KAUNO TECHNOLOGIJOS UNIVERSITETAS
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS
MATEMATINĖS SISTEMOTYROS KATEDRA

Bronius Kučaitis

GRID tinklo klasterių scenarijaus medžio sudarymas naudojant atstatytus duomenis

Magistro darbas

Darbo vadovas
doc. dr. K.Šutienė

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS
MATEMATINĖS SISTEMOTYROS KATEDRA

Bronius Kučaitis

GRID tinklo klasterių scenarijaus medžio sudarymas naudojant atstatytus duomenis

Magistro darbas

Vadovas
doc.dr. K. Šutienė
2012-05-22

Recenzentas
doc. dr. Gytis Vilutis
2012-05-22

Atliko
FMMM-0 gr. stud.
Bronius Kučaitis
2012-05-22

Kaunas, 2012

KVALIFIKACINĖ KOMISIJA

Pirmininkas: Rimantas Rudzkis, profesorius (VU MII)

Sekretorius: Eimutis Valakevičius, docentas (KTU)

Nariai: Jonas Valantinas, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Zenonas Navickas, profesorius (KTU)

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

Kučaitis B. GRID tinklo klasterių scenarijaus medžio sudarymas naudojant atstatytus duomenis : Taikomosios matematikos magistro baigiamasis darbas / mokslinis vadovas doc. dr. K. Šutienė; Kauno technologijos universitetas, Fundamentaliųjų mokslų fakultetas, Matematinės sistemotyros katedra. – Kaunas, 2012. –48 p.

SANTRAUKA

Šio magistro darbo tikslas yra sukurti skaičiuojamųjų GRID tinklų klasterių darbo apkrovos scenarijų medį, kuris leistų spręsti stochastinio optimizavimo uždavinį ir optimizuoti klasterių veiklą. GRID tinklo klasterio scenarijų medžio generavimo metodika susideda iš duomenų atstatymo metodo, klasterio darbo apkrovos duomenų imitacinio generavimo ir klasterizavimo. Duomenų atstatymas buvo reikalingas, kadangi GRID tinklo klasterių darbo apkrovos stebėjimai buvo nepilni. Buvo analizuoti du duomenų atstatymo metodai. Buvo nustatyta, kad naudojant įvairų duomenų kiekį, duomenis geriau atstato didžiausio tikėtinumo tikėtinumo-maksimizavimo metodas. Klasterio darbo apkrovos duomenų trajektorijų imitaciniam modeliavimui buvo panaudotas laiko eilučių GARCH modelis. Šios duomenų trajektorijos buvo klasterizuojamos naudojant hierarchinio klasterizavimo metodą ir taip sudarytas kelių stadijų klasterių darbo apkrovos scenarijų medis.

Kucaitis B. Scenario tree generation for Grid network clusters employing imputed data : Master's work in applied mathematics / supervisor dr. assoc. prof. K. Sutiene; Department of Mathematics Research In System, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2012. – 48 p.

SUMMARY

The aim of this master's work is to generate a computational GRID clusters workload scenario tree, which would enable to solve the problem of stochastic optimization, and to optimize the working of clusters. GRID cluster scenario generation consists of the data recovery method, the cluster workload data simulation and data clustering. Data recovery was needed, since the grid cluster workload observations were incomplete. In this work two data reconstruction methods were analyzed. It was found that using different amounts of data, Maximum Likelihood Expectation-Maximization method is more efficient for the imputation of data. Time series GARCH model was used for simulation of cluster workload data paths. These data paths were clustered using a hierarchical clustering method and in this way GRID cluster workload multiple stage scenario tree was generated.

