталатын (2.2) түрінің генераторы. Интервалда [0; 1) біркелкі таралған $x0$, $x1$,..., xN,сандарының жалған кездейсоқ тізбегін алу үшін жалпыланған аддитивті генератор келесі алгоритм түрінде іске асырылады:

$$x_{n+1} = \{x_n + x_{n+1} + \ldots + x_{n-r}\} . \qquad (2.7)$$

у санының бөлшектік бөлігі

$$\{\psi\} = \psi - \llcorner\psi\lrcorner,$$

мұндағы ⌞ψ⌟–у бүтін бөлігі.

x_0, x_1,..., x_N бастапқы мәні кездейсоқ сан кестесінен таңдалады. Формулада (2.7) қосылғыш саны кез-келген болуы мүмкін. Практикалық жағдайда r = 6...8 шектеледі. Формула (2.7) бағдарламаның келесі негізгі кесінділерін жүзеге асырады:

```
// Жаңа кездейсоқ санды алу үшін қажетті
// кездейсоқ сан мөлшеріprivateconstintR = 6;
private static double Rnd()
{
   double s = 0.0;
   for (int k = 0; k <R; k++)
   {
     s += Rand[k];//кездейсоқ сан соммасы
   }
   for (int k = 1; k <R; k++)
   {
// кездейсоқ санның жылжуыRand[k-1]=
Rand[k];
   }
   s -= Math.Truncate(s);
   //жаңа кездейсоқ
    санRand[R -1]     = s;
```

```
   return s;
static void Main(string[] args)
{
   //Randomстандартты датчигін
   инициализациялау random =
   newRandom();
   for (int i = 0; i <R; i++)
   {
      // бірінші кездейсоқ санның
      түрленуіRand[i] =
      random.NextDouble();
   }
   for (int i = 0; i < N MAX; i++)
   {
      double x = Rnd();
   }
}
```

2.2.1. Кездейсоқ санның қиыстыру генераторы

Циклді қайталанатын тізбекті модельдеу кезінде қолдануды болдырмас үшін (период ұзындығын ұзарту үшін) кездейсоқ санның қиыстыру генераторлары кеңінен қолданылады.

Екі қарапайым генераторды қолданатын осындай генератордың қарапайым мысалын қарастырайық. Олардың көмегімен кездейсоқ санның екі тізбегін алады: біреуі N_1 периодты, басқасы N_2периодты. i-ші тізбегінің (i = 1, 2) мүшесін y_{ij} j-ші арқылы (j= 1, 2, …)және формула бойынша түзілетін жаңа тізбек мүшесінZ_j–j-ші арқылы белгілейік

$$z_j = (y_{1j} + y_{2j}) \bmod m. \qquad (2.8)$$

Егер N_1 және N_2 пероидтары–өзара қарапайым сандар болса, онда нәтижесінде шығатын Z_j тізбегі –N =N_1N_2 периодымен мезгілді.

Процедураны (2.8) бастапқы тізбек санының арту жолымен жалпылауға болады. Мысалы, егер бастапқы тізбектің алғашқы саны r тең болса, онда нәтижесінде шығатынj-шімүше формула бойынша алынады

$$z_j = \left(\sum_{i=1}^{t} y_{ij} \right) \bmod m.$$

Тағы бір мысал ретінде кездейсоқ санды есептеу үшін екі ұқсас мультипликатты датчиктерді қолданатын әмбебап датчиктерді қарастырайық. Бастапқы датчиктердің әрқайсысы төмендегі түрге ие

$$y_{k,n+1} = \left| a_k \left(y_{k,n} \bmod B_k \right) - \frac{c_k y_{k,n}}{B_k} \right|; \qquad (2.9)$$

Мұндағы $X_{k, n+1}$ – интервалға [0; 1) тиесілі кездейсоқ сан; k= 1, 2, 3 – бастапқы датчиктің нөмірі; a_k, b_k, c_k, d_k–бүтін сандар; k-м датчигімен түрленетін және 0-ден(d_k -1) дейін өзгеретін y_{kin} – n-екездейсоқ саны.

Нәтижеленетін кездейсоқ сан формула бойынша есептелінеді

$$^{z}n = \{^{x}1,n + {}^{x}2,n + {}^{x}3,n\}.$$

Бастапқы деректер ретінде келесі бүтін тұрақты шаманы алуға болады:

$a_1 = 171$; $b_1 = 177$; $c_1 = 2$; $d_1 = 30\,269$
$a_1 = 172$; $b_2 = 176$; $c_2 = 35$; $d_2 = 30\,307$;
$a_3 = 170$; $b_3 = 63$; $c_3 = 63$; $d_3 = 30\,323$.

Әмбебап датчикті іске асыратын функцияның түрі келесідей:

```
private double Rnd()
{
  double s = 0.0;
  double[] x = new double[K];
  int[] yN = new int[K];
  for (int i = 0; i <K; i++)
  {
    yN[i] = (int) Math.
    Abs(_A[i]*(_Y[i]%_B[i])                -
    _C[i] * (double)_Y[i] / _B[i]);
    _Y[i] = yN[i];
    x[i] = (double) yN[i] / D[i];
    s += x[i];
  }
  return Math.Abs(s - Math.Truncate(s));
}
```

Сонымен бірге негізгі бағдарламада көмекші ауқымдар мен К тұрақты шама анықталуы керек.

```
…
/ /Алғашқы датчиктер мөлшері
privat const int  K = 3;
// Көмекші
privat int[   A    ne int[K]
privat int[   B    ne int[K]
privat int[   C    ne int[K]
privat int[   D    ne int[K]
…
```

Маклерен-Марсальи әдісі екі бастапқы датчик көмегімен алынатын $x_{1:1}, x_{1,2}, \ldots, x_{1n}, \ldots$ және x_{21}, x_{22}, $x_{2|1}$, $x_{22}, \ldots$, $x_{2,n}, \ldots$,екі кездейсоқ сан тізбегімен алуға негізделген.

Бірінші генератор көмегімен ксанынан $x_{1д}$, $x_{12}, \ldots$, x_{1k} тізбегінен әдеттек = 64, 128, 256 шығарады.

Одан әрі бірінші және екінші датчиктер көмегімен кезектіхі,$n+_к$және $x_{2,n}$сандарн түрлендіреді. Тізбекпен біріктірілгенz_nсанын операцияларды орындау нәтижесінде алады

$$M = [x_{2,\alpha} k] + 1;$$
$$z_\alpha = x_{1,M};$$
$$x_{1,M} = x_{1,\alpha+k}. \qquad (2.10)$$

Операциялар(2.10) қажетті санды бір қайталайды. Нәтижесінде екінші генератормен алынған мән көмегімен ктізбегінің $x_{1,1}$, $x_{12}, \ldots$, x_{1k},сандарынан санның кездейсоқ таңдауы, сонымен қатар жаңа сандармен кездейсоқ толтырылуы іске асырылады.

Маклерен-Марсальи әдісін іске асыру үшін бағдарламаның келесі негізгі кесінділері қажет:

```
//Таңдама көлемі
private const int N MAX = 4000;
//Кездейсоқ санның 1-ші тізбегінің
мөлшеріprivateconstintK = 64;
// 1-ші кездейсоқ тізбек
privatestaticdouble[] Z1 = newdouble[K];
```

```
//Стандартты датчик
private static Random _Random = new Random();

private static double Rnd()
{
   // 1-шітізбектіңкездейсоқсанысdouble g1 =
   Random.NextDouble();
   //2-ші тізбектің кезедейсоқ санысdouble g2 =
   Random.NextDouble(); int m = (int)(g2 *
   K); double res =  Z1[m];
   //1-ші тізбек элементін толтыру
   // жаңа санмен_Z1[m]= g1; returnres;
} static void Main(string[] args)
{
   for (int i = 0; i <K; i++)
   {
      // бірінші К кездейсоқ санның
      түрленуі Z1[i] =
      Random.NextDouble();
   }
   double x;
   for (int i = 0; i < N MAX; i++)
   {
      x = Rnd();
   }
}
```

2.2.2. Сызықсыз формула негізіндегі алгоритмдер

Шаршылы конгруэнтті әдіс. Сызықты конгруэнтті әдісті жалпылау әдісі санның кездейсоқ тізбегін қалыптастырудың шаршылы конгруэнтті әдісі болып табылады

$$Yn+1 = (Ay_a^2 + By_n + C) \bmod m, \qquad (2.11)$$

Мұндағы $m = 2'$. Егер $1 \geq 2$ болса, онда шаршылы конгуэнтті датчик периодының кішігірім мәні $T_{max} = 2^1$, ол Ажұп, С тақ кезінде және егер тақ В $B \bmod 4 = (A + 1) \bmod 4$ жағдайын қанағаттырғанда қол

жеткізіледі.

Формуланы (2.11) іске асыратын функция келесі түрге ие:

```
privatedoubleRnd()
{
   _Y = (_A* _Y * _Y + _B* _Y + _C)   % _Mm;
   return(double) Y/ Mm;
}
```

Сонымен қатар негізгі бағдарламада _A, *_B, _C, _Mm*және бастапқы*_Y* мәні берілуі керек, мысалы:

```
…
 A = 6;
_B = 7;
_C = 3;
 Mm = 4096;
_Y = 4001;
for (int i = 0; i <N; i++)
{
   doublex = Rnd();
   …
}
…
```

Ковэю шаршылы әдісі. Осы әдісті пайдалану кезінде кезекті санды есептеу формула бойынша жүзеге асырылады

$$Уп+i = Уп\ (Уп + 1) \bmod m, \qquad (2.12)$$

Мұндағы $m = 2^l$,ал бастапқы мән $y_0 \bmod 4 = 2$ теңдеуін қанағаттандырады.

Сонымен қатар осындай датчиктің кездейсоқ тізбек кезеңінің ұзындығы 2^{l-2} құрайды.

Формуланы (2.12) іске асыратын функция келесі түрге ие:

```
private double Rnd()
{
   _Y = _Y *  (_Y + 1)      % _Mm;
   return(double) Y/ Mm;
}
```

Сонымен қатар негізгі бағдарламада *_Mm* және _Y берілуі керек, мысалы:

```
_Mm = 512;
_Y = 2135;
for (int i = 0; i <N; i++)
{
   doublex = Rnd();

}
```

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. Біркелкі тараған кездейсоқ шама қалай беріледі?
2. Кездейсоқ санның түрленуінің қандай негізгі әдістерін білесіз?
3. Біткелкі тараған кездейсоқ санның түрлендіретін конгруэнтті әдісі неге негізделген?
4. Түрленудің конгруэнтті әдісінің қандай түрлері сізге белгілі?
5. Жалған кездейсоқ тізбек деген не? Периодтың ұзындығын ұлғайту әдісін сипаттаңыз.
6. Күрделі детчикті қалыптастыру үшін кездейсоқ датчиктердің құраушыларының негізгі қағидалары кандай? Оларды қалыптастырыңыз.

КЕЗДЕЙСОҚ САННЫҢ ТҮРЛЕНУІН ТЕКСЕРУ ӘДІСТЕРІ

3.1. ТЕҢ ЫҚТИМАЛДЫ ТАРАЛУДЫҢ ЖИІЛІК ГИСТОГРАММАСЫ МЕН СТАТИСТИКАЛЫҚ ФУНКЦИЯСЫ

ЭЕМ жүйелердің статистикалық модельдеу тиімділігі мен алынған нәтижелердің дәлділігі көбіне жалған кездейсоқ санның базалық тізбегінің сапасына байланысты. Сондықтан ЭЕМ модельдейтін алгоритмдерді іске асыруға кіріспес алдын жалған кездейсоқ санның бастапқы тізбегі оған қойылған талаптарға жауап беретіндігіне көз жеткізу керек.

Генераторларды тексеру кезінде жалған кездейсоқ сандар тізбегінің негізгі статистикалық сипаттамаларын бағалауға (маткүту, дисперсия және т.б.), жиілік гистограммасы мен статистикалық функцияны құруға және олардың теоретикалық функциялармен көзбен салыстыруға болады.

Кездейсоқ санның сапасын формалды тексеру үшін әдетте әр түрлі N көлемдегі кездейсоқ мән тізбегінің бағдарламалық генераторының жасалған әр түрлі бөліктерін көп реттік тексеруге негізделген бірқатар критерийлер қосымша жасап шығарылған, әрі N айтарлықтай үлкен болуы керек. Одан әрі қарастырылған критерийлер кез-келген таралу заңының кездейсоқ ауқымы үшін қолдануға болады, бірақ одан төмен олар тең ықтимал кездейсоқ санның сапасын тексеру процесінде қолданылады.

Бақылау саны артқанда N саны үлкен болған кездегі кездейсоқ $x_j,...,x_N$ мәнінің статистикалық жиынтығы статистикалық материалды жазудың қолайлы пішіні болудан қалады. Оған үлкен ықшамдылық беру үшін статистикалық қатар деп аталынатын құрылады. Осы мақсатта кездейсоқ ауқымның мүмкін мәнінің барлық диапазоны (0-ден 1-ге дейінгі диапазоны бар біркелкі заң үшін талқыланатын

генератор жағдайында) әрқайсысын $\Delta = 1\ k$ ұзындықты κ тең бөлікке бөледі. Тәжірибелік жағдайда κ тең ретінде 10...20 таңдайды. Сонымен қатар κ Стерджес формуласы бойынша бағалауға болады

$$k = \lfloor 1 + \log_2(N) \rfloor.$$

Одан әрі қолда бар $x_1,...,x_N$ таңдамасы бойынша i-ші бөлікке келетін v_1 мәнінің мөлшерін есептейді. Кездейсоқ ауқымның і-ші интервалға түсуінің p_i ықтималдылығын бағалау төмендегі формуламен анықталады

$$\hat{p}_i = \frac{v_i}{N}.$$

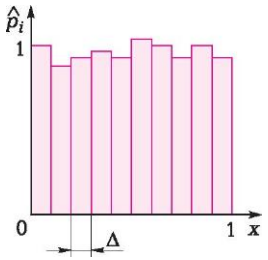
$p_1, p_2, \ldots, p_k$ статистикалық қатарының графикалық көрінісі *жиілік гистограммасы* деп аталады. Барлық интервалдардың p_i жиілігінің соммасы бірге тең болуы керектігі анық.

Тең ықтимал таралу үшін жиілік гистограммасының мысалдық түрі 3.1 суретте көрсетілген. Гистограмманың сыртқы түрі бойынша $x_1,..., x_N$ сандарының тізбегі негізінде жатқан таралу заңы жайлы болжауға болады.

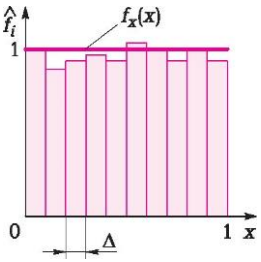
Ықтималдылық тығыздығы бар жиілік гистограммасын салыстыру үшін нормаланған жиілік гистограммасын пайдалануға болады (сур. 3.2), оны құруға арналған мәндер формула бойынша анықталады

$$\hat{f}_i = \frac{v_i}{\Delta N}.$$

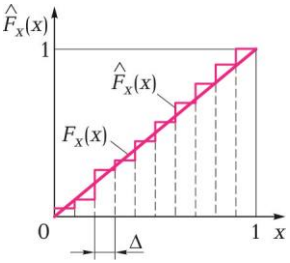
Тәжірибе саны артқан кезде А майдалау интервалдарын беруге болатындығын айта кетейік. Сонымен қатар нормаланған гистограмма ықтималдылығының f_x (x)тығыздығына жақындайды.



Сур. 3.1. Жиілік гистограммасы



Сур. 3.2. Нормаланған жиілікті гистограмма



Сур. 3.3. Таралудың статистикалық функциясы

Статистикалық қатардың дерегін пайдалана отырып, X кездейсоқ шаманың $F_X(x)$ таралуының статистикалық (эмпирикалық) жуық функциясын құруға болады.

Таралудың эмпирикалық функциясы деп әр xмәні үшін уақиғаның салыстырмалы жиілігін$^{X<x}$ анықтайтын функцияны айтады,

$$\hat{F}_X(x) = \frac{N_X}{N},$$

Мұндағы N_X аз $x_1,\ldots,x_N$ жиынтығынан мәнінің саны; N–таңдама көлемі.

Эмпирикалық $F_X(x)$ және теориялық$F_X(x)$ функциялар арасындағы айырмашылық теориялық функция $F_X(x)$ уақиғаның$X < x$ ықтималдылығын анықтайды, ал эмпириялық функция $FX(x)$ –осы уақиғаның салыстырмалы жиілігін анықтауында тұрады.

Мысал ретінде сур. 3.3 тең ықтимал таралуға арналған әдеттегі статистикалық функция көрсетілген.

3.2. ТАРАЛУ КӨРСЕТКІШТЕРІН СТАТИСТИКАЛЫҚ БАҒАЛАУ

x_i, i= 1, Nтаңдамаларындағы деректі өңдеу міндеттің екі класы ретінде анықталады: таңдама бойынша таралу көрсеткіштерін бағалау; статистикалық болжамды тексеру.

Екінші класс міндеттері келесі бөлімшелерде тең ықтимал таралуға қолданып қарастырылады. Бірінші кластың міндеті белгілі таңдама бойынша таралудың қандай да бір сандық сипатамасын (математикалық күтілім, дисперсия, мезет пен с.с.) бағалауға, яғни Ө

$$Ө = w(x_i)$$

көрсеткішінің орнына оның жуықтауы ретінде падалануға арналған таңдама функциясын көрсетуге мүмкіндік береді. Ө статистикасын Ө көрсеткішінің бағасы деп атайды.

Мысалы, математикалық күтілудің жылжытылмаған бағалауы ретінде *таңдалған орташаны* пайдалануға болады

$$\bar{X}(N) = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Дисперсияны бағалау үшін *таңдамалы диспесия* пайдаланылады

$$S^2(N) = \frac{1}{N}\sum_{i=1}^{N}\left[x_i - \bar{X}(N)\right]^2. \qquad (3.1)$$

Машиналық тәжірибені жүргізу кезінде формуланы (3.1) қолдану ыңғасыз болып табылатындығын байқаңыз, өйткені бағаны S^2(N) алу үшін екі іске асыру (екі таңдама) керек, біріншісі $X(N)$ бағалауын алу, ал екіншісі–S^2(N) бағасын алу қызметін атқаратын. Бір таңдаманы екі рет пайдалануға болады, бірақ оны екі рет өңдеуге тура келеді.

Осы кемшілікті болдырмау мақсатында таңдамалы дисперсияны анықтау үшін келесі формуланы қолданады:

$$S^2(N) = \frac{1}{N}\sum_{i=1}^{N} x_i^2 - \bar{X}^2(N),$$

мұндағы екінші бастапқы мезет

$$\hat{\alpha}_2(X) = \frac{1}{N}\sum_{i=1}^{N} x_i^2$$

математикалық күтілумен параллель бағалануы мүмкін. Жоғарырақ қатардың бастапқы мезетінің бағалауы бар

$$\hat{\alpha}_m(X) = \frac{1}{N}\sum_{i=1}^{N} x_i^m, \quad m = 3, 4, \ldots$$

Таңдамалы дисперсия диспесияның жылжыған бағалауы болып табылатындығын байқайық. Жылжымаған бағаны алу үшін *түзетілген таңдамалы дисперсия* пайдаланылады

$$S^2 = \frac{N}{N-1}S^2(N) = \frac{1}{N-1}\sum_{i=1}^{N}\left[x_i - \bar{X}(N)\right]^2.$$

Таңдама көлемі біршама үлкен болған кезде таңдамалы және түзетілген таңдамалы дисперсия айырмашылық аз екендігін айта кетейік. Сондықтан практикада түзетілген диспесияны таңдама көлемі кіші-гірім болған кезде ғана пайдаланады (мысалы, $N < 30$ кезінде).

Таралу сипаты жайлы қосымша ақпаратты ассиметрия көмегімен алуға болады, ол математикалық күтілуге қатысты таралу симметриялығын бағалауға мүмкіндік береді. Ассиметрия төмендегідей бағаланады

$$\hat{v}(X) = \frac{\sum_{i=1}^{N} [x_i - \bar{X}(N)]^3}{N|S^2|^{3/2}}.$$

Егер ассиметрия нөльден көп болса, онда таралу оңға жылжыған, егер нөльден кіші болса, онда солға, егер нөльге тең болса, онда таралу симметриялы.

Осы бағалауды кездейсоқ шаманың тәуелсіздігін жағдайында пайдалануға болады. Бірақ модельдеу нәтижесінде алынған шығатын дерек үнемі дерлік корреляцияланған болып табылады. Бұл жағдайда таңдамалы орташаX(N) m бағасымен жылжымаған күйде қалады, ал түзетілген тадамалы S^2дисперсия о^{2}бағасымен жылжытылмаған

$$E[S^2] = \sigma^2 \left[1 - 2\frac{\sum_{j=1}^{N-1} (1 - j/N)\rho_j}{N-1} \right],$$

күйде қалады, өйткені

мұндағыPj–корреляция коэффициенті.

Осылайша, егер практикада жиі болатын pj > 0 (оң корреляция) болса, түзетілген таңдамалы дисперсияS^2 теріс жылжуға ие болады$S^2 <$ ст^2.

Корреляция коэффициентін былайша бағалауға болады:

$$\hat{\rho}_j = \frac{\hat{C}_j}{S^2},$$

мұндағыCj– ковариация бағасы формула бойынша анықталады

$$\hat{C}_j = \frac{\sum_{i=1}^{N-j} [x_i - \bar{X}(N)][x_{i+j} - \bar{X}(N)]}{n-j}.$$

Ескертетін жайт, бағалардың төмендеуі келесі жайттарға байланысты болады, Егер*N*өте үлкен мәнді қабылдамаса және *j* мәні *N* қатысты аз болса, pj нашар баға алынатындығын айта кетейік.

Одан әрі жалған кездейсоқ санның мультипликативті генераторының мысалында гистограмма, таралудың статистикалық функциясы үшін деректі алуда пайдаланылатын бағдарлама кесінділері, сонымен қатар кейбір таралу көрсеткіштерінің кейбіреулерін статистикалық бағалауы келтірілген:

```
namespace Mult RND
test {
   class
   Program {
   // Кездейсоқ сан мөлшері
   privateconstintN= 1500;
   // Мультипликативті датчик
   көрсеткішіprivatestaticint _A;
   // Модуль
   private static int Mm;
   // Кездейсоқ
   санprivate static int
   Y;
   // Ұсақтау интервалдарының
   саны privatestaticintK;
   /// <summary>Кездейсоқ сан генераторы
   // </summary>
   /// <returns>Кездейсоқ сан</returns>private
   static double Rnd()
   {
     _Y =  (_A *  _Y) %  _Mm;
     return (double) Y / Mm;
   }
   /// <summary>деректер жиынтығын
   қалыптастыру // </summary>
   /// <param name = "parValues">Кездейсоқ сан
   ауқымы /// </param>
   static void GenerateData(out double[]parValues)
   {
     parValues = new double[ N];
     // кездейсоқ санды алуfor
     (int i = 0; i <N; i++)
     parValues [i]   = Rnd();
   }
```

```
/// <summary> Таралудың        тығыздығы мен
    функциясына   арналған            ауқымы
/// алу           </summary>
/// <param name = "parValues" Кездейсоқ сан
/// ауқымы</param
/// <param name = "parDataPlo ">      Жиілік ауқымы
/// </param>
/// <param name = "parDataFun ">      Таралу функциясының
/// ауқымы        </param>
/// <param name = "parMin">Интералдың сол    шекарасы
/// </param>
/// <param name = 'parMax">Интервалды оң     шекарасы


   /// </param>
   private void MakeData(double[] parValues, out
   double[] parDataPlot, out double[] parDataFunc,
   double parMin, double parMax)
      {
      double delta = (parMax - parMin) / K;
      parDataPlot = new double[ K];
      parDataFunc = new double[ K]; for
      (int i = 0; i < N; i++)
      {
        int j = (int) ((parValues[i] - parMin) /
      delta); if (j >= _K) j = _K - 1; else if (j <0)
      j = 0; parDataPlot[j]++;
   }
   for (int i = 0; i < K; i++) parDataPlot[i] /= N;
   parDataFunc[0] = parDataPlot[0]; for (int i = 1; i
   < K; i++)
      parDataFunc[i] = parDataFunc[i -   1]     +
      parDataPlot[i];
   }
   /// <summary>Статистикалық баға алу /// </summary>
   /// < param name = "parValues">Кездейсоқ сан ///
   </param>
   /// <paramname = "parMx">Математикалық күтілім ///
   </param>
   /// <paramname = "parS2">Түзетілген таңдамалы
   дисперсия</param>
   private void Estimate(double[] parValues,
   out double parMx, out double parDx)
   {
```

```
   double m2 = 0; parMx = 0;
   for (int i = 0; i < N; i++)
   {
     parMx += parValues[i];
     m2 += parValues[i] * parValues[i];
   }
   parMx /= _N; m2 / = _N;
   parS2 = (m2 - parMx * parMx) * N / (N -   1);
}
/// <summary>Негізгі бағдарлама</summary>
/// <param name = "args">Дәлелдер</param>
static void Main(string[] args)
{
   double[] values, dataPlot, dataFunc;
    Mm = 4096;
    A = 5;
   _Y = _Mm - 5;
   _K = 15;
   GenerateData(out values);
   MakeData(values, out dataPlot,out
   dataFunc,0.0,1.0); ... // таралудың жиілік
   гистограммасын мен статистикалық функцияны
   құрydoublemx, dx;
   Estimate(values, out mx, out dx);
   ... // статистикалық көрсеткіштерді
     өңдеу }
   }
}
```

3.3. ЖАЗЫҚТЫҚТА ТАРАЛУЫ

$x_j, x_2, \ldots, x_N$ таңдамасы элементтерінің арасындағы тәуелділікті анықтау үшін жазықтықтағы таралу тестін пайдалануға болады. Осы жазықтықтағы тестті пайдалану кезінде X x_i+j) кординаталарымен нүктелерді салады, мұндағыi = 1... N. Осыдан соң алынған сурет талданады. Егер жазықтықта анық «өрнектер»болса, онда таңдамасындағы сандар тізбегі кездейсоқ болып табылмайды.

Егер жазықтықтағы нүктелердің таралуы бей-берекет болса, онда таңдама элементтерінің арасында тәуелділік болмайды.

Мысал ретінде 3.4 суретте тәуелді (а) және тәуелсіз (б) жалған

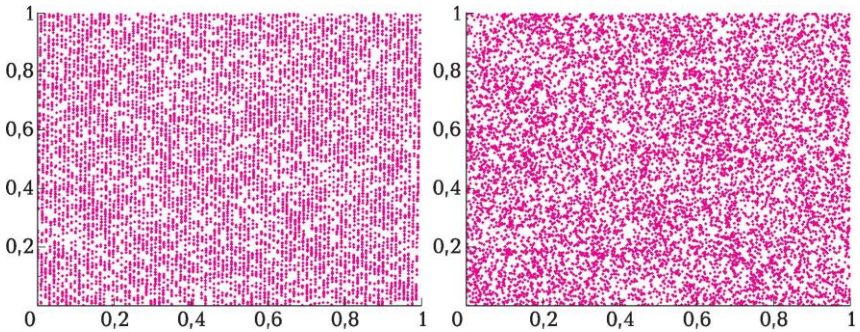
кездейсоқ сандар қалыптастыратын генераторларға арналған жазықтықта таралу көрсетілген. Қарастырылған тексеру әдістерін пайдалану кезінде генератор сапасы жайлы шешімді пайдаланушы қабылдайтындығын байқайық.

Осының арқасында нәтижені түсіндіру әр түрлілігі орын алуы мүмкін. Одан әрі қарастырылған бағалау тесттері (келісім критерийлері) олар сандық сипаттаманы беретіндігімен сипатталады, ол тесттің өткен-өтпегендігін түпкілікті айтуға мүмкіндік береді.

Келісім критрии–бұл $F(x)$таралу функциясы бар белгілі бір таралудаx_x, x_2,..., x_Nбақылау тәуелсіз таңдама болып табылатындығы ресми бағалауда қолданылатын болжамды анықтауға арналған статистикалық критерий. Келісім критерийі келесі нөльдік гипотезаны тексеру үшін пайдаланылады:

H_0: X_t–F(x) тарату функциясы бар тәуелсіз және бірдеу таралған кездейсоқ шама.

«Хи-квадрат» (Пирсон критерийі) критерийі және Колмогоров критерийі сияқты ең жиі қолданылатын келісім критерийлерін қарастырайық.



Сур. 3.4. Тәуелді (а) және тәуелсіз (б) жалған кездейсоқ санды қалыптастыратын генераторларға арналған жазықтықтағы таралу мысалы

3.4. ЖАЗЫҚТЫҚТА ТАРАЛУЫ

Машиналық тәжірибе нәтижесінде алынған N кездейсоқ санның $xj, x_2,..., x_N$ таңдамасы бар делік.

Тағы да алынған мәндер $к$ тобына (интервалға) келтірілген және статистикалық қатар түрінде рәсімделген делік:

$$\begin{array}{cccc} (a_1;a_2) & (a_2;a_3) & \cdots & (a_k;a_{k+1}) \\ \hat{p}_1 & \hat{p}_2 & \cdots & \hat{p}_k \end{array},$$

Мұндағы $pj = m\wedge N$; mj–кездейсоқ X шамасының j-шы интервалға $(aj; aj+j)$ тию саны

Кездейсоқ X шама шынымен де кейбір $f(x)$тығыздық пен таралу заңынан$F(x)$тұратындығы анықтайтын тексеру талап етіледі.

Егер таңдама теориялық таралудан f(x) іске асырылатын болса, таралу тығыздығынf(x) біліп кездейсоқ шаманың j-ші интервалға ($j = 1...$ к) тиюінің күтілетін ықтималдылығын p_tанықтауға болады.

Сонымен бірге

$$p_j = \int_{\alpha_j}^{\alpha_{j+1}} f(x)\,dx = F(\alpha_{j+1}) - F(\alpha_j).$$

Тең ықтимал таралуы және интервалдың бірдей ұзындығы кезінде

$$P_1 = P_2 = \ldots = P_k = 1/k.$$

Теориялық ықтималдылық p_t пен бақыланатын жиілік $P>j$арасындағы айырма шамасы ретінде олардың бірқатар салмақпен ауытқуының шаршы соммасы деп есептеуді, яғни К шамасы

$$U = \sum_{j=1}^{k} c_j(\hat{p}_j - p_j)^2.$$

Пирсон көрсеткендей, егер

болса, онда Nүлкен болған кезде кездейсоқ шаманың таралу заңы келесі қасиетке ие:

$$c_j = \frac{N}{p_j},$$

- f(x) түрі менN тәжірибе санына іс жүзінде тәуелді емес;

■ κ тобының саны ғана анықталынады және N артқан кезде χ^2 (хи-квадрат) таралуына жақындайды.

Сонымен қатар тәжірибелік және теориялық таралу тығыздығы арасындағы айырмашылық өлшемі төмендегідей белгіленеді

$$\chi^2 = N\sum_{j=1}^{k}\frac{(\hat{p}_j - p_j)^2}{p_j} = \sum_{j=1}^{k}\frac{(m_j - Np_j)^2}{Np_j}. \qquad (3.2)$$

N_{pj}–j-ші интервалға түсетін x_t шамасының күтілетін саны болғандықтан, онда теориялық таралу $f(x)$мен деректер арасындағы жақсы сәйкестілік кезінде χ^2 критерийінің статистикасы шамалы болады. Бұл жағдайда H_0гипотезасы қабылданады. Егер X^2статистикасының мәні тым үлкен болса, онда H_0гипотезасы қабылданбайды.

χ^2 таралуы p_j жиілігіне қойылатын κ тобы санынан тәуелсіз жағдай санын алуға тең таралудың «бостандық деңгейінің» саны деп аталатын r көрсеткішіне тәуелді. Тең ықтимал таралуды тексеру кезінде тек бір жағдайды ғана ескеру керектігін айта кетейік:

$$\sum_{j=1}^{k}\hat{p}_j = 1.$$

Бұл балық жағдайда қойылатын шектеу жағдайы.

«Хи-шаршы» критерийі үшін арнайы кестелер құрылған, оны пайдаланып, әр r бос деңгейі санының мәні мен маңыздылық деңгейі αүшін критикалық мән X_{r1} _aтабуға болады. Осындай кестенің қысқартылған нұсқасы 3.1 кестеде көрсетілген.

Теориялық және статистикалық таралу келісімділігін бағалау χ^2 критерийінің қолдану сызбасы келесіге сүйенеді.

1. χ^2 формула (3.2) бойынша айырмашылық өлшемі анықталынады. Бірдей интервал кезіндегі тең ықтимал таралу үшін

$$P_1 = P_2 = \ldots = P_k = 1/k.$$

Практикада әдетте төмендегідей қабылданады

$$k = 20...50,\ N = (10^2...\ 10^3)k.$$

Сонымен қатар әр топқа 10...20 кем емес мән түскені қалаулы, ол таралудың статистикалық тығыздығын айтарлықтай дәр құруды қамтамасыз етеді.

Кесте 3.1. «Хи-шаршы»критерийі үшін $\chi^2_{r,\alpha-1}$ критикалық мән

r	1-α									
	0,990	0,975	0,950	0,900	0,800	0,700	0,600	0,500	0,400	0,300
5	0,55	0,83	1,15	1,61	2,34	3,00	3,66	4,35	5,13	6,06
6	0,87	1,24	1,64	2,20	3,07	3,83	4,57	5,35	6,21	7,23
7	1,24	1,69	2,17	2,83	3,82	4,67	5,49	6,35	7,28	8,38
8	1,65	2,18	2,73	3,49	4,59	5,53	6,42	7,34	8,35	9,52
9	2,09	2,70	3,33	4,17	5,38	6,39	7,36	8,34	9,41	10,66
10	2,56	3,25	3,94	4,87	6,18	7,27	8,30	9,34	10,47	11,78
11	3,05	3,82	4,57	5,58	6,99	8,15	9,24	10,34	11,53	12,90
12	3,57	4,40	5,23	6,30	7,81	9,03	10,18	11,34	12,58	14,01
13	4,11	5,01	5,89	7,04	8,63	9,93	11,13	12,34	13,64	15,12
14	4,66	5,63	6,57	7,79	9,47	10,82	12,08	13,34	14,69	16,22
15	5,23	6,26	7,26	8,55	10,31	11,72	13,03	14,34	15,73	17,32

3.1 кесте жалғасы

r	1-α									
	0,990	0,975	0,950	0,900	0,800	0,700	0,600	0,500	0,400	0,300
16	5,81	6,91	7,96	9,31	11,15	12,62	13,98	15,34	16,78	18,42
17	6,41	7,56	8,67	10,09	12,00	13,53	14,94	16,34	17,82	19,51
18	7,01	8,23	9,39	10,86	12,86	14,44	15,89	17,34	18,87	20,60
19	7,63	8,91	10,12	11,65	13,72	15,35	16,85	18,34	19,91	21,69
20	8,26	9,59	10,85	12,44	14,58	16,27	17,81	19,34	20,95	22,77
25	11,52	13,12	14,61	16,47	18,94	20,87	22,62	24,34	26,14	28,17
50	29,71	32,36	34,76	37,69	41,45	44,31	46,86	49,33	51,89	54,72
100	70,06	74,22	77,93	82,36	87,95	92,13	95,81	99,33	102,95	106,91

2. Бостандық дәрежесінің саны есептелінеді $r = k- m-1$, мұндағы m–топтасқан таңдама бойынша анықталатын таралу көрсеткіштерінің саны. Тең ықтимал таралудың біркелкілігін тексерген жағдайда $г = к -1$.

3. Таңдалған α деңгейі мен бостандық дәрежесінің r саны үшін критикалық $\chi^2_{j\_a}$ мәні орнатылады. Егер алынған χ^2 мәні $\chi^2_{j\_a}$ асып түссе, онда H_0 гипотезаны қабылдамайды, болмаса қабылданады.

Алынған χ^2 мәні $\chi^2_{j\_a}$ біршама кіші болса, онда генератор тығыздық функциясы $f(x)$ бар кездейсоқ таралу талаптарын қанағаттандырмайтынын ерекше атап өткен жөн, өйткені бақыланатын p_t жиілігі теориялық p_t тым жақын және кездейсоқ ретінде қарастырыла алмайды.

«Хи-шаршы» критерийін қолданған кезде негізгі мәселені балық мүмкін таралу мен барлық таңдама көлемі үшін критерийдің жоғары қуаттылығы сақталатындай сан мен интервалдар мөлшерін таңдау тудырады.

Әдетте интервалдар мөлшері $p_j = p_2 = ... = p_k$ (тең ықтимал тәсіл) шарттары орындалатындай етіп таңдау ұсынылады. Бірақ кейбір жағдайларда бұған қол жеткізу қиын. Лайық нәтижеге төменде сипатталған шарттарды орындау кезінде қол жеткізуге болады.

$Npj < 5$ болып табылатын $a = \min Npj$ және $y(5)$ интервал саны болсын делік. Онда интервал саны мен ұзындығын $к > 3$, $a > 5y(5)/k$ шарттары орындалатындай таңдау қалаулы.

C# тіліндегі Пирсон критериі үшін X есептеу функциясының кесіндісі келесідей болады:

```
   /// <summary> Пирсон критериі үшін
   /// есептеу функциясы xi 2</summary>
   /// <param name = "hst"> Кездейсоқ шаманың
    /// интервалға түсу саны</param>
   /// <param name = "pt"> Кездейсоқ шаманың
интервалға
    /// түсуінің теориялық ықтималдылығы //
</param>
   ///              <param name ="k"> Ұсақтау
интервалының саны
   /// </param>
   /// <param name = "n"> Таңдама көлемі
</param>
   /// <returns>Есептеу нәтижесіxi2 //
</returns>
```

```
  double Xi2(double[] hst, double[] pt, byte k,
long n)
  {
  double xi = 0.0;
  for (int i = 0; i < k; i++)
xi += Math. Pow(hst[i] - n * pt[i], 2) / (n*
pt[i]);
returnxi;
  }
```

3.5. КОЛМОГОРОВ КРИТЕРИІ

Ықтималдылық тығыздығымен деректер гистограммасын бір-бірімен салыстыратын «хи-шаршы» критериінің кемшілігі интервалдалды таңдау қиындығында жатыр.

Болжанатын таралу функциясымен эмпирикалық таралу функциясын салыстыруға арналған Колмогоров критериі айтылған кемшіліктер бос, өйткені ол үшін деректерді топтастырудың қажеті жоқ. Бұл критерий N таңдама кез-келген көлемді болған кезде де айқын.

Теориялық және статистикалық таралу арасындағы айырма өлшемі ретінде А.Н. Колмогоров таралудың статистикалық функциясы F(x) мен сай теориялық арасындағы айырма модульінің максималды мәнін пайдалануды ұсынды:

$$D = \max \left| \hat{F}(x) - F(x) \right|.$$

А. Н. Колмогоров үздіксіз кездейсоқ шаманың X таралу функциясы $F(x)$қандай болса да, тәжірибе Nсаны үздіксіз өскен кезде теңсіздік ықтималдылығы

$$D\sqrt{N} \geq \lambda$$

шекке жететіндігін көрсетті

$$P(\lambda) = 1 - \sum (-1)^k e^{-2k^2\lambda^2}. \qquad (3.3)$$

Формула (3.3) бойынша есептелінген ықтималдылық мәні P(λ), 3.2 суретте көрсетілген.

А.Н. Колмогоров критериін қолдану сызбасы келесі сатыларды орындауға апарады.

1. Статистикалық таралу F(x) мен теориялық таралу функциясы $F(x)$ құрылады және Колмогоров критериі болып табылатын олардың арасындағы Dмодулі максимумының айрмасы анықталынады (сур. 3.5).

Кесте 3.2. Ықтималдылық мәнінің кестесі P(λ)

λ	P(λ)	λ	P(λ)	λ	P(λ)
0,0	1,000	0,7	0,711	1,4	0,040
0,1	1,000	0,8	0,544	1,5	0,022
0,2	1,000	0,9	0,393	1,6	0,012
0,3	1,000	1,0	0,270	1,7	0,006
0,4	0,997	1,1	0,178	1,8	0,003
0,5	0,964	1,2	0,112	1,9	0,002
0,6	0,864	1,3	0,068	2,0	0,001

Статистикалық таралу функциясы F(x) формула бойынша анықталады

$$\hat{F}(x) = \frac{M(x)}{N},$$

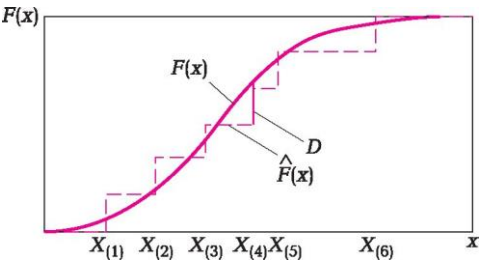
Мұндағы M(x) –х асып түспейтін таңдамадан X_i шама саны. Бұл функция X өсу қатарымен сұрыпталған яғни $^{X}(1) < ^{X}(2) < ... < ^{X}(N)$, $F(X(i = 1, 2, ..., N$ үшін Д = i/N, мұндағы X) ретінде оңай табылуы мүмкінX.

Критерий статистикасын төмендегідей анықтауға болады

$$D = \max\{D^+, D^-\},$$

мұндағы

$$D^+ = \max_{1 \le i \le n}\left\{\frac{i}{n} - F(X_{(i)})\right\}, \quad D^- = \max_{1 \le i \le n}\left\{F(X_{(i)}) - \frac{i-1}{n}\right\}.$$



Сур. 3.5. Теориялық және статистикалық таралу функцияларының арасындағы максимальды айырманы табу

2. Есептегіш шама $\lambda = D\sqrt{N}$.
3. 3.2 кесте бойынша P(λ) ықтималдылығын табады.
4. Егер P(λ) ықтималдылығы айтарлықтай үлкен болса, онда X таралу шамасы $F(x)$ заңыбойынша таралған деп есептейді. Кіші мәнде X шамасының таралуы F(x) таралу заңынан алшақ деп мойындаған жөн.

А.Н. Колмогоров критериі үшін C# тілінде X есептеу функциясының кесіндісі келесідей көрініс болады:

```
/// <summary>Колмогоров критериі үшін
/// Lambdaесептеу функциясы</summary>
/// <param name = "x">Таңдаманың өсуі ретімен
/// сұрыпталған</param>
/// <param name = "n">Таңдама көлемі</param>
/// <returns>ЕсептелгенLambdamәні//
</returns>
double Lambda(double[] x, int n)
{
   double dMax = 0.0;
   for (int i = 0; i <n; i++)
   {
     double dp = Math. Abs((double) (i +   1)
                                          / n -
     Ft(x[i]));
     double dm = Math. Abs(Ft(x[i]) - (double)
     i/ n); // Мұнда Ft –теориялық таралу
     функциясы if (dp >dMax) dMax = dp; if (dm
     >dMax) dMax = dm;
   }
   return dMax * Math.Sqrt(n);
```

3.6. ТОПТАМА КРИТЕРИІ

Топтама критериі кездейсоқ сандардың тәуелсіздігін тексеру үшін пайдаланылады.

xj,x_2,...,x_N–таралу функциясыF(x) бар кездейсоқ санның тізбегі делік.

Осы кездейсоқ сан ауданын екі жекелеген S_0 және S_1 қосалқы жиытыққа бөлейік.

Осы таралуды пайдаланып 0 және 1 мәндері бар жаңа Y кездейсоқ шаманы құрайық:

$$Y = \begin{cases} 0, \text{ егер } x \in S_0; \\ 1, \text{ егер } x \in S_1. \end{cases} \quad (3.4)$$

$P\ (7 = 0) = p$болса делік, онда$P\ (7 = 1) = 1 - p$.

Түрлендіруді (3.4) орындау нәтижесінде, мысалы, Y кездейсоқ шама мәнінің осындай тізбегін аламыз:

001011101000 ... 0110001.

Топтама деп бірдей элементтерден тұратын тізбектің кез-келген кесіндісін айтады. Демек, осы тізбекте топтама бар

00, 1, 0, 111, 0, 1, 000, ..., 0, 11, 000, 1.

Осындай топтаманың Rсаны бірқатар таралуы мен сандық сипаттамасы бар кездейсоқ шама болып табылады:

$$M(R) = 2Np(1-p) + p^2 + (1-p)^2; \quad (3.5)$$

$$D(R) = 4Np(1-p)[1-3p(1-p)] - 2p(1-p)[3-10p(1-p)]. \quad (3.6)$$

Муавр-Лаплас теоремасының негізінде Nүлкен болған кезде Rтоптама санының таралуы (3.5) және (3.6) формулалармен анықталынатын көрсеткіштері бар нормалды таралумен жуықталады.

ТоптаманыңRсанының негізінде P = 1 -α (α маңыздылығы деңгейінде) сенімділігімен $x_1, x_2, \ldots, x_N$ кездейсоқ шаманың тәуелсіздігі жайлы гипотезаны тексеру екі шекті мән R_K және $R_в$ анықтауға келеді, осылайша теңдік орын алады

$$P(R < R_н) = P(R > R_в) = \frac{\alpha}{2}.$$

Егер бақыланатын топтама саны R_Kаз немесеR_mкөп болса, онда кездейсоқ санның $x_1, x_2, \ldots, x_N$тәуелсіздігі жайлы негізгі гипотеза қабылданбайды. Болмаған жағдайда $x_1, x_2, \ldots, x_N$сандарын тәуелсіз деп есептеуға барлық негіз бар.

Топтама санының тестін пайдалану (кездейсоқ санның тәуелсіздігін тексеру үшін) келесі сатыны іске асыруға әкеп соғады.

1. $x1, x2, \ldots, xN$тізбегінен $y_1, y_2, \ldots, y_N$тізбегі қалыптастырылады және бақыланатын топтаманың Rсаны саналады. Бөлуші элемент ретінде бірқатар $p\ (0<p<1)$ мәні таңталынады.

2. (3.5) және (3.6) формулалар бойынша $M\ (R)$ математикалық күтілу, $D(R)$ дисперсия және и $\sigma = \sqrt{D(R)}$орташа шаршы ауытқу анықталынады.

Кесте3.3. Стандартты нормальды таралу t_p квантилия мәні		
Сенімділікβ	Маңыздылық деңгейіα	$t_ß$
0,90	0,10	1,65
0,95	0,05	1,96
0,96	0,04	2,06
0,07	0,03	2,18
0,98	0,02	2,33
0,99	0,01	2,58
0,999	0,001	3,30

3. Маңыздылық деңгейінде α (сенімділік үшін **β**= 1 -**α**) 3.3 кесте бойынша стандартты нормальды таралудың £р квантилия мәні табылады.

4. Шекті мәндер анықталынады

$$R_н = M(R) - t_\beta\sigma(R);\ R_в = M(R) + t_\beta\sigma(R).$$

5. Егер $R_H < R < R_B$ болса, онда x_1, x_2,…, x_N кездейсоқ санның тәуелсіздігі дайлы гипотезаға күмән тудыруға негіз жоқ.

C# тілінде топтама критериі үшін*R*есептеу бағдарламасының кесіндісі төмендегідей бейнеде болады:

```
/// <summary>Топтама критериіне арналған
/// Rесептеуфункциясы </summary>
/// <paramname= "x">тестіленетін генератормен
/// түрлендірілген кездейсоқ санның
ауқымы</param>
/// <paramname= "n">Таңдама көлемі</param>
/// <paramname= "p">Бөлуші элемент ///
</param>
/// <returns>R есептелінген шама</returns>int
getR(double[] x, int n, double p)
{
  bool y1,y2;
  int R = 0;
  // if (x[0] тізбектің 0-ші элементін
  анықтаймыз  <p) y1= false;
  elsey1= true;{
    // if(x[i]тізбектің 1-ші элементін
    анықтаймыз <p)y2 = false;
```

```
        else y2 = true;
        if (y1 != y2) R++; // R санын
        арттырамызy1 = y2;
    }
    returnR;
}
```

Топтама критерийінің әр алуандылығы ретінде топтама тестінің ұзындығын есептеуге болады. Бұл жағдайда кездейсоқ тізбекте y_t, y_2,..., y_N бір түрдің элементтер топтамасын бөледі. Тізбектегі y_t, y_2,..., y_N «нөльдердің» Vтоптамасының ұзындығы мен «бірліктердің» Zтоптамасының ұзындығы бірқатар таралуы бар $P(V = v)$ және $P(Z = z)$ кездейсоқ шама болып табылады. Мысалы үшін алдын келтірілген тізбектің өңделуін елестетейік:

Осындай өңдеу нәтижесінде екі VJ, V_2, ..., V_mжәне Z_1,z2,..., z_n. кездейсоқ тізбек қалыптастырылады.

0 және 1 кездейсоқ сан тәуелсіз және интервалда [0; 1) біркелкі таралған xJ,x2,...,x_Nтізбегінен қалыптасқандықтан, ұзындығы vнөль топтамасының пайда болуының теориялық ықтималдылығы геометриялық таралумен (топтама үнемі с 0басталатындығын

$$P(V = v) = (1 - p)^{v-1} p,$$

ескергенде) анықталады:

мұндағыp–1 пайда болуы ықтималдылығы.

Онда топтамадағы математикалық күтілу мен нөл санының дисперсиясы келесі формула бойынша анықталады:

$$M(V) = \frac{1-p}{p} + 1; \qquad (3.7)$$

$$D(V) = \frac{1-p}{p^2}. \qquad (3.8)$$

xj, x_2,..., x_Nтізбегін өңдеу процесінде топтамадағы нөлдің орташа санын бағалау үшін нөльдіңN_0 санын және нөлдің K_0топтамасының санын анықтау қажет. Онда топтамадағы нөлдің орташа санының бағасы төмендегідей іске асырылады:

$$M_v = \widehat{M}(V) = \frac{N_0}{K_0}. \qquad (3.9)$$

Нөл топтамасы ұзындығының тестін қолдану келесі сатыда іске асыруға әкеп соғады.

1. xj,$x2$,...,x_Nтізбегіен y_j, y_2,…, y_Nтізбегі қалыптастырылады және нөльдіңN_0саны мен нөль топтамасыныңK_0саны есептелінеді.

2. (3.7), (3.8) және (3.9) формулалары бойынша математикалық күтілу$M\ (V)$, дисперсия $D(V)$ және нөл топтамасының M_vорташа ұзындығын бағалау анықталынады.

3. 3.3 кесте бойыншаа = 1 - р маңыздылық деңгейінде £р орналасқан.

4. Шекті мән анықталынады

$$V_н = M(V) - t_\beta\sqrt{\frac{D(V)}{K_0}};\ V_в = M(V) + t_\beta\sqrt{\frac{D(V)}{K_0}}.$$

5. Егер$V_H < M_v < V_B$болса, онда кездейсоқ шама x_1, x_2,..., x_N тәуелсіздігі жайлы гипотезаға күмән келтіруге негіз жоқ.

