ŠIAULIŲ UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

Marius Neimantas

Informatikos specialybės magistrantūros II kurso dieninio skyriaus studentas

INTERNETINĖ MATEMATINIO PROGRAMAVIMO IR MODELIAVIMO SISTEMA. STOCHASTINIO PROGRAMAVIMO PROGRAMŲ PAKETO KŪRIMAS IR ANALIZĖ
ONLINE MATHEMATICAL PROGRAMMING AND SIMULATION SYSTEM. THE DEVELOPMENT AND ANALYSIS OF STOCHASTIC PROGRAMMING PROGRAM PACKAGE

MAGISTRO DARBAS

Darbo vadovas:
prof. habil. dr. L. SAKALAUSKAS

Recenzentas:
doc. dr. K. Žilinskas

Šiauliai, 2012

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

Darbo autoriaus _____

(vardas, pavardė, parašas)

Darbo tikslai ir uždaviniai

Tikslas

Pasinaudojus spietinėmis technologijomis sukurti stochastinio programavimo lygiagrečių programų paketą ir ištirti jo lygiagretinimo efektyvumą.

Uždaviniai

- Išanalizuoti vikinomikos sampratą spietinių programinių sistemų kūrimui;
- Išanalizuoti stochastinio programavimo sampratą;
- Išanalizuoti atviro kodo sampratą ir jo aktualias rūšis, kuriant spietines matematinio modeliavimo sistemas;
- Išanalizuoti lygiagrečių skaičiavimų principus, daugiaprocesorinę techninę įrangą ir paskirstytųjų bei lygiagrečių skaičiavimų taikymo ypatumus;
- Išanalizuoti stochastinio programavimo Monte Karlo metodu nuosekliąją programą ir jos pagrindu sukurti lygiagrečių skaičiavimų programinį paketą;
- Įvertinti programų paketo lygiagretinimo efektyvumą.

Darbo vadovo _____
(vardas, pavardė, parašas)

Turinys

ĮVADAS

Atviro kodo programa ir standartai – tai tema, kuri šiuo metu yra aktuali ir kelianti daug diskusijų tiek Lietuvoje, tiek pasaulio informacinėje visuomenėje[1]. Atviro kodo programos pirmiausia svarbios dėl to, kad jos palengvina prieigą prie programų išeities kodo ir tai gali padaryti kiekvienas norintis asmuo. Atviro kodo principas suteikia galimybę pakartotinai panaudoti, tobulinti autoriaus kūrinį (programą), taip pat minėtas principas gali suteikti kitas laisves, kurios būna nustatytos atviro kodo licencijoje. Šios programos skatina sparčiau plėtoti informacines technologijas, formuoja naujas idėjas, didina konkurencingumą su komercinėmis uždaro kodo programomis. Pažymėtina, kad dauguma šių programų yra nemokamos [1].

Atviro kodo programos suteikia galimybę vartotojams / kūrėjams išreikšti savo nuomonę bei pasidalinti informacinių technologijų srityje turima patirtimi su kitais kūrėjais. Taip susiformuoja glaudus bendradarbiavimas, gali vykti išteklių mainai ir t. t. Visa tai galima apibrėžti kaip siekius, kurių siekiama vikinomikos modelyje.

Šių dienų pasiekimai informacinių technologijų srityje leidžia spręsti vis sudėtingesnius ir didesnės apimties uždavinius. Šių uždavinių sprendimai gali trukti keletą ar net keliolika dienų, savaičių ar net kelerių metų. Tam įtakos gali turėti optimizavimo uždaviniuose esantys ribojimai, jų kiekis. A. Žilinskas (2005) tokių uždavinių sprendimo teoriją ir metodus esant ribojimams įvardina „matematiniu programavimu“.

Pasak minėto autoriaus, „optimizavimo teorija ir optimizavimo metodų bei juos realizuojančios programinės įrangos kūrimas yra aktyviai vystoma mokslo ir technikos šaka“ (p. 8) [24].

Siekiant išnaudoti visas šių dienų technologines galimybes ir turint tikslą padidinti skaičiavimo efektyvumą, buvo pradėti naudoti lygiagretūs bei kitokie paskirstytieji skaičiavimai. Paskirstytos ir bendros atminties sistemos padeda realizuoti minėtus skaičiavimus, jų ypatumai aptarti šiame darbe [9].

Darbo tikslas – pasinaudojus spietinėmis technologijomis sukurti stochastinio programavimo lygiagrečių programų paketą ir ištirti jo lygiagretinimo efektyvumą.

[1] Informacinė visuomenė – visuomenės dalis, kuri savo kasdienėje veikloje (darbe, namuose ir t. t.) aktyviai naudojasi įvairiomis informacinės technologijomis. Tai atvira, išsilavinusi, nuolat besimokanti ir žiniomis savo veiklą grindžianti visuomenė, kurios nariai turi galimybę ir geba visose savo veiklos srityse efektyviai naudotis šiuolaikinėmis IRT priemonėmis. Prieiga per internetą: <http://www.ivpk.lt/main.php?cat=50>

Darbo uždaviniai:

- Išanalizuoti vikinomikos sampratą spietinių programinių sistemų kūrimui;
- Išanalizuoti stochastinio programavimo sampratą;
- Išanalizuoti atviro kodo sampratą ir jo aktualias rūšis, kuriant spietines matematinio modeliavimo sistemas;
- Išanalizuoti lygiagrečių skaičiavimų principus, daugiaprocesorinę techninę įrangą ir paskirstytųjų bei lygiagrečių skaičiavimų taikymo ypatumus;
- Išanalizuoti stochastinio programavimo Monte Karlo metodu nuosekliąją programą ir jos pagrindu sukurti lygiagrečių skaičiavimų programinį paketą;
- Įvertinti programų paketo lygiagretinimo efektyvumą.

Tyrimo metodai: lygiagrečių skaičiavimų realizavimo bibliotekų mokslinė analizė ir sukurtų programų eksperimentinis tyrimas Monte Karlo metodu.

Praktinė darbo reikšmė. Internetinė matematinio programavimo ir modeliavimo sistema leidžia tobulinti programų paketo kodus, leidžia gauti tikslesnius ir greičiau realizuojamus rezultatus. Minėta sistema leidžia kurti naujas programas, skirtas modeliavimo ir matematinio programavimo temoms plėtoti. Visi vartotojai gali pateikti savo realizacijos variantus programų eksploatavimo klausimais.

Mokslinis naujumas. Internetinė svetainė sukurta vikinomikos idėjų pagrindais. Šioje svetainėje visi akademinės visuomenės atstovai gali patalpinti savo programinį kodą, kuris realizuoja matematinio programavimo ar modeliavimo uždavinio sprendimą. Šio kodo tobulinimo, kūrimo procese, modifikavimo procese gali dalyvauti bet kuris akademinės visuomenės atstovas. Be to, minėtoje svetainėje galima koreguoti programinį kodą, jį kompiliuoti ir vykdyti (skirtingai nuo dabar platinamų programavimo sistemų, kuriose šitokie veiksmai yra nenumatomi).

I. Teorinė dalis

1. Temos analizė

1.1. Vikinomikos samprata

Laisvojoje internetinėje enciklopedijoje, vikinomika, vikiekonomika (*wiki* + *ekonomika*, angl. *Wikinomics*) apibrėžiama, kaip „spietinė veiklos forma, kur greitai arba momentiškai apjungus žmones ir skubios saviorganizacijos būdu panaudojus turimus išteklius, sprendžiamos problemos“[2]. Taip pat vikinomika apibūdinama, kaip veiklos, sistemos (organizavimo, veikimo) principas, būdas.

Havajiečių kalbos žodyne, esti žodis *wiki*, kuris išvertus į lietuvių kalbą reiškia *greitas*. Internetinėje erdvėje šis žodis buvo sėkmingai adaptuotas ir panaudotas kaip greito bendradarbiavimo sinonimas. Lietuvių kalboje buvo rastas havajietiško žodžio reikšmę atitinkantis analogas – spiečius.

Vikinomikos sąvokos kūrėjais bei koncepcijos pagrindėjais laikomi Don Tapscott ir Anthony D. Williams, 2006 m. išleidę knygą „Vikinomika – kaip masinis bendradarbiavimas viską keičia“ (angl. *Wikinomics – How Mass Collaboration Changes Everything*). Tais pačiais metais šią publikaciją savaitraštis „The Economist“ ir dienraštis „The Financial Times“ paskelbė metų knyga. Pagrindinis veiksnys, kuris leido formuotis vikinomikai, tai sėkmingas tarptautinis laisvosios interneto enciklopedijos projektas vikipedija, prie kurio kūrimo gali prisidėti kiekvienas norintis.

D. Tapscott suformavo 4 pagrindines idėjas, kuriomis naudojasi vikinomika, tai:

- Atvirumas – paprastai nėra jokių mokesčių, leidimų, reglamentuojančios tvarkos;
- Susivienijimas bendradarbiavimui, išteklių mainai (angl. *peering*);
- Dalinimasis (angl. *sharing*);
- Globalumas.

Vikinomikos modelis apjungia įvairių sričių specialistus, kurie gali įnešti savo idėjų indėlį į tam tikrų klausimų ar problemų sprendimą. Žinoma, šis modelis apima ir asmenų poreikius bei perteklinius išteklius [10].

Vikinomikos modelis yra labai universalus, dėl šios priežasties jis gali būti taikomas daugelyje sričių. Vikinomika veikia atvirumo principu, kuris leidžia kurti, išplėsti, apjungti į vieningą sistemą įvairias sritis, tokias kaip:

- Įmonių, įstaigų darbo organizavimas ir vadyba;

[2]Vikinomika. Prieiga per internetą: <http://lt.wikipedia.org/wiki/Vikinomika>

- Balsavimas, vertinimas, reitingavimas, analizavimas;
- Teisinės sistemos kūrimas, įstatymų leidyba, teisės aktų projektų rengimas;
- Komercija, prekyba;
- Barteriniai mainai;
- Pažintys, žmonių (pvz.: darbuotojų), darbo, išteklių paieška;
- Edukacija [10].

Pirmasis vikinomikos veiklos formos pasirodymas įvyko 2003 m. vasarį, kai buvo pradėtas masinio bendradarbiavimo ir saviorganizacijos projektas žiniatinklyje, pavadintas „Vikipedija. Laisvoji enciklopedija“, kuris taip pat kuriamas ir lietuvių kalba.

Nemokamas vikinomikos naudojimas gali sudaryti klaidingą nuomonę, jog produktų kūrėjai lieka visiškai be atlygio. Tūkstančiai kūrėjų geranoriškai dalijasi savo patirtimi, idėjomis, kuriant ar tobulinant tokius projektus, kaip „Linux“ ar „Apache“, tačiau jie dažniausiai patys turi savo nuosavą sėkmingą verslą.

Darbe siekiama sukurti matematinio programavimo ir modeliavimo sistemą internetinės svetainės pavidalu, pasinaudojant vikinomikos funkcijomis ir modeliu. Svetainės pagrindinė idėja – programinės įrangos kūrėjas perduoda savo sukurtus produktus informacinei visuomenei ir į tolimesnius produktų kūrimus įtraukia informacinės visuomenės atstovus. Minėti asmenys gali pateikti savo realizacijas kiekvieno produkto atžvilgiu. Spietinė metodologija taip pat suteikia galimybę sudėtingus stochastinio programavimo ir statistinio modeliavimo uždavinius spręsti efektyviau bei kokybiškiau.

1.2. Stochastinio programavimo samprata

Stochastinio programavimo uždaviniai, kuriuose dydžiai gali būti nedeterminuoti ir neapibrėžtumai gali būti įvairios prigimties. Šie parametrai dažniausiai sutinkami sprendžiant kaštų ir įplaukų planavimo, darbų padalijimo uždavinius, todėl dažniausiai jie aprašomi atsitiktiniais kintamaisiais. Šios rūšies uždaviniai sprendžiami stochastinio tiesinio arba netiesinio programavimo metodais. Dviejų arba kelių etapų stochastinio tiesinio programavimo uždaviniai yra klasikinio tiesinio programavimo apibendrinimas. Klasikinio tiesinio programavimo uždavinių parametrai yra atsitiktiniai dydžiai [19,23,25,26].

Pagrindinės stochastinio programavimo taikymo sritis:

- Elektros energijos gamyba;
- Produkcijos gamyba ir transportavimas;
- Personalo valdymas;
- Logistika;

- Investicijų valdymas;
- Objektų išdėstymas;
- Mechanizmų stabilumas;
- Biologinių sistemų analizė ir kt. [19,23,25,26]

J. R. Birge ir F. Louveaux (1997) teigia, kad „stochastinio programavimo tikslas yra tiksliai rasti optimalų problemų sprendimą, įtraukiant nepastovius duomenis“ (p. 9). Stochastiniame programavime galimos problemos, kai reikia apskaičiuoti tikslias tikslo funkcijos reikšmes bei gautojo taško priklausomumą leistinajai sričiai tikrinimu. Apskaičiuoti tikslo funkcijos tikslią reikšmę optimaliame taške – beveik neįmanoma, taip pat beveik neįmanomas yra ir ribojimų tikrinimas. Sprendžiant šio tipo uždavinius sprendimo eiga turi būti pasirinkta dar nežinant kai kurių parametrų reikšmių, kurios tikslinamos vėlesniuose etapuose. Šių parametrų reikšmės gali turėti įtakos sprendimo eigos pasirinkimui arba sukurti skirtingą sprendimo eigą [6,19,23,25,26].

Apibendrinat tai, kas išdėstyta, pritartumėme H. Vladimirou ir S. A. Zenios (1999) nuomonei, kad „stochastinis programavimas ir toliau išlieka vienas iš sudėtingiausių skaičiavimų skaitinio optimizavimo srityse“ (p. 88-89) [23]. Papildant minėtų mokslininkų teiginį, būtų galima pridurti, kad stochastinio programavimo uždaviniai gali būti sprendžiami pasitelkiant įvairius sprendimo metodus. Vienas iš tokių metodų – statistinis modeliavimas.

1.3. Statistinis modeliavimas

Statistinis modeliavimas, ar kitaip vadinamas Monte Karlo metodu, yra plačiai taikomas spręsti įvairiems uždaviniams verslo, ekonomikos, telekomunikacijų, fizikos ir kitose srityse. Būtina pabrėžti, kad šių sričių uždaviniai dažniausiai tiriami modeliavimo būdu, kad būtų įmanoma nustatyti sistemų veikimo dėsningumus su tikslu, nustatytu dėsningumo pritaikymu praktikoje. Modeliavimas pasinaudoja matematiniais modeliais, panaudojant šiuolaikines informacines technologijas [19,25,26].

Būtų galima apibrėžti situacijas, kuriose kompiuterinis modeliavimas sėkmingai naudojamas:

- 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 labai sudėtingas arba aprašyti matematiniais modeliais, kurie turėtų analizinius sprendinius, neįmanoma;
- kai sistema aprašyta matematiniu modeliu, bet sprendinių, susijusių su modeliu, gavimas yra neįmanomas, pasinaudojus tik analitiniais metodais;

- kai neįmanoma arba labai brangu atlikti eksperimentus su objektu, tuomet modeliavimo rezultatai gali būti panaudoti testuojant alternatyvias hipotezes [19,25,26].

Statistiniu modeliavimu nagrinėjamos sistemos dažnai pasižymi neapibrėžtumu, kuris teoriškai gali būti aprašytas statistiniais metodais. Šis modeliavimas suprantamas, kaip tiriamo objekto tikimybinis modelio atkūrimas ir tyrimas kompiuteriu [19,25,26].

Statistikos uždaviniai buvo sprendžiami jau seniai su įvestais atsitiktiniais dydžiais, tačiau Monte Karlo metodas nebuvo plačiai paplitęs, nes modeliuoti atsitiktinius dydžius rankiniu būdu be kompiuterių pagalbos yra labai sudėtinga. Tačiau prasidėjus informacinių technologijų erai praeito šimtmečio viduryje Monte Karlo metodas, kaip universalus skaičiavimo metodas ir statistinio modeliavimo teorija buvo pradėti sparčiai plėtoti ir dabar taikomi įvairiuose mokslo ir verslo srityse [19,25,26].

Pasitelkus Monte Karlo metodą galima spręsti tokius uždavinius:

- 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);
- Kiti [19,23,25,26].

Apibendrinant tai, kas išdėstyta, galime numanyti, kad sprendžiant statistinio modeliavimo uždavinius Monte Karlo metodu, būtų tikslinga kartu skaičiuoti ir pasikliautinuosius intervalus. Intervalas, kuriame su tam tikru patikimumo (pasikliovimo) laipsniu tikėtina matuojamo dydžio reikšmė, tai pasikliautinasis intervalas. Minėti intervalai gali būti apskaičiuojami pagal formulę:

$$\left[\bar{x}_N - \eta_\beta \cdot \frac{s_N}{\sqrt{N}}, \bar{x}_N + \eta_\beta \cdot \frac{s_N}{\sqrt{N}}\right] \quad (1.1)$$

čia $\bar{x}_N$ – vidutinė funkcijos reikšmė, s_N – vidutinės funkcijos reikšmės standartinis nuokrypis, β – pasikliovimo tikimybė, N – eksperimentų skaičius. Dažniausiai pasirenkama β =0,95 (η_β =1,96) arba β =0,99 (η_β = 2,58) [19, 16].

2. Darbo srities analizė

Internete ar kitose prekybos vietose galima sutikti gausybę įvairiausių programinių įrangų kategorijų (pvz., Open Source (Atviras kodas), Free Software (Laisvos programos), public domain, Freeware, Shareware ir kt.). Dalis šių kategorijų papuola į laisvų programų apibrėžimą, pvz., Open Source, Free Software. Tačiau ne visi šie terminai reiškia tą patį [1].

Siekiant išlaikyti spietinės metodologijos idėjas (laisvumas, atvirumas ir kt.) reiktų propaguoti programas, kuriose suteikiamos tobulinimo ir modifikavimo galimybės.

2.1. Laisvos programos (Free software)

Laisvoji programinė įranga yra tokia programinė įranga, kurią leidžiama kiekvienam naudotojui, nepriklausomai nuo panaudos srities, keisti, kopijuoti, platinti su atitinkamais pakeitimais ar be jų, už tai gaunant pajamas arba ne, bei tuo pačiu ši programinė įranga platinama su licencija. Taip pat kartu su programa yra platinamas išeities tekstas[3] [1].

Pagrindinė laisvos programinės įrangos idėja yra laisvė, o ne kaina. Šią idėją reikėtų suprasti kaip „laisvę tobulėjimui“, o ne kaip nemokamą produktą [1].

Laisvosios programinės įrangos idėja apibrėžia keturių rūšių laisves, kurios skirtos programinės įrangos naudotojams:

- Laisvė naudoti programą bet kokiems tikslams (laisvė 0).
- Laisvė išstudijuoti programos veikimą ir pritaikyti ją savo reikmėms (laisvė 1). Tam būtinas priėjimas prie pradinių (išeities) programos tekstų.
- Laisvė platinti kopijas tam, kad galėtumėte pagelbėti savo kaimynui (laisvė 2).
- Laisvė tobulinti programą ir išleisti šiuos patobulinimus į viešumą, kad iš to turėtų naudos visa bendruomenė (laisvė 3). Tam būtinas priėjimas prie pradinių programos tekstų [1].

