



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS**  
**MATEMATINĖS SISTEMOTYROS KATEDRA**

**Kristina Salatkaitė**

**ATIDĖTŲ MOKĖJIMŲ ĮTAKA REZERVŲ ĮVERTINIMUI**

Magistro darbas

**Vadovas:**

**doc. V.Karpickaitė**

**KAUNAS,2005**



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS**  
**MATEMATINĖS SISTEMOTYROS KATEDRA**

**TVIRTINU**  
**Katedros vedėjas**

**prof. habil.dr. V.Pekarskas**

**2005 06 11**

**ATIDĖTŲ MOKĖJIMŲ ĮTAKA REZERVŲ ĮVERTINIMUI**

Taikomosios matematikos magistro baigiamasis darbas

**Kalbos konsultantas      Vadovas**

**J. Džežulskienė**

**doc. V. Karpickaitė**

**2005 05 31**

**2005 06 03**

**Recenzentas      Atliko**

**doc.V. Dargis**

**FMMM 3 gr. stud.**

**2005 06 01**

**K. Salatkaitė**

**2005 05 27**

**KAUNAS,2005**

## **KVALIFIKCINĖ KOMISIJA**

**Pirmininkas:** Leonas Saulis, profesorius (VGTU)

**Sekretorius:** Eimutis Valakevičius, docentas (KTU)

**Nariai:** Algimantas Jonas Aksomaitis, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Rimantas Rudzkis, profesorius (MII)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, UAB „Elsis“ generalinio direktoriaus pavaduotojas

**SUMMARY**

The pay delays influence on reserves are very important problem for insurance companies. We consider two particular examples of models that attempt to deal with pay delays in insurance, in this work. The first model provides one explanation for underwriting cycle; the second indicates the effects of delays. We were reckoned reserves by chain – ladder method, multiplicity method, and separation method. We had made pay delays modeling program with generated data; this program should made easier to estimate pay delays influence on reserves. We have done simulation with different damage distribution and come to some conclusion like: the insurance premium was enough for the insurance company, because the reserves were positive. There are no big differences in reserves results, calculated by different methods.

Įvadas	9
1. Bendroji dalis	11
1.1. Atidėti mokėjimai draudime	11
1.2. Pasirašytasis ciklas	13
1.3. Atidėtų mokėjimų laiko atžvilgiu modelis	15
1.4. Atidėtų mokėjimų dėl teisminių procesų modelis	17
1.5. Grandininių laiptų metodas	19
1.6. Multiplikatyvinis metodas	20
1.7. Atskyrimo metodas	22
2. Tiriamoji dalis	25
2.1. Atidėjimų priežastys	25
2.2. Atidėtų mokėjimų modeliavimas	26
2.2.1 Duomenys	26
2.2.1.1. Išmokų skaičius	27
2.2.1.2. Išmokų dydis	28
2.2.1.3. Draudimo įmokos	29
2.2.2 Matematinis modelis	29
2.2.3 Atidėtų mokėjimų įtakos rezervams įvertinimas	30
2.2.4 Netekimų ir išlaidų koeficientų įvertinimas	41
3. Programinė realizacija ir instrukcija vartotojui	44
3.1. Sukurtos modeliavimo programos aprašymas	44
Išvados	46
Literatūra	47
1 priedas. Grafikai	48
2 priedas. Programos tekstas	54
3 priedas. Straipsnio santrauka	79

## LENTELIŲ SĄRAŠAS

1.2.1. JAV "nuosavybės-nelaimingų atsitikimų" kompanijų pasirašytasis ciklas	13
1.5.1. Grandininių laiptų metodo pradiniai duomenys	19
1.5.2. Grandininių laiptų metodo sukauptų mokėjimų lentelė	19
1.6.1. Multiplikatyvinio metodo rezervai	22
1.7.1. Atskyrimo metodo duomenys	22

## PAVEIKSLŲ SĄRAŠAS

2.1.1. Laikas nuo draudiminio įvykio iki pranešimo apie draudiminį įvykį	25
2.1.2. Nuostolių atlyginimo laiko santykinių dažnių histograma	26
2.2.3.1. Rezervai apskaičiuoti grandininių laiptų metodu, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį	30
2.2.3.2. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį	31
2.2.3.3. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį	31
2.2.3.4. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyviniu metodais, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį	32
2.2.3.5. Rezervai su atidėjimais, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį	33
2.2.3.6. Rezervų likutis, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį	33
2.2.3.7. Rezervai apskaičiuoti grandininių laiptų metodu, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį	34
2.2.3.8. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį	35
2.2.3.9. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį	35
2.2.3.10. Rezervai apskaičiuoti grandininių laiptų, atskyrimo ir multiplikatyviniais metodais, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį	36
2.2.3.11. Rezervai apskaičiuoti grandininių laiptų metodu, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį	36
2.2.3.12. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį	37
2.2.3.13. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį	37
2.2.3.14. Rezervai apskaičiuoti grandininių laiptų, atskyrimo ir multiplikatyviniais metodais, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį	38
2.2.3.15. Rezervai apskaičiuoti grandininių laiptų metodu, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį	38

2.2.3.16. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį	39	
2.2.3.17. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį	39	
2.2.3.18. Rezervai apskaičiuoti grandininių laiptų, multiplikatyviniu ir atskyrimo metodais, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį		40
2.2.4.1. Nuostolių koeficientas	41	
2.2.4.2. Išlaidų koeficientas	42	
2.2.4.3. Bendras koeficientas	43	



## ĮVADAS

Laisvosios rinkos sąlygomis kiekvienas verslas yra susijęs su techniniais atidėjimais - tai laiku negauti laukiami pinigai, laiku neišmokėti pinigai ir pan. Draudimo veikloje, kaip ir bet kuriame kitame versle, yra atidėjimai – su draudimo veikla susiję draudiko įsipareigojimai, kurie turi užtikrinti draudimo įmonių prievolių, kylančių iš draudimo sutarčių, vykdymą bei išlaikyti stabilią draudimo įmonių finansinę padėtį.

Šiuo metu draudimo rinka Lietuvoje vis dar vystosi, tačiau jos augimo tempai yra akivaizdūs. Tai patvirtina 1998 metų pagrindiniai draudimo rinką apibūdinantys rodikliai: santykis su bendruoju vidaus produktu yra lygus 0,86% (padidėjo 0,2%), o išsivysčiusiose šalyse siekia 9%; vienam gyventojui tenka 26,73 JAV dol. (padidėjo 9,62 JAV dol.), Šveicarijoje šis rodiklis siekia net 3533 JAV dol. Jaunoje, tik 11 metų, Lietuvos draudimo rinkoje, funkcionuojančioje rinkos sąlygomis, įvyko nemažai bankrotų: 1998 metais iš draudimo rinkos pasitraukė UAB „Vicura“, 1999 metais net trys bendrovės – UAB „Ūkio draudimas“, „Verslo draudimas“ ir „KDK draudimas“. Pagrindinės šių bendrovių bankroto priežastys - neteisingas prisiimtų rizikų įvertinimas, blogas draudimo portfelio subalansavimas, neteisingai apskaičiuoti tarifai, neteisingai įvertinti techniniai atidėjimai. Viso to buvo galima išvengti, atliekant išsamesnę ir tikslesnę vykdomos veiklos analizę.

Tokią analizę draudimo įmonėse paprastai atlieka įmonės vyriausieji finansininkai arba aktuarijai (draudimo matematikai). Lietuvos Respublikos draudimo įstatyme yra numatyta, kad kiekviena draudimo įmonė matematiniams bei statistiniams skaičiavimams atlikti privalo turėti finansininką, o gyvybės draudimo įmonės papildomai dar ir draudimo matematiką – aktuariją. Tačiau daugelyje šalių aktuarijai atlieka ne tik gyvybės draudimo, bet ir ne gyvybės draudimo šakų draudimo tarifų, techninių atidėjimų bei kitų matematinių ir statistinių dydžių skaičiavimus.

Šiame darbe yra nagrinėjamas draudimo įmonių techninių atidėjimų įvertinimas ir jų įtaka rezervams. Draudimo įmonės techniniams atidėjimams, rezervams įvertinti naudojami aktuariniai metodai, taip pat šiuolaikinė teorija, grindžiama matematiniu modeliavimu.

Pagrindinis šio darbo uždavinys - tai techninių atidėjimų įtakos rezervams įvertinimas, žinant pagrindinius draudimo veiklos parametrus: draudimo įmoką, polisų skaičių ir pan. Šiuo metu yra nemažai metodų, įvertinančių atidėjimų įtaką ir apskaičiuojančių pačius atidėjimus, tačiau šie metodai dažniausiai taikomi tais atvejais, kai turime daugiau parametrų. Tikėtina, kad jie yra taikomi išsivysčiusiose Vakarų Europos šalyse, kur draudimo veikla vykdoma jau daugelį dešimtmečių, kur susiformavusios senos draudimo tradicijos.

Mūsų tikslas - parodyti, kokią įtaką atidėjimai gali turėti rezervams. Rezervams apskaičiuoti yra taikomi grandinių laiptų, multiplikatyvinis ir atskyrimo metodai. Nagrinėti sudėtingus, tačiau kartais tikslesnius metodus tampa sudėtinga ir atima daug laiko. Atidėjimų įtakos rezervams įvertinimui naudosime matematinį – kompiuterinį modeliavimą, kuris šiais informacinių technologijų laikais yra vienas pagrindinių šiuolaikinės draudimo teorijos įrankių. Kitas uždavinys yra

matematinio modelio, aprašančio atidėtų mokėjimų procesą, sukūrimas. Tam būtina gerai suprasti ir suvokti, visus draudimo veikloje vykstančius realius procesus. Modeliavimas turi privalumą, nes modelis gali būti bet kokio sudėtingumo, į jį galima įtraukti tiek norime įvairių sudėtinių dalių. Vis dėl to norint sukurti labai detalų modelį, reikėtų jį pritaikyti konkrečios draudimo bendrovės situacijai, draudimo rūšių taisyklėms ir pan. Šiame darbe nagrinėjamas bendresnis modelis, į kurį įtraukti, kai kurie svarbiausi veiksniai.

## **1. BENDROJI DALIS**

### **1.1. ATIDĖTI MOKĖJIMAI DRAUDIME**

Draudikas turi apmokėti nuostolius draudiminio įvykio metu. Mes žinome daug veiksnių, kurie gali būti uždelsto draudiminio įvykio nuostolių apmokėjimo priežastimis. Todėl, Ackman (1985) išskyrė 5 veiksnių grupes:

1. Draudimo sutartimi paremtas įvykis gali pasirodyti neatidėliotinas – pavyzdžiui, darbininko ieškinio kompensacija dėl profesinės ligos; šios ligos priežastis gali būti saugos reikalavimų nesilaikymas ilgesnį laikotarpį ir gali būti nepripažintas kaip draudiminis įvykis, praėjus daug metų nuo draudimo sutarties pradžios.
2. Gali būti įvairūs uždelsimai prieš tai, kai draudikas sužino apie draudiminį įvykį.
3. Teisėti draudiko išpareigojimai gali būti ne visada tiksliai apibrėžti, jei delsiama ilgai, kol paaiškės situacija (galbūt per teismo procesus).
4. Gali būti neįmanoma apibrėžti ieškinio dydžio po draudiminio įvykio, pavyzdžiui, autoavarijos metu ieškinio dydis nustatomas po to, kai transporto priemonė yra apžiūrėta ir įkainota žala; daugiau pavyzdžių galima atrasti asmeninio nuostolio bylose.
5. Gali būti mokėjimų atidėjimai draudiko administracinio skyriaus viduje, kai surenkami ieškiniui reikalingi statistiniai duomenys, suvedama (atnaujinama) ir paruošiama byla, bei apmokami nuostoliai.

Apibendrinant, vėlavimai galimi ir prieš ieškinio pranešimą, ir po pranešimo, bet prieš galutinį išmokėjimą.

Norint tiksliai nustatyti verslo finansinę padėtį tam tikru laiko momentu (pvz., ataskaitinių metų pabaigoje), draudikas turi įvertinti leidžiamą paklaidą, dviems atsakomybės tipams, šiuo atveju teisiniu procesui:

1. Laukiamo mokėjimo ieškiniui įvertinimas, atsirandantis dėl įvykių, kurie atsitiko, bet dar nebuvo pranešti draudikui;
2. Ieškinio sumos, kuri turi būti išmokėta, įvertinimas, apie kurią jau buvo pranešta draudikui, bet dar nebuvo galutinai apsispręsta.

Įvertinimo problema – plačiai nagrinėta įvairioje literatūroje. Įvertinimo procesas susideda iš konstravimo ir modelio pritaikymo praeities duomenims bei tolimesnio proceso naudojimo prognozavime. Šis procesas gali paveikti draudiko nustatytą įmoką. Tokiu būdu draudimo tarifų nustatymas dažniausiai paremtas metiniais duomenimis, kurie yra tikrinami ir nėra pakankamai pilni, kad galėtume juos panaudoti netgi ir mažoms klasėms (pvz. transporto priemonių draudimas). Kaip buvo kalbėta anksčiau, atidėjimai priklauso nuo pranešimų uždelsimo ir nuo duomenų surinkimo bei jų analizės. Prognozavimas sudaromas nuo galiojimo termino iki periodo, kuriam bus apskaičiuojami tarifai, vidurinio taško. Tokios prognozės būtinai turi būti paremtos informacija, surinkta iki projekto dienos (Coutts, 1984; Cummins ir Outreville, 1987).

Atidėti mokėjimai yra pateikiami „tvarkymo-reguliavimo“ procese. Draudimo įmokų tarifai yra reguliuojami daugelyje šalių. Gali būti pareikalauta, kad draudimo kompanijos turėtų patvirtintus valdžios tarifus (pvz. JAV) arba priimti nacionalinį tarifą (pvz. Šveicarijoje). Reguliavimas beveik visada sukuria papildomus atidėjimus ir uždelsimus tarp galiojimo termino ir pakeistų tarifų galiojančios datos. Be to, draudimo įmokos tarifai gali būti keičiami rečiau, negu įstatymų nustatyta tvarka.

Pastebima, kad draudimo sutarties pratęsimo uždelsimas egzistuoja daugelyje šalių beveik kiekviename negyvybės draudime. Skirtingai nuo kitų kainų, pavyzdžiui finansinių aktyvų prekės, draudimo sutarties įmokos negali būti pakeistos staiga, kadangi nebus galima pagrįsti naujos informacijos. Dauguma negyvybės draudimo sutarčių sudaromos vienerių metų laikotarpiui. Pavyzdžiui, jei tarifai perskaičiuojami kartą per metus ir naujas įmokų tarifas įsigalioja sausio 1 d., tuomet įmokos draudimo sutarčiai keičiamos vidutiniškai liepos 1d. ir nauji tarifai neįsigalioja visoms sutartims iki gruodžio 31d.

Duomenų surinkimo ir apdorojimo procesas, prognozavimas ateičiai ir uždelsimų sistemos efektas, kuris parodo, kad nors transporto priemonių draudime galimas tarifų galiojimo laikas vieneri metai ir kiekvienos sutarties ilgis irgi tik metai, tačiau

tarifų įvertinimo proceso prognozavimas gali užsitęsti iki aštuonerių metų. Laiko uždelsimas pranešant apie draudiminį įvykį ir surenkant duomenis gali būti šios pasekmės priežastimi. Laiko, skirto apgalvojimui, prailginimas reikalauja priimti keletą subjektyvių sprendimų, pavyzdžiui:

- ? Yra trijų dalinių įsipareigojimų ieškiniai, kurie atsirado 1990 m., tarkime, tinkami dėl įsipareigojimo ieškinių, kurie turėjo atsirasti 1996m. ir buvo apytiksliai sutvarkyti 1998 m. arba vėliau. Pagrindinė problema slypi išsprendimo aritmetikoje, dėl teismų teisėjų kaitos, keičiantis visuomenės sąlygomis. Tai buvo pavadinta “teisėjų dreifu (srove)”.
- ? Modelis, arba bent požiūris, kad reikalinga praeities ir ateities infliacija.
- ? Ieškinių dažnio pasikeitimo tendencija įkainojama ir prognozuojama. Išoriniai faktoriai, tokie kaip klimatiniai pasikeitimai, degalų kainų pasikeitimo tendencija, potencialių teisių žinojimo ir saugumo įtaka, turi būti svarstomi.

Turime pastebėti, kad pastangos sutrumpinti šiuos uždelsimus gali būti mažesnio tikslumo priežastimi, vadinasi gali būti nenaudinga; ieškinių duomenys gali turėti daugiau netikslumų negu tuo atveju, kai turima daugiau laiko ieškiniui sutvarkyti. Šis pavyzdys gali būti lengvai pritaikytas kitoms bendrojo draudimo klasėms.

## PASIRAŠYTASIS CIKLAS

Cummins ir Outreville (1987) bei Venezian (1985) pastebėjo, jog negyvybės draudimo kompanijų ( Šiaurės Amerikoje jos vadinamos "nuosavybės - nelaimingų atsitikimų" kompanijomis) pelnuose egzistuoja ciklas su maždaug šešių metų periodu.

Kaip pavyzdį galime nagrinėti Roth (1984) sudarytą lentelę (1.2.1 lentelė), kurioje pateiktas JAV "nuosavybės - nelaimingų atsitikimų" kompanijų bendrasis koeficientas. Bendras koeficientas didesnis už 100 parodo pasirašytus nuostolius. Iš šių skaičių pastebime ciklo formą ir apytikslį periodo ilgį.

### 1.2.1 lentelė

**JAV “nuosavybės-nelaimingų atsitikimų“ kompanijų pasirašytasis ciklas (pavaizduota Roth 1984)**

Metai	Nuostolių koeficientas <sup>a</sup>	Išlaidų koeficientas <sup>b</sup>	Bendras koeficientas <sup>c</sup>
1955	58,2	36,7	94,9
1956	63,2	37,1	100,5
1957	66,2	36,7	102,9
1958	63,7	36,3	100,0
1959	62,5	35,3	97,8
1960	63,6	34,8	98,4
1961	64,4	35,0	99,4
1962	64,5	34,5	99,0
1963	66,3	34,7	101,0
1964	68,0	33,9	101,9
1965	69,2	32,7	101,9

1966	66,2	31,9	98,1
1967	67,2	31,7	98,9
1968	68,8	31,2	100,0
1969	40,3	30,3	100,6
1970	69,7	29,6	99,3
1971	66,7	29,1	95,8
1972	66,0	29,4	95,4
1973	68,6	29,6	98,2
1974	75,3	29,7	105,0
1975	78,8	28,7	107,5
1976	74,6	27,4	102,0
1977	70,1	26,9	97,0
1978	69,0	27,6	96,6
1979	71,7	27,9	99,6
1980	73,9	28,5	102,4
1981	75,5	29,4	104,9
1982	78,3	29,9	108,2

aIeškiniai, padalinti iš uždirbtų draudimo įmokų (premijų)

bIšlaidos, padalintos iš pasirašytų draudimo įmokų

cNuostolių ir išlaidų koeficientų suma

Įprastą ciklą sukelia didėjantys pelnai, kurie veda prie didėjančių galimybių, tokiu būdu sąlygodami agresyvios prekybos ir nukrypimų nuo pasirašytų standartų atsiradimus.

Ciklui paaiškinti buvo pasiūlytos kelios konkurencinės hipotezės, pavyzdžiu, Roth (1984), Cummins ir Outreville (1987), Rantala (1988) ir Daykin (1994) pateikė labiau detalizuotas apžvalgas. Kelios iš šių hipotezių priklauso nuo uždelsimų ir nuo anksčiau išnagrinėtų prognozavimo ypatybių.

Pavyzdžiui, Venezian (1985) paaiškino, kad draudikas gali nustatyti įmokas savarankiškai, nepriklausomai nuo rinkos. Jis siūlo ryšį tarp draudiko elgesio ir ciklo, kuris grindžiamas prognozavimu. Venezian aiškina, kad įmokų tarifai, (bent jau tie, kurie praktikuojami JAV), pasikliauja praeities ieškinių išlaidomis tam, kad pranašautų ateities ieškinių išlaidas.

Venezian pateikia teorinę analizę parodančią, kad tokio tipo ekstrapoliacijos procedūra gali sugeneruoti pelnų ciklą. Jis vadovaujasi empiriniais bandymais, atliktais JAV pasirašytų pelnų duomenims ir daro išvadą, kad jie seka iš antros eilės autoregresinio proceso su šešių metų periodu ciklu. Tokiu būdu priemonė atsakinga už pasirašyto ir apdrausto ciklo kūrimą, pritaikant paprastas prognozavimo 'procedūras' (Cummins ir Outreville, 1987).

Ši hipotezė išreiškia draudikų dalinį iracionalumo laipsnį: praeities netekimų tendencija mechaniškai pernešama į ateitį, o kitokia potencialiai reikalinga informacija nustumama į antrą eilę arba ignoruojama. Panašios hipotezės buvo pasiūlytos Balzer ir Benjamin (1980), Balzer (1982) ir Taylor (1987).

Paaškinsime ciklus, priklausančius nuo atidėtų mokėjimų, taip vadinamų uždelsimų. Kaip buvo minėta anksčiau, daugelyje šalių dabartis negyvybės draudimo rinkoje yra:

- ? duomenų rinkimo delsimas;
- ? reguliavimo delsimas;
- ? įmokų atnaujinimo delsimas.

Šie uždelsimai atsiranda iš sutarties ir informacinių savybių, draudiminių sutarčių, galinčių užkirsti kelią draudimo įmokoms, vadinasi ir pelnui.

Turime pastebėti, kad draudimo kompanijų finansinių ataskaitų rezultatai, tokie kaip įmokos ir ieškiniai, paimti iš skirtingų laikotarpių gali būti suvidurkinti. Tai gali padidinti koreliaciją, kuri jau gali būti tarp įmokų ir ieškinių skaičių.

Įmokų skaičiai finansiniams ataskaitiniams metams yra pagrįsti procentiniais skaičiavimo metodais, todėl įmokos yra tos, kurios uždribamos draudiko, teikiant draudimo apimtį per metus. Bet įmokos uždribtos per metus įtraukia ir sutarčių, išleistų dviejų metų laikotarpiu (skaičiuojant nuo pirmos ankstesnių metų dienos iki paskutinės esamų ataskaitinių metų dienos), įmokas. Geriausiai šios įmokos aprašytų tik tą informaciją, kurią galima gauti sutarčių išleidimo momentui. Dėl uždelsimų, aprašytų anksčiau, informacijos turinys retai atspindi dabartį. Tuomet gali atsirasti maišatis tarp įmokų ir ieškinių informacijų, nagrinėjamų tuo pačiu ataskaitiniu laikotarpiu (Cummins ir Outreville, 1987).

