

ŠIAULIŲ UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

Tadas Vaitkus

Informatikos specialybės magistrantūros II kurso dieninio skyriaus studentas

**ENERGETIKOS STOCHASTINIO PROGRAMAVIMO
UŽDAVINIO TYRIMAS**

ANALYSIS OF THE ENERGETIC STOCHASTIC PROBLEM

MAGISTRO DARBAS

Darbo vadovas:

Doc. dr. K. Žilinskas

Recenzentas:

Doc. dr. L. Kaklauskas

Šiauliai, 2013

„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros sąrašę pateiktus informacinius šaltinius bei savo tyrimų duomenis“

Darbo autoriaus _____

(vardas, pavardė, parašas)

Darbo tikslas ir uždaviniai

Tikslas

Ištirti elektros poreikį ir jos pardavimo, pirkimo ir gamybos kainas, sudaryti elektros energijos gamybos modelį bei realizuoti optimizavimo programą *Java* kalba, kuri galėtų apskaičiuoti optimalų investavimą į elektros gamybą.

Uždaviniai

- Nagrinėti elektros gamybos kainas ir elektros energijos poreikį.
- Pasirinkti tinkamus matematinius optimizavimo metodus.
- Sudaryti matematinį uždavinio modelį.
- Parašyti kodą, skirtą optimizavimo uždaviniams *Java* kalba.
- Realizuoti ir testuoti programą.

Darbo vadovo _____

(vardas, pavardė, parašas)

Turinys

I. Įvadas	5
1.1. Įžanga	5
1.2. Tyrimų sritis.....	6
1.3. Problemos aktualumas.....	6
1.4. Mokslinis naujumas.....	6
1.5. Praktinė darbo reikšmė	6
1.6. Darbo tikslas	6
1.7. Uždaviniai	6
II. Analitinė dalis	7
2.1. Temos analizė	7
2.1.2. Elektros energijos poreikis ir kaina	7
2.1.3. Matematinis programavimas.....	11
2.1.4. Stochastinis programavimas.....	11
2.1.5. Atsitiktiniai skaičiai	13
2.1.6. <i>Monte – Karlo</i> metodas.....	14
2.2. Darbinės srities analizė.....	16
2.2.1. Stochastinio programavimo uždavinio sprendimas.....	16
2.2.2. Energetikos uždavinys	16
2.3. Darbinės srities modelis.....	17
2.3.1. Uždavinio sprendimo modelis.....	17
2.3.2 Uždavinio modelio duomenys.....	18
III. Projektinė dalis	19
3.1. Darbų eigos planas	19
3.2. Pradinis projektas	20
3.2.1. Programa	20
3.3. Įrankių ir priemonių pasirinkimas	21
IV. Realizacijos dalis	22
4.1. Darbų eigos aprašymas	22
4.2. Programos tobulinimas	23
4.2.1. Objektai.....	23
4.2.2. Programos sparta, gijos.....	24
4.2.3. Grafikai ir lentelės	25
4.3. Galutinis projekto aprašymas.....	26
4.4. Iškilusių problemų aprašymas.....	29
4.5. Darbo rezultatų analizė.....	30
V. Rekomendacijos	32
VI. Išvados ir rezultatai	33
VII. Literatūros sąrašas	34
Anotacija	36
Summary	37
Priedai	38
1. DVD turinys.....	38
2. Programos naudojimas	38
3. Uždavinio modelio sudarymas.....	41
4. Programos testavimas.....	46
5. Analizuojama programa	59
6. Programos spartos tikrinimas (Gijos).....	64

I. Įvadas

1.1. Įžanga

Lietuvoje uždarius Ignalinos atominę elektrinę svarstoma statyti Visagino atominę elektrinę [7, 22]. Prieš ją statant sprendžiama, ar verta statyti, ar pajėgs konkuruoti su kitomis Lietuvoje esančiomis elektrinėmis [1]. Elektrinės statyboms kainą gali nulemti paskolos ėmimas [20].

Norint optimizuoti Lietuvos energetikos struktūrą pirmiausia reikia ištirti esamą Lietuvos energetikos elektrinių instaliuojamąją galią. Kad šalis būtų nepriklausoma nuo kitų šalių turi turėti elektros energijos gaminimo rezervą. Energetikos struktūroje svarbų vaidmenį sudaro energijos poreikis. Skirtingu metu energijos poreikis skiriasi, žiemą didesnis, o vasarą mažesnis.

Darbe ieškoma kokios ir kokios galios elektrinės jau yra Lietuvoje. Bandoma surasti nuo kiek metų ir kokią sumą investavus būtų verta statyti atominę elektrinę. Taip pat ieškoma, kokias ir kokios galios elektrines optimalu statyti, kai investuojama tam tikra suma tam tikram laikotarpiui.

Energetikos uždaviniui išspręsti kuriama optimizavimo programa. Įrankis realizuojamas *Java* programavimo kalba. Kad būtų lengviau realizuojama, pasirinkta objektinio programavimo paradigma. Šiais laikais kompiuteriai dažniausiai būna daugiau negu vieno branduolio, dėl to papildomai nutarta panaudoti lygiagretaus programavimo paradigma, kad programa galėtų vienu metu skaičiuoti kelis uždavinius, taip sutaupydama laiką.

Kad galėtume išspręsti uždavinį, reikia sumodeliuoti tinkamą energetikos uždavinio modelį. Žinant, kad uždavinyje gali būti atsitiktinių skaičių ir uždavinio parametrai gali keistis, pasirinktas tiesinis dviejų etapų stochastinio programavimo uždavinio modelis.

„Pagrindinės programinės įrangos kūrimo kompanijos (Borland International, Microsoft Corporation ir kt.) daro prielaidą, kad objektinio programavimo technologija palengvins tiek programuotojų, tiek vartotojų darbą su kompiuteriais“ [16]. Objektinis programavimas – objektiškai orientuotas programavimas, kuris remiasi objekto koncepcija [16].

Lygiagretusis programavimas – tai technologija, kuri geba kompiuterio programoje atlikti daugiau nei vieną giją ar procesą lygiagrečiai vienu metu. Ši technologija dažniausiai naudojama daugiau negu vieno branduolio kompiuteriuose, norint išnaudoti visą kompiuterio procesoriaus spartą [8].

1.2. Tyrimų sritis

Tiriamas elektros energijos infrastruktūros bei jos gamybos optimizavimo modelis ir kuriama taikomoji programa, skirta optimizuoti energetikos stochastinio programavimo uždavinį.

1.3. Problemos aktualumas

Pasaulyje vis didėja elektros poreikis. Elektrai gaminti yra daug būdų. Kiekviena įranga gaminti elektrai kainuoja skirtingai. Kartais būna sudėtinga nustatyti ir neaišku, kuris variantas yra optimalus, kad elektros gamybos kaina būtų mažiausia ir kartu jos užtektų.

1.4. Mokslinis naujumas

Mokslinis naujumas – bandymas sudaryti realų energetikos uždavinį pagal Lietuvos situaciją ir jo optimizavimas. Taikomoji programa yra kuriama pasinaudojus *Java* programavimo kalba, kuri remiasi objekcinio programavimo principu.

1.5. Praktinė darbo reikšmė

Modeliuojamas realus Lietuvos energetikos uždavinio modelis. Taikomosios programos pagalba sudaromi pasiūlymai – Lietuvos energetikos struktūrai.

1.6. Darbo tikslas

Ištirti elektros poreikį ir jos pardavimo, pirkimo ir gamybos kainas, sudaryti elektros energijos gamybos modelį bei realizuoti optimizavimo programą *Java* kalba, kuri galėtų apskaičiuoti optimalų investavimą į elektros gamybą.

1.7. Uždaviniai

- Nagrinėti elektros gamybos kainas ir elektros energijos poreikį.
- Pasirinkti tinkamus matematinius optimizavimo metodus.
- Sudaryti matematinį uždavinio modelį.
- Parašyti kodą, skirtą optimizavimo uždaviniams *Java* kalba.
- Realizuoti ir testuoti programą.

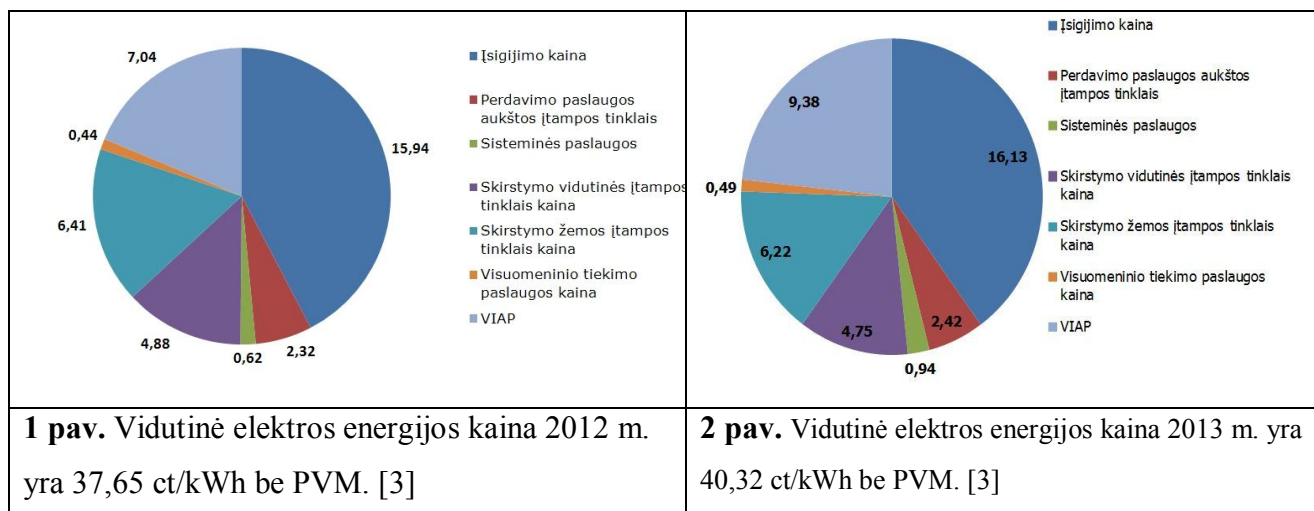
II. Analitinė dalis

2.1. Temos analizė

2.1.2. Elektros energijos poreikis ir kaina

Anot elektros perdavimo sistemos operatoriaus AB „Litgrid“, per devynis 2011 metų mėnesius aukštos įtampos elektros tinklais šalies poreikiams perdavė 6,84 mlrd. kWh elektros energijos arba 1,8 proc. daugiau negu buvo per devynis praėjusių metų mėnesius. Skirstomųjų tinklų operatoriui AB „LESTO“ perduota 5,99 mlrd. kWh – 0,5 proc. daugiau nei 2010 metais, kitiems vartotojams – 0,85 mlrd. kWh – 12,1 proc. daugiau. [14]

Tokiais teiginiais matyti, kad elektros poreikis vis didėja ir tuo tarpu atsiranda poreikis gauti kuo pigesnės elektros, kad būtų galima pasiekti, kuo didesnę pelną.



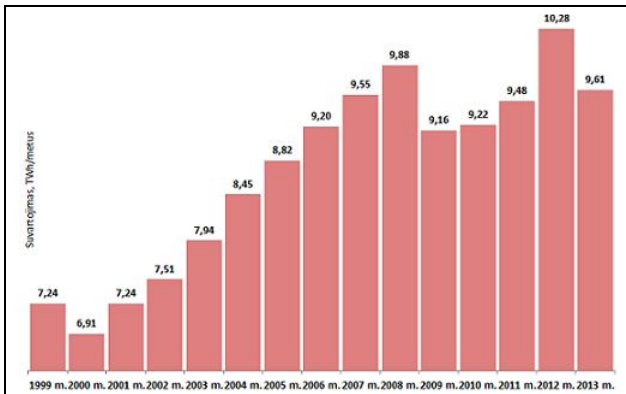
„VIAP (viešuosius interesus atitinkančios paslaugos) „ [3]

Palyginus 1 ir 2 pavaizduotus paveikslėlius, matyti, kad elektros kaina 2013 metais didesnė už 2012 metų elektros kainą.

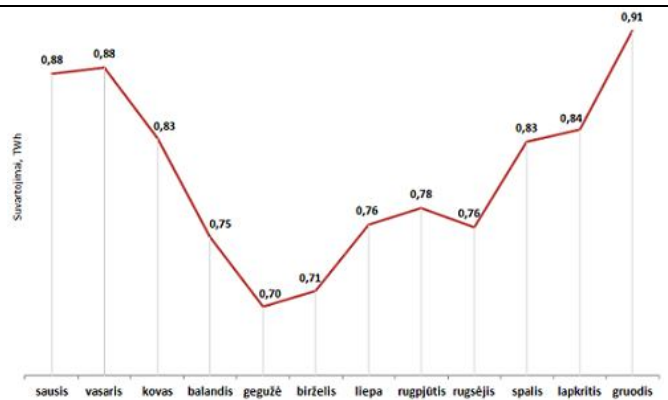
Pagal Pasaulio bankų ataskaitas teigiama, kad per ateinančių 20 metų elektros energijos paklausa Rytų Europoje ir Vidurio Azijoje išaugs beveik dvigubai. Tai konstatuojama naujausioje Pasaulio banko ataskaitoje. [17]

Ekspertai prognozuoja, kad pirminių energijos šaltinių paklausa padidės 50 proc., o elektros energijos poreikis – net 90 proc. Energetikos ministerijos teigimu, kad tokiais tempais didėjant paklausai šalys nepatektų į energijos krizę, jos turi kuo greičiau atnaujinti šio sektoriaus infrastruktūrą ir užtikrinti pakankamą energijos pasiūlą. [17]

Sparčiau didėjant elektros paklausai negu pasiūlai, automatiškai verčia kilti elektros kainoms. Kad to išvengtume, reikia didinti pasiūlą. Statyti elektrines ir gaminti didesnę elektros kiekį už kuo mažesnę kainą. Kad žinotume, ką labiausiai verta statyti, reikia atlikti tyrimus. Elektrinės, kurios gamina elektrą už labai mažą kainą, dažniausiai kainuoja didelius pinigus ir ne visada apsimoka jas statyti, o pigios statybos elektrinės, gali gaminti brangią elektrą.

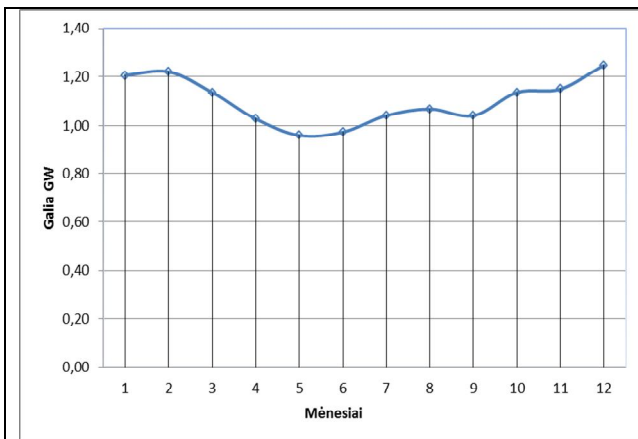


3 pav. Faktiniai Lietuvos elektros energijos vartotojų metiniai suvartojimai (TWh) [5]

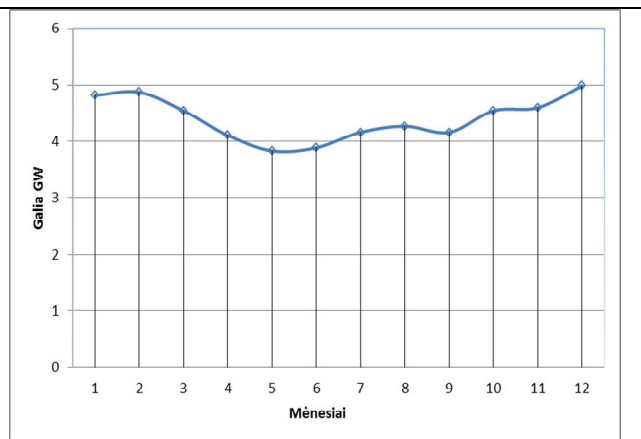


4 pav. Faktiniai 2012 m. Lietuvos elektros energijos vartotojų mėnesiniai suvartojimai (TWh) [5]

Pagal 3 ir 4 paveikslėlius matyti elektros energijos suvartojimas Lietuvoje skirtingais laikotarpiais. Grafikuose matyti, kad elektros poreikis vis auga. Buvo poreikio sumažėjimas, kai prasidėjo ekonominė krizė. [9] Metų laikotarpiui skirtingais metų laikais ar mėnesiais elektros poreikis skiriasi dėl tam tikrų konkrečių gamtinių sąlygų ar ekonominių priežasčių.



5 pav. Metinis elektros galios poreikis, jei dirbtu 100% apkrova



6 pav. Metinis elektros galios poreikis, jei dirbtu 25% apkrova

Remiantis 4 paveikslėliu 5 paveikslėlyje pateikta, kokios galios elektrinės reikėtų aptarnauti tą mėnesį, jei ji dirbtų 100% nenutraukiama apkrova, o 6 paveikslėlyje, jei dirbtų 25% nenutraukiama apkrova.

Remiantis [2] šaltiniu Lietuva 2011 pagamino ir importavo elektros:

Lietuva pagamino (TWh):	4,8
Lietuva importavo (TWh):	8,7
Iš viso (TWh):	13,5

Lentelė 1. Lietuvoje pagaminta ir importuota elektra 2011m.

Remiantis [2] šaltiniu Lietuvos instaliuojama galia: 4,0211 GW.

Remiantis [2] šaltiniu atlikus energetikos galios pagaminimo skaičiavimus, gauti tokie rezultatai:

- Jeigu Lietuvos elektrinės (4,0211 GW) dirbtų 100% apkrova, per metus sugebėtų pagaminti apie 35 TWh;
- 4,8 TWh pagaminti užtektų apie 0,55 GW galingumo, jei dirbtų 100% apkrova;
- 8,7 TWh pagaminti užtektų apie 1 GW galingumo, jei dirbtų 100% apkrova;
- 13,5 TWh pagaminti užtektų apie 1,55 GW galingumo, jei dirbtų 100% apkrova;
- Remiantis skaičiavimais visos Lietuvos elektrinės dirba apie 14% apkrova.

Remiantis [22] šaltiniu, teigiama, kad Visagino atominė elektrinė, būdama 1,35 GW galingumo, galėtų per metus pagaminti apie 10,5 TWh elektros.

Remiantis [4, 6, 11, 10, 13] šaltiniais apskaičiuota, kad Lietuvoje yra tokios elektrinių rūšys su tokiomis galiomis:

Elektrinės	GW
Biodujų elektrinės	1,9633
Hidroelektrinės	1,7272
Vėjo elektrinės	0,2162
Biokuro elektrinės	0,0424
Saulės elektrinės	0,0012

Lentelė 2. Lietuvos elektrinių rūšys ir jų galia

Remiantis [12] šaltiniu, vėjo jėgainių statyba yra prabanga, jas gali leisti tik pasiturinčios valstybės.

Saulės energija superkama skatinamąja kaina. Jų statyba aktuali verslininkams, kurie nori investuoti pinigus į ateitį. Norint sumažinti bendrą Lietuvos elektros kainą, saulės elektrinės nėra tinkamos. [19]

Remiantis [15, 21, 22,] šaltiniais nustatyta, kad elektrines pastatyti kainuotų:

Elektrinė	mln. Lt/ 1GW
Termofikacinė	3500
Kietojo kuro	10500
Branduolinė	12500
Hidroelektrinė	6000

Lentelė 3. Lietuvos elektinių statybos kainos 1GW galingumo

Remiantis [3, 15, 21, 22] šaltiniais nustatyta, kad elektrinių elektros gamybos kainos:

Elektrinė	Gamybos kaina ct/kWh
Termofikacinė	11,8
Kietojo kuro	15,28
Branduolinė	7
Hidroelektrinė	2,3

Lentelė 4. Elektros gamybos kaina

Elektros pirkimo kaina iš išorinių šaltinių 16,13 ct/kWh.

2.1.3. Matematinis programavimas

Matematinis programavimas yra tiesinės lygties išsprendimas su ribojimais. Prieš spręsdami uždavinį, pirmiausia turime susimodeliuoti norimą matematinį modelį. Turint jį, pasirenkame norimą sprendimo metodą.

Vieni iš galimų sprendimų metodų:

- Geometrinis sprendimas;
- Simplekso metodas;
- Modifikuotas simplekso metodas;
- M-metodas.

Matematinio programavimo matematinis modelis sudarytas iš dviejų dalių:

- Tikslų funkcija;
- Ribojimai.

Turint matematinį modelį galima ieškoti uždavinio ekstremumų, t. y. tikslo funkcijos mažiausias ar didžiausias galimas reikšmes, taip pat ir tikslo funkcijos nežinomuosius parametrus. [23]

2.1.4. Stochastinis programavimas

Stochastinis programavimas (toliau SP) yra matematinio programavimo dalis. SP uždavinio modelis yra matematinio programavimo modelis, kuriame yra tikimybinio pobūdžio kintamųjų. SP skirstomas į tiesinį ir netiesinį programavimą.

Tiesinis SP yra tiesinės lygties sprendimas su ribojimais ir atsitiktiniais dydžiais. Tikimybiniai kintamieji būna apriboti pagal kokią nors pasirinktą tikimybę ir pasklidimo aibę. Tikslų funkcija ir ribojimai susidaro iš tiesinių lygčių.

Tiesinis ir netiesinis programavimas gali būti skirstomas į tokias dalis:

- Vieno etapo uždavinys;
- Dviejų etapų uždavinys;
- Daugelio etapų uždavinys.

Vieno etapo uždavinys yra sprendžiamas matematinio modeliu su tikimybinio pobūdžio kintamaisiais.