Programa, kuri gali būti laikoma laisva programinė įranga, turi naudotojams suteikti visas keturias laisvės rūšis. Naudotojams suteikiama galimybė laisvai platinti kopijas bet kam ir bet kur, tiek atlikus pakeitimus, tiek be jų, tiek negaunant pajamų, tiek gaunant pajamų už platinimą. Šią programinę įrangą galima modifikuoti ir naudoti asmeniniams tikslams. Apie atitinkamas modifikacijas nereikia niekam pranešti [1].

Vienas iš pagrindinių programų modifikavimo būdų yra prie programos prijungti laisvas paprogrames ar modulius, tačiau jei modulio (paprogramės) licencijoje nurodoma, kad

[3] Išeities tekstas – tai pageidaujama darbo forma modifikacijoms atlikti.

modulio (paprogramės) jungėjas turi turėti autorines teises, tokia licencijavimo tvarka prieštarauja laisvos programinės įrangos idėjoms [1].

Pradinis laisvos programinės įrangos kūrėjas negali panaikinti licencijos be modifikuotojų leidimo, nes priešingu atveju pradinis kūrėjas negali reglamentuoti programinės įrangos, kaip laisvos [1].

Vis tik yra taisyklių, kurios yra priimtinos, nekonfliktuojančios su pagrindinėmis laisvės idėjomis ir susijusios su laisvos programinės įrangos platinimu. Viena tokių taisyklių, kuri saugo laisvės idėjas, yra „copyleft“. Ši taisyklė apibrėžia tai, jog programos platintojas negali pridėti jokių apribojimų, kurie užkirstų kelią kitiems asmenims į programos laisves [1].

Laisva programinė įranga neatitinka nekomercinės[4] programinės įrangos statuso. Laisva programa turi būti prieinama komerciniam naudojimui, vystymui ir platinimui [1].

Laisvos programinės įrangos kūrime naudojama „copyleft“, todėl teisiškai saugoma laisvės teisė visiems [1].

Naujų licencijų kūrimą prižiūri 1985 m. įkurta nepelno siekianti pasaulinė organizacija Free Software Foundation (FSF; laisvosios programinės įrangos fondas). FSF darbo sritys apima kompiuterių naudotojų laisvių skatinimą ir laisvos programinės įrangos vartotojų teisių gynimą [1].

2.2. Atviras kodas

„Atvirasis kodas“ (angl. Open Source) suteikia galimybę gauti išeities kodą. „Atvirojo kodo“ programinės įrangos platinimas turi atitikti šiuos jam keliamus kriterijus:

1. Laisvas platinimas. Licencija neturi drausti kitiems parduoti ar perduoti programinės įrangos kaip programinės įrangos distribucijos dalies, sudarytos paėmus tas dalis iš keleto skirtingų šaltinių. Licencija neturi reikalauti atlygio ar kitokio mokesčio už tokį pardavimą.

2. Išeities kodas. Programa privalo turėti išeities kodą ir turi leisti platinimą išeities kodu ar sukompiliuota[5] forma. Tokiu atveju, jei nors viena laisvos programinės įrangos dalis neplatinama su išeities kodu, turi būti aiškiai numatyta galimybė įsigyti atvirąjį kodą, apmokant motyvuotas kopijavimo išlaidas arba atsisiųsti nemokamai programą iš interneto. Pagal galiojančias licencijas, išeities kodą būtina pateikti tokia forma, kad ją modifikuoti galėtų bet kuris vartotojas, bet draudžiama tyčia modifikuoti kodą taip, kad jis suklaidintų kitus programuotojus.

[4] Komercinis => komercija – prekybinės operacijos, prekyba.
[5] Sukompiliuoti => kompiliacija – nesavarankiškas, iš kitų raštų ištraukų, dažnai tik mechaniškai sujungtų, sudarytas veikalas.

3. Išvestiniai darbai. Atlikti modifikacijas ir kurti išvestinius tekstus leidžia licencija, kurią leidžiama platinti su ta pačia licencija, kaip ir išeities kodas.

4. Autoriaus išeinamojo kodo neliečiamumas. Licencijos pagalba uždrausti platinti modifikuotą išeities kodą galima tik tada, kai ji leidžia platinti originalų kodą su pataisymų failais, leidžiančiais modifikuoti galutinį produktą kompiliacijos metu. Be to ji gali reikalauti platinti programą su pakeistu pavadinimu ar versijos numeriu.

5. Jokios diskriminacijos prieš asmenis ar grupes. Licencija neturi diskriminuoti jokio asmens ar asmenų grupės.

6. Jokių apribojimų panaudojimo sritims. Programos panaudojimas tam tikroje specifinėje aplinkoje (pvz., komercinėje aplinkoje, genetiniams tyrimams) neturi būti draudžiamas licencijos.

7. Licencijos platinimas. Kiekvienai programinei įrangai ir jos išeities kodui skirta viena konkreti licencija, kuri apibrėžia visas teises ir pareigas, susijusias su šia programine įranga ir jos išeities kodu, todėl programa neturi būti platinama su jokia kita papildoma licencija.

8. Licencija neturi būti specifinė produktui. Nepriklausomai nuo to, ar programa platinama kartu su visu programiniu paketu ar kaip atskira dalis, atitinkanti licencijos sąlygas, kiekvienas programą gavęs vartotojas kartu gauna ir visas teises, kaip ir programą su pilnu programiniu paketu gavęs vartotojas.

9. Licencija neturi riboti kitos programinės įrangos. Laisvai platinama programinė įranga neturi riboti kitų programų, platinamų su atitinkamomis licencijomis. Tai jokiu būdu nereiškia, kad visa toje pačioje laikmenoje platinama programinė įranga turi būti atvirojo kodo.

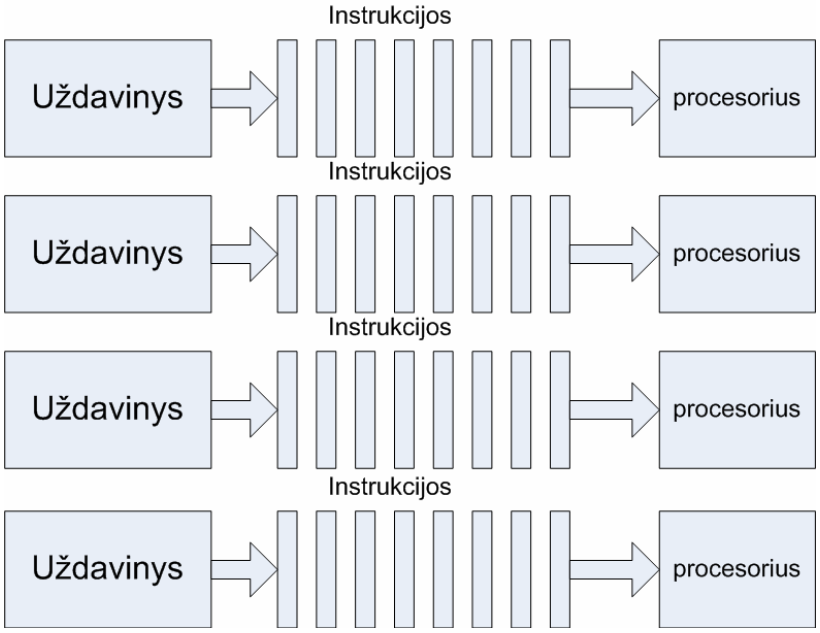
10. Licencija turi būti neutrali technologijų atžvilgiu. Licencijos sąlygos neturi priklausyti nuo individualios technologijos ar naudojamos sąsajos tipo [1].

Aptarus programų kategorijas, manytina, kad toliau darbe pravartu išanalizuoti galimus skaičiavimų būdus ir architektūrinius kompiuterių tipus.

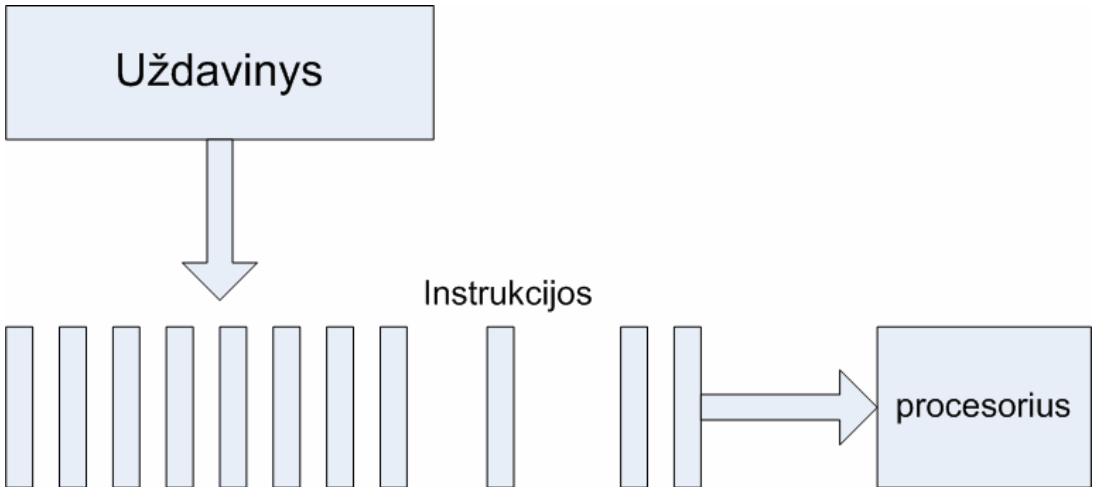
2.3. Lygiagretieji ir paskirstytieji skaičiavimai

Šiandien daugelyje darbo sričių, pvz., inžinerijoje, prekyboje, gamyboje kasdien naudojami kompiuteriai ar kompiuterių klasteriai, kurie turi du (angl. *dualcore)* ar net keturis (angl. *quad core)* branduolius ir ne po vieną procesorių. Tokie kompiuteriai ar klasteriai vienu metu jau geba atlikti keletą veiksmų tuo pačiu metu. Lygiagretieji arba paskirstytieji skaičiavimai yra neišvengiami. Žemiau (1 pav.) pavaizduotas uždavinio sprendimas lygiagrečiuoju ir paskirstytuoju atveju, kai konkreti užduotis (problema) daloma į

smulkesnius uždavinius ir šie vykdomi vienu metu, t. y. sprendžiami lygiagrečiai arba paskirstytuoju būdu. Šiuolaikinės informacinės technologijos pažengusios į priekį, tačiau dar galima aptikti kompiuterius turinčius tik viena procesorių ir sprendžiančius uždavinius nuosekliai, kur uždavinys skaidomas į instrukcijas (2 pav.) [2,8,9,12,13,18,20].



1 pav. Užduoties vykdymas 4 branduolių procesoriuje



2 pav. Uždavinio vykdymas vieno branduolio kompiuterio procesoriuje

2.3.1. Lygiagretieji ir paskirstytieji kompiuteriai

Šiuolaikinių personalinių kompiuterių ar darbo stočių svarbiausios dalys yra procesoriai, kurie sudaryti iš valdymo, skaičiavimo įrenginių ir atminties blokų. Lygiagretieji ir paskirstytieji kompiuteriai turi po kelis skaičiavimo ir valdymo įrenginius, jie klasifikuojami į:

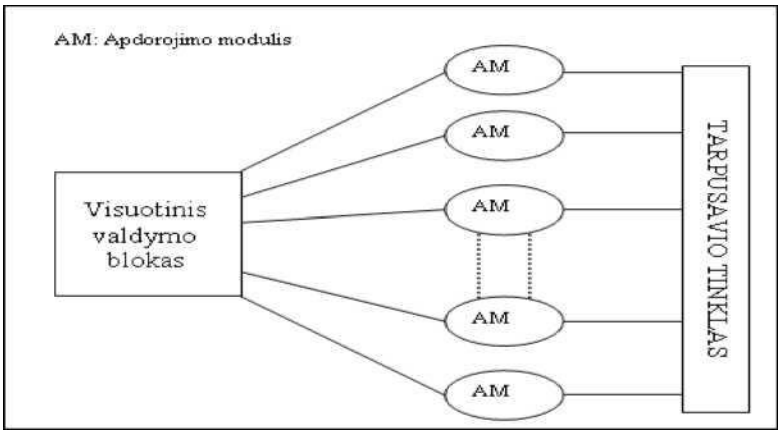
- Turinčius bendrą atmintį;
- Turinčius paskirstytą atmintį.

Paskirstytųjų kompiuterių pagrindinis skirtumas nuo lygiagrečiųjų kompiuterių yra tai, kad skaičiavimo, valdymo įrenginiai ir atminties blokai sujungti į bendrą visumą, pasinaudojus kompiuterinį tinklą ar internetą.

Kompiuterio skaičiavimo greitis priklauso nuo dviejų svarbiausių veiksnių: procesoriaus taktinio dažnio (kuris matuojamas hercais (Hz) ir parodo, kiek instrukcijų kompiuteris įvykdo per vieną sekundę) ir duomenų apsikeitimo tarp procesoriaus bei operatyviosios atminties greičio [2,8,9,12,13,18,20].

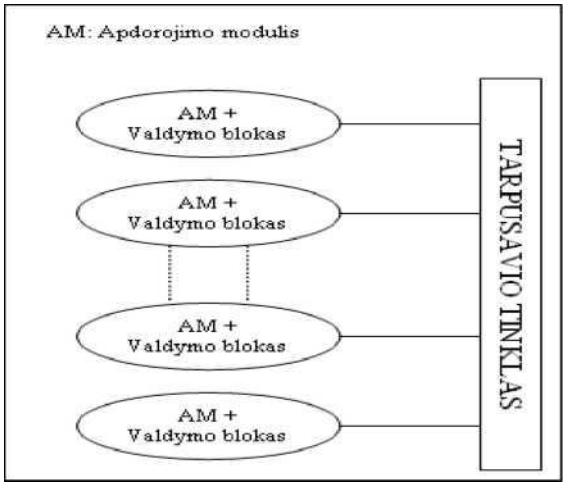
2.3.1.1. Bendrosios atminties lygiagretieji kompiuteriai

Kompiuteryje yra įdiegtas tik vienas valdantis įrenginys ir keli skaičiavimo įrenginiai (3 pav.) Atliekant skaičiavimus visi skaičiavimo įrenginiai skaičiuoja tą pačią operaciją su skirtingais duomenis arba nevykdo jokių veiksmų. Toks skaičiavimo modelis vadinamas SIMD tipo ***(Single Instruction stream – Multiple Data stream).*** Tokie lygiagretieji kompiuteriai yra ypač efektyvūs, kai atliekami veiksmai su matricomis, todėl SIMD tipo kompiuteriai dar vadinami matriciniais kompiuteriais [2,8,9,12,13,18,20].



3 pav. SIMD tipo lygiagretieji kompiuteriai

Jei kompiuteryje įrengti keli procesoriai, kuriuose yra individualūs valdymo ir skaičiavimo įrenginiai, tai turime MIMD tipo ***(Multiple Instruction stream – Multiple Data stream)*** skaičiavimo modelį (4 pav.). Tokiu kompiuteriu galima realizuoti daug bendresnius algoritmus, tačiau sunkiau sinchronizuoti procesorių darbą [2,8,9,12,13,18,20].



4 pav. MIMD tipo lygiagretieji kompiuteriai

Visi procesoriai skaičiuoja uždavinį su tam tikrais duomenimis, kuriuos perskaito ir užrašo į jiems skirtas atminties ląsteles. Bendrosios atminties lygiagretieji kompiuteriai turi tik vieną atminties bloką ir visi procesoriai bet kuriuo laiko momentu gali tiesiogiai pasiekti bet kurią atminties vietą. Kai duomenų persiuntimo greitis yra vienodas visiems procesoriams, turime kompiuterį su tolygiai pasiekiama bendrąja atmintimi. Tačiau dideliam procesorių kiekiui techniškai sunku realizuoti tokią sąlygą, todėl dažnai atmintis skaidoma į dalis, kurios priklauso skirtingiems procesoriams. Šiuo atveju išlieka bendras atminties adresavimas, bet procesoriai greičiau pasiekia duomenis, esančius lokalioje atminties dalyje, nei duomenis, esančius kituose procesoriuose, – tai kompiuteriai, turintys netolygiai pasiekiamą bendrąją atmintį (pasidalintos atminties lygiagretieji kompiuteriai) [2,8,9,12,13,18,20].

2.3.1.2. Paskirstytosios atminties lygiagretieji kompiuteriai

Paskirstytosios atminties lygiagretieji kompiuteriai priklauso MIMD tipui. Tačiau šiuo atveju kiekvienas procesorius gali tiesiogiai perskaityti ir įrašyti tik duomenis, esančius lokalioje atmintyje. Jeigu vykdant algoritmą reikia duomenų, kurie saugomi kito procesoriaus lokalioje atmintyje, tai pirmasis procesorius turi nusiųsti pranešimą antrajam su reikalinga informacija. Pirmasis procesorius laukia, kol ateis pranešimas su reikalinga informacija [2,8,9,12,13,18,20].

Duomenų mainai paskirstytosios atminties lygiagrečiuosiuose kompiuteriuose yra sudėtingesni nei bendrosios atminties lygiagrečiuose kompiuteriuose. Tarpusavio tinklo realizacijai gali būti pasitelkiamas kompiuterinis tinklas ar internetas. Dėl to galima išvengti

problemos, kuri egzistuoja bendrosios atminties kompiuteriuose, kai keli procesoriai vienu metu bando skaityti ir / arba rašyti į tą pačią atminties vietą. Kadangi visus pakeitimus gali atlikti tik tas procesorius, kurio lokalioje atmintyje saugomi duomenys, tai nesunku kontroliuoti duomenų mainus ir jų eiliškumą [2,8,9,12,13,18,20].

2.3.2. Lygiagrečiųjų ir paskirstytųjų algoritmų vertinimo kriterijai

Lygiagretieji ir paskirstytieji algoritmai analizuojami naudojant lygiagretinimo koeficientus. Dažniausiai naudojamas lygiagrečiojo algoritmo spartinimo koeficientas:

$$S_p = \frac{t_1}{t_p}, \tag{2.1}$$

čia t_p yra algoritmo sprendimo, naudojant p procesorių, trukmė, o t_1 yra laikas, per kurį uždavinys išsprendžiamas geriausiu žinomu nuosekliuoju algoritmu. Spartinimo koeficientas, padalytas iš procesorių skaičiaus, yra vadinamas algoritmo efektyvumo koeficientu:

$$e_p = \frac{s_p}{p}. \tag{2.2}$$

Paprastai $1 < s_p < p$ ir $0 < e_p < 1$ [2,8,9,12,13,18].