Kitos pasiūlytos prielaidos apima draudimo verslo eksogeninius faktorius, pavyzdžiui, grindžiamus nenumatyta infliacija ir jos poveikiu ieškinių dydžiui ir su tuo susijusioms išlaidoms; nenumatytais kapitalo srautų padidėjimais; rinkos spaudimu ir konkurencijos laipsniu, bei tarpininkų vaidmeniu rinkoje (Rantala, 1988).

## ATIDĖTŲ MOKĖJIMŲ LAIKO ATŽVILGIU MODELIS

Dabar pereisime prie detalesnio modelio nagrinėjimo tam, kad paaškindume pasirašytų-apdraustų ciklą egzistavimą. Mes eliminuosime išlaidas, mokesčius, investicijų pajamas, sąveiką su šalies rinkos nustatytais įmokos tarifais. Pagrindinė prielaida tokia, kad ciklo dinamika išplaukia iš grįžtamojo ryšio pelno pertekliaus (arba rezervų) su laiko uždelsimais.

Antros eilės regresinė lygtis sugeneruos šešių metų ciklą su tam tikromis parametru reikšmėmis, kaip Venezian (1985) analizėje. Mes nagrinėsime paprastą modelį (žr. Berger, 1988; Daykin, 1994). Jo sudarymui reikia įsivaizduoti du vienerių metų uždelsimo struktūras versle.

Berger (1988) pasiūlytos šios sąlygos:

1. draudikas nustato savo pasirašytą sutartį ateinantiems metams metų pabaigos pertekliaus (arba rezervų) pagrindu, taip, kad kuo finansiškai patikimesnis draudikas, tuo labiau jis bus pasiruošęs pasirašyti kraštutinę riziką;
2. pelno ir netekimų rezultatai išplaukia iš pasirašytos sutarties su vienerių metų uždelsimu;
3. pelno ir netekimų rezultatai tiesiogiai pereina į perteklių (arba rezervus) taip, kad jų skirstymo tarp sutarčių savininkų arba akcininkų;
4. investicijų pajamų veiksmas, išlaidos ir kiti pinigų srautai gali būti ignoruojami.

Tarkime, kad  $P_t, Q_t, S_t$  ir  $p_t$  reiškia draudimo įmoką, kiekį, rezervus (arba perteklių) ir ekonominį pelną metams  $t$  atitinkamai.

Esant 1. sąlygai, mes tariam, kad rinkos perteklius ar kiekis yra pertekliaus funkcija tiesiog ankstesniam periodui (nes, kaip parodė Berger (1988), draudikai labiau pasiruošę pasirašyti įmokas su ribine rizika, kai perteklius padidintas).

Tokiu būdu, sekanti rinkos įmoka ir kiekis taip pat priklausys nuo ankstesnio laikotarpio (kadangi pertekliaus funkcijos padėtis nusakys pertekliaus ir paklausos funkcijos perskirstymą). Taigi  $(P_t, Q_t) = f(S_{t-1})$  kažkokiam  $f$ .

Mes taip pat priimame (iš 2. sąlygos), kad pelnas per metus  $t$  yra įmokos funkcija ir kiekis  $t-1$  metais, t.y.

$$p_t = g(P_{t-1}, Q_{t-1}) = gf(S_{t-2}) = h(S_{t-2})$$

tam tikroms funkcijoms  $g$  ir  $h$ .

Iš 3. sąlygos turime

$$S_t = p_t + S_{t-1} \quad (1.3.1)$$

pagal apibrėžimą,

$$p_t = h(S_{t-2}) = h(S_{t-3} + p_{t-2}) \quad (1.3.2)$$

Jeigu  $h$  atvirkštinis, mes turime antros eilės diferencialinę (skirtuminę) lygtį, galbūt netiesinę,  $p$  atžvilgiu:

$$p_t = h(h^{-1}(p_{t-1}) + p_{t-2}) \quad (1.3.3)$$

Kai  $h$  tiesinis,  $h(S) = aS + b$  ir  $h^{-1}(p) = (p - b)/a$ , tai mes turime

$$p_t = p_{t-1} + ap_{t-2} \quad (1.3.4)$$

kuri yra homogeninė skirtuminė lygtis su sprendiniu  $p_t = k_1 c_1^t + k_2 c_2^t$ , kur  $c_1$  ir  $c_2$  yra lygties  $c^2 = c + a$  sprendiniai, ir  $k_1$  ir  $k_2$  priklauso nuo pradinių sąlygų  $p_0$  ir  $p_{-1}$ . Tokiu būdu,

$$p_0 = k_1 + k_2 \quad \text{ir} \quad p_{-1} = \frac{k_1}{c_1} + \frac{k_2}{c_2}$$

Jei  $1 + 4a < 0$  (t.y.  $a < -1/4$ ),  $c_1$  ir  $c_2$  kompleksiniai jungtiniai,  $c_1 = re^{i\theta}$ ,  $c_2 = re^{-i\theta}$ , kur  $r^2 = -a$  ir  $2r \cos\theta = 1$ , todėl  $\cos\theta = 1/2\sqrt{-a}$ .

Tada

$$\begin{aligned} p_t &= k_1 c_1^t + k_2 c_2^t = \\ &= r^t [(k_1 + k_2) \cos t\theta + i(k_1 - k_2) \sin t\theta] = \\ &= r^t \left[ p_0 \cos t\theta + \left( \frac{2ap_{-1} + p_0}{2r \sin\theta} \right) \sin t\theta \right] \end{aligned}$$

panaikinant  $k_1$  ir  $k_2$ . Taigi

$$p_t = r^t \left[ p_0 \cos t\theta + \frac{2ap_{-1} + p_0}{\sqrt{-4a-1}} \sin t\theta \right] =$$

$= \alpha r^t \sin(t\theta + \beta)$ . kur  $\alpha$  ir  $\beta$  konstantos, kurios priklauso nuo  $a$  ir pradinių sąlygų.

Sinuso funkcija turi savybę, kad  $\sin(t_0\theta + \beta) = \sin(t_0\theta + 2\pi + \beta)$  kažkokiam  $t_0$ , kuris ekvivalentus ciklui (periodui). Tam,

kad galima būtų nustatyti ciklo  $T$  periodą reikalaujam kad,  $(t_0 + T)\theta + \beta = t_0\theta + 2\pi + \beta$ , t.y.  $T = \frac{2\pi}{\theta}$ .

Tokiu būdu,  $P_t$  išplaukia iš ciklo su periodu

$$T = \frac{2\pi}{\theta} = \frac{2\pi}{\cos^{-1}\left(\frac{1}{2\sqrt{-a}}\right)}.$$

Jei  $a = -1$ , tai  $T=6$  metams. Taip pat, jei  $a=-1$ ,  $p_t + p_{t-3} = 0$  atitinka šešių metų ciklą.

Pastebime, kad  $r$  dydis apibrėžia ar ciklai turi didėjančią ( $r>1$ ), mažėjančią ( $r<1$ ), ar pastovią ( $r=1$ ) amplitudę.

## ATIDĖTŲ MOKĖJIMŲ DĖL TEISMINIŲ PROCESŲ MODELIS

Šioje dalyje mes nagrinėsime atidėtų mokėjų dėl teisminių procesų modelį. Šis modelis užsienyje yra labai plačiai taikomas, tačiau mūsų valstybėje jį būtų labai sunku pritaikyti, dėl tokių priežasčių kaip įstatymai, teismo procesų trukmės ir pan.

Kaip mes pastebėjome **1.1.** dalyje esminė draudimo sistemos ypatybė yra uždelsimas dėl teisminių ieškinių. Kai kuriose biznio klasėse - 25 % visų patirtų teisminių ieškinių gali būti nepranešta arba neapmokėta po dviejų metų, tuo metu situacija gali būti labai pasikeitusi; tai įmanoma tokiose biznio klasėse kaip atsakomybės (skolos, įsipareigojimų) draudimas.

Su tais pačiais pažymėjimais kaip dalyje **1.3.** tarsime, kad  $C_t$  yra patirtas teisminis ieškinys metams  $t$ , ir  $1-e$  yra proporcija absorbuota išlaidomis draudimo įmokoje, taip, kad:

$$p_t = eP_t - C_t \quad (1.4.1)$$

Tam, kad nustatytume  $P_t$  nagrinėsime rezultatus “lauk ir stebėk” strategijos. “Lauk ir stebėk” arba laiko uždelsimo strategija galėtų būti lygiareikšmė, tam, kad per  $L$  laiko periodą po draudimo įmokos sumokėjimo, sukauptas likutis (dar vadinamas rezervų likutis) yra nepatikimas ir todėl per  $t$  metų vertė iš  $t-L$  metų turėtų būti panaudota pelno-dalybos “grįžtamojo ryšio” formulėje,

$$P_t = B_t - dS_{t-L} \quad 0 < d < 1 \quad (1.4.2)$$

čia  $B_t$  – teisminių ieškinių srautas;

$d$  – konstanta.

taigi iš lygčių (1.4.1) ir (1.4.2) mes turime

$$p_t = kB_t - C_t - deS_{t-L} \quad \text{ir iš lygties (1.3.1) tada:}$$



$$S_t - S_{t-L} + deS_{t-L} = A_t \quad (1.4.3)$$

kur  $A_t = eB_t - C_t$

Mes pastebime, kad jei sprendinys  $S_t$  konverguos, kai  $t \rightarrow \infty$  į ypatingą vertę  $S$ , tai ir  $A_t$  taip pat konverguos į  $A$ .

Jei reikalaujame, kad  $S > 0$  tada galime tiesiog pareikalauti, kad  $A > 0$  (arba  $eB > C$ ). Jei seka konverguoja ilgam laikotarpiui, tai konverguos į teigiamą (ir nebaigtinę) vertę. Akivaizdu, kad tai konvergavimo negarantuoja. Mes pastebėsime, kad kai  $A = 0$ , tai  $S = 0$  jei konvergavimas įvyktų.

Pagal Balzer ir Bejamin (1980) nagrinėsime keletą specifinių  $L$  variantų.

Jei  $L=1$ , tai lygtis (1.4.3) tampa

$$S_t - mS_{t-1} = A_t \quad \text{kur } m = 1 - de \quad (1.4.4)$$

Kai  $S_0 = 0$  lygtis yra:

$$S_t = \sum_{j=0}^{t-1} A_{t-j} m^j = \sum_{j=0}^{t-1} (eB_{t-j} - C_{t-j}) m^j \quad (1.4.5)$$

Jei  $L=2$ , tai lygtis (1.4.3) yra pavidalo

$$S_t - S_{t-1} + deS_{t-2} = A_t \quad (1.4.6)$$

Sprendinys pavidalu  $S_t = x^t$  sąlygos kvadratinę lygtį  $x^2 - x + q^2 = 0$ , kur  $q^2 = de$  ir turi tokias šaknis  $x_1, x_2 = 1 \pm i\sqrt{4q^2 - 1}/2$  žinant, kad  $4q^2 > 1$  patogumui perrašysime  $x_1, x_2$ , kai  $x_1 = qe^{i\varphi}$  ir  $x_2 = qe^{-i\varphi}$ , kur  $\tan \varphi = \sqrt{4q^2 - 1}$ . Tada su  $S_0 = 0$  lygtis (1.4.6), po keleto algebrinių supaprastinimų įgyja pavidalą:

$$S_t = \sum_{j=0}^{t-1} A_{t-j} q^j (\sin(j+1)\varphi / \sin \varphi) \quad (1.4.7)$$

Tipinės  $e$  ir  $d$  reikšmės gali būti 0,8 ir 0,5, taigi  $m=0.6$   $q=0.6325$  ir  $\varphi = 0.659$ . Tam tikras supratimas patyrimo  $C_1$  gaunamas stebint teisminių ieškinių rezervų likučių ritmą. Dėl tos pačios priežasties mes įvertiname  $B_t = 0, C_2 = C_3 = \dots = 0$  paliekant  $C_1 = x$  ne nulį ir teigiamą. Po šių sąlygų, kai  $L=1$

$$S_t = -m^{t-1} x = -(0.6)^{t-1} x$$

Kai  $L=2$

$$S_t = -q^{t-1} (\sin t\varphi / \sin \varphi) x = -(0.6325)^{t-1} (\sin 0.659 t) x / 0.612$$

kuris yra svyravimo rezultatas su periodu  $T = 2\pi / \varphi = 9.5$  metams.

Kai  $L=1$ ,  $S_t$  rezultatas patenkina atskiras, neprognozuojamas teisminių ieškinių grupes, nežiūrint trukdymo veiksnių ir apima maždaug septynis periodus. Kai uždelsimas  $L$  auga per du periodus, rezultatai tampa svyravimais ir peržengia ribą. Nebandoma kompensuoti nuostolių per du periodus. Kada yra pasibaigęs informacijos apie draudiminį įvykį surinkimas,

draudikas turėtų apmokėti dalį išmokos per artimiausius keturis periodus. Šia susidariusia situacija nėra patenkintas nei draudėjas, nei draudikas. Ši situacija yra susijusi su  $L=I$  atveju (Balzer ir Benjamin,1980). Tai yra bendras principas, kuris yra priežastimi laiko uždelsimo, grįžtamojo ryšio grandinėje.

Kai  $L=5$  skaitmeninis eksperimentavimas su lygtimi (1.4.3) rodo, kad sistema tampa visiškai nestabili, su  $S_t$  svyravimais.  $S_t$  diverguoja, kai  $t \rightarrow \infty$  ir suprantama, kad viena ar

kita pusė greičiau atsitrauktų iš susitarimo negu kentėtų šiuos svyravimus.

Į tai reikia atkreipti dėmesį, kadangi šie svarbūs rezultatai yra gana bendri ir nepriklausomi nuo draudimo tipo, ir draudimo įmokos dydžio pasirinkimo.

## GRANDINIŲ LAIPTŲ METODAS

Rezervams skaičiuoti siūloma labai daug metodų, mes nagrinėsime rezervus apskaičiuotus grandinių laiptų metodu, be infliacijos įvertinimo. Šio metodo esmė ta, kad panaudojant praeities duomenis atliekamas prognozavimas į ateitį. Gautus duomenis surašome į lentelę:

<b>Grandinių laiptų metodo pradiniai duomenys</b>								<b>1.5.1.lentelė</b>
<i>metai</i>	1	2	3	...	$k-1$	$K$	$\infty$	
1	C11	C12	C13		C1k-1	C1k		
2	C21	C22	C23		C2k-1			
3	C31	C32	C33					
...								
$k-1$	Ck-11	Ck-12						
$k$	Ck1							

$$C_{ij} = \sum_{i=1}^j c_{ij}$$

Toliau turime apskaičiuoti sukauptus mokėjimus  $C_{ij}$ , gautus duomenis surašome į lentelę:

<b>Grandinių laiptų metodo sukaupti mokėjimai</b>								<b>1.5.2.lentelė</b>
<i>metai</i>	1	2	3	...	$k-1$	$K$	$\infty$	
1	C11	C12	C13		C1k-1	C1k		
2	C21	C22	C23		C2k-1			
3	C31	C32	C33					
...								
$k-1$	Ck-11	Ck-12						
$k$	Ck1							

Kiekvieną elementą skaičiuojame pagal formulę:

$$C_{ij+1} = m_j \cdot C_{ij} \tag{1.5.1}$$

$$C_{i\infty} = M_k \cdot C_{ik} \tag{1.5.2}$$

Panaudojant šias išraiškas užpildoma apatinė lentelės dalis.

$$m_j = \frac{\sum_{i=1}^{k-j} C_{i,j+1}}{\sum_{i=1}^{k-j} C_{ij}} \quad (1.5.3)$$

Pirmiausia reikia įvertinti  $M_k$ , tai galime padaryti pasinaudodami formule

$$M_k = \frac{C_{1\infty}}{C_{1k}} \quad (1.5.4)$$

$$C_{i\infty} = C_{i,k-i+1} \cdot M_{k-i+1} \quad (1.5.5)$$

Toliau suskaičiuosime rezervus kiekvieniems metams, pasinaudodami formule

$$R_i = C_{i\infty} - C_{i,k-i+1} \quad (1.5.6)$$

$$R = \sum_{i=1}^k R_i \quad (1.5.7)$$

Belieka tik susumuoti:

## MULTIPLIKATYVINIS METODAS

Nagrinėsime rezervus apskaičiuotus multiplikatyviniu metodu, be infliacijos įvertinimo. Šio metodo esmė ta, kad panaudojant praeities duomenis atliekamas prognozavimas į ateitį. Šiame metode naudojami atidėti mokėjimai, kurie užrašomi formule:

$$c_{ij} = x_i \cdot p_j \quad (1.6.1)$$

Čia,  $c_{ij}$  - nepriklausomi atsitiktiniai dydžiai,

$x_i$  - išmokų suma per  $i$  - tuosius metus,

$p_j$  - proporcingumo koeficientas išmokos  $x_i$ , laike metų  $j$ , bei

$$1 = \sum_{j=1}^k p_j \quad (1.6.2)$$

Šie koeficientai apskaičiuojami mažiausių kvadratų metodu:

$$\sum_{(i,j)} w_{ij} \cdot (x_i \cdot p_j - c_{ij})^2 \quad (1.6.3)$$

Laikydami, kad  $w_{ij}$  - svorio koeficientas, lygus 1, suprastinus gauname:

$$\sum_{(i,j)} (x_i \cdot p_j - c_{ij})^2 = S \quad (1.6.4)$$

Koeficientus  $x_i$  ir  $p_j$  reikia parinkti taip, kad minimizuotume (1.6.4) sumą:

$$\frac{\partial S}{\partial x_i} = 0 \quad \frac{\partial S}{\partial p_j} = 0 \quad (1.6.5)$$

Taigi, reikia skaičiuoti dalines išvestines:

$$\begin{cases} \frac{\partial S}{\partial x_i} = 2 \cdot \sum_{(i,j)} (x_i \cdot p_j - c_{ij}) \cdot p_j \\ \frac{\partial S}{\partial p_j} = 2 \cdot \sum_{(i,j)} (x_i \cdot p_j - c_{ij}) \cdot x_i \end{cases} \quad (1.6.6)$$

Įsistačius pradines reikšmes gauname:

$$\begin{cases} 2 \cdot \sum_{(i,j)} (x_i \cdot p_j - c_{ij}) \cdot p_j = 0 \\ 2 \cdot \sum_{(i,j)} (x_i \cdot p_j - c_{ij}) \cdot x_i = 0 \end{cases} \quad (1.6.7)$$

Suprastinus gauname:

$$p_j = \frac{\sum_{j=1}^k c_{ij} \cdot x_i}{\sum_{j=1}^k x_i^2} \quad (1.6.8)$$

$$x_i = \frac{\sum_{j=1}^k c_{ij} \cdot p_j}{\sum_{j=1}^k p_j^2} \quad (1.6.9)$$

Kadangi akivaizdu, kad vienas dydis išsireiškia per kitą, tai pradinius priartėjimus reikia pasirinkti patiems, pagal rekurentines formules apskaičiuojami kiti dydžiai; tikimybių nulinis priartėjimas yra išskaičiuojamas pagal proporcingumą.

Tarp multiplikatyvinio ir gradinio laiptų metodo yra glaudus ryšys, nes sukaupti mokėjimai gali būti išreikšti formule (1.6.9):

$$C_{ij} = \frac{x_i}{M_j} \quad (1.6.10)$$

Čia, 
$$M_j = \frac{1}{\sum_{r=1}^i p_r}$$

Pasinaudodami formule (1.6.1), apskaičiuojame  $c_{ij}$ , gautus duomenis surašome į lentelę:

**Multiplikatyvinio metodo rezervai**

**1.6.1.lentelė**

metai	1	2	3	....	k	R <sub>i</sub>
1	c <sub>11</sub>	c <sub>12</sub>	c <sub>13</sub>		c <sub>1k</sub>	$\sum_{i=1}^k c_{1i}$
2	c <sub>21</sub>	c <sub>22</sub>	c <sub>23</sub>		c <sub>2k</sub>	$\sum_{i=1}^k c_{2i}$
3	c <sub>31</sub>	c <sub>32</sub>	c <sub>33</sub>	....	....	$\sum_{i=1}^k c_{3i}$
...			....	....	....	....
k-1	c <sub>k-11</sub>	c <sub>k-12</sub>	....	....	....	....

$$R = \sum_{i=1}^k R_i$$

Belieka tik susumuoti:

**(1.6.11)**

**ATSKYRIMO METODAS**

Šis metodas skaičiuojamas neįvertinus infliacijos. Atskyrimo metodo esmė, kad naudojant atidėtus mokėjimus, kaip pradinius duomenis, naudodami formulę (1.7.1) sudarome naują lentelę:

$$S_{ij} = p_i \cdot \lambda_{i+j-1} \quad \text{arba} \quad S_{ij} = \frac{c_{ij}}{n_i} \quad (1.7.1)$$

Čia,  $n_i$  - nuostolių skaičius per atitinkamus metus (t.y. draudiminių įvykių skaičius).

**Atskyrimo metodo duomeys**

**1.7.1.lentelė**

metai	1	2	3	....	k-1	k
1	$p_1 \hat{\lambda}_1$	$p_1 \hat{\lambda}_2$	$p_1 \hat{\lambda}_2$	....	$p_1 \hat{\lambda}_{k-1}$	$p_1 \hat{\lambda}_k$
2	$p_2 \hat{\lambda}_2$	$p_2 \hat{\lambda}_3$		....	$p_2 \hat{\lambda}_k$	
3	$p_3 \hat{\lambda}_3$	$p_3 \hat{\lambda}_4$		....		
...				....		
k	$p_1 \hat{\lambda}_k$			....		