Turinys

Įvadas 9
1. Teorinė dalis 11
1.1. Analitinė dalis 11
1.1.1. Darbo reikšmingumas 11
1.1.2. GRID technologijos analizė 11
1.1.3. GRID tinklo apkrovos prognozavimo metodai 13
1.1.4. Scenarijaus medžio taikymas įvariose srityse 15
1.1.5. Metodikos analizuoti duomenims su nepilnais duomenimis 16
1.1.6. Analitinės dalies išvados 17
1.2. Metodologinė dalis 18
1.2.1. Didžiausio tikėtinumo duomenų atstatymo metodas 18
1.2.2. Daugkartinio užpildymo metodas 19
1.2.3. Markovo grandinių Monte Carlo metodas 21
1.2.4. Scenarijų medis 22
1.2.5. Scenarijų medžio generavimas klasterizavimo būdu 26
1.2.6. Klasterinė analizė 27
1.2.7. Hierarchinis klasterizavimas 28
1.2.8. Kophenetinis koreliacijos koeficientas 29
1.2.9. Silueto indeksas 30
1.2.10. Garch modelis 30
1.2.11. Prognozavimo paklaidos matai 31
2. Tiriamoji dalis 33
2.1. Duomenų atstatymas GRID tinklo klasteriams 33
2.2. Scenarijų medžio generavimas 41
Išvados 46
Literatūros sąrašas 47
Priedai 49
1 priedas. Scenarijų generavimo algoritmas 50
2 priedas. Duomenų atkūrimo SAS kodas naudojant didžiausio tikėtinumo metodą 56

Lentelių sąrašas

1.2.4 lentelė. Metodika naudojama scenarijų generavime....25
1.2.7 lentelė. Klasterių panašumo matai....29
2.1.1 lentelė. Klasterių duomenų trūkumo pavyzdys....33
2.1.2 lentelė. Duomenų trūkumo fragmentas....34
2.1.3 lentelė. Pradiniai duomenys....34
2.1.4 lentelė. Pradiniai duomenys po atsitiktinio duomenų pašalinimo....35
2.1.5 lentelė. Hipotezės apie pasiskirstymą pagal normalųjį dėsnį tikrinimo rezultatai. 35
2.1.6 lentelė. Duomenų atstatymo metodų paklaidos pradiniams duomenims....36
2.1.7 lentelė. Duomenų atstatymo metodų paklaidos naudojant 3 parų duomenis....37
2.1.8 lentelė. Duomenų atstatymo metodų paklaidos naudojant 5 parų duomenis....38
2.1.9 lentelė. Pagrindinių momentų vidutinė santykinė paklaida naudojant skirtingą duomenų kiekį....39
2.1.10 lentelė. Vidurkinė kvadratinė paklaida, naudojant skirtingą duomenų kiekį. 40
2.2.1 lentelė. Tikėtinumo santykio testas lyginant su GARCH(1,1) modeliu 42
2.2.2 lentelė. Parametrų nustatymas....42
2.2.3 lentelė. Naudojamo panašumo mato nustatymas naudojant kophenetinį koreliacijos koeficientą 43
2.2.4 lentelė. Realių ir prognozuotų reikšmių palyginimas 45

Paveikslų sąrašas

1.1.2.1 pav. GRID tinklo imitacija. 12
1.1.2.2 pav. Lygiagrečiųjų skaičiavimų pavyzdys 13
1.2.2 pav. Monotoninis (kairėje) ir nemonotoninis (dešinėje) trūkstamų duomenų išsidėstymas....20
1.2.4.1 pav. Scenarijų vėduoklės pavyzdys 23
1.2.4.2 pav. Scenarijų medžio pavyzdys....24
1.2.5 pav. Scenarijų medžio sudarymo schemos pavyzdys 27
2.1.1 pav. Klasterių „pupa.elen.ktu.lt“ (kairėje) ir „grid.fi.lt“ (dešinėje) duomenų histogramos....36
2.1.2 pav. Realių ir atstatytų duomenų palyginimas taikant MCMC ir EM....37
2.1.3 pav. Realių ir atstatytų duomenų palyginimas taikant MCMC ir EM, naudojant 3 parų duomenis ...38
2.1.4 pav. Realių ir atstatytų duomenų palyginimas taikant MCMC ir EM, naudojant 5 parų duomenis ...39
2.2.1 pav. Naudojamų duomenų grafikas. 41
2.2.2 pav. 10 sugeneruotų proceso atsitiktinių trajektorijų....43
2.2.3 pav. Silueto indekso reikšmės skirtingam klasterių skaičiui....44
2.2.4 pav. Pirmos stadijos dendograma 44
2.2.5 pav. Scenarijų medis pirmoms trims stadijoms. 45