Efektyvumo koeficientas parodo, kaip efektyviai panaudojami procesoriai. Sprendžiant kai kuriuos uždavinius, didelis efektyvumas gali būti pasiektas tik esant fiksuotam procesorių kiekiui, tačiau toliau jam didėjant, efektyvumas mažėja. Efektyvumas priklauso ne tik nuo uždavinio, bet ir nuo lygiagrečiojo algoritmo bei lygiagrečiųjų kompiuterių parametrų: granuliacijos, procesorių skaičiavimo greičio, komunikacijos greičių ir pan. To paties algoritmo efektyvumas skirtinguose kompiuteriuose ar jų sistemose gali būti labai skirtingas. Paprastai, kuo didesnis procesorių komunikacijos greitis ir kuo lėtesni procesoriai, tuo geresnis efektyvumas yra pasiekiamas. Daugiaprocesoriniuose superkompiuteriuose santykis tarp procesorių taktinio dažnio ir komunikacijos tarp procesorių greičio yra artimas vienetui, todėl lygiagrečiojo algoritmo efektyvumą daugiausia lemia to algoritmo savybės [2,8,9,12,13,18].

II. Projektinė dalis

1. Įrankių ir priemonių pasirinkimo analizė

Prieš pradedant išlygiagretinti nuosekliąją programą ir sukurti programų paketą buvo atliekama esamų sprendimų lygiagretinimo sferoje analizė. Jos metu nustatyta, kad šiam uždaviniui įgyvendinti galima pasinaudoti openMP standartu arba pranešimų perdavimo sąsaja (MPI), arba lygiagrečių modelių biblioteka (angl. Parallel Pattens Library (PPL)), arba

QTCONCURRENT biblioteka. Toliau darbe būtų tikslinga detaliau aptarti kiekvieną iš šių galimybių, pabrėžiant jų trūkumus ir privalumus [3,4,7,8,9,11,12,13,14,15,17,22].

OpenMP standartas skirtas lygiagretiesiems algoritmams realizuoti bendrosios atminties kompiuteriuose. Atsižvelgiant į tai, kad kompiuteriuose naudojama bendra atmintis visiems procesoriams, programuotojui nereikia rūpintis duomenų pasikeitimu tarp skirtingų procesorių ir tai yra vienas iš šio standarto privalumų, tačiau reikia išskirti tas algoritmo dalis, kurios gali būti vykdomos vienu metu. Šiame standarte naudojamas baigtinis procesorių skaičius. Programuotojas turi nurodyti, kiek procesorių ir kurią programos dalį turi vykdyti pasirinktieji procesoriai. Ši savybė kai kuriomis situacijomis yra labai naudinga, tačiau atsižvelgiant į universalumą – ši savybė nėra labai naudinga, nes programa parašyta šiuo standartu veiktų tik tam tikruose kompiuteriuose, t. y. visi kompiuteriai, kuriuose norėtumėte vykdyti programą, privalėtų turėti vienodą procesorių skaičių, priešingu atveju turėtumėte koreguoti programinį kodą, nurodydami kitus procesorių kiekius [3,8,9,22].

Pranešimų perdavimo sąsaja (MPI) – tai biblioteka, kuri skirta lygiagretiesiems algoritmams realizuoti paskirstytosios atminties kompiuteriuose. Ji apibrėžia paprogramių vardus, parametrus ir jų funkcinę paskirtį. Šią biblioteką galima naudoti rašant programą C/C++ ir Fortran 77/95 programavimo kalbomis. MPI biblioteka apibrėžia tiesiogines (angl. *point to point*) komunikacijas, kai komunikacija vyksta tik tarp dviejų procesų. Ši biblioteka taip pat apibrėžia ir kolektyvines (angl. *collective*) komunikacijas, kai komunikacija ar sinchronizacijos operacijos įtraukia procesų grupes [4,8,9].

Programuotojas, naudodamas MPI standartą, privalo pats pasirūpinti duomenų perdavimu tarp programą vykdančių procesų, nes vienas procesas vienu metu gali tik siųsti arba tik priimti duomenis [4,8,9,11].

Lygiagrečių modelių biblioteka (PPL) skirta programų kūrimui C++ programavimo kalba, kur skaičiavimai ar užduotys atliekami lygiagrečiai. Ši biblioteka atsirado drauge su programavimo įrankiu Visual Studio 2010. Šios bibliotekos pagalba lanksčiai ir nesudėtingai galima pasinaudoti lygiagretaus skaičiavimo savybėmis. Pasinaudojus šia biblioteka, programuotojui nereikia galvoti apie duomenų apsikeitimą tarp procesų, taip pat nereikia nurodinėti, kiek procesorių atlikinės konkrečią užduotį. Programuotojui, rašant programinį kodą, reikia nurodyti, kuri programos dalis turi būti atliekama lygiagrečiai [14,17].

QTCONCURRENT biblioteka skirta programų kūrimui C++ programavimo kalba. Ši biblioteka yra integruota į QtCreator programavimo įrankį, kuris yra nemokamas ir viešai laisvai prieinamas. Dirbant su šia biblioteka, taip pat kaip ir su PPL biblioteka, programuotojui nereikia galvoti apie duomenų apsikeitimą tarp procesų ir nurodinėti kiekio

užduoties realizavimui, nes šių bibliotekų pagalba jos nusistato, kiek procesorių gali naudoti skaičiavimams [7].

QTCONCURRENT ir PPL bibliotekų pagrindinis skirtumas yra sintaksėje. QTCONCURRENT vykdoma funkcija (pvz. mapped (seka, funkcija)), kurioje nurodoma seka (dažniausiai STL bibliotekos list arba vector konteinerio pavidalu) ir lygiagrečiai vykdoma funkcija, o naudojantis PPL bibliotekos funkcija: parallel_for (pradžia, pabaiga, žingsnis) {programinis kodas} [7,14,17].

Iškeltiems uždaviniams pasiekti buvo pasirinkta QTCONCURRENT biblioteka, nes ji yra integruota į QtCreator programavimo įrankį, kuris yra nemokamas, o Visual Studio 2010 yra mokamas produktas. Taip pat atsisakyta openMP ir MPI bibliotekų, nes jose, norint vykdyti lygiagrečias dalis, reikia apibrėžti procesorių kiekius, o tai trukdo padaryti programą visiškai nepriklausomą nuo kompiuterių architektūros ir procesorių skaičiaus.

Kuriamam programų paketui vietoj statinių masyvų buvo panaudota standartinių šablonų biblioteka (angl. Standart Template Library (STL)). STL – duomenų struktūrų ir jiems apdoroti skirtų algoritmų biblioteka. Duomenų struktūros vadinamos konteineriais. Konteineriai skirstomi į:

- Nuosekliuosius (masyvai, vektoriai, sąrašai, eilės);
- Asocijuotuosius (aibės, atvaizdžiai) [5].

STL bibliotekos algoritmai leidžia atlikti visus įmanomus veiksmus su konteineriais – tokius kaip duomenų elementų paieška, kopijavimas, sukeitimas, rūšiavimas ir pan. [5]

STL konteinerių pagalba išspręsta problema dėl išankstinio masyvo elementų skaičiaus apibrėžimo.

2. Projekto vykdymo planas

Planuojamas projekto vykdymo planas.

Metai	2010																
Mėnuo	Rugsėjis				Spalis				Lapkritis				Gruodis				
Savaitės nr. / darbo nr.	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
1																	
2																	
3																	
4																	
5																	
6																	
7																	

Metai	2011																								
Mėnuo	Sausis				Vasaris				Kovas				Balandis				Gegužė					Birželis			
Savaitės nr. / darbo nr.	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
1																									
2																									
3																									
4																									
5																									
6																									
7																									

Metai	2011																								
Mėnuo	Liepa					Rugpjūtis				Rugsėjis				Spalis				Lapkritis				Gruodis				
Savaitės nr. / darbo nr.	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
1																											
2																											
3																											
4																											
5																											
6																											
7																											

Metai	2012																				
Mėnuo	Sausis				Vasaris				Kovas				Balandis					Gegužė			
Savaitės nr. / darbo nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1																					
2																					
3																					
4																					
5																					
6																					
7																					

1. Konsultavimasis su darbo vadovu (susitikimai, el. paštu).
2. Teorijos analizė.
3. Nuoseklios programos analizė ir projektavimas.
4. Programavimo darbai.
5. Testavimo/klaidų taisymo darbai.
6. Eksperimentų atlikimas.
7. Darbo aprašymo rašymas.

3. Pradinis projekto aprašas

Šio projekto vienas iš pagrindinių tikslų išlygiagretinti nuosekliąją programą ir ją pateikti kaip programų paketą spietinei internetinei statistinei modeliavimo ir stochastinio programavimo sistemai. Todėl buvo pasirinkta K. Žilinsko (2007) nuoseklioji programa, realizuojanti Monte Karlo metodą [26]. Ši programa parašyta C kalba. Joje naudojami statiniai masyvai ir skaičiavimams ji naudoja vieną procesorių. Jos skaičiavimo laikai prie skirtingų tikslumo (**ε** - 0,5;1;2) reikšmių, bet su tais pačiais pradiniais duomenimis, pateikiami 5 pav.

III. Darbo eigos aprašymas

1. Darbų eigos grafas

Metai	2010																
Mėnuo	Rugsėjis				Spalis				Lapkritis				Gruodis				
Savaitės nr. / darbo nr.	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
1																	
2																	
3																	
4																	
5																	
6																	
7																	

Metai	2011																								
Mėnuo	Sausis				Vasaris				Kovas				Balandis				Gegužė					Birželis			
Savaitės nr. / darbo nr.	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
1																									
2																									
3																									
4																									
5																									
6																									
7																									

Metai	2011																												
Mėnuo	Liepa					Rugpjūtis				Rugsėjis				Spalis					Lapkritis				Gruodis						
Savaitės nr. / darbo nr.	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52		
1																													
2																													
3																													
4																													
5																													
6																													
7																													

Metai	2012																				
Mėnuo	Sausis				Vasaris				Kovas				Balandis					Gegužė			
Savaitės nr. / darbo nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1																					
2																					
3																					
4																					
5																					
6																					
7																					

1. Konsultavimasis su darbo vadovu (susitikimai, el. paštu).
2. Teorijos analizė.
3. Nuoseklios programos analizė ir projektavimas.
4. Programavimo darbai.
5. Testavimo/klaidų taisymo darbai.
6. Eksperimentų atlikimas.
7. Darbo aprašymo rašymas.

2. Problemų ir jų sprendimų aprašymai ir pagrindimai

Viena iš didžiausių problemų – nustatyti nuosekliosios programos dalis ar dalį, kuri gali būti vykdoma lygiagrečiai. Šios problemos sprendimo raktas slypi nuoseklioje programos kodo analizėje, kurios metu identifikuojami globalieji ir lokalieji kintamieji, jų panaudojimas funkcijose. Analizės metu taip pat yra nustatomi ryšiai tarp funkcijų.

Identifikavus globaliuosius ir lokaliuosius kintamuosius susiduriama su problema, kai programos kodas, atliekamas lygiagrečiai, naudoja tuos pačius globaliuosius kintamuosius ir juose saugomus duomenis. Šios problemos sprendimui galimos kelios alternatyvos. Viena iš

jų – naudoti semaforus[6] globaliesiems kintamiesiems, kuriuose saugomi duomenys. Deja, šis sprendimas turi aiškiai matomą trūkumą: jeigu vienu metu keli procesai norėtų pasiekti tą patį globalaus kintamojo duomenį, šie procesai turėtų sustoti į „virtualią eilę“ ir laukti to momento, kai prieš jį buvęs procesas atlaisvins kintamąjį, nes semaforo pagrindinė mintis – procesas, naudojantis kintamąjį, „užrakina“ prieigą prie šio kintamojo kitiems procesams. Tik pirmajam eilėje procesui baigus naudotis kintamuoju, semaforo pagalba „atrakinamas“ kintamasis ir kiti procesai gali naudotis šio kintamojo duomenimis. Antra alternatyva – vietoj globaliųjų kintamųjų naudoti lokaliuosius kintamuosius, kurie tarp skirtingų procesų neturi jokių ryšių.

Funkcijos, kurios skaičiuojamos lygiagrečiai naudojant vien lokaliuosius kintamuosius, susiduria su problema, kai reikia paduoti pradinius duomenis šiai funkcijai. Šios problemos sprendimas – kiekvienam procesui paduodami unikalūs duomenys, pasinaudojus STL bibliotekos konteinerių pagalba ir juose saugant struktūras.

Programų universalumui dažniausiai trukdantis yra statinių masyvų naudojimo faktorius. Šią problemą taip pat galima išspręsti pasitelkus STL bibliotekos konteinerių pagalbą.

3. Galutinio projekto būvio aprašymas

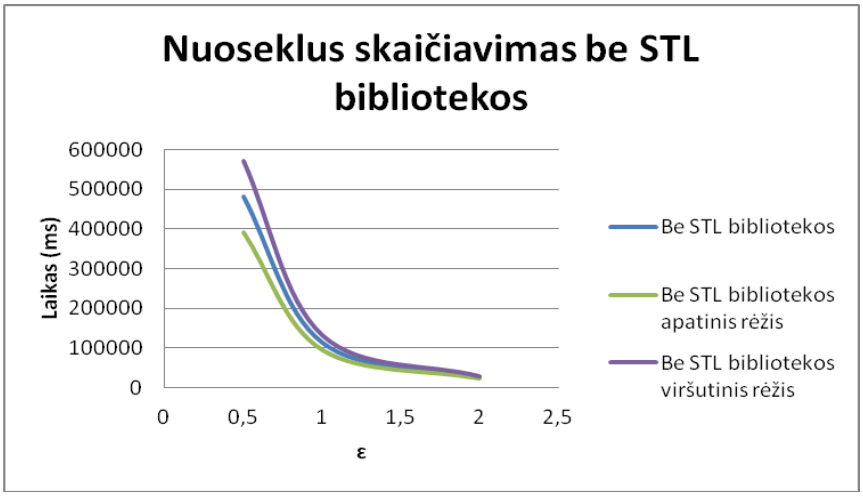
Nuoseklioji Monte Karlo metodą realizuojanti programa išlygiagretinta ir gali būti paleista vykdymui paskirstytose sistemose. Taip pat vietoj statinių masyvų panaudoti STL bibliotekos konteineriai. Darbo eigoje buvo identifikuota programos kodo dalis, kuri gali būti atliekama lygiagrečiai. Šiai kodo daliai sukurta nauja funkcija, kurioje atliekami visi reikalingi skaičiavimai. Šios sukurtos funkcijos lygiagrečiam skaičiavimui pasinaudota QTCONCCURENT funkcija mapped (seka, funkcija1). Mapped funkcija atlieka tiek funkcija1 lygiagrečių skaičiavimų, kiek jų nurodoma elementų sekoje. Šiai mapped funkcijai paduodamas STL bibliotekos List konteineris kaip seka. Tuo pačiu šiame konteineryje saugomi pradiniai duomenys, struktūriniu pavidalu funkcijai funkcija1. Mapped funkcija, vykdydama pirmąjį sekos elementą, funkcijai perduoda ir konteineryje išsaugotus pradinius duomenis. Ši funkcija panaudoja tiek procesorių, kiek jų yra programą vykdančiame kompiuteryje. Mapped funkcija, atlikus visus skaičiavimus pagal seką, grąžina konteinerį su duomenimis, kurie buvo grąžinti iš funkcija1. Su konteineryje saugomais duomenimis toliau atliekami reikalingi skaičiavimai.

[6] **Semaforas** – struktūra, turinti ją sukuriant nustatytą leidimų skaičių vykdymo gijoms dirbti su saugoma duomenų struktūra ar vykdyti saugomą kodo sekciją. [http://lt.wikipedia.org/wiki/Semaforas_(programavimas)]

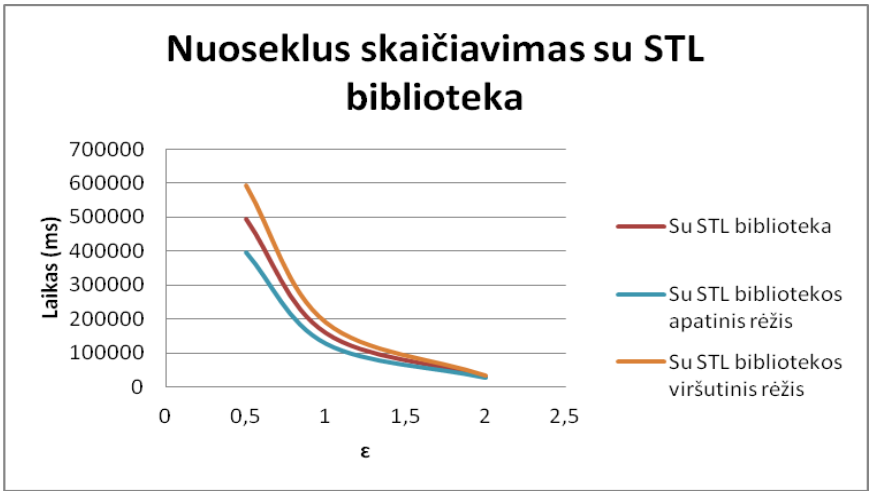
Taip pat išlygiagretintoje programoje atsisakyta statinių masyvų. Vietoj jų naudojami STL bibliotekos Vector konteineriai, tam kad programa įgautų daugiau universalumo.

4. Darbo rezultatų analizė

Šio darbo tikslas ir uždaviniai buvo sėkmingai įgyvendinti. Taip pat papildomai buvo ištirtas nuoseklios programos vykdymo laikas, kai vietoj statinių masyvų panaudoti STL bibliotekos konteineriai (6 pav.).



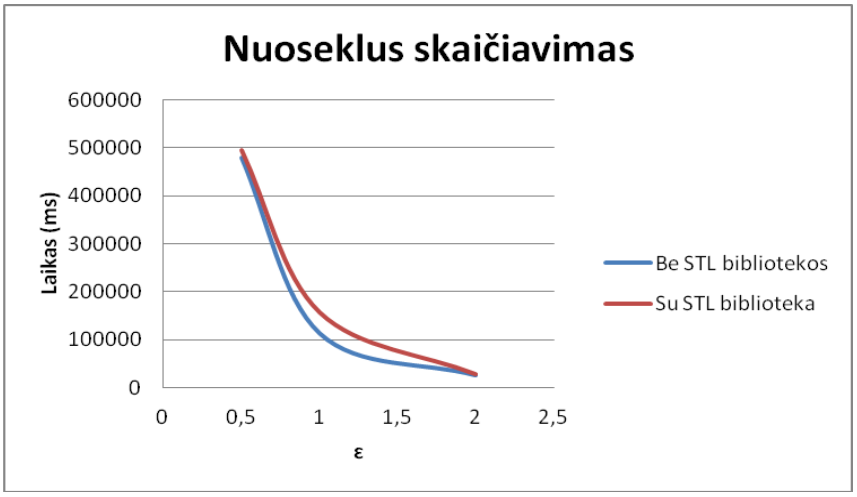
5 pav. Monte Karlo metodo nuoseklios programos be STL bibliotekos skaičiavimo vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95



6 pav. Monte Karlo metodo nuoseklios programos su STL bibliotekos skaičiavimo vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95

Palyginus skaičiavimų vykdymo laikus (7 pav.), labai kardinalus skirtumas tarp šių programų nepastebimas. Grafike akivaizdžiai matyti, jog nuoseklios programos su statiniais masyvais vykdymo laikas trumpesnis už nuoseklios programos su STL biblioteka vykdymo laiką. Tačiau siekiant, kad programa būtų kuo universalesnė ir negalvojant apie atminties

išskyrimą programos pradžioje, tikslinga naudotis STL bibliotekos pagalba. Taip pat galima pastebėti, kad mažinant tikslumo koeficientą **ε,** programos vykdymo laikas trunka mažiau.