Šis metodas įdomus, tuo, kad įvedami tarpiniai parametrai, juos žymėsime:  $v$  ir  $d$ , kurie apskaičiuojami:

$$d_r = \lambda_r \cdot \sum_{i=1}^r p_i = \sum_{l=1}^r S_{l,r-l+1} \quad (1.7.2)$$

$$v_i = \sum_{j=1}^{k-i+1} S_{ij} = p_j \cdot \sum_{i=j}^k \lambda_i \quad (1.7.3)$$

Akivaizdu, kad taikant šį metodą iš atidėtų mokėjimų lentelės pirmiausia yra paskaičiuojama lentelė  $S_{ij}$ , taikant formules (1.7.2) ir (1.7.3), apskaičiuojami tarpiniai kintamieji, kuriais remdamiesi, išsireiškiami kintamieji  $p_i$  ir  $\lambda_j$ :

$$\begin{aligned} \hat{\lambda}_k &= d_k; \quad S_{1k} = v_k; \quad \hat{p}_k = \frac{v_k}{\hat{\lambda}_k} \\ \hat{\lambda}_{k-1} &= \frac{d_{k-1}}{(1 - \hat{p}_k)}; \quad \hat{p}_{k-1} = \frac{v_{k-1}}{\hat{\lambda}_{k-1} + \hat{\lambda}_k} \end{aligned} \quad (1.7.4)$$

-----

$$\hat{\lambda}_r = \frac{d_r}{1 - \sum_{j=1}^{k-r} \hat{p}_{r+j}}; \quad \hat{p}_j = \frac{v_j}{\sum_{s=j}^k \hat{\lambda}_s}$$

Tačiau mes negalime apskaičiuoti pagal duotas formules visų  $\lambda$  reikšmių, todėl tenka likusias reikšmes apskaičiuoti panaudojant regresijos lygtį. Žinoma tiesinės regresijos lygtis:

$$\lambda = a \cdot t + b \quad (1.7.5)$$

Čia,  $a$  ir  $b$  koeficientai ieškomi mažiausių kvadratų metodu,

$$S = \sum_{i=1}^n (\lambda_i - a \cdot i - b)^2 \min_{a,b} S \quad (1.7.6)$$

$$\begin{cases} \frac{\partial S}{\partial a} = 2 \cdot \sum_{i=1}^n (\lambda_i - a \cdot i - b) \cdot (-i) \\ \frac{\partial S}{\partial b} = 2 \cdot \sum_{i=1}^n (\lambda_i - a \cdot i - b) \cdot (-1) \end{cases} \quad (1.7.7)$$

Suprastinus gauname:

$$\begin{cases} \sum_{i=1}^n \lambda_i - a \cdot \sum_{i=1}^n i - b \cdot n = 0 \\ \sum_{i=1}^n i \cdot \lambda_i - a \cdot \sum_{i=1}^n i^2 - b \sum_{i=1}^n i = 0 \end{cases} \quad (1.7.8)$$

Gavę  $a$  ir  $b$  koeficientus įsistatome (1.7.5) lygtį ir gauname likusias  $\lambda$  reikšmes. Turėdami  $\lambda$  visas reikšmes pasinaudojė

formule (1.7.1) apsiskaičiuojame  $S_{ij}$  reikšmes, ir sudarome naują lentelę. Turėdami šias reikšmes prognozuosime reikšmes ateičiai, pasinaudodami formulėmis:

$$\hat{S}_{1k^+}(S_{1\infty}) = C_{1\infty} - C_{1k} \quad (1.7.9)$$

Likusios reikšmės apskaičiuojamos pagal formulę:

$$\hat{S}_{ik^+}(S_{i\infty}) = \hat{S}_{1k^+} \cdot \frac{\hat{\lambda}_{i+k-1}}{\hat{\lambda}_k} \quad (1.7.10)$$

Pagal šias formules gautos reikšmės yra tarpinės, rezervams skaičiuoti taikomos sekančios formulės:

$$M_{k-i+1} = \frac{\sum_{j=1}^k \hat{S}_{ij} + \hat{S}_{ik^+}}{\sum_{j=1}^{k-i+1} \hat{S}_{ij}} \quad (1.7.11)$$

Turėdami visas reikalingas reikšmes rezervams skaičiuoti ir pritaikę formulę:

$$\hat{R}_i = C_{i,k-i+1} \cdot (\hat{M}_{k-i+1} - 1) \quad (1.7.12)$$

Apskaičiavus dalinius rezervus – gautos reikšmės yra sumuojamos:

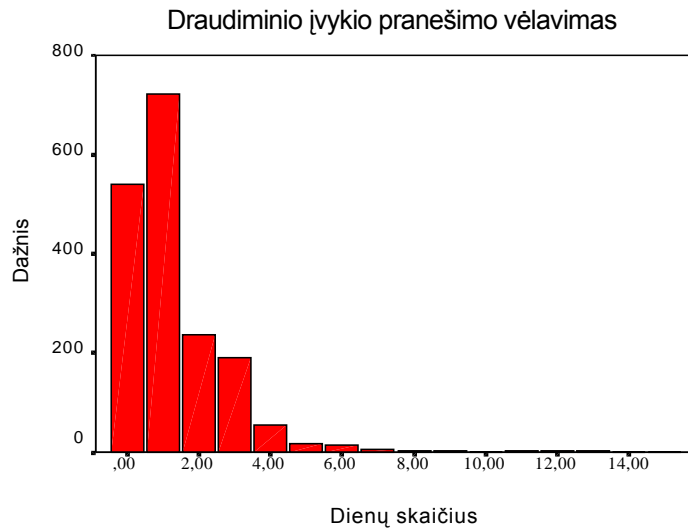
$$R = \sum_{i=1}^k \hat{R}_i \quad (1.7.13)$$

Šis metodas yra gan tikslus, esant didelei statistikai, jei yra žinomi duomenys apimantys 10 metų laikotarpį, tai geriausia iš šių: Grandinio laiptų, multiplikatyvinio ir atskyrimo metodų, taisyti pastarąjį, jis suteikia tikslius rezultatus; tačiau jei yra mažai statistinių duomenų (pvz. 3 metų statistiniai duomenys), geriausia taikyti grandinių laiptų metodą.

## 2. TIRIAMOJI DALIS

## 2.1 ATIDĖJIMŲ PRIEŽASTYS

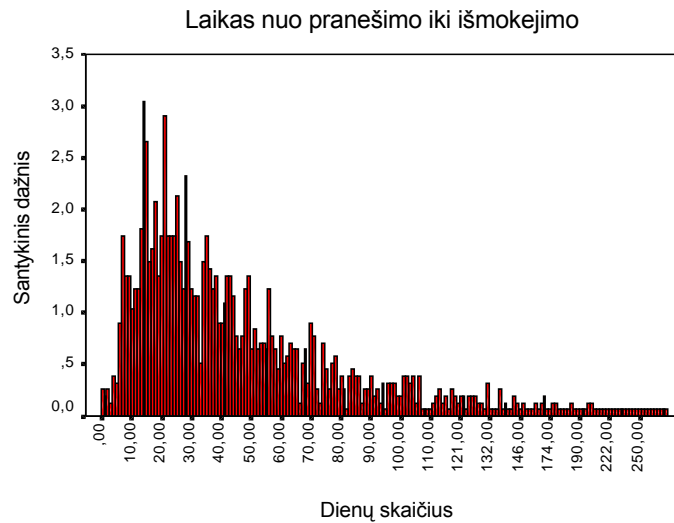
Draudimo teorijoje yra priimta, kad atsiradę nuostoliai yra atlyginami draudėjui iš kart po draudiminio įvykio. Realiame gyvenime draudiminio įvykio atsiradimo ir žalos apmokėjimo laiko momentai nesutampa. Pirmiausia, įvykus draudiminiam įvykiui, draudėjas dažniausiai ne tą pačią dieną, o tik po kelių dienų, praneša apie jį draudikui. Sužinojęs apie draudiminį įvykį, draudikas apmoka žalą ne iš karto, o tik atlikęs draudiminio įvykio tyrimą, kuris gali trukti keletą mėnesių, o kartais ir metus ar daugiau. Be to, atlikus draudiminio įvykio tyrimą, gali paaiškėti, jog draudikas neprivalo atlyginti žalos, tuomet išmokos dydis gali būti lygus ir nuliui. Pagal UAB „Verslo draudimas“ transporto draudimo duomenis pateikiame histogramas.



### 2.1.1 pav. Laikas nuo draudiminio įvykio iki pranešimo apie jį

Nagrinėjant 2.1.1 pav. matyti, kad laikas nuo draudiminio įvykio iki pranešimo apie jį dažnai yra trumpas, vidutiniškai lygus 1,5 dienos, o jo, kaip atsitiktinio dydžio, išsibarstymas apie vidurkį yra nedidelis, todėl šį laiką galime laikyti determinuotu. Iš 2.1.2 pav. pateiktų duomenų galime pastebėti, kad laikotarpis nuo pranešimo apie įvykį iki žalos apmokėjimo (jei apmokėta per kelis kartus, tai iki pirmo išmokėjimo) yra daug ilgesnis ir vidutiniškai lygus 48 dienoms, nors yra nemažai atvejų, kai jis lygus 3 mėnesiams ir daugiau.





### 2.1.2 pav. Nuostolių atlyginimo laiko santykinų dažnių histograma

Modeliuojant žalų atlyginimui reikalingą aktyvų dydį, skaičiuojant nuostolių suregulavimo techninius atidėjimus, tai reiktų įvertinti.

Mus domina ne aktyvai, o atidėtų mokėjimų įtakos rezervams įvertinimas. Tuo tikslu turi būti kaupiama statistika kiekvienoje draudimo kompanijoje. Kadangi daug realių duomenų neturime, todėl generuosime pagal tam tikrus skirstinius. Rezervams skaičiuoti siūloma labai daug metodų. Šiame darbe nagrinėjami grandininių laiptų, multiplikatyvinis ir atskyrimo metodai. Paprastumo dėlei rezervus skaičiuosime neįvertindami infliacijos ir kitų veiksnių.

## 2.2 ATIDĖTŲ MOKĖJIMŲ MODELIAVIMAS

### 2.2.1 DUOMENYS

Norint atlikti bet kokią, net ir paprasčiausią draudimo veiklos analizę, reikalingi statistiniai duomenys. Po Lietuvos nepriklausomybės atkūrimo įsisteigė nemažai privačių draudimo bendrovių, tačiau informacijos kaupimo klausimas tuo metu joms nebuvo aktualus. Tik maždaug 1997 metais dauguma Lietuvos draudimo bendrovių pradėjo daugiau dėmesio skirti kompiuterizacijai ir duomenų kaupimui. Šiuo metu analizei tinkami duomenys surinkti tik 4-6 metų laikotarpiu, todėl šiame darbe naudojami sugeneruoti duomenys, kuriuos sudaro:

- Pradiniai rezervai, Lt.
- Polisių skaičius
- Draudimo įmokos dydis, Lt.
- Atidėtų mokėjimų tikimybės
- Prognozuojami suminiai vienu metų nuostoliai
- Išmokų skaičius
- Išmokų dydis
- Modeliavimo periodų skaičius (mes ėmėme, kad vienas periodas yra vieneri metai)

- Realizacijų skaičius (reikalingas modeliavimui)

Paprastai yra analizuojami ne visi draudimo rūšies polisai kartu, o atskiros, santykinai homogeninės priimtose rizikos atžvilgiu, polisų grupės. Šios grupės dar yra vadinamos tarifų klasėmis, kadangi kiekvienai tokiai polisų grupei yra taikomi skirtingi įmokų tarifai. Naudojant tarifų klases gaunamas adekvatesnis modelis, kadangi realiaame gyvenime draudimo įmokos iš tikrųjų yra skirtingos nevienodo rizikingumo polisams. Norėdami supaprastinti modelį mes tariame, kad yra tik viena tarifo klasė. Įvertinsime tiesioginius draudimo veiklos modeliavimui reikalingus parametrus: išmokų skaičiaus skirstinį ir jo parametrus, bei išmokų dydžio skirstinį ir jo parametrus, vidutinės draudimo įmokas ir t.t.

### 2.2.1.1. IŠMOKŲ SKAIČIUS

Nagrinėjama tema yra atidėtų mokėjimų įtaka rezervams, todėl modeliuojant jiems skiriamas pagrindinis dėmesys. Be to modeliuojant yra svarbu, kad kiekvieną kartą iš kart po draudiminio įvykio, būtų teigiamas rizikos rezervas, kurio pakaktų tam tikro dydžio to draudiminio įvykio nuostolių sureguliuojimo techniniam atidėjimui sudaryti. Techninio atidėjimo dydis yra nustatomas pagal draudimo sumą ir pirminį nuostolio dydžio įvertinimą, naudojant įvairius individualios rizikos modeliuose taikomus metodus. Tačiau kartais gali atsitikti taip, kad rezervai pasidaro neigiami, vadinasi yra rezervų deficitas.

Šiame darbe skaičiuojamas išmokų skaičius per tam tikrą laiką, vadinamą modeliavimo periodu. Dažniausiai naudojamas modeliavimo periodas - vieneri metai (kalendoriniai ar ataskaitiniai). Mūsų darbe šis periodas lygus kalendoriniams metams, vadinasi polisas galioja nuo metų pradžios iki metų pabaigos. Realiaame gyvenime ne visi polisai galioja viso stebimo laikotarpio (mėnesio, ketvirčio, metų) metu. Kiekvieno poliso galiojimo pradžia gali būti bet kuriame stebėjimo laikotarpio taške ir galiojimo trukmė gali būti skirtinga. Vieni polisai gali galioti visą laikotarpį, kiti - tam tikrą jo dalį. Yra polisų, kurie galioja tik keletą dienų, kol baigiasi stebėjimo laikotarpis. Pačių polisų galiojimas nutraukiamas draudėjo iniciatyva.

Kadangi realių duomenų neturime, todėl reikia susigeneruoti išmokų skaičių. Jis bus generuojamas kaip atsitiktinis dydis pasiskirstęs pagal Puasono skirstinį (2.2.1.1.1) su parametru  $\lambda$ :

$$P(X = x) = e^{-\lambda} \cdot \frac{\lambda^x}{x!} \quad (2.2.1.1.1)$$

Yra svarbu tai, kad  $\lambda$  didėjant skirstinys normalizuojasi.

### 2.2.1.2. IŠMOKŲ DYDIS

Kiekvieną kartą įvykus draudiminiam įvykiui, draudikas atlygina draudėjo patirtus nuostolius, sumokėdamas jam tam tikro dydžio išmoką. Išmokos dydis paprastai yra nedidesnis už draudimo sutartyje nurodytą draudimo sumą ir nemažesnis už nustatytą franšizės dydį. Franšizė – tai draudėjo patirtų nuostolių suma, kurios draudikas dar neatlygina. Jei draudėjo patirto nuostolio dydis yra mažesnis už nustatytą franšizės dydį, draudimo išmoka jam nėra išmokama. Įvykus draudiminiam įvykiui, draudimo išmoka gali būti neišmokama ir dėl kitų priežasčių, pavyzdžiui, jeigu paaiškėja, kad draudėjas sudarant draudimo sutartį pateikė neteisingus duomenis apie draudimo objektą, siekė apgaulės būdu gauti draudimo išmoką ar patirtą žalą jam atlygino trečiasis asmuo, kaltas dėl padarytos žalos apdraustajam turtui. Taigi, draudimo išmoka gali būti lygi nuliui.

Išmokos dydis yra iš anksto nežinomas, t.y. atsitiktinis dydis, todėl norint jį aprašyti ir modeliuoti reikia žinoti jo skirstinį. Turime paminėti, kad šiame darbe išmokų dydis nekinta laike, nors realybėje išmokų dydis yra kintamas dėl infliacijos ar kt. priežasčių.

Pagrindinė problema, su kuria susiduriama nustatant išmokų dydžio skirstinius, yra didelių žalų įvertinimas. Paprastai, iš realių draudimo išmokų duomenų gautos empirinės pasiskirstymo funkcijos turi labai „sunkias uodegas“. Tokiems duomenims parinkti statistiškai pagrįstus skirstinius yra labai sunku, kadangi dažniausiai statistiniai metodai hipotezes apie skirstinio tipą atmets. Dėl šios priežasties labai dideli nuostolių dydžiai yra išmetami iš skirstinio įvertinimui naudojamų duomenų, kaip išskirtys ir yra nagrinėjami atskirai, naudojant kitus metodus. Kadangi mes išmokas generuojame, todėl jos yra atsitiktiniai dydžiai ir yra maža tikimybė, kad gali atsirasti išskirtys. Tariame, kad išmokų dydžių skirstiniai gali būti iš sąrašo:  $\chi^2$  (2.2.1.2.1), Erlango, eksponentinis (2.2.1.2.2), lognormalusis (2.2.1.2.3), Veibulo (2.2.1.2.4), Pareto (2.2.1.2.5) arba Burr (2.2.1.2.6).

$$\chi^2 = Y_1^2 + Y_2^2 + \dots + Y_k^2, \text{ kai } Y_i \sim N(0,1) \quad (2.2.1.2.1)$$

$$F(x) = 1 - e^{-\frac{x}{a}}, \text{ kai } E(x) = a \quad (2.2.1.2.2)$$

$$F(x) = N(z, 0, 1), \text{ kai } z = \frac{\ln x - m}{s} \quad m - \text{vidurkis, } s - \text{dispersija} \quad (2.2.1.2.3)$$

$$F(x) = 1 - e^{-(x/\theta)^\tau} \quad (2.2.1.2.4)$$

$$F(x) = 1 - \left( \frac{\theta}{x + \theta} \right)^\alpha \quad (2.2.1.2.5)$$

$$F(x) = 1 - \left( \frac{\theta}{\theta + x^\tau} \right)^\alpha \quad (2.2.1.2.6)$$

### 2.2.1.3. DRAUDIMO ĮMOKOS

Turėdami draudiminių įvykių skaičių ir išmokų dydžio skirstinius, galime apskaičiuoti vidutinės draudimo įmokos. Draudimo įmokos yra skirtos tiesioginėms draudimo išmokų mokėjimo išlaidoms padengti. Draudimo įmoka yra mažesnė už visą įmoką (bruto įmoką), kurią draudėjas turi mokėti. Bruto įmoka susideda iš trijų pagrindinių sudedamųjų dalių: grynosios rizikos įmokos, svyravimų išlyginimo priedo ir organizacinių - valdymo išlaidų priedo.

Grynoji rizikos įmoka yra apskaičiuojama sudauginant būsimų išmokų skaičiaus ir išmokų dydžio vidutinės reikšmes. Vidutinis išmokų skaičius mūsų nagrinėjamame modelyje nekinta laike, todėl ir grynoji rizikos įmoka skirtinguose laiko intervaluose yra vienoda. Atliekant ilgalaikio laikotarpio (daugiau nei 5 metų) draudimo veiklos modeliavimą, įmokų dydžio kitimas realybėje yra įmanomas, kadangi draudimo bendrovės iš tikrųjų nuolat koreguoja draudimo įmokų dydžius. Mūsų modelyje jis lieka pastovus.

## 2.2.2 MATEMATINIS MODELIS

Draudimo veiklos modeliavimui naudosime supaprastintą atidėtų mokėjimų modelį su tam tikrom prielaidom:

- Tarkime, kad nagrinėjama draudimo bendrovė tik pradėjo dirbti ir pirmasis periodas, tai draudimo bendrovės veiklos rezultatai po vienerių metų
- Visi modelyje naudojami dydžiai nekinta laike
- Modeliuojami polisai yra vienerių metų trukmės
- Polisu portfelio struktūra nekinta per visą modeliuojamą laikotarpį
- Draudimo įmokos imamos modelyje yra netto įmokos, todėl, kad neatsižvelgiam į perdraudimą, pajamas iš investicijų ir pan.
- Išmokų skaičius modelyje yra laikomas nepriklausomu atsitiktiniu dydžiu pasiskirsčiusiu pagal Puasono skirstinį su parametru  $\lambda$ .
- Išmokų dydžiai modelyje laikomi atsitiktiniais dydžiais pasiskirsčiusiais pagal lognormalųjį, eksponentinį, Veibulo, Chi-kvadratu ar Pareto skirstinius.

Matematiškai modelį galime užrašyti taip:

$$A(t) = A(t-1) + B(t) - X_p(t)$$

čia  $t$  – modeliavimo periodas (mūsų atveju tai metai);

$A(t)$  – aktyvai momentu  $t$ ;

$A(t-1)$  – Aktyvai momentu  $t-1$ ;

$B(t)$  – pasirašytos įmokos momentu  $t$  (mūsų nagrinėjam modelyje pasirašytos įmokos sutampa su uždirbtomis įmokomis);

$X_p(t)$  – apmokėtos išlaidos momentu  $t$  ir apskaičiuojamos pagal formulę:

$$X_p(t) = \sum_{u=\max(t-d_{\max}+1,1)}^t X_p(u,t)$$

čia  $t$  – modeliavimo periodas (mūsų atveju tai metai);

$d_{\max}$  – maksimalus reguliavimo periodų skaičius;

$X_p(u,t)$  – atidėti mokėjimai įvykę  $u$  metais, bet apmokėti  $t$  metais; apskaičiuojam pagal formulę:

$$X_p(u,t) = \sum_{i=1}^{N(u,t)} X_i$$

čia  $X_i$  – išmokų skaičius, kai  $N(u,t) \sim P(\lambda(u,t))$ :

$$\lambda(u,t) = p(t-u) \cdot \lambda$$

$p(t-u)$  – tikimybė, kad išmokos atsiradusios  $u$  –taisiais metais bus apmokėtos  $t$  –taisiais metais;

$\lambda$  – išmokos.

### 2.2.3 ATIDĖTŲ MOKĖJIMŲ ĮTAKOS REZERVAMS ĮVERTINIMAS

Kai jau žinome išmokų skaičiaus ir išmokų dydžio skirstinius bei jų parametrus, atidėtų mokėjimų tikimybes ir kt., galime modeliuoti ir įvertinti atidėtų mokėjimų įtaką rezervams. Rezervus mes apskaičiuosime pasinaudodami grandininių – laiptų, multiplikatyviniu, bei atskyrimo metodais. Modelio parametrus nustatysime iš gautų sugeneruotų duomenų. Modeliuosime draudimo rezervus naudodami artimus, nustatytiems draudimo bendrovių, įmokų dydžius.

### **2.2.3.1 pav. Rezervai apskaičiuoti grandininių laiptų metodu, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį**

Iš 2.2.3.1 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos. Draudimo bendrovei problema būtų iškilus, jei rezervai būtų kelis periodus iš eilės gaunami, neigiami, t.y. draudimo bendrovė nemoka apskaičiuoti rezervų reikalingų įmonės finansiniams įsipareigojimams įgyvendinti.