Егер бірлік топтамасы ұзындығының тесті пайдаланылатын болса, онда yj, y2,..., yNтізбегін өңдеу процесінде z_1, z_2,…, z_nбірлігінің топтамасы бөлінеді және мәні есептелінеді

$$M(Z) = \frac{p}{1-p};\ D(Z) = \frac{p}{(1-p)^2};\ M_z = \widehat{M}(Z) = \frac{N_1}{K_1},$$

Мұндағы N_1 – y_1, y_2,..., y_Nтізбегіндегі бірлік саны; K_1–бірлік топтамасының саны.

Жүргізілген талдау нәтижесінде бастапқы кездейсоқ тізбектің тәуелсіздігі жайлы гипотеза қабылданады, егер

$$M(Z) - t_\beta\sqrt{\frac{D(Z)}{K_1}} \le M_z \le M(Z) + t_\beta\sqrt{\frac{D(Z)}{K_1}}.$$

Одан әрі нөл топтамасы ұзындығының тестінде қолданылатын у ауқымындағы нөль топтамасының K_0 санын анықтайтын бағдарлама кесіндісі ұсынылған.

```
//0-ден 1-ге өту санын дөңгелектейміз
K0 = 0;
//0-ден 1-ге өту санын есептейміз
for (inti= 1; i<n; i++)
if((y[i - 1]  == 0)     &&
     (y[i]==1))K0++;
//егер соңғы топтама бар болса, оны
ескереміз if(y[n- 1]==0)K0++;
```

Өңдеу нәтижесінде алынған кездейсоқ мәндер нормалы таралумен жуықталған жағдайда, топтама тізбегін бөлуге негізделген критерий Муавр-Лаплас теоремасының шығуына негізделеді. Ұқсас әдісті келесі критериде де қолдануға болады.

3.7. ЖАНАМА БЕЛГІЛЕР БОЙЫНША БІРКЕЛКІЛІКТІ ТЕКСЕРУ

Қалыптасқан нөмерлердің тізбектілігі xj, x_2,..., x_N екі тізбекке бөлінеді:

$$xj, x3, x5,...$$

$$x2, x4, x6,...$$

Кейін келесі тәжірибе жүргізіледі. Егер шарттар орындалатын болса

$$x_{2i-1}^2 + x_{2i}^2 < 1; i = 1, 2, ..., \quad (3.10)$$

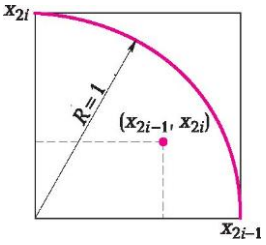
онда бірқатар С уақиғасының келуі тіркеледі және уақиға есептегішіне бірлік қосылады. N/2 тәжірибеден кейінNсаны түрленген кезде, есептегіште$N_C < N/2$ саны болады.

Геометриялық шарт (3.10) нүкте R= 1 радиусты шеңбердің төрттен бірінің ішінде орналасқандығын білдіреді, ол 3.6 суретте бейделенген.

Жалпы жағдайда нүкте(x_{2i-1}, x_{2i}) бірлік шаршының үнемі ішіне түседі, сондықтан үлкен Nжиілікте бастапқы тізбекті біркелкі тарауы кезінде

$$\hat{p}_k = \frac{2N_C}{N}$$

осы нүктенің шеңбердің төрттен біріндегі осы нүктеге түсу ықтималдылығына әкеп соғады.



Сур. 3.6. Генераторды жанама белгі бойынша тексеру әдісін бейнелеу

$$p_k = S_{1/4\text{круга}} / S_{\text{квадрата}} = (\pi R^2/4)/R^2 = \pi/4.$$

Бастапқы тізбектің біркелкі таралуы жайлы а = 1 – маңыздылық деңгейі бар (fp мәнін анықтайтын) гипотезаны, егер төмендегі теңсіздік орындалатын болса, қабылдайды

$$\frac{\pi}{4} - t_\beta\sqrt{\frac{\pi(4-\pi)}{8N}} \le \frac{2N_c}{N} \le \frac{\pi}{4} + t_\beta\sqrt{\frac{\pi(4-\pi)}{8N}}.$$

Бағдарламаша C# тілінде уақыға C басталуыменN_C санын есептеу төмендегідей болады:

```
/// <summary>Жанама белгілері бойынша
біркелкілікті
/// тексеру үшін Nc есептеу функциясы //
</summary>
/// <param name = "x">тестіленетін генератормен
/// түрлендірілген кездейсоқ сан ауқымы</param>
/// <paramname= "n">Таңдама көлемі</param>
/// <returns>ЕсептелінгенR мәні</returns> int
getNc(double[] x, int n)
{
   // xintNc = 0 ауқымындағы жұп мән санын
   анықтаймыз;
   for (int i = 0; i < n /  2; i++)
     if (Math. Pow(x[2*i],2) + Math. Pow(x[2*i +
     1],2)<1) Nc++;
   return Nc;
}
```

3.8. БІРІКТІРІЛГЕН ТЕСТТЕР

Біріктірілген тесттер негізі кейбір уақиғалардың бірігуінен тұратын арнайы критериге негізделген. Қайта қалыптасқан кездейсоқ шама үшін бірқатар тарау функциясы анықталады, ол статикалық таралу функциясымн салыстырылады. Оладың

арасындағы келісім, әдетте, χ2 критериінің көмегімен зерттелінеді.

3.8.1. Покер-тест

Бұл тест шығарылатын тізбек санында санның пайда болу кездейсоқтығын анықтай үшін арнайы құрастырылған. $xj, x_2, \ldots, x_N$– $F(x)$ таралу функциясы бар кездейсоқ шама тізбегі делік. Кездейсоқ шама X мәнінің ауданын нүктелер $a_0 < a_j < \ldots < a_{k-1} < a_k$ көмегімен κтең ықтимал интервалға бөлейік. Егер X шынымен де $F(x)$ таралу

$$P(a_{j-1} \le X < a_j) = \frac{1}{k}.$$

функциясы болатын болса, онда әр интервал үшін$[^a j\text{-}j;\ ^a j)$ болады

$x_i\ e\ [aj; aj+_1)$, $j = 0,1,..., k$ -1 үшін $y_i = j$ пайдаланып, бірдей ықтималдылықпен 0,1,..., κ -1 мәнінің біреуін қабылдайтын жаңа $y,,$ i= 1,2,..., кездейсоқ сан тізбегін аламыз.

Одан әрі барлық $y_{1e}, y_2, \ldots$ тізбекті (У1, *У 2У 5*)·(*у*6, *УУ*10, ...)... бесеуге болуге болады.

Әр түрлі цифр құрамымен ерекшеленетін бестіктің жеті класы бар:

abcde–бестіктегі барлық цифрлар әр түрлі;

aabcd–екі цифр бірдей, қалғандары әр түрлі;

aabbc–цифрдың екі жұбы бірдей;

aaabc–үш цифр бірдей, екеуі әр түрлі;

aaabb–үш цифр бірдей және екеуі бірдей;

aaaab–төрт цифр бірдей;

aaaaa–барлық цифрлар бірдей.

у1, у2,...тізбегіндегі цифрдың 0,1,..., к -1әрқайсысы бірдей ықтималдылықпен пайда болады және осы тізбектің жеке мүшелері тәуелсіз деп болжап, әр түрлі *P(abcde), P(aabcd), P(aabbc), P(aaabc), P(aaabb), P(aaaab), P(aaaaa)* бестіктің таралу ықтималдылығын анықтауға болады.Осы ықтималдылықтың мәні екі жиі қолданылатын к =2, 8 және10 үшін 3.4 кестеде көрсетілген.

Бақыланатын таралумен әр түрлі типті бестіктің таралу келісімін χ^2 критериінің көмегімен тексеріледі.

Кесте 3.4. Покер-тестке арналған уақиға ықтималдылығы			
Ықтималдылық	k =2	k =8	k =10
P(abcde)	–	0,20518	0,3024
P(aabcd)	–	0,51270	0,5040
P(aabbc)	–	0,15381	0,1080
P(aaabc)	–	0,10254	0,0720
P(aaabb)	0,6250	0,01709	0,0090
P(aaaab)	0,3125	0,00854	0,0045
P(aaaaa)	0,0625	0,00024	0,0001

Мысал үшін одан әрі біркелкі тараған сан генераторын тексеру кезінде бестіктің әр түрлі класының пайда болу ықтималдылығын анықтау үшін бағдарлама кесіндісінің нұсқасы ұсынылған.

```
// x-тестіленетін генератормен түрлендірілген
// кездейсоқ сан ауқымы
// n-x ауқымындағы кездейсоқ сан мөлшері
// K-y тізбегімен алынатын тең ықтималды
// интервал мөлшері
inti, j, m;
int []numN = new int[5];
int []repN = new int[5];
int []y = new int[n];
for (i =  0; i <       n; i++) y[i]   =
(int)(x[i] * K);
//бестік санын анықтаймыз intN5 = n / 5;
// әр түрлі класты бестік санын
анықтаймызintNabcde= 0, Naabcd = 0, Naabbc = 0;
intNaaabc = 0, Naaabb = 0, Naaaab = 0, Naaaaa =
0; for (i =0; i <     N5 -  1; i++)
{
   // бестіктегі j-ші санның қайталану
   мөлшері
   // сақталатын ауқымдын нөльдеймізfor (j  =
   0;          j  < 5;  j++) numN[j] = 0;
   for (j  = 0; j     < 5;  j++)
     for (m = 0; m < 5; m++)
```

```
        if(y[j + i * 5] == y[m + i * 5])
        numN[j]++;
    // jқайталанатын сан қанша рет кездескендігі
     // сақталатын ауқымды нөльдеймізfor      (j
        = 0; j <5; j++)    repN[j] =0;
    for (j = 0;j < 5; j++) repN[numN[j]- 1]++;
    // бестік класын
    анықтаймызif
    (repN[0]==5) Nabcde++;
    else
      if (repN[0]==3) Naabcd++;
      else
        if (repN[1]==4)
        Naabbc++; else
          if ((repN[ 1]== 2)  && (repN[2]==
                          3))
          Naaabb++;
          else
            if (repN[3]==4) Naaaab++;
            else
              if (repN[4]==5)
              Naaaaa++; else Naaabc++;
}
// Әр түрлі класты бестіктің пайда болуы
//              (double)Nabcd
ықтималдылығын e              / N5;
double Paabcd = (double)Naabc / N5;
double Paabbc = (double)Naabb / N5;
double Paaabc = (double)Naaab / N5;
double Paaabb = (double)Naaab / N5;
double Paaaab = (double)Naaaa / N5;
double Paaaaa = (double)Naaaa / N5;
```

3.8.1. Коллекция жиюшы критериі

Тест өз коллекциясын толықтырушы коллекция жиюшы қызметіне ұқсас үрдісті іске асырады. Бұл критериді қолдану кезінде покер-тестті қолдану кезіндегідей түзілетін уі, у 2,...тізбегі онда 0,1,..., к -1 санының барлық к пайда болғанға дейін ұзақ байқалады. R1– бақыланатын тізбек кесіндісінің ұзындығы делік. Осыдан соң тізбектің қалған бөлігі үшін «коллекциялау» үрдісі қайта қосылады, оның нәтижесінде R2 тізбегінің жаңа ұзындығы алынады.¬ Бұл

әрекет уj, у2,... барлық тізбегі өңделгенге дейін жалғасады. Нәтижесінде R1, R2, ... кездейсоқ мәнінің тізбегі алынады. R кездейсоқ шамасы төмендегідей анықталатын таралу ықтималдылығына ие

$$P(R = r) = \frac{1}{k^{r-1}} \sum_{j=0}^{k-2} (-1)^j C_{k-1}^j (k-1-j)^{r-1}, \qquad (3.11)$$

мұндағы$r = k, k+1,...$; Cm–m бойынша nэлемент үйлесімінің саны;

$$C_n^m = \frac{n!}{m!(n-m)!}$$

Таралу келісіміне (3.11) бақыланатын бөлінумен R_1, R_2, ... таралуы бар таралу келісімі (3.11) χ^2 Критериінің көмегімен зерттелінеді.

Бақыланатын тізбек кесіндісінің Rұзындығын анықтау үшін бағдарлама кесіндісін келтірейік.

```
   // x –тестіленетін генератормен түрлендірілген
   // кездейсоқ санның ауқымы
   // n –x ауқымындағы кездейсоқ сан мөлшері
   // K –тең ықтимал интервалдардың мөлшеріint i,
у;
   // int R = 0 тізбегінің ағымдағы ұзындығын
нөльдейміз;
   // пайда болып жатқан цифрды белгілейтін
   //ауқымды құрамыз және нөльдейміз
   int []numN= newint[K];
   for (i = 0; i <K; i++) numN[i] = 0;
   //пайда болған санның есептегіш санын
нөльдеймізintM= 0;
   while ((M     != K) && (R<n))
   {
   // у тізбегінің элементін аламызу = (int)(x[R]
*     K);
   //егер мұндай элемент болмасаif (numN[y] == 0)
   {  // она оны ауқымда белгілейміз
   numN[y] = 1;
   //және элемент санының есептегішін артырамыз
```

```
M++;
};
// тізбектің келесі элементіне өтеміз
R++;
}
```

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. Жилік гистограммасы дегеніміз не? Оны құру әдістері қандай?
2. Статистикалық таралу функциясы дегеніміз не? Оны құрудың әдістері қандай?
3. Кездейсоқ шаманың қандай негізгі сандық сипаттамаларын білесіз? Белгілі таңдама бойынша олардың мәнін қалай бағалауға болады?
4. Жазықтықта таралу тесті нені тексеруге мүмкіндік береді? Ол неге негізделген?
5. Жалған кездейсоқ сан датчиктерін тексеру критериі неге керек?
6. Пирсонның $\chi 2$ критериінің мәні неде?
7. Қандай шартты орындаған кезде Пирсонның $\chi 2$ критерийін қолдану мүмкін?
8. $\chi 2$ критериі үшін бостандың дәрежесінің саны қалай анықталады?
9. Колмогоров критериі неге негізделеді?
10. Кездейсоқ санның датчигімен қалыптастырылатын тәуелсіз жалған кездейсоқ шаманы кандай критерий көмегімен тексеруге болады? Осы критериі неден тұрады?
11. Түрлендіретін тізбекте цифрдың кездейсоқтығын қандай критерий көмегімен тексеруге болады?

БЕЛГІЛЕНГЕН ТАРАТУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ МОДЕЛЬДЕУ ӘДІСТЕРІ

4.1. КЕРІ ФУНКЦИЯ ӘДІСТЕРІ

Стохастикалық жүйенің имитациялық модельді құру үрдісінде кездейсоқ үрдістерді немесе кездейсоқ әсер тапсырмаларын бірнеше жүйеге имитациялау қажеттілігі туындайды. Осы кездейсоқ шаманы тарату заңы бірқатар жағдайда біркелкіден ерекшеленеді. Осындай жағдайлар белгіленген бірнеше тарату заңы бар кездейсоқ шаманы бағдарламалық іске асыру қажеттілігін анықтайды. Сонымен қатар, әдетте, дәлділігі, тиімділігі мен күрделілігімен ерекшеленетін қажетті таралуы бар кездейсоқ санды түрлендірудің бірнеше ұқсас нұсқалары бар. *Дәлділік* түрленетін кездейсоқ шаманың талап етілгенге қаншалықты дәл тарағандығын анықтайды. *Тиімділік* талап етілген жады көлемі мен жылдам әрекетімен, ал *күрделілік* – әдісті түсіну және іске асыру күрделілігімен қамтамасыз етіледі.

Талап етілген таралу заңы бар кездейсоқ шаманың модельдеу тапсырмасын шешу әдістерін қарастырайық.

Таралу тығыздығы f(x) бар үздіксіз кездейсоқ X шаманы модельдеу талап етілсін делік. Тапсырма берілген таралуы бар кездейсоқ санды біркелкі таралудың кездейсоқ санымен байланысатын бірқатар функцияның көмегімен интервалда [0; 1) біркелкі тараған кездейсоқ сан тізбегін модельдеу және $r_1, r_2, ..., r_n$ кездейсоқ саны тізбегін $x_j, x_2,..., x_n$ тізбегіне түрлендіру жолымен шешеді. Түрлену (4.1) әр түрлі әдістермен орындалуы мүмкін.

$$x = \Psi(r), \tag{4.1}$$

Кері функция әдісі белгіленген таралу функциясымен $F(x)$ кездейсоқ шаманы түрлендіру үшін қолданылады.

Осы әдіс келесі леммаға негізделген.

Лемма. Егер кездейсоқ шаманың £ таралу тығыздығы f(£) болса, онда кездейсоқ шаманың $\gamma = \int_{-\infty}^{\xi} f(x)dx = F(\xi)$ интервалда [0; 1) біркелкі таралу заңына ие.

Берілген леммадан таралу функциясы бар $F(x)$ кездейсоқ X шаманыңх мәнң теңдеудің шешімі болып табылуы тиіс

$$F(x) = r,$$

мұндағыr– интервалда [0; 1) біркелкі тараған кездейсоқ R шамасының мәні. Осылайша

$$x = F^{-1}(r), \qquad (4.2)$$

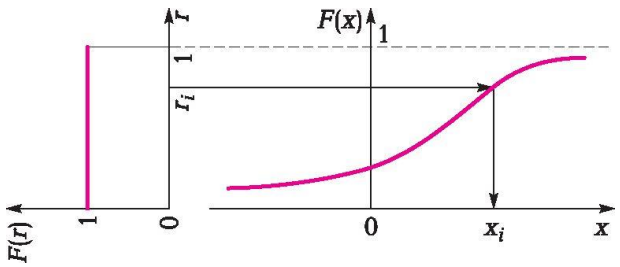
Көріністегі (4.2) жоғарғы индекс «-1» кері функцияның белгісі болып табылатындығын айта кетейік.

Көрнекілік үшін кері функция әдісімен кездейсоқ шаманы алуды графикалық кескіндеуге болады (сур. 4.1). Егер ордината осі бойынша 0-ден 1 дейінгі диапазонда кездейсоқ r_1санын алып және олар үшін стрелкамен көрсетілгендей абсциса осінде сәйкес санды тапса, онда алынған х мәні $F(x)$таралу функциясы бар кездейсоқ шаманың X мәні болып табылады.

Осылайша, берілген $F(x)$ таралу функциясы бар кездейсоқ шама мәнін алу процедурасы екі сатыдан тұрады:

1) интервалда [0; 1) біркелкі тараған кездейсоқ R шаманы іске асыру;

2) формула (4.2) бойынша кездейсоқ шама мәнін есептеу.



Сур. 4.1. Кері функция әдісін бейнелеу

Мысал. Х кездейсоқ шама интервалда [a; b) біркелкі таралуы бар. Интервалда [a; b) кездейсоқ Х шаманы модельдеу бағдарламасын құру.

Шешімі.Кездейсоқ Х шаманың таралу функциясы төмендегідей түрге ие

$$F(x)=\begin{cases}0, & x<a;\\ \dfrac{x-a}{b-a}, & a\le x<b;\\ 1, & x\ge b.\end{cases}$$

Кері функция әдісіне сай

$$F(x)=\frac{x-a}{b-a}=r,$$

одан

$$x= a+r\ (b-a)$$

C# тілінде кездейсоқ шаманы модельдеу бағдарламасын құрастырамыз:

```
static void Main(string[] args)
{
Console. Write(''Интервалдың сол шегін беріңіз:   ");
double a = double.Parse(Console.ReadLine());
Console. Write(''Интервалдың сол шегін беріңіз:   ");
double b = double.Parse(Console.ReadLine());
Console. Write(''n кездейсоқ шама мөлшері: ");
   int n = int.Parse(Console.ReadLine());
   Random random = new
   Random(); double r, x;
   for (int i = 0; i < n; i++)
   {
      r =
      random.NextDouble(); x
      = a + r * (b - a);
      Console. WriteLine("x[{0}] =
                                  {1:f3}", i + 1,  x);
   }
   Console. WriteLine(Модельдеу аяқталды");
   Console.ReadLine();
}
```

Мысал. Кездейсоқ Х шама экспоненциальды (көрсеткішті) таралу заңына ие

$$f(x)=\lambda e^{-\lambda x},\ x\ge 0,$$

мұндағы λ–таралу көрсеткіші, 1 > 0 .

Кездейсоқ X шамасын модельдеу бағдарламасын құрастыру.

Шешімі. Кездейсоқ X шаманың таралу функциясы

$$F(x) = \int_{-\infty}^{x} f(x)\mathrm{d}x = \int_{0}^{x} \lambda e^{-\lambda x}\mathrm{d}x = 1 - e^{-\lambda x}.$$

Кері функция әдісіне сай

$$F(x) = 1 - e^{-\lambda x} = r.$$

Аламыз

$$e^{-\lambda x} = 1 - r;$$

$$-\lambda x = \ln(1 - r);$$

$$x = -\frac{1}{\lambda}\ln(1 - r).$$

Кездейсоқ шама (1 - R)Rсияқты таралуға ие болғандықтан, онда кездейсоқ X шааны табу кезінде формуланы қолданады

$$x = -\frac{1}{\lambda}\ln(1 - r).$$

C# тілінде кездейсоқ шаманы модельдеу бағдарламасын құрастырайық:

```
static void Main(string[] args)
{
   Console. Write(''Таралу көрсеткішін беріңізLambda: ");
   double lambda = double.Parse(Console.ReadLine());
   Console.Write("n кездейсоқ шама мөлшері:      ");
   int n = int.Parse(Console.ReadLine());
   Random random = new
   Random(); double r, x;
   for (int i = 0; i < n; i++)
   {
      r = random.NextDouble();
      x = -Math.Log(r) /
      lambda;
      Console.WriteLine("x[{0}] = {1:f3}", i + 1, x);
   }
   Console. WriteLine("Модельдеу аяқтарды");
   Console.ReadLine();
}
```

Кері функция әдісі берілген $F(x)$ таралу заңымен кездейсоқ X шаманы түрлендіретін әмбебап және дәл әдіс болып табылады. Бірақ

тарату функциясы жазбасының соңғы формасы болмаған жағдайда, ол қолдануға жарамайды. Сонымен қатар, теңдеуді аналитикалық шешу үнемі мүмкін болмайды (4.2).

4.2. КЕСЕК-СЫЗЫҚТЫ АППРОКСИМАЦИЯ ӘДІСІ

Кесек-сызықты аппроксимация әдісін Н.П. Бусленко ұсынған және кері функция үшін бір көрініске қол жеткізе алмай жатқанда қолданылады. $F(x)$ тарату функциясы тура апроксимирленген кесінділерден (немесе кері функциясын табуға болатын басқа келетін қарапайым функциядан) тұратын $F^*(x)$ функциясымен ауыстырылады. Бөлудің интервал бөліктерінің саны ақырғы нәтиженің талап етілген дәлділігіне байланысты. Бірақ әр интервалда біркелкі таралу заңымен (немесе басқа апроксимирленетін қарапайым таралумен) кездейсоқ шама модельденеді. Кездейсоқ X шаманы іске асыруда кескінді алу үшін әр аппроксимирленетін интервалда ауыспалы X қатысты $r = F(x)$ теңдеуін шешу керек.

Мысалы. Таралу функциясы $F(x)$ үш сызықты функциямен аппроксимирленген кездейсоқ X шаманы модельдеу бағдарламасын

$$F(x)=\begin{cases} F_1(x)=0{,}5x, & x\in[0;1); \\ F_2(x)=0{,}3x+0{,}2, & x\in[1;2); \\ F_3(x)=0{,}1x+0{,}6, & x\in[2;4]. \end{cases}$$

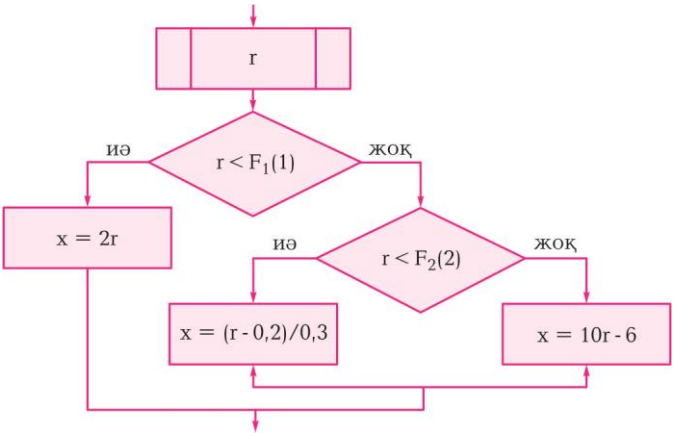
құрастыру:

Шешімі. Әр интервал үшін ауыспалы x қатысты $r = F(x)$ теңдеуін шешейік. Аламыз:

Кездейсоқ шаманы іске асыру алгоритмінің сызбасы 4.2 суретте келтірілген.

$$x=F^{-1}(r)=\begin{cases} 2r, & F_1(0)\le r<F_1(1); \\ \dfrac{r-0{,}2}{0{,}3}, & F_2(1)\le r<F_2(2); \\ 10r-6, & F_3(2)\le r\le F_3(4). \end{cases}$$

C# тіліндегі бағдарлама келесі түрге ие:



Сур. 4.2. Кездейсоқ шама түрленуі алгоритмінің сызбасы

```
static void Main(string[] args)
{
   Console Write ("n кездейсоқ шама мөлшері:   ");
   int n = int.Parse(Console.ReadLine());
   Random random = new Random
   (); double r, x;
   for (int i = 0; i < n; i++)
   {
      r = random. NextDouble();
      if (r < 0.5) x = 2 * r;
      else if (r < 0.8) x = (r - 0.2) / 0.3;
      else x = 10 * r - 6;
      Console.WriteLine("x[{0}] =
                                {1:f3}", i + 1,x);
   }
   Console.WriteLine("Модельдеу аяқталды");
   Console.ReadLine();
}
```

Мысал. Таралу функциясы $F(x)$ екі қарапайым функциямен аппроксимирленген кездейсоқ X шаманы модельдеу бағдарламасын құрастыру:

$$F(x)=\begin{cases}F_1(x)=x^2, & x\in[0;\ 0{,}7071);\\ F_2(x)=\sqrt{x-0{,}4571}, & x\in[0{,}7071;\ 1{,}4571].\end{cases}$$

Шешімі. Ауыспалы x қатысты әр интервалша R= F *(x)* теңдеуін шешеміз. Аламыз:

$$x = F^{-1}(r) = \begin{cases} \sqrt{r}, & F_1(0) \le r < F_1(0{,}7071); \\ r^2 + 0{,}4571, & F_2(0{,}7071) \le r \le F_2(1{,}4571). \end{cases}$$

C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
   Console. Write(n кездейсоқ шама мөлшері:    ");
   int n = int.Parse(Console.ReadLine());
   Random random = new
   Random(); double r, x;
   for (int i = 0; i < n; i++)
   {
      r = random.NextDouble();
      x = r < 0.5 ? Math.Sqrt(r) :r * r + 0.4571;
      Console. WriteLine("x[{0}] =
                                {1:f3}", i + 1,x);
   }
   Console. WriteLine ("Модельдеу аяқталды");
   Console.ReadLine();
}
```

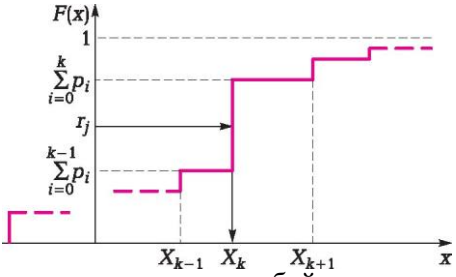
4.3. ЭМПИРИКАЛЫҚ ДЕРЕК БОЙЫНША КЕЗДЕЙСОҚ ШАМАНЫ МОДЕЛЬДЕУ

Әдіс тарату функциясы F (x) үшін көрінісі белгісіз және статистикалық деректер бар болған жағдайда қолданылады. Бұл деректер статистикалық қатар түрінде ұсынылған делік, яғни интервал $(x_i; x_{i+1})$, $i=0...$ $к$ -1, мен кездейсоқ шаманың оларға түсу қтималдылығын p_i бағалау.

Эмпирикалық деректер бойынша кездейсоқ шаманы түрлендіру үшін интервалда [0; 1) біркелкі тараған кездейсоқ R шамасы іске асады. Егер r_1 шамасы интервалға түссе $\left[\sum_{i=0}^{k-1} p_i; \sum_{i=0}^{k} p_i\right)$, онда кездейсоқ X шама $x_j = \frac{x_{k-1} + x_k}{2}$ ретінде *мәніқабылданады* (сур.4.3).

Мысал. Статистикалық қатар бар делік

$(x_t; x_{t+1})$	(0; 0,1)	(0,1; 0,2)	(0,2; 0,3)	(0,3; 0,4)	(0,4; 0,5)	(0,5; 0,6)
p_i	0,1	0,3	0,25	0,2	0,1	0,05



Сур. 4.3. Эмпирикалық деректер бойынша кездейсоқ шаманы түрлендіру әдісінің көрінісі

Кездейсоқ Х шаманы модельдеу бағдарламасын құру.
Шешімі.C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
   Conso1e Write ("n кезейсоқ шама мөлшері:      ");
   int n =
   int.Parse(Console.ReadLine());
   double[] xt = new double []
      {
        0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6
      };
   Double [] p = new double[]
      {
        0.1, 0.3, 0.25, 0.2, 0.1, 0.05
      };
   Random random = new
   Random(); double r, s, x;
   int k;
   for (int i = 0; i < n; i++)
   {
      r =
      random.NextDouble(); s
      = p[0];
      for (k = 1; s < r;
         k++) s += p[k];
      x = (xt[k] + xt[k - 1])  / 2;
      Console.WriteLine("x[{0}] ={1:f3}", i + 1, x);
   }
   Console.WriteLine("Модельдеу аяқталды");
   Console.ReadLine();
}
```

Сипатталған әдісті пайдалану кезінде кездейсоқ шаманың тек интервалдың ортасына тең дискретті мәндері ғана алынатын болады. Бірқатар жағдайда алынатын шама көптеген рұқсат етілгендердің

арасынан кез-келген мәнді қабылдауы тиіс.

Осындай шаманы модельдеу үшін таралудың үздіксіз кесек-сызықты эмпирикалық функциясына кері функция әдісін қолдануға болады. Сонымен қатар осы функцияны топтасқан да, топтаспаған да деректер бойынша құруға болады. Құру процесін қарастырайық.

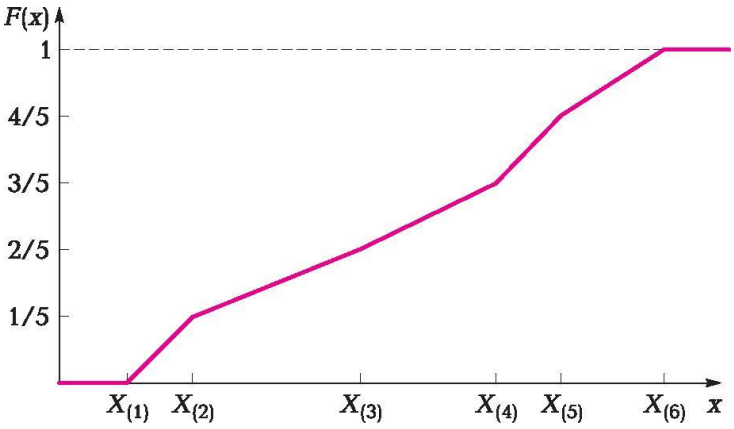
Кездейсоқ X_j, X_2,..., X_nшаманың nбақылауы бар делік. Онда ең алдымен X_iшамасын өсу ретінмен сұрыптап таралудың F үздіксіз кесек-сызықты функциясын анықтауға болады.

$X_{(i)}$мәні $X_{(1)} < X_{(2)} < ... < X_{(n)}$ болатындай X_j, X_2, ..., X_nшаасының і-ші ең кіші мәнін білдірсін делік. Онда Fбылай беріледі

$$F(x) = \begin{cases} 0, \text{ если } x < X_{(1)}; \\ \dfrac{i-1}{n-1} + \dfrac{x - X_{(i)}}{(n-1)(X_{(i+1)} - X_{(i)})}, \text{ если } X_{(i)} \le x < X_{(i+1)}; \\ 1, \text{ если } X_{(n)} \le x. \end{cases}$$

4.4 суретте n = 6 таралу функциясы графигінің мысалы көрсетілген.

Егер жеке бақылаулар болса, яғни F функциясы анықталса, онда кездейсоқ шаманы түрлендіру үшін келесі алгоритмді пайдалануға болады.



Сур. 4.4. Бастапқы деректерден алынған үздіксіз кесек-сызықты эмпирикалық таралу функциясы

1. Интервалда [0; 1) біркелкі таралуы бар R түрленеді.
2. $P = (n-1)R$және $I = LPJ + 1$есептелінеді.

3. $X = X_{(1)} + (P\text{-}I + 1)\ (X_{(I+1)} - X_{(1)})$ қайтып келеді.

Егер деректер топтастырылса, онда эмпирикалық таралуды анықтайтын басқа тәсілді қолдану керек, өйткені жеке X_iшамасының мәні белгілі емес.

nшамасы іргелес $к$интервалдарға $[a_0; a_1)$, $[a_1, a_2)$, ..., $[a_{k-1}; a_k)$топтастырылған жәнеj-$ші$интервалnбақылаудан тұрады, мұндағы$n_1 + n_2 + ... + n_k = n$. $j = 1, 2, ..., к$ үшін $G(a_0) = 0$, $G(aj) = (n_1 + n_2 + ... + nj)/n$болсын делік. Онда кесек-сызықты эмпирикалық таралуды анықтауға болады.

$$G(x) = \begin{cases} 0, \text{ если } x < a_0; \\ G(a_{j-1}) + \dfrac{x - a_{j-1}}{a_j - a_{j-1}}\left[G(a_j) - G(a_{j-1})\right], \text{ если } a_{j-1} \le x < a_j; \\ 1, \text{ если } a_k \le x. \end{cases}$$

Егер деректер топтастырылса, яғни G функциясы анықталса, кездейсоқ шаманы түрлендіру үшін келесі алгоритмді пайдалануға болады.

1. Интервалда [0; 1) біркелкі таралуы бар R түрленеді.
2. Теріс емес бүтін сан анықталынадыJ$(0 < J < к - 1)$, сондықтан $G(a_J) < R < G(a_{J+1})$.
3. $X = aj + [- G(aj)](aj+1 - aj) / [(aj+1) - G(aj)]$ қайтып келеді.

Эмпирикалық таралу бойынша кездейсоқ шама түрленуінің бір анық кемшілігі осылайша түрленген кездейсоқ шама ешқашан $X_{(1)}$ аз немесе$X_{(n)}$ көп болмайды. Сонымен қатар, орташа$F(x)$таңдамалы орташа X_iмәніне тең емес.

4.4. ТАҢДАУ ӘДІСІ

Егер кездейсоқ шаманы түрлендіру үшін дәл әдістерді қолдануға болмайтын болса, жақындатылғандарды пайдалануға болады. Олардың бірі таңдау әдісі (басқаша атауы – режекция әдісі, Нейман әдісі) болып табылады. Осы әдіс бірқатар шектелген [a; b)интервалда анықталған ықтималдылық тығыздығы $f(x)$ функциясының көмегімен берілген кездейсоқ шамын түрлендіруге мүмкіндік береді.

Таңдау әдісін Дж. Нейман ұсынған және оның мәні біркелкі тараған кездейсоқ сан функциясы болып табылатын координаталары бар кездейсоқ X шама тапсырмасының аумағынан нүкте «таңдалынатындығына» негізделген. Егер бұл нүкте X есептеу үшін

қолдану мүмкін болмаса, онда оның «лақтырылуы» және жаңасының «таңдалынуы» жүреді.

Тығыздық $f(x)$ интервалында [a; b) берілген бір өлшемді кездейсоқ X шаманы модельдеу қажет делік. Бұл интервалдан $f(x) = 0$тыс және, сонымен қатар, таралу тығыздығы жоғарыдан шеутелген, яғни $f(x) < C$, мұндағы C–тұрақты.

Кездейсоқ X шаманы алуды таңдау әдісі графикалық түрде 4.5 суретте көрсетілген және келесіге негізделген:

1) интервалда [0; 1) біркелкі тараған кездейсоқ R шаманың екі тәуелсіз r_1 және r_2 мәнін алады;

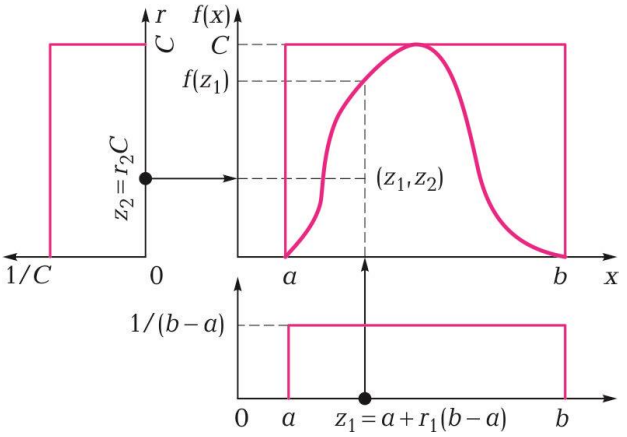
2) координаталармен нүктені құрады (z_1, z_2), мұндағы

$$z_1 = a + r_1(b - a); \quad z_2 = r_2 C;$$

3) егер $z_2 < f(z_1)$ (яғни нүкте $f(x)$ қисығымен және $x = a$, $x = b$ және $y = 0$ туралармен шектелген ауданға түседі) болса, онда X шамасы z_1 мәнін қабылдады деп болжайды; болмаса $z = (z_j, z_2)$ нүктесі лақтырылады және 1 тармақ бойынша жаңа кездейсоқ сан жұбын $(r_1; r_2)$ алып есептеу қайталанады.

Мысал. Ықтималдылық тығыздығының белгіленген таралу функциясымен кездейсоқ шаманы қалыптастыру бағдарламасын

$$f(x) = \begin{cases} 2(x^2 + x^5), x \in [0,1]; \\ 0, x \notin [0;1]. \end{cases}$$



Сур. 4.5. Таңдау әдісін бейнелеу

Шешімі. Бірінші таңдау әдісінің a, bжәнеCкөрсеткіштерін анықтаймыз. f(x)функциясы 0-ден 1 дейінгі интервалда нөльдік мәнге ие. Сәйкесінше, таңдау әдісінің параметрі a= 0, ал парметрib= 1. Осы функция интервалда [0; 1] бірқалыпты өсуші болып табылады. Сәйкесінше, функция максималды мәнді оң шекарада қабылдайды, яғни C = f(1) = 4 көрсеткіші.

Одан әрі осы әдісті қолданатын кездейсоқ шаманы түрлендіру үшін C# тілінде бағдарламаны елестетейік.

```
static void Main(string[] args)
{
   Сопзо1е^г^е("Кездейсоқшама n мөлшері: ");
   int n = int.Parse(Console.ReadLine());
   Randomrandom = newRandom();
   // ықтималдылық тығыздығының функциясы
   // нөльдік шекті береміздоublea= 0, b= 1;
   // ықтималдылық тығыздығының функциясы үшін
   // максималды мәнді береміз
   double C = 4;
   double x, r1, r2 z1, z2;
   bool f;
   for (int i = 0; i < n; i++)
   {
      f =
      true; do
      {
         r1 =
         random.NextDouble(); r2
         = random.NextDouble();
         z1 = a + r1 * (b - a);
         z2 = C * r2;
         if (z2 < 2 * (Math.Pow(z1, 2) + Math.Pow(z1,
         5)))
         {
            x = z1;
            Console.WriteLine("x[{0}] ={1:f3}", i + 1,
                                          x);
            f = false;
         }
      }
      while (f);
   }
   Console.WriteLine("Модельдеу аяқталды");
   Console.ReadLine();
}
```

Таңдау әдісінің тиімділігі z= (z_j, z_2) нүктесі X есептеу үшін қолданылатын ықтималдылықты атайды. Қарастырылған процедурада әдістің тиімділігі тік бұрыштың *C(b- a)* ауданында қисықпен *f (x)*, осьпен x, *x= a* және *x= b* туралармен шектелген ауданға қатысты сипатталынады.

4.5. КЕЗДЕЙСОҚ ШАМАНЫҢ ҚАЛЫПТЫ ТАРАЛУЫ

Таралудың арнайы заңдарын түрлендіру әдістерін қарастырайық.

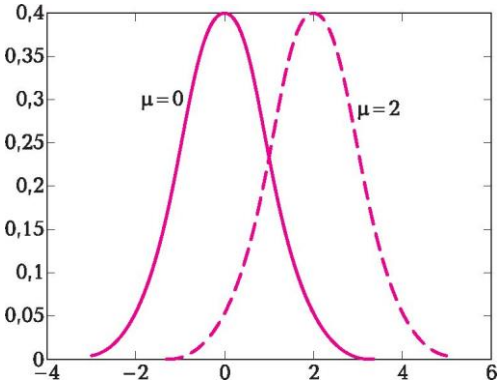
Әр түрлі теориялық заңдардың ішінен үздіксіз кездейсоқ шаманың қалыпты таралу заңы ерекше орын алады, өйткені көптеген практикалық зерттеулерде негізгі болып табылады және онымен өндірімділік пен басқа да процестермен байланысты көптеген кездейсоқ құбылыстар сипатталады. Қалыпты таралу заңына бағынатын кездейсоқ құбылысқа, мысалы, өндірістің көрсеткіштерді өлшеу қателігі, дайындаудың технологиялық қателігінің тарауы, көптеген биологиялық нысандардың бойы мен салмағы, қабыршақты резисторлар көрсеткіштерінің таралуы жатады.

Осыған байланысты әр түрлі құбылысты модельдеу кезінде қарауда қалыпты таралу заңына жауап беретін кездейсоқ шама тізбегінің болу қажеттілігі туындайды.

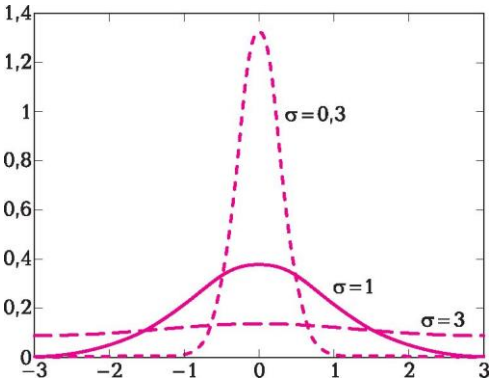
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Қалыпты таралу заңы тығыздықтың функциясы болып табылады

Қалыпты таралу параметрами μжәнеσкөрсеткіштерімен сипатталады және N(μ, σ^2) ретінде белгіленеді. Қалыпты тараған кездейсоқ шаманың математикалық күтілуі μ тең, ал дисперсияσ^2тең екендігін айта кетейік. μмәні әр түрлі болған кездегі ықтималдылық тығыздығы функциясының графигі 4.6 суретте (μөлшеген кезде график жылжиды), ал әр түрлі мәнде σ–сур. 4.7 (σөлшеген кезде граяфик созылады немесе тартылады) көрстілген.



Сур. 4.6. μ және σ = 1 әр түрлі болған кездегі қалыпты таралу тығыздығының функциясы



Сур. 4.7. Әр түрлі σ және μ= 0 кезіндегі қалыпты таралу тығыздығының функциясы

Қалыпты тараған кездейсоқμ= 0 жәнеσ = 1 шамасы қалыпты тараған кездейсоқ шама деп аталынады да, $N(0,1)$ ретінде белгіленеді. Осы шаманың маңыздылығы μ және σ^2 көрсеткіштері **бар қалыпты тараған кездейсоқ** *X′ шамасын шамасын келесі түрлендіру көмегімен стандартты қалыпты тараған кездейсоқ X шаманы алуға болады:*

$$X' = \mu + \sigma X.$$

Сондықтан одан әрі $N(0,1)$ таралуы бар кездейсоқ шаманың түрлендіру әдісін ғана қарастыратын боламыз.

4.5.1. ЖУЫҚТАУ ӘДІСІ

Қалыпты таралуы N (0,1) бар кездейсоқ шама модельдеудің классикалық жалпы әдістерімен іске асырылуы мүмкін. Мысал ретінде осындай әдістерді бірі – жуықтау әдісін қарастырайық.

Жуықтау әдісі *формулада* x = $F^{-1}(r)$ таралу функциясына айналдыру күрделі сандық тапсырма болып табылатын жағдайларда қолданылады. *бірақ кері функция* F^{-1}үшін айтарлықтай жақсы жуықтауды табуға болады.

Мысал. *N(0,1) көрсеткіштерімен бар қалыпты таралуы юар кездейсоқ X шаманы модельдеу бағдарламасын құрастыру.*

$$f(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}},\quad -\infty<x<\infty,$$

Шешімі. Қалыпты таралудың кері функциясын оңай есептелінетін, қарапайым формула түрінде елестетуге болмайды. Егер тығыздықтың кездейсоқ шамасы үшін жуықтауды пайдаланса

$$\sqrt{\frac{2}{\pi}}e^{-\frac{x^2}{2}}\approx\frac{2ke^{-kx}}{\left(1+e^{-kx}\right)^2},\ x>0,$$

мұндағық $k=\sqrt{\frac{8}{\pi}}$, онда оң жартылай ось үшін келесі жуықтауды алуға болады*:*

$$r=F(x)=\int_0^{\infty}\frac{2ke^{-kx}}{\left(1+e^{-kx}\right)^2}dx=\frac{2}{1+e^{-kx}}-1.$$

Онда кері функция келесі түрге ие

$$x\approx\frac{1}{k}\ln\frac{1+r}{1-r},\quad 0\le r<1.$$

Егер x мәніне 0,5 белгі « + » ықтималдылығы мен 0,5 белгі «-» ықтиималдылығын берсе, онда алынған тізбек N(0, 1) таралуы бар кездейсоқ шаманы іске асыру ретінде қарастырылуы мүмкін. C# тіліндегі бағдарлама келесі түрге ие:

```
Staticvoid Main (string [] args)
{
   Console.Write("n кездейсоқ шама мөлшері:    ");
   int n = int.Parse(Console.ReadLine());
   Random random = new Random();
   double r, x;
   double k = Math.Sqrt(8.0 /
   Math.PI);
   for (int i = 0; i < n; i++)
```

```
{
   r = random.NextDouble();
   x = Math.Log((1 + r) / (1 - r)) /
   k;
   r =
   random.NextDouble();
   if (r < 0.5) x = -x;
   Console.WriteLine("x[{0}] =
                              {1:f3}", i + 1,x);
}
Console.WriteLine("Модельдеу аяқталды");
Console.ReadLine();
```

4.5.2. Орталық шекті теореманы пайдалану

Кездейсоқ шаманың қалыпты тараған бағдарламалық іске асырудың ең белгілі әдістерінің бірі орталық шекті теоремаға негізделген: тәуелсіз кездейсоқ шама соммасының таралуы *n* шексіз артуы кезінде қалыптыға жақындайды, егер төмендегі шарттар орындалса:

1) барлық шамалар соңғы математикалық күтілім мен дисперсияға ие;

2) мәні бойынша шаманың бірде біреуі басқа қалғандарынан қатты ерекшеленбейді.

Осы теоремаға сай интервалда [0; 1) біркелкі тараған тәуелсіз кездейсоқ Tj, r_2,..., r_nшама соммасымен таралу жуықтауы $N(0,1)$ негізінде кездейсоқ X шаманы іске асыру алгоритмін құрастыруға болады. Олардың әрқайсысы математикалық күтілуге $M(r_i) = 1/2$ және дисперсияға $D(r) = 1/12$, $i = 1,2,\ldots, n$ ие болғандықтан, онда

$$M\left(\sum_{i=1}^{n} r_i\right) = \frac{n}{2}, \quad D\left(\sum_{i=1}^{n} r_i\right) = \frac{n}{12}.$$

Кездейсоқ шама таралуының орталық шекті теоремасына сай

$$x = \frac{\sum_{i=1}^{n} r_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \qquad (4.3)$$

n жеткілікті үлкен болған к қтайды. Арқырғы *n*таралу кезінде қалыптығыжақын.

Практика көрсеткендей, $n = 12$ болған кезде айтарлықтай жақсы

және есептеу үшін ыңғайлы жуықтау аламыз е.

Модельдеуге арналған формула келесі түрге ие

$$x = \sum_{i=1}^{12} r_i - 6.$$

Мысал. $N(0,1; 2,25)$ таралу талабына жауап беретін кездейсоқ Х шаманы модельдеу бағдарламасын құру.