Dviejų etapų uždavinys yra taip pat sprendžiamas matematinio modeliu su tikimybiniais kintamaisiais, bet jį sudaro papildomas antro etapo modelis. Antro etapo modelį sudaro antro etapo vektorius ir jo ribojimai. Sprendžiant realų uždavinį pasitaiko, kad keičiasi planai ir turi keistis uždavinys. Šiuo atveju yra naudojamas dviejų etapų uždavinys, jo pagalba galima rasti optimalų variantą, kai uždavinio modelis susidaro iš dviejų etapų.

Daugelio etapų uždavinys. Šis metodas naudojamas tada, kai kas kiekvienu etapu gali keistis poreikis, tarkim įmonė atlieka kažkokius darbus ir ties kiekvienu nauju darbu atsiranda naujo užsakovo skirtingi poreikiai. Tokiu atveju kas kiekvienu etapu koreguojami naujo etapo parametrai. Šiuo atveju turi būti kintamasis, kuris lemia etapus.

Netiesinio programavimo uždavinys – tai uždavinio modelis, kuriame naudojamos netiesinės lygtys, tai gali būti netiesinio pobūdžio ribojimai ar tikslo funkcija. Tikimybinio pobūdžio kintamieji gali būti naudojami ne vien tik ribojimuose, bet ir tikslo funkcijoje.

SP tiesinio, vieno etapo, matematinio modelio pavyzdys:

$$\text{Tikslo funkcija: } \sum_{i=0}^{10} c_i x_i \rightarrow \min ,$$

kai turi tokius ribojimus:

$$x \geq 0 ,$$

$$Ax \leq b(\omega) .$$

Kintamųjų reikšmės:

c – tikslo funkcijos x parametų koeficientų vektorius;

x – tikslo funkcijos nežinomųjų parametų vektorius;

A – ribojimų matrica;

$b(\omega)$ – atsitiktinių skaičių vektorius;

ω – atsitiktinio parametro reikšmė.

SP tiesinio, dviejų etapų, matematinio modelio pavyzdys:

$$F(x) = \sum c x + \sum q y(x, \omega) \rightarrow \min,$$

kai

$$Ax = b,$$

$$Tx + Wy = d(\omega),$$

$$x \geq 0, y \geq 0$$

Kintamųjų reikšmės:

$F(x)$ – tikslo funkcija;

$\sum c x$ – pirmas etapas;

$\sum q y(x, \omega)$ – antras etapas;

x – vektoriaus reikšmė;

ω – atsitiktinio parametro reikšmė;

A, T, W, b – ribojamų matricos (gali būti atsitiktiniai);

$d(\omega)$ – dispersija.

[24]

2.1.5. Atsitiktiniai skaičiai

Atsitiktinumas – tai yra koks nors įvykis su įvykio tikimybe. Pavyzdžiui, jei mesime lošimo kauliuką, tai, kad iškristų tam tikras skaičius, tarkim 1, tai tikimybė 1 iš 6. Jei mesime kelis kartus, gausime mestų kauliukų seką. Tarkime, mesdami kauliuką šešis kartus, nebūtinai gali į tą seką pakliūti visi galimi variantai, jie gali ir kartotis.

Sprendžiant sudėtingus uždavinius, kur naudojami ir atsitiktiniai skaičiai, yra svarbu naudoti realią situaciją imituojančius atsitiktinius skaičius (toliau a. s.). Kaip pirmai minėta žmogaus mestas lošimo kauliukas imituoja realią situaciją. Tokie a. s. vadinami *tikrieji atsitiktiniai skaičiai*. Bet kai reikia a. s. aibės tūkstančio ar milijono kintamųjų, tada tokią aibę sudaryti gali būti per brangus ir per daug reikalaujantis laiko procesas. Šitai problemai išspręsti pasitelkiamas kompiuteris. Jo pagalba milijono a. s. aibę galima sugeneruoti žymiai greičiau. Kompiuteriu generuojami a. s. turi baigtinę a. s. aibę. Supaprastinus a. s. po tam tikro skaičiaus sekos prasidės kartotis ta pati a. s. seka. Kad a. s. sekos aibė būtų ilgesnė, naudojami papildomi algoritmai. Kompiuteriu generuojami atsitiktiniai skaičiai, vadinami *pseudo atsitiktiniais skaičiais*.

Egzistuoja ir *kvaziatsitiktiniai skaičiai* – tai tokie skaičiai, kurie nėra atsitiktinai sugeneruoti. Tai jau iš anksto turi sugeneruotą savo seką ir skirti, spręsti konkrečioms uždaviniais. [18]

2.1.6. Monte – Karlo metodas

Monte – Karlo metodas, tai statistinio modeliavimo dalis. Šį metodą sudaro statistinis modelis arba matematinis modelis su tikimybiniais skaičiais bei jo sprendimas. *Monte – Karlo* metodas pagrįstas kompiuteriniu modeliavimu, t.y. uždavinio sprendimas kompiuterio pagalba.

„Kompiuterinis modeliavimas gali būti sėkmingai panaudojamas:

- kai neįmanoma arba labai sudėtinga gauti duomenis apie realius procesus, tuomet modeliavimo rezultatai ir išvados yra būtini prognozėms apie objektą suformuluoti;
- kai nagrinėjamas objektas yra tiek sudėtingas, jog aprašyti jį matematiniais modeliais, turinčiais analizinius sprendinius, neįmanoma;
- kai sistema aprašyta matematinio modeliu, bet sprendinių, susijusių su modeliu, gavimas yra neįmanomas, pasinaudojus tik analitiniais metodais;
- kai neįmanoma arba labai brangu atlikti eksperimentus su objektu, modeliavimo rezultatai gali būti panaudoti testuojant alternatyvias hipotezes.“ [18]

Uždaviniai, kuriems spręsti gali būti pritaikytas *Monte – Karlo* metodas:“

- kartotinių integralų apskaičiavimas;
- aptarnavimo sistemų modeliavimas;
- stochastinis optimizavimas;
- tiesinių lygčių sistemų sprendimas;
- veiksmai su matricomis (atvirkštinės matricos radimas, tikrinių reikšmių ir tikrinių vektorių radimas);
- diferencialinių lygčių sprendimas (Dirichlė uždavinio sprendimas).“ [18]

Monte – Karlo metodo algoritmas:



7 pav. *Monte – Karlo* metodas

Pagal 7 paveikslėlį matyti *Monte – Karlo* metodo algoritmas. Dažniausiai algoritmas atlieka iteracijas, kol tenkina nustatytą tikslumą (rezultatą).

2.2. Darbinės srities analizė

2.2.1. Stochastinio programavimo uždavinio sprendimas

Programai rašyti pasirinkta darbo vadovo doc. dr. Kęstučio Žilinsko daktaro disertacijoje aprašytas algoritmas, bei programa.

Programa rašyta C++ programavimo kalba. Naudotas uždavinio modelio nuskaitymas iš failo ir tekstinė sąsaja. Taip pat naudojama struktūrinio programavimo paradigma ir rodyklės su funkcija, kuri geba iš nurodytos vietos persokti į kitą nurodytą vietą (GOTO). Prieduose yra pateiktas tikslesnis programos aprašymas „Analizuojama programa“.

Uždavinio modelis, kuris pateiktas šioje programoje, skaičiuoja, tarsi statytume viską iš naujo. Naujam modeliui papildomai reikės įvertinti jau esamus elektrinių pajėgumus.

2.2.2. Energetikos uždavinys

Turint programą, reikalingas jau minėtas energetikos uždavinio modelis. Statant elektrinę reikia žinoti, kiek kainuoja jos pastatymas, kokia kaina ji gamina elektrą. Elektrinės gali būti kelių rūšių. Vadinasi, reikia sužinoti, kokios gali būti elektrinės ir jų pastatymo kainos. Gali būti jau pastatytų elektrinių. Taip pat vietoj gamybos galima pirkti elektrą. Vadinasi, reikia įvertinti, ar apsimoka statyti, ar tiesiog pirkti elektrą iš kitur. Dar vienas svarbus kriterijus, reikia žinoti, kiek elektros reikės, kad elektros gaminimas būtų optimalus. Elektrą gaminti galima pasitelkus gamtinius išteklius, vadinasi, reikia atsižvelgti ir į gamtinių išteklių galimybes ir ribojimus. Iš tų visų kriterijų turi susidaryti energetikos modelis bei reikiamose vietose kintantys parametrai, pavyzdžiui, kintantis elektros poreikis. Kad būtų tinkamai išspręstas energetikos uždavinys, turi būti parinktas tinkamas optimizavimo metodas.

2.1. „Temos analizė“ dalyje jau buvo nagrinėti iškelti klausimai šiam uždaviniui. Rasti energetikos uždaviniui reikalingi tam tikri duomenys. Optimizavimo metodui pasirinktas dviejų etapų tiesinis stochastinis optimizavimo metodas ir jo uždavinio modelis.

Šiam uždaviniui viena iš svarbių dalių yra elektros energijos poreikis skirtingu laiku. Žinomas faktas, kad elektrinės, gamindamos elektrą, negamina jos pilnu pajėgumu. To priežastys yra įvairios, kaip remonto darbai, gamtiniai ribojimai. Laikotarpius nuspręsta padalinti į penkias dalis, kurių galingumo poreikis skirstomas į tokias dalis:

- vidurkis;
- didžiausia galia;
- mažiausia galia;
- mediana;
- pirmo ir paskutinio mėnesio vidurkis.

2.3. Darbinės srities modelis

2.3.1. Uždavinio sprendimo modelis

Turime stochastinio programavimo uždavinį:

$$\sum_{i=1}^4 c_i x_i + E \left\{ \min_y \left(\sum_{i=1}^5 \sum_{j=1}^5 \sum_{k=1}^m q_i h_j y_{ijk\omega} \right) \right\} \rightarrow \min,$$

kai

$$\sum_{i=1}^4 c_i x_i \leq \text{invest},$$

$$x_4 \leq \text{hidro},$$

$$y_{ijk\omega} \leq x_i + p_i, \quad i=1,2,3,4, \quad \forall j,k,\omega,$$

$$\sum_{i=1}^5 y_{ijk\omega} \geq D_{jk\omega}, \quad \forall j,k,\omega,$$

$$x \geq 0, \quad y \geq 0,$$

kur

$x=(x_1,x_2,x_3,x_4)$ – elektros energijos gamybos stočių pajėgumo (GW) vektorius,

$y_{ijk\omega}$ – elektros energijos kiekis (GW), kur i -tojo tipo elektros gamybos stotis pagamina k -tųjų metų j -ąjį laikotarpį, kai elektros energijos poreikis $D_{jk\omega}$,

c_i – i -tojo tipo elektros energijos gamybos stoties statybos kaina (mln. Lt/GW),

q_i – i -tojo tipo stoties pagamintos energijos kaina (mln. Lt).

H_j – j -tojo laikotarpio ilgis (val.),

p_i – elektros energijos turimi pajėgumai (GW).

$D_{jk\omega}$ – elektros energijos poreikis (GW) k -tųjų metų j -tąjį laikotarpį, kuris atsitiktinis dydis $N(\mu, \sigma)$.

hidro – hidroelektrinės elektros pagaminimo riba GW į metus.

invest – investuojama suma milijonais litų.

m – kokiam laikotarpiui skaičiuos metais.

2.3.2 Uždavinio modelio duomenys

Uždavinio modelio duomenys susidaro iš tokiu lentelių:

Elektrinė	GW
Termofikacinė	2
Kietojo kuro	0,05
Branduolinė	0
Hidroelektrinė	1,8

Lentelė 5. Uždavinio modelio esami pajėgumai

Elektrinė	ct/kWh
Termofikacinė	11,8
Kietojo kuro	15,28
Branduolinė	7
Hidroelektrinė	2,3

Lentelė 6. Uždavinio modelio gamybos kaina

Elektrinė	mln. Lt/ 1GW
Termofikacinė	3500
Kietojo kuro	10500
Branduolinė	12500
Hidroelektrinė	6000

Lentelė 7. Uždavinio modelio statybos kaina

Laikotarpis	GW	h	Dispersija
1	8,75	730	0,1
2	6,72	730	0,1
3	7,7	4380	0,1
4	7	1460	0,1
5	8,61	1460	0,1

Lentelė 8. Elektros energijos poreikis ir dispersija

Hidroelektrinė naudoja gamtinius išteklius, šiuo atveju – vandenį. Tarkim galingumo riba parinkta 2 GW. Elektrą galima pirkti iš kitų valstybių. Pirkimo kaina bus 16,13 ct/kWh.

III. Projektinė dalis

3.1. Darbų eigos planas

Darbo numeris / mėnesiai	1	2	3	4	5	6	7	8	9	10
2011 rugsėjis										
2011 spalio										
2011 lapkritis										
2011 gruodis										
2012 sausis										
2012 vasaris										
2012 kovas										
2012 balandis										
2012 gegužė										
2012 birželis										
2012 liepa										
2012 rugpjūtis										
2012 rugsėjis										
2012 spalio										
2012 lapkritis										
2012 gruodis										
2013 sausis										
2013 vasaris										
2013 kovas										
2013 balandis										
2013 gegužė										

Lentelė 9. Darbų planas

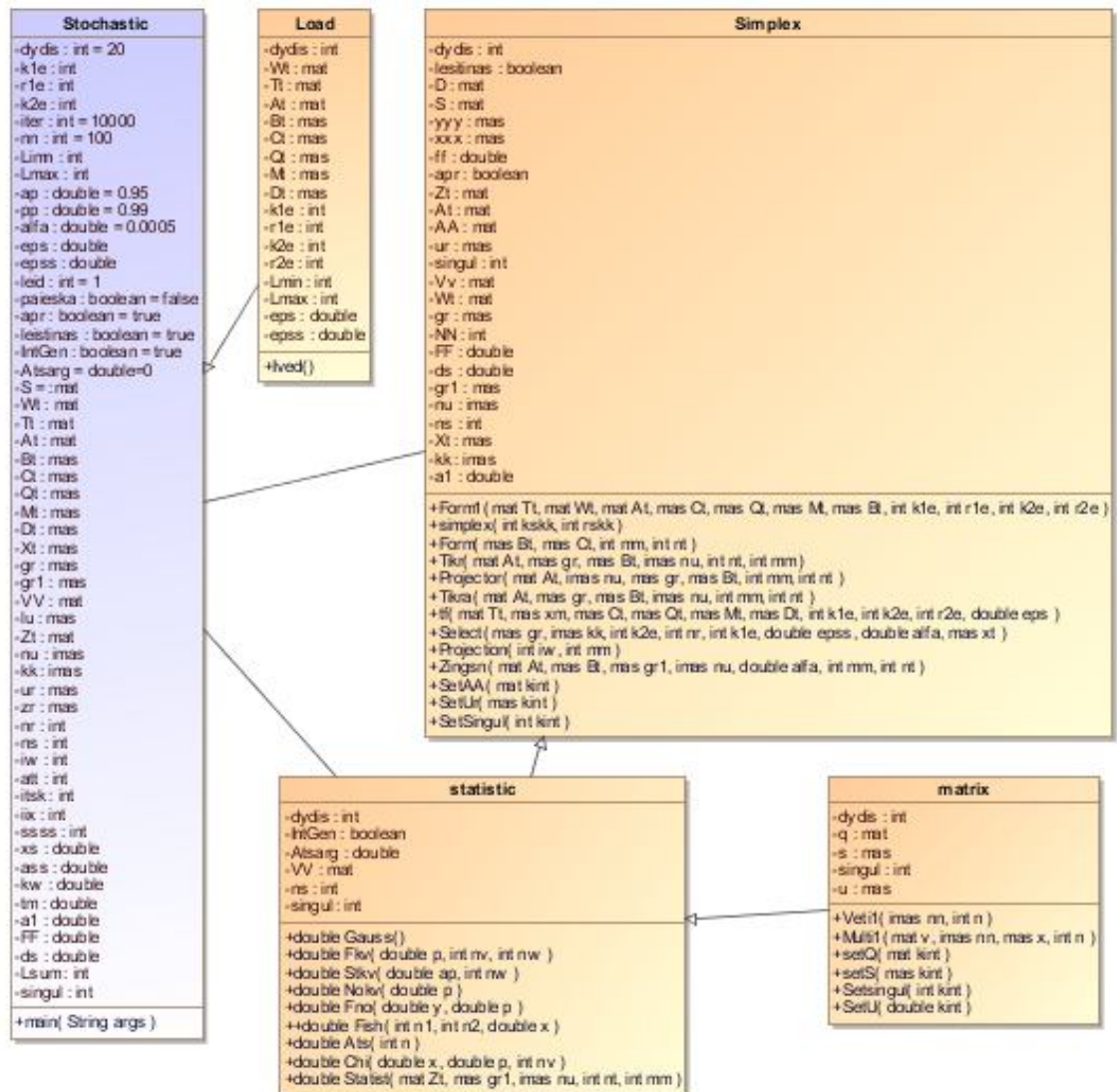
Darbai:

1. *Java* programavimo kalbos mokymasis;
2. Elektros poreikio ir kainos tyrimas;
3. Teorijos tyrimas;
4. Programos analizė;
5. Programos rašymas;
6. Programos tobulinimas;
7. Testavimas;
8. Tyrimų atlikimas;
9. Baigiamojo darbo rašymas;
10. Skaidrių ruošimas.

3.2. Pradinis projektas

3.2.1. Programa

Pradinis programos variantas:



8 pav. Pirmas Java programos variantas

Programos klasių dalys:

- *Stochastic* – pagrindinė programos dalis, kuri kviečiasi kitas klasių funkcijas ir atlieka visą skaičiavimų eilę arba kitaip *Monte – Karlo* metodo algoritmas.
- *Load* – tai klasė, kuri nuskaito duomenis iš failo.
- *Simplex* – klasė su visomis *Simplekso* metodo funkcijomis.
- *Statistic* – klasė atsakinga už statistinius skaičiavimus.
- *Matrix* – pagalbinė klasė matricoms pertvarkyti.

3.3. Įrankių ir priemonių pasirinkimas

Pasirinkti įrankiai:

- *Intellij IDEA* – Java programavimo aplinka;
- *MagicDraw UML* – skirta braižyti *UML* diagramoms;
- *Microsoft Excel 2010* – skirta paskaičiavimams ir grafikams.

Pasirinkta *Java* programavimo kalba, dėl savo plataus pritaikymo. Ji gali būti naudojama darbalaukio programai kurti, pavyzdžiui „*MagicDraw UML*“, aukšto lygio svetainėms kurti, kaip www.registrucentras.lt, www.google.com mobilių įrenginių programos, kurios išplitusios ant „*Android*“ platformos. Šiuo metu aktyviai tobulinami ir plečiami mobilieji įrenginiai. Nesudėtingus uždavinius būtų galima išspręsti ir su jais suskaičiuoti. *Java* programavimo kalba turi daug pagalbinių nemokamų bibliotekų ir labai gerą dokumentaciją bei forumų, kuriuose atsakoma į iškilusius klausimus.

Įrankių programoms kurti, buvo pasirinktas ne vienas. *Java* programavimo kalba reikėjo mokytis ir analizuoti.

Galimybės:	Intellij IDE	NetBeans	Eclipse
Palaiko <i>Java</i>	+	+	+
Kodo tikrinimas	+	+	+
<i>UML</i>	+	plugin	plugin
Kodo generatorius	+	+	+

Lentelė 10. Programų lyginimas

Visos trys programos atitinka reikalavimus ir turi patogią grafinę sąsają. Jos taip pat stebi kodą ir pažymi grubias programavimo klaidas. Programavimo metu pasirinktas *Intellij IDE* programavimo įrankis, nes turi patogų *UML* generatorių, kur kituose reiktų ieškotis ir įsirašyti papildinius. Turi iš anksto patogiai nustatytus greituosius funkcinius mygtukus.

Darbinio kompiuterio parametrai:

Procesorius:	Intel Core i7-2670QM 2.2 Ghz 4 tikri, 8 loginiai branduoliai.
Operatyvioji atmintis:	8 GB DDR3 dual 1333 Ghz magistralė .
Operacinė sistema:	Windows 7 Profesional 64 bitų.

Lentelė 11. Kompiuterio parametrai

Pastaba: Parametrai pateikiami, kad būtų galima tiksliau orientuotis į programos spartos skaičiavimą.

IV. Realizacijos dalis

4.1. Darbų eigos aprašymas

Darbo numeris / mėnesiai	1	2	3	4	5	6	7	8	9	10	11
2011 rugsėjis											
2011 spalio											
2011 lapkritis											
2011 gruodis											
2012 sausis											
2012 vasaris											
2012 kovas											
2012 balandis											
2012 gegužė											
2012 birželis											
2012 liepa											
2012 rugpjūtis											
2012 rugsėjis											
2012 spalio											
2012 lapkritis											
2012 gruodis											
2013 sausis											
2013 vasaris											
2013 kovas											
2013 balandis											
2013 gegužė											

Lentelė 12. Darbų eigos lentelė

Darbų sąrašas:

1. *Java* kalbos mokymasis;
2. Elektros energijos poreikio analizė;
3. Teorijos nagrinėjimas;
4. Programos nagrinėjimas;
5. Programos rašymas;
6. Programos testavimas;
7. Programos tobulinimas;
8. Uždavinio modelio sudarymas;
9. Uždavinio tyrimas;
10. Baigiamojo darbo aprašymo darymas;
11. Skaidrių rengimas.

4.2. Programos tobulinimas

4.2.1. Objektai

Programa gali suskaičiuoti nebūtinai energetikos uždavinį, ji gali ir kito tipo uždavinius spręsti, tik reikia tinkamai aprašyti sprendžiamą modelį. Nutarta programą papildyti dviem papildomomis klasėmis, remiantis objektinio programavimo paradigma. Reikia sukurti dvi papildomas klases: pirma klasė skirta sprendžiamo uždavinio modeliui, o antra – gauto rezultato uždavinio modeliui.



9 pav. Uždavinio pradinio modelio ir galutinio rezultato tobulinimas

Pavaizduoto paveikslėlio reikšmės:

StartObject – pradinis matematinis modelis.

EndObject – gautas rezultatas.

Stochastic – skaičiavimo mechanizmas.