### **2.2.3.2 pav. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį**

Iš 2.2.3.2 paveikslo galime padaryti išvadą, kad draudimo bendrovės nustatytos įmokos yra tinkamos, rezervai apskaičiuoti teisingai, ir įmonė priėmus teisingą sprendimą, dėl reikalingo rezervams kiekio (rezervai gaunami teigiami, visais nagrinėjamais periodais).

### **2.2.3.3 pav. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį**

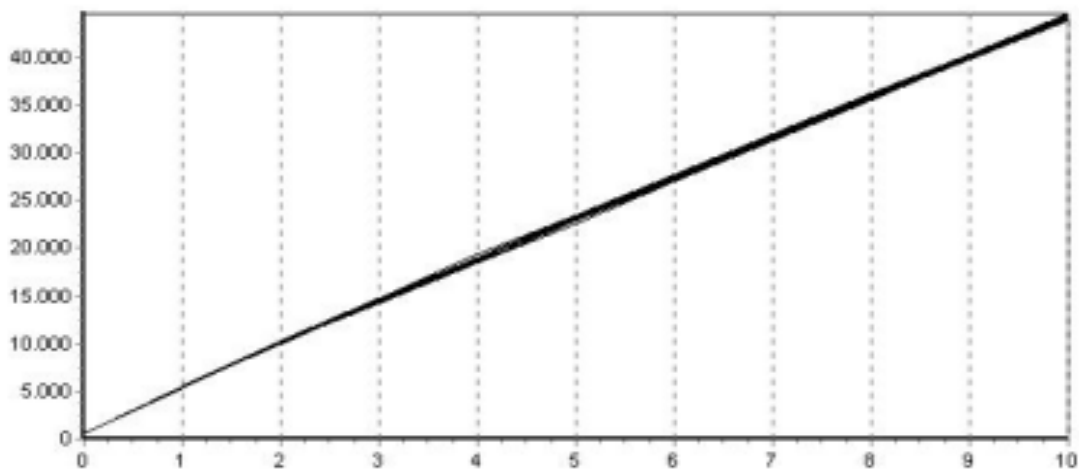
Iš 2.2.3.3 paveikslo galime padaryti išvadą, kad draudimo bendrovės nustatytos įmokos yra pakankamos, rezervai apskaičiuoti teisingai, ir įmonė priėmus teisingą sprendimą, dėl reikalingo rezervams kiekio (rezervai gaunami teigiami, visais nagrinėjamais periodais).

Nubėžiamo grafiką rezervų, rezultatų palyginimui, apskaičiuotų skirtingais metodais.

### **2.2.3.4 pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyviniu metodais, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį**

Iš 2.2.3.4 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, rezervai apskaičiuoti teisingai, metodai tinkami rezervams skaičiuoti, tačiau nereikia pamiršti, kad pats paprasčiausias yra grandininių laiptų metodas, kuris yra siūlomas naudoti, kai turime ne mažiau 5–ių metų duomenis. Iš grafiko matyti, kad didžiausi rezervai gaunami apskaičiavus atskyrimo metodu, šis metodas sudėtingiausias. Rezervai apskaičiuoti tiesioginiu būdu pagal (2.2.3.1) formulę:

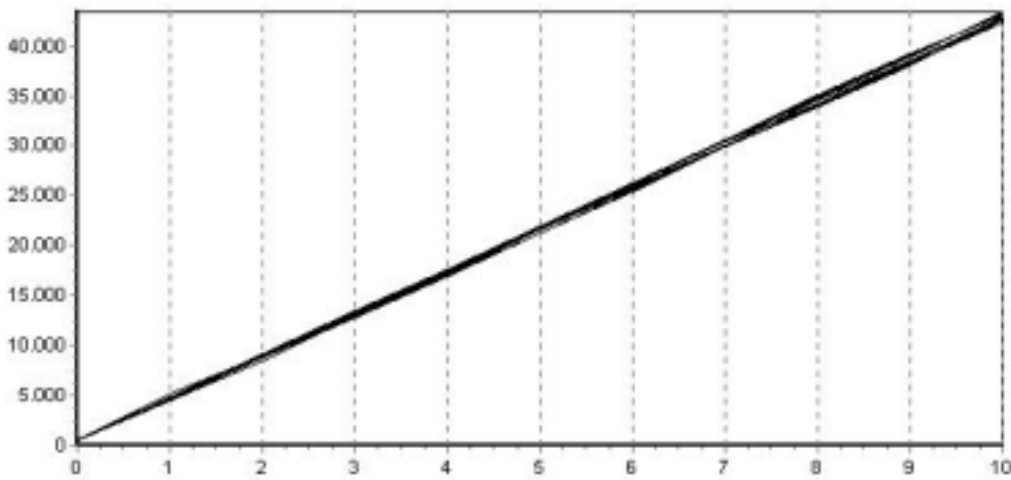
*Aktyvai = Praeito laikotarpio aktyvai + Per nagrinėjamą periodą pasirašytos draudimo įmokos – per nagrinėjamą periodą apmokėti nuostoliai;* (2.2.3.1)



### 2.2.3.5 pav. Rezervai su atidėjimais, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį

Iš 2.2.3.5 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, be to kaip matome iš grafiko rezervai didėja, tai reiškia, kad nuostoliai apmokėti per nagrinėjamą periodą arba buvo labai nedideli, arba reikalingą rezervų dydį nustatėme gerai. Šis metodas praktikoje nėra taikomas todėl, kad jis neįvertina daugelio veiksnių, kurie gali įtakoti mūsų skaičiavimus. Su tais pačiais parametrais mes sumodeliavome rezervų likutį, kai kurioje literatūroje jis yra vadinamas mokumo atsarga ar rizikos rezervais. Rezervų likutį apskaičiuojame formulės (2.2.3.2) pagalba:

$$\text{Rezervų likutis} = \text{Praeito laikotarpio rezervų likutis} + \text{polisų skaičius} * \text{įmokų dydis} - \text{per nagrinėjamą periodą patirti nuostoliai} \quad (2.2.3.2)$$



#### 2.2.3.6 pav. Rezervų likutis, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį

Iš 2.2.3.6 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, rezervų likutis didėja tai reiškia, kad patirtų nuostolių suma per nagrinėjamą periodą nėra didelė. Modeliuojant rezervų likutį mes neįvertinome nei tai, kad parametrai gali kisti laike, nei tai, kad egzistuoja tarifo klasės ir pan. Todėl modeliuojant mums nesigavo neigiamas rezervų likutis, praktikoje tai retas atvejis.

Labai svarbu atlikti analizę ir įvertinti, kuris iš nagrinėjamų metodų: grandininis laiptų atskyrimo, ar multiplikatyvinis geriausias skaičiuojant ir įvertinant rezervus, atidėtų mokėjimų įtaką rezervų skaičiavimams jau įrodėme ir padarėme išvadą, kad šie mokėjimai turi būti prognozuojami ir turi labai didelę reikšmę rezervų sumos apskaičiavimui, taigi sekantis mūsų žingsnis, išanalizuoti rezervams skaičiuoti siūlomus metodus, nes tinkamai ir teisingai nepaskaičiavus įmokų, reikalingos rezervų sumos, įmonės pelnas bus minusinis, įmonei grės bankrotas, įvairių priežiūros komisijų sankcijos, įmonė gali prarasti klientų pasitikėjimą, ir kritiniu atveju, netgi pačius klientus

Todėl norint atlikti metodų skirtų skaičiuoti rezervus analizę, nubrėžėme rezervų grafikus, kai išmokų dydis pasiskirstęs pagal eksponentinį, Veibulo, Chi – kvadratu, Pareto ir Burr skirstinius, kurie yra siūlomi įvairių aktuarijų teorijų.

Rezervai apskaičiuojami grandininio laiptų atskyrimo ir multiplikatyviniais metodais:

#### 2.2.3.7 pav. Rezervai apskaičiuoti grandininio laiptų metodu, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį

Iš 2.2.3.7 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, įmonė priėmus teisingą sprendimą, dėl reikalingo rezervams kiekio (rezervai gaunami teigiami, visais nagrinėjamais periodais).

#### **2.2.3.8 pav. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį**

Iš 2.2.3.8 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, nors kai kuriais periodais gauti neigiami rezervai, būtų iškilusi problema tuo atveju, jei draudimo bendrovei neigiami rezervai būtų gaunami labai ilgą laiką tarpą tada, bendrovei grėstų bankrotas.

#### **2.2.3.9 pav. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį**

Iš 2.2.3.9 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, reikalinga rezervų suma yra teigiama, visais nagrinėjamais periodais. Reikia atkreipti dėmesį, kad atskyrimo metodas yra pats nestabiliausias, kadangi, jo rezultatas priklauso, nuo pradinių duomenų, jei bus atliekamas modeliavimas su klaidingais duomenimis, mes negalėsime analizuoti šio metodo.

Nubėžiame rezervų grafiką, rezultatų palyginimui, apskaičiuotų skirtingais metodais.

#### **2.2.3.10 pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo ir multiplikatyviais metodais, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį**

Iš 2.2.3.10 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, nors kai kuriais periodais gauti neigiami rezervai, būtų iškilusi problema tuo atveju, jei draudimo bendrovei neigiami rezervai būtų gaunami labai ilgą laiką tarpą tada, bendrovei grėstų bankrotas. Kaip matome iš grafiko, atskyrimo metodu apskaičiuoti rezervai yra didžiausi lyginant su kitais dviem metodais.

#### **2.2.3.11 pav. Rezervai apskaičiuoti grandininių laiptų metodu, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį**

Iš 2.2.3.11 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, problema būtų iškilusi tuo atveju, jei draudimo bendrovei rezervai būtų gaunami neigiami, kelis periodus iš eilės, bendrovei grėstų bankrotas. Padarome išvadą, kad draudimo bendrovė dirba sėkmingai, teisingai išanalizavo savo finansinę padėtį ir priėmė teisingą sprendimą dėl rezervų.

#### **2.2.3.12 pav. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį**

Iš 2.2.3.12 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, problema būtų iškilusi tuo atveju, jei draudimo bendrovei rezervai būtų gaunami neigiami, kelis periodus iš eilės, bendrovei grėstų bankrotas.



### **2.2.3.13 pav. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį**

Iš 2.2.3.13 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, draudimo bendrovė dirba sėkmingai, teisingai išanalizavo savo finansinę padėtį ir priėmė teisingą sprendimą dėl rezervų.

Nubėžiame rezervų grafiką, rezultatų palyginimui, apskaičiuotų skirtingais metodais.

### **2.2.3.14 pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo ir multiplikatyviniais metodais, kai išmokų dydis pasiskirstęs pagal Chi – kvadratu skirstinį**

Iš 2.2.3.14 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, draudimo bendrovė dirba sėkmingai, teisingai išanalizavo savo finansinę padėtį ir priėmė teisingą sprendimą dėl rezervų. Kaip matome iš paveikslo didžiausios rezervų sumos gaunamos skaičiuojant atsiskyrimo metodu, mažiausios grandininių laiptų metodu, kuris yra siūlomas, naudoti turint ne mažiau 5-ių metų duomenis. Multiplikatyviniu ir grandininių laiptų metodais apskaičiuoti rezervų reikšmės yra artimos.

### **2.2.3.15 pav. Rezervai apskaičiuoti grandininių laiptų metodu, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį**

Iš 2.2.3.15 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, problema iškiltų jie neigiamos rezervų sumos, būtų gaunamos kelis periodus iš eilės, bendrovei grėstų bankrotas, nes ji neteisingai paskaičiavus rezervus, dėl ko, ji neturėtų laisvų piniginių lėšų, savo įsipareigojimams apmokėti.

### **2.2.3.16 pav. Rezervai apskaičiuoti multiplikatyviniu metodu, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį**

Iš 2.2.3.16 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, problema iškiltų jie neigiamos rezervų sumos, būtų gaunamos kelis periodus iš eilės, bendrovei grėstų bankrotas, nes ji neteisingai paskaičiavus rezervus, dėl ko, ji neturėtų laisvų piniginių lėšų, savo įsipareigojimams apmokėti.

### **2.2.3.17 pav. Rezervai apskaičiuoti atskyrimo metodu, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį**

Iš 2.2.3.17 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, problema iškiltų jie neigiamos rezervų sumos, būtų gaunamos kelis periodus iš eilės, bendrovei grėstų bankrotas, nes ji neteisingai paskaičiavus rezervus, dėl ko, ji neturėtų laisvų piniginių lėšų, savo įsipareigojimams apmokėti.

Nubėžiame rezervų grafiką, rezultatų palyginimui, apskaičiuotų skirtingais metodais.

### **2.2.3.18 pav. Rezervai apskaičiuoti grandininių laiptų, multiplikatyviniu ir atskyrimo metodais, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį**

Iš **2.2.3.18** paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, problema iškiltų jie neigiamos rezervų sumos, būtų ganamos kelis periodus iš eilės, bendrovei gręstų bankrotas, nes ji neteisingai paskaičiavus rezervus, dėl ko, ji neturėtų laisvų piniginių lėšų, savo išpareigojimams apmokėti. Rezervų reikšmės, apskaičiuotos skirtingais metodais yra artimos.

Draudimo priežiūros tarnybos prie Finansų ministerijos valdyba rezervams skaičiuoti siūlo grandininį laiptų metodą, turint ne mažiau kaip 5 – kerių metų duomenis. Galime padaryti išvadą, kad duomenų pakanka ir visi trys metodai tinkami rezervams skaičiuoti, reikia neužmiršti, kad iš nagrinėjamų metodų sudėtingiausias yra atskyrimo, dėl prognozavimo; neteisingai atlikus prognozavimą, neišvengiamos klaidos priimant sprendimus susijusius su rezervų sumos pasirinkimu.

Iš paveikslų pateiktų priede (*žr. 1 priedas*) matome, jog aktyvai, bei rezervų likutis neigiami nesigauna; atlikus analizę su įvairiais pradiniais duomenimis, padarėme išvadą, esant skirtingiems pradiniais duomenims, skirtingais metodais (grandininį laiptų ir multiplikatyviniu), apskaičiuoti rezervai, gaunami artimi. Atskyrimo metodu apskaičiuoti rezervai gaunami, didesni, nei prieš tai paminėtų dviejų metodų, labai didelę reikšmę, šio metodo rezultatams turi pradinių duomenų parinkimas, jei duomenys parenkami neteisingai, gauto rezultato negalime, analizuoti, priimdami sprendimo dėl metodo tinkamumo, ir įmokų, bei rezervų sumos, pasirinkimo.

## 2.2.4 NUOSTOLIŲ IR IŠLAIDŲ KOEFICIENTŲ ĮVERTINIMAS

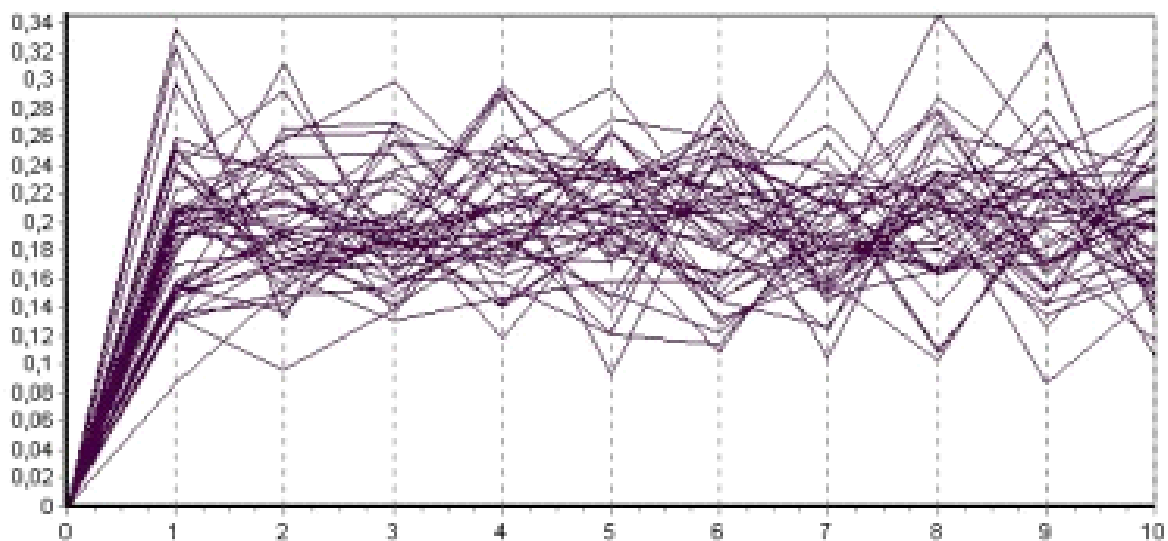
Nuostolių ir išlaidų koeficientai nustato, koks pasirašytasis ciklas yra draudimo kompanijoje. Šie koeficientai yra apskaičiuojami formulėmis **(2.2.4.1)** ir **(2.2.4.2)** :

$$\text{Nuostolių koeficientas} = \text{patirti nuostoliai} / (\text{įmokų dydis} \times \text{polisų skaičius}) \quad \text{(2.2.4.1)}$$

$$\text{Išlaidų koeficientas} = \text{apmokėti nuostoliai} / (\text{įmokų dydis} \times \text{polisų skaičius}) \quad \text{(2.2.4.2)}$$

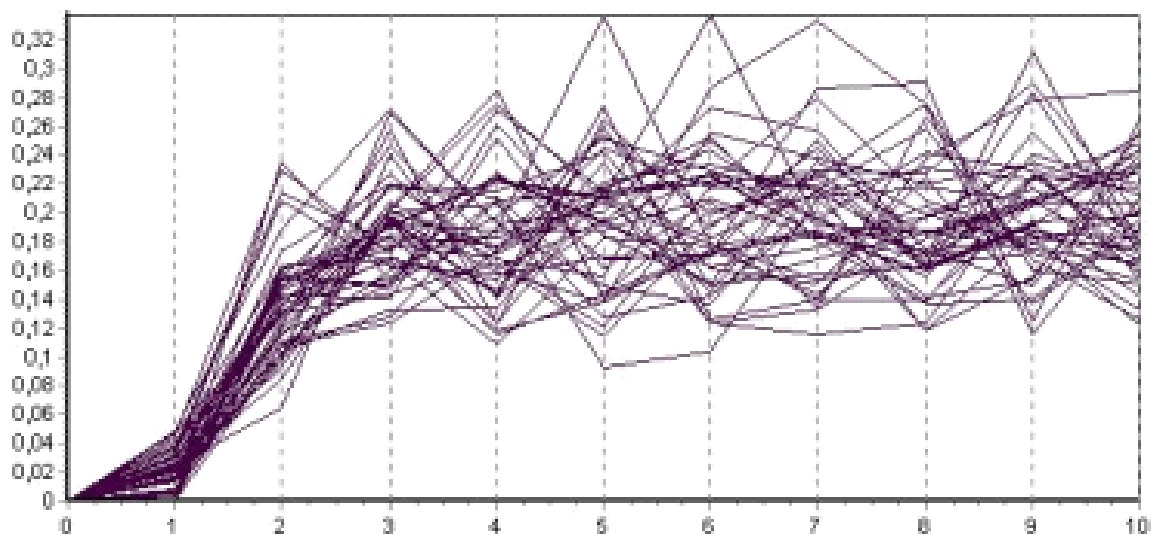
$$\text{Bendras koeficientas} = \text{Nuostolių koeficientas} + \text{Išlaidų koeficientas} \quad \text{(2.2.4.3)}$$

Būtent bendras koeficientas **(2.2.4.3)** pasako, koks yra draudimo kompanijos pasirašytasis ciklas. Mes sumodeliavome nuostolių ir išlaidų koeficientus su generuotais dydžiais ir tokiais dydžiais, kaip įmokos dydis, polisų skaičius ir pan. artimais realiems dydžiams. Gauti rezultatai:



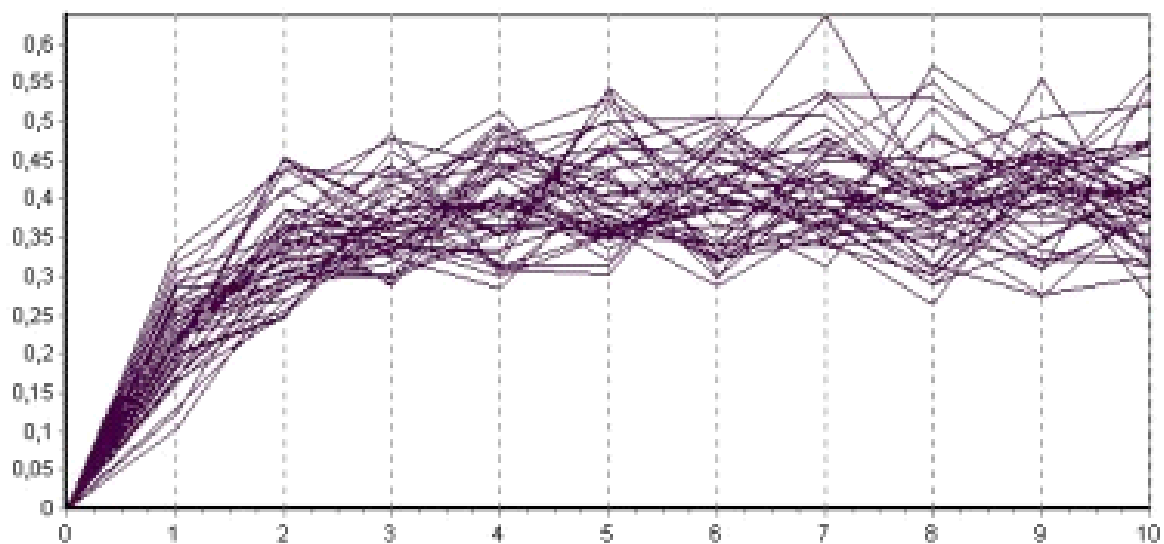
#### 2.2.4.1 pav. Nuostolių koeficientas

Iš 2.2.4.1 paveikslo matome, kad nuostolių koeficientas yra mažas tai gerai, nes reiškia, kad patirti nuostoliai yra nedideli, lyginant su uždirbtomis draudimo įmokomis. Vidutinis nuostolių koeficientas per visą modeliavimo periodą yra lygus vidutiniškai 0,2. Mums šis koeficientas modeliuojant gali gautis mažas ir dėl to, kad mes modeliuojame ne su realiais duomenimis, o su generuotais.