Шешімі.C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
   Console. Write("Математикалық күтілуі: ");
   double m = double.Parse (Console.ReadLine());
   Console. Write(''Дисперсия d:    ");
   double d = double.Parse(Console.ReadLine());
   Console. Write(''n кездейсоқ шама мөлшері:   ");
   int n = int.Parse(Console.ReadLine());
   Random random = new
   Random(); double s, r;
   for (int i = 0; i < n; i++)
   {
      s = 0;
      for (int j = 0; j < 12; j++)
      {
         r =
         random.NextDouble(); s
         += r;
      }
      double x = s - 6;
      x = m + x * Math.Sqrt(d);
      Console. WriteLine("x[{0}] =
                                 {1:f3}", i + 1,x);
   }
   Console. WriteLine(''Модельдеу аяқталды");
   Console.ReadLine();
}
```

Орталық шекті теоремаға негізделген әдістің кемшілігіне көріністен (4.3) байқалғандай, қалыпты таралған сан кез-келген заттық мәнді қабылдай алатын кезде, оның көмегімен ақырғы диапазоннан кездейсоқ сан алынатындығын жатқызуға болады.

4.5.3. БОКС ЖӘНЕ МАЛЛЕР ӘДІСІ

Басқа белгілі қалыпты таралған кездейсоқ шама генераторында Бокс пен Маллердің дәр кері әдісі қолданылады. Әдіс бірлескен

тығыздық $f(x, y)$ берген екі өлшемді кездейсоқ $[X, Y]$шаманы іске асыруға арналған. Егер кездейсоқ XжәнеYшама тәуелсіз болса, онда $f(x, y) = =$(x)f(y) және екі өлшемді кездейсоқ шаманың таралуын модельдеу келесі теңдеу көмегімен кездейсоқ R_1жәнеR_2сандарының интервалында [0; 1) тәуелсіз, біркелкі тараған нормаланған қалыпты xжәнеyшаманың жұбын модельдеуге әкеп соғады:

$$x = \sqrt{-2\ln r_1}\cos(2\pi r_2);$$
$$y = \sqrt{-2\ln r_1}\sin(2\pi r_2).$$

Әдіс жақсы нәтиже береді, оңай бағдарламаланады және айтарлықтай тез жұмыс істейді.

Мысал. N(-1,5; 0,9) жәнеN(5;1,9б) таралуына жауап беретін кездейсоқ екі тізбек шамасын модельдеу бағдарламасын құру.

Шешімі.C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
   Console.WriteLine("Бірінші таралу көрсеткіштері");
   Console.Write(''математикалық күтілім ml:    ");

   double ml = double.Parse(Console.ReadLine());
   Console.Write(''дисперсия d1:    ");

   double d1 = double.Parse(Console.ReadLine());
   Console.WriteLine(''Екінші таралу көрсеткіштері");
   Console.Write("математикалық күтілімm2: ");

   double m2 = double.Parse(Console.ReadLine());
   Console.Write("дисперсия d2:     ");

   double d2 = double.Parse(Console.ReadLine());
   Console.Write(''Кездейсоқn шама мөлшері:   ");

   int n = int.Parse(Console.ReadLine());
   Random random = new Random();
   double r1, r2, x, y;
   for (int i = 0; i < n; i++)
   {
      r1 =
      random.NextDouble();
      r2 =
      random.NextDouble();x
      = Math.Sqrt(-2 *
      Math.Log(r1)) *
```

```
            Math.Cos(2 * Math.PI * r2); x
        = ml + x * Math.Sqrt(d1); y =
        Math.Sqrt(-2 * Math.Log(r1)) *
            Math.Sin(2 * Math.PI * r2); y
        = m2 + y * Math. Sqrt(d2);
        Console. WriteLine("x[{0}] = {1:f3}\ty[{0}] =
            {2:f3}", i + 1, x, y);
    }
    Console. WriteLine("Модельдеу аяқталды");
    Console.ReadLine();
}
```

5.4.5. Марсаль мен Брей әдісі

Марсаль мен Брей әдісі Бокс пен Маллер әдісінің модификациясы болып табылады. Бұл әдіс кездейсоқ X және Y шамалары тәуелсіз болатын жағдайда бірлескен тығыздық функциясы f(x, y) = f(x)f(y)берілген екі өлшемді кездейсоқ [X, У]шаманы іске асыруға арналған. Бұл әдіс есептелінген синустар мен косинустарды алып тастау арқасында жылдамырақ болып табылады. Кезедейсоқ шаманы алу процедурасы келесіге негізделінген:

1) г интервалда [-1; 1) біркелкі тараған тәуелсіз, кездейсоқ U_1жәнеU_2шамалар түрленеді:

$$U_1 = -1 + 2R_1;$$
$$U_2 = -1 + 2R_2,$$

мұндағыR_1, R_2– интервалда [0; 1) біркелкі тараған тәуелсіз, кездейсоқ сан;

2) $U_1^2 + U_2 < 1$ шарттары тексеріледі. Егер шарттар орындалмаса, онда циклді қайта қайталайды. Егер шар дұрыс болса, онда формула бойынша кездейсоқ шаманы іске асырады

$$X = U_1\sqrt{\frac{-2\ln(U_1^2 + U_2^2)}{U_1^2 + U_2^2}},\ Y = X\frac{U_2}{U_1}.$$

Әдіс жақсы нәтижелер береді және оңай бағдарламаланады. Сонымен қатар осы әдіс келесі кемшіліктерге ие:

- айтарлықтай баяу, өйткені онда ұзақ есептілінетін тригонометриялық функциялар қолданылады;
- нөлге жақынr$_1$, кезінде ЭЕМ қатарлық торының шамадан тыс толуы мүмкін.

Осы кемшіліктер кездейсоқ шаманың миллионын алуды талап ететін стохастикалық модельдерде генераторды пайдаланған жағдайда елеулі болып табылады.

Мысал. N(-1,5; 0,9) жәнеN(2,5; 1,96)таралуына жауап беретін екі кездейсоқ шама тізбегін модельдеу бағдарламасын құру.

Шешімі.C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
Console.WriteLine("Бірінші таралу көрсеткіштері");
Console.Write(''Математикалық күтілімml");
double ml = double.Parse(Console.ReadLine());
Console.Write(''дисперсия d1: ");
double d1 = double.Parse(Console.ReadLine());
Console.WriteLine(''Екінші таралу көрсеткіштері");
Console.Write("математикалық күтілім m2: ");
double m2 = double.Parse(Console.ReadLine());
Console.Write("дисперсия d2: ");
double d2 = double.Parse(Console.ReadLine());
Console.Write(''Кездейсоқп шамасының мөлшері:  ");
   int n = int.Parse(Console.ReadLine());
   Random random = new Random();
   double r1, r2, u1, u2, s, x,
   y;
   int i = 0;
   while (i <
   n)
   {
      r1 = random.
      NextDouble(); r2 =
      random.NextDouble();
      u1 = -1 + 2 * r1; u2 =
      -1 + 2 * r2; s = u1 *
      u1 + u2 * u2; if (s <=
      1)
      {
         x =  u1 *Math.Sqrt(-2 * Math.Log(s) /  s);
         y =  u2 *Math.Sqrt(-2 * Math.Log(s) /  s);
         x =  m1 + x * Math.Sqrt(d1);
         y =  m2 + y * Math.Sqrt(d2);
         Console. WriteLine("x[{0}] = {1:f3}\ty[{0}] =
         {2:f3}", i + 1, x, y); i ++;
      }
   }
   Console. WriteLine("Модельдеу аяқталды");
   Console.ReadLine();
}
```

4.6. АРНАЙЫ ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Қосымшада ең жиі кездесетін кездейсоқ шама таралуының бірқатар заңдарында модельдеудің жиі тәсілдері әзірленген, жиі тәсілдер жалпы әдістерді қолдану мүмкіндігін алып тастамайды. Әр нақты тапсырманың шешімі модельдеу алгоритмін таңдауда жеке тәсілді талап етеді. Көсрсеткіштердің бірі біркелкі таралған кездейсоқ сан үшін қолданатын санына таралуын талап ететін жалған кездейсоқ шама мөлшеріне қатысты түсінік деп қабылдайтын *әдістің тиімділігі* болып табылады.

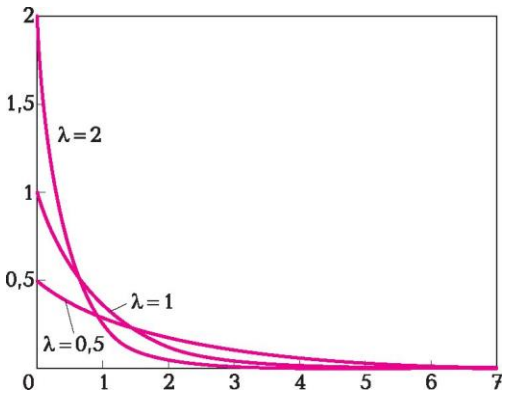
4.6.1.Экспоненциалды таралуды модельдеу

Экспоненциалды немесе көрсеткішті таралу –бұл бірдей уақиғаның екі тізбекті жасауы арасындағы уақытты,құрылғының мүлтіксіз жұмыс істеу уақытын модельдейтін үздіксіз тарату.

Экспоненциалды таралу тығыздығының функциясы теңдеумен беріледі

$$f(x) = \lambda e^{-\lambda x}, \; x \geq 0,$$

мұндағы X –таралу көрсеткіші, $1 > 0$.



Сур. 4.8. Экспоненциалды таралу тығыздығының функциясы

Экспоненциалды таралу үшін маткүтілуі мен дисперсия теңдеуге сай анықталады

$$M(X) = \lambda;$$
$$D(X) = \lambda^2.$$

Ықтималдылық тығыздығы функциясының графигінің әр түрлі мән үшін X көрсеткіші 4.8. суретте көрсетілген.

X көрсеткіші бар экспоненциалды таралудың кездейсоқ шамасын модельдеуге арналған формула төмендегі түрге

$$x = -\frac{1}{\lambda}\ln r.$$

Модельдеу мысалы 4.1 тармақшада қарастырылды.

4.6.2. Бета-таралуды модельдеу

Бета-таралу ақырғы жоғарғы және төменгі шекті мәндермен үздіксіз таралу болып табылады. Осындай шектеулер көптеген нағыз уақиғаларда болуы мүмкін, сондықтан бета-таралуды жеткілікті ақпаратты үлкен мөлшерде алуға дейін нағыз таралуды жуықтау үшін қолдануға болады. Бета-таралу жеткілікті шамада көптеген деректерге сай келеді.

Интервалдағы (0; 1) бате-таралудың тығыздығы формуламен анықталады

$$f_{p,m}(x) = \frac{x^{p-1}(1-x)^{m-1}}{\mathrm{B}(p, m)}, \quad 0 < x < 1,$$

мұндағыр > 0, m > 0–таралу көрсеткіштері.

Бета-функция төмендегі түрге ие

$$\mathrm{B}(p, m) = \int_0^1 x^{p-1}(1-x)^{m-1}\,dx.$$

Бета-таралу үшін маткүтілу мен дисперсия формулаға сәйкес анықталады.

$$M(X) = \frac{p}{p+m};$$

$$D(X) = \frac{pm}{(p+m)^2(p+m+1)}.$$

М және ркөсеткіштерінің әр түрлі мәндері үшін ықтималдылық тығыздығы функциясының графигі 4.9 суретте көрсетілген.

Егер m–бүтін, р–бүтін емес болса, р және m көрсеткіштері бар бета-таралуына ие кездейсоқ Xpm шаманы модельдеуге арналған формула төмендегі түрге ие болады

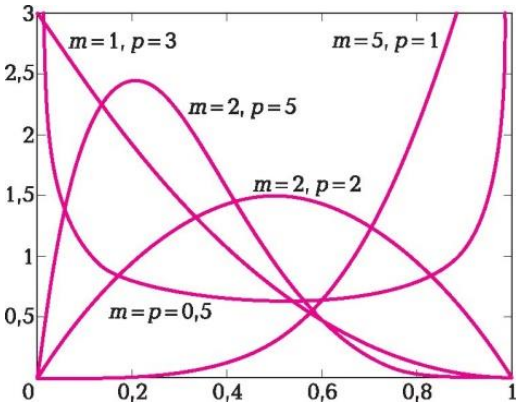
$$x_{p,m} = \prod_{k=1}^{m} r_k^{\frac{1}{p+k-1}} = \exp\left(\sum_{k=1}^{m} \frac{\ln(r_k)}{p+k-1}\right)$$

Мұндағы r1, r2,..., rm–интервалда (0; 1) біркелкі тараған кездейсоқ сандар.

Формуланы m–бүтін емес, р–бүтін болған жағдайда да, У = 1 – х түрінің ауыспалысын алдын ала алмастырып қолдануға болады.

Бүтін емес р > 0, m > 0 көрсеткіштермен бета-таралуды модельдеу үшін суперпозиция әдісін пайдалануға болады. Онда модельдеуге арналған формула төмендегі түрге ие

$$x_{p,m} = \exp\left(\sum_{k=1}^{\lfloor m \rfloor + 1} \frac{\ln(r_k)}{\nu + p + k - 1}\right),$$



Сур. 4.9. Бета-таралу тығыздығының функциясы

мұндағы [m]–*m* бүтін бөлігі; *v*–p(v = k) = p_k, κ =0, 1, 2, ... таралуымен бірге берілген дискретті таралуы бар бүтін сандық кездейсоқ шама:

$$p_k = \frac{\lfloor m \rfloor!}{B(p,m)} \cdot \frac{a(a+1)\ldots(a+k-1)}{k!(k+p)(k+p+1)\ldots(k+p+\lfloor m \rfloor)}, \quad a = \lfloor m \rfloor + 1 - m.$$

Бета-таралуға жауап беретін кездейсоқ шаманы модельдеудің қарапайымдау процедурасын австриялық математик Йонк ұсынды. Процедура келесіге негізделген:

1) интервалда (0; 1) біркелкі тараған кездейсоқ r_1 және r_2 шамалар түрленеді;

2) $r_1^{1/p} + r_2^{1/m} \le 1$, шарты тексеріледі. Егер шарт орындалмаса, онда цикл қайталанады. Егер шарт дұрыс болса, онда формула бойынша кездейсоқ шама іске асырылады

$$r_{p,m} = \frac{r_1^{1/p}}{r_1^{1/p} + r_2^{1/m}}.$$

Мысалы. m= 3, p= 2,5 көрсеткіштері бар бета-таралуға жауап беретін кездейсоқ санды модельдеу бағдарламасын құру.

Шешімі. m–бүтін, p–бүтін емес болған кездегі модельдеуге арналған жалпы формуланы қолданамыз.C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
Console.Write("Тарату көрсеткіштерilambda: ");
int lambda = int.Parse(Console.ReadLine());
Console.Write(''Кездейсоқп шама мөлшері:   ");
int m = int.Parse(Console.ReadLine());
Random random = new Random();
double p, r, x;
for (int i = 0; i < n; i++)
{
p =1;
for (int k = 0; k < lambda; k++)
{
r = random.NextDouble(); p *= r;
}
x = -Math.Log(p);
```

```
   Console.WriteLine("x[{0}] =     {1:f3}", i + 1,
      x);
   }
   Console. WriteLine ("Модельдеу аяқталды");
Console.ReadLine();
   }
```

4.6.3. Гамма таралуды модельдеу

Гамма таралу экономика мен менеджментте, сенімділік пен сынақ теориялары мен практикасында, техника, метеорология және т.б. салаларда кеңінен қолданылады. Айта кетсек, гамма-таралуға көптеген жағдайларда өнімнің жалпы қызмет ету мерзімі, өнімнің коррозия кезінде шекті күйге жету уақыты, *k-ші*тоқтауға дейінгі жұмыс істеу уақыты, *к* =1, 2, ..., және т.б. сияқты шамалар бағынады. Осы таралу қорды басқарудың (логистиканың) экономика-математикалық модульдерінде сұранысты сипаттау үшін жарамды.

Гамма-таралу тығыздығының функциясы формуламен беріледі

$$f_{\alpha\beta}(x) = \frac{x^{\alpha-1} e^{-\frac{x}{\beta}}}{\beta^{\alpha}\Gamma(z)},$$

Мұндағы $\lambda > 0$, $ß>0$–таралу көрсеткіштері; Г(z) –гамма- функция;

$$\Gamma(Z) = \int_0^{\infty} t^{z-1} e^{-t} dt.$$

Жекелеген жағдайда бүтін Z үшін

$$\Gamma(Z + 1) = Z!$$

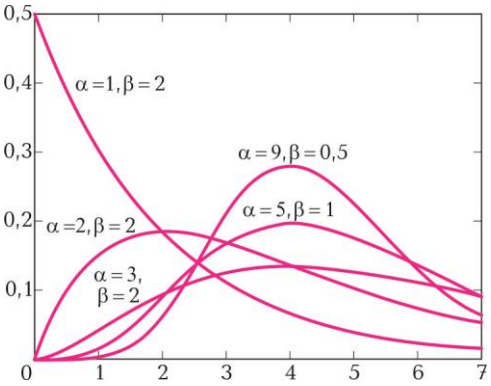
Гамма-таралу үшін маткүтілу мен дисперсия теңдеуге сай анықталады

$$M(X) = \lambda ß;$$

$$D(X) = \lambda ß^2.$$

А және β көрсеткіштерінің әр түрлі мәндері үшін ықтималдылық тығыздығы функциясының графигі 4.10 суретте көрсетілген.

Айта кететін жайт, көрсеткіш мәні α= 1 болған жағдайда, гамма-таралу ертерек қарастырылған β көрсеткіші бар экспоненциалды таралуға сәйкес келеді.



Сур. 4.10. Гамма-таралу тығыздығының функциясы

Бүтін санды $\alpha = n > 1$ көрсеткіші мен $\beta = 1$ көрсеткіші бар гамма-таратын кездейсоқ санды модельдеуге арналған формула келесі түрге ие

$$x = -\sum_{k=1}^{n} \ln(r_k) = -\ln\left(\prod_{k=1}^{n} r_k\right). \qquad (4.4)$$

Бүтін санды α және $\beta \neq 1$ көрсеткішімен гамма-таратуы бар кездейсоқ санды алу үшін формуланы (4.4) β көйбейткен жеткілікті.

Мысал. $\alpha = n, \beta = 1$ көрсеткіші бар гамма-таратуға жауап беретін кездейсоқ шаманы модельдеу бағдарламасын құру.

Шешімі. C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
   {
   Console.Write("Тарату көрсеткіштерilambda:   ");
   int lambda = int.Parse(Console.ReadLine());
   Console.Write(''Кездейсоқп шама мөлшері:     ");
   int m = int.Parse(Console.ReadLine());
   Random random = new Random();
   double p, r, x;
   for (int i = 0; i < n; i++)
   {
   p = 1;
   for (int k = 0; k < lambda; k++)
   {
   r = random.NextDouble(); p *= r;
```

```
    }
    x = -Math.Log(p);
    Console.WriteLine("x[{0}] ={1:f3}", i + 1,   x);
    }
    Console. WriteLine ("Модельдеу аяқталды");
Console.ReadLine();
    }
```

4.6.4. Логарифмді-қалыпты тарату заңын модельдеу

Егер $X = \exp(Y)$ болса, кездейсоқ Э шамасы μ, σ көрсеткіштерімен логарифмды-қалыпты тарауға ие болады, мұндағы Y μ, σ көрсеткіштерімен қалыпты тарауға ие. Осы таралу көмегімен қандай да бір тапсырманың орындалу уақыты, басқа шаманың үлкен санының туындысы болып табылатын шама жиі сипатталынады.

Кездейсоқ шаманың логаррифмді-қалыпты таралу заңы келесі тығыздық функциясымен сипатталынады:

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right],$$

Мұндағы σ >0 және μ–таралу көрсеткіштері.

Осы таралуға арналған маткүтілу мен дисперсия формулаға сай анықталынады

$$M(X) = e^{\mu+\sigma^2/2};$$

$$D(X) = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1).$$

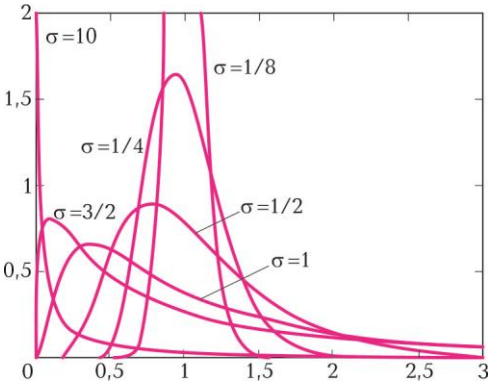
μ= 0 мен σкөрсеткішінің әр түрлі мәніне арналған ықтималдылық тығыздығы функциясының графигі 4.11 суретте көрсетілген.

Логарифмді қалыпты таралуы бар кездейсоқ шаманы модельдеу үшін теңдеу қолданылады

$$x = \exp(\sigma y + \mu),$$

Мұндағы y–μ= 0, σ = 1 көрсеткіштері бар стандартты қалыпты тараған кездейсоқ шама.

Мысал. Логарифмді-қалыпты таратуға жауап беретін кездейсоқ Х шаманы модельдеу бағдарламасын құру.



Сур. 4.11. μ= 0 кезіндегі логарифмді қалыпты тарау тығыздығының функциясы

Шешімі. C# тіліндегі бағдарлама келесі түрге ие:

```
   static void Main(string[] args)
   {
   Console.WriteLine("Таралу көрсеткіші");
   Console.Write("c:      ");
   double c = double.Parse(Console.ReadLine());
    Console.Write("b:     ");
   doubleb       =       double.Parse(Console.ReadLine());
Console.Write(''Кездейсоқп шамасының мөлшері:     ");
   int n = int.Parse(Console.ReadLine());
   Random random = new Random();
   double s, r, x;
   for (int i = 0; i < n; i++)
   {
   s = 0;
   for (int j = 0; j < 12; j++)
   {
   r = random.NextDouble(); s+= r;
   }
   x = s -   6;
   x = Math.Exp(c * x + b);
   Console.WriteLine("x[{0}] =  {1:f3}", i + 1,    x);
   }
   Console.WriteLine("Модельдеу                  аяқталды");
Console.ReadLine();
   }
```

4.6.5. Вейбулл таралуын Модельдеу

Вейбулл таралуы көмегімен қандай да бір тапсырманы орындау уақыты, шамасы, құрылғының дамылсыз жұмыс істеу уақыты жиі сипатталынады.

Вейбулл таралуына арналған тығыздық функциясы келесі түрге ие

$$f(x) = \frac{cx^{c-1}}{b^c}\exp\left(-\left(\frac{x}{b}\right)^c\right),$$

мұндағы$0 \le x < \infty$; $c > 0$; $b > 0$; b, c–бүтін.

Вейбулл таралуы үшін маткүтілу мен диспесия теңдеуге сай анықталады

$$M(X) = \frac{b}{c}\Gamma\left(\frac{1}{c}\right);$$

$$D(X) = \frac{b^2}{c}\left[2\Gamma\left(\frac{2}{c}\right) - \frac{1}{c}\left[\Gamma\left(\frac{1}{c}\right)\right]^2\right].$$

Мұнда Γ( ) – гамма функция.

b= 1 мен c әр түрлі мәнінің көрсеткіштері үшін ықтималдылық тығыздығы функциясының графигі 4.12 суретте көрсетілген.

Вейбулл таралуы келесі формуланы қолдану жолымен модельденуі мүмкін:

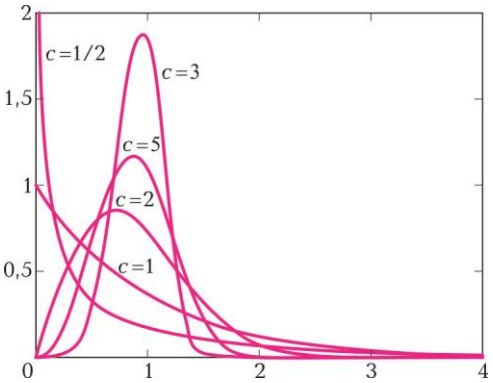
$$x = b[-\ln(r)]^{1/c},$$

мұндағы–интервалда (0; 1) біркелкі тараған кездейсоқ шама. Модельдеу кезінде r шамасы төменгі жағынан кіші шамамен шектелуі керектігін ескерейік.

Мысал. Вейбулл таралуына жауап беретін кездейсоқ X шаманы модельдеу бағдарламасын құрастыру.

Шешімі. C# тіліндегі бағдарлама келесі түрге ие:

```
staticvoidMain(string[] args)
{
   Console. WriteLine("Тарату көрсеткіштері");
   Console. Write("c: ");
   double c = double.Parse(Console.ReadLine());
   Console. Write("b: ");
   double b = double.Parse(Console.ReadLine());
```



Сур. 4.12. b= 1 болған кездегі Вейбулл тарату тығыздығының функциясы

```
 Console Write ("Кездейсоқ n шамасының мәлшері:");
   int n = int.Parse(Console.ReadLine());
   Random random = new
   Random(); double r, x;
   for (int i = 0; i < n; i++)
   {
      r = random.NextDouble();
      x = b * Math. Pow(-Math. Log(r), 1 / c);
      Console.WriteLine("x[{0}] =
                                {1:f3}", i + 1,x);
   }
   Console. WriteLine ("Модельдеу аяқталды");
   Console.ReadLine();
}
```

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. Кері функция әдісі неге негізделген?
2. Кері функция әдісін қандай жағдайда қолдануға болады?
3. Кесек-сызықты жуықтау әдісі неге негізделген? Ол қандай жағдайларда қолданылады?
4. Тарату функциясына арналған теңдеу белгісіз болса, кездейсоқ санды қалыптастырудың қандай әдісін қолдану тиіс? Ол неге негізделген?
5. Таңдау әдісі неге негізделген? Ол қандай жағдайларда

қолданылады?

6. Таңдау әдісінің геометриялық түсіндіруі қандай?
7. Қалыпты тарату заңына арналған тығыздық функциясы қандай?
8. Қалыпты тарату заңы үшін тарату функциясы қалай көрінеді?
9. Қалыпты таралу заңына жауап беретін кездейсоқ шама тізбегінің қалыптасуының қандай әдістері бар?
10. Кездейсоқ шаманың қалыпты таралуын модельдеуге арналған жуықтау әдісі неге негізделеді?
11. Қалыпты таралу заңына жауап беретін кездейсоқ шама тізбегін қалыптастыру үшін орталық шекті теорема қалай қолданылады?
12. Бокс және Маллер әдісінің маңызы неде?
13. Марсаль мен Брей әдісінің мәні неде?
14. Бета тарату тығыздығының функциясы қалай көрінеді?
15. Гамма-тарату тығыздығының функциясы қалай көрінеді?
16. Логарифмді-қалыпты тарату тығыздығының функциясы қалай көрінеді?
17. Вейбулл тарату тығыздығының функциясы қалай көрінеді?
18. Бета-таратуы бар кездейсоқ шаманы модельдеу қалай жүзеге асырылады?
19. Гамма-таратуы бар кездейсоқ шаманы модельдеу қалай жүзеге асырылады?
20. Логарифмді-қалыпты таратуы бар кездейсоқ шаманы модельдеу қалай жүзеге асырылады?
21. Вейбулл таратуын модельдеу үшін қандай әдіс пайдаланылады?

ДИСКРЕТТІ УАҚИҒА МЕН ТАРАЛУДЫ МОДЕЛЬДЕУ

5.1. ЕРІКТІ ДИСКРЕТТІ ТАРАТУДЫ МОДЕЛЬДЕУ

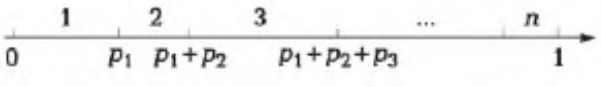
Модельдеу кезінде көптеген дискретті (есептік) мәнді қабылдайын *дискретті кездейсоқ шаманы* түрлендіруге тура келеді. Осында шаманың мысалы ойын сүйегін лақтырған кезде түскен ұпай саны, автобус жолаушыларының саны және с.с. болып табылады.

$p(X = X_i) = p_i$ таралу қатарымен берілген дискретті кездейсоқ X шамасының мүмкін $x_1, x_2, \ldots x_n$мән тізбегін алу талап етілсін делік.

X	x1	x2	…	*xn*
P(X)	p1	p2	…	*pn*

Бұл кезде шарт орындалады $\sum_{i=1}^{n} p_i = 1.$

Осындай дискреті шаманы түрлендіру үшін интервалда (0; 1) біркелкі тараған үздіксіз кездейсоқ*R*шамасын пайдаланамыз. Ұзындығы p_1, p_2, ..., p_n тең интервалды (0; 1) қарастырайық және оны *n*аралықшаларға бөлейік.Бөліну нүктелерінің координатасы $y_1 = p_1$, $y_2 = p_1 + p_2$, ..., $y_{n-1} = p_1 + + p2 + \ldots + pn\text{-}1$болады (сур. 5.1).



Сур. 5.1. Аралықшаларға бөлу мысалы

Кездейсоқ R шамасы интервалда (0; 1) біркелкі тараған, демек, R біршама интервалда болу ықтималдылығы осы интервалдың ұзындығына тең:

$$P\{0 \le R < p_1\} = p_1;$$
$$P\{p_1 \le R < p_1 + p_2\} = p_2;$$
$$\vdots$$
$$P\{p_1 + p_2 + \ldots + p_{n-1} \le R < 1\} = p_n.$$

Нәтижесінде, егер кездейсоқ Rсаны і нөмірлі интервалға түссе, онда кездейсоқ X шама х мәнін қабылдады деп есептеуге болады.

Осылайша, распределения$P_t = P(X = x_t)$ тарату қатарымен берілген дискретті кездейсоқ шаманы түрлендіру келесідей болады.

1. Интервалда (0; 1) біркелкі тараған R түрленеді.
2. $\sum_{i=1}^{I-1} p_i \le R < \sum_{i=1}^{I} p_i$. болып табылатын I анықталынады.
3. $X = I$қайтып келеді (x_i= і деп болжанады).

Мысал. k_i*мәнін қабылдайтын*ықтималдылығы бар кездейсоқ X шамасын модельдеу бағдарламасын құру.

ki	0	1	2	3	4	5
Pi	0,1	0,3	0,25	0,2	0,1	0,05

Шешімі. Қарастырылған алгоритмге сай жұмыс істейтін C# тіліндегі бағдарлама келесі түрге ие.

```
static void Main(string[] args)
{
   Console.Write("Кездейсоқ пшамасының мәлшері: ");
   int n =
   int.Parse(Console.ReadLine());
   double[] p = new double[]
      {
        0.1, 0.3, 0.25, 0.2, 0.1, 0.05
      };
   Random random = new
   Random(); double r, s; int
   x;
   for (int i = 0; i < n; i++)
   {
```

```
        r = random.NextDouble();
        s =
        0.0;
        int k;
        for (k = 0; s < r;
        k++) s += p[k]; x =
        k-1;
        Console.WriteLine("x[{0}] = {1}", i + 1,  x);
    }
    Console. WriteLine("Модельдеу аяқталды");
    Console.ReadLine();
}
```

Қарастырылған тәсіл кез-келген дискретті таралған кездейсоқ шаманы модельдеуге мүмкіндік береді. Бірақ ол таралудың кейбір жеке заңдарын модельдеу үшін барлық кезде ең тиімді болмайды. Сондықтан одан әрі ең жиі кездесетін дискретті таралу мен оларды модельдеу әдісін қарастырамыз.

5.2. БЕРНУЛЛИ ТАРАЛУЫН МОДЕЛЬДЕУ

Бернулли таралуы екі мүмкін мәні (0 және 1) бар дискретті кездейсоқ шаманы сипаттайды. Осындай кездейсоқ шаманың ықтималдылық шамасы (таралу қатары) теңдеумен сипатталады

$$p(x) = \begin{cases} 1 - p, & \text{егер } x = 0; \\ p, & \text{егер } x = 1; \\ 0, & \text{болмаған жағдайда} \end{cases}$$

мұндағыр–тарату көрсеткіші.

Бернуллидің таралуы үшін маткүтілу мен дисперсия теңдеуге сай анықталады

$$M(X) = p;$$
$$D(X) = p(1 - p).$$

Бернулли таралуы бар кездейсоқ шаманы түрлендіру үшін келесі алгоритмді қолдану болады.

1. Интервалда (0; 1) біркелкі таралуы бар R түрленеді.
2. Егер R <p болса, онда X =1 қайтып келеді. Болмаған жағдайда X = 0 қайтып келеді.

Мысал. р көрсеткіші бар Бернулли таралуына жауап беретін кездейсоқ Х шамасын модельдеу бағдарламасын құру.

Шешімі.Ертереуте сипатталған алгоритмге сай жұмыс істейтін C# тіліндегі бағдарлама келесі түрге ие.

```
   static void Main(string[] args)
   {
   Conzole^r^e("p уақиғасының пайда болуы
ықтималдылығы:    ");
   doublep = double.Parse(Console.ReadLine());
Comole^r^e("Кездейсоқ nшамасының мөлшері:");
   int n = int.Parse(Console.ReadLine());
   Random random = new Random(); double r; int x;
   for (int j = 0; j < n; j++)
   {
   r = random.NextDouble(); if (r < p) x = 1; else x =
0;
   Console.WriteLine("x[{0}] = {1}", j + 1,     x);
   }
   Console.WriteLine("Модельдеу аяқталды");
   Console.ReadLine();
```

5.3. БИНОМИАЛЬДІ ТАРАЛУДЫ МОДЕЛЬДЕУ

Биномиальды таралу Бернуллидің tсынағындағы табысты сынақ санын көрсететін дискретті кездейсоқ шаманы сипаттайды, олардың әрқайсысындағы табыс тұрақты және р тең.Биномиальды таралудың ықтималды өлшері теңдеумен беріледі

$$p(x) = \begin{cases} C_t^x p^x (1-p)^{t-x}, \text{ если } x \in \{0,1,2,...,t\}; \\ 0, \quad \text{болмаған жағдайда} \end{cases}$$

Мұндар, t–таралу көрсеткіштері.

Бернулли таралуы бар тәуелсіз және бірдей тараған tшамасының соммасы биномиальды таралуға ие болғандықтан, биномиальды таралуы бар кездейсоқ шаманы түрлендіру алгоритмі осындай.

1. р көрсеткішімен Бернулли таралуы бар кездейсоқ Y1, Y2,..., Yt,шамалар түрленеді.

2. қайтып келеді $X = \sum_{t=1}^{t} Y_t.$

Мысал. Биномиалды таратуға жауап беретін t= 10, p = 0,35көрсеткіші бар кездейсоқ X шамасын модельдеу бағдарламасын құру.

Шешімі. Ертеректе сипаталған алгоритмге сай жұмыс істейтін C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
Console Write уақиғасының қайтып келу ықтималдылығы: ");
   doublep = double.Parse(Console.ReadLine());
   Comole^r^e("Чиалонезависимыхиспытанийт:   ");
   intt = int.Parse(Console.ReadLine());
   Console.Write("Кездейсоқ пшамасының мөлшері:");
   int n = int.Parse(Console.ReadLine());
   Random random = new
   Random(); double r; int x;
   for (int j = 0; j < n; j++)
   {
      int s = 0;
      for (int i = 0; i < t; i++)
      {
         r =
         random.NextDouble(); if
         (r < p) s++;
      }
      x = s;
      Console.WriteLine("x[{0}] = {1}", j + 1, x);
   }
   Console. WriteLine ("Модельдеу аяқталды");
   Console.ReadLine ();
}
```

Биномиальды таралуы бар кездейсоқ шаманы есептеу күрделілігін азайту үшін түлкен жәнеркіші мәнінде таралудың pi= P(X= xi) қатарымен берілген дискретті кездейсоқ шаманы жалпы модельдеу алгоритмінің негізінде орындайтындығын ескерейік. Бұл кездept= = cp(1 - p)t-.

Келесі алгоритмді аламыз.

1. Интервалда (0; 1) біркелкі таралуы бар R түрленеді. *S*= 0, *к* = 0 орнатылады.

2. $S = S + p_k$ анықталынады.

3. Егер $S < R$, онда $X = к$ қайтып келеді. Болмаған жағдайда $к = к + 12$ - қабамға өтеді.

$p_к$ есептеу үшін рекуррентті қатынасты пайдалануға болатынын ескерейік.

$$p_k = p_{k-1} \frac{t-k}{k+1} \frac{p}{1-p}.$$

Бастапқы мән $p_0 = (1 - p)$.

Нәтижесінде кездейсоқ Х шамасы к интеграция санына тең мәнді қабылдайды, оны теңсіздік қанағаттану үшін орындау керек.

$$R > \sum_{i=0}^{k} p_i.$$

Мысал. Биномиалды таратуға жауап беретін n= 5 000,p = 0,01көрсеткіші бар кездейсоқ Х шамасын модельдеу бағдарламасын құру.

Шешімі.C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
          Console. Write(''p уақиғасының қайтып келу
          ықтималдылығы:                          ");
   double p = double.Parse(Console.ReadLine());
   Console. Write("Тәуелсіэт сынағының саны: ");
          int t = int.Parse(Console.ReadLine());
          Console. Write("Кездейсоқ пшамасының мәлшері:
                                                  ");
   int n = int.Parse(Console.ReadLine());
   Random random = new
   Random(); double pk, s, r;
   int x;
   for (int i = 0; i < n; i++)
   {
      pk = Math.Pow(1 - p,
      t); s = pk;
      r =
      random.NextDouble();
      int k;
      for (k = 0; r > s; k++)
      {
         pk *= (t - k) * p / (k + 1)   /  (1 - p);
         s += pk;
      }
```

```
        x = k;
    Console.WriteLine("x[{0}] = {1}", i + 1,   x);
      }
      Console.WriteLine("Модельдеу аяқталды");
      Console.ReadLine();
  }
```

5.4. ГЕОМЕТРИЯЛЫҚ ТАРАЛУЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ МОДЕЛЬДЕУ

Геометриялық таралу Бернуллидің тәуелсіз сынақ тізбегіндегі бірінші табысты сынаққа дейінгі сәтсіз сынақ санын көрсететін дискретті кездейсоқ шаманы сипаттайды, әрқайсысындағы табыс ықтималдылығы p тең. Осындай таралудың ықтимал өлшемі теңдеумен беріледі

$$p(x) = \begin{cases} p(1-p)^x, & \text{если } x \in \{0,1,2,\ldots\}; \\ 0, & \text{болмаған жағдайда} \end{cases}$$

Мұндар–таралу көрсеткіші.

Геометриялық таралуы бар кездейсоқ шама мәніне сүйеніп, осындай кездейсоқ шаманың келесі түрлендіру алгоритмін пайдалануға болады.

1. $\kappa = 0$.
2. p көрсеткіші бар Бернулли таралуымен Үтүрленеді.
3. ЕгерY= 0 болса, ондак = κ + 1 орнатылады және 2-қадамға өтеді. БолмағанжағдайдаX = ккайтып келеді.

Мысал. Геометриялық таратуға жауап беретін p = 0,1 көрсеткіші бар кездейсоқ X шамасын модельдеу бағдарламасын құру.

Шешімі. C# тіліндегі бағдарлама келесі түрге ие:

```
staticvoidMain(string[] args)
{
Console.Write(''p уақиғасының қайтып келу ықтималдылығы:");
    double p = double. Parse(Console.ReadLine());
    Console.Write("Кездейсоқ пшамасының мөлшері:   ");
    int n = int.Parse(Console.ReadLine());
    Random random = new
    Random(); double r;
```

```
    int x, k; bool Y;
    for (int i = 0; i < n; i++)
    {
       k = 0; do
       {
          r = random.NextDouble();
          Y = r < p; if (Y) k++;
       }
       while
       (Y); x =
       k;
       Console. WriteLine("x[{0}] = {1}", i + 1,    x);
    }
    Console. WriteLine("Модельдеу аяқталды");
    Console.ReadLine();
}
```

р кіші мәнінде бір кездейсоқ шаманы алу үшін орта есеппен алгоритм интеграциясының айтарлықтай үлкен саны талап етіледі, содан осындай генератор баяулайды. р кіші мәнінде геометриялық таралуы бар кездейсоқ шаманы түрлендіру үшін келесі алгоритмді пайдалануға болады.

1. Интервалда (0; 1) біркелкі таралуы бар R түрлендіреміз.
2. $X = \lfloor \ln R / \ln(1-p) \rfloor$ қайтып келеміз.

5.5. ПУАССОН ТАРАЛУЫН МОДЕЛЬДЕУ

Пуассонның таралуы уақиға бірдей қарқындылықпен пайда болғанда, уақыт аралығындағы уақиға санын көрсететін дискретті кездейсоқ шаманы сипаттайды.

Пуассон заңы бойынша тараған кездейсоқ шаманың ықтималды өлшемі теңдеумен анықталады

$$p(x) = \begin{cases} \dfrac{\lambda^x e^{-\lambda}}{x!}, & \text{если } x \in \{0, 1, \ldots\}; \\ 0, & \text{болмаған жағдайда} \end{cases}$$

Мұнда X –уақиғаның басталу қарқындылығын анықтайтын

таралу көрсеткіші (бірлік уақытындағы уақиға саны).

Модельдеуші алгоритм келесі белгілі пайымдауға негізделеді: егер кездейсоқ $y_v y_ъ$ ... шама тәуелсіз және барлығы 1 тең математикалық күтілімі бар экспоненциалды таралуы болса, онда теңсіздік орындалатын теріс емес бүтін к саны X көрсеткішімен Пуассон таралуына ие.

$$\sum_{i=1}^{k} y_i \le \lambda < \sum_{i=1}^{k+1} y_i,$$

Yi =- ln (r_i) болғанына байланысты- мұндаr–интервалда (0; 1) біркелкі тараған кездейсоқ шама, шартты төмендегі түрде жазуға болады

$$\prod_{i=1}^{k+1} r_i \le e^{-\lambda} < \prod_{i=1}^{k} r_i.$$

X көрсеткіші бар Пуассон заңы бойынша тараған кездейсоқ шаманы алу процедурасы келесіге негізделген:

1) интервалда (0; 1) біркелкі тараған тәуелсіз кездейсоқ шама r_1-r_2-...- r_nтізбегі іске асырылады;

2) шарт орындалмағанша r_1, r_1r_2, $r_1r_2r_3$... туындысы есептелінеді

$$\prod_{i=1}^{k+1} r_i \le e^{-\lambda} < \prod_{i=1}^{k} r_i.$$

Кездейсоқ X шамасының мәні ретінде к саны қабылданады.Егер теңсіздікті біркелкі тараған r_1санның біріншісі қанағаттандырса, ондах= 0.

Мысал. Пуассон таратуына жауап беретін 1=2 көрсеткіші бар кездейсоқ X шамасын модельдеу бағдарламасын құру.

Шешімі.C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
   Console.Write("Тарату параметрі Lambda:    ");
   double lambda = double.Parse(Console.ReadLine());
   Console.Write(''Кездейсоқ nшамасының мөлшері:
                                                ");
   int n = int.Parse(Console.ReadLine());
   Random random = new Random();
   double r, pr;int x;
for (int i = 0; i < n; i++)
```

```
    {
        r = random.NextDouble(); pr = r; int
        k;
        for (k = 0; pr > Math. Exp(-lambda); k++)
        {
            r =
            random.NextDouble();
            pr *= r;
        }
        x = k;
        Console.WriteLine("x[{0}] = {1}", i + 1,  x);
    }
    Console.WriteLine("Модельдеу аяқталды");
    Console.ReadLine();
}
```

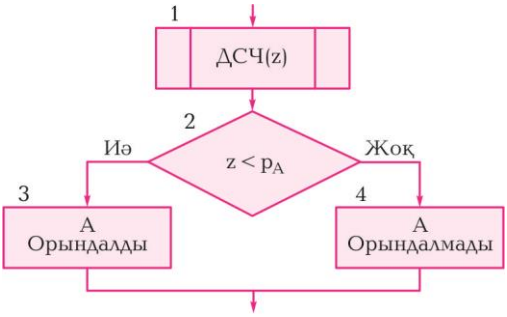
5.6. ҚАРАПАЙЫМ УАҚИҒАНЫ МОДЕЛЬДЕУ

Модельдеу кезінде алдын-ала белгілі ықтималдылықпен кездейсоқ уақиғаны жиі имитациялау талап етіледі. *Кездейсоқ* деп сынақ уақытында болатын немесе болмайтын белгілі бір шарт жинтығы кезіндегі уақиғаны айтады [2]. Көптеген мүмкіннің ішінен әр уақиғаға уақиға ықтималдылығы сай келеді. Міндетті түрле орын алуы керек нақты уақиғаның ықтималдылығы бірлікке тең. Мүмкін емес уақиғаның ықтималдылығы нөльге тең. Жеке уағиғаны модельдеу әдісін қарастырайық.

Орын алу ықтималдылығы p_A(мысалы, мерген нысанаға 0,85 ықтималдылықпен тиеді) тең А уақиғасы бар делік. Көп рет пайдалану кезінде А уақиғасының орын алу жиілігі оның ықтималдылығына ұмтылатын ережені жасап шығару талап етіледі.

Есептегіш көмегімен интервалда (0; 1) біркелкі тараған кездейсоқ санды, бірқатар z санын таңдаймыз және$z<p_A$.ықтималдылығын анықтаймыз. Интервалда (0; 1) біркелкі тараған кездейсоқ z шамасы үшін келесі тәуелділік дұрыс:

$$P(z < p_A) = \int_{-\infty}^{p_A} f(x)\mathrm{d}x = p_A.$$



Сур. 5.2. Қарапайым уақиғаны модельдеу алгоритмі

Осылайша, кездейсоқ zшамасының интервалға (0; 1) тию ықтималдылығырАтең. Сондықтан, егер кездейсоқ шаманы тарату есептегішінің көмегімен алынған z саны осы интервалға түссе, онда А уақиғасы орын алды деп санауға болады. Қарама қарсы уақиға (А емес) (1 - pA) ықтималдылығымен орын алады, яғни егер z>pA болған жағдайда.

Имитациялық модельде қарапайым уақиғаны модельдеу процедурасы алгоритммен сипатталады, оның сызбасы 5.2 суретте көрсетілген.

Осы суретте 1 блокта кездейсоқ z шаманы түрлендіретін кездейсоқ сан есептегішіне КСЕ жүгіну жүзеге асырылады. 2 блок z <pAшартын тексереді. Егер ол орындалса, А уақиғасы орын алды деп есептейді. Болмаған жағдайда, қарама қарсы уақиға (А емес) орын алды деп саналады.

C# тіліндегі А уақиғасының инитациясына арналған осы алгоритмді пайдаланатын бағдарлама келесідей болады:

```
   staticvoidMain(string[] args)
   {
   Console.Write("N тәжірибе саны:  ");
   int n = int.Parse(Console.ReadLine());
   Console.Write("A уақиғасының пайда болу
ықтималдылығы:    ");
   double pA = double.Parse(Console.ReadLine());
Random random = new Random(); for (int i = 0; i <n;
i++)
   {
```

```
   double r = random.NextDouble(); if (r < pA)
   //A уақиғасы орын алған кездегі әрекет
   else

   //A уақиғасы орын алмаған кездегі әрекет
   Console. WriteLine("Модельдеу аяқталды");
Console.ReadLine();
```

5.7. БІРЛЕСПЕГЕН УАҚИҒАЛАРДЫҢ ТОЛЫҚ ТОБЫН МОДЕЛЬДЕУ

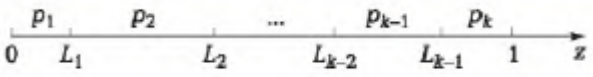
Бірлеспеген A_1, A_2, ...A_k.уақиғалардың толық тобы бар делік. Осы термин ешқандай екі және одан да көп уақиға бір уақытта (яғни бір сынақта) орын ала алмайды, бірақ әр сынақта олардың міндетті түрде біреуі орын алады. A_1, A_2, ..., A_kуақиғаларының ықтималдылығы сәйкесіншерj, p_2, ..., p_kтең делік. Бұл кезде шарт орындалады.ъ

$$\sum_{i=1}^{k} p_i = 1.$$

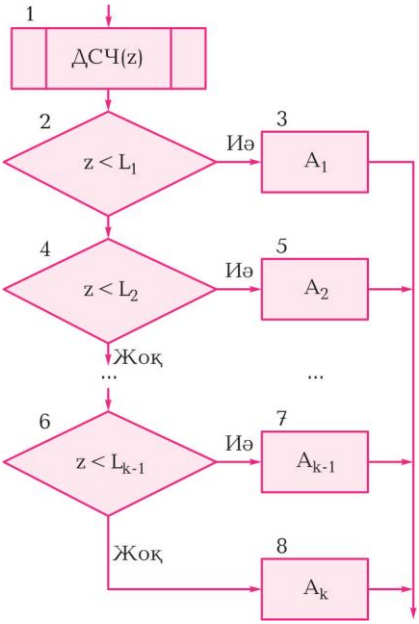
Бірлеспейтін уақиғаның толық тобын модельдеу үшін интервалды (0; 1) *к*қиықтарға бөлеміз, олардың ұзындығы p_j, P_2^r.$Pк$ (сур. 5.3)құрайды.

Егер интервалда (0; 1) біркелкі тараған кездейсоқ сан есептегішімен түрленген кездейсоқz саны, мысалы, *p* аумағына түссе, онда А уақиғасы орын алды деп санауға болатындығын білдіреді.

Онда бірлеспеген уақиғалардың толық тобын модельдеу процедуралары алгоритммен сипатталынады, оның сызбасы 5.4 суретте көрсетілген. Мұнда $L_i = \sum_{j=1}^{i} p_j$.



Сур. 5.3. Аралықшаларға бөлуді көрсету



Сур. 5.4. Бірлеспеген уақиғаның толық тобын модельдеу алгоритмі

1 блокта интервалда (0; 1) біркелкі тараған кездейсоқ сан есептегішіне жүгіну жүреді. Шартты 2 оператор кездейсоқ z шамасының интервалға[0; L_j)түсу шартын тексереді. Егер бұл шарт орындалмаса, онда A_1уақиғасын орын алды деп саналады. Егер 2 операторлағы шарт орындалмаса, онла кездейсоқ шаманың басқа интервалдарға түсу шартын тексереді. A_1, A_2, ..., A_k ішінен бір уақиға міндетті түрде орын алады.

Мысал. Тирде нысанаға тию кезінде 0-ден 5 ұпайға дейін алуға болады делік. Мерген бір атып к ұпай алады деген рықтималдылық одан әрі берілген.

к	0	1	2	3	4	5
P	0,1	0,15	0,2	0,25	0,2	0,1

Бес атудан топтама модельдеу және мерден 5 атуда 15 ұпайдан көп алатын ықтималдылықты бағалау.

Шешімі. Алгоритм сызбасына сай әзірленген, 5.4 суретте ұсынылған C# тіліндегі бағдарлама келесі түрге ие:

```
static void Main(string[] args)
{
Console. Write(N тәжірибе саны:    ");
int n = int.Parse(Console.ReadLine()); double[] L =
new double[]
{// бөлу интервалының шегін береміз 0.1, 0.25, 0.45,
    0.7,  0.9,  1.0 };
Random random = new Random(); int m = 0;
for (int i = 0; i < n; i++)
{
int si = 0;
for (int j = 0; j < 5; j++)
{
double r = random.NextDouble();
if (r < L[0]) si += 0;
else if (r < L[1]) si ++;
else if (r < L[2]) si  += 2
else if (r < L[3]) si  += 3
else if (r < L[4]) si  += 4
else si += 5;
}
if (si > 15) m++;
}
Console. WriteLine(''Ізделіп отырған ықтималдылықр =
    {0:f4}",
(float) m / n);
Console. WriteLine("Модельдеу аяқталды");
Console.ReadLine();
}
```

5.8. КҮРДЕЛІ УАҚИҒАНЫ МОДЕЛЬДЕУ

Бір біріне еретін бірнеше уақиғаның көрінуі болуы мүмкін. рАжәнерВықтималдылықпен екі тәуелсіз А және В уақиғаның жүру жағдайын қарастырайық.Бұл жағдайда сынақтың мүмкін нәтижесі АВ, $\bar{A}$В, А$\bar{B}$ және $\bar{A}\bar{B}$.АВ, $\bar{A}$В, А$\bar{B}$ және $\bar{A}\bar{B}$ уақиғалары ықтималдылығы бар бірлеспеген уақиғаның толық тобын түзеді

$$P(AB) = p_A p_B;$$

$$P(A\bar{B}) = p_A(1 - p_B);$$

$$P(\bar{A}B) = (1 - p_A)p_B;$$
$$P(\bar{A}\bar{B}) = (1 - p_A)(1 - p_B). \quad (5.1)$$

Бірлескен сынақтарды модельдеу үшін процедураның екі нұсқасы қолданылуы мүмкін.

Бірінші нұсқаны ықтималдылықпен (5.1) сәйкес нәтиженің бірін анықтау ретінде бірлеспеген уақиғаның (5.4 суреттегі алгоритм) толық тобына ұқсас құруға болады. Бұл жағдайда бір кездейоқ санмен шектелуге болады, бірақ жалпы жағдайда салыстыруға көп талап етілуі мүмкін.

Екінші нұсқа А, В уақиғаларына қатысты белгіленген ықтималдылықпен (5.2 суреттегі алгоритм) қарапайым уақиғаны ұқсас модельдеу шарттарын тізбекті тексеруден тұрады және екі кездейсоқ сан мен екі салыстыруды пайдалануды талап етеді. Орташа алғанда екінші нұсқа біріншісіне қарағанда тиімдірек болып көрінеді.

Енді А және В уақиғалары тәуелді және p_A, p_Bжәне$p_{B/A}$ белгілі болып табылатын жағдайды қарастырайық.