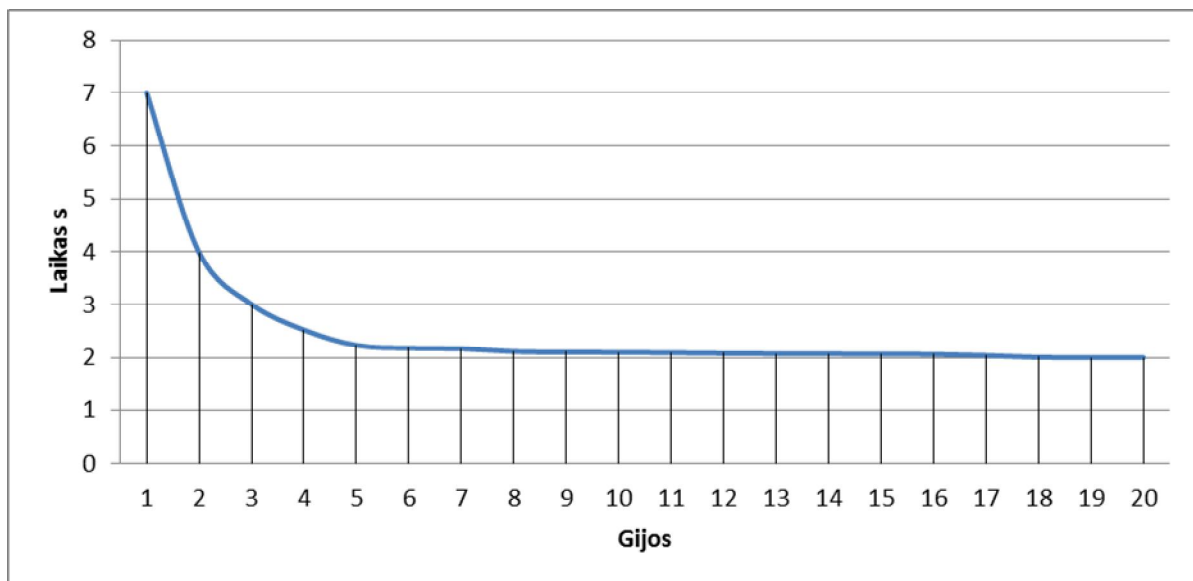
StartObject	EndObject
k1e int	dydis int
r1e int	Xt double[]
k2e int	gr double[]
r2e int	gr1 double[]
dydis int	FF double
Wt double[][]	ds double
Tt double[][]	nn int
At double[][]	Lsum int
Bt double[]	tt int
Ct double[]	ass double
Qt double[]	kw double
Mt double[]	k1e int
Dt double[]	r1e int
Lmin int	k2e int
Lmax int	r2e int
eps double	Nr int
epss double	mcn int
iter int	
ap double	
pp double	
alfa double	
nn int	
test boolean	

10 pav. Uždavinio modelio ir rezultato klasių kintamieji

<p><i>StartObject</i> kintamųjų reikšmės:</p> <ul style="list-style-type: none"> • <i>dydis</i> – nurodo kuriamų matricų dydį; • <i>Wt</i> – gamybos kainų matrica; • <i>Tt</i> – statybos ir gamybos ryšių matrica; • <i>At</i> – elektrinių statybos kainų matrica; • <i>Bt</i> – elektrinių statybos ribojimai. • <i>Ct</i> – elektrinių kainos; • <i>Qt</i> – elektrinių gamybos kainos; • <i>Mt</i> – elektrinių poreikių vidurkis; • <i>Dt</i> – dispersija; • <i>k1e k2e r1e r2e</i> – atitinkamų matricų didžiųjų koeficientai; • <i>Lmin</i> – mažiausias imties tūris; • <i>Lmax</i> – didžiausias imties tūris; • <i>eps</i> – norimo tikslumo skaičius; • <i>epss</i> – epsilon projektavimo koeficientas; • <i>Iter</i> – didžiausias iteracijų skaičius; • <i>ap</i> – pasikliautinojo intervalo tikimybė; • <i>pp</i> – optimalumo hipotezės tikimybė; • <i>alfa</i> – paieškos žingsnio ilgis; • <i>nn</i> – Monte – Karlo imties dydis; • <i>test</i> – nurodoma ar išvesti į ekraną tarpinius paskaičiavimus. 	<p><i>EndObject</i> kintamieji:</p> <ul style="list-style-type: none"> • <i>dydis</i> – matricų dydis; • <i>Xt</i> – ieškomos tikslo funkcijos reikšmės; • <i>gr</i> – gradientas; • <i>gr1</i> – projekcija į gradientą; • <i>FF</i> – tikslo funkcijos reikšmė; • <i>ds</i> – dispersija; • <i>nn</i> – Monte – Karlo imties dydis; • <i>Lsum</i> – scenarijų suma; • <i>tt</i> – Hotelingo T^2 – statistika t-oje iteracijoje; • <i>ass</i> – statistika; • <i>kw</i> – Fišerio kvantilis; • <i>k1e k2e r1e r2e</i> – atitinkamų matricų didžiųjų koeficientai; • <i>Nr</i> – paskutinis iteracijos skaičius; • <i>mcn</i> – Monte – Karlo imties tūris.
---	--

4.2.2. Programos sparta, gijos

Testuojant programą minėtu kompiuteriu, pastebėta, kad programa tenaudoja apie 13-15% kompiuterio procesoriaus. Atliekant ilgus skaičiavimus, nėra optimaliai išnaudojamas kompiuteris. Iškilo poreikis kompiuterio resursus išnaudoti optimaliai, t.y. procesorių išnaudoti 100%. Taip pat sugalvotas papildomas patobulinimas, t.y. dar viena klasė su visais reikalingais kintamaisiais, kuri bus paduodama į funkcijas ir grąžinama atgal. Tos klasės pagalba bus patogiu grąžinti iš funkcijos kelis kintamuosius. Sudarant grafiką, kad būtų tikslesni rezultatai, buvo atlikta po 30 bandymų (žr. Prieduose *Programos sparta*).



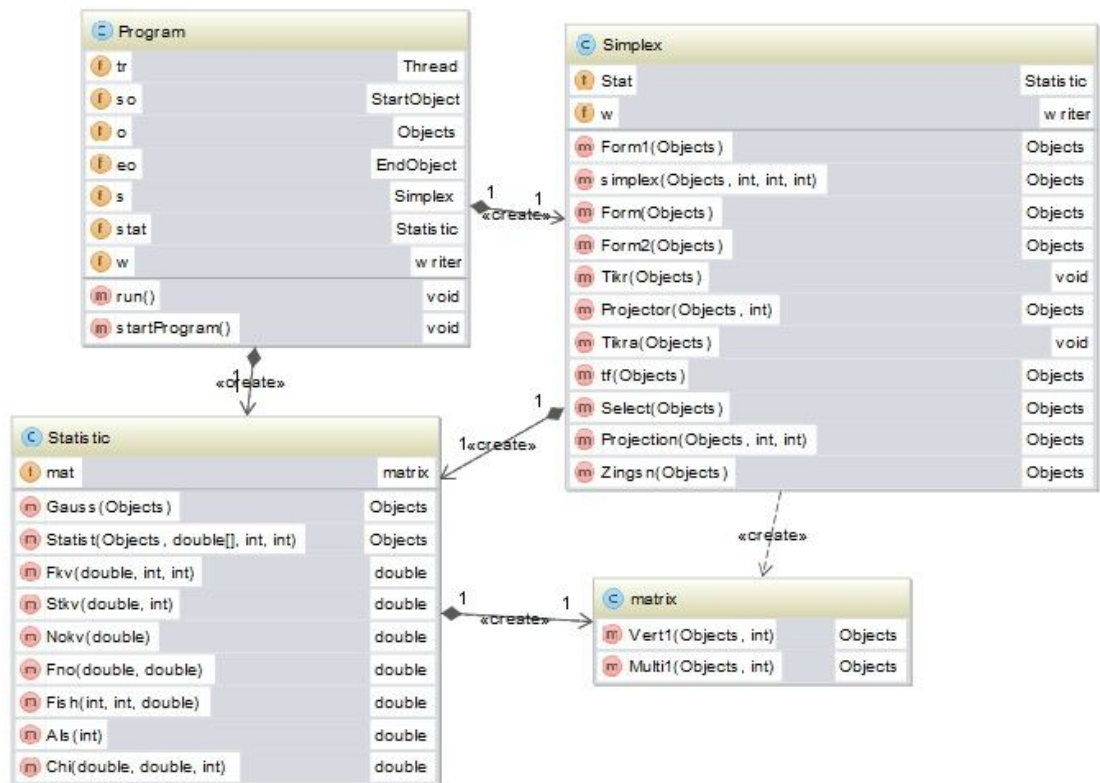
11 pav. Vienos gijos laiko vidurkis gijoms

Po padaryto patobulinimo pavyko išgauti iki 3,5 karto greitesni skaičiavimą. Testavimo metu pastebėta, kai skaičiuojamas didelis skaičius gijų, buvo situacijų, kai didesni gijų skaičių skaičiavo greičiau negu mažesni. To priežastis *Monte – Karlo* metodo vidinio skaičiavimo mechanizmas ir atsitiktinių skaičių generavimas.

4.2.3. Grafikai ir lentelės

Programos skaičiavimus tyrinėti nėra patogiu tekstinėje konsolėje. Todėl nutarta ieškoti patogesnio varianto, taip pat kilo poreikis grafikų piešimui, stebėti, kaip vyksta visas skaičiavimas. Vienas iš sprendimo būdų – naudoti duomenų bazę ir papildomas bibliotekas grafikams piešti. Pasirinktas kitas variantas – visą reikiamą informaciją išvesti į *CSV* (kableliais skirstomas reikšmes) failą. Tokį failą labai patogiu naudoti su *Microsoft Excel* informacija, kuri pateikiama lentelėse ir jos pagalba lengvai galima nusipiešti norimą grafiką.

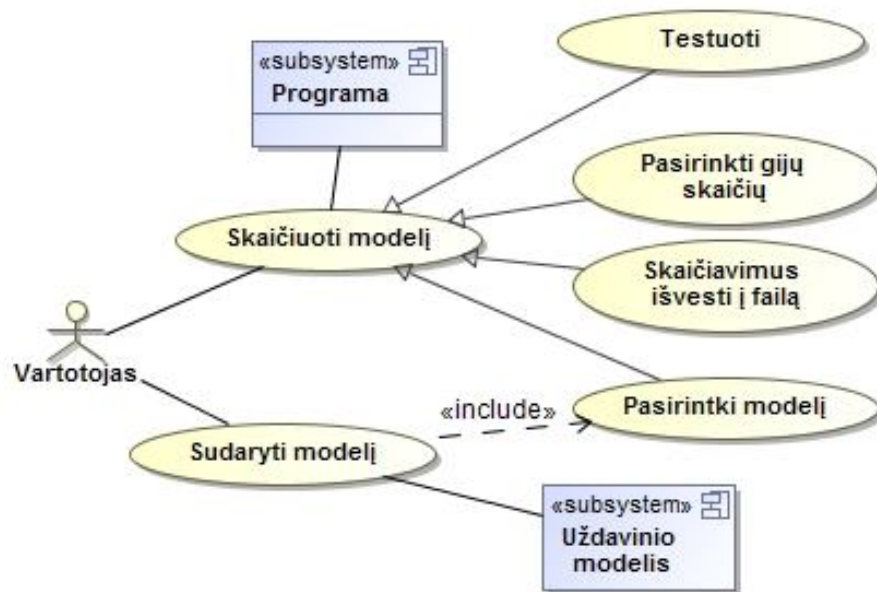
4.3. Galutinis projekto aprašymas



12 pav. Programos variklio schema

Programos klasės:

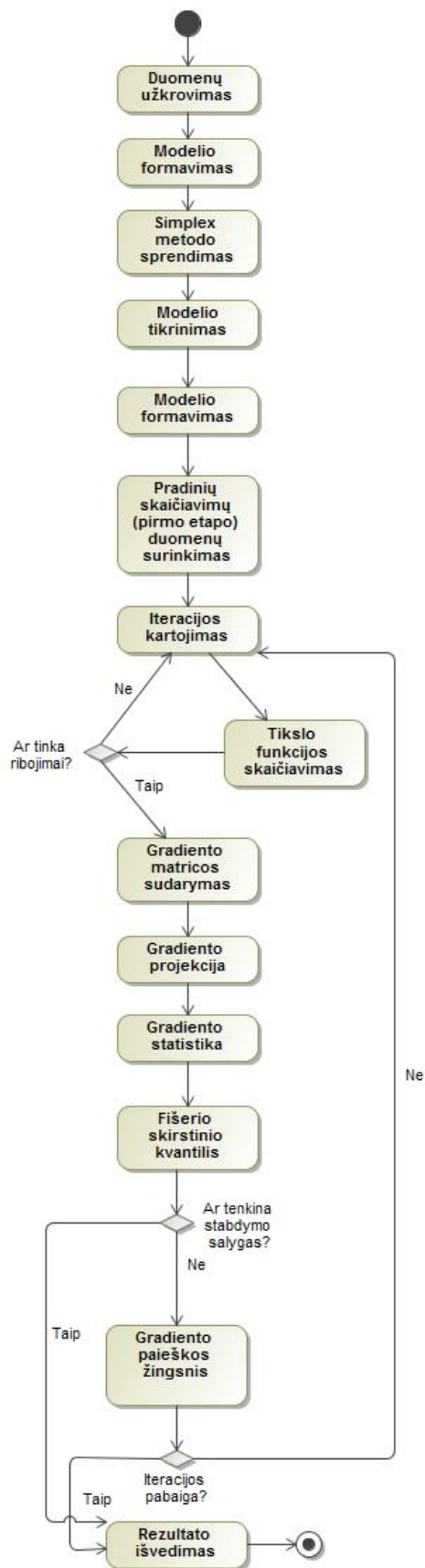
- *Program* – *Mote – Karlo* algoritmo skaičiavimo dalis, kartu su duomenų užkrovimo ir rezultato gavimo klase;
- *Statistic* – statistikos ir atsitiktinių skaičių generavimo klasė;
- *Matrix* – Matricų pertvarkymo klasė;
- *Simplex* – *Simplekso* klasė su visomis reikalingomis funkcijomis skirtomis skaičiuoti.



13 pav. Programos naudojimas

Pateiktame 13 paveikslėlyje vaizduojamas programos naudojimas. Jis susidaro iš dviejų dalių:

- Pirma dalis – *programa*, kuri sprendžia uždavinio modelį ir turi papildomus nustatymus.
- Antra dalis – *uždavinio modelis*, kuris sudaromas „Microsoft Excel“ šablono pagalba.



14 pav. Programos algoritmas

Paveikslėlyje pavaizduota, kaip veikia „Program“ klasės algoritmas.

4.4. Iškilusių problemų aprašymas

Iškilusios problemos ir jos sprendimas:

1. Programavimo kalbų nesuderinamumas. *Java* nepalaiko rodyklių ir „GOTO“ funkcijos. *Sprendimas*: Pertvarkyti programos struktūrą taip, kad nebenaudotų „GOTO“ ir rodyklių. Vietoj „GOTO“ naudoti pasikartojantį kodą, jei reikia ir „break“ (darbo nutraukimo) funkciją. Vietoj rodyklių perduoti parametrus per kitas funkcijas arba pasiimti iš objekto tiesiogiai.
2. Kompiuterio resursų neišnaudojimas. *Sprendimas*: Pasinaudoti Lygiagretaus programavimo paradigma, šiuo atveju gijų technologija. Kompiuteris galės vienu metu skaičiuoti kelis uždavinius, taip turėdamas galimybę apkrauti kompiuterį ir išnaudoti kompiuterio darbą optimaliai.
3. Gijų suderinamumas, kad programa galėtų skaičiuoti lygiagrečiai daugiau negu vieną giją. *Sprendimas*: Bandant paleisti daugiau negu vieną giją susidurta su problema. Programoje naudojami statiniai kintamieji, kurie skaičiuojant gijoms persidengia ir susimaišo visi skaičiavimai. Vietoj statinių kintamųjų perrašyti programą, kad naudotų tik dinaminis kintamuosius.
4. *Java* kalbos kintamųjų perdengimas pagal kintamojo adresą (keičiamos reikšmės, kurių nereikėtų). *Sprendimas*: *Javos* kalboje objektai perduodami ne objektais, o adresais, t.y. priskiriant objektą, paduodamas jo adresas, o ne reikšmė. Atlikus skaičiavimus su tuo objektu, grąžinamas vėl jo adresas. Jei tas naudojamas objektas keičiamas, grąžinamas jau pakeistas objektas. Čia neturėtų būti grąžintas pakeistas objektas. Šiai bėdai išspręsti buvo ieškomi tokie objektai. Vietoj realaus objekto būtina paduoti jo kopiją arba naudoti tarpinį objektą.
5. Energetikos uždavinio modelio informacijos rinkimas. *Sprendimas*: Papildomai naudoti užsienio svetainės, taip pat užsienio kainas ir analizuoti ir parinkti tarpinius apytikslus duomenis.

4.5. Darbo rezultatų analizė

Pavyko sudaryti apytikslį energetikos uždavinį.

Uždavinys nėra visiškai tikslus dėl to, kad:

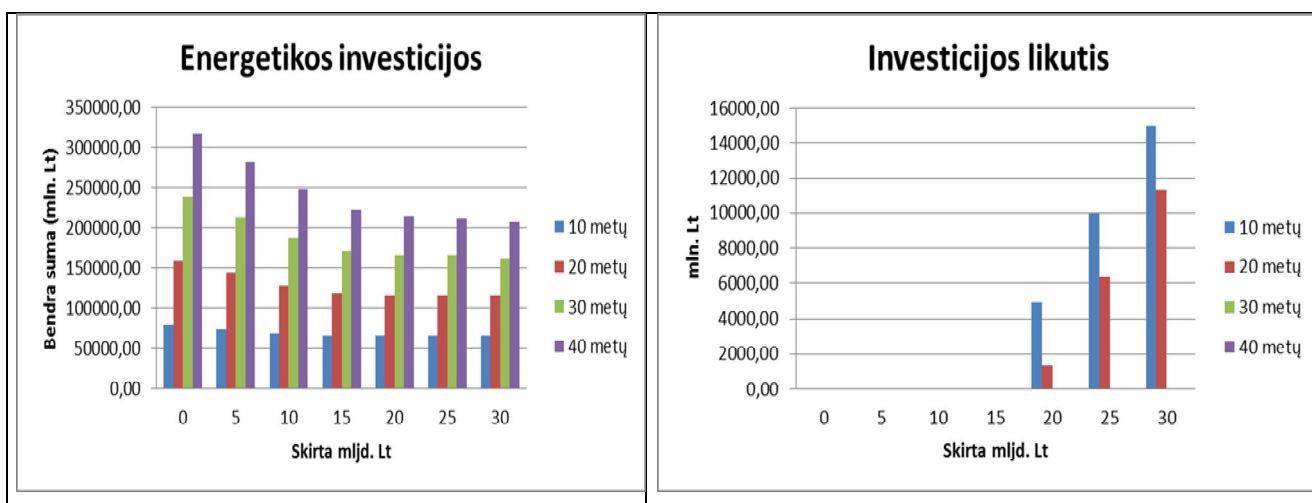
- Nėra naujausios informacijos vienoje vietoje;
- Yra technologijų įvairovė;
- Technologijos nuolat keičiasi;
- Vyksta žaliavų kainų kaita;
- Turi reikšmę pinigų infliacija;
- Riboti gamtiniai ištekliai ir jų kaita;
- Informacija nėra iki galo paviešinama.

Realizuota programa, kuri pajėgi vienu metu suskaičiuoti daugiau negu vieną giją (uždavinį).