ĮVADAS

Mokslinėje veikloje, versle ir kitur nuolat susiduriama su vienokiais ar kitokiais uždaviniais, kuriems spręsti naudojamas kompiuteris. Kuo sudėtingesnis uždavinys tuo daugiau reikia skaičiavimo pajėgumų, todėl ypatingai sudėtingiems uždaviniams spręsti pasitelkiami superkompiuteriai. Kadangi superkompiuterių kainos dažnai neįkandamos mažesnėms mokslo įstaigoms ar įmonėms, todėl kaip alternatyva buvo sukurta GRID technologija, kuri leidžia kompiuterius, bei serverius apjungti į vieną sistemą, kurios skaičiavimo pajėgumai prilyginami superkompiuteriams.

Terminas GRID (lietuviškai „tinklas“) buvo pradėtas vartoti apie 1990-tųjų metų vidurį, jis apibrėžė išplėstinę mokslo ir inžinerijos infrastruktūrą. GRID tinklas – tai lygiagrečiųjų ir paskirstytų skaičiavimų tinklas, kurį sudaro geografiškai nutolę kompiuteriniai skaičiavimų klasteriai (angl. *cluster*). Baltijos šalyse ši technologija pradėta naudoti 2005 metais, kai buvo įkurta BalticGrid organizacija.

Šiame darbe analizuojama kaip galima būt prognozuoti GRID tinklo klasterių darbo apkrovą ir sudaryti GRID tinklo klasterių scenarijų medį, kuris leistų optimizuoti klasterių veiklą sprendžiant stochastinio optimizavimo uždavinį. Norint prognozuoti veiklą susiduriama su duomenų trūkumo problema. Šis duomenų trūkumas atsirado dėl to kad neveikė klasteris arba neveikė klasterio veiklą fiksuojanti monitoringo sistema.

Dėl to šiame darbe galima išskirti du pagrindinius tikslus:

- Rasti tinkamą duomenų atstatymo metodą GRID tinklo klasteriams;
- Sudaryti GRID tinklo klasterių darbo apkrovos scenarijų medį.

Sprendžiant duomenų atstatymo problemą, buvo nagrinėti pagrindiniai nepilniems duomenims analizuoti taikomi metodai. Plačiau buvo analizuojami du duomenų atstatymo metodai, kurie praktikoje dažniausiai naudojami tokiam duomenų trūkumo tipui ir trūkstamų duomenų išsidėstymu kaip ir GRID tinklo klasterių, bei pateikia geriausius atstatymo rezultatus. Tai daugkartinio užpildymo Markovo grandinių Monte Carlo metodas ir didžiausio tikėtinumo metodas naudojant tikėtinumo-maksimizavimo (angl. Expectation-Maximization) algoritmą. Šių metodų palyginimui bei duomenų atstatymo kokybei įvertinti gauti rezultatai buvo palyginti su realiais duomenimis. Ganėtinai tikslūs ir geresni duomenų

atstatymo rezultatai buvo gaunami naudojant didžiausio tikėtinumo metodą, dėl to šis metodas ir buvo naudojamas klasterių duomenims atstatyti.