Модельдеу процедурасының бірінші нұсқасы үшін тағы да ескерейік АВ, АВ АВ және АВ уақиғалары толық топты құрайды. Олардың ықтималдылығын келесідей табуға болады*:*

$$P(AB) = p_A p_{B/A};$$
$$P(A\bar{B}) = p_A(1 - p_{B/A});$$
$$P(\bar{A}B) = (1 - p_A)p_{B/\bar{A}};$$
$$P(\bar{A}\bar{B}) = (1 - p_A)(1 - p_{B/\bar{A}}),$$

мұндағы

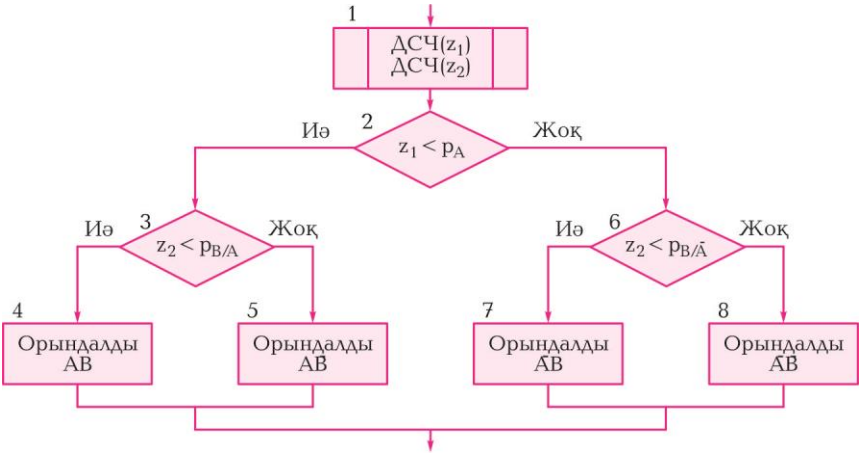
$$p_{B/\bar{A}} = \frac{p_B - p_A p_{B/A}}{1 - p_A}$$

РАотолық ықтималдылық формуласынан анықталады

$$p_B = p_A p_{B/A} + p_{\bar{A}} p_{B/\bar{A}}.$$

Бұл жағдайдағы процедураның екінші нұсқасы келесідей көрінеді. Біркелкі тараған сандар тізбегінен {z}кезектізkсаны алынады және теңсіздіктің әділдігі тексеріледі

$$z_k < p_A. \quad (5.2.)$$



Сур. 5.5. Тәуелді уақиғалардың модельдеу алгоритмі

Егер ол әділ болса, онда А уақиғасы орын алды. Жиынтықтан {zi}кезекті zk+tсан алынады және шарт тексеріледі

$$z_{k+1} < p_{B/A}.$$

Оның сынақ қорытындысының әділдігіне тәуелділігі АВнемесеАВболып табылады.

Егер теңсіздік (5.2) орындалмаса, онда бұл А уақиғасы орын алды дегенді білдіреді. Сондықтан В уақиғасымен байланысты сынақ үшін pBjx ықтималдылығын қолдану қажет. {zi}жиынтықтан zk+1санын таңдайық және теңсіздіктің әділдігін тексереміз

$$z_{k+1} < p_{B/\bar{A}}.$$

Бұл теңсіздіктің әділ немесе әділ еместік нәтижесіне байланысты сынақта ĀВнемесеĀВаламыз.

Бір сынаққа арналған екі тәуелді уақиғаны имитациялау процедураларының екінші нұсқа алгоритмі сур. 5.5 блок-сызбасында бейнеленген.

Айта кететін жайт, егер модельдеу кезінде тәуелді уақиғаның көп санының пайда болуын имитациялауды талап етілсе, онда имитация процедурасының екінші нұсқасы іске асыру үшін қарапайым болып табылады.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. Дискретті және үздіксіз тараған кездейсоқ шама арасындағы айырмашылық неде?
2. Тарату қатарымен берілген дискретті кездейсоқ шаманы модельдеу қалай іске асырылады?
3. Бернулли таралуы қалай беріледі? Сипаттау үшін Бернулли таралуын қолдануға болатын кездейсоқ шама мысалдарын келтіріңіз. Олардың қалай модельдейміз?
4. Биномиальды таралу қалай беріледі? Сипаттау үшін биномиальды таралуын қолдануға болатын кездейсоқ шама мысалдарын келтіріңіз. Олардың қалай модельдейміз?
5. Геометриялық таралу қалай беріледі? Сипаттау үшін геометриялық таралуды қолдануға болатын кездейсоқ шама мысалдарын келтіріңіз. Олардың қалай модельдейміз?
6. Пуассон таралуы қалай беріледі? Сипаттау үшін Пуассон таралуын қолдануға болатын кездейсоқ шама мысалдарын келтіріңіз. Олардың қалай модельдейміз?
7. Бірлеспеген уақиғаның толық тобын модельдеу қалай жүзеге асырылады?
8. Қандай уақиғалар тәуелді деп аталады? Оларды қалай модельдейміз?

СТАТИСТИКАЛЫҚ СЫНАҚТАР ӘДІСІНІҢ КӨМЕГІМЕН МОДЕЛЬДЕУ

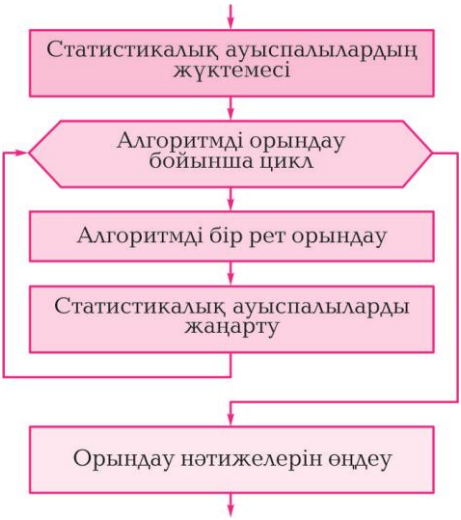
6.1. МОНТЕ-КАРЛО ӘДІСІ

Стохостикалық жүйелердің имитациялық модельдерін құру және іске асыру процесінде статистикалық сынақтар (Монте-Карло) әдісі кеңінен қолданылады. Монте-Карло әдісі — сандық әдістер тобының жалпы атауы, осылайша қалыптасқан стохастикалық (кездейсоқ) процестің іске асырылуының көп мөлшерін алуға негізделген, оның ықтималдық сипаттамалары шешілетін мәселенің ұқсас мәндеріне сәйкес келеді. Әдіс жиі физика, химия, математика, экономика, оңтайландыру, басқару теориясы және т.б. салаларындағы мәселелерді шешу үшін қолданылады.

Монте-Карло әдісінің идеясы келесідей: зерттелген кездейсоқ процессті аналитикалық түрде сипаттаудың орнына, бұл процесті имитациялауға арналған алгоритм құрылады. Алгоритмде кездейсоқтықты модельдеудің арнайы процедуралары бар. Алгоритмнің нақты іске асырылуы оның нәтижелерімен әр уақытта әр түрлі салынады. Математикалық статистиканы қолдана отырып, алгоритмнің көптеген іске асыру нәтижелерін өңдеп, кез-келген сипаттаманы алуға болады: оқиғаның ықтималдығы, математикалық күтілім, кездейсоқ айнымалы дисперсиялар. Сынақтар санының функциясы бұл сипаттамалардың шын мәніндегі айырмашылығы артық белгіленген көлемнен аспайтын ықтималдығы.

Монте-Карло әдісін іске асыратын алгоритмнің жалпыланған схемасы 6.1 суретте көрсетілген. Алгоритмнің бірінші кезеңінде статистикалық айнымалыларды инициализациялау жұмыстары орындалады. Бұл айнымалыларда, мысалы, оқиғаның пайда болу ықтималдығын бағалау кезінде біз үшін қызықтыратын оқиға орын алған бірқатар эксперименттер жиналады.

Содан кейін цикл ұйымдастырылады, оның органында алгоритмнің жеке орындалуы іске асырылады (мысалы, оқиғаны модельдеу орындалады) және статистикалық айнымалылар жаңартылады (мысалы, егер оқиға орын алған болса, тиісті статистикалық айнымалы 1-ге көбейтіледі). Алгоритмнің соңғы блогында модельдеудің нәтижелер



Сурет 6.1.Монте-Карло әдісі бойынша жалпыланған алгоритмін модельдеу

өңдеу орындалады.Бұл жағдайда, мысалы, бізді қызықтыратын оқиғаның ықтималдығы бағалануы мүмкін.

Келесі мысалдармен статистикалық модельдеу әдісінің мәнін түсіндірейік.

Мысал . πсанының мағынасын табу.

Шешуі. Квадраттың қабырғалары r, мысалы, қағаз парағына салынған, 6.2.суретте көрсетілген.Оның ауданыS1= r2. Квадраттың S2= πr 2/4 ауданы радиусы r шеңбердің төрттен бірін құрайды.Квадрат пен шеңбердің төрттен бір аудандарының қатынасы S2/S1= π/4 тең.

Бұл қатынастар, демек, π саны келесі статистикалық сынақтарды орындау арқылы шамамен алуға болады. Үстелде жатқан қағаз бетіне кездейсоқ біркелкі шашылатындай майда түйіршіктерді лақтыратын боламыз. Квадраттың сыртындағы түйіршіктерді ескермеймиз. Есептейміз N1— түйіршіктерсаны, квадратқа түскен, жәнеN2— шеңбердің төрттен бір шегіндегі түйіршіктер саны.Себебі түйіршіктер суреттің кез келген бөлігіне ену ықтималдығы бар, ондаN2/N1қатынастары,лақтырылғандардың жеткілікті улкен саны бар аудандардың қатынасына шамамен тең болады, яғни. π/4.

π санының табу процессін кішкене өзгертуге болады. Ол үшін квадрат қабырғаларының бойымен координаталық өсін жүргіземіз. Масштабты квадраттың қабырғалары бірге тең болатындай етіп таңдаймыз. Түйіршіктермен бірге осы квадратқа кездейсоқ координаталы (x, y)нүкте-лерді түсіреміз. Кездейсоқ координаталар

арқылы бірлік кесіндіні біркелкі тарату сандар деп түсінеміз. Шеңбердің ішіндегі нүктенің π/4 ауданының қатынасына тең ықтималдығы. r санын табу үшін алгоритм құрамыз. Ақиқат бойынша x және y координаттары үшін, бір сынақ (біреуін іске асыру) x2+ y2< 1 теңсіздіктерін тексеруден тұрады (6.3 сурет).

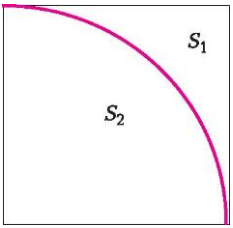
Оны Npet қайталаймыз, сондаN2— нүктелер сан, ол үшін теңсіздік орындалады, және N2/N — бұл оқиғаның жиілігі, бұл π санының мағынасын табуға болатынын біле отырып,

π санының есептеу нәтижесі, бағдарлама көмегі бойынша алынған,суреттелген алгоритмды іске асыру, N нүктелерінің әртүрлі саны үшін 6.1кестеде келтірілген.Монте-Карло әдісінің көмегімен тек қанаπ санының шамамен алынған мәнін ала аламыз және алынған нәтижелердің қателігі неғұрлым аз эксперимент жасалатынын ескереміз.

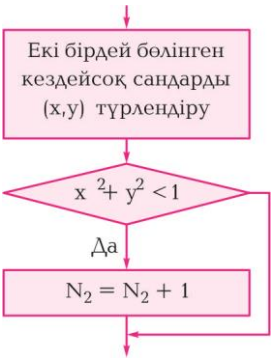
Сонымен қатар, модельдеу нәтижелерін кездейсоқ сандар генераторының сапасына айтарлықтай ықпал ететінін атап өтеміз. Бұл анықталады, егер де нүктелер квадраттың ішінде біркелкі болмаса, онда көптеген эксперименттермен де модельдеу нәтижелері дұрыс болмайды.

Мысал . А және В қатысушылар арасында жекпе-жек болып жатыр. Қатысушы Ар = 0,9 ықтималдылығымен бірінші болып атады.Ату кезінде қатысушы АрА= 0,5 ықтималдылығымен В-ға, ал қатысушы ВА-ға рВ= 0,8 ықтималдылығыментигізеді.Қатысушылар бір рет ғана ата алады. А, В қатысушыларының тең түсу және жеңу мүмкіндігінің ықтималдығын бағалау.

Шешуі.Бұл тапсырма өте оңай болғандықтан, біз қажетті ықтималдықтарды аналитикалық есептеуді орындай аламыз:



Сур. 6.2. π санын табуға арналған суретемме



Сур. 6.3. Теңсіздікті тексеруді жүзеге асыру

Кесте 6.1. p санының мәнін анықтау үшін эксперименттердің нәтижелері	
Нүктелер саны(N)	p мәні
100	3,2000
1000	3,1120
10000	3,1140
100000	3,1374
1000000	3,1391
10000000	3,1409

$$P(A) = pp_A+(1-p)(1-p_B)\ p_A=0,46;$$
$$P(B) = p(1-p_A)\ p_B +(1- p)\ p_B =0,44;$$
$$P(H) = (1-p_A)(1- p_B) =0,1;$$

Бұл нәтижелер модельдеу нәтижелерінің дұрыстығын тексеру үшін пайдалануға болады.

Модельдеу кезінде қатысушылар арасында жеке-жеке оқиғаларын модельдеу (бірінші атқанда таңдау, бірінші атуды түсіру, екінші атуды түсіру) және нәтиженің түрін анықтау (А қатысушысының, қатысушы В жеңісі немесе тең болуы) үшін жекпе-жекті бірнеше рет ойнау керек.

Бірнеше эксперименттерді орындағаннан кейін, әрбір түрдегі оқиғалар саны есептеледі, содан кейін сәйкес оқиғалардың ықтималдығы бағаланады.

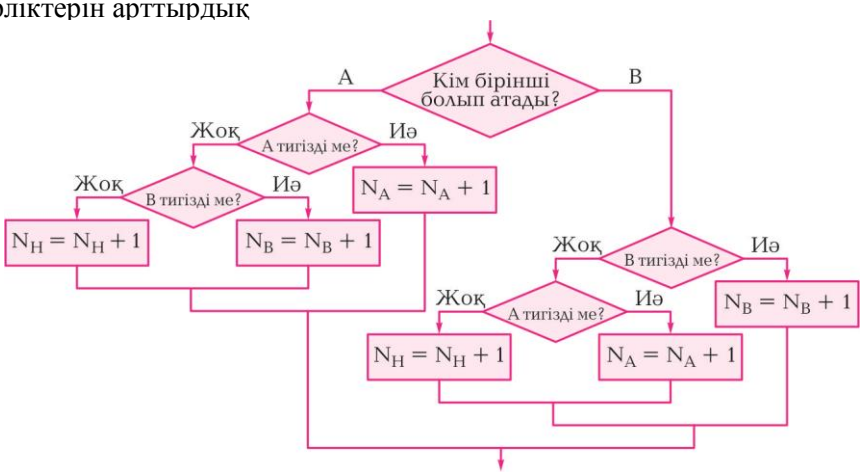
Алгоритмді шешу есебі 6.1 кестедегі жалпы сұлбада көрсетілген. Бір эксперементтің ойнату алгоритмі 6.4 суретте көрсетілген. Бұл алгоритмде үш статистикалық есептеуіш қолданылған (NA, NB, NH), қатысушы А, қатысушы В жеңуі санын және тең түсу санын есептеуді және статистикалық айнымалы мәндерді инициализациялау кезеңінде қолданылады.

Берілген алгоритмде жұмыс бастамастан бұрын оқиғаны медельдеу қажет, бірінші атушының таңдауымен аяқталатын.

Осындай қарапайым оқиғаны модельдеу алгоритмі 5.6.бөлімде қарастырылған. А қатысушысы бірінші болып атқан жағдайда, оның атқаны модельдендіріледі. Бұл әрекетті орындау үшін, қатысушы А-ға қатысушы А жеңіліске ұшырады (ықтималдығы 1 - pA) немесе қатысушы А-ға (егер ықтималдығы бар) соққы берсе, онда оқиға А қатысушысы ойнап, жеңістердің санын көбейтеді.

Егер де А ойыншымұлт кетсе, онда Войыншының атқанын модельдеу қажет. Егер де ол тигізсе, онда Вқатысушысының жеңу саны көбейеді. Кері жағдайда тең үсу болды, және біз есептегіш

бірліктерін арттырдық



Сур. 6.4. Бір тәжірибелік ойынның алгоритмі

NH. Егер В қатысушы бірінші атқан жағдайд , осыған ұқсас болған жағдайда дамиды.

N жекпе-жегі ойнатылғаннан кейін, сіз төмендегі өрнектерді пайдаланып қажетті оқиғалардың ықтималдығын бағалай аламыз:

$$P(A) = N_A/N$$
$$P(B) = N_B/N$$
$$P(H) = N_H/N$$

Қарастырылатын жүйені модельдеуді қолмен жасаймыз. 20 эксперемент (жекпе-жек) бойыншаалынған нәтижелерді, СЛЧИС() MS Excelфункциясы арқылы қалыптастырылған кездейсоқ сандар эксперименті 6.2. кестеде көрсетілген.

А жеңу ықтималдығын бағалау үшін ол қанша рет жеңіп алғанын (бұл жағдайда 7) есептеп және соманы эксперименттердің жалпы саны бойынша бөлу (20).Нәтижесінде А қатысушысын жеңу ықтималдығы бағасы 0,35 теңдігін аламыз.Осындай жолмен біз В қатысушысының жеңуін бағалай аламыз (0,50 аламыз) және тең түсу ықтималдығы (0,15 аламыз). Бұл бағалау ықтималдықтардың нақты мәндеріне ұқсас, бірақ эксперименттің аз мөлшеріне байланысты нәтижелердің қателігі өте жоғары.

Модельдеу қателігін төмендету үшін, 8 тараудағы эксперимент көлемін ұлғайту керектігін көрсетеді.Монте-Карло әдісін қолдануға қызығушылық танытқан ықтималдықтарды бағалауды жүзеге асыратын C# тіліндегі бағдарламасының мысалы, келесі түрде болады:

Кесте 2.Монте-Карло әдісімен модельдеу мысалы

Эксперимент нөмірі	Кездейсоқ сан	Кім бірінші атады	Кездейсоқ сан	Нәтижесі	Кездейсоқ сан	Нәтижесі	Кім жеңді
1	0,3216	A	0,3401	Тигізді	—	—	A
2	0,1922	A	0,3069	Тигізді	—	—	A
3	0,6360	A	0,5904	Мүлт кетті	0,7010	Тигізді	*B*
4	0,6234	A	0,7095	Мүлт кетті	0,9393	Мүлт кетті	Тең түсті
5	0,0983	A	0,5505	Мүлт кетті	0,7170	Тигізді	*B*
6	0,1811	A	0,6871	Мүлт кетті	0,0626	Тигізді	*B*
7	0,0562	A	0,6631	Мүлт кетті	0,6491	Тигізді	*B*
8	0,3536	A	0,4897	Тигізді	—	—	*A*
9	0,4534	A	0,0713	Тигізді	—	—	*A*
10	0,4586	A	0,5352	Мүлт кетті	0,0795	Тигізді	*B*
11	0,2774	A	0,2644	Тигізді	—	—	*A*

6.2.кетенің жалғасы

Эксперимент нөмері	Кездейсоқ сан	Кім бірінші атады	Кездейсоқ сан	Нәтижесі	Кездейсоқ сан	Нәтижесі	Кім жеңді
12	0,0836	A	0,6348	Мүлт кетті	0,8685	Мүлт кетті	Тең түсті
13	0,8119	A	0,5507	Мүлт кетті	0,3903	Тигізді	*B*
14	0,1563	A	0,4240	Тигізді	—	—	*A*
15	0,1373	A	0,6166	Мүлт кетті	0,0428	Тигізді	*B*
16	0,5001	A	0,0578	Тигізді	—	—	*A*
17	0,9247	*B*	0,8491	Мүлт кетті	0,4729	Тигізді	*B*
18	0,4176	*A*	0,7501	Мүлт кетті	0,9962	Мүлт кетті	Тең түсті
19	0,5981	*A*	0,7187	Мүлт кетті	0,0926	Тигізді	*B*
20	0,4908	*A*	0,5251	Мүлт кетті	0,6681	Тигізді	*B*

```
static void Main(string[] args)
{
   Console Write ("А ойыншы бірінші болу
   ықтималды:      ");
   doublep=
   double.Parse(Console.ReadLine());
   Comole^r^e("А ойыншының тигізу
   ықтималдыdoublepA=
   double.Parse(Console.ReadLine());
   Console.Write(''Еойыншы тигізу
   ықтималдылығыdoublepB =
   double.Parse(Console.ReadLine());
   Console.Write('' эксперимент саны N:")
   int n = int.Parse(Console.ReadLine());
   Random random = new Random(); int nA = 0,
   nB = 0, nH = 0; double r;
   for (int i = 0; i < n; i++)
   {
      r = random.NextDouble(); if (r < p)
      { // Аойыншы бірінші атады
        r = random.NextDouble(); if (r < pA)
        nA++; else {
          r = random.NextDouble(); if (r <
          pB) nB++; else nH++;
        }
      }
      else
      { // Войыншы бірінші атады
        r = random.NextDouble(); if (r < pB)
        nB++; else {
          r = random.NextDouble(); if (r <
          pA) nA++; else nH++;
        }
      }
   }
   Console.WriteLine("ізделінген
   ықтималдылық"); Console.WriteLine("pA =
                        {0:f4}",(double)nA/n)
   Console.WriteLine("pE =
                        {0:f4}",(double)nE/n)
   Console.WriteLine("pH =
                        {0:f4}",(double)nH/n)
   Console. WriteLine("Модельдеу аяқталды");
   Console.ReadLine();
```

6.2. БІР ӨЛШЕМДІНІҢ КЕЗДЕЙСОҚ КЕЗУІ

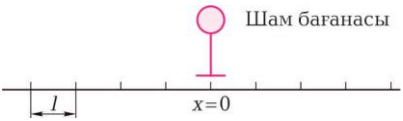
Физика, биология, химия, экономика және көптеген басқа ғылымдардағы әртүрлі процестерді зерттеу барысында пайда болатын кездейсоқ кезуі тапсырмаларын қарастырамыз.Осындай тапсырмаларды статистикалық сынақтар әдісінің көмегімен шешуге болады.

Шамдық бағана қозғалысынан басталатын, x = 0 нүктесіне орналасқан (6.5.сурет.), мастық жаяу жүргіншінің қозғалысының бір өлшемді ең қарапайым мәселесі. Жаяу жүргіншінің барлық қадамдары бірдей ұзақтыққа 1 ие. Жаяу жүргіншінің әр қадамының бағыты алдыңғы бағыттың бағытына байланысты емес. Жаяу жүргінші рықтимадылықпен оңға және q = 1- p ықтималдылықпен солға қадам жасайды. Бұл тапсырмада M сатыдан кейін жаяу жүргінші бағаналы шамдардан x қашықтықта болады немесе M сатыдан кейін бастапқы нүктеден өтетін орташа жаяу жүргіншіні ықтималдығын анықтау қажет.

Бір өлшемді кездейсоқ кезуі проблемасын зерттеу ықтималдық теориясын пайдалана отырып аналитикалық түрде жүргізілуі мүмкін. Бұл жағдайда алынған нәтижелер имитациялық модельді түзету және оның дұрыстығын тексеру барысында пайдаланылуы мүмкін. Орташа ауытқур и дисперсия ст\келесі аналитикалық өрнектерді пайдалана отырып есептеуге болады:

$$\mu = (p - q)Ml;$$
$$\sigma_x^2 = 4pqMl^2.$$

Ары қарай Monte Carlo әдісімен бір өлшемді кездейсоқ кезу модельдеу үшін C # тіліндегі бағдарламасының фрагменті, нәтижесінде жаяу жүргіншінің бастапқы нүктесінен ауытқу диспер-циясы L мен математикалық күтілу бағаланады. Нәтижені алу үшін әрқайсысының бір жаяу жүргінші қозғалысын имитациялайды және оның бағаналы шамнан кейінгі қадамдарының m ауытқуын айқындайтын бірқатар эксперименттер жүргізіледі. Жасалынған экспериметтің нәтижелері жойылады.



6.5.сурет.Мастық жаяу жүргінші қозғалысындағы тапсырма суреттемесі

Айта кету керек, бұл жағдайда бізде тек шамамен алынған нәтижелер,ал

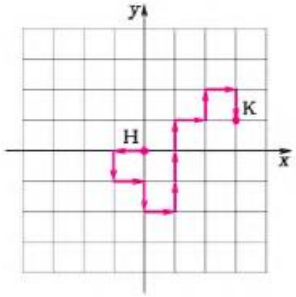
нақтысы эксперименттер санымен анықталады.

```
// n – сынақтар саны
// p – оңға қадам ықтималдылығы
// m – қадам саны
Random random = new Random();
x1 = 0.0;
x2 = 0.0;
for (int i = 0; i < n; i++)
{
   x = 0.0;
   for (int j = 0; j < m; j++)
   {
     double r =
     random.NextDouble(); if (r
     < p) x++; else x     ;
   }
   x1 += x; x2
   += x * x;
}
L = x1 / n;
D = x2 / n – L * L;
```

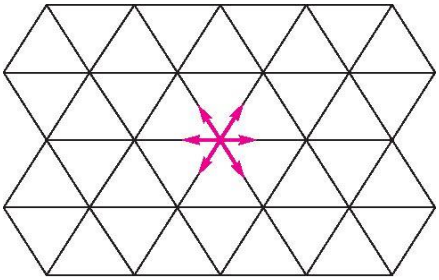
6.3. ЕКІ ӨЛШЕМДІ КЕЗДЕЙСОҚ КЕЗУІ

Көп өлшемді кеңістіктегі кезуі қиынырақ болып келеді. Мысалы, тікбұрышты торда екі өлшемді кездейсоқ кезуі үшін уақыттың әр кезеңінде кездейсоқ қозғалыс төрт ықтимал бағытта: солтүстік, оңтүстік, шығыс немесе батыс жолдарында теңдестірілген бір бағыт түрде жүзеге асырылады (6.6сурет). Қозғалыс бастапқы H нүктесінен басталады,әдетте бастапқы координаталарымен (x = 0; y = 0) сәйкес келеді, және K нүктесінде Мқадамы арқылы яқталады. Бұл жадайда көбінесе K нүктесінің өшіруін анықтап алу қажет,бастапқы координатадан M қадамынан кейін алынған.

Тор басқа пішінде болуы мүмкін екенін ескерміз. Мысалы, үшбұрышты тор кездейсоқ кезуі тапсырмаларда кездеседі. Бұндай қозғалыстың әрбір қадамында кезуі алты мүмкін бағыттардың бірінде тең ықтималдылық орындалады(6.7.сурет).



6.6. сурет. Торлы екі өлшемді кездейсоқ кезуі тапсырмасына суреттеме.

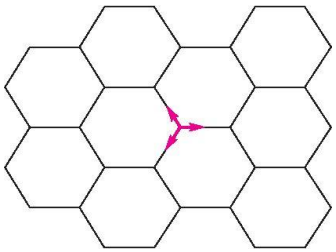


6.7.сурет Үшбұрышты екі өлшемді кездейсоқ кезуі тапсырмасына суреттеме

Жасушаларда кездейсоқ кезуі тапсырмасы белгілі. Әрбір қадамда осындай кезуі арқылы қозғалыс үш ықтимал бағыттардың бірінде тең дәрежеде жүзеге асырылады (6.8 сурет).

C# тіліндегі бағдарлама фрагментті,тік төртбұрышы торда екі өлшемді кездейсоқ кезуі имитацияланады және бастапқы координатадан K қадам m жасалғаннан орташа өшірі L нүктесімен анықталады, бұл келесі түрде болады:

```
// n – сынақтар саны
// m – қадам саны
Random random = new Random();
double L = 0;
for (int i = 0; i < n; i++)
{
   x = y = 0;
```



6.8.сурет.Жасушадағы кезуі тапсырмаға суреттеме.

```
for (int j =  0; j < m; j++)
   {
     double r =
     random.NextDouble(); if (r
     <0.25) x++; else if (r
     <0.5) x—; else if (r
     <0.75) y—; else y++;
   }
   L += Math.Sqrt(x * x + y *  y);
}
L /= n;
```

6.4. СІҢІРМЕЛІ ЭКРАНДА ҚАРАПАЙЫМ КЕЗДЕЙСОҚ КЕЗУІ

Бұрын қаралған кезуі тапсырмаларының аумағында ештеңеден шектелмеген, яғни жаяу жүргіншілер (бөлшектер) бастапқы нүктеден кез-келген қашықтыққа шығуы мүмкін еді.Бірқатар практикалық тапсырмаларда мұндай шектеулер бар. Ойыншының күйреуінің классикалық тапсырмасы мысал бола алады. Онда екі адам бастапқы қаржымен a және bрублі бар, ойын тастармен ойнайды. әрбір ойын тастарын лақтырғаннан кейін бір ойыншы жеңіп 1 рубль алады,ал екінші ойыншы 1 рубль жеңіледі. Олардың біреуінің қаржысы таусылғанша орташа қанша уақыт ойнайтынын білу қажет. Бұл тапсырма сіңірмелі экранда қарапайым кездейсоқ кезуі сипаттамасына ие.

Торда бір өлшемді тапсырма берілгенде,олардың x = -a и x = b (b > -a нүктелерінің түйіндерінде (экраны) «сіңірмелі» бар. Қозғалыс бөлшектері x0 (-a <x0<b) нүктесінен басталады. Әрбір қадамда бөлшек рықтималдылықпен солға және q = 1 – p ықтималдықпен оңға айналады.Қадамының ұзындығы бірге тең. Кез келген бөлшек экранға жеткенде ол сіңіріледі. Бөлшектің орташа кезуі уақытын білу қажет.

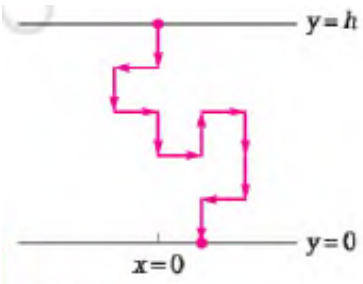
Бұл жағдайда C# тіліндегі бағдарлама фрагменті модельдеу орташа кезуі уақытының T анықтамасы келесі түрге ие:

```
   // n — сынақтар саны // a, b — экраның орналасқан
жері // x0 —бастапқы орналасқан жері
   // p — оңға қадам ықтималдылығыRandom random = new
Random(); double T = 0;
```

```
   for (int i = 0; i < n; i++)
   {
   double x = x0; int time = 0;
   while ((x != a) && (x != b))
   {
   double r = random.NextDouble(); if (r < p) x++; else
x–; time++;
   }
   T += time;
   }
   T /= n;
```

6.5. ЖАУЫН ТАМШЫСЫНЫҢ ТҮСУ МОДЕЛІ

Кездейсоқ кезуі моделінің тағы да бір нұсқасы шектеулі жаңбыр тамшысының моделі болып табылады.Жеңіл желдің кездейсоқ екпінің әсерінің арқасында, жаңбыр тамшыларының түсуін шаршы торда кездейсоқ кезуін модельдеуге болады (6.9-сурет).Қозғалыс көлденең сызықтан (жер үстінен) hқашықтықта орналасқан түйіннен басталады.Төмен, жоғары, сол және оң секіру ықтималдығы тең рі, рр р^, р^. Сонымен қатар төменгі қадам ріықтималдылығы жоғарғы қадам р^ ықтималдылығы үлкенірек. Бұл жағдайда белгілі бір қадамдарды орындағаннан кейін бөлшек жер бетіне жетеді. Тамшының орташа түсу уақытын бағалау қажет.



6.9.сурет. жауын тамшысының түсу тапсырмасына арналған суреттеме

Бұл жағдайда C# тіліндегі бағдарлама фрагменті модельдеу орташа түсу уақытының T анықтамасы келесі түрге ие:

```
// n – сынақтар саны
// pDown, pUp, pLeft, pRight – төменгі қадам
ықтималдылығы, //жоғары, солға және оңға // h
– биіктік
Random random = new Random(); double T = 0;
for (int i = 0; i < n; i++)
{
   double y =
   h; double x
   = 0; int
   time = 0;
   while (y !=
   0)
   {
     double r = random.NextDouble();
     if (r < pDown) y–;
     else if (r < pDown + pUp) y++;
       else if (r < pDown + pUp + pLeft)
     x–; else x++; time++;
   }
   T += time;
}
T /= n;
```

6.6. ПЕРСИСТЕНТТІ КЕЗДЕЙСОҚ КЕЗУІ

Бұрын қарастырылған кездейсоқ кезуі модельдерінде келесі қадамның бағыты алдыңғы кезеңнің бағытына ешқандай қатысы жоқ. Персистентті кездейсоқ кезуі өту (немесе секіру) ықтималдығы соңғы өтуге байланысты.

Персистентті бір өлшемді кездейсоқ кезуі жағдайда, қадамдар тек жақын көршілес түйіндерде жасалады.k – 1 қадам жасады деп ойлайық.Содан соңк-йқадам жасайдысол бағытта а ықтималдығымен, а қарама-қарсы бағытта 1 – ақтималдылығымен қадамжасайды. Мқадамын бастапқы жағдайдан бөлшектің орташа

өшіруін анықтау қажет.

Бұл жағдайда C# тіліндегі бағдарлама фрагменті модельдеу бастапқы жағдайдан бөлшектің орташа өшіруінLкелесі түрде болады:

```
   // n — сынақтар саны
// alpha — сол бағыттағы қадам ықтималдылығы
// M — қадам саны
Random random = new Random();
double r = random.NextDouble();
int dir;//қозғалыс бағыты
//бастапқы кездейсоқ қозғалыс бағытын
таңдаймыз
if (r <0.5)dir = 1;
else dir = -1;
double x;
double L = 0.0;
for (int i = 0; i < n; i++)
{
   x = 0.0;
   // M қадам жасаймыз
   for (int j = 0; j < M; j++)
   {
     r = random.NextDouble();
     if (r>= alpha) dir = -dir^/бағытты
     өзгертемізх += dir^/керекті бағытқы
     қадам жасаймыз
   }
   L += Math.Abs(x);
}
L /= n;
```

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. Монте-Карло әдісінің мәні қандай?
2. Монте-Карло әдісімен алынған нәтижелердің дәлдігін неге байланысты?
3. Бір өлшемді кездейсоқ кезуі модельдеуі қалай іске асырылады?
4. Екі өлшемді кездейсоқ кезуі модельдеуі қалай іске

асырылады?

5. Сіңірмелі экранда қарапайым кездейоқ кезуі модельдеуі қалай іске асырылады?
6. Персистентті кездейсоқ кезуі модельдеуі қалай іске асырылады?

7 Тарау

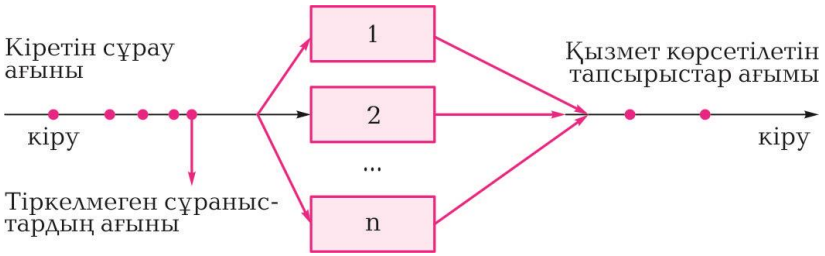
ЖАППАЙ ҚЫЗМЕТ КӨРСЕТУДІҢ ЖҮЙЕЛЕРІН МОДЕЛЬДЕУ

7.1. ЖАППАЙ ҚЫЗМЕТ КӨРСЕТУДІҢ ЖҮЙЕЛЕРІНІҢ НЕГІЗГІ СИПАТТАМАЛАРЫ

Модельдеу кезінде әдетте зерттеуші жаппай қызмет көрсетудің жүйелерінің сипаттамаларын анықтайтын міндеттермен соқтығысады. Жаппай қызмет көрсету жүйелері (ЖҚЖ)— бұл кездейсоқ уақыт мезетінде қызмет көрсетуге тапсырыстар немесе талаптар түсетін жүйелер (Сурет 7.1). Бұл кезде түскен тапсырыстар қызмет көрсету каналдар жүйелерінің меншігі көмегімен жүзеге асады.

ЖҚКЖ мысалдары ретінде банктар мен дүкендердегі кассалар, есептеу мәселелерін шешетін дербес компьютерлер, қызмет көрсету орталықтары, автокөліктерге жанармай құю бекеттері (АЖҚБ), телефон станциялары, зауыттардағы жинақтаушы конвейерлер және т.б. болады.

Жұмыс жасау процесінде ЖҚЖ-не қызмет көрсетуге тапсырыстар түседі. ЖҚЖ-не түскен соң талаптар бұрын түскен талаптар кезегіне келіп қосылады. Қызмет көрсету каналы кейбір ережелерге сәйкес кезектегі талапты таңдайды да оны орындауға көшеді. Кезекті талапқа қызмет көрсету процедурасы аяқталғаннан соң қызмет көрсету каналы егер кезекте келесі талап болса, соған қызмет көрсетуге көшеді. Бұл процесс қызмет көрсету жүйесінің барлық жұмыс жасау мерзімі бойы бірнеше рет қайталанады.



Сур. 7.1. ЖҚЖ типтің графикалық көрінісі

ЖҚЖ-нің кез келген түрінің негізгі компоненттері [1]:

- қызмет көрсетуге түскен талаптар мен тапсырыстардың кіріс ағымы;
- кезек тәртібі;
- қызмет көрсету механизмі жатады.

Тапсырыстың кіріс ағымы кездейсоқ өлшемнің таралу заңымен анықталады, әдетте талаптар түсуі және әрбір талаптағы тапсырыстар санының арасындағы уақыттың берілетін интервалымен анықталады. Аі арқылы талаптар түсуі (і- 1) және талаптар і арасындағы уақытты белгілейміз. Онда 1 = 1/Е(А)өлшемі талаптар түсуінің қарқындылығы деп аталады. Мұндағы Е(А)— талаптар түсуінің арасындағы орташа уақыт.

Кезек тәртібі қызмет көрсету құралы кезектегі қызмет көрсетуге арналған талапты таңдау кезінде қолданатын ережемен анықталады. Әдетте келесі ережелермен анықталатын әдеттегі кезектің тәртіптері қолданылады:

- бірінші келдің— бірінші қызмет көрсетеді (FirstInFirstOut — FIFO);
- соңғыкелдің — бірінші қызмет көрсетеді(LastInFirstOut — LIFO);
- тапсырыстардың кездейсоқ іріктелуі;
- басымдылық критерийі бойынша тапсырыстарды іріктеу.

Қызмет көрсету механизмі қызмет көрсету жүйесінің құрылымы және қызмет көрсету процедурасының сипаттамаларымен анықталады. Қызмет көрсету жүйесінің құрылымы ЖҚКЖ қанша қызмет көрсету каналы бар, бұл каналдар бірдей ме не бірдей емес пе, бұл қаналдар өзара қалай байланысады осыларға негізделеді. Қызмет көрсету процедурасы әдетте тапсырыстарға қызмет көрсету уақытымен анықталатын ықтималдылық заңдарымен сипатталады. Si арқылы і талабының қызмет көрсету уақытын белгілейміз. Онда |т = 1/ Е(S) өлшемі талаптарға қызмет көрсету қарқындылығы деп аталады. МұндағыЕ(S) — талаптарға қызмет көрсетудің орташа уақыты.

ЖҚЖжұмыс жасауы қарқындылығының сипаттамасы ретінде үш негізгі көрсеткіштер (әдетте орташа) тобын таңдауға болады [1].

1. ЖҚЖ қолданудың тиімділік көрсеткіштері:

1.1. ЖҚЖ абсолютті өткізгіштік қабілеттілігі— уақыт бірлігінде ЖҚЖатқара алатын тапсырыстардың орташа саны.

1.2. ЖҚЖ –нің салыстырмалы өткізгіштік қабілеттілігі — қызмет көрсетілетін тапсырыстың осы уақыт аралығында түскен орташа санына уақыт бірлігінде орташа санының қатынасы.

1.3. ЖҚЖ –нің қолдану коэффициенті — ЖҚЖ тапсырысқа қызмет көрсетіп жатқан уақыттың орташа үлесі, және т.б.

2. Тапсырыстарға қызмет көрсету сапасының көрсеткіштері:

2.1. Кезектегі тапсырысты күтудің орташа уақыты.

2.2. ЖҚЖ-де тапсырыстың болуының орташа уақыты.

2.3. Күттірмей қызмет көрсету кезінде тапсырысты қабылдамау ықтималдылығы.

2.4. Түскен тапсырысқа жедел қызмет көрсетілу ықтималдылығы.

2.5. Кезектегі тапсырыстардың орташа саны.

2.6. ЖҚЖ-де жатқан тапсырыстардың орташа саны, және т.б.

3. «ЖҚЖ-тұтынушы» жұбының жұмыс жасау тиімділігінің көрсеткіштері, мұндағы «тұтынушы» барлық иапсырыстар жиынтығы не олардың шығу көзі (мысалы, уақыт бірлігіндегі ЖҚКЖ түсіретін орташа кіріс).

ЖҚКЖ жіктелуінің негізгі жолдарын қарастырайық [1].

1. Тапсырыстар (талаптар) ағымының сипаттамасы және қызмет көрсету ұзақтығы бойынша марков және марков емес ЖҚКЖ деп ажыратады. Марков жүйесінде талаптардың кіріс ағымы мен қызмет көрсетілетін шығыс талаптар (тапсырыстар) ағымы пуассонов болады. Мұндай ағымдар үшін k талаптар саны кез келген уақыт интервалына заңмен таралады

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \; k \geq 0, \; t \geq 0,$$

Мұндағы λ— талаптар ағымының қарқындылығы (уақыт бірлігіндегі талаптар саны).

Пуассонов ағымдары қарапайым аналитикалық шешімдерді жіберетін ЖҚЖ математикалық үлгілерін оңай суреттеуге және құруға мүмкіндік береді. Марков емес процестерде ЖҚЖ сипаттамаларын алу едәуір қиындадйды және әдетте имитациялық модельдеуді қолдануды талап етеді.

Егер тапсырыстың пуассон ағымы (кіріс не шығыс) уақытта өзгермейтін стационарлы болса (мұндай ағымдар қарапайымдар деп те аталады), онда екі кез келген көршілес оқиғалардың арасындағы уақыт аралығы болатын кездейсоқ өлшем (тапсырыс келген аралықта

немесе тапсырысқа қызмет көрсету басында және аяғында) ықтималдылық тығыздығымен көрсеткіштік заңдылығымен таралады

$$f(t) = \lambda e^{-\lambda t},\ t \geq 0,$$

мұндағы λ — ағым қарқындылығы.

2. Қызмет көрсету тәртібіне байланысты ЖҚЖ үш негізгі түрін бөліп қарастырады:

- қабылданбағандары бар жүйелер, барлық каналдар бос болмаған кезде жүйеге түскен тапсырыс қабылданбайды және кезекті босатады;
- кезекті күту жүйелері (кезек), мұнда барлық каналдар бос болмаған кезде жүйеге түскен тапсырыс кезекке тұрады да каналдардың бірі босамайынша күтеді. Мұндай жүйелерде кезектегі тапсырыс өзінің кезегі келмейінше шектеусіз қызмет көрсетуді күтеді;
- аралас түрдегі жүйелер (күтуі шектеулі), кезекке түскен тапсырыстарға кейбір шектеулер жүктеледі. Мысалы, кезек ұзындығы немесе кезекте тұру уақыты шектеледі.

3. Қызмет көрсету каналдарының санына байланысты бір каналды және көп каналды (л-каналды) жүйелерді бөліп қарастырады. Әдетте әрбір канал біруақытта бір ғана тапсырысқа қызмет көрсете алады және әрбір тапсырыс бір каналмен қызмет көрсетіледі деп есептелінеді. Көп каналды ЖҚЖ бірыңғай каналдардан тұрады, не бір тапсырысқа қызмет көрсету ұзақтығымен ерекшеленетін әртүрлі каналдардан тұрады.

4. Қызмет көрсету этаптарының санына қарай барлық ЖҚЖ бір фазалық және көп фазалық жүйелер деп бөледі. Егер ЖҚЖ каналдары бірыңғай болса, яғни қызмет көрсетудің бір операциясын орындайды, оларды бір фазалық ЖҚЖ деп атайды. Егер қызмет көрсету каналдары тізбекті орналасқан және бірыңғай болмаса, қызмет көрсетудің әр түрлі операцияларын орындайды, ондай ЖҚЖ көп фазалы деп атайды. Көп фазалы ЖҚЖ мысалдары ретінде емхана болады, мұнда алдымен тіркелу бөлімінде дәрігерге талон алу керек және содан кейін ғана қарала алады.

5. ЖҚЖ талаптарының шығу көздерінің орналасуы бойынша ашық, яғни шығу көзі жүйеден тыс болғанда және тұйық, мұнда шығу көзі жүйенің өзінде болады. Соңғысына, мысалы, автобус саябағы жатады, мұнда автобустар бұзылыстар көздері болады, сәйкесінше, қызмет көрсетудің талаптарының көздері болады (жөндеу).

Тұйық ЖҚЖ ашыққа қарағанда, талаптардың ағым қарқындылығы

осы кезде кезекте не қызмет көрсетуде тұрған талаптар санына байланысты болады.

7.2. ҚЫЗМЕТ КӨРСЕТУДІҢ БІР ҚҰРАЛЫ БАР ЖҮЙЕЛЕР

ЖҚЖ қарапайымдар сипаттамасының аналитикалық бағалауының сұрақтарын қарастырамыз, егжей-тегжейлі [1] берілген.

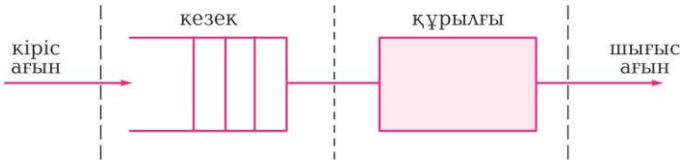
Басында сурет 7.2. көрсетілген бір каналды ЖҚЖ (қызмет көрсетудің бір ғана құралы бар) қарастырамыз. Мұндағы ЖҚЖ де тапсырыстар мен талаптар ағымдары λжәнеμ қарқындылығына сәйкес қарапайымдар болады (яғни ЖҚЖ тапсырыстары орташа есеппен уақыт бірлігінде 1/λарқылы жәнеорташа есеппен 1/μ бойы уақыт бірлігінде қызмет көрсетіледі). Қарастырылып жатқан ЖҚЖ кезегі шектеулі жүйе болса, мұнда шектеулер кезектің ұзындығына жинақталады (кезек ұзындығы ≤m). Мұндайжүйеm=0 кезінде кезексіз ЖҚЖ айналады, ал m=оо болғанда шектеулі кезегі бар ЖҚЖ айналады.

Суреттелген ЖҚЖ тиімділігінің ең маңызды көрсеткіштерін аналитикалық бағалау үшін формулалар келтіреміз. Бірден мән береміз, тапсырыстар мен талаптар ағымы қарапайым болмаса, онда алда көрсетілетін өрнектердің көбі әділ болмайды. Тапсырыстар мен талаптардың басқа ағымдарының аздау санына ЖҚЖ сипаттамасын бағалау үшін өрнектер болады. Күрделі жағдайларда осы мақсаттар үшін компьютерлік модельдеуді қолдану қажет.

Талқыланатын ЖҚЖ қолдану коэффициенті келесідей анықталады:

$$\rho = \frac{\lambda}{\mu}.$$

ЖҚЖ-дегі к тапсырмалардың болу ықтималдылығын өрнектеу үшін Рк келесідей түрге ие болады



Сурет 7.2. Бір каналды ЖҚЖ

$$P_0 = \begin{cases} \dfrac{1-\rho}{1-\rho^{m+2}}, & \rho \neq 1; \\ \dfrac{1}{m+2}, & \rho = 1. \end{cases}$$

$$P_k = \rho^k P_0;\ k = 1, \ldots, m+1.$$

Қызмет көрсетудегі бас тартулар ықтималдығы, яғни қызмет көрсетуде өтініш беру кезінде жүйеде m+1 өтініш болуының ықтималдығы келесі формуламен анықталады:

$$P_{отк} = P_{m+1} = \begin{cases} \dfrac{\rho^{m+1}(1-\rho)}{1-\rho^{m+2}}, & \rho \neq 1; \\ \dfrac{1}{m+2}, & \rho = 1. \end{cases}$$

(7.1) формуланы пайдалана отырып, өтініштің жүйеге қабылдану ықтималдығын анықтауға болады:

$$P_{сис} = 1 - P_{отк}$$

СМО қатысты өткізу қабілеті:

$$Q = P_{сис} = 1 - P_{отк} = \begin{cases} \dfrac{1-\rho^{m+1}}{1-\rho^{m+2}}, & \text{егер } \rho \neq 1; \\ \dfrac{m+1}{m+2}, & \text{егер } \rho = 1. \end{cases}$$

СМО абсолютті өткізу қабілеті:

$$A = \lambda Q$$

Кезектегі өтініштердің орташа саны:

$$\bar{N}_{оч} = \begin{cases} P_0 \dfrac{\rho^2\left[1-\rho^m(m+1-m\rho)\right]}{(1-\rho^{m+2})(1-\rho)}, & \text{егер } \rho \neq 1; \\ \dfrac{m(m+1)}{2(m+2)}, & \text{егер } \rho = 1. \end{cases}$$

Қызмет көрсету үрдісіндегі өтініштердің орташа саны:

$$\bar{N}_{об} = \rho Q = \begin{cases} \dfrac{\rho(1-\rho^{m+1})}{1-\rho^{m+2}}, & \text{егер } \rho \neq 1; \\ \dfrac{m+1}{m+2}, & \text{егер } \rho = 1. \end{cases}$$

СМО өтініштердің орташа саны:

$$\bar{N}_{\text{сис}} = \bar{N}_{\text{оч}} + \bar{N}_{\text{об}}.$$

Кезектегі өтініштердерді күтудің орташа уақыты:

$$\bar{T}_{\text{оч}} = \frac{\bar{N}_{\text{оч}}}{\lambda P_{\text{сис}}}.$$

Жүйедегі өтініштердің болуының орташа саны:

$$\bar{T}_{\text{сис}} = \frac{\bar{N}_{\text{сис}}}{\lambda P_{\text{сис}}}.$$

Берілген формулалар қызмет көрсетуге талаптардың қарапайым ағымдарымен бір каналды ЖҚКЖ сипаттамаларын бағалау үшін қолданылатынын ескертеміз. Егер тапсырыстар ағымы қарапайым болмаса, онда ЖҚКЖ сипаттамаларын алу үшін жиі компьютерлік модельдеу қолданылу қажет.

Шектеулі күтілуі бар бір каналды ЖҚКЖ сипаттамаларын есептеу мысалдарын қарастырамыз.