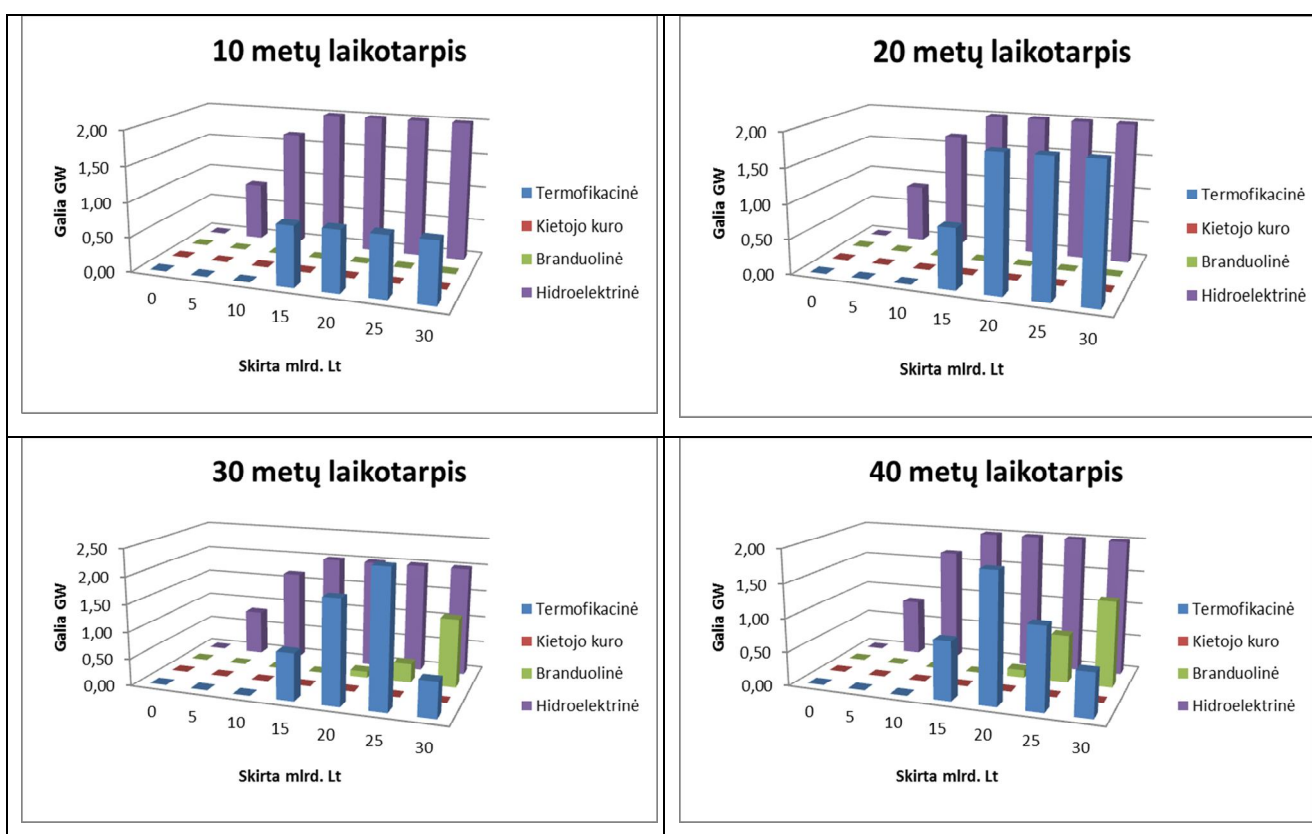
Skaičiuojant gautas toks rezultatas:

Metai:	Skirta mljd. Lt	0	5	10	15	20	25	30
10	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,86	0,87	0,87	0,87
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	0,00	4955,00	9955,00	14955,00
	Bendra suma (mln. Lt):	79604,26	74545,59	69418,66	67154,80	67189,79	67162,11	67177,68
20	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,86	1,90	1,90	1,90
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	0,37	1350,00	6350,00	11350,00
	Bendra suma (mln. Lt)	159247,44	144109,64	128864,24	119343,51	117240,97	117251,55	117255,96
30	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,86	1,87	2,45	0,64
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,12	0,35	1,26
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	0,31	0,09	0,19	0,00
	Bendra suma (mln. Lt)	238970,50	213447,35	188229,02	171521,79	166587,93	167257,74	162681,03
40	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,85	1,87	1,19	0,64
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,12	0,71	1,26
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	13,41	0,07	0,00	0,00
	Bendra suma (mln. Lt)	318328,54	282980,04	247776,66	223699,00	215512,81	211222,05	206962,24

Lentelė 13. Rezultatai



15 pav. Investicijų grafikai



16 pav. Elektrinių statyba

Pateiktuose 15 ir 16 paveikslėliuose matomas elektrinių statybų atsipirkimas ir kada, kokia ir kokios galios elektrines apsimoka statyti.

V. Rekomendacijos

Norint tiksliau išnaudoti kurtą sistemą, vertėtų paanalizuoti energetikos struktūrą skirtingomis sąlygomis. Tarkime, reikia paanalizuoti, kiek turėtų kainuoti statyba ir gamyba, kad apsimokėtų statyti tam tikrą elektrinę. Kurta taikomoji programa dar turi tobulinimo galimybes. Vietoj darbalaukio, programas galima realizuoti ir svetainės pavidalu. Nebūtinai galima nagrinėti Lietuvos energetikos struktūrą, galima bandyti sudaryti kitų šalių Energetikos modelį, taip pat palyginti kitų šalių energetikos gamybos situaciją, patikrinti, kurioje šalyje kokią elektrinę labiausiai apsimoka statyti.

Vienas iš siekių – sukurti universalią optimizavimo programą, su kuria būtų galima spręsti ne tik energetikos, bet ir kito pobūdžio uždavinius. Nuo paprastos lygties su ribojimais bei stochastiniais (atsitiktiniais) skaičiais, kurie lemia uždavinį, iki sudėtingų uždavinių, kurie savo sudėtingumu panašūs į energetikos uždavinį.

Panaudojus lygiagretaus programavimo paradigmą (gijas), kad kompiuteris vienu metu lygiagrečiai galėtų skaičiuoti daugiau negu vieną uždavinį. Sprendžiamas uždavinys gali būti iš kelių uždavinio modelių, kurį kompiuteris galės iki kelių kartų greičiau suskaičiuoti, priklausomai nuo kompiuteryje esančių branduolių skaičiaus.

VI. Išvados ir rezultatai

Darbo metu buvo:

- Sudarytas Lietuvos energetikos uždavinio modelis;
- Sukurta optimizavimo programa *Java* kalba, panaudojus objektinio ir lygiagreto programavimo paradigmas;
- Atlikti energetikos optimizavimo tyrimai.

Remiantis tyrimais, galima teigti, kad:

- Labiausiai verta statyti hidroelektrinę;
- Investuojant į energetikos struktūrą nuo 10 iki 20 metų laikotarpiui, optimali investicija iki 20 mlrd. litų.
- Investuojant į energetikos struktūrą 30 metų laikotarpiui, optimali investicija iki 25 mlrd. litų;
- Investuojant į energetikos struktūrą 40 metų laikotarpiui, optimali investicija nuo 30 ar daugiau mlrd. litų;
- Po hidroelektrinės statybos, kai biudžetas ne mažesnis nei 15 mlrd. litų ir investuojama nuo 10 metų laikotarpio, verta statyti termofikacinę elektrinę;
- Investuojant nuo 10 metų ir nuo 15 mlrd. litų verta statyti termofikacinę elektrinę.
- Investuojant 30 ar daugiau metų ir 20 ar daugiau mlrd. litų, verta statyti atominę elektrinę.

VII. Literatūros sąrašas

1. *Ar statyti atominę elektrinę, spręstis steigiama įmonė* [interaktyvus] [žiūrėta 2013m. gegužės 13d.]. Prieiga per internetą < <http://verslas.delfi.lt/energetika/ar-statyti-atomine-elektrine-spres-steigiama-imone.d?id=61286745>>.
2. *Elektros energetikos sistemos sandara* [interaktyvus] [žiūrėta 2013m. gegužės 2d.]. Prieiga per internetą: < http://www.enmin.lt/lt/activity/veiklos_kryptys/energijos_efektyvumas/elektros_ukis.php?sp_hrase_id=10548&clear_cache=Y>.
3. *Elektros energijos kaina ir tarifai* [interaktyvus] [žiūrėta 2012m. kovo 15d., 2013]. Prieiga per internetą: < <http://www.lrpresidentas.lt/node/320>>.
4. *Elektros gamyba* [interaktyvus] [žiūrėta 2013m. balandžio 12d.]. Prieiga per internetą: < <http://www.energin.lt/elektros-gamyba.html>>.
5. *Elektros vartojimas* [interaktyvus] [žiūrėta 2013m. gegužės 3d.]. Prieiga per internetą: < <http://www.energin.lt/elektros-vartojimas.html>>.
6. *Gamintojai, naudojantys atsinaujinančius energijos išteklius* [interaktyvus] [žiūrėta 2013m. gegužės 3d.]. Prieiga per internetą: < http://www.regula.lt/lt/elektra/gamintojai/gamintojai_1.php>.
7. *Ignalinos AE* [interaktyvus] [žiūrėta 2012m. gruodžio 12d.]. Prieiga per internetą < <http://www.vatesi.lt/index.php?id=11>>.
8. *Introduction to Parallel Computing* [interaktyvus] [žiūrėta 2013m. gegužės 3d.]. Prieiga per internetą < https://computing.llnl.gov/tutorials/parallel_comp/>.
9. S. Jakelūnas. *Krizė Lietuvoje: priežastys, padariniai ir pamokos* [interaktyvus] [žiūrėta 2013m. gegužės 14d.]. Prieiga per internetą <<http://jakeliunas.lt/krize-lietuvoje-priezastys-padariniai-ir-pamokos/>>.
10. *Kauno hidroelektrinė (KHE)* [interaktyvus] [žiūrėta 2013m. gegužės 4d.]. Prieiga per internetą < <http://www.le.lt/lt/veikla/elektros-gamyba/kauno-hidroelektrine-khe/>>.
11. *Kruonio hidroakumuliacinė elektrinė (KHAE)* [interaktyvus] [žiūrėta 2013m. gegužės 4d.]. Prieiga per internetą < <http://www.le.lt/lt/veikla/elektros-gamyba/kruonio-hidroakumuliacine-elektrine-kae/>>.
12. *Lietuvos elektrinė – garantas Lietuvos valstybei* [interaktyvus] [žiūrėta 2013m. gegužės 4d.]. Prieiga per internetą <http://www.madeinlithuania.lt/pramone-straipsnis241-Lietuvos_elektrine_%E2%80%93_garantas_Lietuvos_valstybei>.
13. *Lietuvos elektrinė (LEL)* [interaktyvus] [žiūrėta 2013m. gegužės 4d.]. Prieiga per internetą < <http://www.le.lt/lt/veikla/elektros-gamyba/lietuvoje-elektrine-lel/>>.

14. „Litgrid“: *elektros poreikis auga* [interaktyvus] [žiūrėta 2012m. gruodžio 20d.]. Prieiga per internetą: < <http://www.ekonomika.lt/naujiena/litgrid-elektros-poreikis-auga-16859.html>>.
15. *Mikrohidroelektrinės – papildomas pajamų šaltinis* [interaktyvus] [žiūrėta 2012m. sausio 7d.]. Prieiga per internetą < http://www.manoukis.lt/print_forms/print_st_z.php?s=2197&z=98>.
16. *Objektinis programavimas* [interaktyvus] [žiūrėta 2013m. gegužės 15d.]. Prieiga per internetą: < <http://www.mif.vu.lt/~ragaisis/Inflvadas/ObjProgr.htm>>.
17. *Prognozė: elektros poreikis išaugs dvigubai* [interaktyvus] [žiūrėta 2012m. spalio 5.]. Prieiga per internetą: < <http://www.valstietis.lt/Pradzia/Naujienos/Verslas/Prognoze-elektros-poreikis-isaugs-dvigubai>>.
18. L. Sakalauskas. *Statistinis modeliavimas ir analizė*. Šiauliai, 2012.
19. *Saulės elektrinės* [interaktyvus] [žiūrėta 2013m. gegužės 6d.]. Prieiga per internetą <<http://www.swbaltic.lt/elektros-energijos-gamyba/saul%C4%97s-elekttrin%C4%97s>>.
20. A. Sytas. R. *Kuodis: Atominės elektrinės kaina gali siekti 6 mlrd. Eurų* [interaktyvus] [žiūrėta 2013m. gegužės 6d.]. Prieiga per internetą: < <http://www.lrytas.lt/-13312827541328939058-r-kuodis-atomin%C4%97s-elekttrin%C4%97s-kaina-gali-siekti-6-mlrd-eur%C5%B3.htm>>.
21. *The Economics of Nuclear Power* [interaktyvus] [žiūrėta 2013m. gegužės 2d.]. Prieiga per internetą: < <http://www.world-nuclear.org/info/Economic-Aspects/Economics-of-Nuclear-Power/#.UYTpxbUdckR>>.
22. *Visaginos atomine elektrinė* [interaktyvus] [žiūrėta 2013m. gegužės 13d.]. Prieiga per internetą <<http://www.vae.lt/faktai/>>.
23. K. Žilinskas. *Matematinis programavimas. I dalis. Tiesinis programavimas*. Šiauliai, 2007.
24. K. Žilinskas. *Stochastinio tiesinio programavimo Monte Karlo metodu tyrimas*. Daktaro disertacija. Vilnius, 2007.

Anotacija

Vaitkus T. Energetikos stochastinio programavimo uždavinio tyrimas. Vadovas: doc. K. Žilinskas. – Šiaulių universitetas, Matematikos ir informatikos fakultetas, 2013.

Šio darbo tikslas – ištirti stochastinio programavimo energetikos uždavinį. Darbe tyrinėta Lietuvos energetikos struktūra, taip pat tyrinėti optimizavimo metodai. Sudarytas Lietuvos energetikos uždavinio modelis. Optimizavimo modelis realizuotas *Java* programa. Optimizavimo programoje panaudotos objektinio ir lygiagreto programavimo paradigmos. Atlikti energetikos uždavinio parametrų tyrimai.

Reikšminiai žodžiai: stochastinis programavimas, energetikos uždavinys, matematinis modelis, Monte – Karlo metodas.

Summary

Vaitkus T. Analysis of the energetic stochastic programming problem. Tutor: doc. K. Žilinskas. – Šiauliai University, Faculty of Mathematics and Informatics, 2013.

Primary goal of this work was to research stochastic programming model of the energetic system. In this paper we also studied Lithuanian energetic sector structure and made Lithuanian energetic model. This model was realized in Java programming language. Created program is written using object-oriented and parallel programming paradigms. Also we researched different parameters influence on the model.

Keywords: stochastic programming, the energy task model, mathematical model, Monte - Carlo method.

Priedai

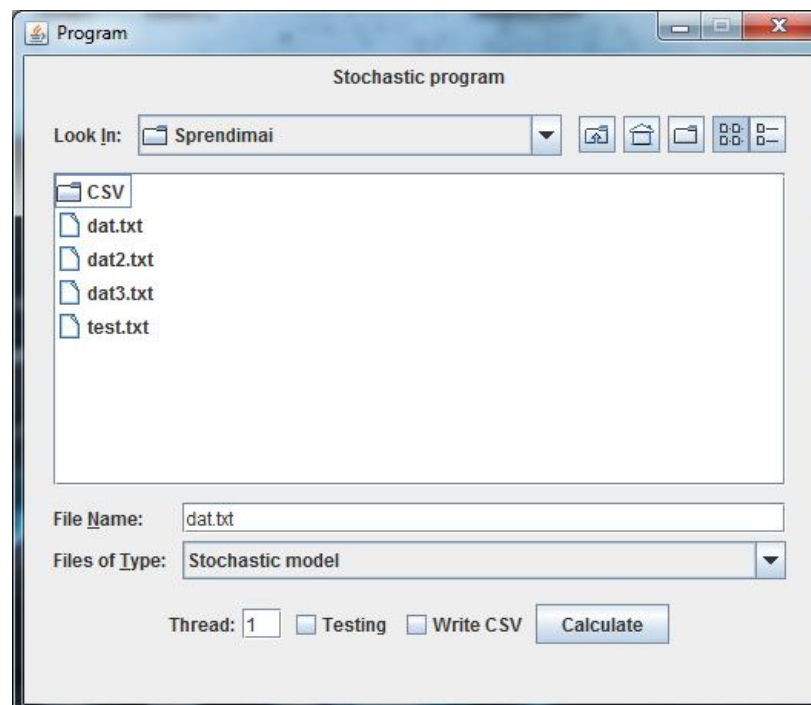
1. DVD turinys

Diske yra:

- Darbo aprašymas pdf ir doc;
- Kodas;
- Programa;
- Uždavinių modelis;
- Uždaviniai;
- CSV skaičiavimo rezultatai;
- Tyrimai;
- Instrukcija.

2. Programos naudojimas

Programą sudaro šios trys dalys:



17 pav. Programos valdymas

```

C:\Windows\system32\cmd.exe

-----
gradient:
1000,0000 3000,0000 6494,6749 9503,1700 0,0000 0,0000
-----
projection of gradient:
0,0010 0,0000 0,0000 0,0000 -1,0000 0,0000
-----
Nr=2 FF=583110,747+- 0,001 NN=1000000 Lsum=971 TT= 0,00 as=0,0000000260299 ==
-----
Calculation endet... sto.txt1
-----
point:
140,2507 0,0000 0,0000 0,0000 9749,2974 5,0000
-----
gradient:
1000,0000 3000,0000 6494,6749 9503,1700 0,0000 0,0000
-----
projection of gradient:
0,0010 0,0000 0,0000 0,0000 -1,0000 0,0000
-----
Nr=2 FF=583110,747+- 0,001 NN=84 Lsum=971 TT= 11,56 as=0,0000000260299
-----

```

18 pav. Programos išvedimo langas

Išvedimo lango reikšmės:

- *point* – x parametrai, šiuo atveju išvedami reikalingi elektrinių pajėgumai, investicijos likutis ir ribojimo likutis;
- *gradient* – gradientas, šiuo atveju elektrinių statybos kainos;
- *Projekction of gradient* – grafiendo projekcija;
- *Nr* – iteracijos skaičius;
- *FF* – tikslo funkcija;
- +- - tikslumas;
- *NN* – Monte – Karlo imties dydis;
- *Lsum* – scenarijų suma;
- *TT* – Hotelingo T^2 – statistika t-oje iteracijoje;
- *as* – statistika.

```

sto - Užrašinė
Failas Redaguoti Formatuoti Rodyti Žinynas
6      2
50     25
1000000
0,95
0,999
0,5
10000
10000  1000000
100    0,5

      1000,0  3000,0  5000,0  7000,0  0,0  0,0
      1000,0  3000,0  5000,0  7000,0  1,0  0,0
      0,0     0,0     0,0     1,0     0,0  1,0

      150000,0      5,0

3258,72 3744,9 1764,045      755,55 5913,0 2172,48 2496,6 1176,03 503,7
0      0      0      0      0      0      0      0      0      0
-1     0      0      0      0      0
0      -1     0      0      0      0
0      0      -1     0      0      0
0      0      0      -1     0      0
-1     0      0      0      0      0
0      -1     0      0      0      0
0      0      -1     0      0      0
0      0      0      -1     0      0
-1     0      0      0      0      0
0      -1     0      0      0      0
0      0      -1     0      0      0
0      0      0      -1     0      0
-1     0      0      0      0      0
0      -1     0      0      0      0
0      0      -1     0      0      0

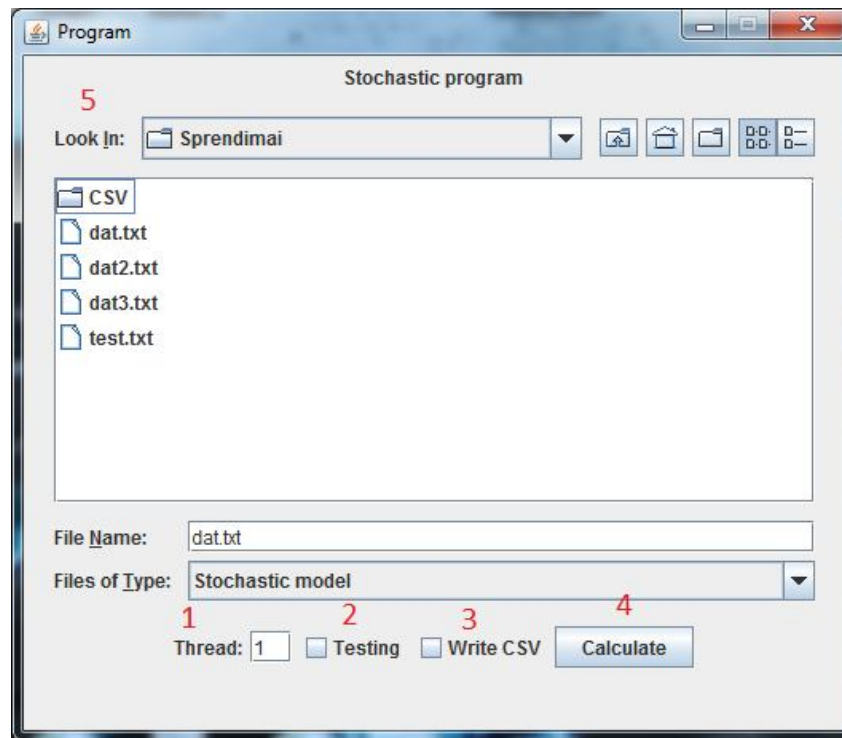
```

19 pav. Uždavinio modelis

Programos reikalavimai:

- *Java JDK*. Rekomenduojama 1.7 versija.
- Norint pajusti gijų naudą, reikia mažiausiai 2 branduolių kompiuterio. Rekomenduojama 4 branduolių ar daugiau.
- Testuota tik su *Windows 7* operacine sistema.

Programa paleidžiama per **run.bat** failą. Naudojama komanda: „java -jar JavaThread.jar“.



20 pav. Programos dalys

Programos pagrindinės dalys:

1. Nustatomas gijų skaičius;
2. Uždėjus varnelę, vedami tarpiniai kintamieji – testavimo režimas;
3. Uždėjus varnelę išvedami skaičiavimai į csv failą;
4. Paspaudus mygtuką, pradedama skaičiuoti.
5. Uždavinio (failo) susiradimo interfeisas.

Pastaba: Paspaudus kelis kartus mygtuką „Calculate“, pasirinktą uždavinio failą skaičiuos tiek kartų, kiek bus paspausta. Vienu metu galima skaičiuoti kelis skirtingus uždavinius. Jei nėra pasirinktas testavimo režimas, programa išves užkrautus duomenis ir galutinį suskaičiuotą rezultatą. Užkraunant uždavinį ir išvedant rezultatą, prie jų rašomas failo (uždavinio) pavadinimas ir gijos skaičius.

3. Uždavinio modelio sudarymas

Uždavinio modeliui sudaryti naudojamas *Microsoft Excel* programos šablonas. Naudojamos uždavinio lentelės iš minėto modelio. Sudarytas *Simpleks*o matematinis energetikos modelis ir jo skaičiavimas pagal pateiktus duomenis. Sudarytas energetikos modelis, su statistiniais kintamaisiais, skirtais stochastiniam uždaviniui spręsti. Keliamas į tekstinį failą ir skaičiuojama su programa.