GRID tinklo klasterių scenarijaus sudarymui buvo reikalingas šios klasterio veiklos prognozavimas. Buvo išanalizuoti pagrindiniai klasterių darbo apkrovos prognozavimo metodai. Kadangi laiko eilučių metodas skirtingai nei dauguma metodų gali pateikti ilgalaikes klasterių darbo apkrovos prognozes ir taip pat ši metodika yra įdiegta į pagrindinius statistinius programinius paketus, ji buvo naudojama klasterių veiklai prognozuoti. Klasterių darbo apkrovai prognozuoti buvo naudojamas laiko eilučių GARCH modelis. Buvo imitaciniu būdu sumodeliuotos šio proceso duomenų trajektorijos. Tada scenarijų medžiui sugeneruoti, šios duomenų trajektorijos buvo klasterizuojamos. Klasterizavimui atlikti buvo naudojamas hierarchinis klasterizavimas, o nustatyti optimalų klasterių skaičių buvo naudojamas Silueto indeksas. Taip buvo sugeneruotas kelių stadijų scenarijų medis. Scenarijų medžio generavimo algoritmas buvo realizuotas Matlab programiniame pakete.

Dalis magistro darbų rezultatų buvo paskelbta X-oje studentų konferencijoje „Taikomoji matematika“ ir išspausdinta konferencijos leidinyje.

1. TEORINĖ DALIS

1.1. ANALITINĖ DALIS

Šioje dalyje bus apžvelgti GRID tinklai, aplikta GRID tinklų darbo apkrovos prognozavimo ir duomenų atstatymo metodų analizė.

1.1.1. DARBO REIKŠMINGUMAS

GRID lygiagrečiųjų skaičiavimų tinklai sparčiai populiarėja, kadangi tai alternatyva pakeisti superkompiuterius, kuria gali naudotis mažesnės mokslo organizacijos ar įmonės. Šiame darbe analizuojama kaip galima būt numatyti šių tinklo klasterių veiklos darbo apkrovą ir sudaryti šios apkrovos scenarijų medį. Tai leistų efektyviau paskirstyti resursus. Tuo galėtų naudotis tinklų administratoriai, kurie prižiūri tokius tinklus.

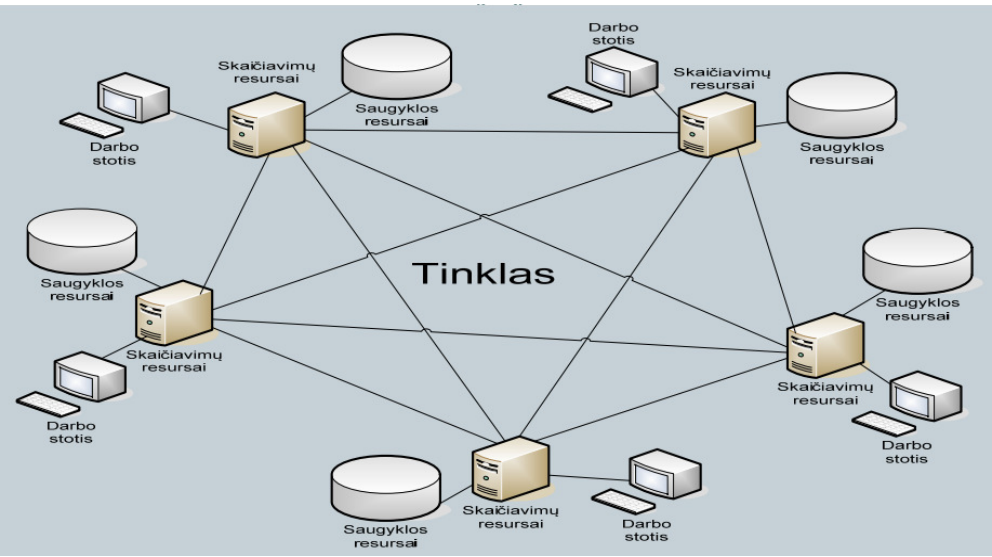
1.1.2. GRID TECHNOLOGIJOS ANALIZĖ

GRID technologija – vieni nuo kitų geografiškai nutolę kompiuterių telkiniai, apjungti į tinklą. Šie kompiuteriai apjungti spręsti uždaviniams reikalaujantiems didelių lygiagrečių skaičiavimo resursų. Tokiems uždaviniams spręsti naudojami superkompiuteriai, tačiau jų kaina yra labai didelė ir paprastos institucijos neišgali jų įsigyti. Todėl GRID skaičiavimų tinklai sukurti tam, kad pakeistų brangius didelių skaičiavimų įrenginius.