7 pav. Monte Karlo metodo nuoseklių skaičiavimų vidutinis laikas su STL biblioteka ir be STL bibliotekos

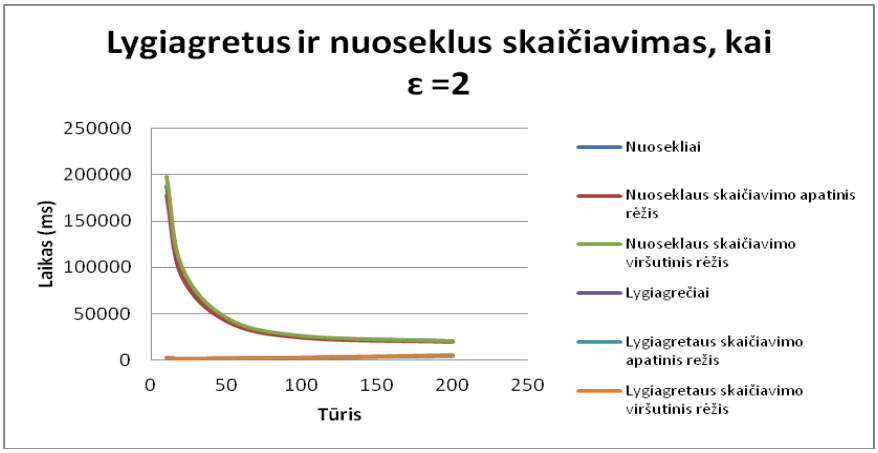
Su išlygiagretinta programa buvo atlikta 3000 eksperimentų su testiniais duomenimis, kurie skirti spęsti uždavinį, kuriame yra 20 pirmo etapo kintamųjų ir 20 antro etapo ribojimų (Priedas nr. 1). Taip pat buvo atlikta 300 eksperimentų su testiniais duomenimis, kuriuose yra 60 kintamųjų ir 60 ribojimų. Visi eksperimentai buvo atliekami tuo pačiu kompiuteriu. Šio kompiuterio duomenys: centrinis procesorius Intel Xeon X5670 4 branduolių 2,93 GHz, 48 GB RAM, operacinė sistema Windows server 2008 R2 Enterpice SP1. Visi eksperimentai buvo atlikti keičiant tikslo funkcijos tikslumo reikšmę ε, kuri įgijo reikšmes 0,5, 1, ir 2. Taip pat keičiant lygiagrečiai skaičiuojamų ciklų skaičių (skaičiuojamų scenarijų kiekis visiems procesoriams, o ne kiekis vienam procesoriui), arba kitaip vadinamą tūrį, buvo suteiktos šios reikšmės: 10, 20, 50, 100 ir 200. Eksperimentai su išlygiagretinta programa buvo atliekamai ir nuosekliai t. y. šią programą paleidus priverstinai ant vieno kompiuterio branduolio.

Lygiagretaus ir nuoseklaus skaičiavimo eksperimentų rezultatai (Priedas nr. 2) pateikiami 8 pav., kai ε=2. Galima pastebėti, jog kintant tūriui, t. y. jam didėjant, nuoseklaus skaičiavimo laikas trumpėja (9 pav.).

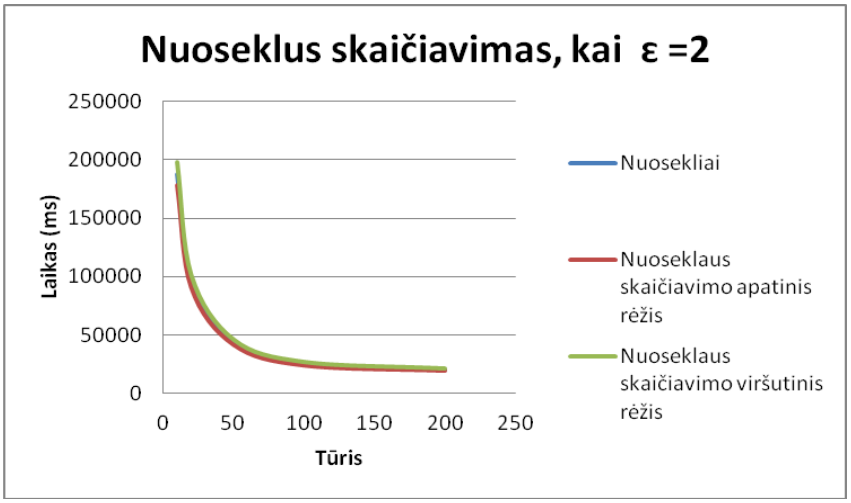
Taip pat pastebėtina, jog pasikliautųjų intervalų (1.1) ilgis sudaro apie 10 % vidutinio programos vykdymo laiko atžvilgiu – tai reiškia, jog rasta tikslo funkcijos reikšmė yra patikima. Pasikliautinųjų intervalų plotis nekinta keičiant programos vykdyme naudojamą tūrį, skaičiuojant nuosekliai, tačiau skaičiuojant uždavinį lygiagrečiai (10 pav.) galima pastebėti, kad pasikliautinųjų intervalų plotis didėja atitinkamai didėjant tūriui. Iš 10 pav.

galima pastebėti, kad programos vykdymo laikas didėja, didėjant tūriai. Šiems testiniams duomenims ir tikslumui ε=2 optimalus tūris būtų 20.

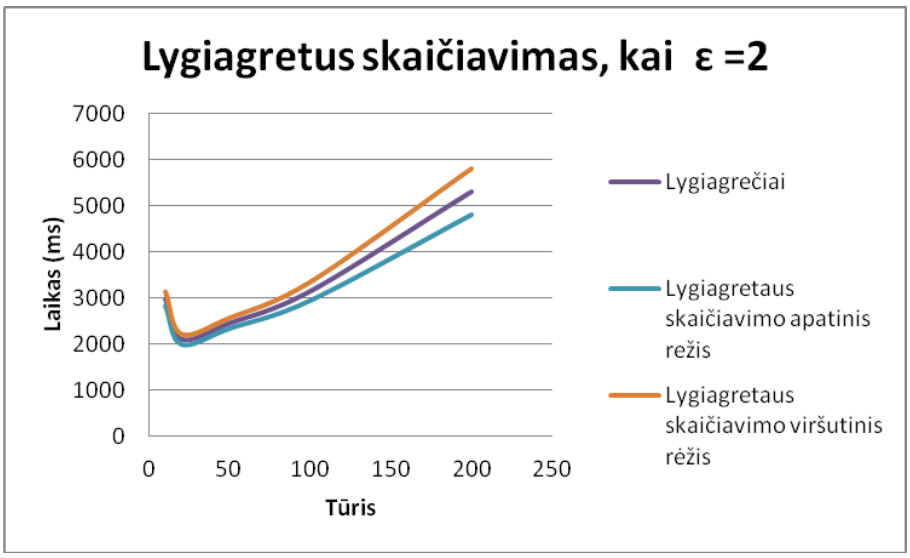
Pagal gautus rezultatus buvo apskaičiuoti spartinimo koeficientai ir efektyvumo koeficientai, kurie pateikti 1 lentelėje. Didžiausias spartinimo ir efektyvumo koeficientas gaunamas, kai tūris lygus 10.



8 pav. Lygiagrečių ir nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=2



9 pav. Nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=2



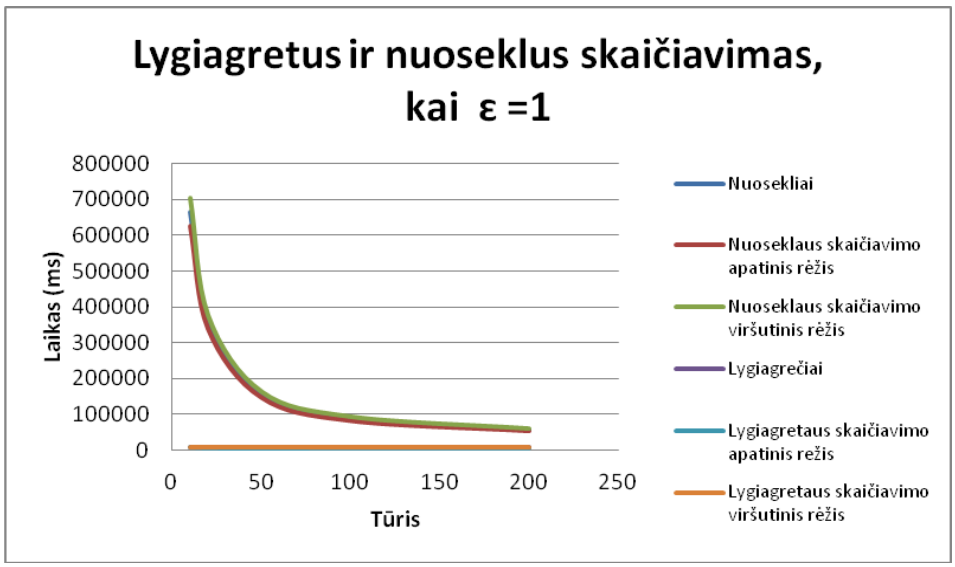
10 pav. Lygiagrečių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=2

Tūris	Spartinimo koeficientas	Efektyvumo koeficientas
10	63,07	15,77
20	45,66	11,41
50	17,89	4,47
100	8,08	2,02
200	3,82	0,96

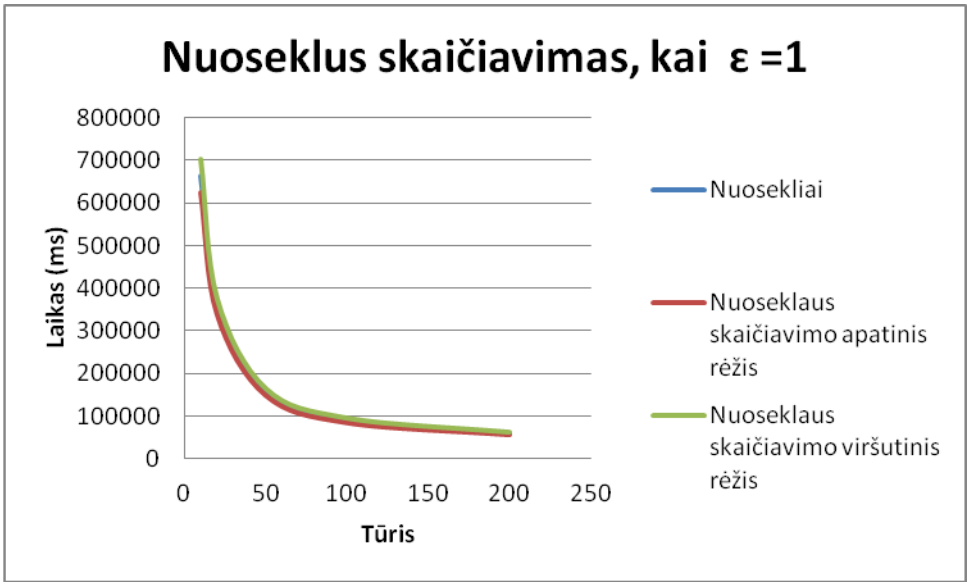
1 lentelė. Lygiagrečios programos spartinimo (2.1) ir efektyvumo (2.2) koeficientai, kai ε=2

Lygiagretaus ir nuoseklaus skaičiavimo eksperimentų rezultatai (Priedas nr. 2) pateikiami 11 pav., kai ε=1. Šių eksperimentų rezultatų tendencijos mažai skiriasi nuo anksčiau aptartų. Padidinus norimą tikslumą, pailgėjo ir skaičiavimo laikas. Pateiktuose nuoseklaus skaičiavimo rezultatų (12 pav.) duomenyse galima pastebėti, kad didėjant tūriui, skaičiavimo laikas mažėja, bet pasikliautinųjų intervalų pločiui tai neturi jokios įtakos. Pasinaudojus lygiagrečių skaičiavimų rezultatų duomenimis, galima teigti, jog šiems testiniams duomenims prie šio tikslumo optimalus tūris būtų 50. 13 pav. matyti, kad tikslo funkcijos reikšmė randama greičiausiai, kai tūris lygus 50.

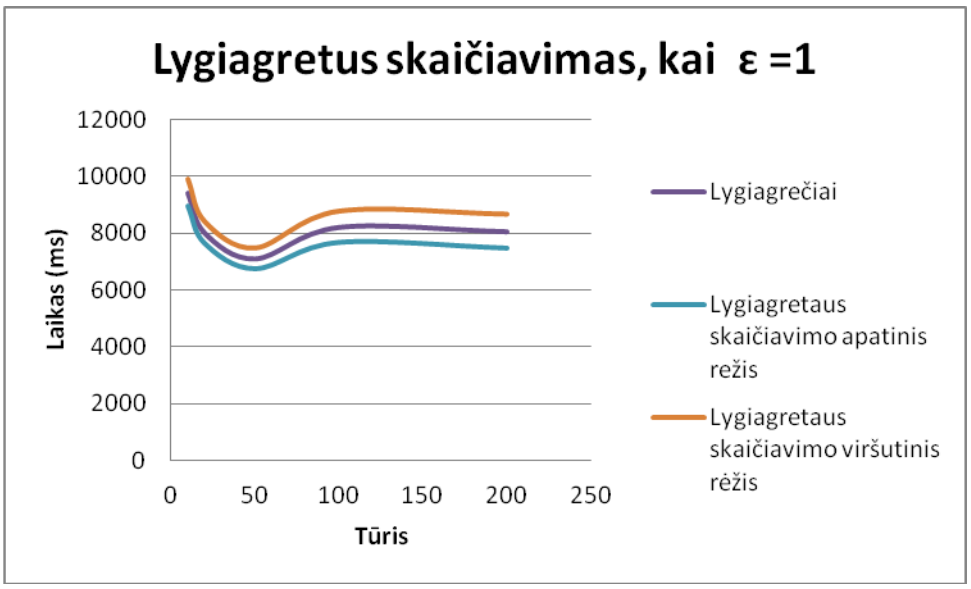
Pasinaudojus gautais eksperimento duomenims apskaičiuoti spartinimo ir efektyvumo koeficientai. Geriausias koeficientas gautas, kai tūris – 10 (2 lentelė).



11 pav. Lygiagrečių ir nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=1



12 pav. Nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=1

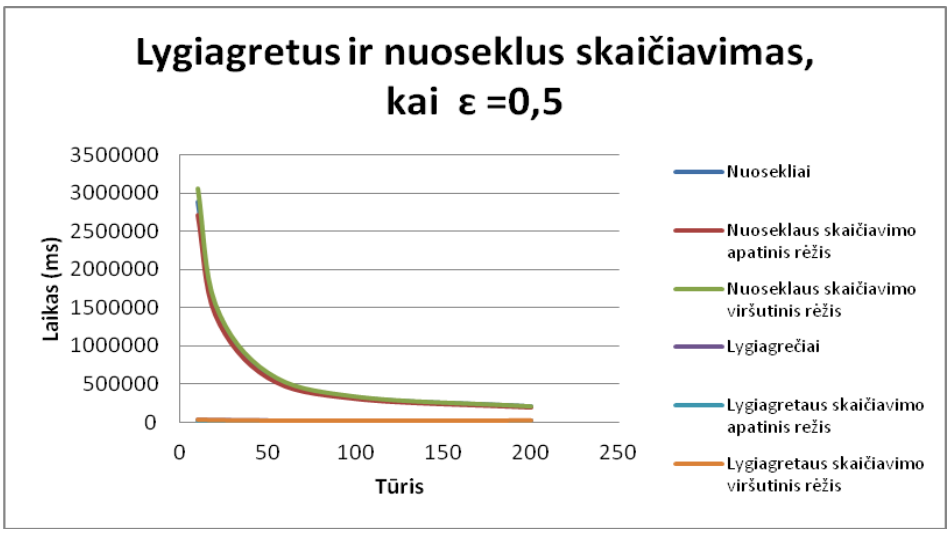


13 pav. Lygiagrečių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=1

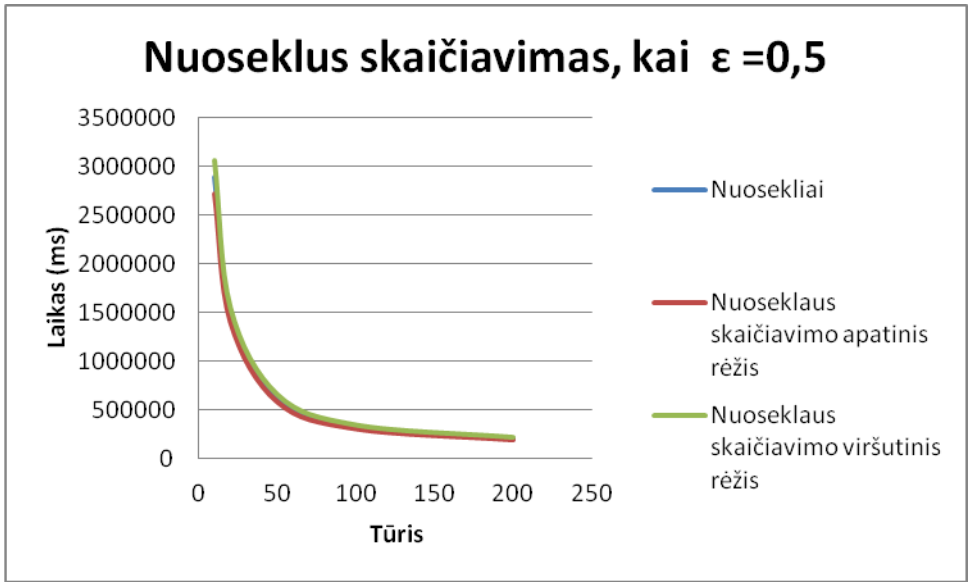
Tūris	Spartinimo koeficientas	Efektyvumo koeficientas
10	70,46	17,61
20	44,64	11,16
50	21,93	5,48
100	10,83	2,71
200	7,22	1,80

2 lentelė. Lygiagrečios programos spartinimo (2.1) ir efektyvumo (2.2) koeficientai, kai ε=1

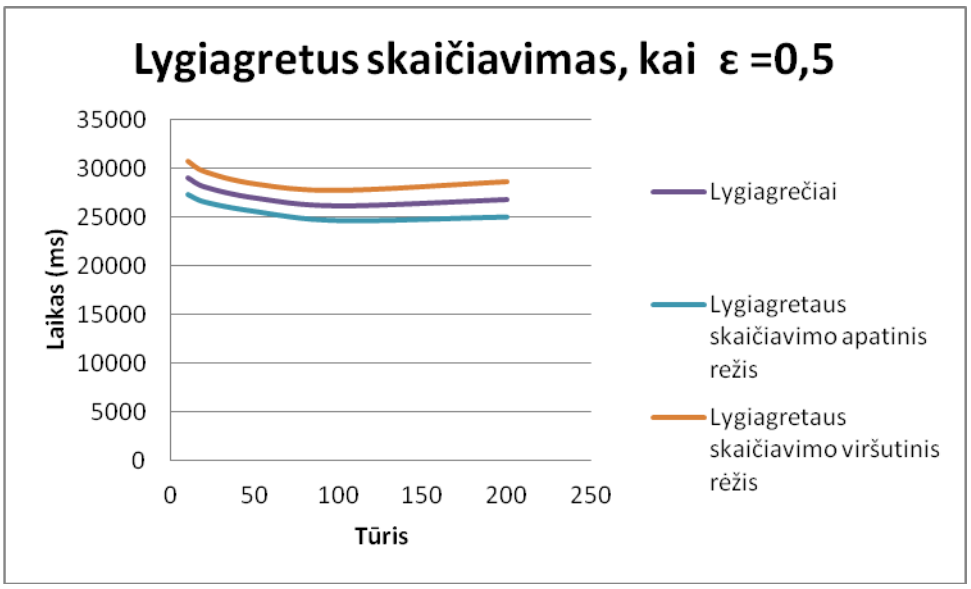
Gauti eksperimentų rezultatai (Priedas nr. 4), kai ε=0,5, pateikiami 14 pav. Kaip ir prieš tai atliktuose eksperimentuose, skaičiavimo laiko tendencija nesikeičia ir galima pastebėti, kaip tūrio didėjimas daro įtaką tikslo funkcijos reikšmės radimo greičiui (15 pav.). Pasinaudojus 16 pav. pateiktos diagramos duomenimis, galima teigti, jog lygiagrečiam skaičiavimui optimalus tūris yra 100. Tačiau ir pasirinkus tūrį 200, didelio skirtumo funkcijos reikšmės radimo greičiui tai neturės. Kaip ir prieš tai atliktuose eksperimentuose, didžiausias spartinimo ir efektyvumo koeficientas išlieka pasirinkus tūrį 10 (3 lentelė).