Modelis naujas vad - Microsoft Excel

Failas Pagrindinis Įterpimas Puslapio maketas Formulės Duomenys Peržiūra Rodymas

Calibri 11

Įklijuoti

Šriftas Lygiuotė Skaičius

Bendra

Sąlyginis formatavimas

Formatuoti kaip lentelę

Langelių stiliai

Įterpti Naikinti Formatuoti

Langeliai

Rūšiuoti ir filtruoti

Rasti ir žymėti

Redagavimas

A1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1			Įterpti funkcija											
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														
35														
36														
37														
38														
39														
40														
41														
42														
43														
44														
45														

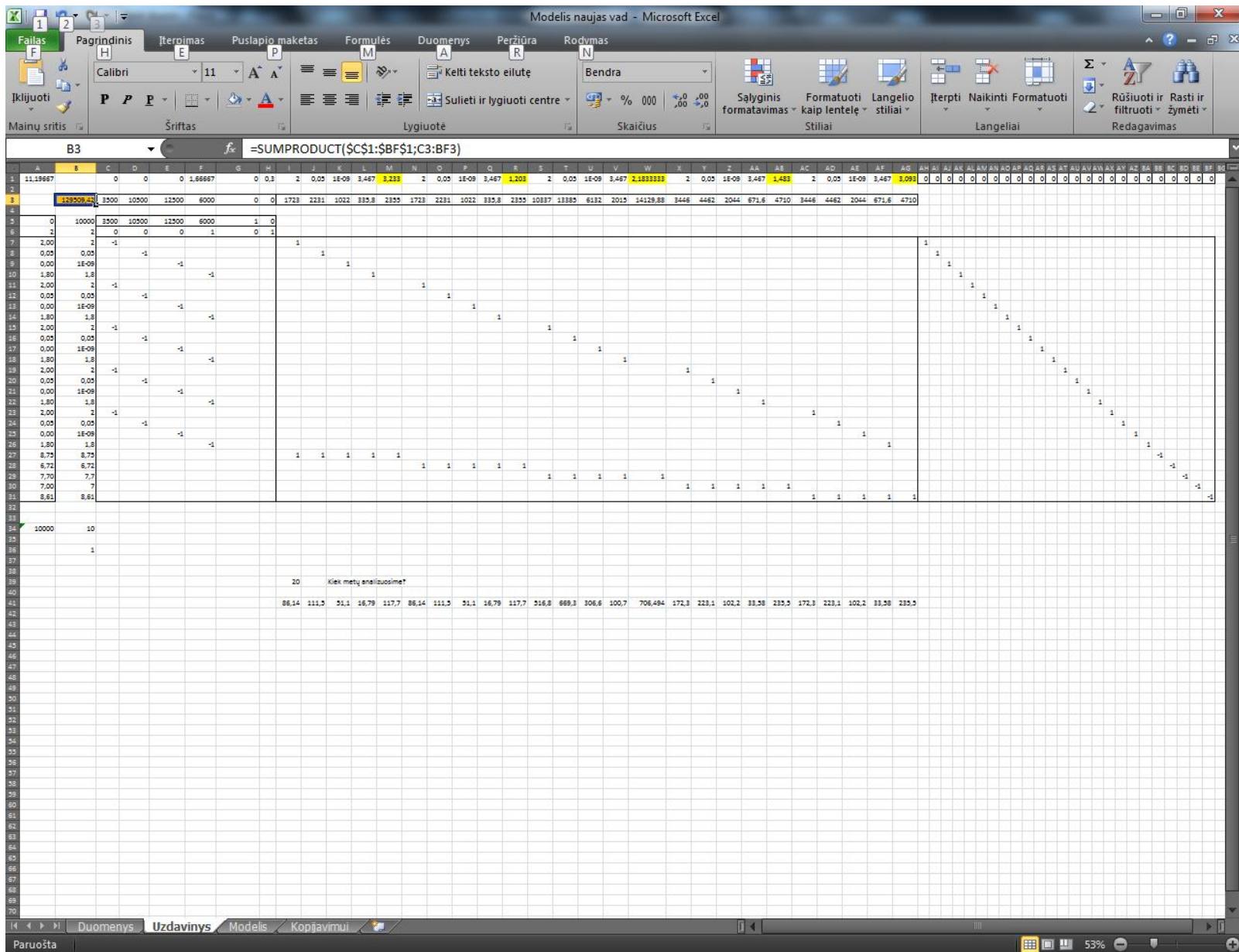
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45

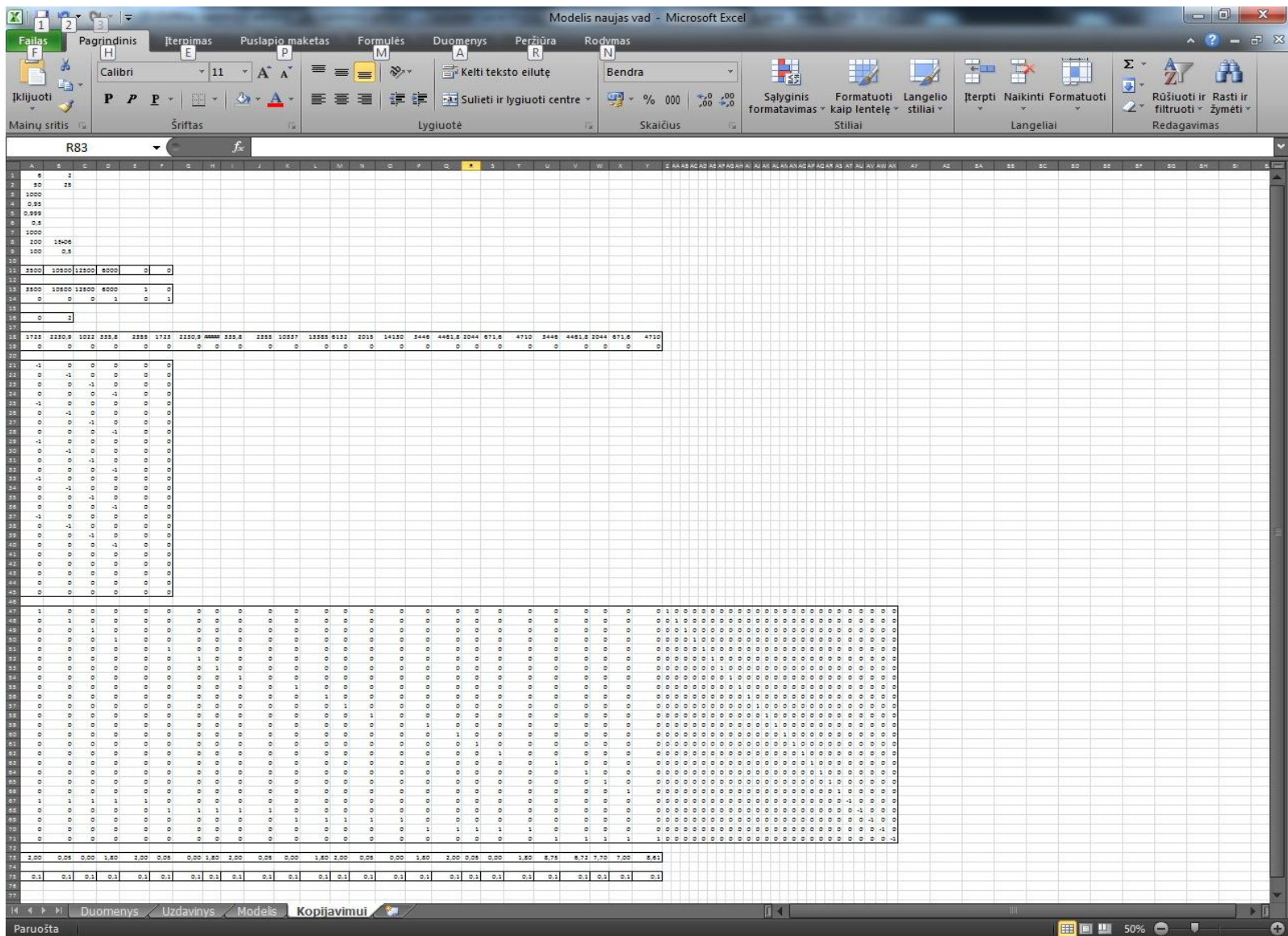
Redaguoti

85%

21 pav. Uždavinio modelio duomenys



22 pav. Uždavinio modelio duomenų matricos ir vektoriai



24 pav. Uždavinių modelis, skirtas kopijuoti

4. Programos testavimas

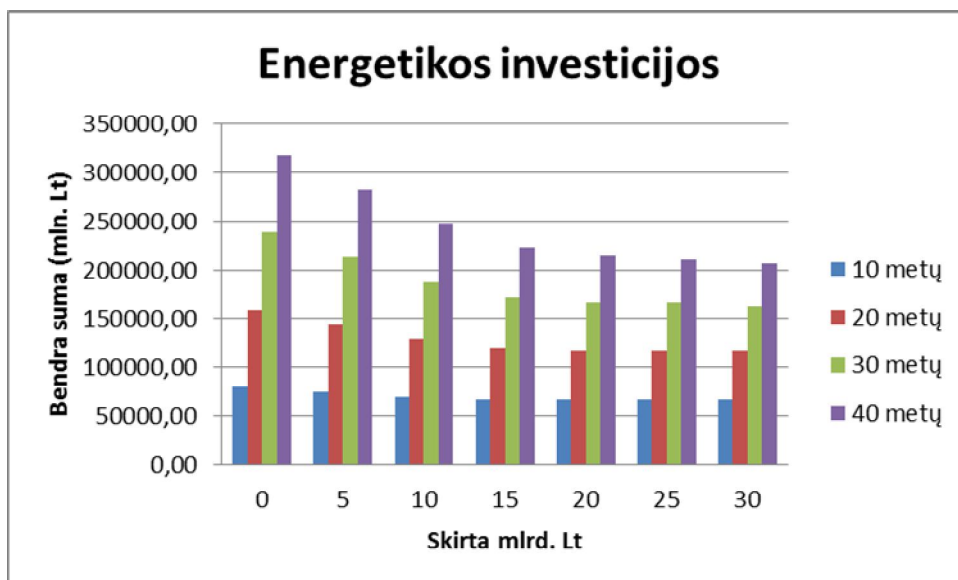
4.1. Elektrinių statyba

4.1.1. Duomenys

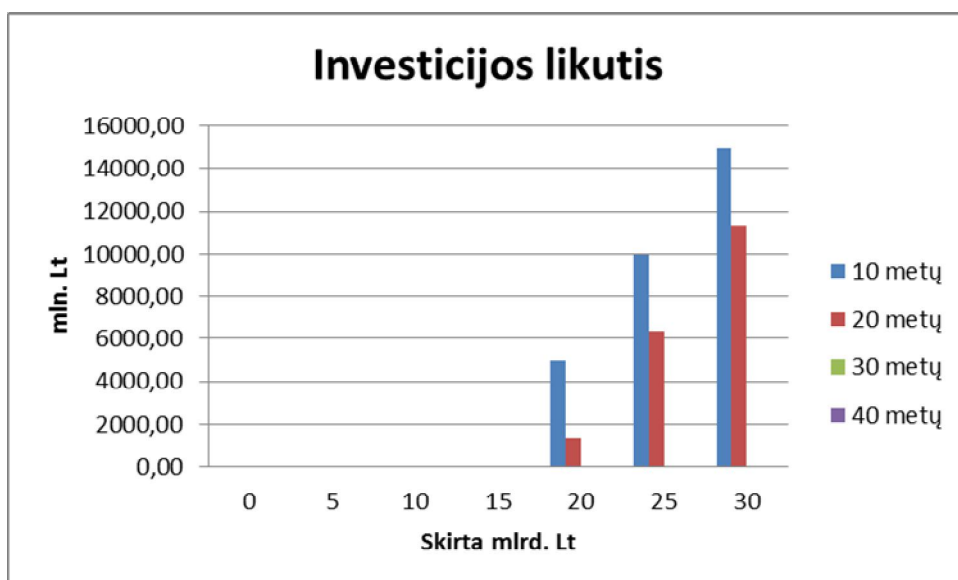
Testavimas pateiktas, kai hidroelektrinės riba 2 GW, dispersija 0,1. Kiekvienas skaičiavimas buvo atliktas po 30 kartų ir paimtas vidurkis.

Metai:	Skirta mljd. Lt	0	5	10	15	20	25	30
10	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,86	0,87	0,87	0,87
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	0,00	4955,00	9955,00	14955,00
	Bendra suma (mln. Lt):	79604,26	74545,59	69418,66	67154,80	67189,79	67162,11	67177,68
20	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,86	1,90	1,90	1,90
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	0,37	1350,00	6350,00	11350,00
	Bendra suma (mln. Lt)	159247,44	144109,64	128864,24	119343,51	117240,97	117251,55	117255,96
30	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,86	1,87	2,45	0,64
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,12	0,35	1,26
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	0,31	0,09	0,19	0,00
	Bendra suma (mln. Lt)	238970,50	213447,35	188229,02	171521,79	166587,93	167257,74	162681,03
40	Elektrinė	GW	GW	GW	GW	GW	GW	GW
	Termofikacinė	0,00	0,00	0,00	0,85	1,87	1,19	0,64
	Kietojo kuro	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Branduolinė	0,00	0,00	0,00	0,00	0,12	0,71	1,26
	Hidroelektrinė	0,00	0,83	1,67	2,00	2,00	2,00	2,00
	Liko (mln. Lt)	0,00	0,00	0,00	13,41	0,07	0,00	0,00
	Bendra suma (mln. Lt)	318328,54	282980,04	247776,66	223699,00	215512,81	211222,05	206962,24

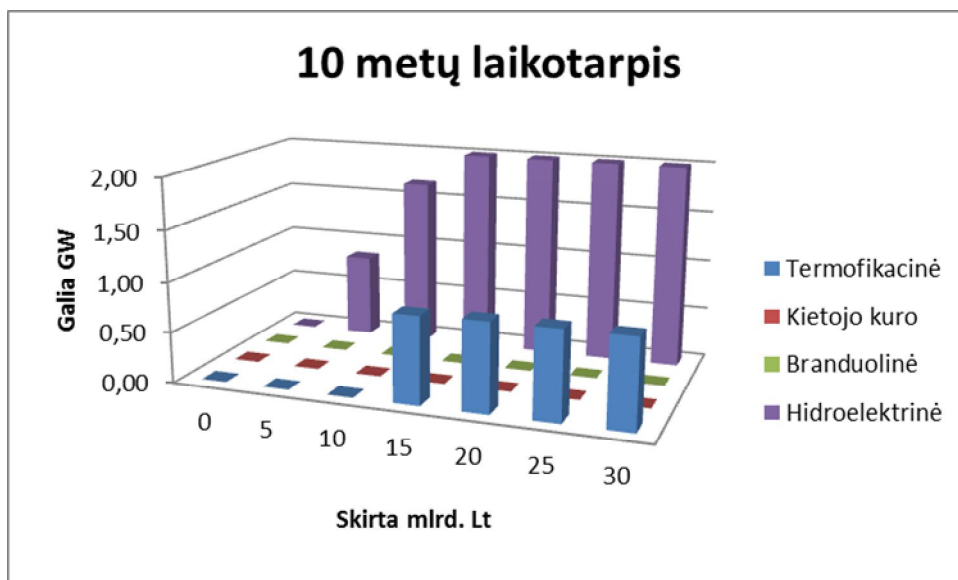
Lentelė 14. Elektrinių statybos pagal metus ir skirtą sumą



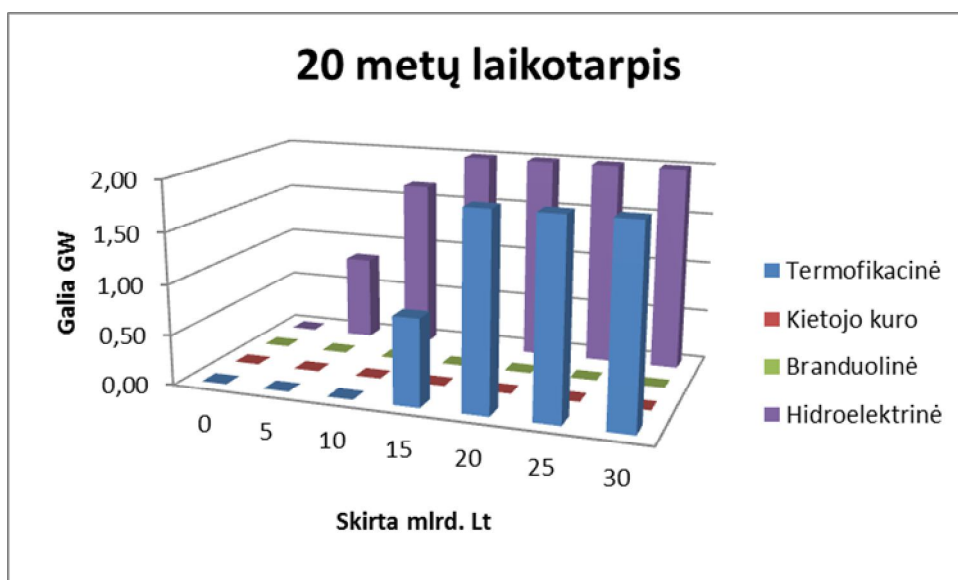
25 pav. Energetikos investicijos



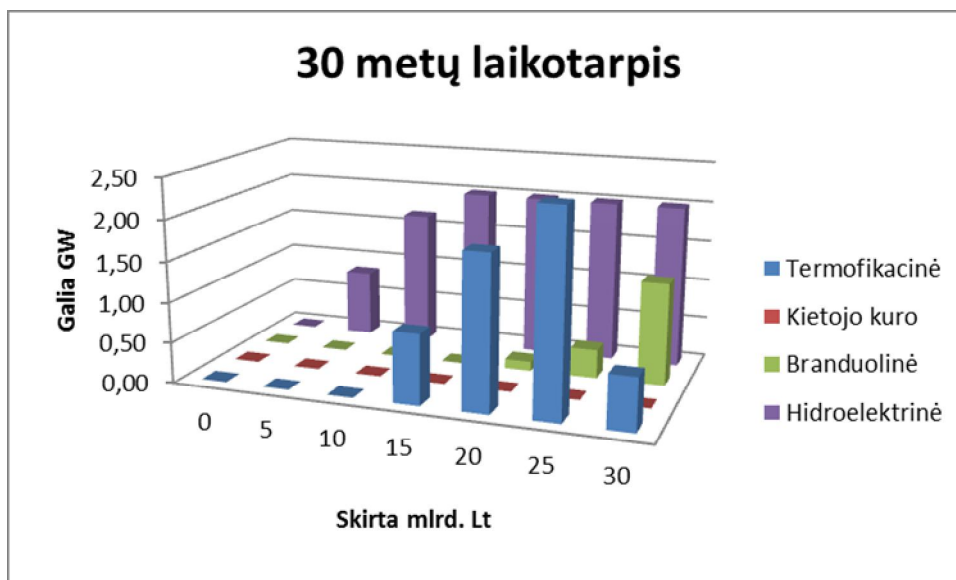
26 pav. Investicijos likutis



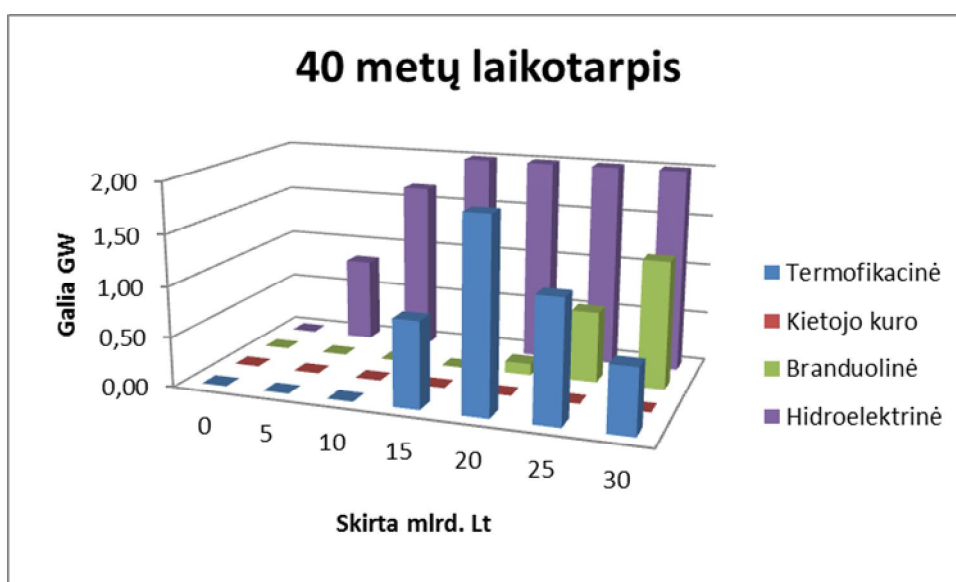
27 pav. Elektrinių statyba 10 metų laikotarpiui



28 pav. Elektrinių statyba 20 metų laikotarpiui



29 pav. Elektrinių statyba 30 metų laikotarpiui



30 pav. Elektrinių statyba 40 metų laikotarpiui

4.1.1. Išvados

Statant elektrines, matyti, kad parinktas skirtingas laikas ir investuojama suma turi įtaką uždavinio rezultatams. Priklausomai nuo investicijos ir laiko parenkamos atitinkamos elektrinės ir jų galios poreikis.

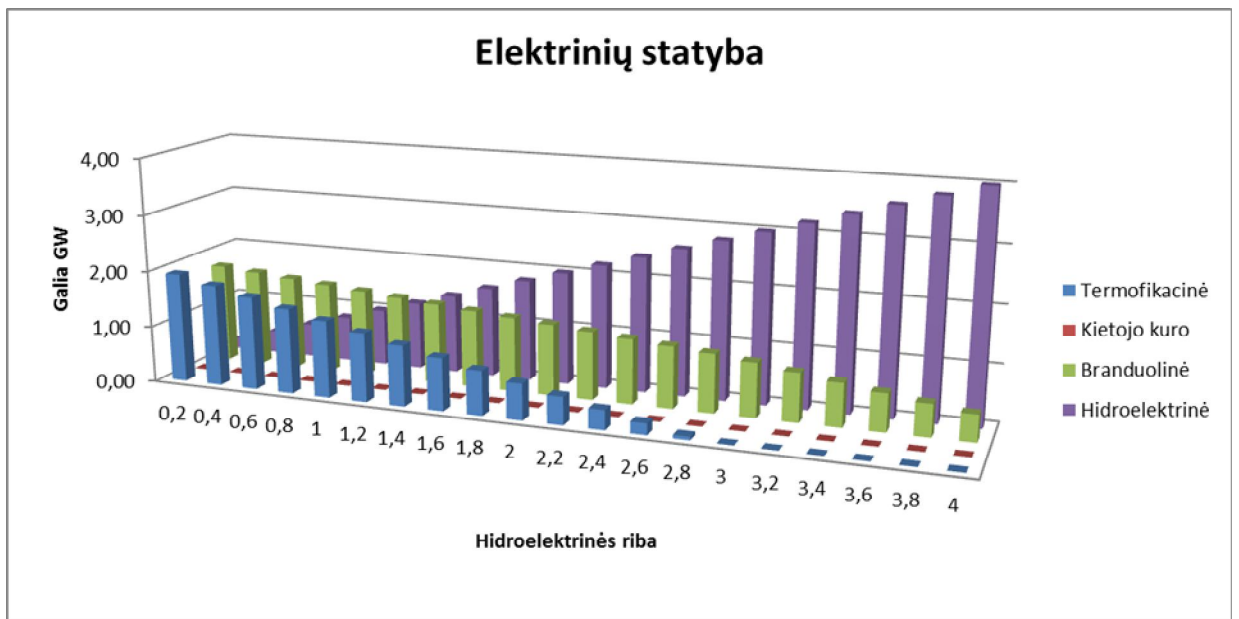
4.2. Hidroelektrinės ribojimo įtaka.