GRID tinklą sudaro klasteriai (angl. *cluster*). Klasterį gali sudaryti mums įprasti namų kompiuteriai su įprasta įranga (motininė plokštė, procesorius, atmintis, kietasis diskas, tinklo ar kita ryšio plokštė) arba specializuoti serveriai.

Vieną klasterį gali sudaryti 10, 20 ir daugiau kompiuterių, serverių skaičius neribotas. Visą GRID tinklą sudaro keletas ar kelios dešimtys tarpusavyje sujungtų klasterių. GRID skaičiavimų technologija

pagrįsta bendradarbiavimu, kiekviena organizacijoje dalyvaujanti institucija savo išteklius pateikia kitiems organizacijos nariams. Tokio tinklo iliustracija pavaizduota 1.1.2.1 paveiksle.

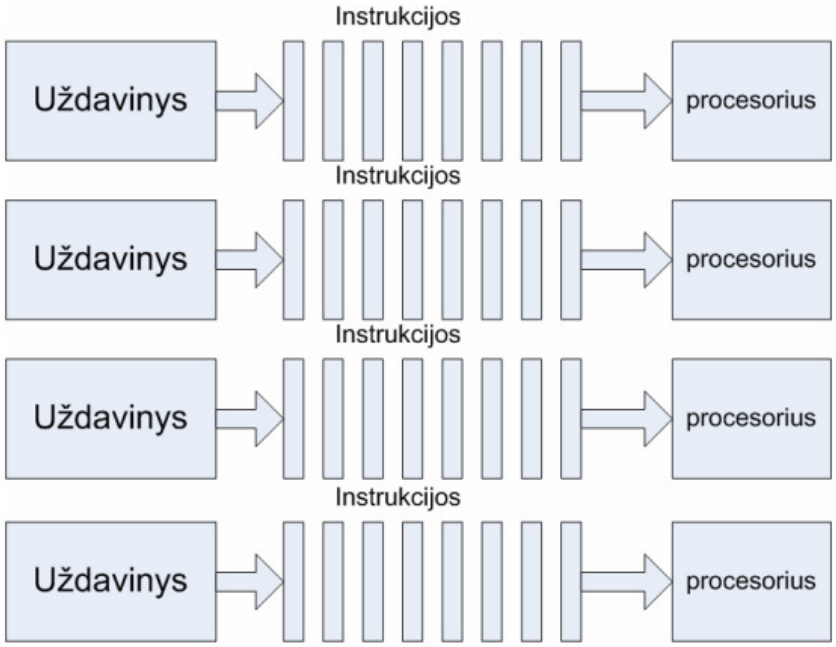


1.1.2.1 pav. GRID tinklo imitacija.

Kiekvienas GRID klasteris yra aprūpintas programine įrangą, kurią galima skaidyti į tokias dalis:

1. Operacinės sistemos programinę įrangą;
2. GRID apjungimo, valdymo programinę įrangą;
3. Papildomą programinę įranga (angl. *midleware*).

Realiame pasaulyje yra daug lygiagrečių procesų kaip pavyzdžiui planetų judėjimas, automobilių surinkimas, oro ir vandenynų judėjimas, tektoninių plokščių judėjimas, įvairių cheminių, fizikos dalelių judėjimas. Todėl lygiagretieji skaičiavimai yra neišvengiami. Žemiau 1.1.2.2 paveiksle pavaizduotas uždavinio sprendimas lygiagrečiuoju atveju, kada užduotis skaidoma į smulkesnes užduotis ir šie uždaviniai vykdomi vienu metu, lygiagrečiai.