Мысал. Нотариалды кеңсе бір каналды ЖҚКЖ болып табылады. Нотариуске кезекті күту бөлмесінде орындар саны шектеулі және екіге тең. Егер бөлмедегі барлық орындар бос болмаса, онда жаңадан келген клиент кезекке тұрмайды. Кеңес алуға келген клиенттер сағатына $\lambda = 8$ клиент қарқындылығымен қарапайым болады. Қызмет көрсету уақыты қызмет көрсетудің орташа уақытымен $t = 7$ мин. экпоненциалды заңдылықпен таралады.

Стационарлық режимде жұмыс жасайтын нотариалды кеңсенің ықтимал сипаттамаларын анықтау.

Шешімі. 1. Алдымен клиенттерге қызмет көрсету ағымының қарқындылығын анықтаймыз. Нәтижені λ сияқты өлшемде аламыз:

$$\mu = \frac{1}{\bar{t}} = \frac{1}{7/60} = 8{,}571.$$

2.СМО пайдалану коэффициентін λжәне μ қарқындылықтарының қатынасы ретінде есептейік:

$$\rho = \frac{\lambda}{\mu} = \frac{8}{8{,}571} = 0{,}933.$$

3.Клиенттерде СМО болу ықтималдығын есептейік:

$$P_0 = \frac{1-\rho}{1-\rho^{\alpha+2}} = \frac{1-0{,}933}{1-0{,}933^4} \approx 0{,}276;$$

$$P_1 = \rho^1 P_0 = 0{,}933 \cdot 0{,}276 \approx 0{,}258;$$

$$P_2 = \rho^2 P_0 = 0{,}933^2 \cdot 0{,}276 \approx 0{,}241;$$

$$P_3 = \rho^3 P_0 = 0{,}933^3 \cdot 0{,}276 \approx 0{,}225.$$

4.Клиентке қызмет көрсетудегі бас тарту ықтималдығы:

$P_{отк} = P_{m+1} = 0{,}225.$

5.Нотариалды кеңсенің қатысты өткізу қабілеті:

$Q = P_{снс} = 1 - P_{отк} = 1 - 0{,}225 = 0{,}775.$

6.Нотариалды кеңсенің абсолютті өткізу қабілеті:

$A = \lambda \cdot Q = 8 \cdot 0{,}775 = 6{,}2.$

7.Кезектегі клиенттердің орташа саны:

$$\bar{N}_{оч} = P_0 \frac{\rho^2\left[1-\rho^m(m+1-m\rho)\right]}{(1-\rho^{m+2})(1-\rho)} =$$

$$= 0{,}276\frac{0{,}933^2\left[1-0{,}933^2(2+1-2\cdot 0{,}933)\right]}{(1-0{,}933^4)(1-0{,}933)} \approx 0{,}69.$$

8.Қызмет көрсетілетін клиенттердің орташа саны:

$\bar{N}_{об} = \rho Q = 0{,}933 \cdot 0{,}775 \approx 0{,}72.$

9.Жүйедегі клиенттердің орташа саны:

$\bar{N}_{снс} = \bar{N}_{оч} + \bar{N}_{об} = 0{,}69 + 0{,}72 = 1{,}41.$

10.Кезектегі клиенттердің қызмет көрсетілудегі орташа уақыты, минут:

$$\bar{T}_{оч} = \frac{\bar{N}_{оч}}{\lambda P_{снс}} = \frac{0{,}69}{8 \cdot 0{,}775} = 0{,}1113\ ч = 6{,}68.$$

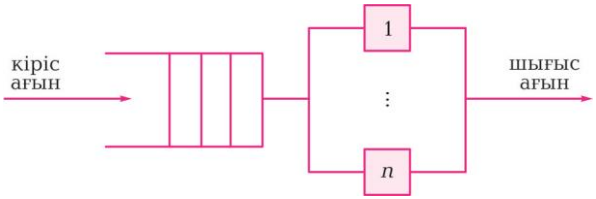
11.Жүйедегі клиенттердің қызмет көрсетілудегі орташа уақыты, минут:

$$\bar{T}_{снс} = \frac{\bar{N}_{снс}}{\lambda P_{снс}} = \frac{1{,}41}{8 \cdot 0{,}775} = 0{,}228\ ч = 13{,}68.$$

Қарастырылған нотариалды кеңсенің жұмысын қанағаттандырылған деп санауға болады, өйткені ол орташа есеппен 22,5% жағдайында клиенттерге қызмет көрсетпейді (Рқабылдамау = 0,225). Аналитикалық есеппен кезекке қосымша орын енгізу арқылы қызмет көрсетуде қабылдамауларды 17 % дейін төмендетуге болады.

7.2. ҚЫЗМЕТ КӨРСЕТУДІҢ БІР ҚҰРАЛЫ БАР ЖҮЙЕЛЕР

Көп каналды ЖҚЖ с n (n> 2) қызмет көрсетудің бірдей құралымен 7.3. суретте көрсетілген. Қарастырылған бір каналды жүйелер үшін бұл ЖҚЖ шектеулі күтілуі бар жүйе болғандай, (кезек ұзындығы <m), тапсырыстар мен талаптардың кіріс ағымы λ эәне μ қарқындылығымен қарапайым болады.



Сурет 7.3. Көп каналды ЖҚЖ

Бұл берілген ЖҚЖ үшін жүктелу коэффициенті келесідей анықталады:

$$\rho = \frac{\lambda}{\mu}.$$

Бір каналға келетін жүктелу коэффициенті келесідей есептеледі:

$$\psi = \frac{\rho}{n}.$$

Барлық каналдар бос болу ықтималдылығы:

$$P_0 = \begin{cases} \left[\sum_{k=0}^{n} \frac{n^k}{k!} \psi^k + \frac{n^n}{n!} \frac{\psi^{n+1}\left(1-\psi^m\right)}{1-\psi} \right]^{-1}, \psi \neq 1; \\ \left[\sum_{k=0}^{n} \frac{n^k}{k!} + \frac{n^n}{n!} m \right]^{-1}, \psi = 1. \end{cases}$$

ЖҚЖ ктапсырысының болу ықтималдылығы келесідей есептеледі:

$$P_k = \begin{cases} \frac{n^k}{k!} \psi^k P_0, k = 1, \ldots, n; \\ \frac{n^n}{n!} \psi^k P_0, k = n+1, \ldots, n+m. \end{cases}$$

Тапсырысқа қызмет көрсетуді қабылдамау ықтималдылығы жүйеде n+ m тапсырысының бар болу ықтималдылығына тең болады:

$$P_{отк} = P_{n+m} = \frac{n^n}{n!} \psi^{n+m} P_0.$$

(7.2) формуласын қолданып тапсырыс жүйеге қабылдану ықтималдылығын анықтауға болады:

$$P_{\text{сис}} = 1 - P_{\text{отк}} = 1 - \frac{n^n}{n!}\psi^{n+m}P_0.$$

ЖҚЖ салыстырмалы өткізгіштік қабілеттілігі

$$Q = P_{\text{сис}} = 1 - P_{\text{отк}} = 1 - \frac{n^n}{n!}\psi^{n+m}P_0.$$

ЖҚЖ абсолюттік өткізгіштік қабілеттілігі

$$A = \lambda Q.$$

Бос емес каналдардың орташа саны (яғни қызмет көрсетілу үстіндегі тапсырыстардың орташа саны)

$$\bar{N}_{\text{об}} = \frac{A}{\mu} = \rho Q = \rho\left(1 - \frac{n^n}{n!}\psi^{n+m}P_0\right).$$

Кезектегі тапсырыстардың орташа саны

$$\bar{N}_{\text{оч}} = \begin{cases} \dfrac{n^n}{n!}\psi^{n+1}\dfrac{1-(m+1)\psi^m + m\psi^{m+1}}{(1-\psi)^2}P_0, \psi \neq 1; \\ \dfrac{n^n}{n!}\dfrac{m(m+1)}{2}P_0, \psi = 1. \end{cases}$$

ЖҚКЖ тапсырыстардың орташа саны

$$\bar{N}_{\text{сис}} = \bar{N}_{\text{оч}} + \bar{N}_{\text{об}}.$$

Кезектегі тапсырыстардың күтілуінің орташа уақыты

$$\bar{T}_{\text{оч}} = \frac{\bar{N}_{\text{оч}}}{\lambda P_{\text{отс}}}.$$

Жүйедегі тапсырыстың болуының орташа уақыты

$$\bar{T}_{\text{отс}} = \frac{\bar{N}_{\text{сис}}}{\lambda P_{\text{сис}}}.$$

Келтірілген формулалар ЖҚЖ үшін тапсырыстар мен талаптар қарапайым ағымдарымен есептеу жүргізуге және имитациялық модельдеудің алынған нәтижелерімен оларды салыстыруға мүмкіндік береді.

Бір каналды ЖҚЖ келтірілген мысалды аздап өзгертіп, көп каналды ЖҚЖ сипаттамаларын есептеу мысалдарын қарастырайық.

Мысал. Нотариалды кеңсе екі каналды ЖҚЖ ұсынады. Нотариуска кезекке тұрып күтубөлмесінде орындар саны шектеулі

және үшке тең. Егер бөлмедегі барлық орындар бос болмаса, онда келген клиент кезекке тұрмайды. Кеңес алуға келген клиенттер ағымы сағатына қарқындылығы λ= 12 клиент болатын қарапайым. Қызмет көрсету уақыты қызмет көрсетудің орташа уақытымен t = 7 мин. экпоненциалды заңдылықпен таралады.

Стационарлық режимде жұмыс жасайтын нотариалды кеңсенің ықтимал сипаттамаларын анықтау.

Шешімі. 1. Сағатына клиенттерге қызмет көрсету ағымының қарқындылығы.

$$\mu = \frac{1}{t} = \frac{1}{7} \cdot 60 = 8{,}571.$$

2. ЖҚЖ жүктелу коэффициенті:

$$\rho = \frac{\lambda}{\mu} = \frac{12}{8{,}571} = 1{,}4.$$

3. ЖҚЖ бір каналға жүктеу коэффициенті:

$$\psi = \frac{\rho}{n} = \frac{1{,}4}{2} = 0{,}7.$$

4. ЖҚЖ кклиенттің болу ықтималдылығы:

$$P_0 = \left[\sum_{k=0}^{n} \frac{n^k}{k!}\psi^k + \frac{n^n}{n!}\frac{\psi^{n+1}(1-\psi^n)}{1-\psi}\right]^{-1} =$$

$$= \left[\sum_{k=0}^{2} \frac{2^k}{k!}0{,}7^k + \frac{2^2}{2!}\frac{0{,}7^{2+1}(1-0{,}7^3)}{1-0{,}7}\right]^{-1} \approx 0{,}205;$$

$$P_1 = \frac{n^1}{1!}\psi^1 P_0 = \frac{2^1}{1!}0{,}7^1 \cdot 0{,}205 \approx 0{,}287;$$

$$P_2 = \frac{n^2}{2!}\psi^2 P_0 = \frac{2^2}{2!}0{,}7^2 \cdot 0{,}205 \approx 0{,}201;$$

$$P_3 = \frac{n^3}{n!}\psi^3 P_0 = \frac{2^3}{2!}0{,}7^3 \cdot 0{,}205 \approx 0{,}140;$$

$$P_4 = \frac{n^4}{n!}\psi^4 P_0 = \frac{2^4}{2!}0{,}7^4 \cdot 0{,}205 \approx 0{,}098;$$

$$P_5 = \frac{n^5}{n!}\psi^5 P_0 = \frac{2^5}{2!}0{,}7^5 \cdot 0{,}205 \approx 0{,}069.$$

5. Клиентке қызмет көрсетуден бас тарту ықтималдылығы:

$$P_{отк} = P_{n+n} = 0{,}069.$$

6. Нотариалды конторaның салыстырмалы өткізу қабілеті:

$$Q = P_{\text{снс}} = 1 - P_{\text{отк}} = 1 - 0{,}069 = 0{,}931.$$

7. Нотариалды конторaның сағатына клиенттерді кіргізуінің абсолюттік қабілеті:

$$A = \lambda Q = 12 \cdot 0{,}931 = 11{,}17.$$

8. Кезектегі клиенттердің орташа саны:

$$\bar{N}_{\text{оч}} = \frac{n^n}{n!}\psi^{n+1}\frac{1-(n+1)\psi^n + n\psi^{n+1}}{(1-\psi)^2}P_0 =$$

$$= \frac{2^2}{2!}0{,}7^{2+1}\frac{1-(3+1)0{,}7^3 + 3\cdot 0{,}7^{3+1}}{(1-0{,}7)^2}0{,}205 \approx 0{,}544.$$

9. Қызмет көрсетуде тұрған клиенттердің орташа саны:

$$\bar{N}_{\text{об}} = \rho Q = 1{,}4 \cdot 0{,}931 = 1{,}303.$$

10. Жүйедегі клиенттердің орташа саны:

$$\bar{N}_{\text{снс}} = \bar{N}_{\text{оч}} + \bar{N}_{\text{об}} = 0{,}544 + 1{,}303 = 1{,}847.$$

11. Клиенттердің кезекте отырған орташа уақыты, мин:

$$\bar{T}_{\text{оч}} = \frac{\bar{N}_{\text{оч}}}{\lambda P_{\text{снс}}} = \frac{0{,}544}{12 \cdot 0{,}931} = 0{,}048\ \text{ч} = 2{,}92.$$

12. Жүйедегі клиенттерге қызмет көрсетудегі орташа уақыт, минут:

$$\bar{T}_{\text{снс}} = \frac{\bar{N}_{\text{снс}}}{\lambda P_{\text{снс}}} = \frac{1{,}847}{12 \cdot 0{,}931} = 0{,}165\ \text{ч} = 9{,}92.$$

Қарастырылған нотариалды конторaның жұмысын қанағаттандырарлық деп санауға болады, өйткені ол орташа есеппен барлығы 6,9 % (Рқаб-мау = 0,069) жағдайда клиентке қызмет көрсетпейді.

7.4. ЖАБЫҚ ЖАППАЙ ҚЫЗМЕТ КӨРСЕТУ ЖҮЙЕЛЕРІ

Қарастырылған ЖҚЖ кіретін өтініштің λ ағысының қарқындылығы жүйе күйіне тәуелді болмады. Бұл жағдайда өтініш көзі ЖҚЖ қатысты сыртқы болды және талаптың шектелмеген ағысын түрлендірді. Бірқатар жағдайда өтініш ағысының λ жүйе күйіне тәуелді, әрі талап көзі ішкі болып табылады және шектелген өтініш ағысын түрлендіреді. Осыған ұқсас жүйелерді жаппай қызмет көрсететін жабық жүйе деп атайды.

Жабық ЖҚЖ мысалы, бұзылған кезде автомеханик (N

>R)бригадаларымен R жөнделетін Nавтобустардан тұратын автобус паркі болып табылады. Сонымен қатар әр автобус тек бір бригадамен жөнделеді. Осы жүйеде автобустар қызмет көрсетуге өтініш көзі, ал автомеханиктер бригадасы – қызмет көрсететін каналдар болып табылады. Бұзылған автобусқа қызмет көрсеткеннен кейін бағытта пайдаланылады және қызмет көрсетуге талаптың пайда болуының ықтимал көзіне айналады.

λ өтініш ағысының қарқындылығы қазіргі кезде қанша автобус жұмыс істеп тұрғаны(N - K)мен қанша автобус жөнделуде немесе қызмет көрсетуге кезекте тұрғанына (K) байланысты болатындығы анық.

Қарастырылған модельде талап көзінің сыйымдылығы шектелген деп санаған жөн. Кіретін талап ағысы кездейсоқ уақыт мезетінде істен шығатын және қызмет көрсетуді талап ететін пайдаланылатын машинаның шектелген(N - K) санынан шығады.

Сонымен қатар пайддаланудағы әр машина(N - K)λ қарқындылығы бар пуассонов ағыстарының талаптарын басқа нысандарға тәуелсіз түрлендіреді. Сонда жалпы жиынтықты кіруші ағыс λ(N - K) қарқындылығына ие. Ең болмаса бір канал бос болған кезде жүйеге түскен талаптар бірден қызмет көрсетуге кетеді. Егер талаптар барлық каналдардың басқа талаптарға қызмет көрсетумен бос болмаса, онда ол жүйеден кетпейді, кезекке тұрып, каналдардың бірі бос болғанға дейін күтеді.

Жабық ЖҚЖ тиімділігінің ең маңызды көрсеткіштерін аналитикалық бағалауға арналған формуланы келтірейік. Берілген баға [1] көрсетілген.

Жүйедегі кнысандар қызмет көрсетуде немесе кезекте болу ықтималдылығы төмендегідей анықталады

$$P_k = P_0 \varphi_k,$$

мұндағыP0–нысанның бірде бірі қызмет көрсетуде болмау ықтималдылығы;

$$\varphi_k = \begin{cases} \dfrac{N!\rho^k}{k!(N-k)!}, & \text{для } 1 \le k < R; \\ \dfrac{N!\rho^k}{R!R^{k-R}(N-k)!}, & \text{для } R \le k \le N, \end{cases}$$

$$\rho = \frac{\lambda}{\mu},$$

Мұнда λ–бір қызмет көрсетілетін нысанға шаққандағы ағыс талаптарының қарқындылығы; μ–бір қызмет көрсету каналымен нысанға қызмет көрсету қарқындылығы.

Р0ықтималдылығы теңдікте болады

$$\sum_{k=0}^{N} P_k = 1.$$

Аламыз

$$\sum_{k=0}^{N} P_k = P_0 + P_0\varphi_1 + P_0\varphi_2 + \ldots + P_0\varphi_N = P_0(1 + \varphi_1 + \varphi_2 + \ldots + \varphi_N) = 1.$$

одан

$$P_0 = \frac{1}{1 + \varphi_1 + \varphi_2 + \ldots + \varphi_N}.$$

Қызмет көрсетуге кезекте тұрған нысандардың орташа саны

$$\bar{N}_{оч} = \sum_{k=R}^{N} (k - R)P_k.$$

Жүйеде (қызмет көрсетуге және кезекте) тұрған нысандардың орташа саны,

$$\bar{R}_{п} = \sum_{k=0}^{R-1} (R - k)P_k.$$

Іркілген қызмет көрсету каналдарының орташа саны

$$\alpha_1 = \frac{\bar{N}_{оч}}{N}.$$

Іркілген қызмет көрсету нысандарының коэффициенті

$$\alpha_2 = 1 - \frac{\bar{N}_{сис}}{N}.$$

Нысанды пайдалану коэффициенті

$$\alpha_3 = \frac{\bar{R}_{\text{п}}}{R}.$$

Қызмет көрсететін каналдардың іркілу коэффициенті

$$\bar{T}_{\text{оч}} = \frac{1}{\lambda}\left(\frac{1-\alpha_2}{\alpha_2}\right) - \frac{1}{\mu}.$$

Нысанның кезектег Жабық ЖҚЖ сипаттамаларын есептеу мысалын қарастырайық.

Мысал. 15 компьютерден тұратын есептегіш орталықты қарастырайық, олар бұзылған кезде ЖК жөндеу бойынша екі маман жөндейді. Сонымен қатар әр компьютерді бір маман ғана жөндейді. Істен шыққан компьютер жөнделген соң есептегіш орталықта пайдаланылады және қызмет көрсетуге ықтимал талаптың пайда болу көзіне айналады.Бір компьютерден қабылдамау ағысының қарқындылығы 100 сағ аралығынла 0,5 ретке тең, компьютерді жөндеудің орташа уақыты 10 сағ құрайды.

Есептегіш орталықтың сипаттамаларын анықтау.

Шешімі.Біздің тапсырма үшінN= 15, R= 2, λ = 0,5, μ = 10.

1. р коэффициентін λ және μ қарқындылығына қатысты анықтаймыз:і орташа күту уақыты

$$\rho = \frac{\lambda}{\mu} = \frac{0,5}{20} = 0,05.$$

2. Жүйедегі ккомпьютерлерге қызмет көрсетуде немесе кезекте болу ықтималдылығын есептейміз.

Ол үшін біріншік = 1 ...N арналған φктабайық

$$\varphi_1 = \frac{10!0,05^1}{(10-1)!} = 0,5; \quad \varphi_2 = \frac{10!0,05^2}{2!2^{2-2}(10-2)!} = 0,1125;$$

$$\varphi_3 = \frac{10!0,05^3}{2!2^{3-2}(10-3)!} = 0,0225; \quad \varphi_4 = \frac{10!0,05^4}{2!2^{4-2}(10-4)!} = 3,9\cdot 10^{-3};$$

$$\varphi_5 = \frac{10!0,05^5}{2!2^{5-2}(10-5)!} = 5,9\cdot 10^{-4}; \quad \varphi_6 = \frac{10!0,05^6}{2!2^{6-2}(10-6)!} = 7,4\cdot 10^{-5};$$

$$\varphi_7 = \frac{10!0,05^7}{2!2^{7-2}(10-7)!} = 7,6\cdot 10^{-6}; \quad \varphi_8 = \frac{10!0,05^8}{2!2^{8-2}(10-8)!} = 5,5\cdot 10^{-7};$$

$$\varphi_9 = \frac{10!0,05^9}{2!2^{9-2}(10-9)!} = 2,8\cdot 10^{-8}; \quad \varphi_{10} = \frac{10!0,05^{10}}{2!2^{10-2}(10-10)!} = 6,9\cdot 10^{-10}.$$

$$P_0 = \frac{1}{1+\varphi_1+\varphi_2+\ldots+\varphi_N} = \frac{1}{1,6396} = 0,6099.$$

Есептейміз P_k:

$P_1 = P_0\varphi_1 = 0{,}3050;\ P_2 = P_0\varphi_2 = 0{,}0686;$

$P_3 = P_0\varphi_3 = 0{,}0137;\ P_4 = P_0\varphi_4 = 2{,}4 \cdot 10^{-3};$

$P_5 = P_0\varphi_5 = 3{,}6 \cdot 10^{-4};\ P_6 = P_0\varphi_6 = 4{,}5 \cdot 10^{-5};$

$P_7 = P_0\varphi_7 = 4{,}5 \cdot 10^{-6};\ P_8 = P_0\varphi_8 = 3{,}4 \cdot 10^{-7};$

$P_9 = P_0\varphi_9 = 1{,}7 \cdot 10^{-8};\ P_{10} = P_0\varphi_{10} = 4{,}2 \cdot 10^{-10}.$

3.Қызмет көрсетуге кезектегі компьютерлердің орташа саны

$$\bar{N}_{оч} = \sum_{k=2}^{N}(k - R)P_k = 0{,}0198.$$

4. Қызмет көрсету немесе кезектегі компьютерлердің орташа саны

$$\bar{N}_{сис} = \sum_{k=0}^{N} kP_k = 0{,}4951.$$

5. Жөндау бойынша іркіліп тұрған мамандардың орташа саны

$$\bar{R}_п = \sum_{k=0}^{R-1}(R - k)P_k = 2P_0 + P_1 = 1{,}5248.$$

6. Компьютердің іркілу коэффициенті N_{04}

$$\alpha_1 = \frac{\bar{N}_{оч}}{N} = \frac{0{,}0198}{10} = 1{,}98 \cdot 10^{-2}.$$

7. Компьютерді қолдану коэффициенті

$$\alpha_2 = 1 - \frac{\bar{N}_{сис}}{N} = 1 - \frac{0{,}4951}{10} = 0{,}9505.$$

8. Жөндеу бойынша маманның іркілу коэффициенті

$$\alpha_3 = \frac{\bar{R}_п}{R} = \frac{1{,}5248}{2} = 0{,}7624.$$

9. Компьютердің жөндеуге кезекте тұрған орташа уақыты

$$\bar{T}_{оч} = \frac{1}{\lambda}\left(\frac{1 - \alpha_2}{\alpha_2}\right) - \frac{1}{\mu} = \frac{1}{0{,}05}\left(\frac{1 - 0{,}9505}{0{,}9505}\right) - \frac{1}{10} = 4{,}2 \cdot 10^{-3}.$$

Кезектегі алынған орташа күту уақыты жүздеген сағатпен өлшенеді, яғни орташа есеппен компьютерлер жөндеуді 0,42 сағ күтеді.

7.5. ЖАППАЙ ҚЫЗМЕТ КӨРСЕТУ ЖҮЙЕЛЕРІН КОМПЬЮТЕРЛІК МОДЕЛЬДЕУ

Егер зерттелінетін ЖҚЖ үшін оның сипаттамаларын бағалауға арналған аналитикалық көрініс алу мүмкін болмаған немесе берілген көрініс практикада қолдану үшін тым күрделі болып шыққан жағдайда, ЖҚЖ тиімділігін компьютерлік модельдеу көмегімен зерттеуді орындауға болады. ЖҚЖ модельдеудің әр түрлі тәсілдерінің нақты мысалдары [2] қарастырылған.

ЖҚЖ модельдеу үшін алгоритмдерді құру әдісінің келесі негізгі әдістері қолданылуы мүмкін [2]:

- ауыспалы қадамы бар мезгілді модельдеуі (немесе модельді уақыттың уақиғадан уақиғаға қозғалу әдісі);
- тұрақты қадамы бар мезгілді модельдеу;
- өтініштің тізбекті өткізуі;
- өтініштің сатылы тізбекті өткізілуі.

Берілген алгоритмдер бір мәннен басқасына модельді уақыт сағатында сақталатын модельді уақыттың қозғалу әдісімен ерекшелінеді. Бірінші екі алгоритмдердің мәнін қысқаша қарастырайық, одан кейін көбіне ЖҚЖ модельдеу кезінде қолданылатын өтініштің тізбекті өткізілу әдісін егжей-тегжейлі сипаттайық.

Уақыттың уақиғадан уақиғаға қозғалуын пайдалану кезінде бастапқы күйдегі модельді уақыттың сағаты нөлге орнатылады және жүйе күйінің өзгеруіне әкеп соғатын болашақ уақиғаның пайда болу уақыты анықталынады [5]. Осыдан соң модельді уақыт сағаты жақын уақиғаның пайда болу уақытына өтеді және сол мезеттк өткен уақиғаны ескере отырып жүйе күйі, сонымен қатар болашақ уақиғаның пайда болу уақыты жайлы мәлімет жаңаланады. Кейін модельді уақыт сағаты келесі жақын уақиғаның пайда болу уақытына жылжиды, жүйе күйі жаңаланады және болашақ уақиға уақыты анықталынады және с.с. модельді уақыттың бір уақиғаның пайда болу уақытынан басқасының райда болу уақытына жылжу процесі алдын-ала көрсетілген қандай да бір тоқтату шарты орындалмағанға дейін жалғасады. Осындай имитациялық модельде барлық өзгерістер тек уақиға пайда болған кезде ғана орын алатындықтан, жүйенің әрекетсіздік периоды жай өткізіп жіберіледі және сағат бір уақиғаның пайда болу уақытынан басқасының пайда болу уақытына ауыстырылады.

Тұрақты қадамы бар модельді уақыттың жылжуы кезінде модельді уақыт сағаты нақты Δt мәніне арналған уақыттың дәл Δt бірлігіне жылжижы. Сағаттың әр жаңаруынан кейін алдыңғы уақыт Δt интервалы

аралығында қандай да бір уақиға жүргенін анықтау мақсатымен тексеру орындалады. Егер осы интервалға бір немесе бірнеше уақиға жоспарланған болса, берілген уақиға интервалдың аяғында жүрді деп есептелінеді, содан кейін жүйенің күйі мен статистикалық есептегіштер сәйкесінше жаңарады.

Тұрақты қадам арқасында уақыттың жылжуы екі кемшілікке ие:

- уақиғаның жүру аралығында интервал соңындағы уақиғаны өңдеумен байланысты қатенің пайда болуы;
- егер шынымен де әр түрлі уақытта жүретін уақиға бір уақыттық деп қарастырылса, қандай уақиғаны бірінші өңдеуді шешу қажеттілігі.

Осындай мәселелерді Δt интервалын ұзақтығы азырақ қылып жартылай шешуге болады, бірақ онда уақиғаның пайда болуын тексеру саны артады, ол тапсырманың орындалу уақытының көбеюіне әкеліп соғады. Осыдан ЖҚЖ модельдеуде тізбекті уақиғалар арасындағы уақыт интервалдары ұзақтығы бойынша біршама ерекшеленетін кезде, тұрақты қадам көмегімен уақыттың жылжуы қолданылмайды.

Бірқатар жағдайда ЖҚЖ өтінішке қызмет көрсету процестерін модельдеу кезінде өтінішті тізбекті өткізу әдісі бойынша модельдеуші алгоритм құрған ыңғайлы [2]. Бұл әдіс жүйеде жүріп жатқан уақиғалар бір біріне тәуелді болмаған кезде қолданылуы мүмкін. Модельдеуші алгоритм жеке өтініштердің тарихын олардың жүйегі түсу тәртібінде бірізді қайта жаңғыртады. Алгоритм басқа өтініштер жайлы ақпаратқа берілген өтінішке одан әрі тәртіпте қызмет көрсету жайлы мәселені шешу үшін қажет болса ғана жүгінеді.

Тізбекті өткізілім әдісінің екі түрі бар:

- жалғыз өтінішті өткізу;
- өтініш ағысын өткізу.

Берілген әдіс өтініштің қалыптасу және қызмет көрсету ретімен ғана ерекшеленеді. Бірінші күтілімі бар бір каналды ЖҚЖ модельдеуге қолданылатын жалғыз өтінішті өткізу[2] әдісін қарастырайық. Модельдеу процесінде кезекті өтінішті қалыптастыру (бұл үшін өтініштің келу уақытының арасындағы уақыт интервалын анықтайтын кездейсоқ сан мен оған қызмет көрсетудің кездейсоқ уақыты түрлендіріледі) және қызмет көрсету уақытының басталуы мен аяқталуын анықтап, оның қызмет көрсетуін орындау қажет.

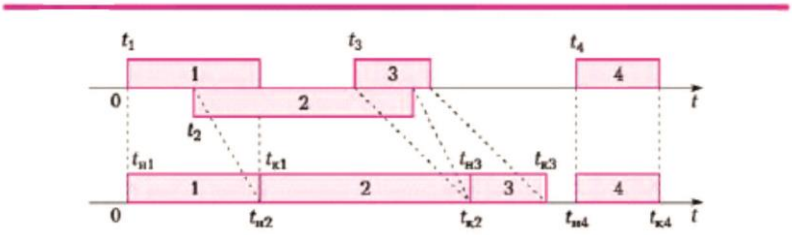
Модельдеудің басында модельді уақыт сағаты нөлге орнатылады. Бірінші өтініш жүйеге $t_1=A_1$ уақыт мезетінде түсті делік. Мұнда A_1– өтініштің түсуі арасындағы уақыт болып табылатын кездейсоқ

шаманы іске асыру. Уақыт t_1мезетінде қызмет көрсету каналы бос болғандықтан, өтінішке қызмет көрсету уақыт $t_{K1}= t_1$мезетінде басталады.Кейін бірінші S_1өтінішке қызмет көрсету уақытының кездейсоқ шамасын алу керек. Сонда 1-ші өтінішке қызмет көрсету мезетінің аяқталуы $t_{K1}= t_{R1}+ S_j$.Сонымен қатар қызмет көрсетілген өтініштің есептегіш санына бірлік қосылады.

Одан әрі келесі 2-ші өтініштің түсу уақыты анықталады. Ол үшін өтініштің A_2түсуі арасындағы уақыт қалыптастырылады және 2-ші өтініштің түсу уақытының мезетін анықталынады $t_2= t_1+ A_2$. Егер берілген мезетте, мысалы канал бос болмаса, онда 2-ші өтініш үшін ұзақтығы $\Delta T_{кут}= t_{к1} - t_2$күту периоды басталады. Бұл өтінішке қызмет көрсету $Д_2 = t_{R1}$мезетінде басталады. Одан әрі 2-ші өтінішке қызмет көрсету уақытының кездейсоқ шамасын S_2анықталынады. Онда 2-ші өтінішке қызмет көрсетудің аяқталу мезеті $t_{к2} = t_{п2} + S_2$.Қызмет көрсетілген өтініш есептегішінің санына бірлік қосылады. Кейінгі өтініштерге ұқсас тәсілмен қызмет көрсетіледі.

Төрт өтінішті қалыптастыру және оған қызмет көрсету процесі 7.4 суретте көрсетілген. Жоғарғы уақыттық осі өтініштің келу уақыты (тік бұрыштың сол шекарасы) мен қызмет көрсетілуін (тік бұрыштың ұзындығы), ал төменгісі – өтінішке қызмет көрсету уақытының басталуы (тік бұрыштың сол шекарасы) мен аяқталуын (тік бұрыштың оң шекарасы) көрсетеді.

Өтініштің тізбекті өткізілім әдісінің екінші нұсқасы бірінші барлық өтініштің ағымы қалыптасатын, ал кейін өтінішке қызмет көрсету процесі басталатын әдіс болып табылады [2]. Сонымен



Сур. 7.4. Өтініштің бірізді өткізілім әдісінің көрінісі

қатар барлық өтініштер біркелкі деп болжанады. Бұл жағдайда өтініш жайлы деректерді ЭЕМ жадында сақтау үшін сандардың ауқымын құру керек. Осындай ауқымның мөлшері алдын ала белгісіз болғандықтан, оны белгілі-бір қормен таңдауға тура келеді,

ол жады босқа шығындауға әкеп соғады.

Одан әрі өтініштің бірізді өткізу әдісінің екінші нұсқасы қолданылатын, шексіз күтілімі бар бір каналды ЖҚЖ C# тілінде модельдеу бағдарламасының мәтінінің кесіндісі көрсетілген.

Кіретін өтініш ағысын қалыптастыру

```
   : double time = 0;//Модельді уақытты нөлдеу
int i, Nin = 0;//Түсетін өтініш есептегішін нөльдеу
//кіретін өтінішті қалыптастыру
for (i= 0; i<Zmax; i++)
{
   //Кезекті өтініштің түсу уақытын қалыптастырytime=
   time+ rndexpo(lambda);
   //Егер модельдеу уақыты аяқталмасаif
   (time<modelTime)
   {
      // онда түсетін өтініш санын арттыруNin++;
      // өтініштің келу уақытын есте сақтауzarrive[i]
      = time;
      // өтінішке қызмет көрсету ұзақтығын
      қалыптастыру
      zserve[i] = rndexpo(mu);
   }
   elsebreak; //модельдеу уақыты аяқталу кезіндегі
   шығу
```

Бағдарламада қолданылатын тұрақты келесі тағайындауға ие:

lambda–кіретін өтініш ағымының қарқындылығы;

mu–өтінішке қызмет көрсету ағымының қарқындылығы;

Zmax–қызмет көрсетілген өтініштің максималды саны (өтініштің келу уақыты мен қызмет көсетілуі жайлы ақпарат сақталған z_arriveжәнеz_serve ауқымының мөлшерін анықтайды);

modelTime–модельдеу уақыты.

Бағдарламаның ұсынылған кесіндісінде функция дәлелімен анықталынған параметрі бар экспоненциалды таралған кездейсоқ шаманы түрлендіретін rnd_expo пайдаланушы функциясы қолданылған.

Одан әрі өтініш өткізілу бағдарламасының кесіндісін келтіреміз.

```
time= 0;   // Модельді уақытты нөльдеу
intNserve= 0;    // Қызмет көрсетілген есептегіш
санын
//нөлдеу
//барлық түскен өтінішті сұрыптуfor
(i= 0; i<Nin; i++)
{
   // Ағымдағы өтінішке қызмет көрсету басталуын
   анықтауif (z arrive[i] <time) timeServe = time;
   else timeServe = z arrive[i];
   //егер модельдеу уақыты аяқталмасаif
   (timeServe<modelTime)
   {
     // онда өтінішке қызмет көрсету уақытының
     басталуын
     //есте сақтау
     z begin[i] = timeServe;
     time = timeServe + z serve[i];
     //қызмет көрсету уақытының аяқталуын есте
     сақтауzend[i] = time;
     //қызмет көрсетілген өтініш санын
     арттыруNserve++;
   }
   elsebreak; //модельді уақыттың аяқталуы кезіндегі
   шығу
}
// соңғы өтінішке қызмет көрсету уақытының
аяқталуыdoubletimeEnd= time;
```

Бағдарламаны әзірлеу кезінде, моделді уақыттың аяқталуына дейін қызмет көрсетіліп бастаған өтінішке міндетті түрде қызмет көрсетілетіндігі болжанғанын ескерейік. Одан кейінгі өтініштерге қызмет көрсетілмейді.

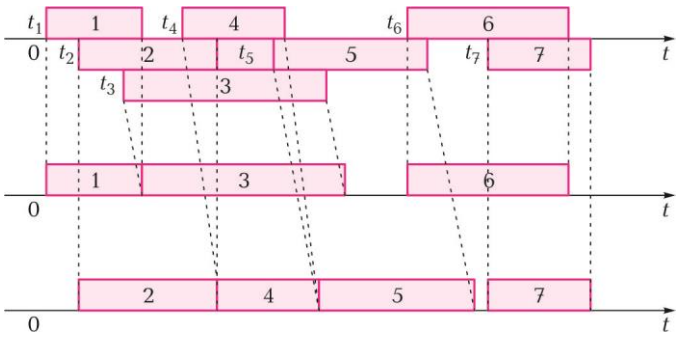
ЖҚЖ қызмет ету сипаттамаларын есептеу мен экранға шығаруға арналған бағдарлама кесіндісін елестетейік.

```
double timeWait = 0;  // өтініштің кезекте күту
уақыты
double timeSys = 0;// өтініштің жүйеде болу уақыты
double timeWork = 0; // жүйенің жұмыс істеу уақыты
//қызмет көрсетілген
өтініштерді сұрыптауfor (i =
0; i <Nserve; i++)
{
```

```
    //кезекте күту уақытын
    жинақтаytimeWait += (z begin[i] - z
    arrive[i]);
    // жүйеде болу уақытын жинақтаytimeSys
    += (zend[i] - zarrive[i]);
    // жүйенің жұмыс істеу уақытын
    жинақтаytimeWork += zserve[i];
}
Console.WriteLine("Түскен өтініш саны:{0)", Nin);
Console.WriteLine("Қызмет көрсетілген өтініш
саны:{0)", Nserve);
Console.WriteLine("Кезекте күткен орташа
уақыт:{0^4)", (double) timeWait/ Nserve);
Console.WriteLine("Жүйеде болған орташа уақыт:
{0:f4}", (double) timeSys/ Nserve);
Console.WriteLine("ЖҚЖ қолдану коэффициенті:
{0:f4}", (double) timeWork/timeEnd);
```

Өтінішті бірізді өткізудің қарастырылған әдісін көп каналды ЖҚЖ модельдерін құру үшін де қолдануға болады.

Мысал үшін 7.5 суретте екі каналды ЖҚЖ жұмысын модельдеу процесі көрсетілген. Суретте көрсетілгендей, берілген жағдайда



Сур. 7.5. Екі каналды ЖҚЖ модельдеуге арналған өтініштің бірізді өткізу әдісінің көрінісі

Өтініштің кіретін ағымы (жоғарғы уақыттық ось) қызмет көрсетудің екі құрылғысына (төменгі уақыттық ось) таралады. Әрі, өтініштің келген мезетінде бос құрылғылар бар болса, онда оған олардың кез-келгенімен қызмет көрсетіледі (әдетте бірінші босы

таңдалынады). Егер барлық қызмет көрсету құрылғылары бос болмаса, онда өтініш ерте босайын құрылғығы бағытталады.

Одан әрі servCount канал санымен күтілетін көп каналды ЖҚЖ өтінішке қызмет көрсету және сипаттамаларды анықтау бағдарламасының кесіндісі ұсынылған(бұл кезде өтінішті қалыптастыру бір каналды ЖҚЖ өтінішті қалыптастыруға толығымен сәйкес келеді).

```
// басында барлық қызмет көрсететін
құралдар бос for (j = 0; j <servCount; j++)
endTime[j] = 0;
int Nserve = 0; // Қызмет көрсетілген өтінішті
есептегіш
//санын нөльдеу
double timeWait = 0;  // өтініштің кезекте күту
уақыты
double timeSys = 0; // өтініштің жүйеде болу
уақыты
double timeWork = 0;  // жүйенің жұмыс істеу
уақыты
//түскен өтініштерді
сұрыптауfor (i= 0; i<Nin;
i++)
{
   //Ерте босайтын ind каналының нөмірін
анықтаймыз
   ind = 0;
   for (j = 1; j <servCount; j++)
   if (endTime[j] < endTime[ind])
   ind = j;
   // егер барлық қызмет көрсету
   құралдары бос //болмаса қызмет
   көрсетудің басталу уақытын
   // анықтауif (z arrive[i] <
   endTime[ind]) timeServe =
   endTime[ind]; else
     timeServe = z arrive[i];
   // егер уақыт аяқталмасаif
   (timeServe < modelTime)
   {
// онда өтінішке қызмет көрсетудің басталу уақытын
есте сақтауз
     zbegin[i] = timeServe;
     // қызмет көрсетудің аяқталу уақытын есте
```

```
        сақтау szend[i] = timeServe + zserve[i];
        // жүйедегі жұмыс уақытын жинақтаytimeWork
        += z serve[i]; endTime[ind] = z end[i];
        // қызмет көрсетілген өтініш санын
        арттыруNserve++;
        // кезектегі күту уақытын
        жинақтаytimeWait+= (zbegin[i] -
        zarrive[i]);
        // жүйеде болу уақытын жинақтаytimeSys +=
        (z end[i] - z arrive[i]); if (timeEnd <
        endTime[ind]) timeEnd = endTime[ind];
    }
    elsebreak; //модельдеуді аяқтау
}
Console. WriteLine("Түскен өтініш саны:{0)", Nin);
Console.WriteLine("Қызмет көрсетілген өтініш
саны:{0)", Nserve);
Console.WriteLine("Кезекте күткен орташа
уақыт:{0^4)", (double) timeWait/ Nserve);
Console.WriteLine("Жүйеде болған орташа уақыт:
{0:f4}", (double) timeSys/ Nserve);
Console.WriteLine("ЖҚЖ қолдану коэффициенті:
{0:f4}", (double) timeWork/timeEnd/servCount);
```

ЖҚЖ соңғы талқыланатын модельдеу әдісі – өтінішті сатылы бірізді өткізу әдісі, әр түрлі басымдылықғы бар ЖҚЖ модельдеуге арналған. Берілген әдіс өтініштің бірізді өтілімі әдісіне негізделген және бірінші басымдылығы жоғарырақ өтінішке қызмет көрсету процесі, ал кейін басымдылығы төмендеу өтінішке қызмет көрсету процесі модельденеді. Сонымен қатар қықмет көрсету құрылғысының жұмыс уақытының бөлігі басымдылығы жоғарырақ өтініштерге қызмет көрсетумен бос емес болатындығы ескеріледі. Басқаша айтқанда, басымдылығы төмендеу өтініштерге қызмет көрсету құрылғысы бос тұрған уақытта ғана қызмет көрсетілкі керек. Сонымен бірге абсолютті басымдылық іске асырылады, онда басымдылығы төмендеу өтінішке қызмет көрсету мезетінде келген басымдылығы жоғарылау өтініш басымдылығы төмендеу өтінішті ығыстырады. Берілген өтініш қызмет көрсету құрылғысы босаған соң қызмет алуын жалғастырады.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1. ЖҚЖ деген не?Осындай жүйеге мысал келтір.
2. ЖҚЖ негізгі компоненті қандай? Олар қандай параметрлерді береді?
3. ЖҚЖжіктеудің негізгі әдістері қандай? ЖҚЖ жіктеу беріңіз.
4. ЖҚЖ қандай негізгі сипаттамаларын білесіз?
5. Бір каналды ЖҚЖ деп нені түсінеді? Оны суреттеңіз.
6. Бір каналды ЖҚЖ сипаттамаларын қалай аналитикалық есептейді?
7. Көп каналды ЖҚЖ қалай түсінесіз? Оны суреттеңіз.
8. Көп каналды ЖҚЖ қалай аналитикалық есептеуге болады?
9. Қандай ЖҚЖ жабық деп аталады?
10. Уақығадан уақығаға модельді уақытты жылжытып модельдеу әдісі неге негізделген?
11. Тұрақты қадамы бар модельді уақытты жылжытып модельдеу әдісі неден тұрады?
12. Өтініштің бірізді өтілімі әдісі неге негізделген?
13. Өтініштің сатылы бірізді өткізілім әдісі неден тұрады?

КОМПЬЮТЕРЛІК ЭКСПЕРИМЕНТТІ ЖОСПАРЛАУ

8.1. НЕГІЗГІ ҰҒЫМДАР

Эксперименттің мақсаты белгіленгенін және машиналық имитация моделін қолдану шешімі қабылданғыннан кейін, алдағы эксперименттің бастапқы жоспарлауын жүргізу тиімді болып саналады.

Қажетті ақпаратты нысаналы және тиімді алу үшін зерттеушіде толық жоспар болу керек. Уақыт пен шығынға арналған жоспарлық шектеулер зерттеушінің құзырындағы ресурстарға сәйкес келуі қажет. Эксперимент неғұрлым қиын әрі қымбат болса, сол кезеңге соғұрлым назар аудару қажет.

Эксперимент жоспары төмендегілерді анықтайды:

- Компьютердегі есептеулер көлем;
- Компьютердегі есептеулерді жүргізу реті порядок;
- Жиынтық тәсілдері және үлгілеу нәтижелерін статистикалық өңдеу.

Эксперименттерді жоспарлау келесі мақсаттардан құралады:

- Нәтижелердің дәлдігі мен сенімділігіне көзделген талаптарды жүргізу барысында үлгілеудің ортақ уақытын қысқарту;
- әрбір бақылаудың ақпараттылығын арттыру;
- зерттеу процесінің құрылымдық негіщін құру. Эксперименттерді

жоспарлау теориясының негізгі ұғымдарын құрастырамыз, толығырақ [12 қосымшада. Экспериментті жоспарлау кезіндегі айрықша келісімді үлгі «қара жәшік» (8.1сурет) деп аталатын абстрактылы сызба болып табылады.

- Сондай тәсілдеме кезінде ішкі және сыртқы айнымалалар ажыратылады: $x_1, x_2,..., x_k; y_1, \mathbf{y_2}, \bullet\bullet\bullet, \mathbf{y}_m$. Кез келген айнымалы не факто, не реакция болып табылады.

8.1 сур «Қара жәшік» күйіндегі жүйе үлгісі

Мысалы, тек екі түрлі айнымалы бар делік. Олар: *х*пен*у*. Егер эксперименттің мақсаты х айнымалысының у айнымалысына әсерін зерттеу болса, онда х — фактор ал у — реакция болып табылады.

Әрбір фактор x_i, i =1...кэксперименте бірнеше айнымалылардың арасынан *деңгей* деп аталатын бір мәнге ие бола алады. Барлық факторлардың белгіленген деңгейлер жиынтығы қарастырылып жатқан жүйенің мүмкін жағдайларының біреуін анықтайды.

Факторлардың әрбір белгіленген деңгейлер жиынтығы *факторлық кеңістік* деп аталатын көпөлшемді кеңістіктегі анықталған нүктеге сәйкес келеді.

Байланыс күйінде өрнектеуге болатын факторлар деңгейлері мен жүйелер реакциясының арасындағы толақтай анықталған байланыс бар.

$$\tilde{y}_j = \varphi_j(x_1, x_2, \ldots, x_k),\ j = \overline{1,m}.$$

Факторлармен реакцияларды байланыстыратын функцияна реакция функциясы деп атайды, ал оған сәйкес геометриялық бейне реакция жазықтығы деп аталады. Реакция функциясы жайлы хабардар болу зерттеушіге жүйенің жұмысы туралы толық ақпарат береді. Ол жүйедегі қызықтыратын сипаттамаларды басқаруды, белгіленген тиімділік өлшемдеріне сәйкес факторлардың тиімді мәндерін жинақтауды оңай үйрене алады. Сондықтан зерттеушіге үлгілеу кезінде реакция қандай қалыпта факторға бағынышты болатындығын білу маңызды.

Сонымен қатар зерттеушіге у, j =l...m тәуелділік түрлері адын ала белгісіз, сол себепті жуық байланыс қолданылады:

$$\tilde{y}_j = \varphi_j(x_1, x_2, \ldots, x_k),\ \varphi = \overline{1,m}.$$

Әдетте jl тәуелділігі ретінде бірінше және екінші дәрежелі полиномдар сайланады. Олардың коэфициенттері эксперименттің мағлұматтары бойынша (мысалы, ең кіші квадраттар тәсілінің көмегімен) табылады. Коэфициенттерді есептеленетін ресустарының минималды шығыны (мысалы, сынақтардың

минималды саны) кезінде ішкі айнымалыдардың мәндер ережелерін арнайы тұдырымдаулары бойынша түрлендіре математикалық үлгіні құрастырып, оның сипаттамаларын бақылайтындай етіп орнату қажет.

Эксперименттерді жоспарлау кезінде факторлардың негізгі ерекшеліктерін анықтау қажет. Эксперименттерді жүргізу кезіндегі факторлар басқарылатын және басқарылмайтын, зерттелетін және зерттемейтін, бақыланатын және бақыланбайтын, есептік және сапалық, белгіленген және кездейсоқ болуы мүмкін.

Фактор басқарылатын деп аталады, егер оның деңгейлері эксперимент кезінде зерттеушімен нысаналы түрде анықталған жағдайда. Фак¬тор бақыланатын деп аталады, егер оның мәндері бақыланады және тіркеледі. Фактор зерттелетінге жатады, егер ол қосымша нысана (мысалы, эксперименттің дәлдігін арттыру үшін) ретінде емес жүйенің ерекшеліктерін зерттеуге арналған үлгіге кірген жағдайда. Фактор есептік болып табылады, егер оның мәні— реакцияға әсер ететін сандық шамалар, қарама-қайшы жағдайда фактор сапалық болып есептелінеді. Мысалы, БҚКЖ есептік факторлар деп сұрыныстардың қарқынды кіріс ағыны, қызмет көрсету ағынының қарқыны, жинақтауыш сыйымдылығы, қызмет көрсететін арналардың саны және т.б. есептелінеді, ал сапалық факторлар— тізімге тәртіп орнатулар, тізімнен талғаулар, сұраныстарға ағындармен қызмет көрсету және т.б.

Фактор белгіленген деп аталады, егер экспериментте экспериментаторды қызықтыратын факторлардың барлық мәндері зерттелетін болса, ал егер экспериментаторды қызықтыратын факторлар мәндерінің қосындысынан тек кейбір кездейсоқ сұрыптауларды зерттейтін болса онда фактор кездейсоқ деп аталады. Кездейсоқ факторлар негізінде болжамалы тұжырымдар және экспериментте зерттелмеген фактор мәндері жасалынуы мүмкін.

Факторға қойылатын негізгі талаптар факторды басқару талаптар, нысанға тура әсер ету талаптары, факторлардың үйлесімділігі мен тәуелсіздігі талаптары болып табылады.