14 pav. Lygiagrečių ir nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=0,5



15 pav. Nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=0,5

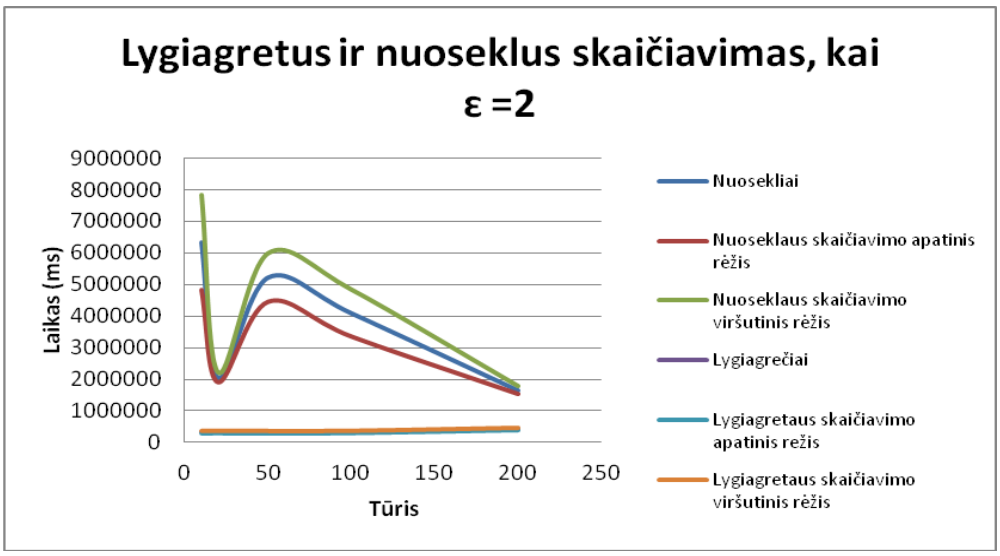


16 pav. Lygiagrečių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=0,5

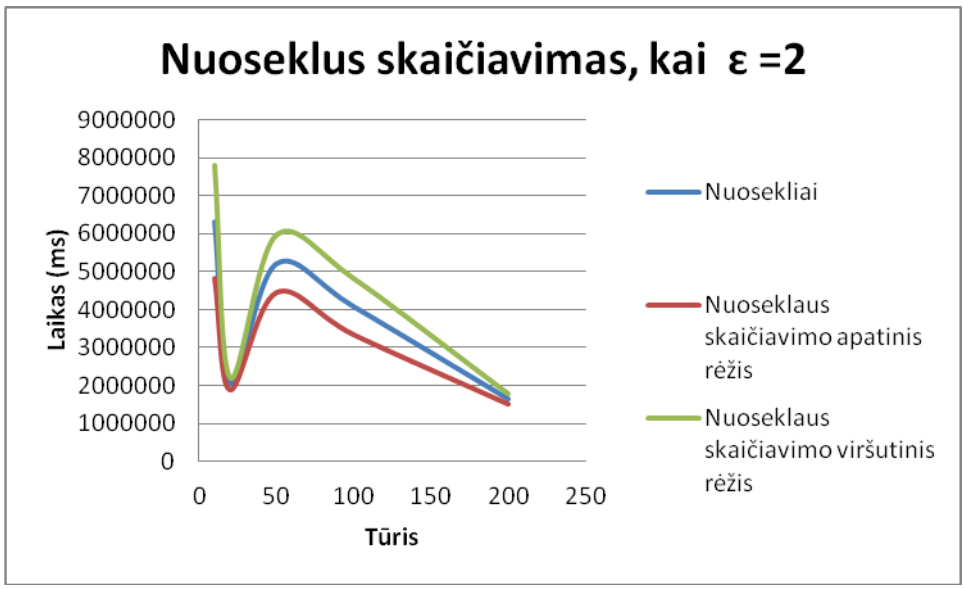
Tūris	Spartinimo koeficientas	Efektyvumo koeficientas
10	99,46	24,87
20	52,56	13,14
50	22,86	5,71
100	12,35	3,09
200	7,57	1,89

3 lentelė. Lygiagrečios programos spartinimo (2.1) ir efektyvumo (2.2) koeficientai, kai ε=0,5

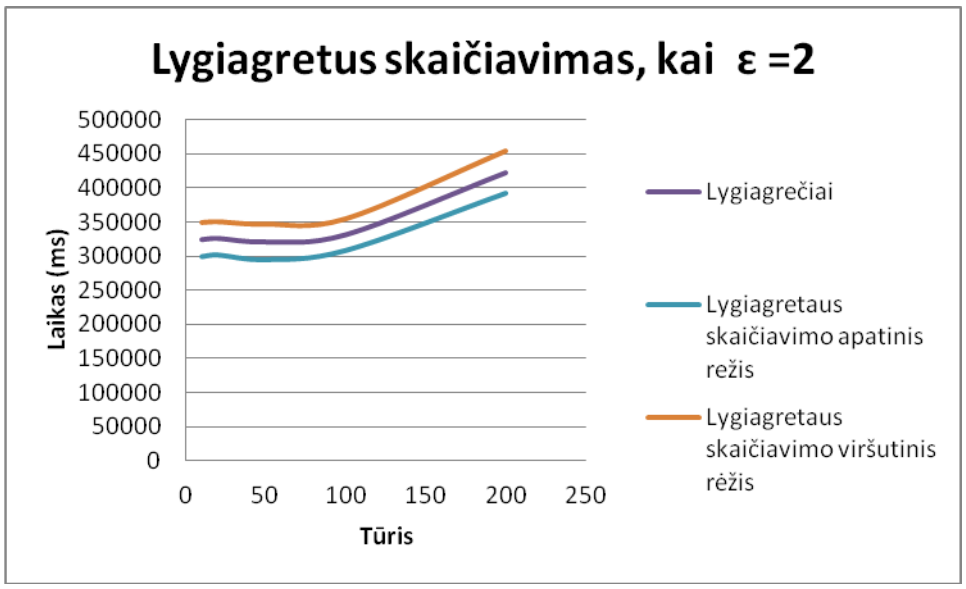
Atlikus eksperimentus su išlygiagretinta programa, kur testiniai duomenys skirti spęsti uždavinį kuriame yra 60 pirmo etapo kintamųjų ir 60 antro etapo ribojimų ir ε=2, gauti rezultatai (Priedas nr. 5) pateikti 17 pav. Iš gautų rezultatų galima pastebėti, jog atliekant skaičiavimus nuosekliai (18 pav.), greičiausiai randamos tikslo funkcijos reikšmės norimu tikslumu, kai tūris yra 20 ir 200. Taip pat šiuose taškuose pasikliautinųjų intervalų rėžiai (viršutinis ir apatinis) mažiausiai nutolę nuo gautų reikšmių lyginant su kitomis reikšmėmis. Lygiagretaus skaičiavimo rezultatai pateikiami 19 pav. Galima pastebėti, kad skaičiavimo laikas didėja, kai tūris yra didesnis nei 100, o pasikliautinųjų intervalų rėžių nutolimo dydžiai nekinta keičiantis tūriui. Didžiausias spartinimo ir efektyvumo koeficientas gaunamas, kai tūris yra 10 (4 lentelė).



17 pav. Lygiagrečių ir nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=2



18 pav. Nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=2



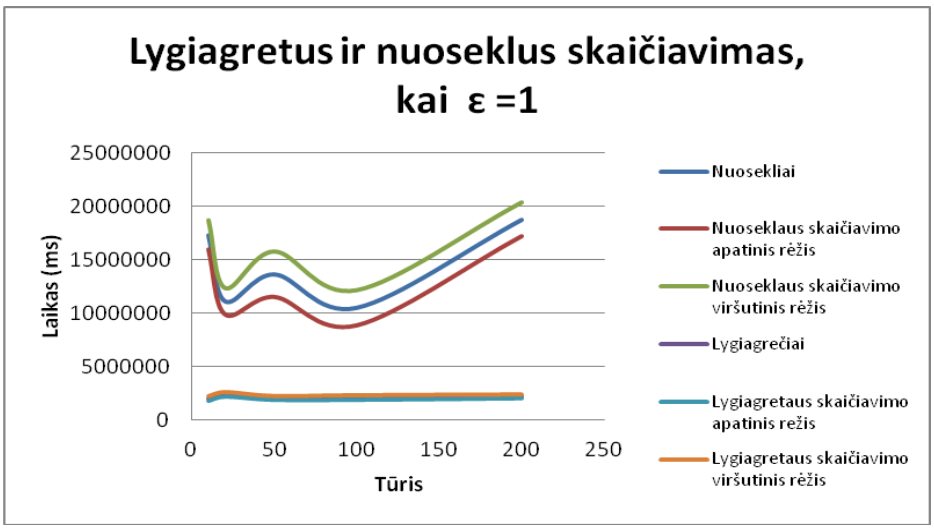
19 pav. Lygiagrečių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=2

Tūris	Spartinimo koeficientas	Efektyvumo koeficientas
10	19,48	4,87
20	6,31	1,58
50	16,25	4,06
100	12,33	3,08
200	3,92	0,98

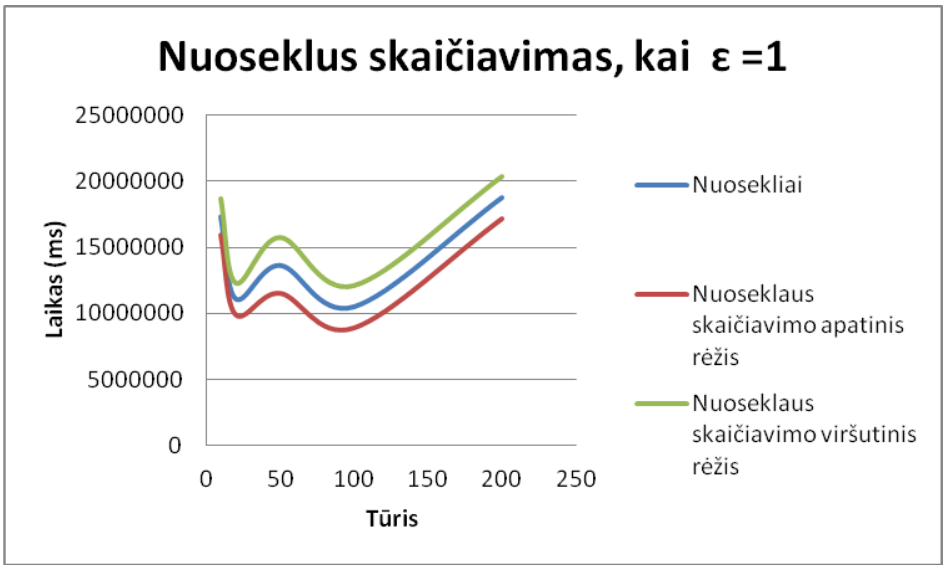
4 lentelė. Lygiagrečios programos spartinimo (2.1) ir efektyvumo (2.2) koeficientai, kai ε=2

Eksperimentų rezultatai (Priedas nr. 6), kai ε=1 pateikti 20 pav. Analizuojant rezultatus galima pastebėti, kad skaičiuojant nuosekliai, didinant tūrį daugiau nei 100, tikslo funkcijos reikšmės radimo laikas didėja. Iš 21 pav. galima matyti, kad tikslo funkcijos radimo geriausius laikus įmanoma gauti pasirinkus tūrį 20 arba 100, o pasikliautinųjų intervalų rėžiai vienodai nutolę, nepaisant tūrio kitimo.

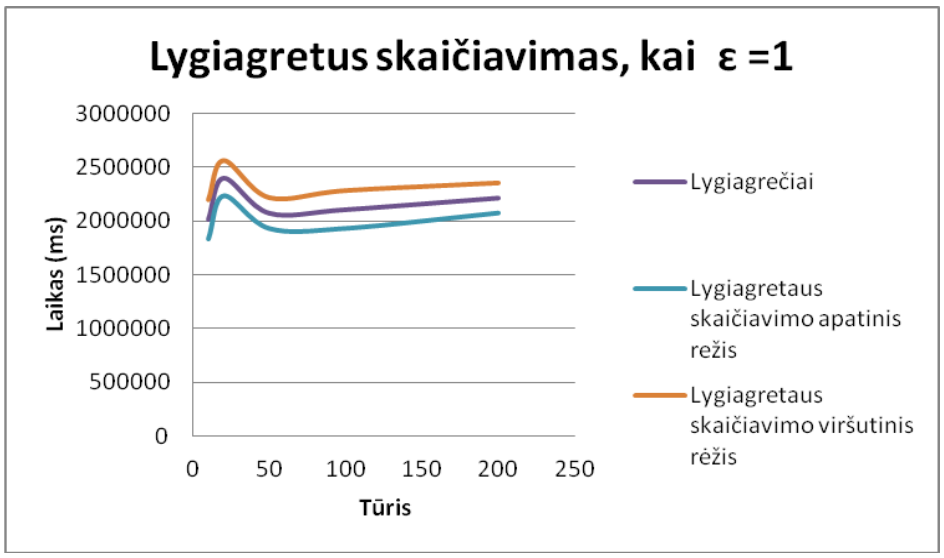
Lygiagretaus skaičiavimo rezultatai pateikti 22 pav. Akivaizdu, jog lygiagrečiai skaičiuojant, skaičiavimo laikas didėja didinant tūrį, tačiau kaip matyti iš grafiko – didžiausias skaičiavimo laikas yra kai tūris yra 20, o geriausias laikas – kai tūris yra 10. Šio eksperimento didžiausi spartinimo ir efektyvumo koeficientai gaunami, kai tūris yra 10 (5 lentelė).



20 pav. Lygiagrečių ir nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=1



21 pav. Nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=1



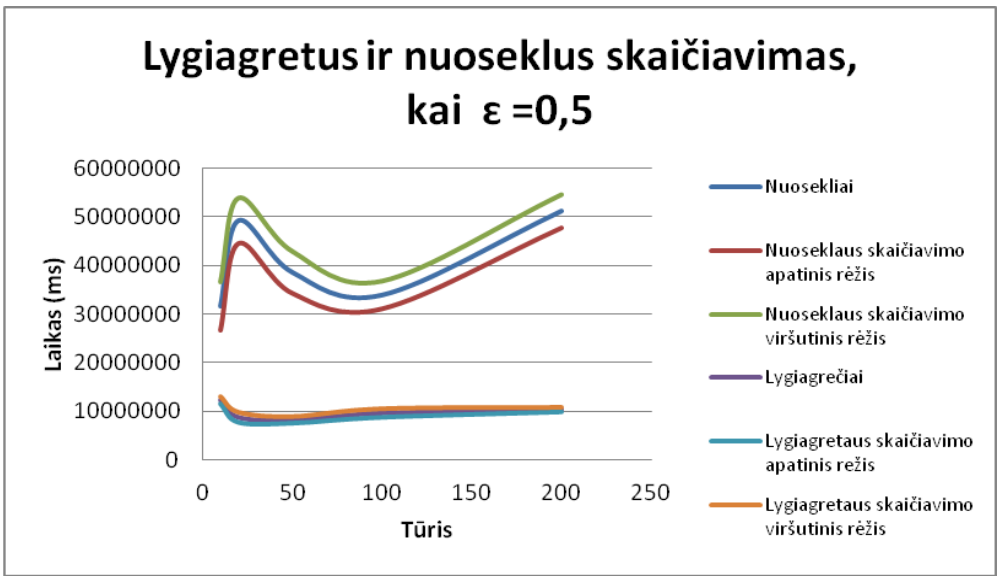
22 pav. Lygiagrečių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=1

Tūris	Spartinimo koeficientas	Efektyvumo koeficientas
10	8,57	2,14
20	4,62	1,16
50	6,56	1,64
100	4,97	1,24
200	8,46	2,12

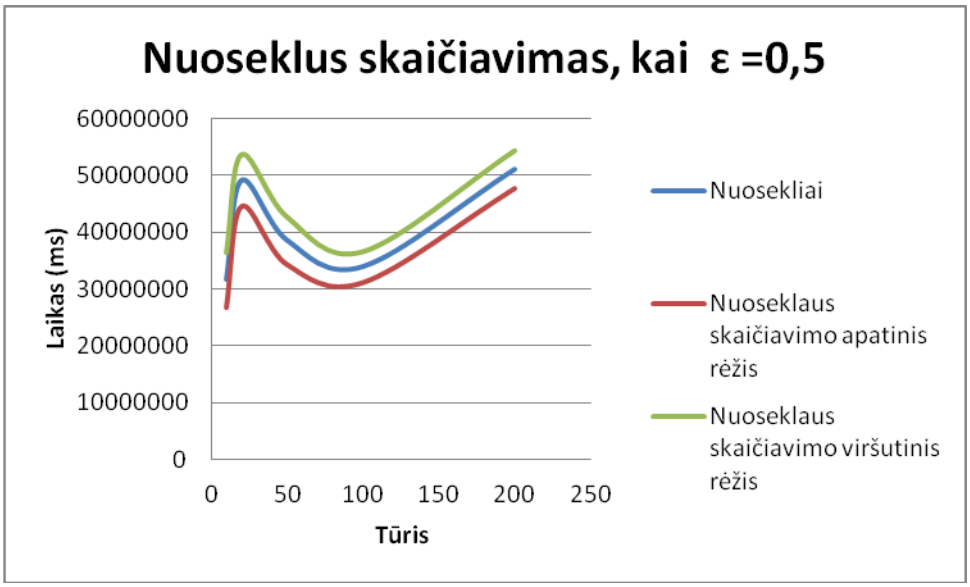
5 lentelė. Lygiagrečios programos spartinimo (2.1) ir efektyvumo (2.2) koeficientai, kai ε=1.