**2.2.4.2 pav. Išlaidų koeficientas**

Iš **2.2.4.2** paveikslo matome, kad išlaidų koeficientas yra mažas, tai gerai, nes reiškia, kad apmokėti nuostoliai yra nedideli, lyginant su uždirbtomis draudimo įmokomis. Vidutinis išlaidų koeficientas per visą modeliavimo periodą yra lygus vidutiniškai 0,18. Kaip matome iš paveikslo gaunam labai didelį šių koeficientų išsisklaidymą lyginant su netekimų koeficientu. Šis koeficientas modeliuojant gali gautis mažas ir dėl to, kad mes modeliuojame ne su realiais duomenimis, o su generuotais.



### 2.2.4.3 pav. Bendras koeficientas

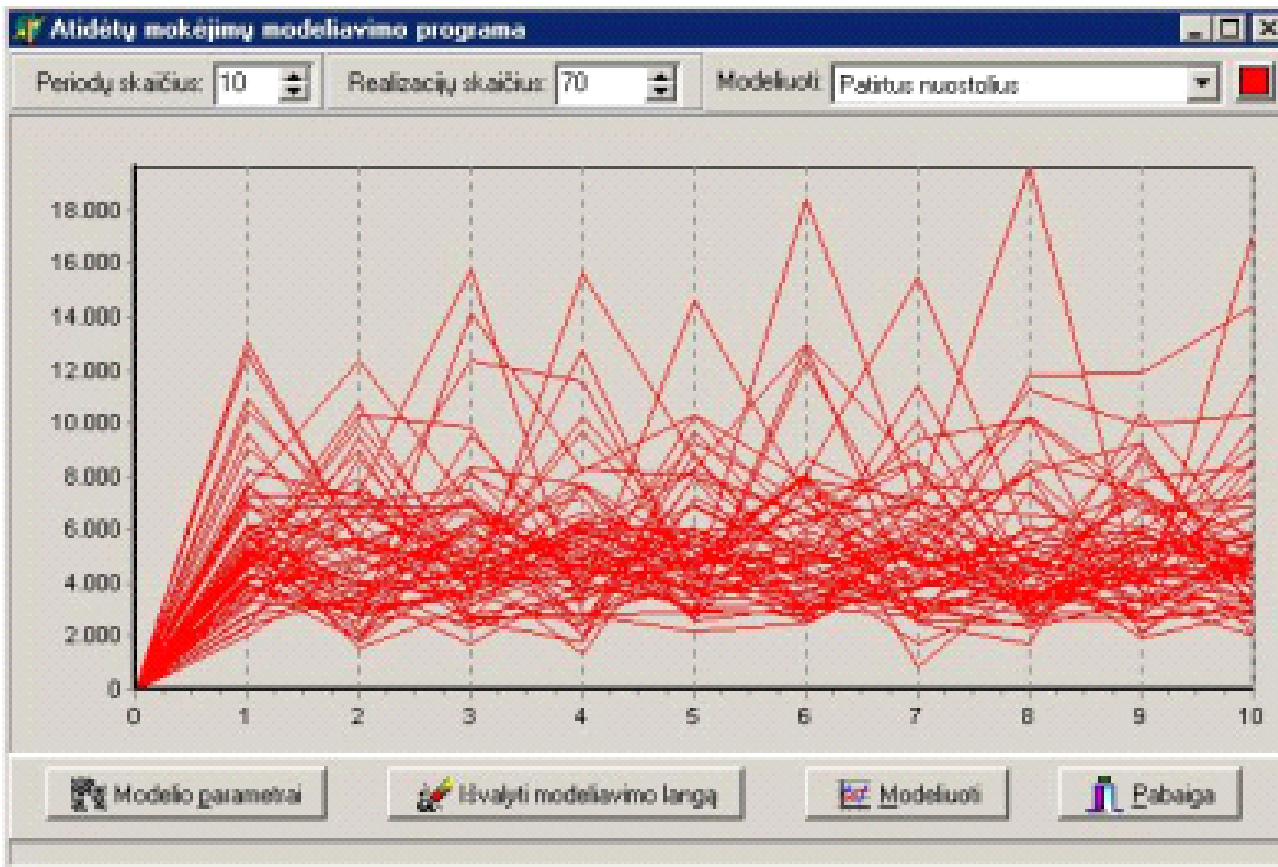
Iš 2.2.4.3 paveikslo matome, kad bendras koeficientas yra didesnis lyginant su prieš tai nagrinėtais taip yra todėl, kad šis koeficientas apskaičiuojamas kaip pastarųjų koeficientų suma. Tačiau šis koeficientas gaunamas labai mažas lyginant su tuo bendru koeficientu, kuris yra pateiktas šio darbo “Bendrojoje dalyje”, taip gaunasi dėl to, kad mes šį koeficientą modeliuojame ne pagal realius draudimo kompanijos duomenis, generuojame atsitiktinius dydžius pasiskirsčiusius pagal tam tikrus skirstinius, šis koeficientas gali gautis mažas nes mes tariame, kad nagrinėjame vieną draudimo rūšį. Bendras koeficientas labai priklauso nuo to kokius parametrus užsiduosime pačioje modeliavimo programoje; su kitais parametrais jis gali gautis dar mažesnis. Iš gautų rezultatų mes matome, jog negauname pasirašytojo ciklo.

### **3. PROGRAMINĖ REALIZACIJA IR INSTRUKCIJA VARTOTOJUI**

#### **3.1. SUKURTOS MODELIAVIMO PROGRAMOS APRAŠYMAS**

Kad būtų patogiau atlikti modeliavimą buvo sukurta draudimo įmonės atidėtų mokėjimų, rezervų įvertinimo, bei išlaidų ir nuostolių koeficientų modeliavimo programa. Nuostolių ir išlaidų koeficientai parodo pasirašytus – apdraustus ciklus. Pagrindinė programos paskirtis yra įvertinti rezervus ir palyginti rezervų kitimą priklausomai nuo to ar yra atidėti mokėjimai ar ne. Vertinant rezervus pagrindiniame programos lange galima nurodyti periodų skaičių, t.y. kiek periodų mes modeliuosime, norimą realizacijų skaičių, bei grafiką, kurį mes norime, kad mums išvestų, galima pasirinkti iš: patirtų nuostolių, apmokėtų nuostolių, aktyvų, išsipareigojimų ar rezervų apskaičiuotų grandininių laiptų, multiplikatyviu ir atskyrimo metodais. Atlikto

modeliavimo rezultatai yra išvedami pačiame pagrindiniame programos lange.



Naudojantis programa galima interaktyviai keisti modelio parametrus: polisų skaičių, atidėtų mokėjimų tikimybę, įmokų

dydį, išmokų skaičiaus ir dydžio skirstinius bei jų parametrus, užsiduoti pradinis rezervus ir t.t. Galima naudoti šiuos išmokų dydžio skirstinius:

- eksponentinį,
- Chi - kvadratu,
- lognormalųjį,
- Veibulo,
- Pareto,
- Burr.

Išmokų skaičius gali būti pasiskirstęs tik pagal Puasono skirstinį, kadangi šis skirstinys tinkamiausias, pagal įvairias aktuarijų teorijas, modeliuojant išmokų skaičių.

Taip pat galima nurodyti draudimo įmokos dydį (Lt.), keliems periodams galime užsiduoti atidėtų mokėjimų tikimybes. Visa tai galime padaryti kitame programos lange:

Programa yra universali ir gali būti naudojama įvairių draudimo rūšių atidėjimams, rezervams, bei nuostolių ir išlaidų koeficientams modeliuoti.



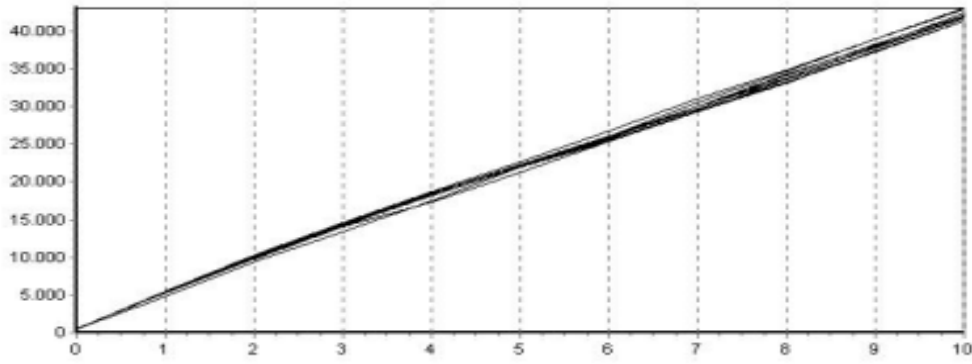
## IŠVADOS

- Sudarytas pakankamai bendras atidėtus mokėjimus aprašantis matematinis modelis, kai draudimo portfelio dydis ir išmokų parametrai (skaičius ir dydis) nekinta laike.
- Įvertinta atidėtų mokėjimų įtaka rezervams, apskaičiuotiems grandininiu laiptų, atskyrimo ir multiplikatyviniais metodais; grandinis laiptų metodas yra siūlomas Draudimo priežiūros tarnybos prie Finansų ministerijos valdybos, turint ne mažiau kaip 5 –rių metų duomenis.
- Gauti rezervų rezultatai, apskaičiuoti grandinių laiptų ir multiplikatyviniais metodais, su įvairiais nuostolių skirstiniais parodė, kad: rezervai, yra artimi, todėl iš to seka išvada, kad metodas, skirtas rezervams skaičiuoti, didelės įtakos rezervams neturi.
- Sukurta atidėtų mokėjimų modeliavimo, rezervų, rezervų likučių ir bendro koeficiento įvertinimo programa. Programa yra universali ir gali būti naudojama įvairių draudimo rūšių atidėtų mokėjimų modeliavimui, kai modelio parametrai nekinta laike. Išmokų skaičių ir dydį galima modeliuoti pasiskirsčiusius pagal įvairius skirstinius, jų parametrus galima keisti.

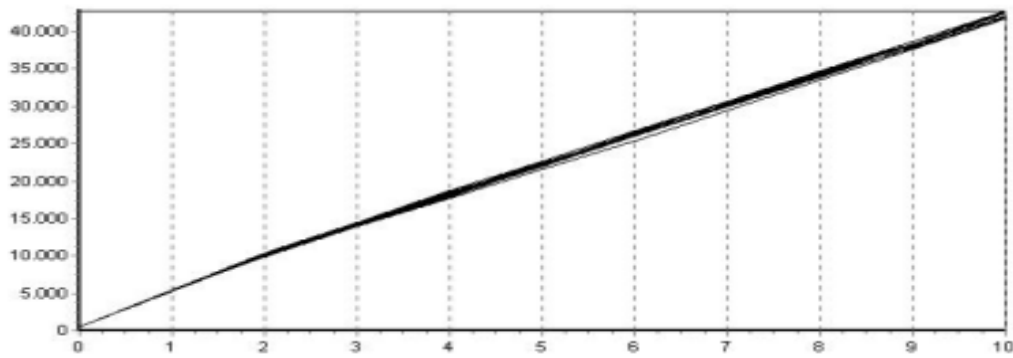
## LITERATŪRA

1. Booth P., Chadburn R. Modern actuarial theory and practice.-Chapman and Hall/CRC,1999.-716p.
2. Klugman S.A., Panjer H.H., Willmot G.E. Loss Models: from data to decisions. – New York: John Wiley & Sons, 1998.
3. Čepinskis J., Raškinis D., Stankevičius R., Šernius A. Draudimas. – Kaunas: Pasaulio lietuvių kultūros, mokslo ir švietimo centras, 1999.
4. Bepirštytė N., Simanavičienė Ž. Draudimo raida Lietuvoje. – Kaunas: Mažasis prekeivis, 1999.
5. Гофман В., Хомоненко А. Delphi 5. – Санкт-Петербург: БХВ, 2000.
6. Daykin C.D., Pentikäinen T., Pesonen M. Practical Risk Theory for Actuaries. – London: Chapman & Hall, 1994.

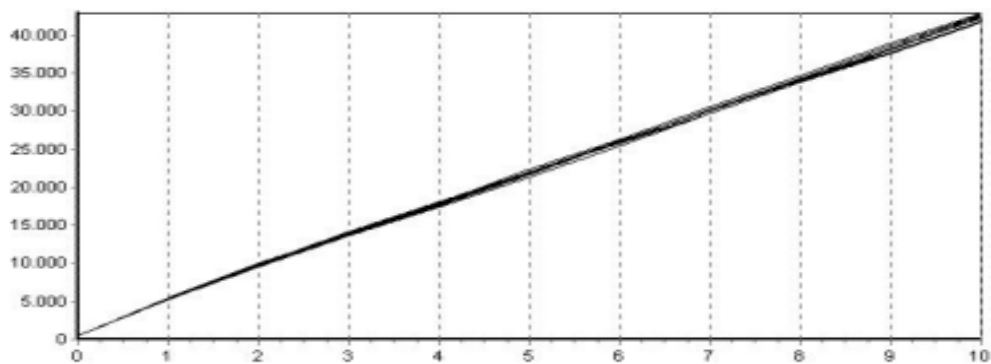
# 1 PRIEDAS. GRAFIKAI



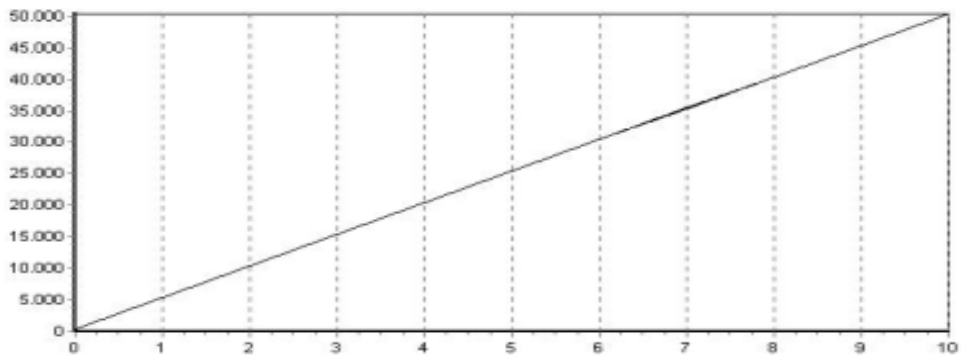
1 pav. Rezervai su atidėjimais, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį



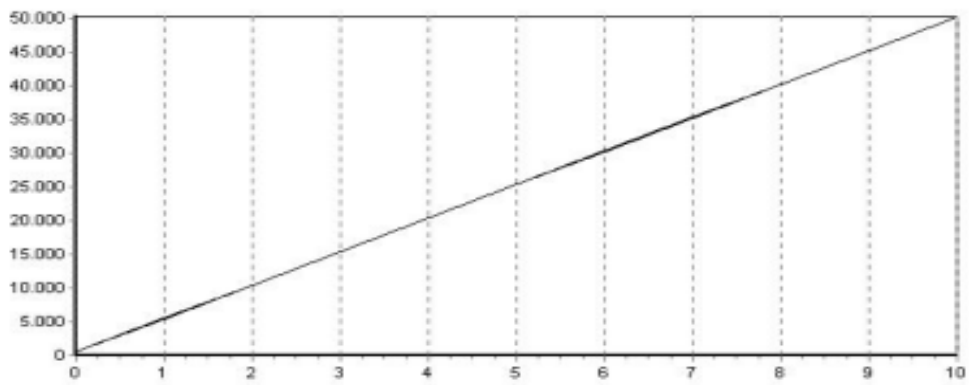
2 pav. Rezervai su atidėjimais, kai išmokų dydis pasiskirstęs pagal chi - kvadratu skirstinį



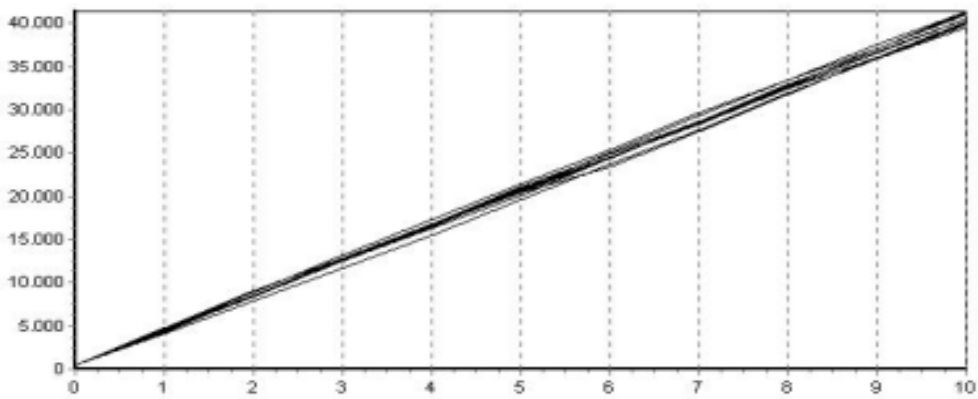
3 pav. Rezervai su atidėjimais, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį



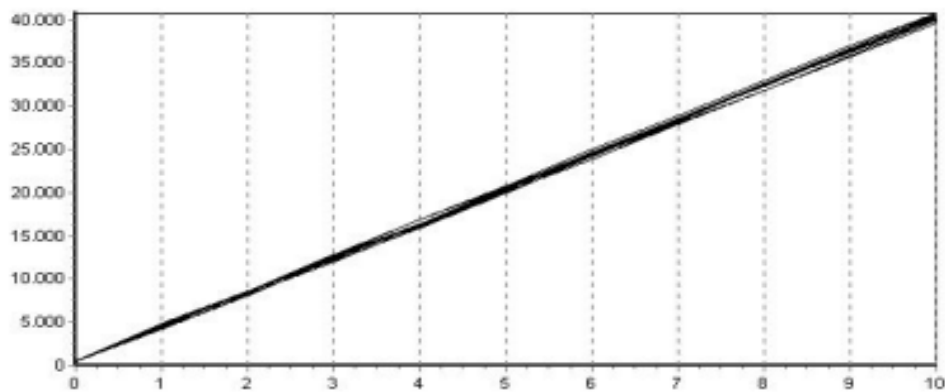
4 pav. Rezervai su atidėjimais, kai išmokų dydis pasiskirstęs pagal Pareto skirstinį



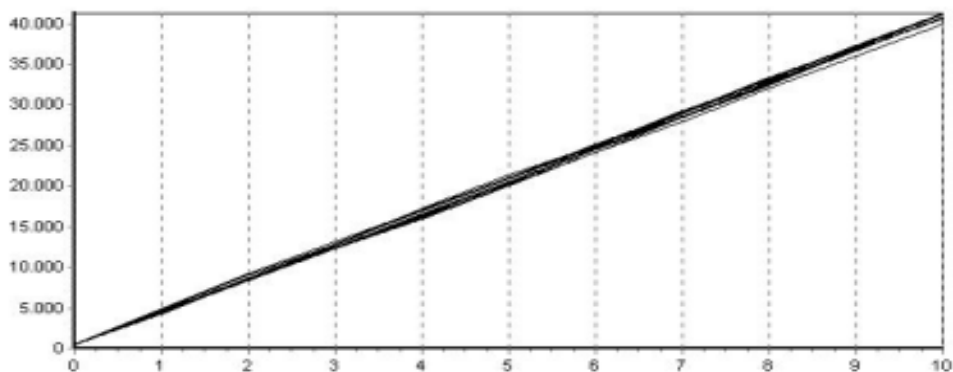
5 pav. Rezervai su atidėjimais, kai išmokų dydis pasiskirstęs pagal Burr skirstinį



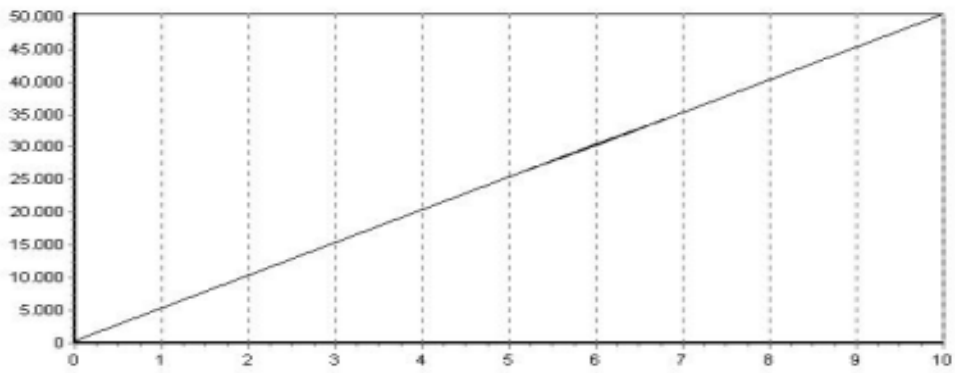
6 pav. Rezervų likutis, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį



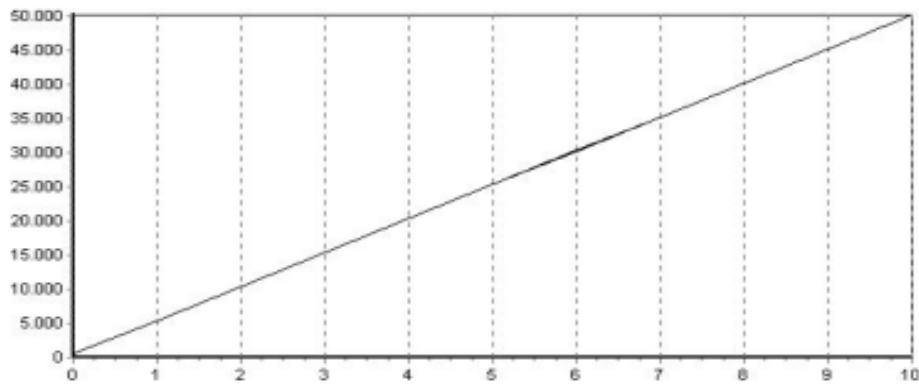
7 pav. Rezervų likutis, kai išmokų dydis pasiskirstęs pagal chi - kvadratu skirstinį