1.1.2.2 pav. Lygiagrečiųjų skaičiavimų pavyzdys

Pagrindinės GRID naudojimo priežastys:

1) GRID tinklai kainuoja mažiau, nei superkompiuteriai, o skaičiuojamoji galia didelė, prilyginama superkompiuteriams. Ne visos mokslo organizacijos išgali nusipirkti superkompiuterius, todėl šie tinklai yra gera išeitis;
2) GRID tinklai leidžia atlikti sudėtingus uždavinius, o daugelis mokslinėje veikloje vykdomų uždavinių yra gana sudėtingi;
3) GRID tinklai konkurencingi – leidžia bendradarbiauti ir apjungti darbus virtualioje aplinkoje;
4) Įgalina naudoti nutolusius skaičiavimo resursus. Resursai gali būti apjungti į kitus tinklus ir taip padidinama skaičiavimo galia.

1.1.3. GRID TINKLO APKROVOS PROGNOZAVIMO METODAI

GRID tinkluose procesoriai gali prisijungti ir atsijungti bet kuriuo metu, kas daro tinklo apkrovą labai šokinėjančią (angl. *bursty*). Žemiau pateikti pagrindiniai metodai, kurie taikomi prognozuojant tinklo darboapkrovą.

1) Homeostatiniai (angl. *homeostatic*) prognozavimo metodai. Šita metodika buvo pasiūlyta Yang [12]. Abu šitie prognozavimo metodai yra vieno žingsnio į priekį, kurie naudoja skirtingą strategiją prognozuojant tinklų apkrovą. Homeostatinių prognozavimo metodai priima prielaidą, kad jei apkrovos dabartinė reikšmė yra didesnė (mažesnė) negu praeities reikšmių vidurkis, sekanti reikšmė yra linkusi sumažėti (padidėti). Pagrindas šito metodo yra tai, kad apkrova yra auto-korekcinė ir sugrįžtanti į praeities duomenų vidurkį. Kiekviename prognozavimo žingsnyje, augimo ar slopinimo reikšmė gali būti statinė visiems prognozavimo žingsniams arba dinaminė ir apskaičiuojama kiekviename žingsnyje, pagal tai homeostatiniai prognozavimo metodai skirstomi į 4 klases: nepriklausomi statiniai, nepriklausomi dinaminiai, santykinai statiniai ir santykinai dinaminiai.

2) Tendencija paremti prognozavimo metodai:
   a) Tendencija paremti prognozuoją sekančią reikšmę remiantis laiko eilučių kaitos tendencija. Šitas metodas remiasi prielaidą, kad jei esama reikšmė padidėja (sumažėja), sekanti reikšmė taip pat padidėja (sumažėja). Kaip ir homeostatiniame prognozavimo modeliuose, variacija (sumažėjimo ir padidėjimo reikšmė), gali būti nepriklausoma arba santykinai proporcinga esamai reikšmei [12].
   b) Taip pat tendenciją paremtą modelį, kur prognozavimas paremtas laiko eilučių atitikimu tam tikro aukštesnio laipsnio polinomui (angl. *polynomal fitting*) pasiūlė Zhang[27]. Pagrindinė šios ir praeitos metodikos trūkumas tas, kad neįmanoma apskaičiuoti kada laiko eilutė pakeis kryptį.