4.2.1. Duomenys

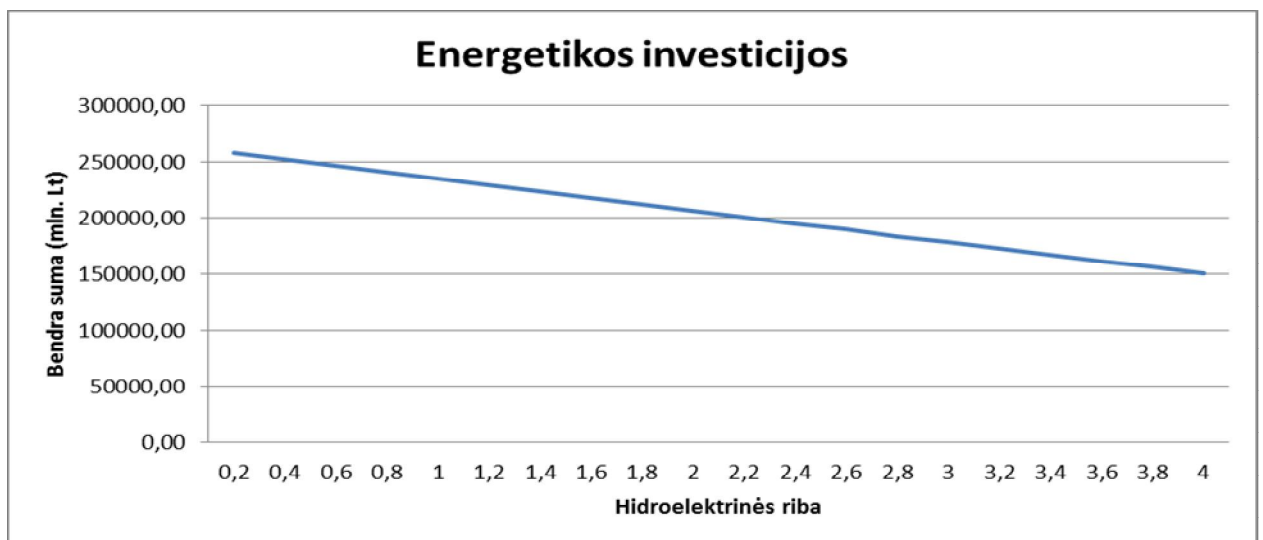
Pasirinkta, kai investuojama 30 mlrd. litų 40 metų laikotarpiu, kai hidroelektrinės riba kinta nuo 0,2 iki 4 GW įtaka. Kiekvienas rezultatas skaičiuotas po 30 kartų.

Hidroelektrinės riba (GW):	0,2	0,4	0,6	0,8	1
Elektrinė	GW	GW	GW	GW	GW
Termofikacinė	1,94	1,79	1,65	1,51	1,36
Kietojo kuro	0,00	0,00	0,00	0,00	0,00
Branduolinė	1,76	1,71	1,65	1,59	1,54
Hidroelektrinė	0,20	0,40	0,60	0,80	1,00
Liko (mln. Lt)	0,00	0,00	0,00	0,00	0,00
Bendra suma (mln. Lt):	258574,35	252572,82	246923,00	241233,67	235462,62
Hidroelektrinės riba (GW):	1,2	1,4	1,6	1,8	2
Elektrinė	GW	GW	GW	GW	GW
Termofikacinė	1,22	1,07	0,93	0,78	0,64
Kietojo kuro	0,00	0,00	0,00	0,00	0,00
Branduolinė	1,48	1,43	1,37	1,32	1,26
Hidroelektrinė	1,20	1,40	1,60	1,80	2,00
Liko (mln. Lt)	0,00	0,00	0,00	0,00	0,00
Bendra suma (mln. Lt):	229835,12	223979,98	218339,91	212569,36	206926,59
Hidroelektrinės riba (GW):	2,2	2,4	2,6	2,8	3
Elektrinė	GW	GW	GW	GW	GW
Termofikacinė	0,49	0,35	0,21	0,06	0,00
Kietojo kuro	0,00	0,00	0,00	0,00	0,00
Branduolinė	1,21	1,15	1,09	1,04	0,96
Hidroelektrinė	2,20	2,40	2,60	2,80	3,00
Liko (mln. Lt)	0,00	0,00	0,00	0,00	0,00
Bendra suma (mln. Lt):	201168,94	195411,06	189693,36	183908,30	178466,16
Hidroelektrinės riba (GW):	3,2	3,4	3,6	3,8	4
Elektrinė	GW	GW	GW	GW	GW
Termofikacinė	0,00	0,00	0,00	0,00	0,00
Kietojo kuro	0,00	0,00	0,00	0,00	0,00
Branduolinė	0,86	0,77	0,67	0,58	0,48
Hidroelektrinė	3,20	3,40	3,60	3,80	4,00
Liko (mln. Lt)	0,00	0,00	0,00	0,00	0,00
Bendra suma (mln. Lt):	172693,60	167191,02	161599,86	156264,88	150757,31

Lentelė 15. Elektrinių statyba pagal hidroelektrinės statybos ribojimą



31 pav. Elektrinių statyba pagal hidroelektrinės statybos ribojimą grafikas



32 pav. Energetikos investicija pagal hidroelektrinės statybos ribojimą grafikas

4.2.2. Išvados

Pagal skaičiavimo tyrimus matyti, kad parinktam uždavinio modeliui hidroelektrinė statoma visa leistina galia. Hidroelektrinės ribojimas turi įtaka kitų elektrinių statybai.

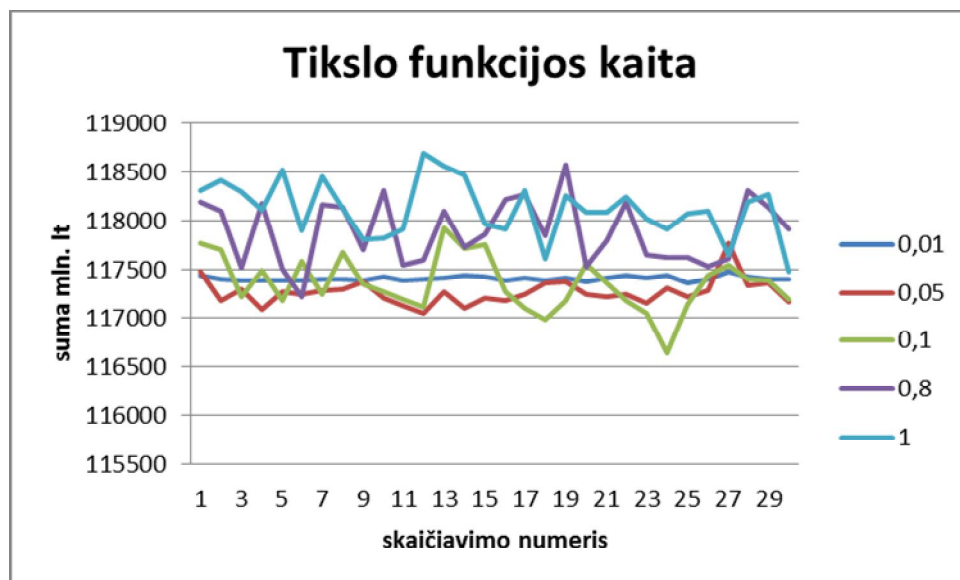
4.3. Dispersijos įtaka

4.3.1. Duomenys

Tiriant dispersijos įtaką uždaviniui, parinkta 20 metų laikotarpis, kai investuojama 25 mlrd. litų ir iki 500 mln. litų tikslumu:

Dispersija:	0,01	0,05	0,1	0,8	1	10
1	117437,934	117481,6983	117768,1487	118191,5052	118314,2496	157862,0349
2	117406,5592	117185,0136	117704,5135	118101,3605	118417,1397	157631,9294
3	117394,9198	117298,0015	117215,9141	117526,7348	118296,8761	157989,3748
4	117394,8488	117081,4197	117490,9334	118173,9851	118114,9865	157808,3482
5	117392,9977	117268,079	117176,3244	117509,7496	118518,8056	156754,6133
6	117390,5289	117247,7944	117588,2086	117222,8027	117906,0558	157511,6833
7	117398,8502	117290,5015	117244,9365	118167,1999	118456,7185	157650,6047
8	117401,6386	117296,2967	117680,929	118135,6242	118125,2803	157707,8848
9	117395,655	117376,9974	117356,9582	117700,1071	117812,8173	157731,8552
10	117431,6597	117206,4742	117266,6547	118311,4305	117828,0176	157760,9963
11	117394,6027	117124,2079	117198,8917	117550,1146	117913,4782	157114,0718
12	117398,7246	117045,2712	117109,3718	117599,6128	118689,7903	157876,9267
13	117415,4508	117277,1576	117931,5148	118105,8591	118562,9583	157265,5306
14	117442,9856	117096,1545	117719,4711	117726,7032	118463,8444	157754,1729
15	117426,0975	117206,2982	117759,9713	117858,8549	117967,7148	157652,1165
16	117396,5466	117180,0039	117268,317	118225,2767	117915,3773	157288,9288
17	117411,8703	117248,6908	117100,0541	118268,1153	118306,1768	156930,8726
18	117396,4083	117358,0171	116988,3405	117857,0033	117608,9319	157570,0167
19	117421,1808	117383,4924	117183,304	118569,8529	118264,4792	157568,4677
20	117379,4932	117246,4778	117565,3532	117541,3919	118082,3259	157542,5611
21	117414,6914	117217,7599	117363,328	117798,5651	118090,5333	157350,0891
22	117440,1098	117242,5139	117183,0602	118188,1632	118251,6317	157733,4746
23	117419,1244	117151,0803	117051,5548	117656,4118	118016,2471	157341,6888
24	117437,6068	117317,0793	116636,6963	117631,0503	117914,6314	157516,1224
25	117369,5783	117219,5963	117144,5639	117631,0353	118068,5232	157324,9562
26	117398,4624	117278,7522	117443,9226	117531,0954	118107,1939	157643,8112
27	117479,7973	117767,2655	117546,3853	117608,7526	117652,004	157965,1519
28	117432,7425	117333,0155	117404,7116	118315,2338	118186,3332	157614,0654
29	117408,9071	117364,9737	117391,3467	118135,1799	118277,4039	157795,5912
30	117403,7497	117166,9602	117198,8917	117920,3213	117479,5442	157861,0327

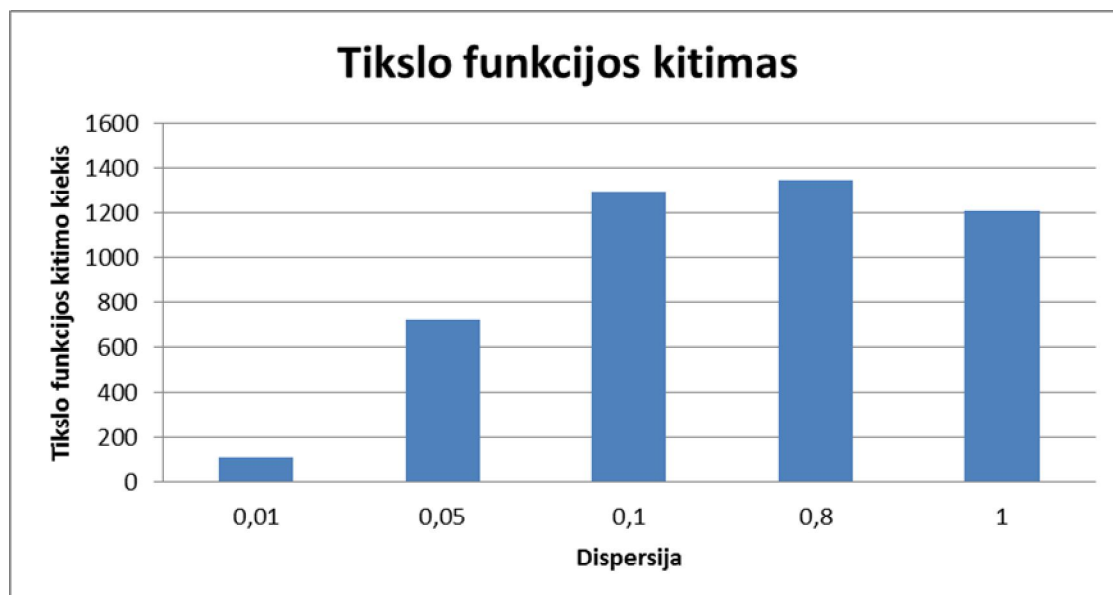
Lentelė 16. Tikslų funkcijos kaita pagal dispersiją



33 pav. Tikslo funkcijos radimas su dispersijos įtaka

Dispersija:	0,01	0,05	0,1	0,8	1
Mažiausia reikšmė:	117369,6	117045,3	116636,7	117222,8	117479,5
Didžiausia reikšmė:	117479,8	117767,3	117931,5	118569,9	118689,8
Skirtumas:	110,219	721,9943	1294,819	1347,05	1210,246

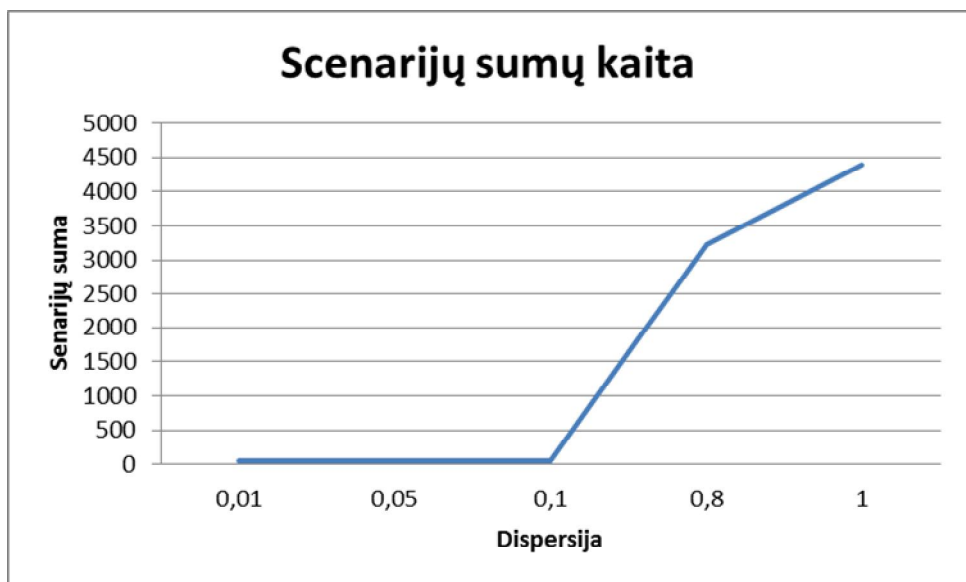
Lentelė 17. Tikslo funkcijos didžiausia mažiausia reikšmė, jų skirtumas pagal dispersija



34 pav. Energetikos investicija uždavinio sprendimo dispersijos įtaka

Dispersija:	0,01	0,05	0,1	0,8	1
Scenarijų suma:	50	50	50	3231	4389,1
Tikslumas:	37,93862	201,1062	413,8688	499,9112	499,9442607

Lentelė 18. Uždavinio scenarijų ir tikslumo reikšmės pagal dispersija



35 pav. Energetikos investicija uždavinio sprendimo dispersijos įtaka scenarijų skaičiui



36 pav. Energetikos investicija uždavinio sprendimo dispersijos įtaka tikslumui

4.3.2. Išvados

Pagal atliktus tyrimus, matyti, kad kuo didesnė dispersija, tuo didesnis skirtumas tarp paskaičiuotų rezultatų. Kuo didesnė disperija tuo daugiau laiko reikia paskaičiuoti (daugiau scenarijų). Tikslumo ribojimas įtakoja galutinį rezultatą.

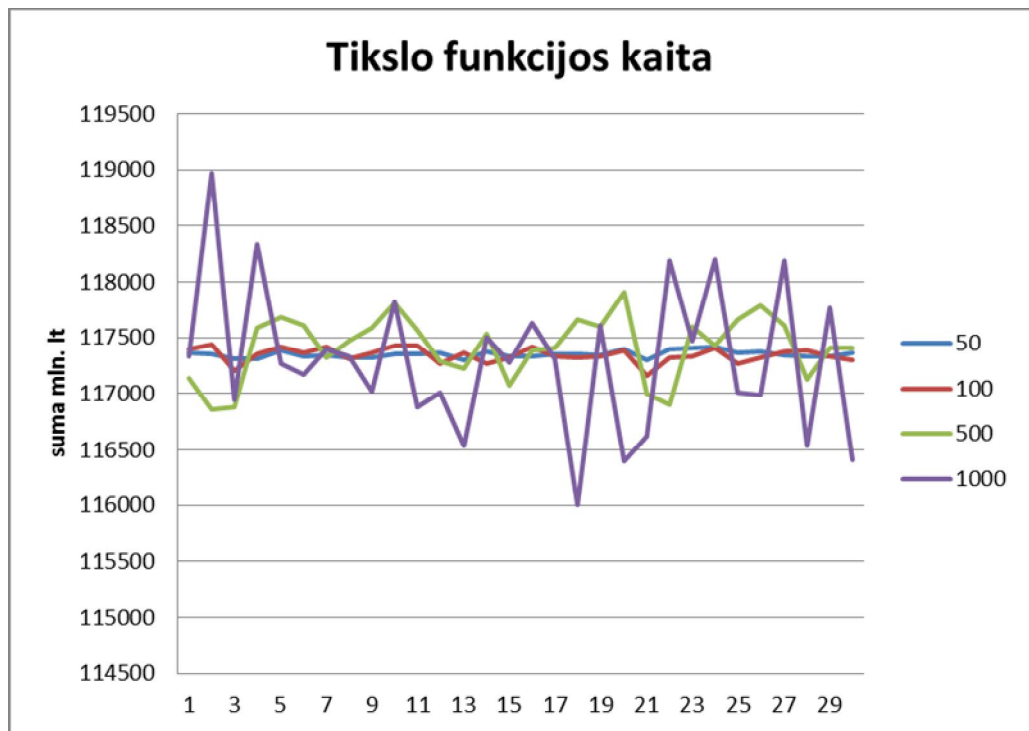
4.4. Tikslumo įtaka

4.4.1. Duomenys

Sprendimo modelis parinktas, kai investuojama 20 metų laikotarpiui 25 mlrd. litų. Dispersija parinkta 0,5.