8 pav. Rezervų likutis, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį



9 pav. Rezervų likutis, kai išmokų dydis pasiskirstęs pagal Pareto skirstinį



10 pav. Rezervų likutis, kai išmokų dydis pasiskirstęs pagal Burr skirstinį

11. pav. Rezervai apskaičiuoti grandininio laiptų, atskyrimo, ir multiplikatyvinio metodais, kai išmokų dydis

**pasiskirstęs pagal eksponentinį skirstinį**

**12. pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyvinio metodais, kai išmokų dydis pasiskirstęs pagal eksponentinį skirstinį**

**13. pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyvinio metodais, kai išmokų dydis pasiskirstęs pagal lognormalųjį skirstinį**

**14. pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyvinio metodais, kai išmokų dydis pasiskirstęs pagal Chi - kvadratu skirstinį**

**15. pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyvinio metodais, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį**

**16. pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyvinio metodais, kai išmokų dydis pasiskirstęs pagal Pareto skirstinį**

**17. pav. Rezervai apskaičiuoti grandininių laiptų, atskyrimo, ir multiplikatyvinio metodais, kai išmokų dydis pasiskirstęs pagal Pareto skirstinį**



## 2 PRIEDAS. PROGRAMOS TEKSTAS

**unit Unit1;**

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, TeeProcs, TeEngine, Chart, ExtCtrls, Spin, ComCtrls, Series, Math;

type

TForm1 = class(TForm)

Panel1: TPanel;

Chart1: TChart;

Panel3: TPanel;

Label1: TLabel;

SpinEdit1: TSpinEdit;

Panel6: TPanel;

Label4: TLabel;

SpinEdit2: TSpinEdit;

Series1: TFastLineSeries;

ProgressBar1: TProgressBar;

Panel2: TPanel;

ComboBox1: TComboBox;

Label2: TLabel;

BitBtnPabaiga: TBitBtn;

BitBtnStop: TBitBtn;

BitBtnParametrai: TBitBtn;

BitBtnModeliuoti: TBitBtn;

sbSpalva: TSpeedButton;

sSpalva: TShape;

```

ColorDialog1: TColorDialog;
BitBtnIsvalyti: TBitBtn;
procedure PabaigaClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnStopClick(Sender: TObject);
procedure BitBtnModeliuotiClick(Sender: TObject);
procedure BitBtnParametraiClick(Sender: TObject);
procedure BitBtnStopClick(Sender: TObject);
procedure sbSpalvaClick(Sender: TObject);
procedure BitBtnIsvalytiClick(Sender: TObject);

```

```

private { Private declarations }
public { Public declarations }
end;

```

```

TVektorius = array of extended;
TMasyvas = array of TVektorius;

```

```

TParametrai = record

```

```

  DMax: integer;           //maksimalus atidetu mokejimu periodu skaicius
  TikMasyvas: TVektorius; //atidejimu tikimybiu masyvas
  PradRez: extended;     //pradiniai rezervai
  PolSkaicius: integer;  //polisu skaicius
  ImokDydis: extended;   //imokos dydis
  Cb: extended;          //Prognozuojamos vienu metu sumines ismokos

```

```

  IsmSkSkirst: byte;      //ismoku skaciaus skirstinys
  IsmSkPar1: extended;    //ismoku skaciaus skirstinio pirmas parametras
  NyraNorm: boolean;     //Puasono skirstinio parametras yra polisui (normuotas)

```

```

  IsmDydSkirst: byte;    //ismoku dydzio skirstinys
  IsmDydPar1: extended;  //ismoku dydzio skirstinio pirmas parametras
  IsmDydPar2: extended;  //ismoku dydzio skirstinio antras parametras
  IsmDydPar3: extended;  //ismoku dydzio skirstinio trecias parametras
  VidStd: boolean;       //Nurodyti ne parametrus, o vidurka ir dispersija (lognormaliajam)
end;

```

```

var

```

```

  Form1: TForm1;
  Param: TParametrai;
  apismokos: TMasyvas; //apmoksetos (atidetos) ismokos
  t1: integer;         //modeliavimo periodu skaicius
  RealizSk: integer;  //realizaciju skaicius
  stop: boolean;      //stabdyti modeliavima
  spalva: TColor;
  sl: integer;        //sluoksniu skaitliukas

```

```

function RandPuasono(liambda: extended): integer;
function RandChiKvadrat(df: extended): extended;
function RandEksp(Mean: extended): extended;

```

```

function RandLogN(Mean, StdDev : extended; poz: boolean):extended;
function RandVeibulo(teta,tau: extended):extended;
function RandPareto(alfa,teta: extended):extended;
function RandBurr(alfa,teta,tau: extended):extended;
function GenerIsmDyd(skirstipas: integer; par1,par2,par3: extended; poz: boolean): extended;

```

```

function atidmokejimai(u:integer; t:integer):extended;
function atidmokejimai1(t:integer):extended;
procedure apmoketosismokos(t:integer);
procedure sukauptosismokos(t:integer; apismokos:TMasyvas; var sukauptosismokos:TMasyvas);
function NaujiAktyvai(t:integer; Aktyvai: extended): extended;
function Grandininis(t:integer; sukauptosismokos:TMasyvas):extended;
function multiplikatyvinis(t:integer; atidimokejimai:TMasyvas):extended;

```

implementation

```

uses Unit3;
{$R *.DFM}

```

//-----Atsitiktiniu dydziu generavimas-----

```

function RandPuasono (liambda:extended):integer;
//Puasono skirstinio atsitiktinio skaièiaus generatorius
var ats, pk, p0, a, b, tik :extended; k:integer;
begin
ats:=Random; k:=0;
if(liambda<=0)then
begin ShowMessage('Puasono parametro reikðmë turi bûti teigiama'); Abort; end;
if(liambda>100)then p0:=0.0000001
else p0:=Exp(-liambda);
a:=0;b:=liambda;
pk:=p0; tik:=p0;
while tik<ats do
begin
k:=k+1;
pk:=(a+(b/k))*pk;
tik:=tik+pk;
end;
RandPuasono:=k;
end;

```

//-----

```

function RandPuasono1 (liambda:extended):integer;
//Puasono skirstinio atsitiktinio skaièiaus generatorius dideliems liambda
var liek,maxliambda:extended; i,sveik,randp:integer;
begin
maxliambda:=500;
randp:=0;
if liambda<maxliambda then RandPuasono1:=RandPuasono(liambda)
else
begin
sveik:=floor(liambda/maxliambda);
liek:=liambda-maxliambda*sveik;

```

```

for i:=1 to sveik do randp:=randp+RandPuasono(maxliambda);
if liek>0 then randp:=randp+RandPuasono(liek);
RandPuasono1:=randp;
end;
end;
//-----
function RandLogN(Mean, StdDev : extended; poz: boolean):extended;
//Log-normaliojo skirstinio atsitiktinio skaièiaus generatorius
//jei poz=True, tai Mean ir StdDev yra lognormaliojo a.d. vidurkis ir stand.
//nuokrypis, priešingu atveju, tai yra skirstinio parametrai (atinkamo Gauso skirstinio vidurkis ir stand. nuokrypis.)
var m,s: extended;
begin
if StdDev<=0 then
begin
ShowMessage('Neteisingai ávesti lognormaliojo skirstinio parametrai!');
Abort; end;
if poz then begin
s:=sqrt(ln(power(StdDev,2)/power(Mean,2)+1));
m:=ln(Mean)-power(s,2)/2;
RandLogN:=exp(RandG(m,s));
end
else RandLogN:=exp(RandG(Mean,StdDev));
end;
//-----
function RandEksp(Mean: extended):extended;
//Ekspontentinio skirstinio atsitiktinio skaièiaus generatorius
begin
if Mean<=0 then
begin
ShowMessage('Neteisingai ávesti eksponentinio skirstinio parametrai!');
Abort; end;
RandEksp:=-Mean*ln(1-Random);
end;
//-----
function RandVeibulo(teta,tau: extended):extended;
//Veibulo skirstinio atsitiktinio skaièiaus generatorius
begin
if (teta<=0)or(tau<=0) then
begin
ShowMessage('Neteisingai ávesti Veibulo skirstinio parametrai!');
Abort; end;
RandVeibulo:=teta*power(-ln(1-Random),1/tau);
end;
//-----
function RandPareto(alfa,teta: extended):extended;
//Pareto skirstinio atsitiktinio skaièiaus generatorius
begin
if (alfa<=0)or(teta<=0) then
begin
ShowMessage('Neteisingai ávesti Pareto skirstinio parametrai!');

```

```

Abort; end;
RandPareto:=teta*(power(1-Random,-1/alfa)-1);
end;
//-----
function RandBurr(alfa,teta,tau: extended):extended;
//Burr'o skirstinio atsitiktinio skaièiaus generatorius
var lnx: extended;
begin
if (alfa<=0)or(teta<=0)or(tau<=0) then
begin
ShowMessage('Neteisingai ávesti Burr'o skirstinio parametrai!');
Abort; end;
lnx:=(teta*(power(1-Random,-1/alfa)-1))/tau;
RandBurr:=Exp(lnx);
end;
//-----
function RandChiKvadrat(df: extended): extended;
//Chi Kvadratu atsitiktinio skiaciaus generatorius
var i: integer; x: extended;
begin
if (df<=0) then
begin
ShowMessage('Neteisingai ávesti Chi-kvadratu skirstinio parametrai!');
Abort; end;
x:=0;
for i:=1 to Round(df) do x:=x+power(RandG(0,1),2);
RandChiKvadrat:=x;
end;
//-----Skaiciavimai-----
function atidmokejimai(u:integer; t:integer):extended;
//Generuojame ismokas, atsiradusias metais u ir apmoketas metais t (vienas langelis)
var liambda1:extended; ismokuskaicius,i:integer; ismokusuma:extended; ismokudydis:extended;
begin
liambda1:=param.tikmasyvas[t-u]*param.IsmSkPar1;
if param.NyraNorm then liambda1:=liambda1*param.PolSkaicius;
ismokuskaicius:=randpuasono1(liambda1);
ismokusuma:=0;
if ismokuskaicius>0 then
for i:=1 to ismokuskaicius do
begin
ismokudydis:=GenerIsmDyd(param.IsmDydSkirst, param.IsmDydPar1, param.IsmDydPar2, param.IsmDydPar3,
param.VidStd);
ismokusuma:= ismokusuma+ismokudydis;
end;
atidmokejimai:=ismokusuma;
end;
//-----
function patirtinuostoliai():extended;
var i,ismokuskaicius:integer; liambda,ismokusuma,ismokudydis: extended;
begin

```

```

liambda:=param.IsmSkPar1;
if param.NyraNorm then liambda:=liambda*param.PolSkaicius;
ismokuskaicius:=randpuasono1(liambda);
ismokusuma:=0;
if ismokuskaicius>0 then
  for i:=1 to ismokuskaicius do
    begin
      ismokudydis:=GenerIsmDyd(param.IsmDydSkirst, param.IsmDydPar1, param.IsmDydPar2, param.IsmDydPar3,
param.VidStd);
      ismokusuma:= ismokusuma+ismokudydis;
    end;
  patirtinuostoliai:=ismokusuma;
end;
//-----
function atidmokejimai1(t:integer):extended;
//Generuojame t-taisiais metais apmoketas ismokas
var stulpelis, langelis: extended; dmax,d: integer;
begin
  dmax:=min(param.DMax, t);
  stulpelis:=0;
  for d:=0 to (dmax-1) do
    begin
      langelis:=atidmokejimai(t-d,t);
      stulpelis:=stulpelis+langelis;
    end;
  atidmokejimai1:=stulpelis;
end;
//-----
procedure atidmokejimai_mas(t:integer; var atididismokos:TMasyvas);
//Suformuoja atidetu mokejimu masyva t-tiesiems metams
var i,j,dmax:integer;
begin
  dmax:=min(param.DMax, t);
  SetLength(atididismokos, dmax);
  for i:= 0 to dmax-1 do Setlength (atididismokos[i], dmax-i);

  for j:=0 to dmax-1 do      //eilute
    begin
      for i:= 0 to dmax-1-j do      //stulpelis
        begin
          atididismokos[i,j]:=atidmokejimai(t-j,t);
        end;
      end;
    end;
end;
//-----
procedure apmoketosismokos(t:integer);
//Apmoketu ismoku masyva papildu t-taisiais metais apmoketu ismoku stulpeliu
var dmax,d: integer; langelis: extended;
begin
  dmax:=min(param.DMax, t);

```

```

SetLength(apismokos, t+1); //tam, kad t butu stulpelio numeris, nulinis stulpelis bus tuscias
SetLength(apismokos[t], dmax);
for d:= 0 to (dmax-1) do
begin
  langelis:=atidmokejimai(t-d, t);
  apismokos[t,d]:=langelis;
end;
end;
//-----
procedure sukauptismokos(t:integer; apismokos:TMasyvas; var sukauptosismokos:TMasyvas);
//Suformuoja sukauptu ismoku masyva t-tiesiems metams
var i,j,dmax:integer; suma:extended; tarpinis:TMasyvas;
begin
  dmax:=min(param.DMax, t);
  SetLength(sukauptosismokos, dmax);
  for i:= 0 to dmax-1 do Setlength (sukauptosismokos[i], dmax-i);

  SetLength(tarpinis, dmax);
  for i:= 0 to dmax-1 do Setlength (tarpinis[i], dmax-i);

  for i:= 0 to dmax-1 do      //stulpelis
    for j:= 0 to dmax-1-i do  //eilute
      begin
        tarpinis[i,j]:=apismokos[t-(dmax-1-i)+j,i];
      end;

  for j:=0 to dmax-1 do      //eilute
    begin
      suma:=0;
      for i:= 0 to dmax-1-j do  //stulpelis
        begin
          sukauptosismokos[i,j]:=suma+tarpinis[i,j];
          suma:=suma+tarpinis[i,j];
        end;
      end;
      tarpinis:=Nil;
    end;
end;
//-----
function NaujiAktyvai(t:integer; Aktyvai: extended): extended;
//Apskaiciuoja t-tojo periodo aktyvus pagal t-1 periodo aktyvus
begin
  NaujiAktyvai:=Aktyvai+param.PolSkaicius*param.ImokDydis-atidmokejimai l(t);
end;
//-----
function NaujiRizLikutis(rizlikutis: extended): extended;
//Apskaiciuoja t-tojo likuti pagal t-1 periodo likuti
begin
  Naujirizlikutis:=rizlikutis+param.PolSkaicius*param.ImokDydis-PatirtiNuostoliai();
end;
//-----

```

```

function Grandininis(t:integer; sukauptosismokos:TMasyvas):extended;
//t-tuju metu rezervu skaiciavimas Grandininiu laiptu metodu
var i,j,k,dmax:integer; Cbb,Grand:extended; Rez,Cbbb,mm,M:Tvektorius;
skaitiklis, vardiklis: extended;
begin
  dmax:=min(param.DMax, t);
  Cbb:=sukauptosismokos[dmax-1,0]+param.Cb;
  SetLength(mm,dmax);
  k:=dmax-1;
  for j:=dmax-1 to 1 do
  begin
    skaitiklis:=0;
    for i:=0 to k-j do skaitiklis:=skaitiklis+sukauptosismokos[j,i];
    vardiklis:=0;
    for i:=0 to k-j do vardiklis:= vardiklis+sukauptosismokos[j-1,i];
    if vardiklis=0 then
    begin
      MessageDlg('Klaida: taikant grandininá laiptø metodà dalybos vardiklis gautas nulis!'
        +'#13#10+Iðmokø parametrø reikðmës nurodytos per maþos.', mtWarning, [mbOk], 0);
      Abort;
    end;
    mm[j-1]:=skaitiklis/vardiklis;
  end;
  SetLength(M, dmax);
  M[k]:=Cbb/sukauptosismokos[k,0];
  for i:=k-1 to 0 do
  M[i]:=mm[i]*M[i+1];
  SetLength(Cbbb,dmax);
  for i:=0 to k do Cbbb[i]:=sukauptosismokos[k-i,i]*M[k-i];
  SetLength(Rez, dmax);
  for i:= 0 to k do Rez[i]:=Cbbb[i]-sukauptosismokos[k-i,i];
  Grand:=0;
  for i:=0 to k do Grand:=Grand+Rez[i];
  Rez:=Nil; Cbbb:=Nil; mm:=Nil; M:=Nil;
  Grandininis:=Grand;
end;
//-----
procedure xx(sukauptosismokos:TMasyvas;p:Tvektorius;k:integer; var X:Tvektorius);
var j,i:integer;
skaitiklis, vardiklis:extended;
begin
  SetLength(X,k+1);
  skaitiklis:=0;
  vardiklis:=0;
  for j:=0 to k do
  begin
    skaitiklis:=skaitiklis+sukauptosismokos[0,j]*p[j];
  end;
  skaitiklis:=skaitiklis+param.Cb*p[k+1];
  for j:=0 to k+1 do

```



```

begin
  vardiklis:=vardiklis+p[j]*p[j];
end;
X[0]:=skaitiklis/vardiklis;
for i:=1 to k do
begin
skaitiklis:=0;
vardiklis:=0;
for j:=0 to k-i do
begin
skaitiklis:=skaitiklis+sukauposismokos[i,j]*p[j];
end;
for j:=0 to k-i do
begin
vardiklis:=vardiklis+p[j]*p[j];
end;
X[i]:=skaitiklis/vardiklis;
end;
end;
//-----
procedure ppp(sukauposismokos:TMasyvas;X:Tvektorius;k:integer; var p:Tvektorius);
var j,i:integer;
skaitiklis, vardiklis:extended;
begin
SetLength(p,k+2);
for i:=0 to k do
begin
skaitiklis:=0;
vardiklis:=0;
for j:=0 to k-i do
begin
skaitiklis:=skaitiklis+sukauposismokos[j,i]*X[j];
end;
for j:=0 to k-i do
begin
vardiklis:=vardiklis+X[j]*X[j];
end;
p[i]:=skaitiklis/vardiklis;
end;
p[k+1]:=param.Cb/X[0];
end;
//-----
function multiplikatyvinis(t:integer; atidimokejimai:TMasyvas):extended;
var i,j,k,dmax:integer;
p_t,skaitiklis, vardiklis,Cb,rezervaiM:extended;
p,p1,X,m,mm,MuM,mCB,RezM:Tvektorius;
nc,C:Tmasyvas;
begin
dmax:=min(param.DMax, t);
SetLength(p,dmax+1);

```

```

SetLength(p1,dmax+2);
SetLength(X,dmax);
SetLength(m,dmax);
SetLength(mm,dmax);
SetLength(MuM,dmax);
SetLength(mCB,dmax);
SetLength(RezM,dmax);
SetLength(nc,dmax+2);
for i:= 0 to dmax+1 do Setlength(nc[i], dmax+2);

SetLength(C,dmax+2);
for i:= 0 to dmax+1 do Setlength(C[i], dmax+2);
k:=dmax-1;
//t skaiciavimas
skaitiklis:=atidimokejimai[0,0];
vardiklis:=0;
for i:=0 to k do vardiklis:=vardiklis+atidimokejimai[0,i];
p_t:=skaitiklis/vardiklis;

if (p_t<=0)or(p_t>=1)
then
begin
  MessageDlg('Klaida: klaidingi multiplikatyvinio metodo pradiniai parametrai', mtWarning, [mbOk], 0);
  Abort;
end;
//*****
for i:=1 to k do
begin
  skaitiklis:=0;
  vardiklis:=0;
  for j:=0 to k-i do skaitiklis:=atidimokejimai[j,i]+skaitiklis;
  for j:=0 to k-i do vardiklis:=atidimokejimai[j,i-1]+vardiklis;
  if vardiklis=0 then
  begin
    MessageDlg('Klaida: taikant multiplikatyvini metoda dalybos vardiklis gautas nulis!'
      +'#13#10+Iðmokø parametrø reikðmës nurodytos per maþos.', mtWarning, [mbOk], 0);
    Abort;
  end;
  p[i]:=skaitiklis/vardiklis;
end;
  vardiklis:=atidimokejimai[k,0];
  if vardiklis=0 then
  begin
    MessageDlg('Klaida: taikant multiplikatyvini metoda dalybos vardiklis gautas nulis!'
      +'#13#10+Iðmokø parametrø reikðmës nurodytos per maþos.', mtWarning, [mbOk], 0);
    end
  else p[dmax]:=param.Cb/vardiklis;

p[1]:=p[1]*p_t;
for j:=2 to k do

```

```

begin
  p[j]:=p[j]*p[j-1];
end;
p[dmax]:=p[dmax]*p[k];
skaitiklis:=0;
for j:=1 to k+1 do
  begin
    skaitiklis:=skaitiklis+p[j];
  end;
p[0]:=1-skaitiklis;
//-----
xx(atidimokejimai,p,k,X);
ppp(atidimokejimai,X,k,p1);
for i:=1 to k do
  begin
    xx(atidimokejimai,p1,k,X);
    ppp(atidimokejimai,X,k,p1);
  end;

for i:=0 to k do
  begin
    for j:=0 to k+1 do nc[i,j]:=X[i]*p1[j];
  end;

for i:=0 to k do C[i,0]:=nc[i,0];
for i:=0 to k do
  begin
    for j:=1 to k do
      begin
        if j>=Dmax-i then C[i,j]:=0
        else C[i,j]:=nc[i,j-1]+nc[i,j];
      end;
    end;
end;
Cb:=C[0,k]+param.Cb;

for i:=1 to k do
  begin
    skaitiklis:=0;
    vardiklis:=0;
    for j:=0 to k-i do skaitiklis:=skaitiklis+C[j,i];
    for j:=0 to k-i do vardiklis:=vardiklis+C[j,i-1];
    mm[i]:=skaitiklis/vardiklis;
  end;

for i:=0 to k-1 do m[i]:=mm[i+1];

MuM[k]:=Cb/C[0,k];
for i:=k-1 to 0 do MuM[i]:=m[i]*MuM[i+1];
for i:=0 to k do mCB[i]:=C[i,k-i]*M[k-i];
for i:=0 to k do RezM[i]:=mCB[i]-C[i,k-i];

```

```

RezervaiM:=0;
for i:=0 to k do RezervaiM:=RezM[i]+RezervaiM;
multiplikatyvinis:=RezervaiM;
end;

```

```

//-----
function GenerIsmDyd(skirsttipas: integer; par1,par2,par3: extended; poz: boolean): extended;
begin