Skaičiavimo rezultatai (Priedas nr. 7), kai ε=0,5 pateikiami 23 pav. Skaičiuojant nuosekliai (24 pav.) išlieka ta pati tendencija kaip ir prieš tai pateiktuose rezultatuose – skaičiavimo laikas didėja didinant tūrį.

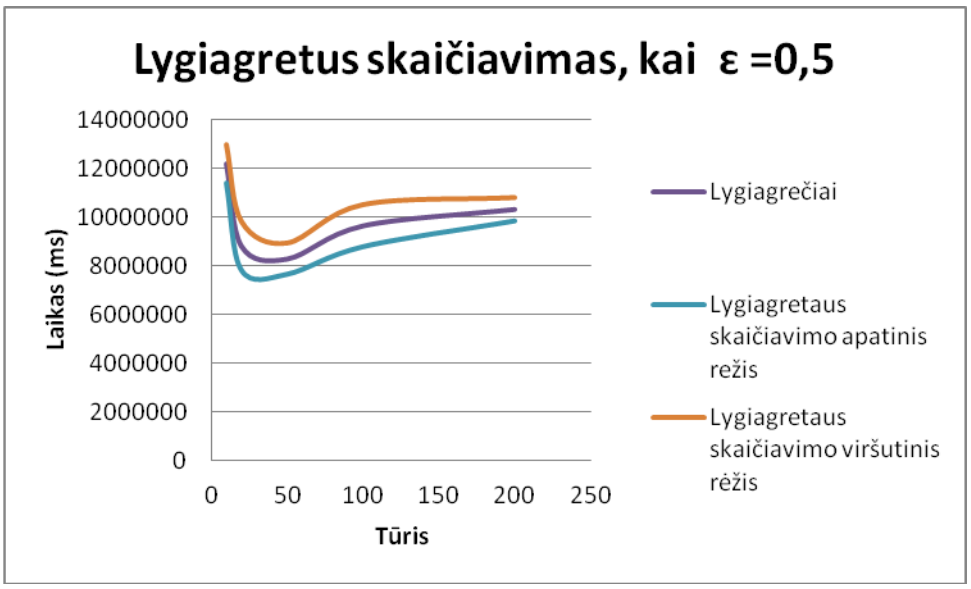
Tačiau šis eksperimentas parodo kitas tūrio reikšmes, prie kurių gaunamas geriausias tikslo funkcijos radimo laikas, t. y. tūris lygus 10 ir 100, o skaičiuojant lygiagrečiai (25 pav.) geriausias rezultatas pasiekiamas, kai skaičiuojama su tūriu 50. Skaičiuojant su tūriu 10 gaunamas dar prastesnis rezultatais nei su tūriu 200, tačiau galima pastebėti laiko didėjimo tendenciją, kai tūris didesnis nei 50. Spartinimo ir efektyvumo koeficientai pagal gautus rezultatus geriausi, kai tūris 20 (6 lentelė).



23 pav. Lygiagrečių ir nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=0,5



24 pav. Nuoseklių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=0,5



25 pav. Lygiagrečių skaičiavimų vidutinis laikas su pasikliautinaisiais intervalais (1.1), kur reikšmingumo lygmuo 0,95 ir ε=0,5

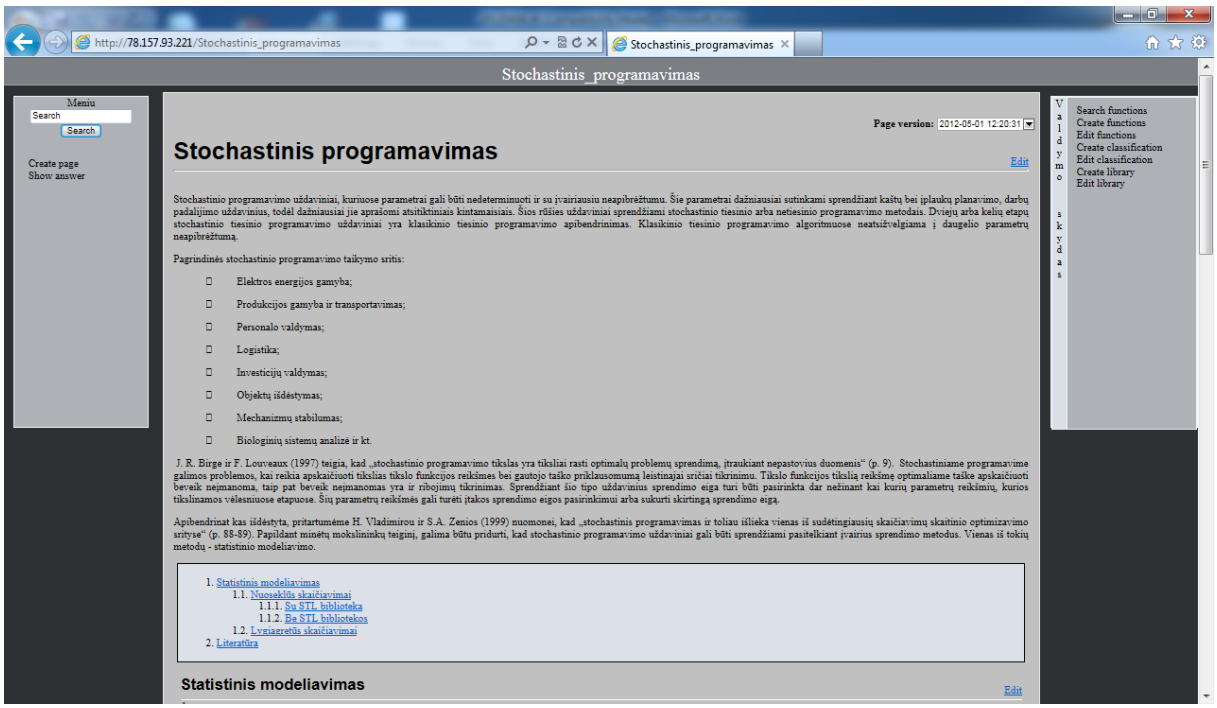
Tūris	Spartinimo koeficientas	Efektyvumo koeficientas
10	2,59	0,65
20	5,58	1,39
50	4,65	1,16
100	3,51	0,88
200	4,95	1,24

6 lentelė. Lygiagrečios programos spartinimo (2.1) ir efektyvumo (2.2) koeficientai, kai ε=0,5

Apibendrinant tyrimo duomenis, pravartu paminėti, kad Monte Karlo metodą realizuojančios programos t. y. nuoseklioji programa be STL bibliotekos, nuoseklioji programa su STL biblioteka ir lygiagrečioji programa yra patalpintos vikinomikos modelio pagrindu realizuotoje internetinėje matematinio programavimo ir modeliavimo sistemoje[7], kurioje buvo sukurtas naujas puslapis tema – Stochastinis programavimas[8] (26 pav.), kartu pateikiami ir šios temos potemės.

Internetinėje matematinio programavimo ir modeliavimo sistemoje pateikiami anksčiau minėtų programų išeities kodai. Joje taip pat galima ir atlikti išeities kodų tobulinimo darbus, sukompiliuoti programinį kodą ir atlikti skaičiavimus su sukompiliuota programa.

[7] Internetinė matematinio programavimo ir modeliavimo sistema. Prieiga per internetą: <http://78.157.93.221/>
[8] Stochastinis programavimas. Prieiga per internetą: <http://78.157.93.221/Stochastinis_programavimas>



26 pav. Stochastinio programavimo puslapis internetinėje matematinio programavimo ir modeliavimo sistemoje

Internetinėje matematinio programavimo ir modeliavimo sistemoje buvo sėkmingai atlikti bandomieji kompiliavimo ir programų vykdymo veiksmai. Būtina paminėti, kad šio darbo sukurtų programų eksperimentai buvo atlikti ne su minėta sistema, bet panaudojus kitą serverį. Tai atlikta todėl, kad internetinė matematinio programavimo ir modeliavimo sistema tuo metu pilnai nefunkcionavo bei panaudoto serverio techniniai parametrai pranašesni, nei kuriamos minėtos sistemos serverio parametrai.

5. Patarimai, pastebėjimai, rekomendacijos

Atliekant skaičiavimus su išlygiagretinta programa patariama atlikti keletą eksperimentų ir pasirinkti optimalų tūrį, kurį naudojant gaunamas trumpiausias funkcijos reikšmės radimo laikas. Pasinaudojus gautais eksperimento rezultatais, galima teigti, jog didinant tikslumą, t. y. mažinant ε reikšmę, reikia pasirinkti kuo didesnę tūrio reikšmę.

Siekiant greičiau rasti tikslo funkcijos reikšmę laiko atžvilgiu patartina pasirinkti kompiuterį, su kuo daugiau procesorių ar branduolių, nes išlygiagretinta programa juos išnaudoja maksimaliai ir greičiau atlieka lygiagrečius skaičiavimus.

Atliekant programavimo darbus pastebėta, jog kompiliuojant su skirtingomis QT versijomis, pateikiami skirtingi perspėjimai tam pačiam realizaciniam kodui taip pat pateikiamas perspėjimas „QWaitCondition: Destroyed while threads are still waiting“ po atliktų skaičiavimų. Atlikus analizę forumuose ir straipsniuose dėl šio perspėjimo buvo

priimta išvada, kad šis klaidos pranešimas pateikiamas, nes bibliotekoje QtConcurrent yra klaida. Tačiau ši klaida neturi įtakos programos skaičiavimui ir jos rezultatams.

Lygiagretinant nuosekliąją programą ir siekiant identifikuoti globaliuosius bei lokaliuosius kintamuosius, ir nustatant tarpusavio ryšius, būtina atlikti išsamią išeities kodo arba dokumentacijos analizę.

IŠVADOS IR REZULTATAI

1. Išanalizavus vikinomikos ir atviro kodo sampratą, galima teigti, kad spietinių programinių sistemų kūrimui (išlaikant vikinomikos modelio idėjas) tikslinga naudoti laisvas ir atviro kodo programas;
2. Atlikta stochastinio programavimo sampratos analizė parodė, kad stochastinio programavimo uždaviniuose naudojami parametrai yra modeliuojami atsitiktiniais dydžiais, o programinė įranga turi realizuoti optimizavimo ir statistinio modeliavimo algoritmus;
3. Išanalizavus lygiagrečių skaičiavimų atliktų eksperimentų rezultatus, darytina išvada, kad tikslo funkcijos radimo spartai turi įtakos tinkamai pasirinktas lygiagrečiai skaičiuojamų scenarijų kiekis;
4. Pasinaudojus atliktų eksperimentų rezultatais su uždaviniu, kuriame yra 20 pirmo etapo kintamųjų ir 20 antro etapo ribojimų, nustatyta, kad nuosekliame skaičiavime scenarijų kiekio didėjimas daro didelę įtaką programos veikimo spartai (9 pav., 12 pav., 15 pav.);
5. Įvykdžius eksperimentus su uždaviniu, kuriame yra 20 pirmo etapo kintamųjų ir 20 antro etapo ribojimų, prieita prie išvados, kad didinant tikslo funkcijos tikslumo reikšmę ε lygiagrečios programos efektyvumas didėjo, t. y. pasiektas didesnis 4 branduolių išnaudojimas, kai ε=2 efektyvumo koeficientas 15,77 (1 lentelė), ε=1 efektyvumo koeficientas 17,61 (2 lentelė), ε =0,5 efektyvumo koeficientas 24,87 (3 lentelė). Tačiau atlikus eksperimentus su uždaviniu, kuriame yra 60 pirmo etapo kintamųjų ir 60 antro etapo ribojimų nustatyta, kad didinant tikslo funkcijos tikslumo reikšmę ε lygiagrečios programos efektyvumas mažėjo, t. y. pasiektas mažesnis 4 branduolių išnaudojimas, kai ε=2 efektyvumo koeficientas 4,87 (4 lentelė), ε=1 efektyvumo koeficientas 2,14 (5 lentelė), ε=0,5 efektyvumo koeficientas 1,39 (6 lentelė).

LITERATŪRA IR INFORMACINIAI ŠALTINIAI

1. Atviras kodas Lietuvai. Prieiga per internetą: <http://www.akl.lt/ak>. Žiūrėta: 2010.11.05.
2. Baravykaitė, M. M. (2006). Lygiagrečiųjų algoritmų šablonų tyrimas ir kūrimas. Daktaro disertacija. Vilnius. 84 p.
3. Barney, B. OpenMP. Prieiga per internetą: <https://computing.llnl.gov/tutorials/openMP/#Introduction>; žiūrėta: 2011.03.15.
4. Baumann, W. Writing Message-Passing Parallel Programs with MPI. Paskaitų konspektas. Prieiga per internetą: <http://www.zib.de/zibdoc/mpikurs/mpi-course.pdf>. Žiūrėta: 2011.04.01.
5. Belevičius, R. (2012). Programavimas C++. Mokomoji knyga. Vilnius: Technika. 222 p.
6. Birge, J. R., Louveau, F. (1997). Introduction to Stochastic Programming. Springer. 421 p.
7. Concurrent Programming. Prieiga per internetą: <http://qt-project.org/doc/qt-4.8/threads-qtconcurrent.html>. Žiūrėta: 2011.02.01
8. Čiegis, R. (2001). Lygiagretieji algoritmai. Vilnius: Technika. 226 p.
9. Čiegis, R. (2005). Lygiagretieji algoritmai ir tinklinės technologijos. Vilnius: Technika. 320 p.
10. D. Tapscott, A. D. Williams (2006). Wikinomics. JAV. 324 p.
11. Derived Datatypes in MPI. Prieiga per internetą: <http://beige.ucs.indiana.edu/I590/node100.html>. Žiūrėta: 2011.04.02.
12. Ivanikovas S. (2009). Lygiagrečių skaičiavimų taikymo daugiamačiams duomenims vizualizuoti problemos. Daktaro disertacija. Vilnius. 112 p.
13. Juozapavičius A., Lapienis S., Tamulienė J. Paskirstytieji ir lygiagretūs skaičiavimai jau Lietuvoje. Prieiga per internetą: <http://www.elektronika.lt/articles/computers/4192/>. Žiūrėta: 2010.11.05.
14. Kerr, K. Visual C++ 2010 And The Parallel Patterns Library. MSDN Magazine, 2009, February. Prieiga per internetą: <http://msdn.microsoft.com/en-us/magazine/dd434652.aspx>. Žiūrėta: 2011.02.01.
15. Lygiagretusis programavimas. Prieiga per internetą: <http://lt.wikipedia.org/wiki/Lygiagretusis_programavimas>. Žiūrėta: 2010.11.05
16. Olsson U., Engstrand U., Rupšys P. (2007). Statistiniai metodai. SAS ir MINITAB.

Mokomoji knyga. Akademija. 138 p. Prieiga per internetą: <www.lzuu.lt/file.doc?id=33908>. Žiūrėta: 2012.04.20.

17. Parallel Patterns Library (PPL). Prieiga per internetą: <http://msdn.microsoft.com/en-us/library/dd492418.aspx>. Žiūrėta: 2011.02.03.
18. Paulavičius R. (2010). Globalusis optimizavimas su simpleksiniais posryčiais. Daktaro disertacija. Vilnius. 134 p.
19. Sakalauskas, L. (2009). Statistininis modeliavimas ir analizė. Paskaitų konspektas. 100 p.
20. Sapeiga A. (2006). GRID technologija mokymo procese: duomenų vizualizavimas. Šiaulių Universitetas. 56 p.
21. Skaičiavimų tinklai. Prieiga per internetą: <http://grid.mif.vu.lt/>. Žiūrėta: 2010.11.05.
22. The OpenMP® API specification for parallel programming. Prieiga per internetą: <http://openmp.org/wp/>; žiūrėta: 2011.03.15.
23. Vladimirou, H., Zenios, S. A. Scalable parallel computations for large-scale stochastic programming // Annals of Operations Research 90. 1999. p. 87-129
24. Žilinskas, A. (2005). Matematinis programavimas.Prieiga per internetą: <http://www.mii.lt/antanas/uploads/MathematicalProgramming.pdf>; žiūrėta: 2011.04.17.
25. Žilinskas, K. (2007). Matematinis programavimas. I dalis. Tiesinis programavimas. VŠĮ Šiaulių universiteto leidykla. 303 p.
26. Žilinskas, K. (2007). Stochastinio tiesinio programavimo Monte Karlo metodu tyrimas. Daktaro disertacija. Vilnius. 104 p.

Neimantas M. Internetinės matematinio programavimo ir modeliavimo sistemos / Stochastinio programavimo programų paketo kūrimo ir analizės magistro baigiamasis darbas. Vadovas prof. habil. dr. L. Sakalauskas. – Šiaulių universitetas, Matematikos ir informatikos fakultetas, 2012. – 54 p.

ANOTACIJA

Magistro baigiamajame darbe, remiantis spietinėmis technologijomis, sukurtas stochastinio programavimo lygiagrečių programų paketas ir ištirtas jo lygiagretinimo efektyvumas. Pirmoje darbo dalyje teoriniu aspektu pateikiama vikinomikos, stochastinio programavimo, statistinio modeliavimo, laisvų programų, atviro kodo, lygiagrečiųjų ir paskirstytųjų skaičiavimų samprata ir šių sąvokų taikymo sritys bei ypatumai. Antroje, projektinėje dalyje analizuoti esami sprendimai lygiagretinimo sferoje, pateiktas sukurtų programų eksperimentinio tyrimo vykdymo planas. Trečioje darbo dalyje aprašyta darbo eiga, pateiktas darbo eigos grafas, identifikuotos darbo metu kilusios problemos, aprašyti jų sprendimai, pristatytas galutinis projekto būvis ir atlikto eksperimento rezultatai bei rekomendacijos.

Pagrindiniai žodžiai: vikinomika, atviras kodas, stochastinis programavimas, statistinis modeliavimas, lygiagretieji ir paskirstytieji skaičiavimai.

Neimantas M. Master thesis on the Online Mathematical Programming and Simulation System / The Development and Analysis of Stochastic Programming Program Package. Tutor: Prof. habil. dr. L. Sakalauskas. – Šiauliai University, Faculty of Mathematics and Informatics, 2012. – 54 p.

ABSTRACT

In the Master thesis the package of parallel programmes of stochastic programming is created and the parallelism effectiveness is examined according to wikinomic technologies. The first part of the thesis in a theoretical aspect presents the concepts of the wikinomics, staochastic programming, statistical designing, free programmes, open source, parallel and distributed computing, also the application spheres of these terms and peculiarities. The second, projective part of the work, analyses the existing solutions in the parallelism sphere, presents the implementation plan of the experimental research of created programmes. In the third part of the thesis the process of the work, the graph of the working process are stated, the problems arising within the working process are identified, solutions of the problems are described, the final state of the project, results and recommendations of the performed experiment are introduced.

Key words: wikinomic, open source, stochastic programming, statistical designing, parallel and distributed computing.