Tikslumas:	50	100	500	1000
1	117358,2	117398,2	117140,7	117326,2
2	117350,4	117437,4	116860,6	118975,9
3	117308,7	117199,5	116881,6	116949,1
4	117310,4	117353,1	117583,7	118343
5	117379,2	117414,4	117680,2	117271,1
6	117328,7	117359,2	117608,8	117173,3
7	117344,8	117412,2	117324,4	117398,7
8	117323,9	117306,9	117471,1	117332,4
9	117318,1	117363,2	117582,9	117023,5
10	117355,9	117423,9	117810,5	117826,6
11	117353,5	117429,9	117566,9	116876,9
12	117363,5	117269	117290,5	117004,1
13	117302,1	117359	117221,8	116538,9
14	117372,3	117268,4	117532,5	117502,3
15	117325,7	117322,4	117076,4	117282,6
16	117332,2	117418,3	117382,3	117630,5
17	117348,4	117334	117409,6	117305,9
18	117354,7	117324	117662,1	115999,4
19	117344,5	117334,5	117594,8	117609,4
20	117393,7	117387,5	117904,6	116400
21	117303,8	117154,7	116999,8	116621,6
22	117398,5	117325,4	116900,1	118189,1
23	117408,2	117333	117602,7	117474,4
24	117415,6	117408,8	117426,9	118201,4
25	117360,6	117271,6	117667,4	117009,4
26	117378	117323,6	117796,2	116992,6
27	117346,4	117375,8	117614,4	118184,4
28	117326,2	117387	117127,3	116535,8
29	117334,1	117333,1	117410,1	117772,9
30	117360,4	117301,2	117402,1	116409,8

Lentelė 19. Uždavinio paskaičiuotas rezultatas pagal nurodyta tikslumą



37 pav. Energetikos investicija uždavinio sprendimas tikslumo įtaka

Tikslumas:	50	100	500	1000
Mažiausias:	117302,1	117154,7	116860,6	115999,4
Didžiausias:	117415,6	117437,4	117904,6	118975,9
Skirtumas:	113,4818	282,7775	1044,052	2976,523

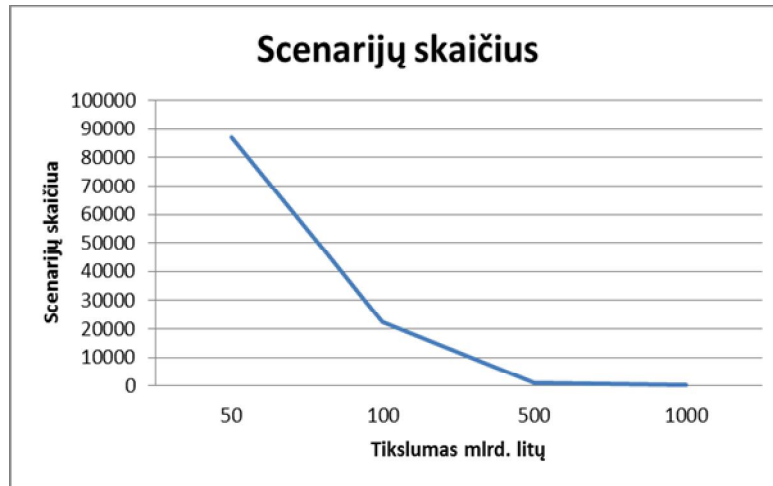
Lentelė 20. Uždavinio rastos rezultatas pagal nurodyta tikslumą



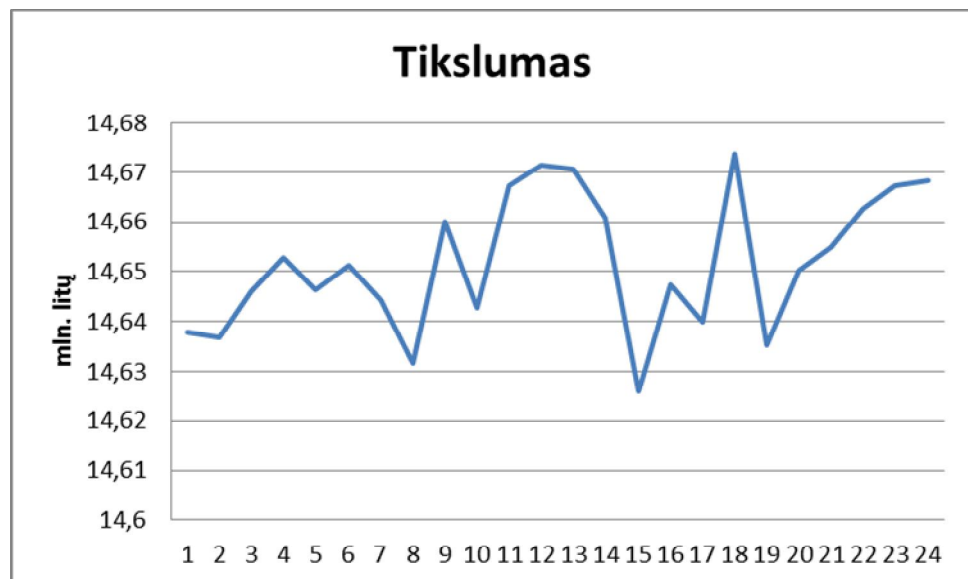
38 pav. Uždavinio rastos rezultatas pagal nurodyta tikslumą

Tikslumas:	50	100	500	1000
Scenarijų skaičius:	87065,83	22406,7	854,0667	217,1667

Lentelė 21. Uždavinio scenarijų skaičius pagal nurodyta tikslumą



39 pav. Uždavinio scenarijų skaičius pagal nurodyta tikslumą



40 pav. Uždavinio tikslo siekiama nurodžius 10 mlrd. litų

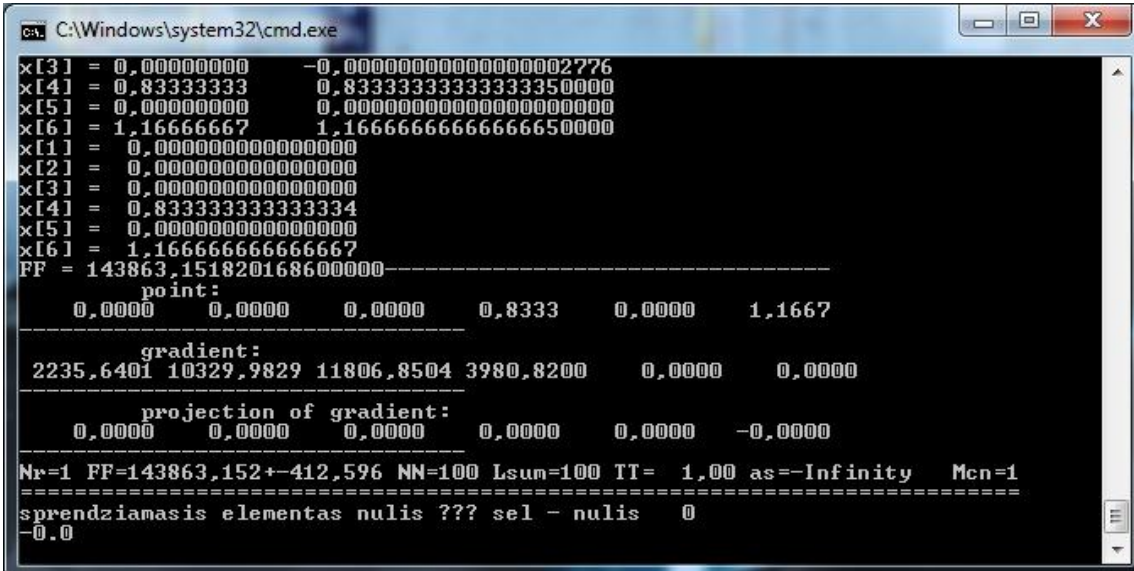
4.4.2. Išvados

Pagal pateiktus duomenis galima teikti, kad programoje turi įtaką nurodomas tikslumas. Kuo didesnis tikslumas tuo mažesnis tikslo funkcijos rezultato išsimėtymas. Kuo didesnis tikslumas, tuo ilgiau programa skaičiuoja (didesnis scenarijų skaičius). Nustačius labai didelį tikslumą programa gali nesugebėti rasti norimą tikslumą (žr. 39 pav).

4.5. Programos klaidos

Sprendžiant stochastinį uždavinį galima susidurti su neišsprendžiamais uždaviniais.

Kaip viena iš šių situacijų:



```
C:\Windows\system32\cmd.exe
x[3] = 0,00000000 -0,000000000000000002776
x[4] = 0,83333333 0,83333333333333350000
x[5] = 0,00000000 0,00000000000000000000
x[6] = 1,16666667 1,16666666666666650000
x[1] = 0,0000000000000000
x[2] = 0,0000000000000000
x[3] = 0,0000000000000000
x[4] = 0,8333333333333334
x[5] = 0,0000000000000000
x[6] = 1,1666666666666667
FF = 143863,151820168600000-----
point:
0,0000 0,0000 0,0000 0,8333 0,0000 1,1667
-----
gradient:
2235,6401 10329,9829 11806,8504 3980,8200 0,0000 0,0000
-----
projection of gradient:
0,0000 0,0000 0,0000 0,0000 0,0000 -0,0000
-----
Nr=1 FF=143863,152+-412,596 NN=100 Lsum=100 IT= 1,00 as=-Infinity Mcn=1
=====
sprendziamasis elementas ??? sel - nulis 0
-0,0
```

41 pav. Neišsprendžiamas uždavinys

Šiuo atveju, kad galėtume jį išspręsti turime keisti statistikos parinktus duomenis, tarkim didinti nn ir $Lmin$.

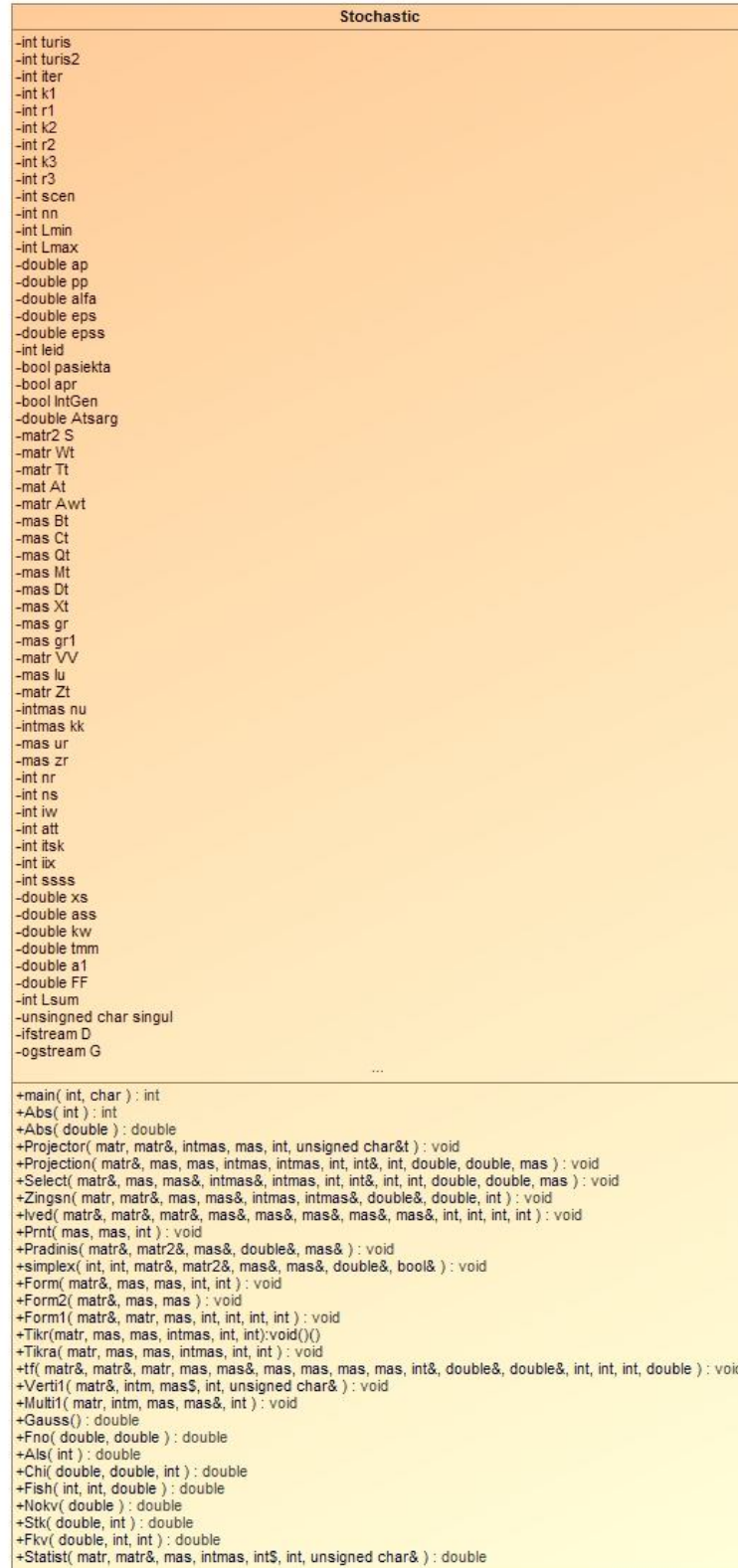
Kartais susiduriama reikšmė *Infinity*. Tai reiškia, kad programa nepajėgia suskaičiuoti, viršija leistina riba. Darant tyrimus šiuo atveju *Infinity* lygus 0.

Atliekant skaičiavimus, galima pasižiūrėti CSV faile vedamus duomenis, bet negalima skaičiavimo metu išsaugoti, nes tada nutrauks programos darbą, dėl pakeisto naudojamo failo.

5. Analizuojama programa

UML schema

Pateikiama struktūrinės programos UML schema:



42 pav. Analizuojamos programos funkcijos ir kintamieji

Tiesinio programavimo uždavinio sprendimo algoritmas

Tikslas: tiesinio programavimo tiesioginio uždavinio bei jam dualaus uždavinio sprendinių ir tikslo funkcijos optimaliosios reikšmės radimas

Pradinės sąlygos (*preconditions*): tiesinio programavimo uždavinio formalizuotas aprašymas.

Galinės sąlygos (*postconditions*): tiesioginio uždavinio sprendinio vektorius, dualaus uždavinio sprendinio vektorius, tikslo funkcijos optimali reikšmė.

1. Inicializuoti M uždavinio matricą, t. y. suformuoti pradinę kintamųjų bazę ir parinkti pradinį leistinąjį sprendinį.

2. **While** leistinasis sprendinys nėra optimalus **do**

2.1. Rasti sprendžiamąjį stulpelį *sst*.

2.2. **If** uždavinys neaprežtas **then** konstatuoti uždavinio neišsprendžiamumą.

2.3. Rasti sprendžiamąją eilutę *seil* ir sprendžiamąjį elementą *sel*.

2.4. Sukeisti bazinį kintamąjį x_{seil} su laisvąjį kintamąjį x_{sst} vietomis. t. y. suformuoti naują kintamųjų bazę.

2.5. Atlikti simplekso metodo Žordano žingsnį sprendžiamajam elementui.

2.6. Apskaičiuoti naująjį leistinąjį sprendinį.

end while.

3. Apskaičiuoti uždavinio tiesioginį bei dualųjį sprendinius ir tikslo funkcijos reikšmę.

Dviejų etapų tiesinio stochastinio programavimo uždavinio sprendimo algoritmas

Tikslas: rasti dviejų etapų tiesinio stochastinio programavimo uždavinio sprendinį ir tikslo funkcijos reikšmę.

Pradinės sąlygos (*preconditions*): dviejų etapų tiesinio stochastinio programavimo uždavinio formalizuotas aprašymas.

Galinės sąlygos (*postconditions*): dviejų etapų tiesinio stochastinio programavimo uždavinio sprendinio vektorius $x^*=(x_1, x_2, \dots, x_n)$, tikslo funkcijos optimali reikšmė $F(x^*)$.

1. Surašyti uždavinį apibūdinančius parametrus: pirmojo ir antrojo etapų kintamųjų ir ribojimų skaičius n_1, m_1, n_2, m_2 , pradinį, minimalų ir maksimalų Monte Karlo imčių ilgį N_0, N_{min}, N_{max} , tikslo funkcijos tikslumą δ , paieškos žingsnio ilgį α , pasikliautinojo intervalo tikimybę β , optimalumo hipotezės tikimybę $1-\gamma$, maksimalų iteracijų skaičių *iter*.

2. Inicializuoti uždavinio koeficientų matricas A, W, T bei vektorius c, b, q, μ, σ .
3. Nustatyti pradinį gradientinės paieškos tašką $x^0 = (x_1^0, x_2^0, \dots, x_{n_l}^0)$ sprendžiant deterministinį uždavinio analogą, kai atsitiktiniai dydžiai pakeičiami jų vidurkiais $h_i = \mu_i, i = 1, 2, \dots, m_l$. (Simplekso programa).

4. Patikrinti pradinį tašką:

4.1. Patikrinti pirmojo etapo uždavinio ribojimus $Ax = b$ pradiniame taške

$$x^0 = (x_1^0, x_2^0, \dots, x_{n_l}^0).$$

4.2. **If** pradinis taškas netenkina ribojimų **then** suprojektuoti tašką $x^0 = (x_1^0, x_2^0, \dots, x_{n_l}^0)$ į ribojimų sritį $D: \pi_D(x^0)$.

5. **While** iteracijų skaičius t neviršija maksimalaus dydžio *iter do*

5.1. Patikrinti ribojimus apskaičiuotame taške. $X^t = (x_1^t, x_2^t, \dots, x_{n_l}^t)$.

5.2. Apskaičiuoti tikslo funkcijos $f(x, y^t), i = 1, 2, \dots, N^t$ ir gradiento $g(x, y^t), i = 1, 2, \dots, N^t$ reikšmes pagal formules pasinaudojant Monte Karlo imties dydžiais.

5.3. Apskaičiuoti tikslo funkcijos ir gradiento įverčius $\tilde{F}(x^t)$ ir $\tilde{g}(x^t)$, gradiento dispersiją $\tilde{D}^2(x^t)$.

5.4. Sudaryti gradiento kovariacijos matricą $Z(x^t)$.

5.5. Suprojektuoti gradientą ϵ -leistinąja kryptimi $\tilde{g}_\epsilon^t = \tilde{g}(x^t)_{V_\epsilon(x^t)}$

5.6. Apskaičiuoti gradiento statistiką T_t^2

5.7. Patikrinti stabdymo sąlygą:

5.7.1. Apskaičiuoti Fišerio skirstinio kvantilį $Fish(\gamma, n_l, N^t - n_l)$.

5.7.2. **If** statistika mažesnė už Fišerio kvantilį $T_t^2 \leq Fish(\gamma, n_l, N^t - n_l)$

and imties dispersijos pasikliautiniosios ribos neviršija tikslumo $\frac{\eta_\beta \cdot \tilde{D}(x^t)}{\sqrt{N^t}} \leq \delta$ **then**

gražinti tikslo funkcijos reikšmę $F(x^*)$ ir uždavinio sprendinį $x^* = (x_1, x_2, \dots, x_n)$.

5.8. Apskaičiuoti gradientinės paieškos žingsnio ilgį $p_{x^i}(g)$

5.9. Rasti kitą tašką $x^{t+1} = (x_1^{t+1}, x_2^{t+1}, \dots, x_{n_l}^{t+1})$.

5.10. Apskaičiuoti Monte Karlo imties ilgį N^{t+1} .

6. **done**.

Dviejų etapų tiesinio stochastinio programavimo uždavinio tikslo funkcijos ir gradiento įverčių bei kovariacijos matricos skaičiavimo algoritmas.

Tikslas: rasti Monte Karlo imties tikslo funkcijos ir gradiento įverčius bei kovariacijos matricą.

Pradinės sąlygos (preconditions): pradinis taškas $x^t = (x_1^t, x_2^t, \dots, x_{n1}^t)$, Monte Karlo imties ilgis N^t .

Galinės sąlygos (postconditions): dviejų etapų tiesinio programavimo uždavinio tikslo funkcijos įvertis $\tilde{F}(x^t)$, gradiento įvertis $\tilde{g}(x^t)$, kovariacijos matrica $Z(x^t)$.

1. **While** i neviršija Monte Karlo imties ilgio N^t **and** imties dispersijos pasikliautiniosios ribos neviršija

Tikslumo $\frac{\eta_\beta \cdot D(x^t)}{\sqrt{N^t}} \leq \delta$ **do**

1.1. Sugeneruoti atsitiktinius dydžius $h_i = \sigma_i \cdot \text{Gauss}() + \mu_i$, $i=1, 2, \dots, m_2$. $\text{Gauss}()$ yra funkcija, generuojanti standartinio normaliojo skirstinio atsitiktinį dydį.

1.2. Rasti antrojo etapo uždavinio dualųjį sprendinį $u^i = (u_1^i, u_2^i, \dots, u_{m2}^i)$ (Simplekso programa).

1.3. Apskaičiuoti tikslo funkcijos gradientą $g(x^t, u^i)$.

1.4. Apskaičiuoti tikslo funkcijos reikšmę $f(x^t, u^i)$.

2. **done**.

3. Apskaičiuoti tikslo funkcijos įvertį $\tilde{F}(x^t)$.

4. Apskaičiuoti gradiento įvertį $\tilde{g}(x^t)$.

5. Apskaičiuoti dispersijos įvertį $\tilde{D}^2(x^t)$.

6. Sudaryti kovariacijos matricą $Z(x^t)$.

Vektoriaus projektavimo į aibę algoritmas

Tikslas: rasti vektoriaus projekciją aibėje $D = \{x : Ax = b\}$.

Pradinės sąlygos (preconditions): pradinis vektorius $z = (z_1, z_2, \dots, z_n)$, formalizuotas aibės D aprašymas.

Galinės sąlygos (postconditions): vektoriaus projekcija aibėje D .

1. Apskaičiuoti matricų sandaugą $AI = A \cdot A^T$.

2. Rasti matricą $A2$, atvirkštinę matricai AI .

3. Rasti projektorių $PI = I_n - AI \cdot z \cdot A2$, I_n – n -tosios eilės vienetinė matrica.

4. Rasti projektorių $P2 = A \cdot b \cdot AI$.

5. Sudaryti projektorių $P=p2+PI$.

Gradiento ε -projektavimo algoritmas.

Tikslas: suprojektuoti gradientą ε -leistinosiomis kryptimis $V_\varepsilon(g)$.

Pradinės sąlygos (preconditions): pradinis gradientas $g=(g_1, g_2, \dots, g_n)$, pradinis taškas $x^t=(x_1^t, x_2^t, \dots, x_{n1}^t)$, ε -projektavimo parametras.

Galinės sąlygos (postconditions): gradiento ε -projekcija.

1. Nustatyti taško $x^t=(x_1^t, x_2^t, \dots, x_{n1}^t)$ nulines komponentes ir sudaryti leistinųjų krypčių aibę (aktyvių ribojimų aibę) $V(x)$
2. Suprojektuoti gradientą aktyvių ribojimų aibėje.
3. Sudaryti ε -leistinųjų krypčių aibę (beveik aktyvių ribojimų aibę) $V_\varepsilon(g)$
4. Suprojektuoti gradientą beveik aktyvių ribojimų aibėje.

Hotelingo T^2 -statistikos skaičiavimo algoritmas.

Tikslas: rasti gradiento Hotelingo T^2 -statistiką.

Pradinės sąlygos (preconditions): gradientas $g=(g_1, g_2, \dots, g_n)$, kovariacijos matrica $Z(x^t)$.

Galinės sąlygos (postconditions): Hotelingo T^2 -statistika.

1. Rasti matricą ZI , atvirkštinę kovariacijų matricai $Z(x^t)$.
2. Apskaičiuoti Hotelingo T^2 -statistiką.

Gradientinės paieškos žingsnio skaičiavimo algoritmas.

Tikslas: rasti kitą gradientinės paieškos tašką.

Pradinės sąlygos (preconditions): pradinis taškas $x^t=(x_1^t, x_2^t, \dots, x_{n1}^t)$, gradientas $g=(g_1, g_2, \dots, g_n)$.

Galinės sąlygos (postconditions): kitas gradientinės paieškos taškas $x^{t+1}=(x_1^{t+1}, x_2^{t+1}, \dots, x_{n1}^{t+1})$.