3) Tinklo orų servisas (angl. *Network Weather Service*) - tai specializuota programa, kur atlieka dinaminį duomenų stebėjimą ir atlieka vieno žingsnio į priekį prognozavimą, taikant įvairius metodus . Šie metodai yra tokie : slenkančio vidurkio (angl. *running average*), slankiojančio lango vidurko (angl. *sliding window average*) , paskutinio matavimo (angl. *last measurement*), adaptyvaus lango vidurkio (angl. *adaptive window average*), medianinio filtro (angl. *median filter*), adaptyvaus lango medianos, α- nupjautojo vidurkio (angl. α- trimmed mean), autoregresijos ir stochastinio gradiento metodai. Sistema fiksuoja visų prognozių tikslumą ir pateikia geriausią metodą. Pagrindiniai trūkumas yra tas, kad ši sistema turi būt įdiegta GRID tinklo programinėje įrangoje. [28]

4) Markovo grandinėmis paremtas prognozavimo modelis pasiūlytas Shi [24]. Jame prognozavimas paremtas esama būsena ir nepriklauso nuo praeities duomenų, todėl prognozavimas gali būti atliktas tik trumpalaikis.

5) Remiantis kryžminę koreliacija (angl. *cross-corelation*) tarp apkrovos ir kitų parametrų (kaip atminties būsena) [16]. Šiam metodai reikalinga, kad būtų fiksuojami ne tik apkrovos duomenys.

6) Laiko eilučių prognozavimo metodai. Išsami analizė naudojant tiesinius modelius atlikti apkrovos prognozavimą buvo atlikta Dinda[5]. Jis atliko metodus su autoregresijos (AR), slenkančio vidurkio (MA), autoregresijos ir slenkančio vidurkio metodą (ARMA) , autoregresinis integruotas slenkančio vidurkio metodas (ARIMA) ir kt. Atlikti skaičiavimai parodė, kad ir paprastu AR modeliu remiantis praeities duomenimis galima sėkmingai prognozuoti apkrovą. Dar vienas šio metodo privalumas toks, kad laiko eilučių modeliai yra daugumoje statistinių paketų, todėl jais patogu naudotis. Taip pat šie metodai, skirtingai nei dauguma anksčiau minėtų metodų, gali pateikti ilgalaikes prognozes.

1.1.4. SCENARIJAUS MEDŽIO TAIKYMAS ĮVARIOSE SRITYSE

Sprendimų priėmimo modeliuose esant neapibrėžtumams susiduriama su neapibrėžtumų reprezentacijos problema, juos reikia pateikti tokia forma, kuri tinka kiekybiniams modeliams. Jei neapibrėžtumai išreikšti per daugiamačius tolydžiuosius skirstinius ar diskretų skirstinį su be galo daug baigčių, stochastinis programavimas negali išspręsti tokios problemos. Vienas iš būdų spręsti tokią problemą yra diskretus aproksimavimas duoto pasiskirstymo. Šis aproksimavimas atliekamas per scenarijaus medžio generavimą. Vietoj to kad pateiktų įvertį daugiamačių atsitiktinių kintamųjų, scenarijaus medžio metodas pateikia tikėtinus ateities scenarijus su jų tikimybėmis.

Scenarijų medžio generavimas plačiai taikomas sprendimų priėmimo procese esant neapibrėžtumams taikomos finansuose kaip, pavyzdžiui, ekonominio pobūdžio problemoms tokioms kaip optimaliam lėšų paskirstymui [13], kapitalo rinkos modeliui generuoti [18], pensijų fondo lėšų įsipareigojimams[14,21], rizikos valdyme [20], taip aptarnaujančių grandinių planavimo problemose, transportavimo ir logistikos problemose, telekomunikacijose. Nors šis metodas sėkmingai pritaikytas daugelyje sričių, šis metodas kol kas ne taip plačiai taikomas modeliuoti įvairių apkrovų stochastiniams procesams, nors yra ir šitiems procesams scenarijų metodų taikymo pavyzdžių [9] [19].

1.1.5. METODIKOS ANALIZUOTI DUOMENIMS SU NEPILNAIS DUOMENIMIS

Kadangi GRID tinklo klasterių duomenys paprastai būna nepilni, dėl to kad neveikė klasteris ar jo monitoringo sistema, apžvelgsime pagrindines duomenų metodikas, kurios naudojamos dirbti su duomenimis, kurie yra duomenų trūkumas. Visi žemiau išvardinti metodai naudojami kai prielaida apie duomenų trūkumą yra MAR (angl. *Missing-At-Random*) pagal kurį trūkstama kintamojo reikšmė nepriklauso nuo kintamojo įgyjamos reikšmės. Pagrindinės metodikos nepilnų duomenų analizei yra tokios [2,8,23] :

1) Pilnų atvejų analizė.