PRIEDAI

1 priedas. Skaičiavimo rezultatas su testiniais duomenimis 20x20, kai ε =2 ir tūris 200

Matrix W

0.000 0.000 0.000 0.000 0.000 0.000 1.500 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 -1.500 0.000 0.000 0.000 0.000
-9.600 -9.100 8.600 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 2.100
0.000 0.000 0.000 0.000 0.000 8.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 2.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 -2.000 0.000 0.000
0.000 1.950 0.000 3.150 0.000 0.000 0.000 0.000 3.400 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 -8.500 0.000 0.000 0.000
7.900 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -
8.800 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.900 0.000
0.000 7.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 -4.800 -2.200 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 2.500 0.000
0.000 0.000 0.000 0.000 0.000 -2.500 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -5.050 0.000 0.000 0.000 1.800 0.000 0.000 0.900 0.000 0.000
0.000 0.000 0.000 0.000 1.500 0.000 0.000 0.000 0.000 0.850
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -5.100 0.000 0.000 0.000
0.000 -0.300 0.000 5.400 0.000 0.000 0.000 0.000 0.000 0.000
0.000 -6.700 0.000 0.000 0.000 0.000 -4.700 0.000 0.000 0.000 0.000 3.600 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 7.800 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 -1.600 -6.400 0.000 0.000 0.000 8.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 -5.800 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 6.000 -0.200 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 8.300 0.000 0.000 -8.300 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -2.000 0.000 0.000 0.000 0.000 0.000 2.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 -1.900 0.000 0.000 0.000 2.600 -8.400 0.000 0.000 0.000 0.000 0.000 0.000 0.000 5.800 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
3.400 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 -4.300 0.000 0.000 0.900 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 3.950 4.400 0.000
0.000 0.000 0.000 0.000 0.000 0.000 -8.350 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.500 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 -1.500 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 1.400 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -
1.900 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -0.500
0.000 0.000 0.000 0.000 0.000 0.000 0.000 -3.900 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -4.400 0.000 0.000
0.000 8.300 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

Matriix T

0.000 0.000 0.000 0.000 6.200 -1.000 0.000 0.000 0.000 6.600 0.000 1.000 0.000 0.000 -3.300 0.000 -4.400 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 -0.300 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 6.200 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 -9.200 -5.800 -7.500 -7.800 0.000 0.000 2.200 -0.600 0.000 0.000 0.000 0.000 0.000 0.800 0.800 0.000 1.300 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.100 0.000 8.500 -0.400 0.000 -3.600 9.100 0.000 0.000 0.000 0.000 0.000 3.600 0.000 1.300 -8.500 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 -5.900 0.000 -4.600 1.700 0.000 0.000 0.000 -6.000 -6.700 0.000 9.300 0.000 0.000 -8.300 0.000 0.000 0.000 0.000 8.900
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 2.000 0.000 -5.300 2.400 -5.200 5.300 0.000 0.000 0.000 -4.700 -4.600 8.500 -1.800 0.000 -9.200 1.700 0.000 6.700
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.800 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 -3.000 0.000 0.000 9.000 0.000 -5.900 -0.500 -6.300 0.000 9.700 -8.300 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 8.200 -2.800 0.000 0.000 0.000 2.100 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 3.600 0.000 0.000 -4.600 -8.700 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.500 0.000 5.900 0.000 0.000 0.000 1.100 0.000 6.800 0.000 0.000 0.000 0.000 1.400 1.500 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.500 0.000 7.100 0.000 -0.300 0.000 -0.700 0.000 6.800 0.600 0.000 9.700 0.000 0.000 -5.500 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.300 -5.500 0.000 0.600 0.000 -7.300 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -7.400 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
-5.600 0.300 7.000 0.000 -4.100 0.000 -0.900 0.000 0.000 0.000 0.000 0.000 -4.800 -4.200 -0.900 -1.100 -8.900 -6.600 -3.100 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 -3.200 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 3.400 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

0.000 -5.000 0.000 0.000 0.000 -6.300 0.000 0.000 -1.100 -1.500 0.000 0.000 0.000 0.000 0.000 0.000 0.000 4.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 5.100 0.000 3.900 0.000 6.000 0.000 0.000 -1.300 -7.700 0.000 0.000 0.000 -7.600 0.000 -3.300 -5.700 8.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 8.800 2.400 0.000 0.000 -0.700 0.000 0.000 0.000 0.000 0.000 0.000 0.000 4.300 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 -2.000 0.000 0.000 6.300 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 1.400 0.000 -5.300 -2.700 2.700 -0.800 0.000 0.000 0.000 0.000 -6.700 0.000 -3.600 0.000 -4.200 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
Vector Q
6.20 9.63 8.78 7.06 7.19 2.66 6.79 8.19 8.13 6.82 8.81 3.74 1.79 1.25 8.04 6.38 3.85 1.06 7.69 9.35 0.28 7.65
1.49 8.50 1.80 7.73 3.56 6.67 1.22 8.12
Vector Mean
-4.30 0.30 4.70 -3.80 3.40 2.30 -4.20 0.30 -3.10 -1.50 2.90 -2.40 4.70 0.30 -0.90 3.30 -0.70 2.30 4.40 0.30
Vector D2
1.29 0.09 1.41 1.14 1.02 0.69 1.26 0.09 0.93 0.45 0.87 0.72 1.41 0.09 0.27 0.99 0.21 0.69 1.32 0.09
Vector b
3.16 -1.09 1.30 -5.02 3.47 3.32 -3.25 1.15 -0.21 1.76
Matrix A
-3.30 -4.00 -2.30 0.00 0.00 0.00 -2.00 0.00 0.00 8.00 0.00 3.10 4.10 5.10 0.00 0.00 -5.80 0.00 6.80 0.00
0.00 0.00 0.00 -4.00 5.80 0.00 0.00 -3.70 0.00 0.00 0.00 1.20 0.00 0.00 0.00 0.00 0.00 -6.60 0.00 0.20
0.00 0.00 -3.40 0.00 0.00 0.00 -6.60 8.60 6.40 0.00 0.00 0.00 0.00 0.00 0.00 0.00 -3.90 2.90 9.90 0.00
0.00 1.80 3.20 0.00 7.30 0.00 0.00 -4.50 0.00 -1.70 6.70 -3.00 0.00 0.00 0.00 0.00 -3.70 -8.10 -5.80 -7.60
0.00 0.00 8.70 8.30 0.00 0.00 0.00 -7.10 8.70 0.00 0.00 0.00 -0.40 0.00 0.00 0.00 0.00 0.00 -3.00 2.90
0.00 0.00 0.00 0.00 0.00 0.00 6.90 -3.20 -9.10 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 8.20 8.50 9.80
0.00 -5.30 -3.90 -1.30 -1.60 0.00 1.60 0.00 -8.40 0.00 0.00 0.00 5.40 0.00 3.30 0.00 -9.60 0.00 0.00 0.00
0.00 0.00 0.00 7.40 8.70 0.00 -4.90 0.00 4.60 0.00 0.00 0.00 -3.80 0.00 0.00 0.00 0.00 -4.50 0.00 0.00
0.00 -3.50 0.30 4.20 -2.40 -9.10 0.00 -1.90 6.30 6.40 0.00 0.00 -6.30 0.00 0.00 0.00 -3.20 0.00 0.00 0.00
0.00 -8.00 0.00 8.80 -1.60 -1.20 -9.10 0.00 0.00 1.50 0.00 8.10 0.10 0.00 0.00 -7.90 4.90 -6.90 0.00 0.00
Vector C
-0.30 -0.90 1.40 2.70 2.40 -1.40 0.10 -2.90 1.70 1.70 0.10 0.40 -1.30 0.50 0.00 0.00 -1.10 -0.70 1.50 2.10
x[1]= 0.0000000000
x[2]= 0.0000000000
x[3]= 0.4817069669
x[4]= 0.1122166089
x[5]= 0.0336411404
x[6]= 0.0000000000
x[7]= 0.0000000000
x[8]= 0.3231870812
x[9]= 0.0058519940
x[10]= 0.1222918389
x[11]= 0.0000000000
x[12]= 0.2433421589
x[13]= 0.0000000000
x[14]= 0.6626114321
x[15]= 0.5240031214
x[16]= 0.3523720448
x[17]= 0.2970509337
x[18]= 0.0000000000
x[19]= 0.1292364311
x[20]= 0.3376471573
---
point:
0.0000 0.0000 0.4817 0.1122 0.0336 0.0000 0.0000 0.3232 0.0059 0.1223 0.0000 0.2433 0.0000 0.6626 0.5240
0.3524 0.2971 0.0000 0.1292 0.3376
---
gradient:
19.8769 57.3190 -19.1573 40.9049 117.8814 23.5570 5.8835 -39.5097 24.6111 134.8978 10.9218 33.1354 40.5056 -4.8231
-33.1055 -9.1957 -20.4070 6.0178 2.5915 -11.2839
---
projection of gradient:
0.0000 0.0000 21.4501 -3.5852 6.9236 0.0000 0.0000 19.5217 -7.3271 12.3582 0.0000 13.6237 0.0000 -17.1425 -
18.0826 5.2208 -9.1872 0.0000 -8.4740 6.9206

Nt=1 FF= 189.171+- 3.905 NN=200 Lsum=200 TT= 1.00 as= 23.613 Mcn=40
==========
---
point:
0.0000 0.0000 0.4710 0.1140 0.0302 0.0000 0.0000 0.3134 0.0095 0.1161 0.0000 0.2365 0.0000 0.6712 0.5330
0.3498 0.3016 0.0000 0.1335 0.3342
---
gradient:
19.8769 57.2958 -21.0281 40.5568 110.0676 28.0138 5.5890 -42.4702 26.2704 122.7902 11.8862 35.2551 37.5953 -5.6054
-26.7809 -8.9403 -12.7237 5.1909 1.5606 -11.9006
---
projection of gradient:
0.0000 0.0000 16.9438 -2.0993 5.0567 0.0000 0.0000 14.4705 -6.1867 9.5614 0.0000 12.5747 0.0000 -15.3758 -
13.3236 7.2469 -6.6085 0.0000 -5.3551 3.6250

```
---------------------------------
Nt=2 FF= 186.523+- 3.387 NN=200 Lsum=400 TT=  2.00 as= 21.908   Mcn=43
=========================================================================
---------------------------------
       point:
   0.0000   0.0000   0.4625   0.1151   0.0277   0.0000   0.0000   0.3062   0.0126   0.1113   0.0000   0.2302   0.0000   0.6789   0.5397
0.3461   0.3049   0.0000   0.1362   0.3324
---------------------------------
       gradient:
  19.8769  57.3245 -23.8636  40.9299 101.2962  37.1879   3.5361 -46.4641  27.8034 107.8369  13.6868  39.0063  32.1628  -4.4995
-19.6158  -9.2561  -6.5076   3.2938  -0.2744 -11.0288
---------------------------------
       projection of gradient:
   0.0000   0.0000  11.0819  -0.4358   2.8123   0.0000   0.0000   8.4521  -4.6179   5.8549   0.0000  10.9596   0.0000 -12.5875  -
9.2261   8.7863  -4.0425   0.0000  -2.1435   0.3310
---------------------------------
Nt=3 FF= 185.578+- 3.661 NN=200 Lsum=600 TT=  3.00 as= 22.276   Mcn=42
=========================================================================
---------------------------------
       point:
   0.0000   0.0000   0.4570   0.1153   0.0262   0.0000   0.0000   0.3020   0.0149   0.1084   0.0000   0.2248   0.0000   0.6852   0.5443
0.3417   0.3070   0.0000   0.1372   0.3322
---------------------------------
       gradient:
  19.8769  57.3711 -23.0744  41.5930  89.6871  40.2581   1.0862 -44.1748  27.7661  97.1812  14.0008  35.1715  31.3318  -0.8235  -
14.9824  -9.7685  -3.6103   3.7299  -0.5426  -8.1313
---------------------------------
       projection of gradient:
   0.0000   0.0000   7.7633  -0.1815   1.8946   0.0000   0.0000   5.7233  -3.2912   4.0937   0.0000   7.9076   0.0000  -9.0941  -5.9823
6.6731  -2.6217   0.0000  -1.2108  -0.1371
---------------------------------
Nt=4 FF= 182.745+- 3.361 NN=200 Lsum=800 TT=  4.00 as= 10.112   Mcn=93
=========================================================================
---------------------------------
       point:
   0.0000   0.0000   0.4531   0.1154   0.0253   0.0000   0.0000   0.2991   0.0166   0.1064   0.0000   0.2208   0.0000   0.6897   0.5473
0.3384   0.3083   0.0000   0.1378   0.3323
---------------------------------
       gradient:
  19.8769  57.3622 -26.4096  41.4303  74.0265  48.7527  -0.1870 -49.2576  31.8981  76.5748  15.8315  38.8090  25.5564  -1.0877
-3.2738  -9.6710   9.6533   2.0734  -2.4557  -8.3396
---------------------------------
       projection of gradient:
   0.0000   0.0000  -0.4283   2.3390  -1.4038   0.0000   0.0000  -3.2642  -1.1078  -0.9135   0.0000   5.4730   0.0000  -5.3947
2.7701   9.5763   2.0128   0.0000   4.1976  -5.7353
---------------------------------
Nt=5 FF= 181.143+- 2.670 NN=200 Lsum=1000 TT=  5.00 as= 11.593   Mcn=81
=========================================================================
---------------------------------
       point:
   0.0000   0.0000   0.4533   0.1142   0.0260   0.0000   0.0000   0.3007   0.0171   0.1068   0.0000   0.2181   0.0000   0.6924   0.5459
0.3336   0.3073   0.0000   0.1357   0.3351
---------------------------------
       gradient:
  19.8769  57.3711 -24.5355  41.5707  77.4749  46.5666  -0.3694 -46.1717  29.2336  81.7421  14.8064  36.2741  27.1237   0.5297
-7.8052  -9.7685   3.7399   3.1974  -1.3796  -7.0647
---------------------------------
       projection of gradient:
   0.0000   0.0000   2.1060   1.3135  -0.2300   0.0000   0.0000  -0.0712  -1.6779   0.6373   0.0000   5.8104   0.0000  -6.0091  -1.0634
7.6549   0.1075   0.0000   1.9122  -3.2398
---------------------------------
Nt=6 FF= 182.174+- 3.116 NN=200 Lsum=1200 TT=  6.00 as= 10.369   Mcn=90
=========================================================================
---------------------------------
       point:
   0.0000   0.0000   0.4522   0.1135   0.0261   0.0000   0.0000   0.3008   0.0180   0.1065   0.0000   0.2152   0.0000   0.6954   0.5465
0.3298   0.3072   0.0000   0.1348   0.3368
---------------------------------
       gradient:
  19.8769  57.3424 -24.4642  41.1684  79.5653  44.6252   0.8486 -46.4639  29.5042  84.1348  14.5234  36.4826  28.3134  -1.1446
-8.4435  -9.4527   4.5749   3.0522  -1.1214  -8.3844
---------------------------------
       projection of gradient:
   0.0000   0.0000   2.9754   1.3102  -0.0428   0.0000   0.0000   0.4518  -2.0268   1.1581   0.0000   6.5419   0.0000  -6.9833  -0.8904
8.4503   0.0883   0.0000   1.9744  -3.4470
---------------------------------
Nt=7 FF= 184.481+- 3.332 NN=200 Lsum=1400 TT=  7.00 as= 10.971   Mcn=85
=========================================================================
---------------------------------
```

```
      point:
   0.0000   0.0000   0.4508   0.1129   0.0261   0.0000   0.0000   0.3005   0.0190   0.1059   0.0000   0.2119   0.0000   0.6989   0.5469
0.3256   0.3072   0.0000   0.1338   0.3385
----------------------------------
      gradient:
  19.8769  57.3567 -25.2471  41.3618  78.6477  46.8135  -0.0289 -47.4518  30.0328  82.2056  15.0968  37.7270  26.5603  -0.4188
-7.3742  -9.6106   4.6487   2.7693  -1.6992  -7.8123
----------------------------------
      projection of gradient:
   0.0000   0.0000   1.9743   1.5422  -0.3892   0.0000   0.0000  -0.4609  -1.7449   0.5061   0.0000   6.2345   0.0000  -6.4197  -0.6662
8.5033   0.3517   0.0000   2.3450  -3.8046
----------------------------------
Nt=8 FF= 185.859+- 3.197 NN=200 Lsum=1600 TT=  8.00 as= 11.265   Mcn=83
================================================================================
----------------------------------
      point:
   0.0000   0.0000   0.4498   0.1121   0.0263   0.0000   0.0000   0.3008   0.0198   0.1057   0.0000   0.2088   0.0000   0.7021   0.5472
0.3213   0.3070   0.0000   0.1326   0.3404
----------------------------------
      gradient:
  19.8769  57.3642 -23.1563  41.4836  84.5502  44.3639  -1.2504 -43.4995  27.2223  90.6931  14.3394  34.7320  27.5682   2.5540 -
13.4761  -9.6927  -4.7989   4.0854  -0.9030  -5.4690
----------------------------------
      projection of gradient:
   0.0000   0.0000   5.6323  -0.0140   1.3463   0.0000   0.0000   4.2376  -2.5238   2.7712   0.0000   6.5165   0.0000  -7.1090  -6.2260
5.4044  -2.4425   0.0000  -1.0775  -0.0252
----------------------------------
Nt=9 FF= 182.969+- 3.236 NN=200 Lsum=1800 TT=  9.00 as=  8.791   Mcn=106
================================================================================
----------------------------------
      point:
   0.0000   0.0000   0.4470   0.1121   0.0257   0.0000   0.0000   0.2987   0.0211   0.1043   0.0000   0.2055   0.0000   0.7057   0.5503
0.3186   0.3082   0.0000   0.1332   0.3404
----------------------------------
      gradient:
  19.8769  57.3567 -24.6181  41.3555  66.9723  50.9917  -2.9283 -45.3686  30.3093  71.6798  15.2341  34.8312  23.1054   4.1458
-3.4006  -9.6106   6.0942   3.2545  -1.8419  -4.2143
----------------------------------
      projection of gradient:
   0.0000   0.0000  -1.4829   1.8175  -1.3281   0.0000   0.0000  -3.1733  -0.4118  -1.4321   0.0000   3.3756   0.0000  -2.9372
1.4942   6.3831   1.4517   0.0000   3.0854  -4.0947
----------------------------------
Nt=10 FF= 181.086+- 3.046 NN=200 Lsum=2000 TT= 10.00 as=  6.444   Mcn=145
================================================================================
----------------------------------
      point:
   0.0000   0.0000   0.4477   0.1112   0.0263   0.0000   0.0000   0.3002   0.0213   0.1050   0.0000   0.2038   0.0000   0.7071   0.5496
0.3154   0.3075   0.0000   0.1316   0.3425
----------------------------------
      gradient:
  19.8769  57.3909 -23.2895  41.8561  75.4009  46.7885  -2.2796 -43.4476  28.6046  82.8283  14.5030  33.7682  26.1488   4.0689
-9.1992  -9.9868  -0.5037   4.1653  -1.0396  -4.2749
----------------------------------
      projection of gradient:
   0.0000   0.0000   2.6275   0.5920   0.3051   0.0000   0.0000   1.2875  -1.5294   1.0683   0.0000   4.6987   0.0000  -4.8938  -2.8629
5.0944  -0.8443   0.0000   0.4400  -1.3814
----------------------------------
Nt=11 FF= 183.455+- 3.276 NN=200 Lsum=2200 TT= 11.00 as=  5.769   Mcn=162
================================================================================
----------------------------------
      point:
   0.0000   0.0000   0.4464   0.1109   0.0262   0.0000   0.0000   0.2996   0.0221   0.1045   0.0000   0.2015   0.0000   0.7096   0.5510
0.3129   0.3079   0.0000   0.1314   0.3431
----------------------------------
      gradient:
  19.8769  57.3300 -22.1970  41.0065  85.5615  43.2369  -1.9661 -41.5067  26.6725  91.8974  14.0867  32.5914  27.4623   3.9935 -
14.4210  -9.3165  -6.2181   4.9994  -0.6831  -4.3343
----------------------------------
      projection of gradient:
   0.0000   0.0000   6.5534  -0.5368   1.8391   0.0000   0.0000   5.4783  -2.6147   3.4560   0.0000   6.0330   0.0000  -6.8524  -6.8348
4.0369  -2.9578   0.0000  -1.9832   1.0810
----------------------------------
Nt=12 FF= 182.201+- 3.055 NN=200 Lsum=2400 TT= 12.00 as=  4.861   Mcn=192
================================================================================
----------------------------------
      point:
   0.0000   0.0000   0.4431   0.1112   0.0252   0.0000   0.0000   0.2969   0.0234   0.1027   0.0000   0.1985   0.0000   0.7130   0.5545
0.3109   0.3094   0.0000   0.1324   0.3426
----------------------------------
```

gradient:
19.8769 57.3172 -24.1059 40.7964 72.6837 51.4995 -4.3812 -43.8013 27.8982 75.2494 15.3848 34.3631 21.5876 6.2665 -7.2940 -9.1756 -1.0510 3.7548 -2.0478 -2.5427
---
projection of gradient:
0.0000 0.0000 0.3790 1.0111 -0.4071 0.0000 0.0000 -0.5777 -0.8566 -0.4051 0.0000 3.8643 0.0000 -3.4209 -2.9445 4.7927 -0.4857 0.0000 0.9945 -1.8467