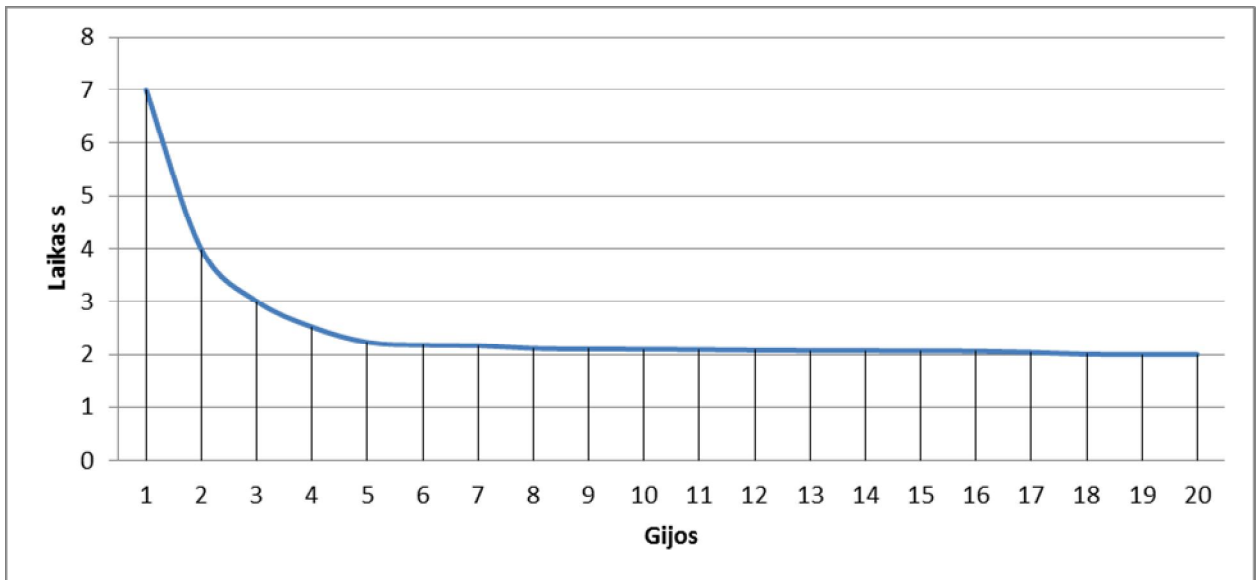
1. Rasti gradientinės paieškos žingsnio ilgį $p_{x^t}(g)$.
2. Apskaičiuoti sekantį galimą tašką $x^t=(x_1^t, x_2^t, \dots, x_{n1}^t)$.
2. **If** taškas nepriklauso ribojimų sričiai D **then** suprojektuoti tašką į sritį D .

6. Programos spartos tikrinimas (Gijos)

Testuojant programą, parinktas uždavinys, kuris skaičiuotų vienodą iteracijų kiekį.

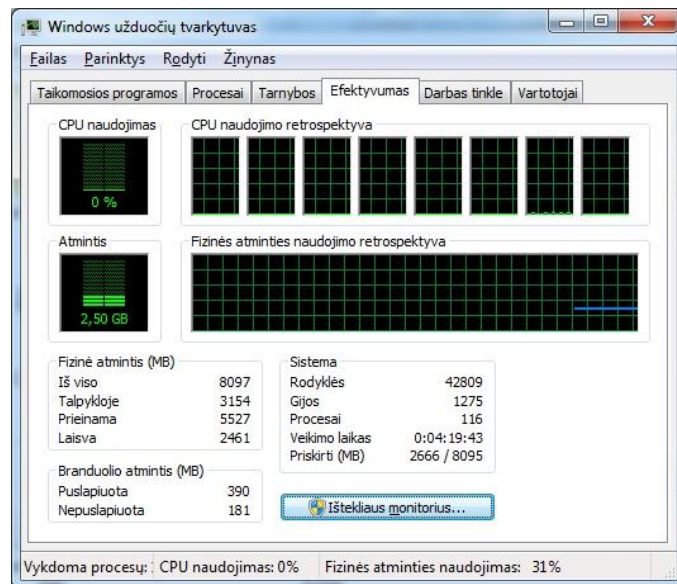
Nr/Gija	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	7	8	9	10	11	13	15	17	19	21	23	25	27	29	31	33	34	36	39	41
2	7	8	9	10	11	13	15	18	19	21	23	25	27	30	31	31	35	36	40	40
3	7	8	9	10	12	13	15	17	19	21	23	25	27	30	31	32	35	38	41	40
4	8	8	9	9	11	14	16	17	19	20	23	26	29	29	32	34	32	38	39	42
5	7	8	9	10	11	12	15	17	18	20	22	26	27	29	32	33	36	37	34	40
6	7	8	9	10	10	12	15	16	18	21	23	25	27	28	32	31	32	37	39	40
7	7	8	9	10	11	13	14	17	19	21	23	25	28	26	30	33	34	34	38	43
8	7	7	9	10	12	13	15	17	19	22	22	28	27	29	29	33	34	34	34	46
9	6	7	9	10	12	13	15	17	19	21	20	25	25	29	29	35	33	35	38	38
10	7	8	10	11	10	15	15	17	18	22	23	25	27	27	31	33	36	35	36	39
11	7	8	9	10	12	16	13	19	19	20	23	20	27	27	32	29	37	35	37	40
12	7	8	9	11	11	12	15	17	20	21	24	25	26	26	31	33	33	34	37	38
13	7	8	9	11	10	13	15	17	19	21	23	25	28	28	31	34	35	36	38	39
14	7	9	9	10	12	13	15	17	19	21	24	25	27	29	30	33	34	36	40	37
15	8	7	9	10	10	12	16	17	20	22	23	24	27	32	30	31	34	35	41	40
16	7	8	9	11	12	13	15	18	19	22	23	24	27	31	32	34	36	32	38	42
17	7	8	9	10	10	13	15	17	18	21	24	25	24	29	31	32	32	37	38	42
18	7	8	8	9	11	16	14	17	19	21	24	25	27	29	31	33	36	37	41	43
19	8	8	9	10	12	11	15	16	19	23	25	26	28	28	30	33	34	39	39	37
20	7	8	9	10	11	13	15	17	19	21	22	25	27	29	29	35	36	38	36	39
21	7	8	9	10	12	13	17	17	19	19	23	25	28	30	28	33	34	38	36	41
22	6	8	9	9	11	13	15	17	19	21	23	24	27	32	30	33	32	37	37	40
23	7	9	10	10	12	13	15	15	20	21	23	25	27	31	30	34	35	37	38	40
24	7	8	9	12	12	14	15	16	18	21	23	25	27	30	32	32	34	36	40	41
25	7	8	9	10	12	14	16	15	19	21	24	25	27	30	31	33	36	34	41	39
26	7	8	9	9	10	13	16	17	19	21	22	25	28	31	32	32	35	35	37	38
27	6	8	9	10	12	12	15	17	19	22	23	26	27	30	32	33	36	36	38	39
28	7	8	9	11	10	13	16	17	18	21	23	24	26	28	34	36	35	36	38	38
29	7	8	9	11	11	13	16	17	19	20	23	25	25	26	31	34	38	37	37	37
30	7	8	9	9	11	11	16	17	19	19	22	25	27	27	33	34	36	36	35	40
Vid:	7	7,96666667	9,03333333	10,1	11,1666667	13,0666667	15,1666667	16,9	18,9	20,9666667	22,96667	24,93333	26,93333	28,96667	30,93333	32,96667	34,63333	36,03333	38	39,96667
Gijos laikas:	7	3,98333333	3,01111111	2,525	2,23333333	2,17777778	2,16666667	2,1125	2,1	2,09666667	2,087879	2,077778	2,071795	2,069048	2,062222	2,060417	2,037255	2,001852	2	1,998333

43 pav. Programos spartos skaičiavimas

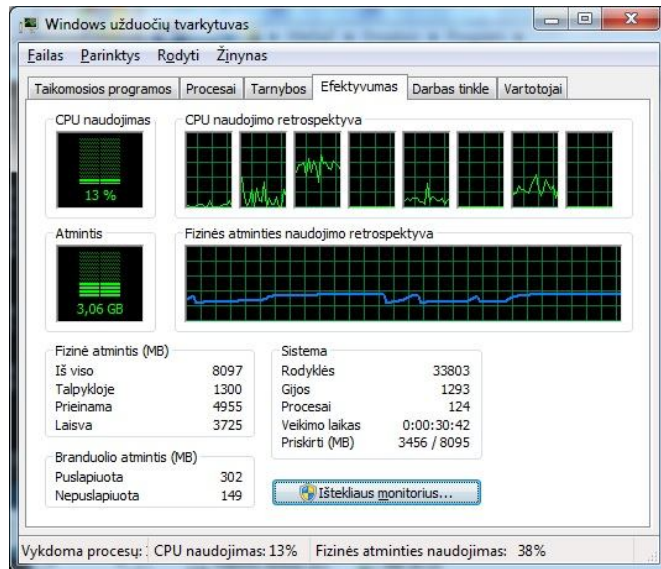


44 pav. Vienos gijos laiko vidurkis gijoms

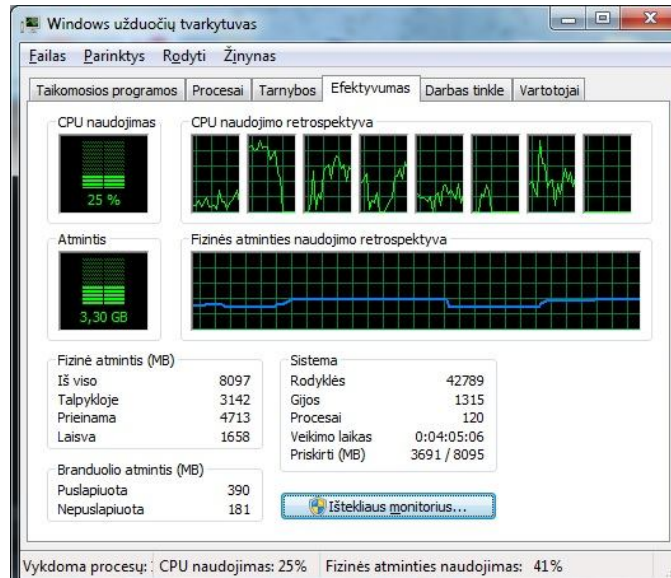
Testuojant programos spartą, pavyko išgauti iki 3,5 karto spartesnę skaičiavimą laiko atžvilgiu skirta vienai gijai. Pastebėta, kad skaičiuojant didesnę gijų skaičių susidurta su situacijomis, kai daugiau parinktų gijų suskaičiuoja greičiau negu mažesnę gijų skaičių.



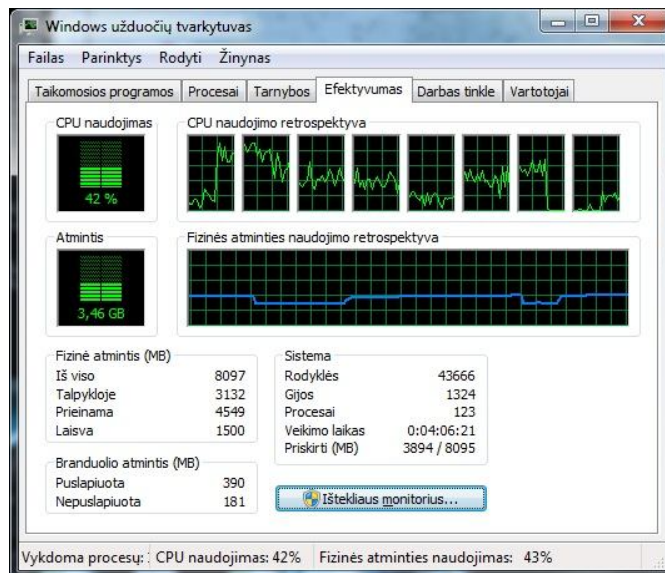
45 pav. Programos spartos skaičiavimas 0 gijų



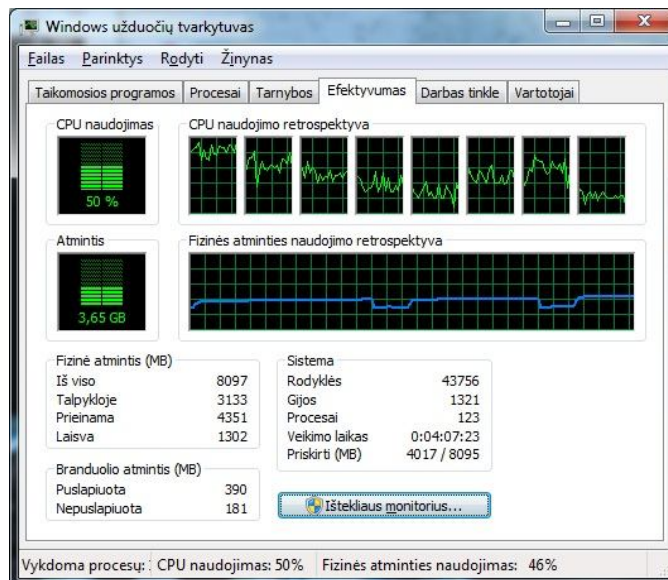
46 pav. Programos spartos skaičiavimas 1 gija



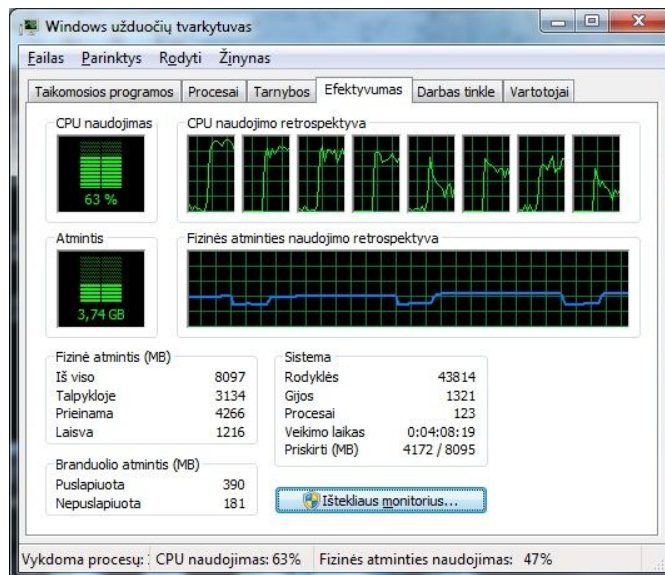
47 pav. Programos spartos skaičiavimas 2 gijos



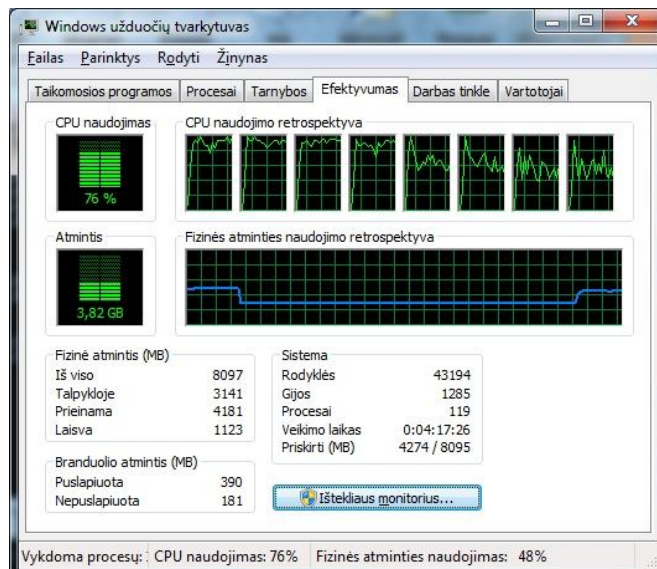
48 pav. Programos spartos skaičiavimas 3 gijos



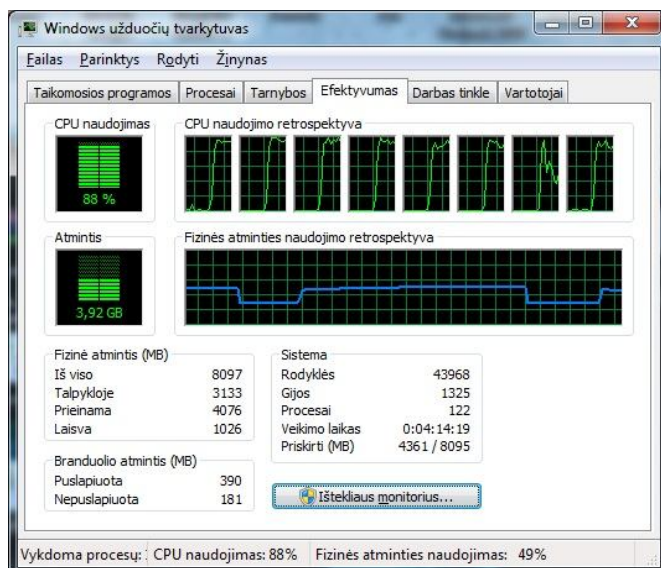
49 pav. Programos spartos skaičiavimas 4 gijos



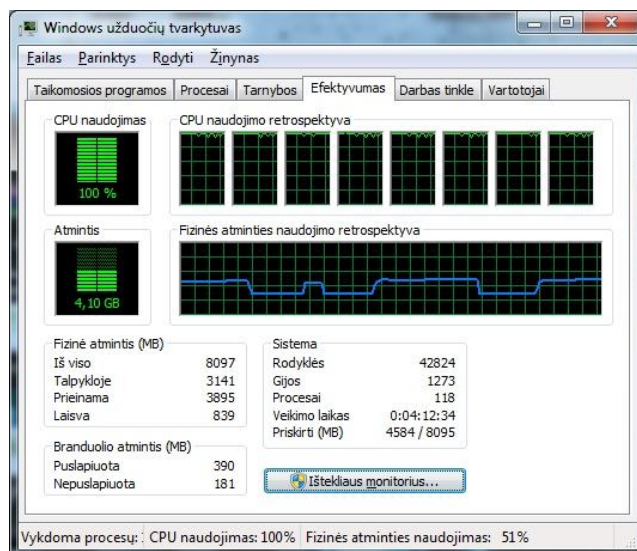
50 pav. Programos spartos skaičiavimas 5 gijos



51 pav. Programos spartos skaičiavimas 6 gijos

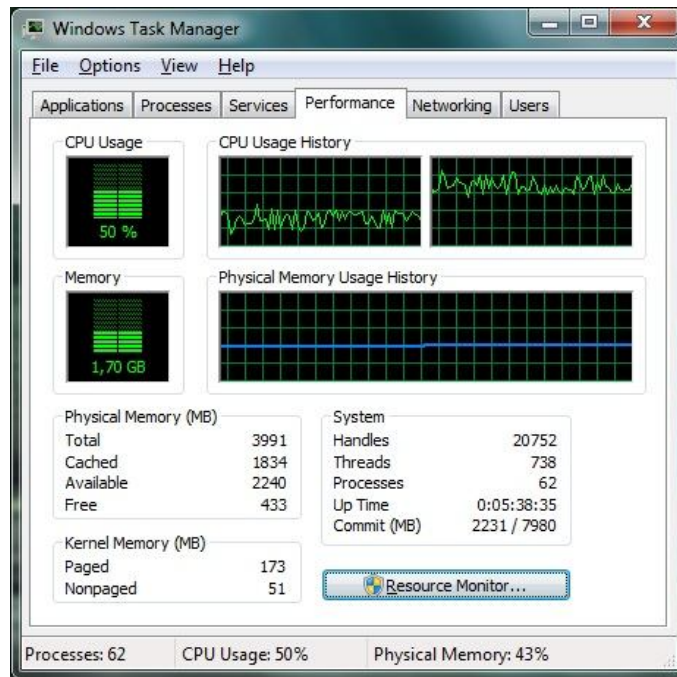


52 pav. Programos spartos skaičiavimas 7 gijos

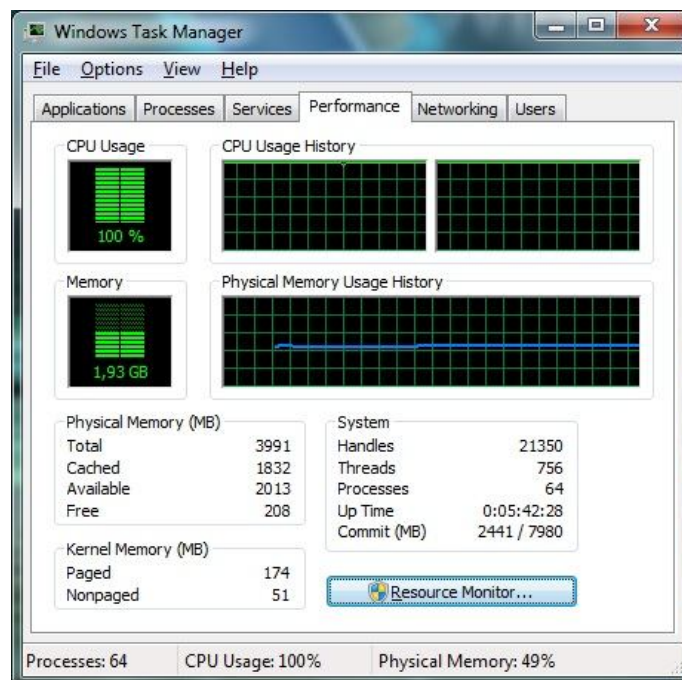


53 pav. Programos spartos skaičiavimas 8 gijos

Stebint kompiuterio apkrovą, pastebėta, kad 8 branduolių (i7-2670QM) kompiuteris, vienai gijai skiria 1 branduolio apkrovą. Patikslinti duomenims pasirinktas antras kompiuteris 2 branduolių E8400 3GHZ procesorius.



54 pav. Programos spartos skaičiavimas 2 branduoliai (E8400) 1 gija



55 pav. Programos spartos skaičiavimas 2 branduoliai (E8400) 2 gija

Pateiktuose paveikslėliuose matyti, kad 2 branduolių (E8400) procesorius 1 gijai skiria 50 procentų apkrovos, tai yra vienai gijai skiriamas 1 branduolys.