Фактордың басқарымдылығы деп барық сынақ аралығында таңдалған керекті фактор деңгейін тұрақты орнату және ұстап тұру немесе берілген бағдарламаға сай өзгеретін мүмкіндік деп түсінеді. Нысанға тура әсер ету талаптары үлкен маңыздылыққа ие, өйткені фактор басқа факторлардың функциясы болып табылса, оны басқару қиын. Факторлардың үйлесімділігі олардың барлық қиыстыруларын іске асыруға болатындығын, ал тәуелсіздік басқалардың деңгейіне тәуелсіз кез-келген деңгейде факторды орнату мүмкіндігіне сай

келетіндігін білдіреді.

Машиналық тәжірибелерді жоспарлау кезіде екі құраушыны бөлуге болады: стратегиялық және тактикалық жоспарлау.

Стратегиялық жоспарлаутәжірибе қоюшының қарамағында бар қордың шектеуін ескере отырып, ЭЕМ іске асырылған модель көмегімен жүйе жайлы қажетті ақпаратты алу тапсырмасын шешуді өзінің негізгі мақсатына қояды. Машиналық тәжірибені жоспарлау кезіндегі жиі тапсырмалардың санына модельдеуге кететін машина уақытын азайту, модельдеу нәтижесінің дәлділігі мен нақтылығын арттыру модельдің барабарлығын тексеру міндеттері жатады.

Тактикалық жоспарлаутәжірибе жоспарында қарастырылған машина модельінің әр сынақ топтамасын жүргізу әдісін анықтауды көрсетеді.

8.2. ТӘЖІРИБЕНІ СТРАТЕГИЯЛЫҚ ЖОСПАРЛАУ

Модельдеудің стратегиялық жоспарын құру процесі негізгі екі сатыдан тұрады: тәжірибенің құрылымдық моделін құру; функционалды модельді құру.

Құрылымдық модельді құру кезінде факторлар саны мен әр факторға арналған деңгей саны таңдалады. Осы көрсеткіштерді таңдау тәжірибенің мақсатына, фактор өлшемдерінің длділігіне, сызықсыз эффектке қызығушылық және с.с. анықталынады. Осы таңдауға қордың шектелгендігі арқасында орын алатын мүмкін өлшеу сандарыныңшектелгендігі әсер етпеуі керек, яғни құрылымдық модель не істеу керектігіне сүйеніп таңдалынады.

Құрылымдық модель, сәйкесінше, түрге ие

$$N_s = q_1 q_2 \ldots q_k,$$

МұндағыNs–тәжірибе элементтерінің саны; k–тәжірибе факторларының саны; qi–і-ші фактор дәрежесінің саны.

Мұнда элемент ретінде фактордың нақты мән жиынында жүргізілетін қарапайым тәжірибе деп түсінеді.

Қажетті фактордың саны мен түрін таңдау кезінде төмендегідей мәселелер бірізді шешіледі:

1)бізді бірінші кезекте қызықтыратын қандай жауапты (шығатын ауыспалы) анықтайды;

2)осы ауыспалыларға қандай факторлар әсер ететіндігін

белгілейді. Жүйенің көпшілігі Парето қағидасына сай жұмыс істейтіндіктен, онда ең маңызды факторлардың 20 %жүйенің қасиетінің 80 %анықтайды, ал фактордың қалған 80 % оның қасиетінің тек 20 % анықтайды;

3)өлшенетін және бақарылатын факторларды бөледі және олардың мүмкін деңгейін анықтайды. Әр фактор үшін деңгей санын минимальды мүмкін, бірақ тәжірибе мақсатына жету үшін жеткілікті таңдау керек. Барлық факторлар үшін бірдей q санының бірдей қашықтықтағы деңгейі (екі-үш) модель зерттеуін айтарлықтай жеңілдетеді. Бұл жағдайда құрылымдық модель төмендегі түрге ие

$$N_s = q^k.$$

Функционалды модельжауаптың нағыз өлшеміне қызмет ететін құрылымдық модель элементтерінің санын қамтамасыз етеді, яғни әр түрлі ақпараттық нүктенің қанша болуы қажеттігін анықтайды. Ұқсас функционалды модельдер не аяқталған не аяқталмаған болуы мүмкін. Егер жауап өлшемінде оның барлық элементтері қатысса, функционалды модель аяқталған деп аталады, яғни Nf= Ns.Егер орын алған жауап саны элемент санынан аз болса, функционалды модель аяқталмаған деп аталады, яғни Nf<Ns.Мінсіз жағдайда функционалды модель құрылымдыққа сай келеді. Бірақ модельді зерттеулердің көпшілігі уақытқа, ақшалай қаражатқа және есептегіш машиналардың өндірімділігіне негізделген шектеуге ие болғандықтан, функционалды модельді құру кезінде қалау мен ресурс арасында лайықты ымыраны таңдау керек. Функционалды модельді құру кезіндегі одан кейінгі әрекет жүргізілетін зерттеудің мақсатына тәуелді. Мысалы, реакция бетін зерттеу қажет жағдайда $\tilde{y}_j = \varphi_j(x_1, x_2, \ldots, x_k),\ j = \overline{1, m}$ тәуелділік функциясы коэффициентін анықтау үшін жеткілікті толық немесе толық емес факторлы тәжірибені таңдау жүзеге асырылады.

Стратегиялық жоспарды құру жайлы толығырақ ақпаратты [5, 12] алуға болады.

8.3. КОМПЬЮТЕРЛІК ТӘЖІРИБЕНІ ТАКТИКАЛЫҚ ЖОСПАРЛАУ

Жүйенің машиналық модельімен тәжірибені тактикалық жоспарлау тәжірибе үшін бөлінген машина ресурстарын тиімді

пайдалану және стратегиялық жоспарлау кезінде құрылған тәжірибе жоспарымен белгіленген модель сынағын жүргізудің нақты әдістерін анықтау мәселелерімен байланысты. Сонымен қатар мәселелер шешіледі:

- бастапқы шарттырдың анықтау және олардың модельдеу кезінде орнатылған жетістікке әсерін анықтау;
- модельдеу нәтижелерінің дәлдігі мен нақтылығын қамтамасыз ету;
- жүйе модельдерімен имитациялық тәжірибенің автоматты түрде тоқтату ережелерін таңдау.

8.3.1. Орнатылған тәртіпке жетудің бастапқы шарттары мен олардың әсер етуі

Машиналық тәжірибені жүргізу кезіндегі бірінші мәселе модельді іске қосулың бастапқы шартының әсерінен машиналық модельдің бастапқы периодының бұрмалануына негізделеді. Бұл кезде N таңдама көлемінің өзгеруі кезіндегі бағаланатын параметр p (p параметрі ретінде қызмет көрсетуг.е өтінімді қабылдау ықтималдылығы алынған) қадімгі тәртібі 8.2 суретте көрсетілген.

Бастапқы периодтың алынатын деректің әсерін азайтудың үш негізгі жолы бар.

1. Ауыспалы периодтың деректер саны орнатылған тіртңп дерегінің санымен салыстырғанда мардымсыз болуы үшін, айтарлықтай ұззын есептегіш аралықты пайдалану, бірақ бұл кезде модельдеу уақыты біршама артады.

2. Модельдің орнатылған жұмыс тәртібіне жету үшін талап етілетін бастапқы аралық периодын қарастыруды алып тастау. Бұл жағдайда ауыспалы процес уақытын анықтау қиындық тудырады. Егер оны өте аз таңдаса, онда бастапқы шарттың әсері қалады. Егер оны тым үлкен таңдаса, онда пайдалы деректің бөлігі тасталынуының арқасында модельдеу уақыты артады.

Сур. 8.2. Параметр бағасының таңдама көлемімен өзгеру тәуелділігі

3. Қарапайымға жақын осындай орнатылған бастапқы шарттың күйін таңдау, және сол арқылы ауыспалы периодты азайту. Бұл жағдайда модельдеу кезінде жүйенің қарапайым күйін анықтау мәселесі туындайды.

Осылайша, осы әдістердің барлығы модельмен машиналық тәжірибені жүргізу кезінде ауыспалы процестің әсерін тек азайтуға, бірақ нөлге келтірмеуге мүмкіндік береді.

Одан әрі модельдеудің берілген дәлділігі мен нәтиженің нақтылығын қамтамасыз етуге мүмкіндік беретін таңдама көлемін анықтау мәселелерін қарастырамыз, ол [10]сипатталған.

8.3.2. Іріктеменің жиынтықтың орташа мағынасын бағалау

Есеп белгілі бір дәлдікпен және сенімділікпен минималды x уақыт ішінде орташа мәнінен кейбір зерттелген параметрдің белгісіз математикалық күту m бағалауын алудан тұрады. Орташа іріктемех,әртүрлі бастапқы жағдайлармен модельдеудің бірнеше имитациялық аралықпен алынған, кездейсоқ шама X ретінде қарастыруға болады (x іріктеуден іріктемеге өзгереді) және іріктеме мағынасының белгілері x_1, x_2,..., xN— бірдей тәуелсіз таралған кездейсоқ шамалар сияқты X_1, X_2, ..., X_N математикалық күтілімentжәне орташа квадраттық ауытқуыσ. Алдымен біз тәуелсіздіктің болжамын және модельдік жауаптардың қалыпты таралуын қолданамыз.Бұл болжам ықтималдықтар теориясының орталық шекті теоремасын қолдануда негізделеді, оның кездейсоқ шамаларын таралуын мәні бірдей қабылдауы, бірдей таратулары ықтималдық бар көптеген тәуелді емес шамалардың үлкен сандар сомасы жиынтығы болып табылатын, қалыпты таратуға жақын мәндерінен тұрады.Көптеген жұмыстың аяғында тәуелсіздік пен бірдей тарату талаптары қажет емес екендігі көрсетіледі. Жиі бұл шағын салдардың көп сан сомасына жауап қайтармауы жеткілікті болады. Сондықтан кездейсоқ айнымалы мәндердің көп мөлшерін аддитивтік әрекетінің нәтижесі болып табылатын күрделі имитациялық модельдеуге жауап қайтармау айнымалысы шамамен қалыпты түрде

таратады. Оның нәтижесінде, егер кездейсоқ шама X жақсы таратылса, онда орташа іріктеме x,тәуелсіз қадағалау бойынша табылған, сонымен қатар математикалық күтілімі жақсы таратылған

$$M(x) = \mu$$

және орташа квадраттық ауытқуы

$$\sigma(\bar{x}) = \sigma/\sqrt{N}.$$

Қатынастардың орындалуын талап етеміз

$$P(\mu - d \le \bar{x} < \mu + d) = \beta, \qquad (8.1)$$

Мұндағы x— іріктеме орташа мағынасы; d— бағаның дәлдігі; P — ықтималдылығы, μ ± d аралығыхқұрайды (сенімділік, зерттеушінің айтуымен).

Берілген сенімділік ß формуламен алынған x бағасының маңыздылығына байланысты

$$\beta = 1 - \alpha.$$

Егер математикалық күтілімі m белгілі бір дәлдікпен және сенімділік P (мәнінің деңгейі a) берілген, онда бұл дәлдікті қамтамасыз ететін минималды іріктеу көлеміформула бойынша анықталады

$$N = \frac{t_\beta^2 \sigma^2}{d^2},$$

мқндағы $t_ß$ саны теңсіздікпен анықталады $2\Phi(t_ß) = \beta$ немесе $\Phi(t_ß) = \beta/2$

Лапласа функциясының кестесі бойынша (бұл жерде

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{z^2}{2}} dz$$

Егер дәлділік d белгісіз болса және орташа квадраттық ауытқуσ, онда мәселені төмендегідей етіп орнатуға болады: іріктеу көлемі қандай болуы керек, ықтималдықпен қажетті бағалау үшін, β = 1 - a шектік табу m ± Δσ, мұнда Δσ — кейбір үлес σ (мысалы, Δσ = σ/2; σ/4; σ/6; σ /10; σ /20; ...).

Мысалы, кезінде Δσ = o/4 және P = 0,95 (σ = 0,05) аламыз β= 1,96, онда

$$N = \frac{(1{,}96\sigma)^2}{(\sigma/4)^2} = 61.$$

Басқа жағдайларда тәжірибелік эксперимент арқылы шығыс

дисперсиясын анықтау және дисперсияның s2бағасын алу керек, содан кейін қажетті бақылаудың жалпы санын есептеңіз. Онда іріктеу көлемі келесі өрнекпен анықталады:

$$N = \frac{t_\beta^2 S^2(n)}{d^2},$$

мұндағы дисперцияны бағалау формула бойынша алынған

$$S^2(n) = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2,$$

Мұндағы n— сынамалы эксперименттегі іріктеу көлемі.

8.3.3. Чебышев теоремасын қолдану

Чебышев теңсіздігінде айтылады, бұл берілген белгілі бір санға k (кемінде бір) және ерікті іріктеу x_1, x_2,..., xN көлемінде N кем дегенде 1 – 1/ k^2 өлшем үлесі жақын орналасады орташа мәнінен аспайтын қашықтықта - орташа квадраттық ауытқу. Бұл теңсіздік кез-келген таралуына жиынтығында әділ. Егер біз қалыпты таралу туралы болжамнан шықпасақ (әрқашан жеткілікті дәлдікпен қанағаттандырылмаса), Чебышев теңсіздігін қолдануға болады, ол келесі түрде болады

$$P(|x - \mu| > k\sigma) \le \frac{1}{k^2}.$$

Шарт қоямыз, онда бағалау кезіндеХаралықта болуы керек μ ± σ/m (m= 2, 4, 6, 10, ...) ықтималдылығымен P = 1 - α :

$$P\left(|\bar{x} - \mu| > \frac{\sigma}{m}\right) \le \sigma,$$

онда, теңдікті қолданып (8.1), аламыз

$$P\left(|\bar{x} - \mu| > \frac{\sqrt{N}}{m}\frac{\sigma}{\sqrt{N}}\right) \le \alpha = \frac{m^2}{N},$$

өйткені

$$k = \frac{\sqrt{N}}{m};\ \frac{1}{x^2} = \frac{m^2}{N}.$$

Кесте 8.1. Әр түрлі мағынадағы қателік кезіндегі іріктеу көлемі		
Ауытқу	Керекті іріктеу көлемі *N*	
	Орташа шекті теоремасы бойынша	Чебышев теоремасы бойынша
σ/2	15	80
σ /4	61	320
σ /6	138	720
σ /8	246	1 280
σ /10	384	2 000
σ /12	553	2 880
σ /20	1 537	8 000

Осы жерден керекті машиналық тәжірибенің іріктеу көлемін табамыз

$$N = \frac{m^2}{\alpha}$$

Алынған іріктеу көлемі қалыпты әлдеқайда көп, таралу жиынтық кезінде жеткілікті. Әр түрлі мағынадығы іріктеу көлемid кезінде α = 0,05 (β = 0,95) кест. 8.1. келтірілген

8.3.4. Пайыздық қатынастарды бағалау

Көптеген жағдайларда А машина тәжірибеніңінің мақсаты зерттелетін жүйенің жұмыс процесінің жай-күйімен анықталған оқиғаның P = P(A) жағдайының ықтималдығын бағалауды алу болып табылады. Ықтималдылық санын бағалау ретіндержиілік p = Ө/N,где Ө —тәжірибенің тиімді шығыс саны.

Жиілікті көрсетеміз p = Ө /N түрінде

$$\hat{p} = \frac{1}{N}\sum_{i=1}^{N} x_i,$$

мұндағы оқиғаның бақылау саны Аберілгенді жүзеге асыруN-ненбақылау кездейсоқ шама болып табылады, мағынасын

қабылдайтын $x_i = 1$ ықтималдылықпенр и $x_i = 0$ қосымша ықтималдылықпен 1 - р.

Онда

$$\bar{X}(N) = p;$$

$$S^2(N) = \frac{p(1-p)}{N}.$$

Лаплас теоремасы бойынша, жеткілікті үлкен N шамасындағы 0/N жиілігін кездейсоқ шама ретінде қарастыруға болады, ол математикалық болжамы X (N) және таңдаулы дисперсиясы $S^2(N)$ бар ықтималдықтарды бөлудің қалыпты заңымен сипатталады.Сондықтан Ɛ бағалау дәлдігі және β нақтылығы N санымен байланысты болып келеді. Ɛ дәлдігі және β нақтылығы бар бағалауды алуға қажетті таңдау көлемі осы арқылы анықталады.

$$P\left(p - \varepsilon < \frac{\theta}{N} < p + \varepsilon\right) = \Phi\left(\frac{p + \varepsilon - p}{\sqrt{p(1-p)}}\sqrt{N}\right) - \Phi\left(\frac{p - \varepsilon - p}{\sqrt{p(1-p)}}\sqrt{N}\right) = \beta,$$

Р ықтималдығы туралы алдын-ала деректерді алу мүмкіндігі болмаған жағдайда абсолютті дәлдікті пайдалану өз мәнін жоғалтады. Мұндай жағдайларда модельдеу нәтижелерінің қатысты дәлдігін анықтау қажет болады:

$$N = \frac{t_\beta^2 p(1-p)}{\varepsilon^2}.$$

Р мәні белгісіз болған жағдайда, тәжірибені тактикалық жоспарлау кезінде ерікті таңдалған n мәніне арналған алдын-ала модельдеу жүргізіледі. Алдын-ала модельдеу нәтижесі бойынша, $p_0 = \theta/n$, анықталады, одан кейін (8.2) қатынасы бойынша N өткізу саны есептеледі, ол үшін р орнына р0 мәні қолданылады. Мұндай N бағалау процедурасы машиналық тәжірибе кезінде бірнеше рет орындалады.

$$\varepsilon_0 = \frac{\varepsilon}{p}.$$

Онда қатынастар (8.2) келесі түрде

$$N = \frac{t_\beta^2(1-p)}{\varepsilon_0^2 p}.$$

Формула (8.3) сирек кездесетін оқиғалардың статистикалық

модельдеудің нақты ерекшелігін көрсетеді, бұл шағын р ықтималдықтарды жоғары дәлдікпенкеректі бағалау үшін өте үлкен санды іске асыру қажет. Тәжірибелік жағдайда ықтималдылықты рет ретімен бағалау үшін 10-кмақсаттық санын ең болмаса10k+l теңдей таңдап іске асыру.

8.3.5. Жиынтық дисперциясын бағалау

Жүйенің тиімділігінің көрсеткіші ретінде σ^2 дисперсиясын көрсетуге болады, бағасы s^2 ол машиналық тәжірибелердің жүргізу нәтижесі ретінде алынуы керек,кейбір сенімділік $\beta = 1 - a$.

Талап етеміз, қатынастар орындалуын

$$P\left(s^2 - d \le \sigma^2 \le s^2 + d\right) = \beta.$$

Екі еселік теңсіздікке айналдырамыз

$$s^2 - d \le \sigma^2 \le s^2 + d$$

мәндес

$$s^2(1-q) \le \sigma^2 \le s^2(1+q),$$

Мұнда q — сан, бағалаудың жақындығының дәрежесін сипаттайтын s^2 шынайы дисперцияға σ^2; $q = d/s^2$.

Қолдануға тиімді болу үшін қарастырылған сатистиканы қолданамыз

$$\chi^2 = \frac{s^2(N-1)}{\sigma^2},$$

заң бойынша таралған χ^2 («хи-квадрат») N - 1 еркіндік дәрежесімен.

Таралу тығыздығы χ^2 келесі түрде болады

$$k_N(x) = \begin{cases} \dfrac{1}{2^{N/2}\Gamma\left(\dfrac{N}{2}\right)} x^{\frac{N}{2}-1} e^{-x/2}, x > 0;\ N \ge 1; \\ 0, x \le 0. \end{cases}$$

Мұнда $\Gamma(\lambda)$ — гамма-функциясы, анықталған

$$\Gamma(\lambda) = \int_0^{\lambda} t^{\lambda-1} e^t \mathrm{d}t.$$

Атап айтқанда аргументтің жалпы мағынасы Г(n +1) = n!

Үлкен N үшін орталық шектік теоремасына сәйкес қалыпты шамалардың таралуы $(\chi^2 - N)/\sqrt{2N}$ стандартты қалыпты таралуына ұмтылады

$$P\left(\frac{\chi^2 - N}{\sqrt{2N}} < x\right) \approx \Phi(x),$$

Осыдан теңдікті алуға болады

$$t_\beta^2 = \frac{q^2(N-1)}{2}$$

немесе

$$N = 1 + \frac{2t_\beta^2}{q^2},$$

мұнда t_β — стандартты қалыпты таралу квантильі, мәнін теңдеуден табамыз $2\Phi(t_\beta) = \beta$ Лапласа функциясының кетесі .

8.3.6. Автокорреляциялық деректер

Егер модельдің шығысындағы мәндері автокорреляцияланған болса, яғни кейінгі таңдалған мәндері алдыңғысына тәуелді болса, онда бұл іріктеуде тәуелсіз деректер іріктеуіне қарағанда аз ақпарат болады. Сонымен қатар талап етілетін іріктеу көлемі автокорреляция шамасына өте сезімтал.

Автокорреляция функциясын бағалау әдісін қарастырайық және оның параметрлік бағалауларын құрастыру кезінде қолдануға болады, олар корреляциялық деректердің жағдайда

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i;$$

$$\sigma_N^2 = \frac{\sigma_x^2}{N}\left[1 + 2\sum_{p=1}^{m}\left(1 - \frac{p}{m+1}\right)\rho_{p,x}\right],$$

Мұнда σ^2_x — жиынтық дисперсия; m — қарастырып жатқан автокорреляцияның максималды тереңдігі; p_{px} - p-й автокорреляция

коэффициентті; p= 1,2,..., m.

Корреляциялық коэффициенттерді бағалау үшін сынақ іріктеуімен тәжірибе жүргізу және формуланы қолдану қажет

$$\rho_{p,x} = \frac{\sum_{i=1}^{N-p}\left[(x_i - \bar{x})(x_{i+p} - \bar{x})\right]}{S^2},$$

Мұнда S^2 — жөнделген іріктеу дисперсиясы, негізінде алынған N сынау тәжірибелерінде өлшенген.

Минимальды қажетті іріктеу көлемін формула бойынша анықтауға болады

$$N = \frac{t^2 S^2\left[1 + 2\sum_{p=1}^{m}\left(1 - \frac{p}{m+1}\right)\rho_{p,x}\right]}{(\bar{x}d)^2},$$

где d— шынайы орташа мәннің салыстырмалы қателігі.

Есептелген корреляция коэффициенттерінің максималды тереңдігі m тәжірибеде таңдалған өлшемдердің N санының шамамен 10% құрайды.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1 Машиналық тәжірибені жоспарлау дегеніміз не?
2 Компьютерлік тәжірибені тактикалық жоспарлаудың мәні қандай?
3 Бастапқы жағдайлар модельдеу нәтижелеріне қалай әсер етеді??
4 Берілген дәлдік пен тәжірибенің минимальды құнын қамтамасыз ететін іріктеу көлемін анықтау мәселесі қалай шешіледі?
5 Іріктеу жиынтығының орташа мәні қандай дәлдікпен анықталады?
6 Оңтайлы іріктеу көлемін бағалау үшін Чебышевтің теңсіздігін қалай пайдалануға болады?
7 Белгілі дәлдікпен оқиғаның пайда болу ықтималдығын бағалау кезінде іріктеу көлемін қалай анықтауға болады?
8 Берілген дәлдікпен ауытқуды бағалау кезінде іріктеу көлемін қалай анықтауға болады?

9 Автокорреляциялық деректер үшін іріктеу көлемі қалай есептеледі?

9 Тарау

ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТТЕРІ

9.1. ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТТЕРІНІҢ ТАҒАЙЫНДАЛУЫ

Имитациялық модельдерді құру кезінде оларды әзірлеу үшін қажетті кейбір типтік функционалды мүмкіндіктерді бөлуге болады [5].

Осындай мүмкіндіктерге:

- 0-ден 1 дейінгі интервалда біркелкі тараған кездейсоқ санды түрлендіру;
- берілген таралу ықтималдылығы бар кездейсоқ шаманы түрлендіру;
- модельді уақытты қозғалтуды ұйымдастыру;
- шығатын статистикалық деректерді жинақтау және нәтиделері бар есеп құру және с.с. жатады.

Модельдеуші бағдарламаларда жалпы функционалды мүмкіндіктердің болуы имитациялық модельдерді жасап шығару процесін біршама жеңілдету мен тездетуге мүмкіндік беретін имитациялық модельдеудің арнайы бағдарламалық пакеттерін немесе сжүйелерін әзірлеуге әкеп соқты.

Әмбебап бағдарламалау тілін қалданумен салыстырғанда имитациялық модельдеу пакетін қолдану бірқатар ерекшеліктерді береді [5].

1. Имитациялық модельдеу пакеттері имитациялық модельді құру үшін талап етілетін функционалды мүмкіндіктердің көпшілігін автоматты түрде ұсынады, ол бағдарламалау үшін қажетті уақытты және жобаның жалпы құнын біршама қысқартуға мүмкіндік береді.

2. Имитациялық модельдеу пакеттері имитациялық модельдерді құру үшін қалыпты ортаны қамтамасыз етеді. Олардың негізгі

модельдеуші құрылымы сәйкес бағдарламаның әмбебап тіліндегі құрылымына қарағанда имитациялық модельдеуге көбірек келеді.

3. Имитациялық моделі пакеттердің көмегімен құрылған модельдеу, әдетте, түрлендіруге және пайдалануға оңай.

4. Имитациялық модельдеу пакеттері неғұрлым жетілдірілген тетіктері табылған қателерді анықтау механизмдерін қамтамасыз етеді, өйткені олар көптеген түрлердің автоматты түрде қателерді тексеруін орындайды. Және де модельге құрылымдық компоненттердің үлкен санының қажет болмағандығынан, кез келген қатені жасау мүмкіндігі азаяды.

Сол уақытта әмбебап бағдарламалау тілдерін қолдану, сондай-ақ бірқатар артықшылықтарға ие [5].

1. Бағдарламалау тілдерін көптеген әзірлеушілер біледі, имитациялық модельдеу пакеттері туралы айту мүмкін емес.

2. Әмбебап программалау тілдерінде жазылған имитациялық модельдерінің жылдамдығы, әдетте жоғары, модельдеу пакеттерінің көмегімен жасалған модельдерге қарағанда.

3. Бағдарламалау кезінде әмбебап тілдері имитациялық модельдеу пакеттеріне қарағанда икемділікті қамтамасыз етеді.

4. Әмбебап бағдарламалау тілдерінінің қолданылатын кезінде пайдаланылатын бағдарламалық жасақтама құны әдетте имитациялық модельдеу пакеттерінің құнынан төмен болады.

Осылайша, екі әдісті пайдалану артықшылығы бар, сондықтан модельді әзірлеуші олардың әрқайсысын мұқият қарастыруы керек.

Қазіргі күні бір бірімен әмбебаптылық деңгейімен, функционалды мүмкіндіктерімен, пайдалану саласы мен көптеген басқа да көрсеткіштер бойынша ерекшеленетін имитациялық модельдердің көптеген пакет саны әзірлнген. Одан әрі басқа модельдеу пакеттерімен салыстырғанда бірқатар артықшылықтарға ие Pilgrim атты имитациялық модельдеу пакеттерінің бірін қарастырамыз.

9.2. PILGRIM ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТІ

Pilgrim пакеті имитациялық объектілердің уақытша, кеңістіктік және қаржылық динамикасын модельдеуге арналған кең мүмкіндіктерді ұсынады [3].Оның көмегімен дискреттік-үздіксіз модельдер жасауға болады. Мәтін үлгісінде стандартты C ++ тілін

қолданып кез келген блоктарды кірістіруге болады.Pilgrim жүйесіндегі модельдер құрастырылады және сондықтан жоғары жылдамдықпен әсер етуге ие, ол аса жылдамдатылған уақыт масштабында басқару шешімдерін жасау мен нұсқаның бейімделген таңдауы үшін маңызды. Компиляциядан кейін алынған нысан кодты жасалып жатқан бағдарламалық кешендерге енгізуге немесе тапсырыс берушіге беруге (сатуға) болады, өйткені модельдерді пайдалану кезінде Pilgrim пакетінің құралдық заты қолданылмайды.

Pilgrim имитациялық модельдеуінің пакетінің [3] сипатталған ұсынылымын келтірейік.

9.2.1. Модельдің негізгі нысандары

Модельденетін жүйе тұжырымдамасы негізделетін алты негізгі түсінік бар.

1.Кез-келген имитациялық модель сызбасын модельдің бағытталған бағаны ретінде ұсынуға болады, олардың биіктіктері қарапайым үрдістердің құраушылары болып табылады, ал доғалар модельденетін жүйедегі басқарушы әрекеттерді және өтініштер ағынының бағытын анықтайды. Қарапайым СМО сипаттайтын баған мысалы 9.1 суретте көрсетілген.

2.Транзакт – қандай-да бір қызмет көрсетуге формалды өтініш. Жаппай қызмет көрсету модельдерін талдау кезінде қарастырылатын өтініштерге қарағанда, транзакт динамикалық ерекше қасиеттер мен параметрлерден тұрады. Модель бағаны бойынша транзакттарды миграциялау жолдары бағана түйініндегі модель құраушыларының логикалық қызмет етуімен анықталады.

Транзакт келесідей әрекеттерді орындай алады:

- басқа транзактілер тобын тудыра алады;
- нақты түрдегі басқа транзактілерді жұта алады;
- ресурстарды қамтып, оларды бірнеше уақыт пайдалана алады, одан кейін босата алады;
- қызмет көрсету уақыттарын анықтай алады, өтілген жол туралы ақпаратты жинап, басқа транзактілердің ары-қарайғы жолдары туралы ақпараттарды жинай алады.

Транзактінің негізгі параметрлеріне жатады:

Сурет 9.1 Модель бағанының мысалы

■ транзактінің бірегей сәйкестендіргіші;

■ транзакт тиесілі болатын топ нөмірі (сәйкестендіргіші);

■ транзакт қамти алатын және қандай-да бір уақытта пайдалана алатын әртүрлі ресурстар жиынтығы;

■ транзакт өмірінің уақыты;

■ басымдылық – теріс емес сан; басымдылық жоғары болған сайын, транзакт те басым болады (мысалы, кезекте тұрғанда);

■ қандай-да бір қызмет көрсетуші құрылғыдағы қызмет көрсету параметрлері (ықтимал сипаттамаларды қоса алғанда).

Транзакт мысалдарына жатады: ақша аудару талабы; фирмада жұмыс орындауға өтініш; сатып алушы дүкенде.

3.Желі бағанының түйіндері транзактілер бойынша қызмет көрсету орталығынан тұрады (жаппай қызмет көрсетуде міндетті емес). Транзактілер түйіндерде кідіруі, қызмет етілуі, жаңа транзактілер топтарын тудыруы, басқа транзактілерді жоюы мүмкін. Әрбір түйінде есептеу үрдісі тарапынан тәуелсіз үрдіс туындайды. Есептеу үрдістері параллельді орындалады және бір-бірлерін байланыстырады. Олар бірегей модельді уақытта, бір кеңістікте орындалады, уақыт, кеңістікті және қаржылық динамиканы есепке алады. Түйін мысалдары: өндірістік аумақ (жөндеу); билет сатуға арналған касса; ресурстарды бір кеңістіктен екіншісіне ауыстыратын тасымалдау құралы; ресурстар қоймасы.

4.Оқиға ретінде бір транзактінің түйінінен шығу қарастырылады. Оқиға әрдайым уақыттың белгілі кезеңдерінде пайда болады. Олар кеңістік нүктесімен де байланысуы мүмкін. Модельдегі көршілес оқиғалар арасындағы аралық – бұл әдетте кездейсоқ шамалар. Модельді дайындаушы оқиғаларды қолымен басқара алмайды (мысалы, бағдарламадан). Сондықтан, оқиғаны басқару функциясы арнайы басқару бағдарламасына беріледі – модель құрамына автоматты түрде кіретін үйлестірушіге.

5.Ресурс – модельдеу үрдісіндегі табиғатына тәуелсіз үш ортақ параметрмен сипатталуы мүмкін: қуаттылықпен, қалдықпен және тапшылықпен. Ресурс қуаттылығы – ресурсты бірліктердің максималды саны, оларды әртүрлі мақсатта пайдалануға болады. Ресурс қалдығы - транзактілерді қанағаттандыру үшін пайдалануға болатын бос бірліктер саны. Ресурс тапшылығы – аталған ресурс бойынша кезекте тұрған транзактілердің жиынтық өтінішіндегі ресурс бірлігінің саны.

Әдетте ресурстардың үш негізгі түрін бөліп қарастыруға болады: материалды, ақпараттық және қаржылық.

6.*Кеңістік* - декарттық, географиялық жазықтық (басқаларын да енгізуге болады). Түйіндер, транзактілер және ресурстар кеңістік нүктелеріне байланысып, сонда тасымалдануы мүмкін. Модельді ішкі жүзеге асыру экономикалық үрдістерді ұсынудың бағытталған-объективті тәсілін пайдаланады. Транзактілер, түйіндер, жағдайлар мен ресурстар – имитациялық модельдің негізгі объектілері. Осы объектілердің өзара әрекеттері баған құрылымымен анықталады.

Түрлі модельдеуші жүйелерде баған түйіндерін ұсынудың әртүрлі тәсілдері болады. Бұл осындай жүйелердің ерекшеленетін қасиеттерімен байланысты болып келеді. Мысалы, GPSS жүйесіндегі түйіндер блоктар деп аталады; блоктардың әртүрлі блоктарының саны жүзден асады, бұл модель бағанын қабылдауды күрделендіреді. Piligrim пакетінде түйіндердің 17 түрі болады, олар GPSS барлық блоктарын қамтиды және қосымша құралдарды ұсынады, олар GPSS жүйесінде болмайды:

- үздіксіз үрдістермен жұмыс істеу мүмкіндігі;
- кеңістікті динамиканы модельдеу;
- ақша және материалды құндылықтардан тұратын ресурстармен жұмыс істеу, бухгалтерлік шот есептері, банктік шоттар.

Модель бағанын «оқуға» мүмкіндік беретін түйіндер мәнінің жүйесі болады. Әрбір түйін графикалық мәннен, функционалды атаудан, ерікті бірегей нөмірден және ерікті атаудан тұрады (мысалы, атауы – serv, нөмірі -123, аты –«Мастерская»). Транзактілер жолдары доғалармен анықталады - бағыттаушысы бар тұтас сызықтардан.

9.2.2. Имитациялық модель түйіндерінің түрлері

Имитациялық модель түйіндері модельденетін бағдарламаны жазу кезінде пайдаланушымен белгіленетін параметрлермен сипатталады. Түйін параметрлері тұрақты және ауыспалы мәннен тұрады. Кез-келген түйіннің бірінші параметріне p1 модельдегі символдық атауы, түйіннің логикалық бағыты жатады. Ары қарай Piligrim пакетінің кейбір түйіндері және оның параметрлерінің бағыттары сипатталған.

ag түйіні – транзактілер генераторы. ag түйіні шексіз сыйымдылықтан тұратын транзактілер генераторынан құралады. Ол имитациялық модель транзактілерін құру үшін қолданылады. Транзактілер белгілі-бір уақытта пайдаланушымен бір-бірлеп түрленеді. Генератор тапсырмасы үшін ag (p1,p2,p3,p4,p5,p6,p7,p8). Параметрлер келесідей мәннен тұрады:

p1 – түйіннің символдық атауы (баған ұзындығы 14 символға дейін);

p2 –генератор-түйіннің нөмірі (int);

p3-әрбір түрленетін транзактіге арналған басымдылық, диапазондағы сан -1.....32767 (сан үлкен болған сайын, басымдылық та жоғары болады), егер басымдылық керек болмаса, p3=none.

p4-екі тізбекті түрленген транзактілер арасындағы уақыт аралығын бөлу функциясының түрі болып табылады. Шартты мәндер пайдаланылады: norm-қалыпты бөлу; unif- бірдей бөлу; expo-экспоненциалды бөлу; erln – Эрланг бойынша бөлу; beta-үшбұрышты бөлу; none-анықталған шама болып табылатын аралық.

p5-екі тізбекті түрленген транзактілер арасындағы уақыт аралығының математикалық болжамы (float), p4 =norm, expo, unif. Осы интервал қосылғышының математикалық болжамы (p4 = erln), немесе интервалдың минималды мәні (p4 = beta) немесе осы интервалдың тұрақты шамасы (p4 = none).

p6- бөлу функциясының түріне тәуелді болатын шама (float): орташаквадратты ауытқу (p4 =norm), немесе орташа шамада есептелген максималды ауытқу (p4 = unif), немесе zero (p4 = expo,none), немесе қосылғыш интервалдар саны (p4 = erln, бұл жағдайда p4> 0), немесе екі тізбекті түрленген транзактілер арасындағы уақыт интервалының болжамды мәні (p4 = beta);

p8-түрленетін транзакт берілетін түйін нөмірі.

Генератор параметрлері cheg командасының (пәрменінің) көмегімен модельмен жұмыс істеу кезінде өзгеруі мүмкін.

Serv түйіні – абсолютті басымдылықтары бар көпканалды қызмет көрсетуші аспап. Serv түйіні – модельді уақыт ішінде транзактілердің қандай-да бір қызметін көрсететін қызмет көрсетуші аспап.

Имитациялық модель тарапынан «қызмет көрсету» уақыттың белгіленген аралығында транзактінің кідірісімен түсіндіріледі. Сервер – көпканалды қызмет көрсетуші аспап, ол абсолютті басымдылықтар ережелері бойынша жұмыс істейді немесе оларсыз

істейді, «үзілген» транзактілер үшін стегтен тұрады (қатысты басымдылықтар ержесі queue түріндегі түйінде орындалады – кезек).

Имитациялық модель тарапынан «қызмет көрсету» уақыттың белгіленген аралығында транзактінің кідірісімен түсіндіріледі. Сервер – көпканалды қызмет көрсетуші аспап, ол абсолютті басымдылықтар ережелері бойынша жұмыс істейді немесе оларсыз істейді, «үзілген» транзактілер үшін стегтен тұрады (қатысты басымдылықтар ержесі queue түріндегі түйінде орындалады – кезек).

Сервер (p1,p2,p3,p4,p5,p6,p7,p8) түйінді функциясының көмегімен анықталады. Параметрлер келесідей мәнге ие болады:

p2 – қызмет көрсетуші каналдар саны (int), 1<p2 <32767;

p3-қызмет көрсету тәртібі: abs - басымды транзактіден аз үзінді қызмет көрсетуден тұратын басымдылықты, немесе none – басымдылықсыз. p1,p4..... p8 параметрлерінің бағыты ag түйіні параметрлерінің бағытымен сәйкес келеді. Егер p3= abs белгілейтін болсақ, онда басымдылықсыз транзактілермен үздіксіз жұмыс істеудің екі мүмкіндігі бар: оларға толығырақ қызмет көрсетіледі немесе үзу сипаты қызмет көрсетуді жаңартуды талап етеді. Мысалы, егер компьютерде мәтін теру жүзеге асырылса, онда қорек көзінің аяқ асты өшуі (басымдылықты транзакт ретінде түрленеді) мәтінді теруді басынан бастауға итермелейді (басымдылықсыз транзакт). Транзактінің «қызмет көрсетуі» әдепкі қалпы бойынша қосымша қызмет көрсету режимінде жалғасатын болады.

Дегенмен, serv функциясының алдында иелену операторы орындалатын болса, t-ga=again, онда транзакт қызмет көрсетуді қайтадан бастау белгісін алады. Serv шыққаннан кейін, бұл белгі жоғалады.

Serv түйініндегі транзактілердің қызмет көрсетуінің орташа уақыты және ондағы кідірістердің орташа уақыты – әртүрлі уақыттар. Транзактінің қызмет көрсетуіндегі кідірістер – қызмет көрсетудің таза уақыты, ал түйінде басымдылықсыз транзактінің кідіріс уақытында ол қызмет көрсету уақыты және осы транзактінің үзілетін күйдегі уақыты ретінде енеді (яғни, басымдылықты транзактінің қызмет көрсетуі кезінде).

queue түйіні – қатысты басымдылықтары бар кезек. queue түйіні транзактілер кезегін модельдейді. Бұл кезек екі ереженің біреуі бойынша құрылады: немесе транзактілер түсу тәртібі бойынша реттеледі, немесе, одан басқа, басымдылықты транзактілер кезектің басына жақын орналасады, ал басымдылығы төмен транзактілер –

соңына қарай орналасады. Екінші жағдайда қатысты басымдылықтар ережесі жұмыс істейді. Аталған түйінді сипаттау үшін queue функциясы пайдаланылады (p1,p 2,p 3). p1 және p 3 параметрлері түйіннің символдың атауларын және бағандағы келесі түйін нөмірін анықтайды, ал p 2 кезекті ұйымдастыру түрін белгілейді. p 6 кезекті ұйымдастыру түрін белгілейді (егер p2=prty болса, онда басымдылықты кезек қолданылады, ал егер p2=none болса, онда басымдылықсыз қолданылады).

Term түйіні – транзактілер терминаторы. Term түйіні-транзактілер терминаторы, оның бағыты транзакті кіретін модельден шығарады және оның өмір сүру уақытын бекітеді, осы транзактінің генератордан шығуынан бастайды. Аталған түйінді сипаттау үшін term функциясы (p1) пайдаланылады, p1 түйіннің символдық атауын анықтайды.

Key түйіні-транзакт жолындағы клапан. Key түйіні – «шлагбаум» қағидасы бойынша жұмыс істейтін клапан немесе кілт. Клапан жабық болғанда, транзакт оған кіре алмайды. Егер клапан ашық болса, онда транзакт кідіріссіз келесі түйінге енеді. Мұндай түйіннің жабық күйде болуының орташа уақыты автоматты түрде есептеледі. Мұндай клапанды немесе кілтті басқару hold және rels пәрмендері арқылы орындалады.

Аталған түйін key функциясының көмегімен (p1,p2) сипатталады, оның параметрлері түйіннің символдық атауын және бағандағы түйін нөмірін анықтайды.

Creat түйіні – транзактілердің басқарылатын генераторы. Creat түйіні транзактілердің жаңа тобын құру үшін қолданылады. Барлық транзактілер қандай-да бір топқа жатады. Қарапайым генератордан шығатын транзактілер (ag) 0 нөміріндегі топқа жатады. Басқарылатын генераторды сипаттау үшін түйінді функциялар қолданылады: creat (p1,p2,p3,p4,p5,p6). Аталған түйіннің қызмет ету логикасы келесідей:

- түйін арқылы туындатушы транзакт өтеді, ол f1 тобына жатады және келесі p6 түйініне түседі;
- бір уақытта түйінде p3 жаңа транзактілердің түрлері f1= p2 нөмірімен түрленеді, олар p5 түйініне бағытталатын болады. f1 және f2 топтарының нөмірлері сәйкес келуі мүмкін.

Қалған параметрлердің бағыттарын келтірейік. p1 параметрі түйіннің символдық атауын анықтайды. p4 параметрі келесідей мәндерден тұрады: тудырушы транзактінің параметрлерін тираждау

үшін әрбір тудырушыға, немесе none – әрбір тудырушы транзактіге нөлдік мәндер параметрлерін беру үшін қолданылатын мән.

Delet түйіні – транзактілердің басқарылатын терминаторы. Delet түйіні топтардың кейбір диапазондарына жататын транзактілер тобын жою үшін қолданылады. Аталған түйін delete функциясымен сипатталады (p1,p2,p3,p4,p5,p6).

Бұлт түйіндердің қызмет көрсету логикасы келесідей: түйінге p4 тобының жоюшы транзакті кіреді және сонда p5 түскенге дейін болады: p2<Number<p3, оларды бірдей жоя алады. Аталған түйінде жұтылатын транзактілердің уақыты бекітіледі. Транзактілердің талап етілетін саны түскеннен кейін транзакт p6 түйініне ауысады.

Send және direct түйіндері – бухгалтерлік шоттары бар операция. Piligrim пакетінің негізгі объектілері (түйіндер, транзактілер, жағдайлар) кәсіпорынның (фирманың) қаржылық динамикасын сипаттау үшін ыңғайлы болып келеді. Бухгалтерлік есеп шотын (қосалқы шот) сипаттау үшін send түйіні қолданылады; түйін нөмірі і деп санайық. і түйініне кіретін транзакт - і шотынан басқа шотқа, белгілі бір сомада жазба жүргізуге өтініш. Жазбаны жүзеге асыру үшін і шотында (яғни і түйінінде) талап етілетін сома болу керек. Мұндай сома болмаған жағдайда, транзакт і шотына жеткілікті қаржының түсуін күту режиміне өтеді. Басқа сөзбен айтқанда, і нөмірі бар түйін бухгалтерлік жазбаны қалыптастырады, бұл транзактілердің арнайы кезегі болып табылады.

Келесі send түйіні міндетті түрде direct түйінінен тұруы керек, ол «қаржылық директор» рөлін атқарады. Direct түйіні клапан болып табылады, сол арқылы жазба орындауға болатын транзактілер ғана өте алады. Direct бір түйіні модельдегі send барлық түйіндеріне қызмет көрсете алады.

Send түйінін сипаттау үшін send функциясын пайдалану қажет (p1,p2,p3,p4,p5), олардың параметрлері келесі мәндерден тұрады: p1-түйіннің символдық атауы; p2-аталған сома аударылатын түйін-шот; p3-аталған сома өлшемі (double түрі). Қаржылық құралдарды өлшеу бірліктері – кез-келген валютада болады (рубльдер, долларлар және т.б). Нүктеден кейін міндетті түрде бір немесе екі санды белгілеу қажет - өлшемдердің пайдаланылатын бірліктерінің үлестері. Мысалы: 1 000 000.00 (бір млн руб.00 тиын):

p4- басымдылықтармен жұмыс істеу мүмкіндігін анықтаушы параметр (prty мәні) немесе осы жазбалардағы қажеттіліктердің пайда болу кезеңдері бойынша жазбаларды қатаң реттеу (none мәні).

Егер prty мәні белгіленсе, жазба сомасы аз болған сайын, сұраныс та басым болады;

p5-direct түріндегі түйін нөмірі, ол қаржылық менеджментті жүзеге асырады және қажет болған жағдайда жазбалар жүргізеді.

Send әрбір түйінінде saldo ішкі атрибуты болады, ол і шотындағы қаржы қалдықтарын көрсетеді. Бухгалтерлік есеп шотындағы қаржының жетіспеушілігі defic атрибутында болады. Егер і түйініндегі saldo атрибуты нөлдік мәннен тұрса, бұл жазба тудыратын түйінде транзактілер болады (бір немесе бірнеше), осы транзактілермен талап етілетін соманың жиынтық тапшылығы defic атрибутында автоматты түрде көрінеді. Send түйінін басқару үшін assign пәрмені қолданылады. Оның көмегімен бухгалтерлік шоттағы қаржы қалдығын өлшеуге болады. Send түріндегі кезектерге қызмет көрсету бір немесе бірнеше «қаржылық директор» түйіндерін пайдалану арқылы орындалады. Мұндай түйінді сипаттау үшін direct (p1,p2) функциясы қолданылады. Бұл жердегі p1 параметрі түйіннің символдық атауын анықтайды, ал p2 параметрі транзактінің қабылдаушы-түйінінің нөмірін анықтайды, ол бухгалтерлік жазбаны орындайды.

Attach және manage түйіндері – ресурс қоймаларымен жұмыс істеу. Piligrim пакетінде имитациялық модельдерді құру кезінде модельденетін үрдістердің өту уақытын ғана емес, сонымен бірге, осы үрдістермен пайдаланылатын, жұмысқа қажетті ресурстарды қолдану да қарастырылады. Ресурс ретінше қоршаған әлемнің кез-келген объектісі есептеледі (шикізат, құрылғы, персонал). Модельдегі ресурс көлемі бүтін сандарда өлшенеді. Үрдіспен пайдаланылатын ресурстардың барлық түрлері арнайы attach түйін-қоймаларында сақталады. Attach түйінін транзактімен қараған кезде ондағы сақталатын ресурстың бірлік саны алынады. Егер қоймадағы ресурс саны аз болса (транзакті талабынан), онда ресурс көлемі көп болмағанға дейін ол түйінде қала береді, яғни, attach түйіні ресурс алу бойынша өтініштер кезегінде қалады. Өтініштер кезегі қарапайым болуы мүмкін, және өтініш басымдылығын есепке алуы да мүмкін. Басымдылық жоғары болған сайын, талап етілетін ресурс көлемі де аз болады. Модельдегі attach түйінінен кейін міндетті түрде manage түйіні тұруы керек, ол қойманы басқарады. Бұл түйін әрбір транзакт үшін attach түйініне келетін ресурстың талап етілетін санын тексереді. Егер ресурс жеткілікті болса, онда транзакт manage түйіні арқылы ары қарай өтеді. Әйтпесе транзакт attach түйінінде қалады.

Attach түйіні attach функциясымен орындалады (p1,p2,p3,p4). Оның параметрлерінің мәні келесідей:

p1-түйіннің символдық атауы;

p2-ресурс бірлігінің талап етілетін саны;

p3-prty немесе none – өтініштер басымдылығы есебінің жалаушасы. Send түйіні үшін де, өтініштер көлемі аз болған сайын, оның басымдылығы да жоғары болады;

p4-manage түріндегі түйін нөмірі, аталған қойманы басқарады.

Ресурс қалдығы және тапшылық rsal және rdef атрибуттарында бекітіледі. Тапшылық барлық кезекте тұрған транзактілердің attach түйініндегі өтініштердің жиынтық көлеміне тең болады (ресурстың жетіспеушілігіне байланысты қызмет көрсетілмейтіндер).

Транзактілерге қызмет көрсетумен manage түріндегі түйін айналысады – ол қойманы басқарады. Аталған түйінді сипаттаушы функция manage түрінен тұрады (p1,p2). Аталған функцияның p1 параметрі түйіннің символдық атауын анықтайды, ал p2 параметрі транзакт қабылдаушысының нөмірін анықтайды.

Қоймада алынған транзакт ресурс модель бойынша «тасымалданады» және оны қоймаға қайтара алады. Транзакт ресурсты detach пәрменінің көмегімен модельдің кез-келген түйінінен қайтаруға болады. Ресурсты қайтару міндетті емес. Қоймадағы ресурс көлемін өзгерту үшін supply пәрмені қолданылады.

9.2.3. Тораптарды басқару пәрмендері

Тораптарды басқару командалары қажет болған жағдайда кез келген түйінде пайдаланылуы мүмкін. Әрбір үлгі торабы кез келген басқа түйінді және өзі басқара алады. Команда әрбір транзактті көрсеткен түйінді енгізген сайын іске қосылады.

Блокталған delet торапты - freed пәрменімен босатыңыз.Freed (p1) командасы delet түйінінің блокталуымен күресуге арналған. Ол келесі логика бойынша жұмыс істейді.

Мысалы, p1 санымен delet торабында транзакт «тоқталды» (яғни, жойылатын транзакциялардың қажетті саны ұзақ уақытқа қабылданбайды) деп есептейік.Freed функциясы арқылы өтетін ағымдағы транзакт дереу қосалқы транзакті жасайды және оны p1 торабына жібереді.Бұл қосалқы транзакт кептеліп, «сыртқа шығады», deletқалыпты жағдайға алып келеді, ал өзі өледі. Қосалқы мәмілеге

келу фактісі delet-ге тіркеледі және статистикалық нәтижелерде көрінеді, бірақ оның өмір сүру уақыты нөлге тең. Кептелген транзакция келесі delet түйініне жіберіледі.