```

```

case skirsttipas of
0: GenerIsmDyd:=RandEksp(par1);
1: GenerIsmDyd:=RandChiKvadrat(par1);
2: GenerIsmDyd:=RandLogN(par1,par2,poz);
3: GenerIsmDyd:=RandVeibulo(par1,par2);
4: GenerIsmDyd:=RandPareto(par1,par2);
5: GenerIsmDyd:=RandBurr(par1,par2,par3);
else GenerIsmDyd:=0;
end;
end;

```

```

//-----
procedure ni_generavimas(ilgis:integer; var ni:Tvektorius);
var i:integer;
    liambda1:extended;

```

```

begin
SetLength(ni,ilgis+1);
liambda1:=param.IsmSkPar1;
for i:=0 to ilgis do ni[i]:=randpuasono1(liambda1);
end;

```

```

//-----
function geometrinis(t:integer; atidimokejimai:TMasyvas; ni:Tvektorius):extended;
var i,j,k,n,dmax:integer;

```

```

Ra,AB0,AB1,AB2,tarpinis,skaitiklis, vardiklis,Cb,rezervaiG:extended;
Ri3,Rs,Md, Sbe,h,lamda_n,lamda_g,p_g,lamda_gp,p,w,E,RezG,pp,ll:TVektorius;
A,Sd,s:Tmasyvas;

```

```

begin
dmax:=min(param.DMax, t);
Cb:=param.Cb;
k:=dmax-1;
SetLength(w,dmax+2);
SetLength(E,dmax+2);
SetLength(RezG,dmax+2);
SetLength(lamda_g,dmax+2);
SetLength(p_g,dmax+2);
SetLength(lamda_gp,dmax+2);
SetLength(p,dmax+2);
SetLength(pp,dmax+2);
SetLength(ll,dmax+2);
SetLength(h,dmax*2);
SetLength(Sbe,dmax+2);
SetLength(Md,dmax+2);
SetLength(Rs,dmax+2);
SetLength(Ri3,dmax+2);

```

```

SetLength(s,dmax+2);
for i:= 0 to dmax+1 do Setlength(s[i],dmax+2);

SetLength(Sd,dmax+2);
for i:= 0 to dmax+1 do Setlength(Sd[i],dmax+2);

SetLength(A,dmax+2);
for i:= 0 to dmax+1 do Setlength(A[i],dmax+2);

SetLength(lamda_n,dmax+2);

for i:=0 to k do
begin
  for j:=0 to k do
  begin
    if j<=k-i then s[i,j]:=atidimokejimai[i,j]/ni[i]
    else s[i,j]:=0;
  end
end;

for j:=0 to k do
w[j]:=1;
begin
  for i:=0 to k-j do w[j]:=w[j]*s[i,j];
end;

for j:=0 to k do
E[j]:=1;
begin
  for i:=0 to j do E[j]:=E[j]*s[i,j-i];
end;

//lamda_g ir p skaiciavimas
//-----
lamda_g[k]:=Power(E[k],1/(k+1));
p_g[k]:=w[k]/lamda_g[k];
for i:=k-1 to 0 do
begin
  tarpinis:=1;
  for j:=i+1 to k do tarpinis:=p_g[j]*tarpinis;
  tarpinis:=tarpinis*E[i];
  lamda_g[i]:=Power(tarpinis,1/(i+1));
  tarpinis:=1;
  for j:=i to k do tarpinis:=tarpinis*lamda_g[j];
  tarpinis:=w[i]/tarpinis;
  p_g[i]:=Power(tarpinis,1/(k-i+1));
end;
//-----
lamda_gp[k]:=Power(E[k],1/(k+1));

```

```

p[k]:=w[k]/lamda_gp[k];
for i:=k-1 to 0 do
begin
  tarpinis:=1;
  for j:=i+1 to k do tarpinis:=p[j]*tarpinis;
  tarpinis:=tarpinis*E[i];
  lamda_gp[i]:=Power(tarpinis,1/(i+1));
  tarpinis:=1;
  for j:=i to k do tarpinis:=tarpinis*lamda_gp[j];
  tarpinis:=w[i]/tarpinis;
  p[i]:=Power(tarpinis,1/(k-i+1));
end;

```

```

//-----
for i:=0 to k do
begin
  vardiklis:=0;
  for j:=0 to k do vardiklis:=vardiklis+p[j];
  pp[i]:=p[i]/vardiklis;
end;

```

```

for i:=0 to k do
begin
  tarpinis:=0;
  for j:=0 to k do tarpinis:=tarpinis+p[j];
  ll[i]:=lamda_g[i]*tarpinis;
end;

```

```

//-----
// AB koef skaiciavimas
//-----
tarpinis:=0;
for i:=0 to k do tarpinis:=tarpinis+(i+1)*ll[i];
skaitiklis:=tarpinis;
tarpinis:=0;
for i:=0 to k do tarpinis:=tarpinis+ll[i];
tarpinis:=tarpinis/k;
skaitiklis:=skaitiklis-tarpinis;

```

```

tarpinis:=0;
for i:=0 to k do tarpinis:=tarpinis+(i+1)*(i+1);
vardiklis:=tarpinis;
tarpinis:=0;
for i:=0 to k do tarpinis:=tarpinis+(i+1);
tarpinis:=tarpinis/k;
vardiklis:=vardiklis-tarpinis;

```

```

AB0:=skaitiklis/vardiklis;

```

```

tarpinis:=0;

```

```

for i:=0 to k do tarpinis:=tarpinis+l[i];
skaitiklis:=tarpinis;
tarpinis:=0;
for i:=0 to k do tarpinis:=tarpinis+(i+1);
tarpinis:=tarpinis*AB0;
skaitiklis:=skaitiklis-tarpinis;

AB1:=skaitiklis/k;

n:=k+1;
for i:=n to 2*n-2 do lamda_n[i-n]:=AB0*i+AB1;

//visos lenteles uzpildymas pagal zinomus p ir lamda
for i:=0 to k do h[i]:=lamda_g[i];
for i:=n to k*2 do
begin
  tarpinis:=0;
  for j:=0 to k do tarpinis:=tarpinis+p[j];
  h[i]:=lamda_n[i-n]/tarpinis;
end;
for i:=0 to k do
begin
  for j:=0 to k do Sd[i,j]:=p[j]*h[i+j];
end;

for i:=0 to k do A[i,0]:=atidimokejimai[i,0];
for i:=0 to k do
begin
  for j:=1 to k do
begin
  if j<=k-i then A[i,j]:=A[i,j-1]+atidimokejimai[i,j]
  else A[i,j]:=0;
end;
end;

//Sbe skaiciavimas
for i:=0 to n-1 do h[i]:=lamda_g[i];
for i:=n to (n-1)*2 do
begin
  tarpinis:=0;
  for j:=0 to k do tarpinis:=tarpinis+p[j];
  h[i]:=lamda_n[i-n]/tarpinis;
end;
if ni[0]=0 then ni[0]:=1;
Sbe[0]:=Cb/ni[0];
for i:=1 to n-1 do Sbe[i]:=Sbe[i-1]*(h[n-1+i]/h[n-2+i]);

// skaiciuojama M ivercius
for i:=0 to n-1 do
begin

```

```

skaitiklis:=0;
for k:=0 to n-1 do skaitiklis:=skaitiklis+Sd[n-1-i,k];
skaitiklis:=skaitiklis+Sbe[n-1-i];
vardiklis:=1;
for j:=0 to i do vardiklis:=vardiklis+Sd[n-1-i,j];
Md[i]:=skaitiklis/vardiklis;
end;

```

```

//Skaiciuoja Rs
for i:=0 to n-1 do Rs[i]:=Md[n-1-i]*A[i,n-1-i];

```

```

// skaiciuoja rezervus atskyrimo metodais
for i:=0 to n-1 do Ri3[i]:=A[i,n-1-i]*(Md[n-1-i]-1);

```

```

// Ra
Ra:=0;
for i:=0 to n-1 do Ra:=Ra+Ri3[i];
geometrinis:=Ra;
end;

```

```

//-----Modeliavimas-----

```

```

procedure TForm1.PabaigaClick(Sender: TObject);
begin
Application.Terminate;
end;

```

```

//-----

```

```

procedure TForm1.FormCreate(Sender: TObject);
begin
RealizSk:=0;
bitbtnStop.Hide; stop:=False;
spalva:=clBlack;
sl:=0; //sluoksniu skaitliukas
end;

```

```

//-----

```

```

procedure TForm1.btnStopClick(Sender: TObject);
begin
stop:=True;
end;

```

```

//-----

```

```

procedure TForm1.BitBtnModeliuotiClick(Sender: TObject);
var t,i,t0,t1: integer; aktyvai,rizlikutis,grezervai,netkoef,islkoef,bendrkoef,tst_knt,mult_rez, geo_rez: extended;
sukaptomsismokos,atidimokejimai: TMasyvas;
ni: TVektorius;
ilgis:integer;
begin
try
t0:=0; t1:=SpinEdit2.Value; RealizSk:=SpinEdit1.Value;
ProgressBar1.Max:=RealizSk*t1;
bitbtnStop.Show; bitbtnModeliuoti.Hide; Form1.Repaint;

```

```

randomize; stop:=False;

```



```

Chart1.SeriesList[0].SeriesColor:=clBlack;
for t:=t0 to t0+t1 do Chart1.SeriesList[0].AddXY(t,0,"clTeeColor);

for i:=1 to realizsk do begin {realizacijø skaièius}

randomize;
aktyvai:=param.PradRez;
rizlikutis:=param.PradRez;

Chart1.AddSeries(TFastLineSeries.Create(Self)); sl:=sl+1;
Chart1.SeriesList[sl].SeriesColor:=spalva;
case ComboBox1.ItemIndex of //Breziame pradini taska
0,1,5,6,7: Chart1.SeriesList[sl].AddXY(t0,0,"clTeeColor);
2,3: begin
    Chart1.SeriesList[sl].AddXY(t0,param.PradRez,"clTeeColor);
    end;
end; {case}
for t:=(t0+1) to t1 do {modeliavimo periodai}
begin {---}
Application.ProcessMessages; if Stop then Abort;

case ComboBox1.ItemIndex of
0: begin
    Chart1.SeriesList[sl].AddXY(t,patirtinuostoliai(),"clTeeColor);
    ProgressBar1.StepIt;
    end;
1: begin
    Chart1.SeriesList[sl].AddXY(t,atidmokejimai1(t),"clTeeColor);
    ProgressBar1.StepIt;
    end;
2: begin
    aktyvai:=NaujiAktyvai(t,aktyvai);
    Chart1.SeriesList[sl].AddXY(t,aktyvai,"clTeeColor);
    ProgressBar1.StepIt;
    end;
3: begin
    rizlikutis:=Naujirizlikutis(rizlikutis);
    Chart1.SeriesList[sl].AddXY(t,rizlikutis,"clTeeColor);
    ProgressBar1.StepIt;
    end;
4: begin
    if param.DMax<2 then
    begin
    MessageDlg('Klaida taikant grandininá laiptø metodà'+#13#10+
'Maksimalus reguliavimo periodø skaièius turi bûti didesnis.',
    mtWarning, [mbOk], 0);
    Abort;
    end;
    apmoksetosismokos(t); //gaunamas apismokos dvimatis masyvas

```

```

if t>1 then
begin
sukaupismokos(t,apismokos,sukauposismokos); //sukaupos ismokos masyvo formavimas
grezervai:=Grandininis(t,sukauposismokos); //vienos reikšmes rezervo skaičiavimas prie konkretaus t
Chart1.SeriesList[sl].AddXY(t,grezervai,"clTeeColor);
sukauposismokos:=Nil; //apnulinama sukaupos ismokos
end;
ProgressBar1.StepIt;
end;
5: begin
netkoef:=patirtinuostoliai()/(param.ImokDydis*param.PolSkaicius);
Chart1.SeriesList[sl].AddXY(t,netkoef,"clTeeColor);
ProgressBar1.StepIt;
end;
6: begin
islkoef:=atidmokejimai1(t)/(param.ImokDydis*param.PolSkaicius);
Chart1.SeriesList[sl].AddXY(t,islkoef,"clTeeColor);
ProgressBar1.StepIt;
end;
7: begin
netkoef:=patirtinuostoliai()/(param.ImokDydis*param.PolSkaicius);
islkoef:=atidmokejimai1(t)/(param.ImokDydis*param.PolSkaicius);
bendrkoef:=netkoef+islkoef;
Chart1.SeriesList[sl].AddXY(t,bendrkoef,"clTeeColor);
ProgressBar1.StepIt;
end;
8: begin
if param.DMax<2 then
begin
MessageDlg('Klaida taikant multiplikatyvini metodà!'+#13#10+
'Maksimalus reguliavimo periodø skaièius turi bûti didesnis.',
mtWarning, [mbOk], 0);
Abort;
end;
apmuketosismokos(t); //gaunamas apismokos dvimatis masyvas
if t>1 then
begin
atidmokejimai_mas(t,atidimokejimai); //sukaupos ismokos masyvo formavimas
mult_rez:=multiplikatyvinis(t,atidimokejimai);
Chart1.SeriesList[sl].AddXY(t,mult_rez,"clTeeColor);
atidimokejimai:=Nil; //apnulinama sukaupos ismokos
end;
ProgressBar1.StepIt;
end;
9: begin
if param.DMax<=1 then
begin
MessageDlg('Klaida taikant geometrini metodà!'+#13#10+
'Maksimalus reguliavimo periodø skaièius turi bûti didesnis.',
mtWarning, [mbOk], 0);

```

```

Abort;
end;
apmoketosismokos(t); //gaunamas apismokos dvimatis masyvas
if t>=2 then
begin
atidimokejimai_mas(t,atidimokejimai); //sukauptos ismokos masyvo formavimas
ilgis:=min(param.DMax, t);
//ilgis:=length(sukauptosismokos);
ni_generavimas(ilgis,ni);
geo_rez:=geometrinis(t,atidimokejimai,ni);
//geo_rez:=length(ni);
Chart1.SeriesList[sl].AddXY(t,geo_rez,"clTeeColor);
atidimokejimai:=Nil; //apnulinama sukauptos ismokos
ni:=Nil;
end;
ProgressBar1.StepIt;
end;

end; {case}
end; {---}
apismokos:=Nil;
        end; {realizacijø skaièius }
finally
ProgressBar1.Position:=0; {ProgressBar1.Hide;}
bitbtnStop.Hide; bitbtnModeliuoti.Show;
apismokos:=Nil;
end;
end; {BitBtnModeliuotiClick}
//-----
procedure TForm1.BitBtnParametraiClick(Sender: TObject);
begin
Form3.RodytiParam(Form3);
Form3.ShowModal;
end;
//-----
procedure TForm1.BitBtnStopClick(Sender: TObject);
begin
stop:=True;
end;
//-----
procedure TForm1.sbSpalvaClick(Sender: TObject);
begin
if ColorDialog1.Execute then spalva:=ColorDialog1.Color;
sSpalva.Brush.Color := spalva;
end;
//-----
procedure TForm1.BitBtnIsvalytiClick(Sender: TObject);
begin
While Chart1.SeriesList.Count>1 do Chart1.SeriesList[0].Free;
Chart1.SeriesList[0].Clear; sl:=0;

```

end;  
end.

---

**unit Unit3;**

```
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Grids, ComCtrls, Math,
  Spin, Buttons, Unit1;
type
  TForm3 = class(TForm)
    Panel4: TPanel;
    Panel1: TPanel;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    PradRezEdit: TEdit;
    Label2: TLabel;
    PolSkEdit: TEdit;
    GroupBox2: TGroupBox;
    Label3: TLabel;
    ImokaEdit: TEdit;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    Label25: TLabel;
    cbIsmSkSkirst: TComboBox;
    lbIsmSkSkirstFormule: TLabel;
    lbIsmSkVidFormule: TLabel;
    GroupBox13: TGroupBox;
    lbIsmSkPar1: TLabel;
    ckbNyraNorm: TCheckBox;
    edIsmSkPar1: TEdit;
    GroupBox5: TGroupBox;
    cbIsmDydSkirst: TComboBox;
    Label28: TLabel;
    lbIsmDydSkirstFormule: TLabel;
    GroupBox6: TGroupBox;
    lbIsmDydPar1: TLabel;
    lbIsmDydPar2: TLabel;
    lbIsmDydPar3: TLabel;
    edIsmDydPar1: TEdit;
    edIsmDydPar2: TEdit;
    edIsmDydPar3: TEdit;
    GroupBox7: TGroupBox;
    GroupBox8: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    CbEdit: TEdit;
    Label6: TLabel;
    StringGrid1: TStringGrid;
    Label7: TLabel;
    TikSumaEdit: TEdit;
```

```

ckbIsmDydVidStd: TCheckBox;
DmaxSpinEdit: TSpinEdit;
BitBtnAtsaukti: TBitBtn;
BitBtnPriimti: TBitBtn;
BitBtnAtstatyti: TBitBtn;
procedure btnAtsauktiClick(Sender: TObject);
procedure btnPriimtiClick(Sender: TObject);
procedure StringGrid1SetEditText(Sender: TObject; ACol, ARow: Integer;
  const Value: String);
procedure FormCreate(Sender: TObject);
procedure cbIsmDydSkirstChange(Sender: TObject);
procedure DmaxSpinEditChange(Sender: TObject);
procedure BitBtnPriimtiClick(Sender: TObject);
procedure RodytiParam(Sender: TObject);
procedure InitParam();
procedure BitBtnAtstatytiClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form3: TForm3;
procedure Perskaiciuoti(var KintStringGrid: TStringGrid; var KintEdit: TEdit);
procedure ReiksmiuSkPasikeite(KintEdit: TEdit; var KintEdit1: TEdit; KintUpDown: TUpDown;
  var KintStringGrid: TStringGrid);
procedure RSkPasikeite(KintSpinEdit: TSpinEdit; var KintEdit1: TEdit;
  var KintStringGrid: TStringGrid);
function SkaiciusSveikas(Sk:String):boolean;
function SkaiciusRealus(Sk:String; var x: real):boolean;

```

#### implementation

```

{$R *.DFM}
//-----
function SkaiciusSveikas(Sk:String):boolean;
//Tikrina ar skaicius sveikas
const Skaiciai: set of Char = ['0'..'9'];
var S: String;
begin
  S:=Sk; SkaiciusSveikas:=False;
  while (S <> '') and (S[1] in Skaiciai) do Delete(S, 1, 1);
  if (S = '') and (Sk <> '') then SkaiciusSveikas:=True;
end;
//-----
function SkaiciusRealus(Sk:String; var x: real):boolean;
//Tikrina ar skaicius realus
var code: integer; S: String; simb: char;

```

```

begin
SkaiciusRealus:=False; code:=0; simb:=#49; S:=Sk;
if Sk <> " then begin
  Val(S,x,code);
  if code <>0 then simb:=S[code];
  if (Ord(simb)=44) then begin
    Delete(S,code,1); Insert('.',S,code);
    Val(S,x,code); end;
  if code=0 then SkaiciusRealus:=True;
    end;
end;
//-----
procedure Perskaiciuoti(var KintStringGrid: TStringGrid; var KintEdit: TEdit);
//Perskaiciuoja tikimybiu suma
var i: integer; suma,x: real;
begin
suma:=0;
for i:=1 to KintStringGrid.RowCount-1 do begin
if SkaiciusRealus(KintStringGrid.Cells[1,i],x) then
  suma:=suma+x;
  end;
KintEdit.Text:=FloatToStr(suma);
end;
//-----
procedure RSkPasikeite(KintSpinEdit: TSpinEdit; var KintEdit1: TEdit;
var KintStringGrid: TStringGrid);
var i: integer; RSk: Integer;
begin
RSk:=KintSpinEdit.Value;
if (RSk>=KintSpinEdit.MinValue)and(RSk<=KintSpinEdit.MaxValue) then
  begin
RSk:=Min(RSk,KintSpinEdit.MaxValue);
for i:=RSk+1 to KintStringGrid.RowCount-1 do KintStringGrid.Cells[1, i]:=?;
KintStringGrid.RowCount:=RSk+1;
for i:=1 to KintStringGrid.RowCount - 1 do KintStringGrid.Cells[0, i]:=IntToStr(i-1);
Perskaiciuoti(KintStringGrid,KintEdit1);
  end;
end;
//-----
procedure ReiksmiuSkPasikeite(KintEdit: TEdit; var KintEdit1: TEdit; KintUpDown: TUpDown;
var KintStringGrid: TStringGrid);
var i: integer; RSkaicius: String; RSk: Integer;
begin
RSkaicius:=KintEdit.Text;
if SkaiciusSveikas(RSkaicius)and(StrToInt(RSkaicius)>=1) then
  begin
RSk:=Min(StrToInt(Trim(RSkaicius)),KintUpDown.Max);
for i:=RSk+1 to KintStringGrid.RowCount-1 do KintStringGrid.Cells[1, i]:=?;
KintStringGrid.RowCount:=RSk+1;
for i:=1 to KintStringGrid.RowCount - 1 do KintStringGrid.Cells[0, i]:=IntToStr(i-1);

```

```

Perskaiciuoti(KintStringGrid,KintEdit1);
end;
end;
//-----
procedure TForm3.btnAtsauktiClick(Sender: TObject);
begin
Form3.Close;
end;

procedure TForm3.btnPriimtiClick(Sender: TObject);
begin
Form3.Close;
end;
//-----
procedure TForm3.StringGrid1SetEditText(Sender: TObject; ACol,
ARow: Integer; const Value: String);
begin
Perskaiciuoti(StringGrid1,TikSumaEdit);
end;
//-----
procedure TForm3.FormCreate(Sender: TObject);
begin
StringGrid1.ColWidths[0]:= 30;
StringGrid1.Cells[0, 0]:= 'd';
StringGrid1.Cells[1, 1]:= '1';
StringGrid1.Cells[0, 1]:= '0';
StringGrid1.Cells[1, 0]:= 'p(d)';
cbIsmSkSkirst.ItemIndex :=0;
cbIsmDydSkirst.ItemIndex :=0;
InitParam();
end;
//-----
procedure TForm3.cbIsmDydSkirstChange(Sender: TObject);
procedure RodytiParam(k:integer);
begin
case k of
1: begin
lbIsmDydPar1.Show; lbIsmDydPar2.Hide; lbIsmDydPar3.Hide;
edIsmDydPar1.Show; edIsmDydPar2.Hide; edIsmDydPar3.Hide;
end;
2: begin
lbIsmDydPar1.Show; lbIsmDydPar2.Show; lbIsmDydPar3.Hide;
edIsmDydPar1.Show; edIsmDydPar2.Show; edIsmDydPar3.Hide;
end;
3: begin
lbIsmDydPar1.Show; lbIsmDydPar2.Show; lbIsmDydPar3.Show;
edIsmDydPar1.Show; edIsmDydPar2.Show; edIsmDydPar3.Show;
end;
end; {case}
end;