Nt=13 FF= 180.305+- 2.865 NN=200 Lsum=2600 TT= 13.00 as= 11.685 Mcn=80
==========
---
point:
0.0000 0.0000 0.4429 0.1107 0.0255 0.0000 0.0000 0.2971 0.0238 0.1029 0.0000 0.1965 0.0000 0.7147 0.5559 0.3085 0.3096 0.0000 0.1319 0.3435
---
gradient:
19.8769 57.3424 -20.6116 41.1903 75.7489 42.8456 -2.7159 -38.7564 26.7151 87.0803 13.2091 28.9036 27.6684 6.3560 -12.7950 -9.4527 -5.4132 5.8713 0.2441 -2.4722
---
projection of gradient:
0.0000 0.0000 5.0805 -0.5727 1.5127 0.0000 0.0000 4.4430 -1.9397 2.7271 0.0000 4.2753 0.0000 -4.9140 -5.4433 2.4606 -2.4124 0.0000 -1.8111 1.2205

Nt=14 FF= 183.383+- 3.398 NN=200 Lsum=2800 TT= 14.00 as= 2.201 Mcn=423
==========
---
point:
0.0000 0.0000 0.4404 0.1110 0.0247 0.0000 0.0000 0.2949 0.0248 0.1016 0.0000 0.1944 0.0000 0.7172 0.5586 0.3072 0.3108 0.0000 0.1328 0.3429
---
gradient:
19.8769 57.3578 -22.6773 41.3842 66.4948 50.6469 -6.2513 -40.6348 28.1725 73.2091 15.3084 30.1461 22.2648 9.4842 -5.9367 -9.6228 -1.6190 5.0069 -1.9176 -0.0064
---
projection of gradient:
0.0000 0.0000 -0.1242 0.5060 -0.2657 0.0000 0.0000 -0.4457 -0.3114 -0.4119 0.0000 1.7299 0.0000 -1.3852 -1.8007 2.1142 -0.3203 0.0000 0.4196 -0.7986

Nt=15 FF= 180.482+- 1.755 NN=600 Lsum=3400 TT= 5.67 as= 7.694 Mcn=365
==========
---
point:
0.0000 0.0000 0.4404 0.1107 0.0248 0.0000 0.0000 0.2952 0.0249 0.1018 0.0000 0.1935 0.0000 0.7179 0.5595 0.3062 0.3110 0.0000 0.1326 0.3433
---
gradient:
19.8769 57.3369 -24.2280 41.0764 63.6833 51.7082 -3.8789 -44.2399 29.8031 67.6527 15.3469 33.2023 22.4003 5.5256 -2.1604 -9.3923 6.4693 3.7555 -1.9838 -3.1267
---
projection of gradient:
0.0000 0.0000 -2.4628 1.8497 -1.5710 0.0000 0.0000 -3.8982 -0.0044 -1.9580 0.0000 2.4202 0.0000 -1.8213 2.1793 5.5829 1.7648 0.0000 3.2386 -4.0859

Nt=16 FF= 182.726+- 2.233 NN=400 Lsum=3800 TT= 9.50 as= 9.722 Mcn=192
==========
---
point:
0.0000 0.0000 0.4417 0.1098 0.0256 0.0000 0.0000 0.2971 0.0249 0.1028 0.0000 0.1923 0.0000 0.7188 0.5585 0.3034 0.3101 0.0000 0.1310 0.3454
---
gradient:
19.8769 57.3711 -23.7892 41.5664 77.4959 50.0016 -4.8526 -43.0529 27.3656 81.4362 15.5771 34.0718 23.0073 6.8764 -9.9150 -9.7685 -5.1909 4.1396 -2.1803 -2.0619
---
projection of gradient:
0.0000 0.0000 2.4210 0.2842 0.4697 0.0000 0.0000 1.7849 -1.3455 0.8740 0.0000 4.2159 0.0000 -4.0912 -5.2079 3.6474 -1.7132 0.0000 -0.5242 -0.2119

Nt=17 FF= 181.921+- 2.895 NN=200 Lsum=4000 TT= 20.00 as= 8.668 Mcn=108
==========
---
point:
0.0000 0.0000 0.4405 0.1096 0.0254 0.0000 0.0000 0.2962 0.0256 0.1023 0.0000 0.1902 0.0000 0.7208 0.5611 0.3016 0.3110 0.0000 0.1312 0.3455
---
gradient:
19.8769 57.3766 -22.6915 41.6543 75.5157 48.3316 -4.6480 -41.3510 27.1573 82.0288 14.9389 31.9034 24.4120 7.3631 -10.2564 -9.8289 -5.1474 4.6353 -1.5104 -1.6783

```
      projection of gradient:
   0.0000   0.0000   2.8672   0.0035   0.7128   0.0000   0.0000   2.3521  -1.3536   1.2618   0.0000   3.7715   0.0000  -3.8352  -4.9437
2.8502  -1.7990   0.0000  -0.8922   0.2849
-----------------------------------
Nt=18 FF= 188.425+- 3.074 NN=200 Lsum=4200 TT= 21.00 as=  5.584   Mcn=167
===================================================================================
-----------------------------------
      point:
   0.0000   0.0000   0.4390   0.1096   0.0250   0.0000   0.0000   0.2950   0.0263   0.1017   0.0000   0.1883   0.0000   0.7227   0.5635
0.3001   0.3119   0.0000   0.1317   0.3453
-----------------------------------
      gradient:
  19.8769  57.3280 -21.6401  40.9720  73.4384  47.0988  -4.8165 -39.5496  26.9275  81.8431  14.2635  29.9465  24.0042   8.4246 -
10.8103  -9.2948  -5.8920   5.3940  -0.8693  -0.8416
-----------------------------------
      projection of gradient:
   0.0000   0.0000   3.1419  -0.2503   0.9143   0.0000   0.0000   2.8460  -1.3266   1.4949   0.0000   3.3792   0.0000  -3.5110  -5.2288
2.0246  -2.0315   0.0000  -1.3360   0.8563
-----------------------------------
Nt=19 FF= 184.237+- 3.294 NN=200 Lsum=4400 TT= 22.00 as=  3.460   Mcn=269
===================================================================================
-----------------------------------
      point:
   0.0000   0.0000   0.4375   0.1098   0.0246   0.0000   0.0000   0.2936   0.0270   0.1009   0.0000   0.1866   0.0000   0.7245   0.5661
0.2991   0.3129   0.0000   0.1323   0.3449
-----------------------------------
      gradient:
  19.8769  57.3340 -23.9757  41.0396  62.3112  51.4720  -4.3435 -43.5384  30.0206  66.9383  15.4727  32.1127  22.5660   6.1206
-1.5272  -9.3611   6.6752   3.8551  -2.1180  -2.6577
-----------------------------------
      projection of gradient:
   0.0000   0.0000  -2.6203   1.7571  -1.5661   0.0000   0.0000  -3.9574   0.1353  -1.9672   0.0000   1.8823   0.0000  -1.3341   2.7569
5.0198   1.9168   0.0000   3.2055  -3.9468
-----------------------------------
Nt=20 FF= 183.216+- 2.034 NN=400 Lsum=4800 TT= 12.00 as=  7.827   Mcn=239
===================================================================================
-----------------------------------
      point:
   0.0000   0.0000   0.4388   0.1089   0.0253   0.0000   0.0000   0.2956   0.0269   0.1019   0.0000   0.1857   0.0000   0.7252   0.5648
0.2966   0.3119   0.0000   0.1307   0.3469
-----------------------------------
      gradient:
  19.8769  57.3640 -23.0114  41.4710  61.9425  51.1328  -5.6646 -41.3993  29.1257  68.3303  15.4054  29.9607  22.9016   8.4104
-2.7807  -9.6903   3.3272   4.5922  -2.0108  -0.8528
-----------------------------------
      projection of gradient:
   0.0000   0.0000  -1.8824   1.1281  -1.0409   0.0000   0.0000  -2.6150   0.1540  -1.4142   0.0000   1.1306   0.0000  -0.6713   1.3424
3.0480   1.1122   0.0000   1.9637  -2.4141
-----------------------------------
Nt=21 FF= 181.026+- 2.105 NN=400 Lsum=5200 TT= 13.00 as=  3.209   Mcn=582
===================================================================================
-----------------------------------
      point:
   0.0000   0.0000   0.4397   0.1083   0.0259   0.0000   0.0000   0.2969   0.0268   0.1026   0.0000   0.1851   0.0000   0.7255   0.5641
0.2951   0.3114   0.0000   0.1298   0.3481
-----------------------------------
      gradient:
  19.8769  57.3810 -21.6135  41.7258  65.7718  47.7101  -6.4428 -38.7039  29.1907  77.3683  15.0446  27.6078  24.3156  10.1690
-6.4195  -9.8776  -2.3863   5.3976  -1.6147   0.5334
-----------------------------------
      projection of gradient:
   0.0000   0.0000   1.1870  -0.1022   0.3384   0.0000   0.0000   1.0142  -0.4755   0.6165   0.0000   1.1121   0.0000  -1.2427  -1.3774
0.7138  -0.5822   0.0000  -0.3953   0.2325
-----------------------------------
Nt=22 FF= 180.138+- 1.974 NN=400 Lsum=5600 TT= 14.00 as=  0.436
FINISH ! (  14.00)
 Sugaistas laikas (1234)
```

2 priedas. Skaičiavimo rezultatų vidutinės reikšmės su testiniais duomenimis 20x20, kai ε =2

Tūris	Vidutinė tikslo funkcijos reikšmė	Vidutinė paklaida	Vidutinė iteracijų suma	Vidutinis laikas	Nuoseklaus skaičiavimo apatinis rėžis	Nuoseklaus skaičiavimo viršutinis rėžis
Nuosekliai						
10	181,8268	1,98419	9440	187994,7	178237,7561	197751,6439
20	181,79291	1,97692	9602	95890,52	90706,09366	101074,9463
50	182,16153	1,94271	10005	43867,57	41733,89111	46001,24889
100	182,07423	1,89613	10546	25427,02	24041,80052	26812,23948
200	182,05292	1,75848	12392	20286,2	19428,39085	21144,00915
Lygiagrečiai					Lygiagretaus skaičiavimo apatinis rėžis	Lygiagretaus skaičiavimo viršutinis rėžis
10	181,8869	1,98645	9466	2980,54	2829,797959	3131,282041
20	182,03994	1,97777	9541	2100,28	1990,088365	2210,471635
50	181,82492	1,94189	9945	2451,83	2340,304432	2563,355568
100	181,99972	1,90181	9894	3145,89	2947,967563	3343,812437
200	181,93641	1,78572	11826	5307,5	4814,107009	5800,892991

3 priedas. Skaičiavimo rezultatų vidutinės reikšmės su testiniais duomenimis 20x20, kai ε =1

Tūris	Vidutinė tikslo funkcijos reikšmė	Vidutinė paklaida	Vidutinė iteracijų suma	Vidutinis laikas	Nuoseklaus skaičiavimo apatinis rėžis	Nuoseklaus skaičiavimo viršutinis rėžis
Nuosekliai						
10	182,3373	0,99827	33087,4	664577,2	625028,7124	704125,7076
20	182,3306	0,99672	35291,2	358164,4	339489,3506	376839,4694
50	182,258	0,992	35934	155996	147914,1804	164077,8796
100	182,3171	0,98661	37949	89114,01	83718,7516	94509,2684
200	182,5129	0,97442	39580	58246,25	55298,23458	61194,26542
Lygiagrečiai					Lygiagretaus skaičiavimo apatinis rėžis	Lygiagretaus skaičiavimo viršutinis rėžis
10	182,2858	0,9982	34767,6	9432,18	8968,403692	9895,956308
20	182,2899	0,99667	35651	8024,17	7646,993678	8401,346322
50	182,4332	0,99274	36029	7113,36	6751,348584	7475,371416
100	182,3399	0,98563	36788	8229,2	7685,815965	8772,584035
200	182,3081	0,97469	37256	8072,78	7479,689024	8665,870976

4 priedas. Skaičiavimo rezultatų vidutinės reikšmės su testiniais duomenimis 20x20, kai ε =0,5

Tūris	Vidutinė tikslo funkcijos reikšmė	Vidutinė paklaida	Vidutinė iteracijų suma	Vidutinis laikas	Nuoseklaus skaičiavimo apatinis rėžis	Nuoseklaus skaičiavimo viršutinis rėžis
Nuosekliai						
10	182,5151	0,49992	142868,3	2886537	2712667,221	3060407,739
20	182,4571	0,4997	145926,8	1477285	1404458,265	1550111,755
50	182,5138	0,49907	141501,5	616876,2	579566,5853	654185,7947
100	182,4658	0,49814	137309	323466,4	304594,0353	342338,8447
200	182,494	0,49663	142778	203086,1	191231,8742	214940,3858
Lygiagrečiai					Lygiagretaus skaičiavimo apatinis rėžis	Lygiagretaus skaičiavimo viršutinis rėžis
10	182,3886	0,49997	131243	29021,09	27301,11745	30741,06255
20	182,4374	0,49974	139875,2	28107,62	26551,69088	29663,54912
50	182,4925	0,49916	149086	26989,37	25596,52745	28382,21255
100	182,4797	0,49805	143639	26194,97	24678,12204	27711,81796
200	182,4331	0,49623	139708	26828,72	25036,82976	28620,61024

5 priedas. Skaičiavimo rezultatų vidutinės reikšmės su testiniais duomenimis 60x60, kai ε =2

Tūris	Vidutinė tikslo funkcijos reikšmė	Vidutinė paklaida	Vidutinė iteracijų suma	Vidutinis laikas	Nuoseklaus skaičiavimo apatinis rėžis	Nuoseklaus skaičiavimo viršutinis rėžis
Nuosekliai						
10	300,428	1,9962	89444	6319253	4820631,112	7817874,888
20	301,4978	1,9901	34320	2058425	1894505,411	2222343,589
50	301,5827	1,9742	111795	5215603	4447009,625	5984196,175
100	300,6117	1,9493	95580	4088622	3342670,908	4834572,092
200	301,5278	1,8937	40680	1659283	1522211,127	1796355,473
Lygiagrečiai					Lygiagretaus skaičiavimo apatinis rėžis	Lygiagretaus skaičiavimo viršutinis rėžis
10	301,1941	1,9936	27171	324468,1	299972,4648	348963,7352
20	301,1503	1,9897	29908	326184,6	302341,2318	350027,9682
50	301,4597	1,9845	30760	321035,9	295632,8597	346438,9403
100	301,7054	1,9405	31460	331676	308913,3935	354438,6065
200	301,7705	1,9236	39960	423180,6	391990,3318	454370,8682

6 priedas. Skaičiavimo rezultatų vidutinės reikšmės su testiniais duomenimis 60x60, kai ε =1

Tūris	Vidutinė tikslo funkcijos reikšmė	Vidutinė paklaida	Vidutinė iteracijų suma	Vidutinis laikas	Nuoseklaus skaičiavimo apatinis rėžis	Nuoseklaus skaičiavimo viršutinis rėžis
Nuosekliai						
10	300,7122	0,9992	368582	17312626	15938163,7	18687087,7
20	301,5135	0,9994	183558	11095795	9888508,339	12303082,06
50	301,2532	0,997	292160	13627570	11509007,31	15746131,89
100	300,7587	0,9945	245690	10488896	8874242,393	12103550,21
200	301,2103	0,9917	462640	18762623	17163333,24	20361912,76
Lygiagrečiai					Lygiagretaus skaičiavimo apatinis rėžis	Lygiagretaus skaičiavimo viršutinis rėžis
10	301,2204	0,9995	170408	2019752	1836264,056	2203240,344
20	301,4989	0,9983	221590	2399308	2232223,93	2566391,07
50	301,2325	0,9979	200440	2077585	1932326,127	2222842,873
100	301,4875	0,9954	204960	2111368	1933985,995	2288749,405
200	301,4385	0,9894	210920	2217333	2075203,607	2359463,193

7 priedas. Skaičiavimo rezultatų vidutinės reikšmės su testiniais duomenimis 60x60, kai ε =0,5

Tūris	Vidutinė tikslo funkcijos reikšmė	Vidutinė paklaida	Vidutinė iteracijų suma	Vidutinis laikas	Nuoseklaus skaičiavimo apatinis rėžis	Nuoseklaus skaičiavimo viršutinis rėžis
Nuosekliai						
10	301,3453	0,5	936848	31640906	26743386,54	36538424,46
20	301,3144	0,5	949998	49177093	44578592,21	53775594,59
50	301,3092	0,4997	816440	38579485	34341908,29	42817061,91
100	301,3538	0,4996	1188410	33929762	31152962,16	36706561,84
200	301,325	0,4982	1909040	51126125	47770178,65	54482071,75
Lygiagrečiai					Lygiagretaus skaičiavimo apatinis rėžis	Lygiagretaus skaičiavimo viršutinis rėžis
10	301,1446	0,5	1029461	12205005	11410156,09	12999853,31
20	301,2139	0,5	815530	8816431	7796328,142	9836533,058
50	301,3948	0,4995	803970	8298343	7634463,696	8962222,304
100	301,3303	0,4996	935180	9657519	8777323,632	10537714,57
200	301,4513	0,4985	989140	10335324	9843624,096	10827024,5