Ag транзакт генераторын көшіру - cheg пәрмені. Генераторды P2 нөмірімен жаңа параметр мәндеріне орнату үшін cheg(p2, p3, P4, P5) командасы қолданылады.Модельдік уақыт мағынасында осындай қайта жүктеу «бірден» болады. Бұл пәрменнің барлық параметрлері генератордың ag параметрлері (генератордың символдық атауы командада көрсетілмеген) сияқты бірдей мақсатқа ие.

Клапанды басқару key - hold және rels командалары. Hold (p1) және rels (p1) командалары кез келген басқа түйіндерден p1 санымен клапанды басқару үшін қолданылады. Holdорындаудан кейін, клапан p1 күйі «жабық», ал кейін rels - «ашық» жағдайда алады.

Бухгалтерлік шотты басқару - команда assign.Assign (p1, p2, p3) командасы p1 (double) ақша p1 сомасын орналастырады.Егер p2 операциялық кодының мәні add мәніне тең болса, онда шоттың балансы p3 (қосу режимі) сомасына көбейтіледі; егер p2 = none болса, онда шоттың балансы тең p3 (ауыстыру режимі) болады.

Ресурс қоймасын басқару - supply және detach пәрмендері. Supply (p1, p2, p3) командасы p1 қоймасындағы ресурстық бірліктердің p3 (long) мәнін орнатады.Егер p2 операциялық кодының мәні add-ге тең болса, қоймадағы ресурстың қоры p3 (қосу режимі) сомасына көбейтіледі; егер p2 = none болса, онда ресурстың қоры p3 (ауыстыру режимі) тең болады.

Detach (p1, p2) командасы p1 нөміріне p1 нөмірімен қоймадағы ресурс бірліктерінің санын қайтарады. Моделді дұрыс пайдалану үшін ресурстың көрсетілген көлемі бұрын қойма пәрменінде көрсетілген қайтару мәмілесі бойынша қабылданады (транзакция қайтарылады, ол detach пәрменін іске қосады).

9.2.4. Транзакт параметрлері және тораптардың жай-күй параметрлері

Транзакт параметрлері бар, олардың кейбіреуі пайдаланушыға қол жетімді. Транзакт параметріне кіру тек модель түйініне енген сәтте мүмкін болады және торапқа көшірудің атауы болып табылатын жүйе айнымалы t арқылы жасалады.

Мәміленің негізгі параметрлері болып табылады: t → iu0, t → iul, t → iu2, t → iu3 - пайдаланушының еркін бүтін параметрлері (мәміленің жай-күйі туралы ақпаратты сақтау үшін пайдаланылады);

t → ru0, t → rul, t → ru2, t → ru3 - өзгермелі нүктесі бар ауыспалы

нысаны бар ерікті пайдаланушы параметрлері;

t → ga - қайтадан мәміле жасаудың белгісі, ол serv функциясын қарағаn кезде көрсеткендей,меншіктеу оған ғаламдық тұрақты операциясымен again тағайындалуымен анықталады;

t → pr - мәміленің басымдығы;

t → ft - бұл транзакт тиесілі отбасының саны.

Модельдің күтпеген мінез-құлқынан қорықпай, тек қана өзгеріске ғана ие бола алмайды, тек қана, t → iu1, t → iu2, t → iu3, t → ru0, t → ru1, t → ru2, t → ru3, t → ga,және , кейде t → pr және t → ft. Басқа параметрлерін өзгерту тек Pilgrim-геғана сеніп тапсырылуы тиіс.

Pilgrim пакетіндегі үлгі түйіндері, сондай-ақ параметрлерге ие. Кейбір түйін параметрлері талдау үшін пайдаланушыға қол жетімді (бірақ оларды өзгерту үшін емес).Параметрлер жүйелік addr алқабында бар. Параметрге

```
addr <n> -><параметр>
```

мекен-жайының өрнегі арқылы кіруге болады, мұндағы <n> - үлгідегі түйін нөмірі.

Кесте 9.1. Тораптар параметрлері

Аты	Параметр мақсаты
nc	Тораптағы арналардың саны
na	Модельдік уақытында түйін арқылы өткізілген транзакциялар саны
tn	Модельдік уақытында торапта болатын мәмілілер саны
ts	Қазіргі уақытта есептелген орташа жұмыс уақыты
°p	Кеу типіндегі тораптың күйі. Бұл клапан ашық болғанда true мәнін қабылдайтын бүтін айнымалылар, немесе егер ол жабық болса, false
se	Delet типті түйіннің күйінің ерекшелігі. Бұл торапта деструктивті операция болмаған кезде nil мәнін қабылдайтын бүтін айнымалылар
saldo,	Saldo үлгідегі торабына байланысты есеп
defic	шотындағы қаражат тапшылығы және қалдығы
rsal,	Қоймадағы ресурстық тапшылық және қалдық
rdef	rdef торабы үлгідегіattach-қабайланысты

Түйіндер параметрлерінің аттары мен тапсырмалары Кестеде келтірілген. 9.1Кез келген түйіннің параметрлерін талдау үшін оның <n> нөмірін пайдалануға болады.Мысалы, егер транзакт delet типті түйіннің жағдайын 5 санымен талдауға тура келсе және жоюшы транзакция болмаса, операторды орындаңыз, ол келесі өрнектен өтуі керек:

```
if (addr [5] -> se == nil) < оператор>;
```

Бұл жағдайда транакттың өзінде бұл кезде еркін түйінде болуы мүмкін.

9.2.5. Кестелік модельдеу нәтижелері

Модельдеу нәтижелері туралы есеп бір немесе бірнеше беттен тұратын кесте ретінде көрсетіледі.Кесте модельдік бағдарламада модельдің авторы көрсететін жеке файлда жазылады. Осы кестенің өрістерін жіктеу 9.2.кестеде келтірілген.

9.2-кесте. Модельдеу нәтижелерімен кесте өрістерін тағайындау

Баған	Толық Жазылуы
№	Торап нөмірі
Торап түрі	Торап түрі (ag, serv, key, queue, term, creat, delet, proc, dyna, send, direct, attach, manage)
Нүкте	Модельдеу аяқталған кезде creat, delet немесе процтүрінің торабы орналасқан кеңістіктегі соңғы нүктенің саны
Жүктеу,% жол, км	Serv, key немесе proc типті түйіндерді пайыздық көрсеткіште пайдалану коэффициенті есептеледі. Key түйіндері транзакциялар арқылы өтетін кезде пайдаланылады. Attach торабы үшін бұл ресурсты пайдалану коэффициенті
M [t] орташа	Түйіннің түріне байланысты түйінде немесе басқа уақыт аралығындағы транзакцияның кешігу уақытының орташа мәні:1) d - түйінге жұмсалған орташа уақыт (абсолютті басымдықты ережені пайдалану кезінде артық емес мәмілелерге қызмет көрсету уақытынан ұзағырақ болуы мүмкін);

9-2. кесте жалғасы

Баған	Толық Жазылуы
M [t] орташа	2) queue, send және attach - кезектегі орташа кідіріс уақыты; 3) ag - екі генерацияланғантранзакт арасындағы орташа уақыты; 4) term немесе delet - жойылған мәмілелердің орташа мерзімі; 5) key - жабық жағдайда орташа уақытты өткізу уақыты
C2 [t] вариациясы	Кешігу уақытының дисперсиясының кешігу уақытының орташа мәнінің квадратына қатынасы (вариация коэффициенті, квадратқа бой көтергені)
Кіріс есептегіші	Торап арқылы жасалған транзактылардың саны немесе жасалынған транзактылардың саны (ag немесе creat үшін) немесе жойылған (term немесе delet үшін). Key үшін берілген hold пәрмендерінің саны
Арналардың саны	Тораптағы арналардың саны
Транз – актілердің қалдығы	Модельдеу аяқталған кезде түйінде сақталған транзактылардың саны
Осы сәтте торап жағдайы	Модельді іске қосу соңында түйіннің жағдайы: келесі транзактіні енгізу үшін түйін «ашық» немесе «жабық» болуы мүмкін (тораптар serv және key).Сервер ашық, онда кемінде бір бос арна бар. Send және attach түйіндері үшін кестенің қалған бөлігі және қор тапшылығы (тиісінше «S» және «D» әріптері)

Сонымен осы стандартты Pilgrim құралдарымен қатар пайдаланушы әрқашан өздерінің құжаттарын басып шығаруға немесе олардың нәтижелерін модельдеу үрдісінде де, соңында да шығаруға мүмкіндік алады.Бұл C ++ тілінде жазылған және Pilgrim бағдарламасының сәйкес нүктелерінде орналасқан пайдаланушы анықтайтын операторлар немесе функциялар көмегімен жасалады.

9.2.6. Модельдерді сипаттау тілі

Pilgrim бағдарламасының жалпы құрылымы. Pilgrim пакетіндегі имитациялық моделін сипаттайтын бағдарлама құрылымы келесідей:

```
#include <pilgrim.h>
[константт пайдаланушының сипаттамасы]
[ауыспалы пайдаланушының сипаттамасы] forward {
 modbeg(...);
 ag(...);
 [бастапқы пәрмендер assign және supply]
 [бастапқы пайдаланушының функциясы]
 network(...)
 {
  < блок торабының сипаттауы>
  fault(...);
  }
  [Пайдаланушының аяқтау функциялары]
  modend(...);
  return 0;
}
```

Модельдің бірінші жолы - «Pilgrim» жүйелік файлын қосу командалары.Содан кейін пайдаланушы C ++ тілінің ережелеріне сәйкес қажетті тұрақты және айнымалыларды сипаттай алады.

Modbeg функциясынан кейін барлық ag генераторларын сипаттау жүреді.Үлгінің басқа түйіндері түйіннің сипаттамасы блоктар көмегімен сипатталады. Нөмірлеуге қарамастан кез келген тәртіпте түйіндерді сипаттауға болады.

Үлгіні инициализациялау modbeg функциясы. Модельдеу бағдарламаларын іске қосу операторы modbeg (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9). Бұл функцияның дәлелдері келесі мағынаға ие:

p_1 - бұл модельдің символикалық атауы;

p_2 - бұл үлгі түйіннің максималды саны (int), $2 \leq P_2 \leq 1{,}024$;

P_3 - модельдеуді орындау қажет уақыт кезеңі (float);

p_4 - жалған кездейсоқ шамаларды (long) сенсорларын теңшеу үшін пайдаланылатын ерікті бүтін сан.P4 ретінде компьютердің таймерінің мәнін пайдалануға болады, оның қол жетімділігі келесі пішінге ие: (long time(NULL).Бұл жағдайда түрлі уақыттарда

жүргізілген модельдеу нәтижелері әр түрлі болады;

p_5 - кеңістіктік модельдеу режимінің белгісі: earth - Жер беті, plane - декарттық жазықтық, cosmos - еркін кеңістік.Егер кеңістіктік модельдеу пайдаланылмаса, none пайдалануға болады;

p_6 - кезектердің біреуінің (queue түрінің торабының) саны, ол осы кезектегі кідірістердің динамикасын нәтижелерді графикалық көрсету арқылы талдау үшін уақытында қадағалауға тиіс.Мысалы, 3 санын көрсетсеңіз және модельде бұл нөмірмен кезек бар болса, онда осы кезектің динамикалық кескінін тікелей имитациялауға болады;

p_7 - бұл кеңістікте де, нәтижелерді графикалық көрсету кезінде де бақылануға тиіс процестердің біреуінің (proc типті торабы) саны.Егер кеңістіктік модельдеудің графикалық интерпретациясы қажет болмаса, онда none көрсетіледі;

p_8 - терминатордың нөмірі (типті term), оның кіріспесінде имитациялау барысында транзакциялар ағынының қарқындылығын сақтау қажет. Егер мұндай қажеттілік жоқ болса, онда none көрсетіледі;

p_9 дәлдігі (int саны 1-ден 6-ға дейін немесе none), яғни уақыт аралығына сәйкес келетін нәтижелердің шығуындағы ондық үтірден кейінгі сандар саны. Егер p9 = none болса, нәтижелер бүтін мәндер бойынша дөңгелектенеді.

Үлгі үйлестірушісін іске қосу – network функциясы.Функция network, (p_1, p_2) модельдер бөлімшелеріндегі мәмілелерді диспетчерлендіреді, оқиғаларды біртұтас уақыт режимінде жоспарлайды және модельдің үздіксіз компоненттерін белсендіреді.Бағдарламалау тілінің синтаксисінің тұрғысынан, бұл функция құрылымдық мәлімдеме болып табылады, одан кейін модельдік графиктің сипаттамасымен блокты ашатын бұйра сызық {.Бұл функция шақыру modbeg функциясы және барлық ag функциялары орындалғаннан кейін ғана жасалады. P1 және p2 аргументтері интегралдауды жүзеге асыратын кіші бағдарламалардың атаулары (мекен-жайлары), айырмашылық теңдеуін шешу, формулалар бойынша есептеу және т.б.Ең қарапайым модельдерде бұл функциялар қажет емес, ал dummy сөзі p1 және p2 параметрлері ретінде пайдаланылады. Бүкіл модельдеу кешенінің орындалуы бірінші қатеге дейін немесе modbeg функциясында p3 аргументінде көрсетілген уақыт аяқталғанға дейін жоспарланған.

Қате өңдегіш - fault (err) функциясы.Бағанның сипаттамасының ең соңында, егер пайдаланушының назарынсыздығына байланысты транзакцияны жоқ түйінге жылжыту үшін әрекет жасалса, берілген fault (err) функциясын жазу керек.Мұндай әрекеттен кейін модельдеу тоқтайды және аяқталады err кодымен, мұндағы err - әдетте 101-ден 32 767-ге дейінгі диапазонда еркін анықталатын нөмір болып табылады. Бұл жағдайда пайдаланушыға осы қате орын алған түйін және келесі түйіннің дұрыс нөмірі деген хабарлама беріледі.

Үлгінің соңы - modend функциясы. Modend (p_1, p_2, p_3, p_4) функциясы ішкі жай-күйі noerr бұзылғаннан кейін орындалуы керек, ол модельде жалпы қателігі бар немесе modbeg мәлімдемесінде көрсетілген модельдеу уақыты аяқталғанын анықтайды.Ол компьютердің жадынан моделдеу үрдісінде жасалған басқару құрылымдарын жояды. Бұған қоса, бұл функция монитордың экранында графикалық нәтижелерді көруге және соңғы нәтижелерді есеп файлына шығаруға мүмкіндік береді. Параметр мәндері:

p_1 - модельдеу нәтижелерінің кестесі бар файлдың атауы;

p_2 - есептің бірінші бетінің нөмірі;

p_3 - әр беттегі жолдар саны;

p_4 - екі мағынасы бар: кез-келген бет, егер сіз бет файлына аударманың белгісін есеп файлына қою керек болса немесе none.

Пайдаланушының бастапқы және аяқтау функциялары. Modbeg және network функцияларының арасында модельдің таймерінде нөлдік мән болғанда, сондай-ақ кестенің түйіндерінің сипаттамасын сипаттағаннан кейін, жабылатын бұйра сызығымен аяқталған } - modend функциясынан бұрын, үлгі таймерінде соңғы мән бар болғанда - C ++ бағдарламасындағы кез-келген бағдарлама немесе C ++-та жазылған кез-келген функцияға сілтеме жасаңыз, ол негізгі модель жұмыс істегенге дейін немесе кейін жұмыс істейтін болады.

Торап сипаттамасының блогы. Торап сипаттамасы блогының құрылымы келесідей:

```
top(<n>): [<операторлар пайдаланушы>]
           <торап түрі>(<торап параметрлері>);
          [clcode < пайдаланушының операторы>]
           place;
```

Top (<n>) бастап, трансакт барлық операторларға кешіктірусіз (модель уақытының мағынасында) өтіп, келесі түйінге өту шарттары пайда болғанша place операторына енеді.Кейбір түйіндерде бір уақытта бірнеше транзакт болуы мүмкін. Олардың барлығы бірдей place мәлімдемесінде. Place-да транзактының кешігуі - бұл торапта қалу уақыты болып келеді.Түйін түрі (түйін функциясы) алдында тұрған операторлар түйінге кірер алдында транзактының орындау керек әрекеттері болып табылады. Clcode белгісін бақылайтын операторлар кіргеннен кейін орындалатын әрекеттер болып табылады. Барлық айнымалы мәндер - түйін параметрлері транзакт түйінді енгізбес бұрын анықталуы керек.

Торап түріне қарайтын операторлар транзакция құлыпталған түйінге (бос сервис немесе жеке кілт) кіруге тырысқан кезде бірнеше рет орындалуы мүмкін.Модель жұмысының барысында желілік үйлестіруші мезгіл-мезгіл«Талқылаулар» босатылған-істемейтінін тексеріп, түйіндерді оқшаулап отырады."Сауалнама" кезінде түйіннің түрін орындайтын операторлар автоматты түрде орындалады.Осылайша, осы жерде осы операторларға жаза алмайды, олардың қайтадан орындалуы модель жұмысында қате әкеледі. Clcode кейінгі операторлар тек бір рет орындалады.

Торап сипаттамасы блогында C ++ тілінде және Pilgrim командаларының кез келген жинақтарын орналастыруға болады, сондай-ақ транкзакт параметрлері мен түйін күйінің параметрлерін, бағдарлама үзінділерін, C ++ тілінде жазылған функцияларын және жүйелік қоңырауларға қол жеткізуге болады. Алайда, бұл белгілі бір ережелерге сәйкес жасалуы тиіс.Біріншіден, C ++ тілінің goto үлгідегі операциясын тек бір түйінде (top белгісі функциясы мен тиісті place операторы арасында) ғана емес, сонымен қатар басқа түйінді өту үшін қолдануға болмайды. Модельдеуді аяқтауға мәжбүрлеу үшін goto орнына тек операцияны пайдалануға болады

```
error = коды,
```

онда «код» - «коды»> 100 жағдайын қанағаттандыратын пайдаланушының кез келген қате коды.

Осындай операциядан кейін бақылау координатордан таңдалады және операторға баған сипаттамалық келесі блок үшін (мысалы, функция modend) сәйкес беріледі.

Екіншіден, бірнеше орындалу мүмкіндігінің себебінен, xd = f (x) немесе x = x + 1 типті реквизиттерді top жапсырмасы мен түйін түрі

және т.б. арасында пайдалануға тыйым салынады және маңызды компьютерлік ЭВМ процессор уақытын талап ететін ресурстық қарқынды есептеулерді жазу қажет емес .Тек тапсырма операциясын және, қажет болса, if немесе switch операциясын қолданған жөн. Мысалы:

```
top(3):if (a>0) b =4;
else b = 5;
    key("Кілт",b); place;
```

Ресурстық қарқынды есептеулерді орындау қажеттілігі туындаса, кейбір айнымалылармен (мысалы, x = f (x) немесе x = x + 1) регурсияларды қолданыңыз, жүйелік қоңыраулармен жұмыс істеңіз, ресурсқа негізделген есептеу функцияларына қолжеткізіңіз, бұл түйін түрін анықтау функциясымен тиісті place құрал арқылы арнайы clcodeоператоры.

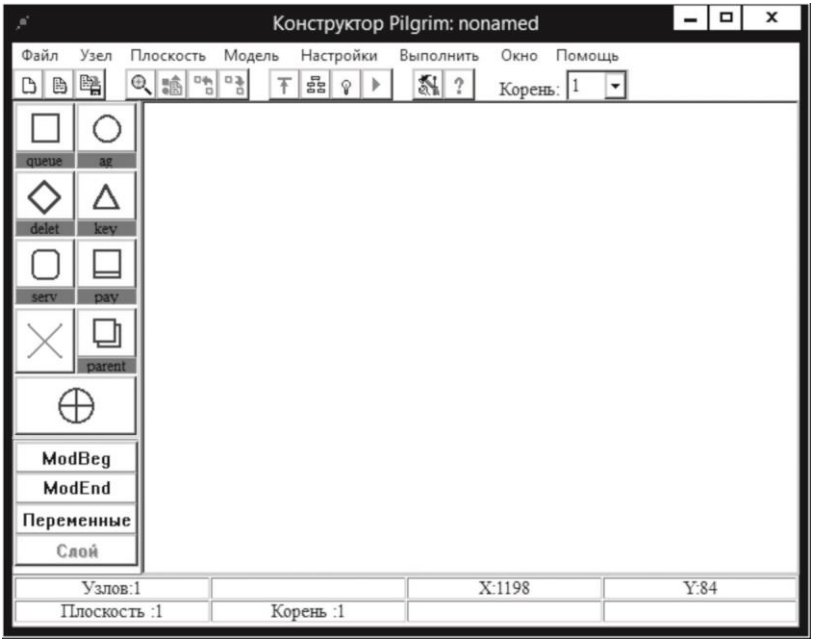
Түрін анықтаған операторлардан кейін сіз dd пішімінде жазылған C ++ тілінің кез келген мәлімдемелерінің блогын жаза аласыз

```
Clcode
{
<пайдаланушы операторлар блогы>
}
немесе жекеclcodeмәлімдемесі
```

9.3. PILGRIM ИМИТАЦИЯЛЫҚ МОДЕЛЬДЕУ ПАКЕТІ

Pilgrim модельдеу жүйесінде модель жасаған кезде графикалық дизайнер Gem көмегімен модельдеу бағдарламасын жасау ыңғайлы, ол сізге үлгі диаграмма жасауға, жалпы модельдеу параметрлерін және графикалық түйін параметрлерін орнатуға және C ++ үлгі кодын жасақтауға мүмкіндік береді.Болашақта бұл код Microsoft Visual Studio жүйесінде жасалған жобада пайдаланылады. Графикалық дизайнер Gem көмегімен модельдеудің негізгі принциптерін қарастырайық.

Графикалық дизайнер басталғанда негізгі бағдарлама терезесі пайда болады (9.2-сурет). Бұл терезеде басты мәзір мен екі құралдар тақтасы бар, олардың біреуі бар



Сурет. 9.2. Графикалық конструктордың негізгі терезесі - бағандардың моделітұратын Gem.

Бұл жағдайда бір түйме түйіндердің бірнеше түріне сәйкес болуы мүмкін (үлгі параметрлерінде арнайы түйін түрі таңдалады). Мысалы, queueтүймесі queue, sent және attachтүйіндерін анықтауға мүмкіндік береді, ag және term түйіндерін анықтау үшін ag түймесі, delete және create түйіндерін анықтау үшін delete түймешігі, key, manager және directтүйіндерін анықтау үшін key түймешігі, pay түймешігі pay rent және downtүріндегі түйіндер.

Жаңа үлгі жасау үшін Файл /Құру Жаңа мәзір пәрменін орындаңыз немесе құралдар тақтасындағы Жаңа үлгі түймешігін басыңыз. Осыдан кейін сіз модель бағандарын жасай алатын жаңа ұшақ жасай аласыз.Үлгі торабын жазықтыққа орналастыру үшін, оны құралдар тақтасынан ұшаққа дейін апару керек. Осыдан кейін, блоктың мәтінмәндік мәзіріндегі түйменің Параметрлер командасын (немесе блокты екі рет басу арқылы) пайдаланып, сіз торап параметрлерін теңшеу үшін терезені аша аласыз. Түйіндер арасындағы сілтемелерді көрсету үшін, шеңберде көк косы бар түймешікті пайдаланыңыз. Бұл түймені сілтеме шыққан түйінге

сүйреп апару керек, содан кейін осы сілтемені қамтитын түйінді басыңыз.Бағанның элементтерін жою үшін қызыл кресті бар батырманы пайдаланыңыз. Оны жойғыңыз келетін нысанға сүйреп апару керек.

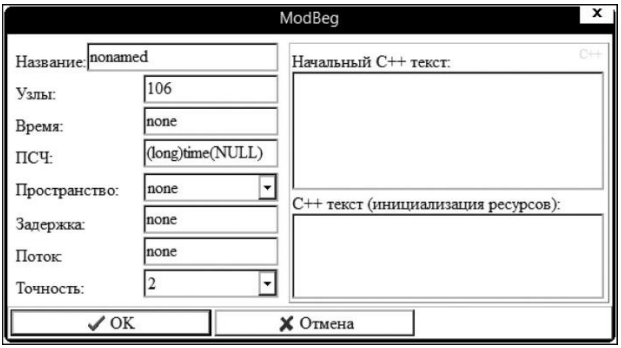
Жалпы модельдеу параметрлерін орнату үшін құралдар тақтасындағы ModBeg батырмасын пайдаланыңыз, басқан кезде параметрлері бар терезе ашылады (9.3 суретті қараңыз).Мұнда сіз модельдің атын (атау өрісі), модель уақыт бірліктеріндегі уақытты модельдеу уақытын, queue түрінің түйін нөмірі, имитациялық (Кешігу өрісі), уақыт түйінінің нөмірі, көрсетілетін кіріс ағынының динамикасы көрсетілетін кешіктірілу динамикасы көрсетіледі. Модельдеу кезінде (Ағын өрісі), алынған нәтижелерге арналған дәл сандардың саны (дәлдік өрісі) көрсетіледі және модельдік операцияның бастапқы кезеңінде (бастапқы C ++ мәтін өрісі) орындалатын C ++ пәрмендері көрсетіледі.

ModEnd батырмасын қолданып, модельдеу нәтижелері көрсетілетін файлдың атын көрсетуге және шығыс параметрлерін көрсетуге болатын терезе шақырылады.

Айнымалылар түймесінің көмегімен сіз модельдеуде пайдаланылатын өзгермелі айнымалыларды анықтай аласыз және олардың түрін және бастапқы мәнін белгілей аласыз.

Gem графикалық дизайнерде жасалған графикалық модельдің мысалы 9.4-суретте көрсетілген.

Модельдік бағанды жасағаннан кейін оның дұрыстығын тексеріп, *Орындау/дұрыстығын* графикасының дұрыстығын тексеріп, C ++ файлын жасаңыз.



9.3-сурет. Үлгінің жалпы параметрлерін көрсету терезес тапсырмалары

Сурет 9.4. Gem графикалық құраушысында құрылған СМО моделі бағанының мысалы

С ++ файлын іске *қосу / жасау*. Бұл С ++ файлын Visual С ++ жобасында пайдалануға болады.Жасалынған үлгі одан әрі пайдалану үшін *Файл / Сақтау* pgf-файлы ретінде сақталады.

9.4. ИМИТАЦИЯЛЫҚ МОДЕЛЬ ДАЙЫНДАУ МЫСАЛЫ

СМО моделін дайындау үрдісін қарастырайық (екі нотариусы бар нотариалды кеңсе), олардың сипаттамалары 7.3 бөлімде аналитикалық есептелген. Ыңғайлылық үшін тағы да аталған жүйені сипаттайық.

Нотариалды кеңсе екі арналы СМО тұрады делік. Кезекті күту бөлмесіндегі орындар саны шектеулі және үшке тең. Егер күту бөлмесіндегі барлық орындар бос болмаса, онда қайта келген клиент кезекке тұрмайды. Кеңес алуға келген клиенттер ағыны қарапайым қарқындылықпен сағатына 12 клиентті құрайды. Қызмет көрсету уақыты экспоненциалды заң бойынша бөлінген және t=7 минутты құрайды.

Алдымен тапсырмалар талдауын орындайық және модель бағанасын құрайық. Осыдан кейін есептік талдауды және тәжірибені жүргізейік.

Аталған тапсырмада тек бір түрдегі өтініштер бар –клиенттер. Бұл өтініштер түйін көмегімен қалыптасады – транзактілер, генераторлар көмегімен. Клиенттер ағыны қарапайым болғандықтан, клиенттердің келуі арасындағы уақыт интервалын бөлу заңы экспоненциалды болып келеді және оның орташа мәні: 60/ λмин.

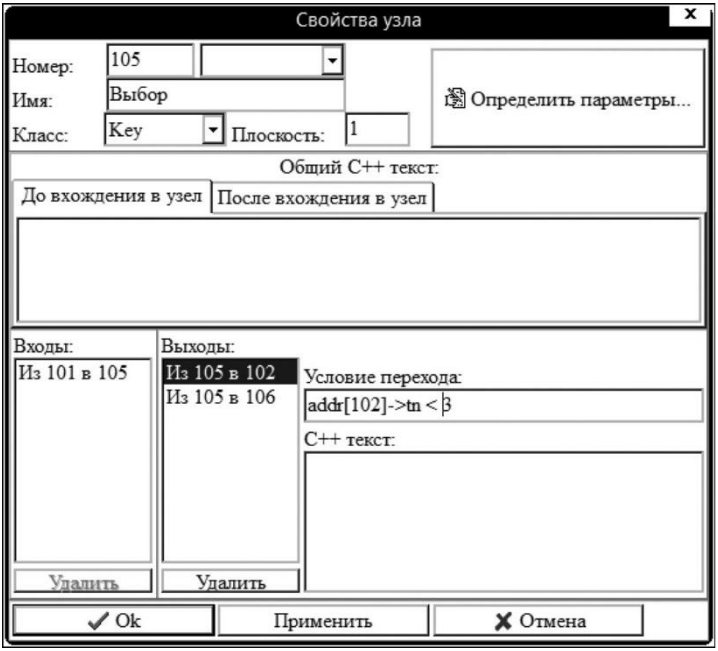
Генераторда қалыптастырылған транзактілер бөлігі қызмет көрсету бойынша кезекке тұрады (егер кезектегі клиенттер саны үштен аз болса), ал кейбір бөлігі бірден жүйеден шығып, қызмет көрсетілмеген өтініштер терминаторына түседі. көмекші түйінді қолдануға болады, мысалы key типі, егер tn өрісі түйін кезегінде үштен аз параметр және басқа жағдайда

қызмет көрсетілмеген транзакт терминаторы кезінде сұранымдар кезекке түседі (бұл шарттар сәйкес ауысуды таңдау кезінде кезекке шығу үшін 9.5-суретте көрсетілгендей клапан параметрлері және Ауысу шарты өрісіне беріледі).

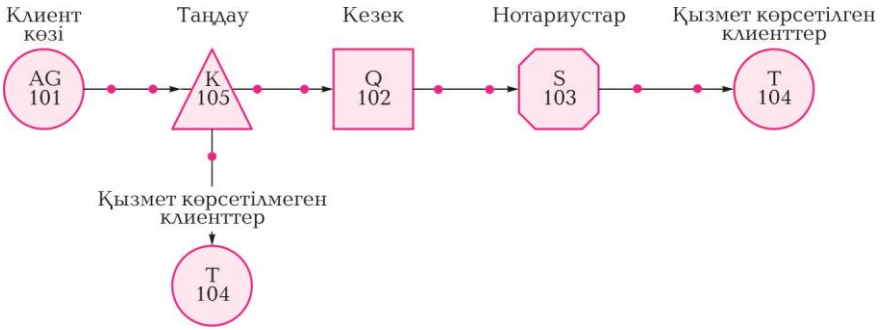
Түйіннен кейін сұраным кезегі екі қызмет көрсету каналы бар қызмет көрсету құрылғысына түседі. Есеп шарты бойынша қызмет көрсету уақыты экспопонциалды заң бойынша 7 мин орташа қызмет көрсету уақытымен бөлінеді (бастапқы деректер бір үлгідегі уақытта беріледі). Содан кейін сұранысқа қызмет көрсету құрылғысы қызмет көрсетілген сұранымдар терминаторына түсуі қажет.

Есепке жасалған талдау негізінде Gem графикалық конструкторының көмегімен 9.6-суретте көрсетілген графикалық модельді құруға болады.

Осыдан кейін ModBed (9.7-сурет) батырмасының көмегімен модельдің жалпы параметрлерін береміз, қажет болғанда ModEnd батырмасының көмегімен нәтижені шығаруға арналған параметрлерді береміз.



Сур. 9.5. клапан параметрлері



Сур. 9.6. Бағандардың моделі

Рис. 9.7. Общие параметры модели

Содан кейін мәзір пәрменін пайдаланып, үлгі баған моделінің дұрыстығын тексеремізІске қосу / графиктің дұрыстығын тексеріңіз және қате болмаған жағдайда, C ++ файлын Орындау/Генерациялау C ++ файлының мәзір командасының көмегімен жасаймыз. Үлгіні Файл/Сақтау пәрменін пайдаланып pgf-файлында сақтау қажет, қажет болған жағдайда оны өзгертуге болады.
Төменде келтірілген C ++ файлының мәтіні келтірілген.

```
#include <Pilgrim.h>
forward
{
   int fw;
```

```
modbeg("Нотариалдық контора", 107, 1000000,
   (long time(NULL), none, 102, none, 104, 2);
тозақ («Клиент көзі», 101, none, expo, 60.0 / 12,
  none, none, 105);
желі (күлгін, күлгін)
{
       жоғарғы (102):
       Кезек («кезек», жоқ, 103); орын;
       жоғарғы (103):
       serv («Notaries», 2, none, expo, 7.0,
     none, none, 104); орын;
       жоғарғы (104):
       мерзімі («Қызмет көрсетілетін
     клиенттер»); орын;
       жоғарғы (105):
       егер (addr [102] -> tn <3)
       {
       Else
       {
       fw = 106
       }
   key("Таңдау", fw);
   place;
   top(106):
   term("Қызмет көрсетілмеген клиенттер");
   place;
   fault(123);
   }
modend("pilgrim.rep", 1, 8, бет);

return 0;

   }
```

Pilgrim модельдік жобасын құру және іске қосу. Модельдеу моделі Microsoft Visual C ++ ортасында жасалады. Мысал ретінде Visual Studio 2000-ні қолдана отырып, үлгіні құру процесін қарастырайық. Алдымен үлгі жобаны жасауыңыз керек.Ол үшін Visual C ++ қабықшасының File мәзірін New... ішкі мәзірін таңдаңыз, содан кейін Projects қойындысында Win32 Application терезесінің қалқымалы терезесінде таңдалады. Сол терезеде, жоба атауын және оның орналасқан жерін дискіде көрсету керек. Содан кейін OK түймесін басу бос жобаны жасайды.

Содан кейін үлгінің мәтінін қамтитын жобаға файл қосу қажет. Бұл әрекетті орындау үшін Files қойындысының қалқымалы терезесінде C ++ Source File таңдаңыз, файл атауын енгізіңіз, Add to project құсбелгісін қойыңыз, содан кейін OK түймешігін нұқыңыз Visual C ++ қабығының File мәзірін таңдауыңызға болады New... ішкі мәзірін таңдай аласыз.Жасалған файлда үлгінің мәтінін (мысалы, бұрын ұсынылған) немесе бұрын дайындалған файлдан көшіріп алу қажет.

Содан кейін жобаға Comctl32.lib (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Lib \ COMCTL32.LIB), Pilgrim.lib (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Lib \ PILGRIM.LIB) кітапханалары қосылған, Pilgrim.res ресурстар файлы (C: \ Program Files \ Microsoft Visual

Studio \ VC98 \ projects \ PILGRIM.RES) және тақырыптама Pilgrim.h (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Include \ Pilgrim.h) және Simulate.h (C: \ Program Files \ Microsoft Visual Studio \ VC98 \ Include \ SIMULATE.H).Осы файлдарды қосу үшін, Сіз Project мәзірі → Project To Project файлдарын таңдап, содан кейін пайда болатын терезеде қалаған файлды таңдап, OK түймесін басыңыз. Visual Studio 2005 және 2008 үшін жоба сипаттарында (Alt + F7 деп аталатын) Configuration Properties/ Linker / Input» сипатын таңдаңыз және «libc.lib» мәтінін Ignore Specific Library өрісіне енгізіңіз

Содан кейін, модель құрастырылып, құрастырылды: Build → Rebuild Allқұрылымды қалпына келтіріңіз. Үлгінің мәтінінде қате болмаған жағдайда, Build → Executeмәзірін пайдаланып қарамастан, қарамастан немесе Visual C ++ ортасында іске қосылатын орындалатын файл жасалады.

Үлгі басталғанда, негізгі терезе пайда болады (сурет 9.8), модельдеу үрдісі арқылы басқарылатын мәзір элементтерінің көмегімен. Төменде ең маңызды мәзір элементтері бар.

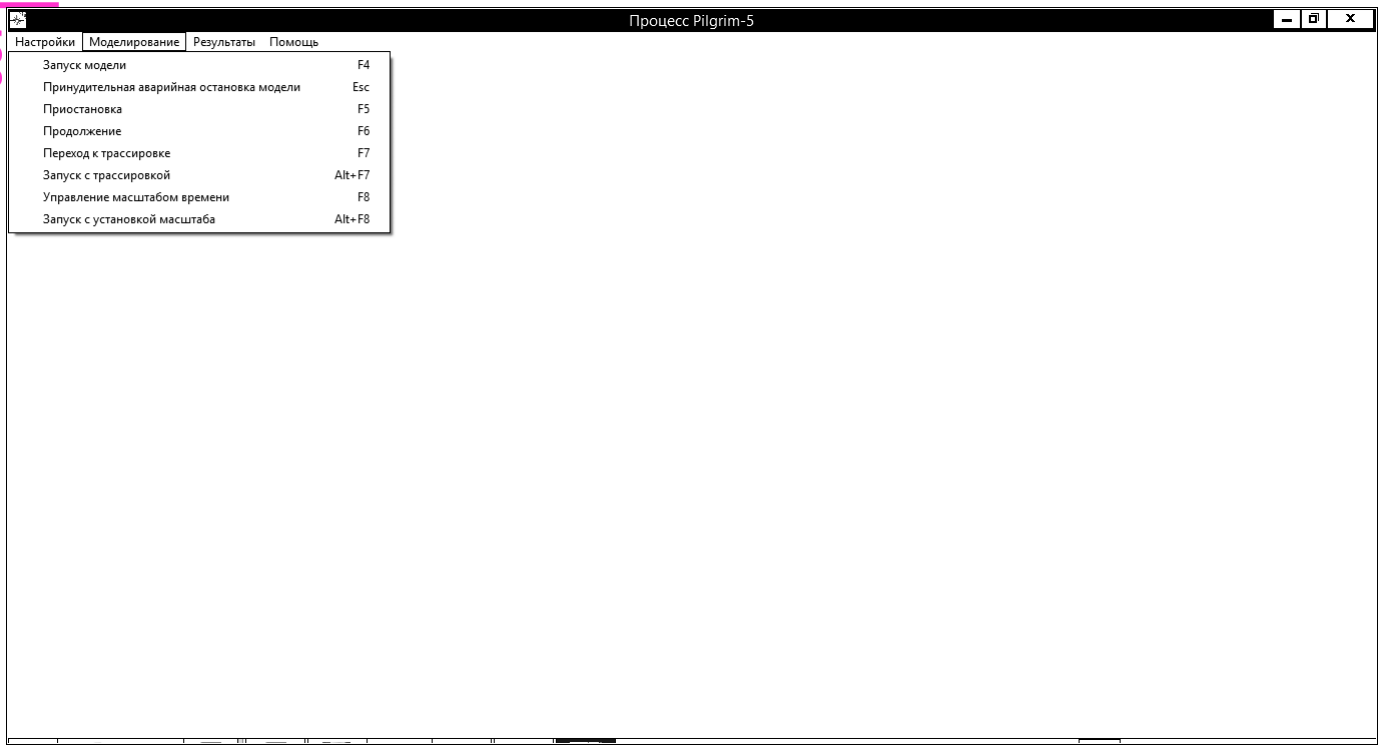
Модельдеу → Модельді іске қосу (F4) - модельдеуді бастаңыз.

Модельдеу → тоқтата тұру (F5) —модельдеуді тоқтату.

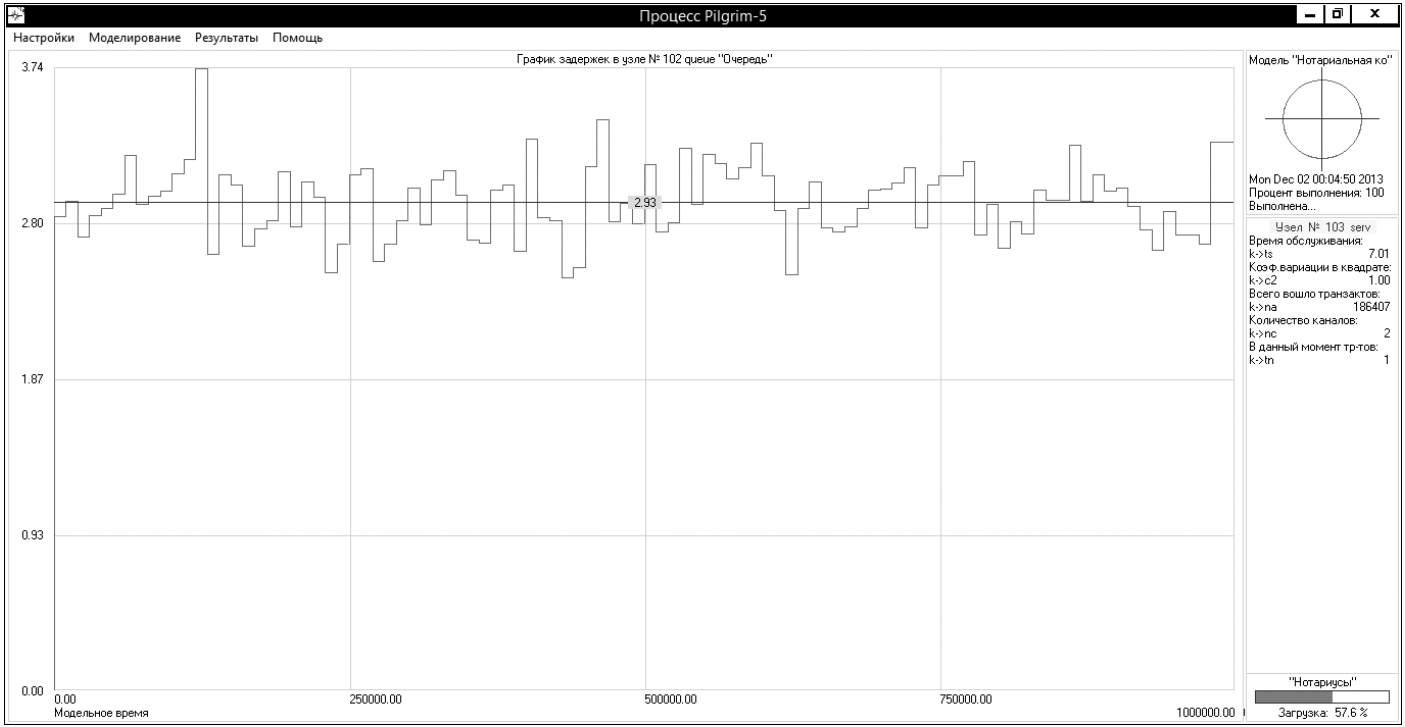
Модельдеу → Жалғасы (F6) —модельдеуді қайтару.

Нәтижелері → торап Параметрлері (F9) - параметрлер терезенің оң жағында көрсетілетін түйінді таңдаңыз.

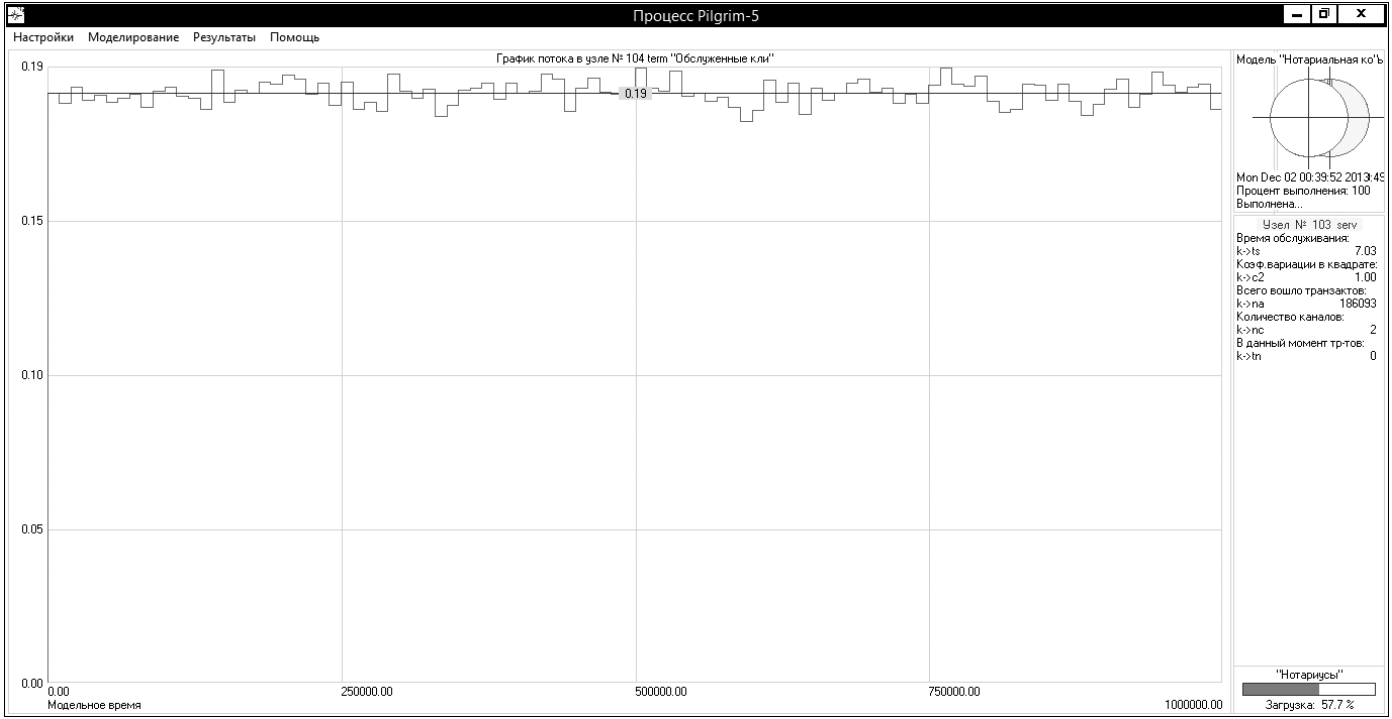
Нәтижелер → Кезектегі кешіктіру динамикасы (F10) - modbeg операторының p6 параметрімен белгіленген кезекке кешігу динамикасын көрсетуге мүмкіндік береді (9.9 суретті қараңыз).



Сурет. 9.8. Pilgrim моделінің терезесі



Сурет. 9.9. Pilgrim моделінің терезесі



Сурет. 9.10. Pilgrim моделінің терезесі

```
*-------------------------------------------------------------------------------------------*
|  НАЗВАНИЕ МОДЕЛИ:     Нотариальная контора                                                |
|  ВРЕМЯ МОДЕЛИРОВАНИЯ: 1000003.70                                          Лист: 1         |
|-------------------------------------------------------------------------------------------|
|    |                |      |    |       |          |   2      |       |    |    |              |
| No | Наименование   | Тип  |Точ-|Загруз-|  M [t]   |  C [t]   |Счетчик|Кол.|Ост.|Состояние узла|
|узла|     узла       | узла | ка |ка(%=),| среднее  |квадрат   |входов |кан.|тр. |в этот момент |
|    |                |      |    |Путь(км)| время   |коэф.вар.|и hold |    |    |              |
|----+----------------+------+----+-------+----------+----------+-------+----+----+--------------|
|    |                |      |    |       |          |          |       |    |    |              |
| 101 Источник клиент ag       -      -        5.00       1.00  200166    1    1         открыт |
| 102 Очередь         queue    -      -        2.93       2.73  186407    1    0         открыт |
| 103 Нотариусы       serv     -   %= 57.6     7.01       1.00  186407    2    1         открыт |
| 104 Обслуженные кли term     -      -        9.94       0.73  186406    0    0         открыт |
| 105 Выбор           key      -   %=  0.0     0.00       1.00       0    1    1         открыт |
| 106 Необслуженные к term     -      -        0.00     999.99   13758    0    0         открыт |
|    |                |      |    |       |          |          |       |    |    |              |
*-------------------------------------------------------------------------------------------*
```

Сур. 9.11. Модельдеу нәтижелері

Нәтижелер → Ағыс динамикасы (F11) —modbeg операторының белгілі бір р8көрсеткішінің терминаторға кіруінде транзакт ағысының динамикасына көрсетуге мүмкіндік береді (сур. 9.10).

Құрылған модельге арналған модельдеудің кестелік нәтижелері сур. 9.11 келтірілген.

Модельдеу нәтижелерін талдау. Модельдің аралық нәтижесінің кестесінде (сур. 9.11 қараңыз) 1000003,7 сағ тең периодтағы модель жұмысы жайлы статистикалық деректі көреміз. Бастапқы деректі тексеру үшін клиенттің келуі мен орташа қызмет көрсету уақыты аралығындағы орташа уақытты білуге болады. Ол 5 мин және 7,01 мин (101 және 103 M[t] түйінінің алаңы) құрайды, ол бастапқы дерекке сай келеді. Бұл уақыт ішінде нотариустың орташа жүктеуі 57,6% (103 түйінді жүктеу) құрады. Кезектегі клиенттің орташа күту уақыты 2,93 мин (узла 102 түйіннің M[t]) құрады. Қызмет көрсетілген өтінімдер жүйесіндегі орташа болу уақыты 9,94 мин (104 түйіннің M[t] алаңында) тең болды.

Нотариалды конторaға барлығы 200 166 клиент (101 түйіннің Есептегіш кірісінің алаңы). Олардың ішінен 13 758 (106 түйіннің Есептегіш кірісінің алаңы) қызмет көрсетуден баст тартылды. Осылайша, қызмет көрсетуден бас тарту ықтималдылығы 13 758/200 166 = 0,0687 құрады.

Аналитикалық есептеу келесі нәтиже бергенін есіңізге салайық.

1.Кезекта болған орташа уақыт — 2,92 мин.

2.Жүйеде болған орташа уақыт — 9,92 мин.

3.Қызмет көрсетуден бас тарты ықтималдылығы — 0,069.

Осылайша, модельдеудің алынған нәтижелері аналитикалық есептің алдын орындалған нәтижесімен келісіледі.

Pilgrim модельдеу пакетінде құрылған модель, оның параметрлерін оңай өзгертуге мүмкіндік беретіндігін атап өткен жөн. Мысалы, қызмет көрсету каналдарының санын, кезектегі орын санын арттыруғы немесе қысқартуға, клиенттің келу уақытының аралының таралу заңы мен қызмет көрсету уақыты мен олардың параметрлерінің таралу заңын өзгерртуге болады. Сонымен қатар бұны C++-файлды қайта түрлендіріп, Gemграфикалық конструкторында да және сәйкес түйін функциясы параметрлерінің мәнін өзгертіп, қалыптастырылған бағдарлама кодында істеуге болады.