```

```

begin
case cbIsmDydSkirst.ItemIndex of
0: begin {Ekspontentinis}
  RodytiParam(1); ckbIsmDydVidStd.Hide;
  lbIsmDydSkirstFormule.Caption:=F(x)=1-exp(-(1/a)*x), E(X)=a';
  lbIsmDydPar1.Caption:=a';
  end;
1: begin {Chi-kvadratu}
  RodytiParam(1); ckbIsmDydVidStd.Hide;
  lbIsmDydSkirstFormule.Caption:=X=Y1^2+Y2^2+..+Yk^2, Yi~N(0,1)';
  lbIsmDydPar1.Caption:=k=';
  end;
2: begin {Lognormalusis}
  RodytiParam(2); ckbIsmDydVidStd.Show;
  lbIsmDydSkirstFormule.Caption:=F(x)=N(z,0,1), z=(ln(x)-m)/s';
  lbIsmDydPar1.Caption:=m='; lbIsmDydPar2.Caption:=s=';
  end;
3: begin {Veibulo}
  RodytiParam(2); ckbIsmDydVidStd.Hide;
  lbIsmDydSkirstFormule.Caption:=F(x)=1-exp{-[x/teta]^tau}';
  lbIsmDydPar1.Caption:=teta='; lbIsmDydPar2.Caption:=tau=';
  end;
4: begin {Pareto}
  RodytiParam(2); ckbIsmDydVidStd.Hide;
  lbIsmDydSkirstFormule.Caption:=F(x)=1-(teta/[teta+x])^alfa';
  lbIsmDydPar1.Caption:=teta='; lbIsmDydPar2.Caption:=alfa=';
  end;
5: begin {Burr}
  RodytiParam(3); ckbIsmDydVidStd.Hide;
  lbIsmDydSkirstFormule.Caption:=F(x)=1-(teta/[teta+x^tau])^alfa';
  lbIsmDydPar1.Caption:=teta='; lbIsmDydPar2.Caption:=alfa=';
  lbIsmDydPar3.Caption:=tau=';
  end;
end; {case}
end;

```

```

//-----
procedure TForm3.DmaxSpinEditChange(Sender: TObject);
begin
  RSkPasikeite(DMaxSpinEdit,TikSumaEdit,StringGrid1);
end;

```

```

//-----
procedure TForm3.BitBtnPriimtiClick(Sender: TObject);
//Nuskaito parametrus is lauku i kintamuosius
var i: integer; MyErrorString: string; TempParam: TParametrai;
begin
  try
    TempParam.PradRez:=StrToFloat(PradRezEdit.Text);
    TempParam.PolSkaicius:=StrToInt(PolSkEdit.Text);
    TempParam.ImokDydis:=StrToFloat(ImokaEdit.Text);
    TempParam.DMax:=DmaxSpinEdit.Value;
  
```



```

SetLength(TempParam.TikMasyvas,DMaxSpinEdit.Value);
for i:=1 to DMaxSpinEdit.Value do
  TempParam.TikMasyvas[i-1]:=StrToFloat(StringGrid1.Cells[1,i]);
TempParam.Cb:=StrToFloat(CbEdit.Text);
TempParam.IsmSkSkirst:=cbIsmSkSkirst.ItemIndex;
TempParam.IsmSkPar1:=StrToFloat(edIsmSkPar1.Text);
TempParam.NyraNorm:=ckbNyraNorm.Checked;
TempParam.IsmDydSkirst:=cbIsmDydSkirst.ItemIndex;
TempParam.IsmDydPar1:=StrToFloat(edIsmDydPar1.Text);
TempParam.IsmDydPar2:=StrToFloat(edIsmDydPar2.Text);
TempParam.IsmDydPar3:=StrToFloat(edIsmDydPar3.Text);
TempParam.VidStd:=ckbIsmDydVidStd.Checked;
Param:=TempParam; TempParam.TikMasyvas:=Nil;

except
  MyErrorString := 'Blogai ávestos parametrø reikðmës!'+#13#10+'Klaida: ' + IntToStr(Error) + #13#10;
  MessageDlg(MyErrorString, mtWarning, [mbOk], 0);
end;
end;
//-----
procedure TForm3.RodytiParam(Sender: TObject);
//Nuskaito parametrus is kintamuju i laukus
var i: integer;
begin
  PradRezEdit.Text:=FloatToStr(Param.PradRez);
  PolSkEdit.Text:=IntToStr(Param.PolSkaicius);
  ImokaEdit.Text:=FloatToStr(Param.ImokDydis);
  DmaxSpinEdit.Value:=Param.DMax; //DmaxSpinEditChange(Form3);
  for i:=1 to Param.DMax do
    StringGrid1.Cells[1,i]:=FloatToStr(Param.TikMasyvas[i-1]);
  CbEdit.Text:=FloatToStr(Param.Cb);
  cbIsmSkSkirst.ItemIndex:=Param.IsmSkSkirst;
  edIsmSkPar1.Text:=FloatToStr(Param.IsmSkPar1);
  ckbNyraNorm.Checked:=Param.NyraNorm;
  cbIsmDydSkirst.ItemIndex:=Param.IsmDydSkirst; cbIsmDydSkirstChange(Self);
  edIsmDydPar1.Text:=FloatToStr(Param.IsmDydPar1); //ShowMessage('edIsmDydPar1.Text :=
'+FloatToStr(Param.IsmDydPar1));
  edIsmDydPar2.Text:=FloatToStr(Param.IsmDydPar2);
  edIsmDydPar3.Text:=FloatToStr(Param.IsmDydPar3);
  ckbIsmDydVidStd.Checked:=Param.VidStd;

end;
//-----
procedure TForm3.InitParam();
//Priskiria parametru reiksmes pagal nutylejima
begin
  Param.PradRez:=0;
  Param.PolSkaicius:=0;
  Param.ImokDydis:=0;
  Param.DMax:=1;

```

```
SetLength(Param.TikMasyvas,Param.Dmax);  
  Param.TikMasyvas[0]:=1;  
Param.Cb:=0;  
Param.IsmSkSkirst:=0;  
Param.IsmSkPar1:=0;  
Param.NyraNorm:=True;  
Param.IsmDydSkirst:=2;  
Param.IsmDydPar1:=0;  
Param.IsmDydPar2:=0;  
Param.IsmDydPar3:=0;  
Param.VidStd:=True;
```

end;

```
procedure TForm3.BitBtnAtstatytiClick(Sender: TObject);  
begin  
  InitParam();  
  RodytiParam(Form3);  
end;  
end.
```

### 3 PRIEDAS. STRAIPSNIO SANTRAUKA

#### ATIDĖTŲ MOKĖJIMŲ ĮTAKA REZERVŲ ĮVERTINIMUI

V. Karpickaitė, M. Peičius, K. Salatkaitė

*Kauno technologijos universitetas*

Laisvosios rinkos sąlygomis kiekvienas verslas yra susijęs su techniniais atidėjimais, tai laiku negauti laukiami pinigai, laiku neišmokėti pinigai ir pan. Draudime, kaip ir bet kuriame versle egzistuoja atidėjimai su draudimo veikla susiję draudiko išsipareigojimai, kurių paskirtis užtikrinti draudimo įmonių prievolių, kylančių iš draudimo sutarčių, vykdymą, išlaikyti stabilią draudimo įmonių finansinę padėtį. Draudimo įmonės techniniams atidėjimams, rezervams įvertinti naudosime aktuarinius metodus, bei matematinį modeliavimą.

Nagrinėsime techninių atidėjimų įtaką rezervams, žinodami pagrindinius draudimo veiklos parametrus, tokius kaip draudimo įmoka, polisų skaičius ir pan. Šiuo metu yra nemažai metodų šiai įtakai įvertinti ir apskaičiuoti pačius atidėjimus. Tačiau šie metodai yra taikomi dažniausiai tais atvejais, kai turime daugiau parametrų, todėl tikėtina, kad jie yra taikomi išsivysčiusiose vakarų Europos šalyse, kur draudimo veikla vykdoma jau daugelį dešimtmečių, susiformavusios senos draudimo tradicijos ir pan. Rezervams apskaičiuoti taikėme grandininį laiptų metodą ir paprastumo dėlei rezervus skaičiavome neįvertindami infliacijos ir kitų veiksnių. Nes nagrinėti sudėtingus, tačiau kartais tikslesnius metodus tampa sudėtinga. Atidėjimų įtakos rezervams įvertinimui naudosime matematinį – kompiuterinį modeliavimą, kuris šiais informacinių technologijų laikais tapo vienu iš pagrindinių šiuolaikinės draudimo teorijos įrankiu. Tam reikia sukurti matematinį modelį, aprašantį atidėtus mokėjimus.

Draudimo veiklos modeliavimui naudosime supaprastintą atidėtų mokėjimų modelį su tam tikrom prielaidom:

- Tarkime, kad nagrinėjama draudimo bendrovė tik pradėjo dirbti ir pirmasis periodas, tai draudimo bendrovės veiklos rezultatai po vienerių metų
- Visi modelyje naudojami dydžiai nekinta laike
- Modeliuojami polisai yra vienerių metų trukmės
- Polisų portfelio struktūra nekinta per visą modeliuojamą laikotarpį
- Draudimo įmokos imamos modelyje yra Netto įmokos, todėl, kad neatsižvelgiam į perdraudimą, pajamas iš investicijų ir pan.
- Išmokų skaičius modelyje yra laikomas nepriklausomu atsitiktiniu dydžiu pasiskirsčiusiu pagal Puasono skirstinį su parametru  $\lambda$ .
- Išmokų dydžiai modelyje laikomi atsitiktiniais dydžiais pasiskirsčiusiais pagal Lognormalųjį, Eksponentinį, Veibulo, Chi - Kvadratu, Burr ar Pareto skirstinius.

Matematinį modelį galime užrašyti taip:

$$A(t) = A(t-1) + B(t) - X_p(t)$$

čia  $t$  – modeliavimo periodas;

$A(t)$  – aktyvai momentu  $t$ ;

$A(t-1)$  – Aktyvai momentu  $t-1$ ;

$B(t)$  – pasirašytos įmokos momentu  $t$  (nagrinėjamame modelyje pasirašytos įmokos sutampa su uždirbtomis įmokomis);

$X_p(t)$  – apmokėtos išlaidos momentu  $t$  ir apskaičiuojamos pagal formulę:

$$X_p(t) = \sum_{u=\max(t-d, \max+1, 1)} X_p(u, t)$$

čia  $t$  – modeliavimo periodas;

$d_{\max}$  – maksimalus reguliavimo periodų skaičius;

$X_p(u, t)$  – atidėti mokėjimai įvykę  $u$  metais, bet apmokėti  $t$  metais; apskaičiuojami pagal formulę:

$$X_p(u, t) = \sum_{i=1}^{N(u, t)} X_i$$

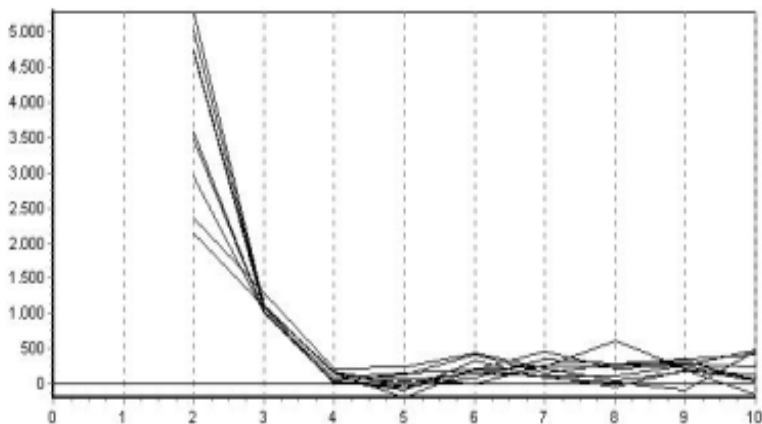
čia  $X_i$  – išmokų skaičius, kai  $N(u, t) \sim P(\lambda(u, t))$ :

$$\lambda(u, t) = p(t - u) \cdot \lambda$$

$p(t - u)$  – tikimybė, kad išmokos atsiradusios  $u$  – taisiais metais bus apmokėtos  $t$  – taisiais metais;

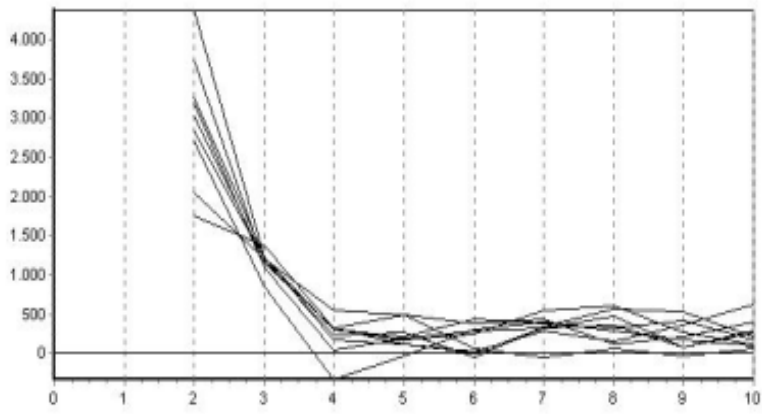
$\lambda$  – išmokos;

Išmokų skaičiaus ir išmokų dydžio skirstinius bei jų parametrus, atidėtų mokėjimų tikimybes, galime modeliuoti ir įvertinti atidėtų mokėjimų įtaką rezervams.

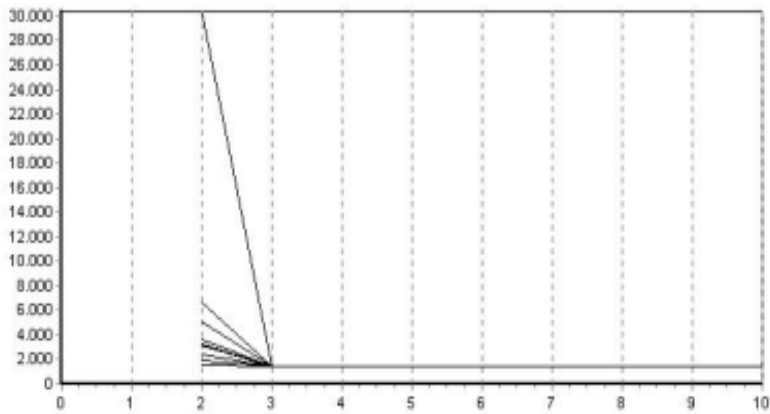


**1 pav.** Rezervai, apskaičiuoti grandininio laiptų metodu, kai išmokų dydis pasiskirstęs pagal Lognormalųjį skirstinį

Iš 1 paveikslo matyti, kad draudimo bendrovės nustatytos draudimo įmokos buvo pakankamos, nors kai kuriais periodais gauti neigiami rezervai. Tačiau būtų iškilusi problema tuo atveju, jei draudimo bendrovei neigiami rezervai būtų gaunami labai ilgą laiko tarpą. Tada, bendrovei grėstų bankrotas.



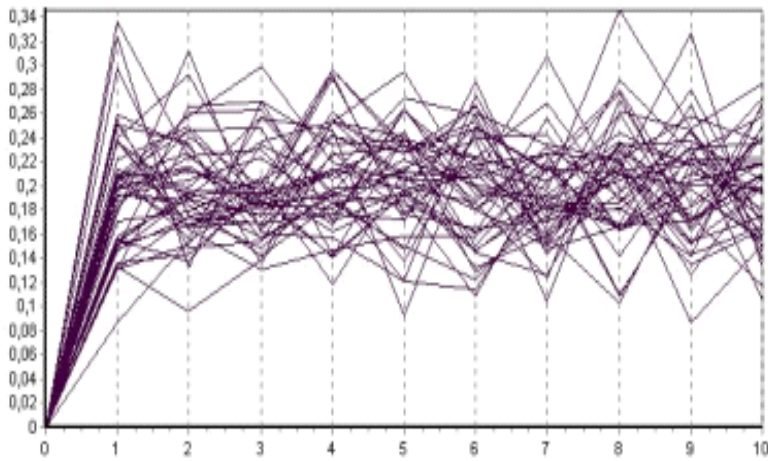
**2 pav.** Rezervai, apskaičiuoti grandinių laiptų metodu, kai išmokų dydis pasiskirstęs pagal Veibulo skirstinį  
 Iš 2 paveikslą matyti, kad rezervai nagrinėjamo periodo eigoje yra labiau išsisklaidę, turi didesnę dispersiją.



**3 pav.** Rezervai, apskaičiuoti grandinių laiptų metodu, kai išmokų dydis pasiskirstęs pagal Pareto skirstinį

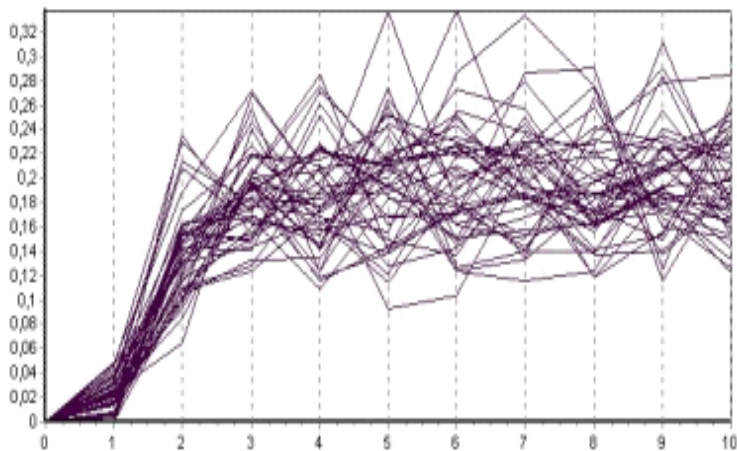
Iš 3 paveikslo matyti, kad rezervai nagrinėjamo periodo eigoje nusistovi - draudimo bendrovė dirba sėkmingai.

Buvo nagrinėti nuostolių ir išlaidų koeficientai, kurie apsprendžia, koks bus gautas pasirašytasis ciklas draudimo kompanijoje. Šį ciklą sukelia didėjantys pelnai, kurie veda prie didėjančių galimybių, tuo būdu sąlygodami agresyvios prekybos ir nukrypimų nuo pasirašytų standartų atsiradimus.



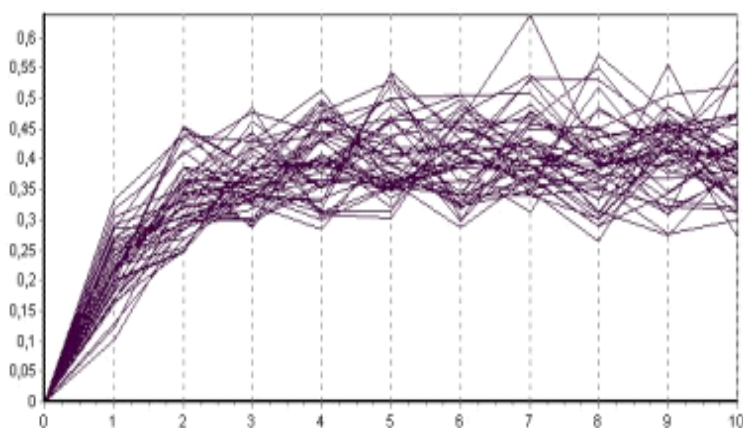
4 pav. Nuostolių koeficientas

Iš 4 paveikslo matome, kad nuostolių koeficientas yra mažas. Tai gerai, nes reiškia, kad patirti nuostoliai yra nedideli, lyginant su uždirbtomis draudimo įmokomis. Vidutinis nuostolių koeficientas per visą modeliavimo periodą yra lygus vidutiniškai 0,2.



### 5 pav. Išlaidų koeficientas

Išlaidų koeficientas, per visą modeliavimo periodą yra lygus vidutiniškai 0,18. Kaip matome iš paveikslo gaunam labai didelį šių koeficientų išsisklaidymą lyginant su nuostolių koeficientu.



### 6 pav. Bendras koeficientas

Iš 6 paveikslo matome, kad bendras koeficientas yra didesnis lyginant su prieš tai nagrinėtais taip yra todėl, kad šis koeficientas apskaičiuojamas kaip pastarųjų koeficientų suma. Bendras koeficientas labai priklauso nuo to kokius parametrus užsiduosime pačioje modeliavimo programoje; su kitais parametrais jis gali gautis dar mažesnis.

#### Išvados:

1. Įvertinome atidėtų mokėjimų įtaką rezervams, apskaičiuotiems grandininio laiptų metodu. Šis metodas yra siūlomas Valstybinės draudimo priežiūros tarnybos prie Finansų ministerijos valdybos, turint ne mažiau kaip 5 – rių metų duomenis.
2. Gauti rezultatai įvairių nuostolių skirstinių atveju parodė, kad:
  - a) Pareto ir Burr skirstinių atveju rezervai labai greitai nusistovėjo;
  - b) Lognormaliojo, Veibulo, Eksponentinio ir Chi – kvadratu skirstinių atveju rezervai nusistovi iš lėto.

## Literatūra

1. Booth P., Chadburn R. Modern actuarial theory and practice.-Chapman and Hall/CRC,1999.
2. Klugman S.A., Panjer H.H., Willmot G.E. Loss Models: from data to decisions. – New York: John Wiley & Sons, 1998.
3. Daykin C.D., Pentikäinen T., Pesonen M. Practical Risk Theory for Actuaries. – London: Chapman & Hall, 1994.
4. Čepinskis J., Raškinis D., Stankevičius R., Šernius A. Draudimas. – Kaunas: Pasaulio lietuvių kultūros, mokslo ir švietimo centras, 1999.
5. Гофман В., Хомоненко А. Delphi 5. – Санкт-Петербург: БХВ, 2000.

## THE PAY DELAYS INFLUENCE ON RESERVES

V. Karpickaitė, M. Peičius, K. Salatkaitė

## Summary

In this paper we were studying pay delays influence on reserves. Reserves were reckoned by chain – ladder method.