БАҚЫЛАУ СҰРАҚТАРЫ МЕН ТАПСЫРМАЛАРЫ

1 Имитациялық модельдеу пакеттері не үшін керек?
2. Имитациялық модельдерді жасап шығару кезінде имитациялық модельдеу пакеттері мен бағдарламалаудың әмбебап тілдерін пайдаланудың қандай артықшылықтары мен кемшіліктері бар?
3. Pilgrim имитациялық модельдеу пакеті қандай мүмкіндіктерге ие?
4. Pilgrim пакетіндегі модель қандай негізгі нысандардан тұрады?
5. Какие основные типы узлов модели в пакете Pilgrimпакетінде модель түйінінің қандай негізгі типтерін білесіз?
6. Pilgrim пакетіндегі транзакт қандай параметрлерге ие?
7. Pilgrim пакетіндегі түйін қандай параметрлерге ие?
8. Pilgrim пакетіндегі бағдарламаның жалпы құрылымы қандай?
9. Pilgrim пакетіндегі модель жұмысының нәтижесі қандай түрде шығарылады?
10. Gem графикалық конструкторы нені істеуге мүмкіндік береді?

ЗЕРТХАНАЛЫҚ ТӘЖІРИБЕ

10.1 ЖАЛҒАН КЕЗДЕЙСОҚ САННЫҢ БАЗАЛЫҚ ГЕНЕРАТОРЛАРЫН ЗЕРТТЕУ

Тең ықтималды таратумен [0; 1) аралықта бірге жалған кездейсоқ сандарды генерациялау үшін бағдарламаны (кіші) құрастыру және ретке келтіру. 10.1 кестеден нұсқаны таңдаңыз. Бұл кездейсоқ сандардың генераторының түрін, бастапқы шарттарын және т.б. көрсетеді. Берілген іріктеу көлемі және ұсақтау аумағы үшін аралық саны [0; 1) жиіліктер мен статистикалық бөлу таралу функциясының гистограммасын құрастыру, математикалық күтудің бағдарламаланған бағаларын, дисперсияны, екінші және үшінші сәттерді алу. Алынған нәтижелерге талдау жасау.

Кесте 10.1. Зертханалық жұмыс № 1 нұсқалық тапсырмалар

Нұсқаның нөмері	Тетік түрі	Бастапқы мәліметтер	Іріктеу көлемі	Ұсақтау аумағының саны
1	Мультипликативті, формула (2.4)	Y_0= 4 003, M = 4 096	256	16
2	Әмбебап, формула (2.9), κ = 3	Y_k— кез келген	500	16
3	Аддитивті, формула (2.6)	Y_1= 3 215, Y_2= 4 073, m=4096·4	5 000	10
4	Универсальный, формула (2.9), κ = 1	Y — кез келген	1 600	18

10.1 кесте жалғасы

Нұсқаның нөмері	Тетік түрі	Бастапқы мәліметтер	Іріктеу көлемі	Ұсақтау аумағының саны
5	Квадратты конгруэнттік әдіс, формула (2.11)	*Y*— кез келген, *I* = 12	5 000	25
6	Ковэю квадраттыәдісі, формула (2.12)	*I* = 9, ҮjшартынанҮmod4 = 2	5 000	12
7	Квадратты конгруэнттік әдіс, формула (2.11)	*Y*— кез келген, *I* = 12	7 000	16
8	Макларена— Марсальи Әдісі	*к* = 256	1 000	21
9	Аралас, формула (2.5)	*Y*— кез келген	4 000	16
10	Аддитивті, формула (2.6)	Y = 4091, Y_2= *m* - 5, *m*= 4 096·4	1 000	16
11	Жалпыланған аддитивті , формула (2.7)	*r* = 6, ..., *X*6кездейсоқ сандар кестесінен	6 000	16
12	Жалпыланған аддитивті , формула (2.7)	*r*= 10, x^ ..., x_{10}кездейсоқ сандар кестесінен	6 000	16
13	Аддитивный, формула (2.6)	Y = 3 971, Y_2 = 1 013, *m* = 4 096·4	2 000	21
14	Макларена— Марсальи Әдісі	*к*= 64	2 000	16
15	Аралас, формула (2.5)	*Y* = 3 845	1 000	16
16	Жалпыланған аддитивті, формула (2.7)	*r* = 8, xj, ..., x_8кездейсоқ сандар кестесінен	6 000	26

10.1 кесте жалғасы

Нұсқаның нөмері	Тетік түрі	Бастапқы мәліметтер	Іріктеу көлемі	Ұсақтау аумағының саны
17	Макларена—Марсальи Әдісі	k = 128	5 000	26
18	Аралас, формула (2.5)	Y = 4 001	1500	16
19	Әмбебап, формула (2.9), k = 2	Y_k— кез келген	4 000	21
20	Мультипликативті, формула (2.4)	Y_0= 3 091, M = 4 096	2 000	21

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 2. ЖАЛҒАН КЕЗДЕЙСОҚ САННЫҢ САПАСЫН ТЕКСЕРУ

Зертханалық жұмыста 1 жасағанда алынған нәтижелерді пайдалана отырып, Пирсон, Колмогоров өлшемдері арқылы жалған кездейсоқ сандардың сапасын тексеріңіз, сонымен қатара кестеде 10.2. көрсетілген өлшемдер:

Кесте 10.2. Зертханалық жұмыс № 2 нұсқалық тапсырмалар

Нұсқаның нөмері	Өлшемі
1	Жанама белгісі бойынша сапасын тексеру
2	Серия санының өлшемі, бөлетін элемент p= 0,25
3	Нөлдік серия ұзындығының сынақтамасы, бөлетінэлемент p= 0,3
4	Бірлік серия ұзындығының сынақтамасы, бөлетінэлемент p= 0,4
5	Покер-сынақтама, k= 2
6	Коллекционер өлшемі

10.2 кесте жалғасы

Номер варианта	Өлшемі
7	Покер-сынақтама, k= 8
8	Серия санының өлшемі, бөлетін элемент p= 0,5
9	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент p= 0,5
10	Бірлік серия ұзындығының сынақтамасы, бөлетінэлемент p= 0,25
11	Коллекционер өлшемі
12	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент p= 0,4
13	Серия санның сынақтамасы, бөлетін элемент p= 0,45
14	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент p= 0,45
15	Бірлік серия ұзындығының сынақтамасы, бөлетін элемент p= 0,45
16	Серия санның сынақтамасы, бөлетін элемент p= 0,6
17	Покер-сынақтама, k= 10
18	Жанама белгісі бойынша сапасын тексеру
19	Нөлдік серия ұзындығының сынақтамасы, бөлетін элемент p= 0,65
20	Бірлік серия ұзындығының сынақтамасы, бөлетінp= 0,65

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 3. БЕЛГІЛЕНГЕН ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Кесте 10.3. арқылы анықталған тапсырманың нұсқасына сәйкес кездейсоқ шамаларды түрлендіру бағдарламасын құрастыру.Іріктеу бағдарламасы бойынша алынған, жиілік гистограммасын және статистикалық тарату функциясын құрастырады, кездейсоқ шаманы математикалық күтілімі мен дисперсиясын бағалау.Эмпирикалық деректер мен теориялық таралу арасындағы сәйкестік Пирсон өлшемі немесе Колмогоров өлшемі арқылы тексеріледі.Кездейсоқ шаманың іріктеу көлемі 1 000-нан кем емес. Ұсақтау аралықтарының саны k = 15 немесеk = 25.

Кесте 10.3. Зертханалық жұмыс № 3 нұсқалық тапсырмалар

Нұсқаның нөмері	Таралу заңы	Құрылу әдісі
1	$F(x)=\begin{cases}0,4(x-1)^3+0,4, & x\in[0;0,5);\\ 0,3x+0,2, & x\in[0,5;1,5);\\ 0,4(x-1)^3+0,6, & x\in[1,5;2]\end{cases}$	Іріктеу әдісі
2	$F(x)=\begin{cases}\sqrt{0,25-(x-0,5)^2}, & x\in[0;0,5);\\ 0,3125x+0,34375, & x\in[0,5;1,3);\\ 1,25x-0,875, & x\in[1,3;1,5]\end{cases}$	Кері функци әдісі
3	$F(x)=\begin{cases}0,25x^2, & x\in(0;1);\\ 1,14x-0,89, & x\in[1;1,5);\\ 1-0,08(x-3)^2, & x\in[1,5;3]\end{cases}$	Іріктеу әдісі
4	$F(x)=\begin{cases}x, & x\in[0;0,5);\\ 0,5, & x\in[0,5;1);\\ 2(x-1)^2+0,5, & x\in[1;1,5]\end{cases}$	Кері функци әдісі
5	$F(x)=\begin{cases}0,3x, & x\in[0;0,5);\\ 3x-1,35, & x\in[0,5;0,7);\\ 0,25x+0,575, & x\in[0,7;1,7]\end{cases}$	Іріктеу әдісі
6	$F(x)=\begin{cases}x^2, & x\in[0;0,5);\\ 1,1x-0,3, & x\in[0,5;1);\\ 0,4x+0,4, & x\in[1;1,5]\end{cases}$	Кері функци әдісі

10.3 кесте жалғасы

Нұсқаның нөмері	Таралу заңы	Құрылу әдісі
7	Ұшбұрышты $f(x)=\begin{cases}0, x \le a;\\ \dfrac{2(x-a)}{(b-a)(c-a)}, a < x \le c;\\ \dfrac{2(b-x)}{(b-a)(b-c)}, c < x \le b;\\ 0, x > b.\end{cases}$ $a=1;\ b=5;\ c=2$	Іріктеу әдісі
8	$F(x)=\begin{cases}0{,}2\cdot 10^{x}-0{,}2, & x \in [0;0{,}3);\\ 1{,}5x-0{,}25, & x \in [0{,}3;0{,}7);\\ 0{,}25x+0{,}625, & x \in [0{,}7;1{,}5]\end{cases}$	Кері функци әдісі
9	$F(x)=\begin{cases}0{,}15x, & x \in [0;1);\\ 0{,}35x-0{,}2, & x \in [1;2);\\ 0{,}875x-1{,}25, & x \in [2;2{,}4);\\ 0{,}15x-0{,}49 & x \in [2{,}4;3{,}4]\end{cases}$	Іріктеу әдісі
10	$F(x)=\begin{cases}0{,}8x^{2}, & x \in [0;0{,}5);\\ 0{,}7x-0{,}15, & x \in [0{,}5;1);\\ 1-e^{-0{,}8x}, & x \in [1;\infty)\end{cases}$	Кері функци әдісі
11	Ұшбұрышты (7 тасырмаға қарау) $a=0;\ b=10;\ c=5$	Іріктеу әдісі
12	$F(x)=\begin{cases}0{,}05x^{3}, & x \in [0;2);\\ 1-2e^{-0{,}602x}, & x \in [2;\infty)\end{cases}$	Кері функци әдісі
13	$F(x)=\begin{cases}\sqrt{x}, & x \in [0;0{,}25);\\ 0{,}25x+0{,}4375, & x \in [0{,}25;2{,}25]\end{cases}$	Іріктеу әдісі

10.3 кесте жалғасы

Нұсқаның нөмері	Таралу заңы	Құрылу әдісі
14	$F(x)=\begin{cases}\frac{2(e-1)}{e}x, & x\in[0;0,5);\\ 1-e^{-2x}, & x\in[0,5;\infty)\end{cases}$	Кері функци әдісі
15	Үшбұрышты (7 тасырмаға қарау) a= 4; b= 5; c = 4,7	Іріктеу әдісі
16	$F(x)=\begin{cases}1-e^{-2x}, & x\in[0;1);\\ \frac{x-3+2e^2}{2e^2}, & x\in[1;3]\end{cases}$	Кері функци әдісі
17	$f(x)=\sqrt{R^2-(x-a)^2};$ $R=\sqrt{\pi/2}; a=-2$	Іріктеу әдісі
18	$F(x)=\begin{cases}0,4(x-1)^3+0,4, & x\in[0;0,5);\\ 0,3x+0,2, & x\in[0,5;1,5);\\ 0,4(x-1)^3+0,6, & x\in[1,5;2]\end{cases}$	Кері функци әдісі
19	$f(x)=\frac{\pi}{2}\sin(\pi x),\quad 0<x\le 1$	Іріктеу әдісі
20	Үшбұрышты (5 тасырмаға қарау) a = -3; b = 10; c = 0	Кері функци әдісі

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 4. ҚАЛЫПТЫ ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Қалыпты таралу заңының әдісі бойынша кездейсоқ шаманы түрлендіру үшін, орталық шекті теоремаға негізделген, сонымен қатар тапсырманың нұсқасына (10.4-кесте) сәйкес анықталған әдісі арқылы бағдарлама құрастыру. Көрсетілген таралу заңының параметрі N(m; o2) түрінде болады.Іріктеу бағдарламасы бойынша алынған, жиілік гистограммасын және статистикалық тарату функциясын құрастырады, кездейсоқ шаманы математикалық күтілімі мен дисперсиясын бағалау.Эмпирикалық деректер мен теориялық таралу арасындағы сәйкестік Пирсон өлшемі немесе Колмогоров өлшемі арқылы тексеріледі. Кездейсоқ шаманың іріктеу көлемі 1 000-нан кем емес. Ұсақтау аралықтарының саны k = 15 немесе k = 25

Кесте 10.4. Зертханалық жұмыс № 4 нұсқалық тапсырмалар

Нұсқа	Таралу заңы	Құрылу әдісі
1	Қалыпты, N(3; 1)	Аппроксимация әдісі
2	Қалыпты, N(0; 1)	Бокса және Малера әдісі
3	Қалыпты, N(3; 1)	Марсальи және Брея процедурасы
4	Қалыпты, N(-2; 0,81)	Аппроксимация әдісі
5	Қалыпты, N(4.3; 0,5)	Бокса және Малера әдісі
6	Қалыпты, N(2; 0,9)	Марсальи және Брея процедурасы
7	Қалыпты, N(3.5; 0,9)	Аппроксимация әдісі
8	Қалыпты, N(2; 0,2)	Бокса және Малера әдісі
9	Қалыпты, N(-1,5; 1,7)	Марсальи және Брея процедурасы
10	Қалыпты, N(0; 0,1)	Аппроксимация әдісі
11	Қалыпты, N(2; 1)	Бокса және Малера әдісі
12	Қалыпты, N(-2; 1)	Марсальи және Брея процедурасы
13	Қалыпты, N(2; 1)	Аппроксимация әдісі

10.4 кесте жалғасы

Нұсқа	Таралу заңы	Құрылу әдісі
14	Қалыпты, N(1; 0,7)	Бокса және Малера әдісі
15	Қалыпты, N(5; 1)	Марсальи және Брея процедурасы
16	Қалыпты, N(4,7; 0,6)	Аппроксимация әдісі
17	Қалыпты, N(1,5; 0,1)	Бокса және Малера әдісі
18	Қалыпты, N(-4; 0,1)	Марсальи және Брея процедурасы
19	Қалыпты, N(3; 0,1)	Аппроксимация әдісі
20	Қалыпты, N(2,75; 2,8)	Бокса және Малера әдісі

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 5. ЖИІ ҚОЛДАНЫЛАТН ТАРАЛУ ЗАҢЫ БАР КЕЗДЕЙСОҚ ШАМАНЫ ТҮРЛЕНДІРУ

Бета-тарату, гамма-тарату, логарифмдік-қалыпты тарату және Вейбулла таралуына бағынатын кездейсоқ шаманы түрлендіру бағдарламаларын құрастырамыз. Іріктеу бағдарламасы бойынша алынған, жиілік гистограммасын және статистикалық тарату функциясын құрастырады, кездейсоқ шаманы математикалық күтілімі мен дисперсиясын бағалау. Эмпирикалық деректер мен теориялық таралу арасындағы сәйкестік Пирсон өлшемі немесе Колмогоров өлшемі арқылы тексеріледі. Кездейсоқ шаманың іріктеу көлемі 1 000-нан кем емес. Ұсақтау аралықтарының саны k = 15 немесе k = 25

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. МОНТЕ-КАРЛО ӘДІСІ БОЙЫНША МОДЕЛЬДЕУ

Тапсырманы шешуге арналған машиналық модельдеу (Монте-Карло әдісі) арқылы тапсырма нұсқасына сәйкес бағдарламаны құрастыру.бағаланған ықтималдықтардың нақты мәнін қамтитын b = 0.95 дәйектіілікпен алынған бағалау үшін сенімді аралықтарды құру. Нәтижелердің дұрыстығын тапсырманың аналитикалық шешімімен тексереді.

Нұсқаның тапсырмалары

1.Екі ракетамен қаруланған жойғыш, ауада жаулап алу мақсатында жіберіледі. Мұндай жағдайдағы жоюшының шығару ықтималдылығы, оның ішінде мақсатты шабуыл жасау мүмкіндігі, p1-ге тең. Егер жоюшы осы позицияға жеткізілсе, онда ол әрқайсысы бір-біріне тәуелсіз екі ракетаны босатып, ықтималдықпен p2 ықтималдығы бар нысанның маңына алынады. Егер ракета нысананың маңайына түсірілсе, ол оны ықтималдық p3-ға түседі. Мақсаттың жеңіліс табу ықтималдығын бағалаңыз.

2.Атуды бір-біріне жақын бір сызықтың бойында орналасқан жанғыш бүйір бойымен екі снаряд жүзеге асырады (k> 2). Әрбір снаряд басқалардың тигізуіне қарамастан, бірінші багқа ықтималдықпен p1 түседі, екіншісі - ықтималдығы p2 және тағы да сол сияқты. Багты жандыру үшін бір багқа екі соққылар немесе іргелес багқа екі рет тигізу қажет. Багты жандыру ықтималдығын бағалау.

3.Үш снаряд бойынша нысанаға ату жүзеге асырылады. Снарядтар бір-біріне тәуелсіз нысанаға тиеді.Әрбіт снаряд үшін нысанаға тигізу ықтималдылығы p0тең. Егер де нысанаға бір наряд тигізсе, ол нысанды (оны саптан шығарады) p1ықтималдылықпенжеңіледі; егер де екі снаряд — p2 ықтималдылықпен; егер де үш снаряд —p3 қтималдылықпен. Мақсатынан жеңілудің жалпы ықтималдылығын бағалау.

4.Екі ұшақ арасында әуе шайқасы болып жатыр: жойғыш және бомбылаушы. Атуды жойғыш бастайды: ол бомбылаушыны бір рет атадыжәне p1 ықтималдылықпен құлатады. Егер де бомбылаушы бұл атқанда құламаса, ол жойғышты атады да және p2ықтималдылықпен құлатады. Егер де жойғыш бұл атқанда құламаса, ол тағы да бомбылаушыны атады да және p3 ықтималдылықпен құлатады. Келесі шайқас қорытындысының ықтималдылығын бағалау:

A — бомбылаушыны құлату;

B — жойғышты құлату;

C — кем дегенде бір ұшақты құлату.

5.Натуды бір біріне тәуелсіз әрқайсысы өзінше нысанаға атуды жүргізеді. Олардың әрқайсысы шайқас қорының кпатрондары бар. Бір атқанда нысанаға тигізу ықтималдылығы, бірінші атқандағырг (г = 1, 2, ..., N) тең. Өзінің нысанасына бірінші тигізгеннен бастап атқыш атуын тоқтатады.Келесі оқиғаның ықтималдылығын бағалау:

A — Барлық атқыштарда бірдей кем дегенде бір жұмсамаған патрон қалуы;

B — атқыштардың ешкайсысында барлық шайқас қоры жұмсалынбайды;

C — әйтеуір атқыштың біреуі барлық шайқас қорын жұмсайды, алқалғандары — барлығын емес.

6.Natуды кезекпен бір нысанаға атады. Ату бірінші тигізгенге дейін жалғасады. Әрбір атқыш үшін нысанаға тигізу ықтималдылығы pi (i = 1, 2, ..., N) тең. Қайсысы нысанаға бірінші тигізсе, ол жеңген болып саналады.Әрбір атқышта қорында nпатрон бар болады. 1-ші атқыштың жеңіске жету ықтималдығын бағалау.

7.Бомбылаушы және оның екі шабуылдаушы жойғыштарының арасында әуе шайқасы болуда. Бомбылаушы атуды бастайды; ол әрбір жойғышқа бір рет атады және оны p1 ықтималдылығымен құлатады. Егер де бұл жойғыш құламаса, онда ол бақаның тағдырына қарамастан бомбылаушыны атады және оны p2 ықтималдылығымен құлатады.

Келесі шайқас қорытындысының ықтималдылығын бағалау:

A — бомбылаушыны құлату;

B — екі жойғышты құлату;

C — ең болмаса бір жойғышты құлату;

D— ең болмаса бір ұшақты құлату;

E — дәл бір жойғышты құлату;

F— дәл бір ұшақты құлату.

8.

Адам, белгілі бір халық тобына жатады, қара торы p1ықтималдылығы болады, сарғыш шашты адам —p2ықтималдылығымен, ақсары — p3ықтималдылығымен, қызғылт сары — p4 ықтималдылығы. Жорамалмен алты адамнан тұратын топ таңдалады.Келесі оқиғаның ықтималдылығын бағалау:

A — топтың құрамында кем дегенде төрт ақсарының болуы;

B — топтың құрамында ең болмаса қызғылт сарының болуы;

C— топтың құрамында сарғыш мен ақсарының тең болуы.

9.Құрал үш түйіннен тұрады.Құралды қосқан кезде p1ықтималдылығымен бірінші түйінде ақау болады, екінші түйінде — p2ықтималдылығы, үшінші түйінде — p3ықтималдылығымен. Түйіндердегі ақау бір біріне тәуеліз болады. Үш түйіннің әрқайсысы, сөзсіз құралмен жұмыс жасағанда қажет. Түйіннің істен шығуы үшін, кем дегенде екі ақау болу керек. Құрылғы nқосқандакауіпіз тұру ықтималдылығын бағалау.

10.Ұшақтар тобының құрамында: бір басқарушы және екі жетектегіш,нысанда бомбылауға бағытталған.әркайсысы бір бір бомбадан алып жүреді.Басқарушы ұшақ көздеуде болады, жетектегіштер — басқарушы белгі бермейінше бомбылауды жүзеге асыра алмайды. Нысанға келместен бұрын топ әуе шабуылына қарсы қорғаныстан өтеді,ондағы әрбір ұшақ бір біріне тәуелсіз p ықтималдылығымен құлатады. Егер нысанаға басқарушы ұшақ екі бірдей жетекшісімен келетін болса, олар P12 ықтималдылығымен нысанды қиратады.Басқарушы ұшақ бір жетекшісін алып шықса, Pu ықтималдылығымен нысанды қиратады.Бір бақарушы ұшақ жетекшісіз, P1,0ықтималдылығымен нысанды қиратады.Егер де

басқарушы ұшақ құласа, онда әрбір жетекші егер де ол сақталса нысанға шығады да Р0,1 ықтималдылығымен қиратады. Нысанға қарсы әрекет ету кезіндегі толық ықтималдығын бағалау.

11.Зауыт өнім өндіреді, олардың әрқайсысында р ықтималдылықпен ақау болады. Цехта үш бақылаушы бар; өнімді тек бір бақылаушы тексеріп қарайды,бірдей ықтималдылықпен бірінші, екінші немесе үшінші. Бірінші бақылаушының ақауды табу (егер де болса) ықтималдылығы рг(i= l, 2, 3) тең. Егер де өнім цехта жарамсыз болса, ол зауыттың техникалық бақылау бөліміне (ТББ)түседі, онда егер ақау бар болса,р0ықтималдықпен анықталады.

Келесі оқиғаның қтималдылығын бағалау:

А — өнім жарамыз болады;

В — өнім цехта жарамсыз болады;

С — өнм зауытта ТББ жарамсыз болады.

12.Радиолокациондыстанцияы нысанда бақалауға алады,кедергілері пайдаланылатын немесе пайдаланбайтын болуы мүмкін. Егер нысан кедергіні пайдаланбаса, онда бір анлалымдағы танция шолуы р0 ықтималдылығын көрсетеді; егер пайдаланса — р1< р0 ықтималдылығымен. Айналу кезінде кедергі қолданылатындығы р ықтималдығы және қалған айналымдарда кедергі қалай қолданылатынына байланысты емес.Нысанның шолу айналым-дарында n кем дегенде бір рет көрету ықтималдылығын бағалау.

13.Топта, үш барлаушы-ұшақтардан тұратын, зымыранмен оқ жаудыру болжамымен, нысанның координаталары білу мақсатында жаудың аймағына жіберіледі. Нысанды жеңу үшін n зымырандар бөлінеді. Нысанның анықталған координаталарында бір зымыранның жеңіліс ықтималдығыр1,ал анықталмаған үшін - р2. Әрбір барлаушы нысана аймағына шығудан алдын жаудың әуе шабылуына қарсы құрылғымен құлатуы мүмкін; оның ықтималдылығы р3. Егер де барлаушы құлатылмаса, ол нысанның координаталарын радио арқылы хабарлайды. Барлаушының радиоаппараты р4 сенімділігі бар. Координатаны анықтау үшін бір ғана барлаушының хабарламасы жеткілікті.Барлаушының әрекетін ескере отырып нысанның жеңілу ықтималдылығын бағалау.

14.Natқыштар ішінен төрт топты таңдап алуға болады: а1өте жақсы атқыш, а2жақсы, а3орташажәне а4нашар. Бірінші топтағы атқыштың бір атқанда нысаныға тигізу ықтималдылығы Pi (i= 1, 2, 3, 4) тең. Жорамалмен екі атқыш шақырылады, сол бір нысанаға ататын.Нысанаға ең болмаса бір рет тигізу ықтималдылығын бағалау.

15.Нысана алма мен екі сақинадан тұрады.бір рет атқандағы алмаға тигізу ықтималдылығы р0тең, бірінші сақина — р1, ал екінші— р2; нысанаға тигізе алмау ықтималдылығы р3. Нысанада бес рет атылыс жүзеге асыралды. Екі рет алмаға тигізу және бір рет екінші сақинаға тигізу ықтималдылығын бағалау.

16.Құрылғы 10 түіннен тұрады. Әрбір түйіндегі сенімділік (увқыт ішіндегіттоқтаусыз жұмыс ықтималдылығы) р тең. Түйіндер саптан бір біріне тәуелсіз шығады.Уақыт ішіндегітықти-малдылығын бағала: а) кем дегенде бір түйіннің қабылдамауы; б) дәл бір түйіннің қабылдамауы; в) дәл екі түйіннің қабылдамауы; г) кем дегенде екі түйіннің қабылдамауы.

17.Мақсаты бойынша төрт тәуелсіз атыс жүргізіледі. Әр атыс кезіндегі тигізу ықтималдылығы әр түрлі және тең: p1, p2, p3, p4. Ықтималдылықты бағалауP04; P14; P24; P34; P44бірде бір,бір,екі, үш, төрт рет тигізу; ең дегенде бір реттигізу R14ықтималдылығы; кем дегенде екі рет тигізуR24ықтималдылығы.

18.Зауыт өнім өндіреді, олардың әрқайсысы сандық төрт түрімен тартылады. Бірінші сынақ p1 ықтималдықпен өнімді қауіпсіз түрде өтеді; екінші —p2 ықтималдылықпен; үшінші — p3 ықтимал-дылықпен және төртінші —p4 ықтималдылықпен. Өнімді жақсы өткізу ықтималдығын бағалау:

A — барлық төрт сынақ;

B— дәл екі сынақ (төртеуінің ішінен);

C— кем дегенде екі сынақ (төртеуінің ішінен).

19.

Зауыт өнім өндіреді, олардың әрқайсысы г (бір біріне тәуелсіз) ықтималдылықпен ақауының бар болуы. Ақауды тексеру кезінде, егер де ол болса р ықтималдылықпен анықталады.Бұйымды бақылау кезінде зауыттан n өнім таңдалады. Келесі оқиғаның ықтималдылығын бағалау:

A — ешбір өнімде ақаулық болмайды;

B— өнімnішінен дәл екі ақаудың болуы;

C — өнімnішінен кем дегенде екі ақаудың болуы;

20.Ұшақ оқ жаудырадыптәуелсіз атыстармен; атыстың әр қайсысы p1ықтималдылығымен аймаққа тигізеді, ол ұшақты жедел түрде жеңіліс тапқызады; жанармай шанына p2ықтималдылықпен тигізіледі және ұшаққа p3ықтималдылықпен тигізеді. Снаряд, жанармай шанына тиген, онда сағатына жанармай клитрі ағып жатқан тесік қалдырады.Жанармай М литрін жоғалтқан ұшақ шайқасқа жарамсыз болып қалады. Ұшаққа оқ жаудырғаннан бір сағаттан кейін шайқасқа жарамсыз болу ықтималдылығын бағалау.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. КЕЗДЕЙСОҚ КЕЗУІ МОДЕЛЬДЕУ

Әрбір тапсырманың бағдарламалық өңдеуді кездейсоқ кезуі имитациялық модельін көздейді, оның көмегімен қажетті нәтижелер алуға болады. Тәжірибелердің белгілі бір санын жүргізу нәтижесінде зерттелетін параметрдің статистикалық таралуын (гистограмма және эмпирикалық

бөлу функциясы) құру және белгілі заңдардың бірі (қалыпты, экспоненциалды, логорифмдік-қалыпты және т.б.) арқылы алынған таратуды мақсаттылық жуықтамасы анықтайды.

Нұсқаның тапсырмасы

1.Бір өлшемді кездейсоқ кезуі. M = 20 қадам үшін жаяу жүргіншілерге кететін қашықтығын анықтайтын модельін жасаңыз.

2.Екі өлшемді кездейсоқ кезуі. M = 10 қадам үшін жаяу жүргіншілерге кететін қашықтықты анықтайтын модельін жасаңыз.

3.Сіңіретін экранда қарапайым кездейсоқ кезуі. Кезуі уақытын анықтау үшін машиналық модельін жасаңыз.

4.«Аралар» квадраттық торда.«Омарта» N-нен «Ара» бастапқыда координаттар басында ортасымен бір шеңберде орналасқан. Әрбір кезеңде әрбір «ара» төрт бағыттың бірінде кездейсоқ түрде жылжиды: солтүстікке, оңтүстікке, шығысқа және батысқа.M = 8 қадам үшін жаяу жүргіншілерге кететін қашықтықты анықтайтын модельін жасаңыз. Әрбір уақытша аралықта

ішінде «Аралар» бірлік ұзындығының қадамын жасайды. Орташа есеп N «Аралар» бойынша орындалады.

5.Ұшбұрышты торда кезуі. Үшбұрышты торда «Аралар» кездейсоқ кезуі имитациялық модельін жасаңыз. Уақыт бойынша әрбір қадамда «Ара» алты ықтимал бағыттардың бірінде теңдей ауысады.M = 8 қадам үшін «Ара»қандай қашықтыққа кетеді?

6.Жауын тамшыларының түсу моделі. Жеңіл желдің кездейсоқ екпінің әсерінің арқасында, жаңбыр тамшыларының түсуін шаршы торда кездейсоқ кезуін модельдеуге болады(сур. қар. 6.9). Қозғалыс көлденең сызықтан (жер үстінен) h қашықтықта орналасқан түйіннен басталады. Сонымен қатар төменгі қадам рі ықтималдылығы жоғарғы қадам p^ ықтималдылығы үлкенірек. Секіру ықтималдылығын бірдей таңдаған абзал:pi= 0,5; pi= 0,1; p^_ = = p^ = 0,2.

Тамшылаудың көлденең сызыққа жететін уақытын және t -ның h функциясына тәуелділігін (4 ... 6 мәндері) анықтаңыз.

7.Кездейсоқ кезуді шектеу. Сіңіретін экранда қарапайым кездейсоқ кезуі тапсырмасы келесі түрлендірулер түрінде қарастырылуы мүмкін. Бір өлшемді торда нүктелерде сіңіретін түйіндері (тұзақ) бар x = 0 жәнех = a (a> 0). Бөлшек x0(0 <x0<a) нүктесінен қозғалысты бастайды және бірдей ықтималдылықпен жақын көршілес түйіндерге өтеді. Бөлшектің оның сіңуіне дейін өту tуақытын анықтаңыз.

8.Жасушада кезуі. Жасушаларда имитациялық кездейсоқ кезуі модельін жасаңыз. Уақыт бойынша әрбір қадамда аралар үш бағыттың біреуінде бірдей ықтималдылықта қозғалады. M = 8 қадам үшін «ара» қанша қашықтыққа кетеді?

9.Үш өлшемді торда кездейсоқ кезуі. Бөлшектің кететін қашықтықтығын бағалау, бұл үш өлшемді торда кезуі бірдей ықтималды

болады.Кезуі қадамдардың саны M = 10. Зерттеліп жатқан процеспен қатар, барлық үш координаттар үшін бастапқыдан кететін қашықтықты бөлек анықтау.

10.Перистентті кездейсоқ кезуі. Персистентті кездейсоқ кезуі өту немесе «секіру»ықтималдығы соңғы өтуге байланысты.Персистентті бір өлшемді кездейсоқ кезуі жағдайда, қадамдар тек жақын көршілес түйіндерде жасалады. k – 1 қадам жасады деп ойлайық. Содан соң k-й қадам жасайды сол бағытта а ықтималдылығымен, а қарама-қарсы бағытта 1 – аықтималдылығымен қадам жасайды. ¬

M = 8 қадамын онда а = 0,2 и а = 0,4 бастапқы жағдайдан бөлшектің орташа кетуін анықтау қажет.

11.Айнымалы қадаммен кездейсоқ кезуі. Барлық рұқсат етілген секіру ұзындығы бар бір өлшемді кездейсоқ кезуді қарастырыңыз. Ықтималдылығы, қадам ұзындығы j тең,P(j)= exp(-j) түрде болады. Бастапқы күйінен 10 қадамнан кейін қашықтыққа кететінін анықтау.

12.Айнымалы қадаммен тор.Кездейсоқ кезуін 11 тапсырмадағыдай қарастырамыз, қадам ұзындығының таралу ықтималдылығыP(j) = a j2. Бұл жердеа= 6/p2. Бастапқы күйінен сегіз қадамнан кейін қашықтыққа кететінін анықтау.

13.Үздіксіз кездейсоқ кезуі. Кез-келген кездейсоқ кезудің алғашқы үздіксіз модельдерінің бірі Рейлидің 1919 жылы ұсынған болатын. Рейли үлгісінде әр қадамның ұзындығы - а ықтималдық тығыздығы p (a) және әрбір қадамның кездейсоқ бағытымен бөлінген кездейсоқ айнымалы.

Ықтималдылығының тығыздығы p (a) 0,5-тен 1,5-ке дейінгі диапазонда теңдестірілсе және қозғалыс бағыты 1 ° дәлдікпен бірдей таңдалса, жаяу жүргіншінің бастау нүктесінен бес қадам кеткенін анықтау.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. ЖҮЙЕНІҢ ЖАППАЙ ҚЫЗМЕТІН МОДЕЛЬДЕУ

ЖЖҚ модельдеу бағдарламасн құру, нұсқа тапсырмаларымен анықталған. ЖЖҚ тиімді функциялаудың негізгі сипаттамасын бағалау. Нәтижелерін нақты мағынасымен салыстыру, тарау 7 келтірілген формула бойынша аналитикалық есептеу кезінде алынған.

Нұсқа тапсырмалары

1.Бір арналы ЖЖҚ бас тартуы бірыңғай телефон желісін көрсетеді. Жолдың бос емес уақытына келген тапсырыс (қоңырау) бас тартылды. Оқиғалардың барлық ағындары ең қарапайым. Ағынның қарқындылығы минутына X = 0,95 шақыру болып табылады. Әңгіменің орташа ұзақтығы t = 1 мин. Жұмыс тәртібі орнатылған ЖЖҚ ықтималдық сипаттамаларын анықтаңыз.

2.Бір арналы ЖЖҚ бас тартуы минутына X = 0,5 сұрау қарқындылығы бар тапсырыстардың қарапайым ағымын алады. Тапсырыстың қызмет

көрсету уақыты t = 1.5 мин. Жұмыс тәртібі орнатылған ЖЖҚ ықтималдық сипаттамаларын анықтаңыз.

3.Компьютер орталығында бес дербес компьютер бар. Есептеу орталығына түсетін тапсырыстардың қарапайым ағымы сағатына X = 10 тапсырмасының қарқындылығына ие. Тапсырманы шешудің орташа уақыты - 12 минут. Барлық компьютерлер бос болмаса, тапсырыстан бас тартылады. Қызмет көрсету жүйесінің ықтималдық сипаттамаларын (есептеу орталығы) табыңыз.

4.Аудиторлық ұйымда тәулігіне X = 1,5 тапсырыс қарқындылығы бар қызметтерге қарапайым сұраныс ағыны алады. Қызмет мерзімі көрсетілген заң бойынша бөлінеді және орта есеппен үш күнге тең болады. Аудиторлық ұйымда аудитті жүзеге асыратын бес тәуелсіз бухгалтер бар (тапсырыс қызметі). Тапсырыстың кезегі шектелмейді. Аудиторлық ұйымның стационарлық тәртібінде жұмыс істейтін ЖЖҚ ретінде ықтималдық сипаттамаларын анықтаңыз.

5.Техбайқау пунктінде сағатына X = 4 қарқындылығы бар қарапайым тапсырыстар (автомобильдердің) ағынына ие болады. Байқау уақыты көрсетілген заңға сәйкес бөлу және орташа есеппен 17 минутқа тең, кезекте тұрған автокөлік бесеуден аспау керек. Техбайқау пунктінің ықтималдық сипаттамаларын орнатылған тәртіпте анықтаңыз.

6.Кәсіпорынның бухгалтериясында кассирлер бар, олардың әрқайсысы сағатына орта есеппен 30 адам жұмыс істей алады. Жалақы алатын қызметкерлердің ағымы - қарапайым,қарқынды, сағатына 40 қызметкерге тең . Кассалық бөлімде кезек шектелмеген. Қызмет көрсету уақыты экспоненталық бөлу туралы заңға бағынады. ЖЖҚ-ның ықтималдық сипаттамаларын стационарлы тәртіппен есептеп,осындай өнімділікпен алғашқы екі жұмыс істейтін болса кәсіпорын үшін үшінші кассирді қабылдаудың мақсаттылығын анықтаңыз.

7.Жинау цехының құрастыру бөлімінде үш қойма қызметкері жұмыс істейді. Орта есеппен 1 минутта 0,8 жұмысшы (X = 0,8) құрылғы келеді. Бір жұмысшыға қызмет көрсету қоймашының t = 1,0 минутын ұстайды. Кезекте шектеу жоқ. Құралдың артында кезекте жұмысшылардың ағымы Пуассон болып табылады, және қызмет көрсету уақыты экспоненталық бөлу заңына бағынады. Қызметкердің жұмысының 1 минутының құны - 30 д.е., ал қойма қызметкері - 15 д.е. Стационарлы жұмыс тәртібінде аспаптық бөлімшедегі (қызмет көрсету уақытының құны)

Осы қызмет ұйымы үшін семинардың орташа шығындарын табыңыз.

8.Билеттер кассасы үздіксіз жұмыс истейді. Билетті бір кассир сатады. Орташы қызмет көрсету уақыты — 2 мин әрбір адамға. Жолаушылардың орташа саны, бір сағат ішінде билет кассасында билеттерді сатып алғысы келетін, сағатына X = 20 жолаушы. Жүйенің барлық ағындары қарапайым болып табылады. Стационарлы жағдайындағы жұмыс тәртібі ЖЖҚ сипаттамаларын анықтаңыз.

9.Автокөлікті диагностикалау қызметі - бұл бір арналы ЖЖҚ бас тартуға ұшыраған. Қызмет орны бос емес кезде қабылданған диагностикаға арналған тапсырыстар қабылданбайды. Диагностикаға арналған аппараттардың ағынының қарқындылығы сағатына X = 0,5 автомобиль. Диагностиканың орташа ұзақтығы t = 1,2 сағат. Жүйенің барлық оқиғалары қарапайым болып табылады. Жүйенің орнатылған тәртібінің сипаттамаларын анықтаңыз.

10.Автокүю станциясы ЖЖҚ бір арнада қызмет ететін және бір бағанмен қамтиды. АЖҚС-дағы аумағында бір мезгілде үш автокөлікке жанармай құю үшін кезекте тұруға рұқсат етілген. Кезекте кез-келген үш автокөлік бар болса, станцияға келетін тағы бір автокөлік, бірақ айналып өтетін болады. Май құюға келген автокөліктердің қозғалысы минутына X = 0.7 автомобиль қарқындылығына ие. Құю процесі орташа алғанда 1,25 минутты құрайды. Барлық ағындар қарапайым.Жүйенің орнатылған тәртібінің ЖЖҚ сипаттамаларын анықтаңыз.

11.Темір жолды сұрыптау тау-кен құрамдары келеді сағатына X = 2 қарқындылығы бар. Тау-кен құрамының қызмет көрсететін орташа уақыт 0,4 сағат. Сол уақытта келген құрамдар,тау-кен бос болмағанда, кезекке тұрып,және келу паркінде күтіледі,кез-келген бір құрамды күтуге болатын үш қосалқы жүру жолы бар. Сол уақытта келген құрамдар, барлық үш қосалқы жүру жолы келу паркінде бос емес болса, сыртта осы жолға кезекке тұрады. Оқиғалардың барлық ағындары ең қарапайым. Жүйенің орнатылған тәртібі ЖЖҚ сипаттамаларын анықтаңыз.

12.АЖҚС станциясының жұмысы қарастырылған, онда үш толтыру бағандары бар. Бір автокөлікке құю орташа алғанда 3 минутты құрайды. Орташа алғанда, жанармай құю бекетінде әрбір минут сайын көлік жанармай құюға мұқтаж. Кезекте орын саны шектеусіз. Автожанармай құюға кезекте тұрған барлық автокөліктер өздерінің кезектерін күтеді. Жүйенің барлық ағындары қарапайым болып табылады. Стационарлық режимде жанармай құю станциясының жұмысының ықтималдық сипаттамаларын анықтаңыз.

13.Автокөліктің техникалық қызмет станциясына орта есеппен екі сағат сайын бір көлік келеді. Станцияда алты жөндеу бекеті бар. Қызметті күткен автомобильдердің кезегі шексіз. Бір машинаның орташа қызмет көрсету уақыты - 2 сағат. Жүйенің барлық ағындары ең қарапайым. Автомобильді техникалық қызмет көрсету станциясының сипаттамаларын анықтаңыз.

14.Екі арналы қарапайым ЖЖҚ бар бас тартуға ұшыраған. Оның тапсырыс сағатына X = 3 қарқындылығы бар оған кіру ағынын алады. Тапсырысты бір рет қолдануға арналған орташа жұмыс уақыты t = 0,5 сағат. Әр тапсырыста 5 ш.б. кіріс пайда болады. Арнаның мазмұны - 3 сағ/ш.б өтеді. ЖЖҚ арналарының санын үшке дейін ұлғайтудың экономикалық жағынан тиімді екендігін анықтаңыз.

15.Дүкенде сағатына орташа 30 тұтынушыға қызмет көрсете алатын бір сатушы жұмыс істейді. Сатып алушылардың ағыны қарқындылығы сағатына 60 тұтынушыға тең қарапайым. Барлық сатып алушылар «шыдамсыз» және кезек кезегінде бес адам болған жағдайда кетеді (оларға қызмет көрсететіндерге қосымша). Оқиғалардың барлық ағындары ең қарапайым. Тұрақты жұмыс істеу үшін дүкеннің сипаттамаларын анықтаңыз.

16.Телефон аппараттарының тоғыз жұмыс арнасымен кіруі сағатына орташа есеппен 120 тапсырыс қабылдайды. Барлық арналар бос болмаса, тапсырыстан бас тартылады. Бір арнадағы орташа қызмет көрсету уақыты - 4 минут. Жүйенің барлық ағындары қарапайым болып табылады. Телефон станцияларының сипаттамаларын анықтаңыз.

17.АЖҚС станциясының жұмысы қарастырылған, онда үш толтыру бағандары бар. Бір автокөлікке құю орташа алғанда 4 минутты құрайды. Орташа алғанда, жанармай құю бекетінде әрбір минут сайын көлік жанармай құюға мұқтаж. Кезекте орын саны шектеусіз. Автожанармай құюға кезекте тұрған барлық автокөліктер өздерінің кезектерін күтеді. Жүйенің барлық ағындары қарапайым болып табылады. Стационарлық режимде жанармай құю станциясының жұмысының ықтималдық сипаттамаларын анықтаңыз.

18.Кезек шарты үш орынмен қарапайым бір арналы ЖЖҚ үшін ықтималдық сипаттамаларын есептеңіз, X = 4 тапсырыс бойынша сағатына, t = 0,5 сағ.Егер де кезек орындардың төртке дейін көбейтсек, осы сипаттамалардың қалай өзгеретінін біліңіз.

19.Есептеуге арналған тапсырмалар қабылдайтын бір арналыЖЖҚ - ЭЕМ. Тапсырыстар ағыны ең қарапайым болып табылады, бұл тапсырыстар арасындағы орташа уақыт аралығы t = 10 минут. Қызмет мерзімі экспоненталық заңға сәйкес, ^ bs = 8 минуттың математикалық күтуімен бөлінеді. ЖЖҚ-дағы тапсырыстардың орташа санын, кезектегі тапсырыстың орташа саны, жүйеде және кезекте тапсырыстардың қалыпты уақытын анықтау.

20.Жүйенің жалпы қызметін — үш терезеден тұратын билет кассасы (үш кассиримен) және шектелмеген кезекпен. Билетті сатып алғысы келетін жолаушылар, 20 минут сайын орташа есеппен бес адам келеді. Жолаушылар ағымы қарапайым деп санауға болады. Кассир орташа есеппен 10 минут ішінде үш жолаушыға қызмет көрсетеді. Қызмет көрсету уақыты экспоненталық бөлу туралы заңға бағынады.Стационарлы жағдайындағы жұмыс тәртібі ЖЖҚ сипаттамаларын анықтаңыз.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. ЭКСПЕРИМЕНТТІҢ АЙЛАЛЫ ЖОСПАРЛАУ

Әрбір тапсырма кездейсоқ кезуі бағдарламалық өңдеуді модельдеуді

дамытуды көздейді, оның көмегімен зертханалық жұмыстың екі бөлігіне қажетті нәтижелер алына алады.Жұмыстың бірінші бөлігінде көрсетілген параметрдің орташа мәнін бағалау үшін эксперименттің нәтижелері жоспарланып, қабылданады.Екінші бөлімде белгілі бір дәлдікпен берілген параметрдің үлгі талғамын бағалау үшін эксперименттің нәтижелерін жоспарлайды және алады.Сынамалық эксперимент жүргізу барысында жоспарлау кезінде параметрдің статистикалық таралуы жасалады және қалыпты аппрокцимациялық мақсаттылық таралу осы заңымен анықталады.Осыған байланысты эксперименттің көлемін анықтау үшін таңдалады. Нұсқаның тапсырмасыЗертханалық жұмыс № 7 алынады.

ЗЕРТХАНАЛЫҚ ЖҰМЫС № 6. PiLGRiM МОДЕЛЬДЕУ ҚҰРАЛ ТӘСІЛІМЕН ЖҮЙІНІҢ ЖАЛПЫ ҚЫЗМЕТІН МОДЕЛЬДЕУ

Pilgrim модельдеу құрал тәсілдерін пайдалану көмегімен № 8зертханалық жұмыстың белгілі бір нұсқа тапсырмаларыменЖҚЖ модельдеуді орындау. Модельдеу нәтижесін № 8 зертханалық жұмыс алынған нәтижелермен салыстыру..

Әдебиеттер тізімі

1. *Бережная Е.В. Математические методы моделирования экономических систем : оқу құралы / Е.В.Бережная, В. И. Бережной. – М. : Қаржы және статистика, 2002.*

2. *Варфоломеев В.И. Алгоритмическое моделирование элементов экономических систем: Практикум : оқу құралы / В. И. Варфоломеев. – М. : Қаржы және статистика, 2000.*

3. *Емельянов А.А. Имитационное моделирование экономических процессов : оқу құралы / А. А. Емельянов, Е.А.Власова, Р. В.Дума. – М. : Қаржы және статистика, 2002.*

4. *Золотарев В.В. Компьютерное моделирование : оқу құралы/ В.В. Золотарев, Г. В. Овечкин, П. В. Овечкин. – Рязань : РГРТУ, 2008.*

5. *Кельтон В. Имитационное моделирование. Классика CS/ В. Кель- тон, А.Лоу. – 3-ші басылым. – СПб. : Питер ; Киев : BHV, 2004.*

6. *Кнут Д. Искусство программирования. Том 2. Получисленные алгоритмы / Д. Кнут. – 3-ші басылым. – М. : Вильямс, 2007.*

7. *Лабскер Л.Г. Теория массового обслуживания в экономической сфере : оқу құралы / Л.Г.Лабскер, Л. О.Бабешко. – М. : Банктер және биржа ; ЮНИТИ, 1998.*

8. *Нейлор Т. Машинные имитационные эксперименты с моделями экономических систем / Т. Нейлор. – М. : Мир, 1975.*

9. *Основы компьютерного моделирования систем / Д. Е. Артемкин,*
В. В. Баринов, Г. В. Овечкин, И. М. Степнов ;. А. Н. Пылькин редакциясымен. – М. : Бином. Білімдер зертханасы, 2004.

10. *Пузикова Л.А. Основы имитационного моделирования на ЭВМ. Методические указания к лабораторным работам / Л. А. Пузикова. – Рязань : РГРТУ, 1991.*

11. *Рыжиков Ю.И. Имитационное моделирование. Теория және технология / Ю. И. Рыжиков. – СПб. : КОРОНА принт ; М. : Альтекс-А, 2004.*

12. *Советов Б.Я. Моделирование систем : оқулық / Б.Я.Советов, С. А.Яковлев. – М. : Жоғарғы мектебі., 2001.*

13. *Шеннон Р. Имитационное моделирование систем: наука и искусство / Р. Шеннон. – М. :Әлем, 1978.*

Мазмұны

Оқу баспасы

Овечкин Геннадий Владимирович,
Овечкин Павел Владимирович

Компьютерлік модельдеу

Оқулық

Редактор Л. В. Толочкова, Ә.Шапауов
Компьютерлік беттеу: Л. М. Беляева
Түзетуші Л. В. Гаврилина

Шығ. № 101115958. Қол коюға жіберілді 29.09.2014. Пішімі 60 х 90/16.
Гарнитурасы «Балтика». Қағаз офсетті. Басу офсетті. Шар. бас. Парағы 14,0.
Таралымы 2500 дана. Тапсырыс №

«Академия» баспа орталығы». ЖШСwww.academia-moscow.ru129085, Мәскеу, Мир д-лы, 101В, б. 1.
Тел./факс: (495) 648-0507, 616-00-29.
29.04.2014№ РОСС RU. АЕ51. Н16592 санитарлы-эпидемиологиялық қорытынды.
Баспаның электронды тасығышынан басылған.

«Тверь полиграфиялық комбинаты» АОҚ, 170024, Тверь қ., Ленин д-лы, 5.
Телефон: (4822) 44-52-03, 44-50-34. Телефон/факс: (4822) 44-42-15.
Homepage–www.tverpk.ru Электронды пошта(E-mail) –sales@tverpk.ru