Dauguma statistinių analizės programų pagal nutylėjimą naudoja pilnų atvejų analizę, taikant subalansuotos duomenų dalies naudojimą (angl. *listwise deletion*). Nagrinėjamai tik duomenų įrašai, kurie turi pilnus stebėjimus visiems priklausomiems ir nepriklausomiems kintamiesiems. Subalansuotos duomenų dalies naudojimas, kai yra gana daug trūkstamų duomenų gali vesti prie analizės, kai nagrinėjama mažas procentų duomenų dėl naudojamos redukcijos, netgi jei yra mažas kintamųjų skaičius. Tai sumažina statistinę galią - veda prie didesnių standartinių paklaidų ir platesnių pasikliautinų intervalų.

2) Esamų atvejų analizė.

Kadangi naudojant pilnų atvejų analizę prarandami esami duomenys, kaip alternatyva gali būti naudojamas kitas metodas – poromis subalansuotos duomenų dalies panaudojimas (angl. *pairwise deletion*). Šita metodika dažnai siūloma statistiniuose paketuose apskaičiuojant aprašomąją statistiką. Daugumai tiesinių modelių (pvz. tiesinei regresijai, faktorinei analizei) ieškomi parametrai gali būti išreiškiami kaip funkcijos nuo populiacijos vidurkiu, variacijų ir koreliacijos. Tada kiekvienas iš šių momentų yra apskaičiuojamas naudojant visus turimus duomenis. Tada šie gauti momentai yra įstatomi į populiacijos parametrų formules.

Jei duomenys yra MAR, poromis subalansuotos duomenų dalies panaudojimas gali pateikti iškreiptus parametrų įverčius. Taip pat sunku gauti tiksliais standartines paklaidas, nes kiekviena koreliacija tarp kintamųjų apskaičiuojama naudojant skirtingą duomenų skaičių, kuris priklauso nuo duomenų trūkumo išsidėstymo.

3) Nesąlyginis vidurkio pakeitimas.

Vienas iš paprasčiausių variantų yra pakeitimas trūkstamos reikšmės kintamojo vidurkiu. Tai yra prastas metodas, pirmiausia dėl to, kad gaunamas netikslus variacijos įvertinimas, priklausomai nuo duomenų trūkumo proporcijos su visais duomenimis. Taip pat koreliacija tarp kintamųjų gaunama iškreipta. Dėl to šis metodas netinkamas praktiniam naudojimui.

4) Sąlyginis vidurkio pakeitimas.

Taip pat galima naudoti trūkstamų duomenų užpildymus paremtais sąlygine duomenų informacija. Dažniausiai naudojamas duomenų užpildymas, remiantis tiesine regresija. Nors šis metodas žymiai pranašesnis už nesąlyginį duomenų užpildymą, tačiau ir šiuo metodu galima gauti iškreiptus parametrų įverčius, taip pat gaunamos netikslios standartinės paklaidos, dėl to statistikų reikšmės yra netinkamos naudoti hipotezėms. Taip pat šis metodas neatspindi neapibrėžtumo tinkamos trūkstamos reikšmės parinkime. Dėl to šis metodas, kaip ir prieš tai paminėti, nėra tinkami praktiniam naudojimui.

5) Didžiausio tikėtinumo metodu.

Didžiausio tikėtinumo metodu gauti parametrų įverčiai yra nuoseklūs ir efektyvūs pagal MAR sąlyga. Taip pat jei galioja ši prielaida apie trūkstamų duomenų tipą, kitų parametrų įverčiai, kaip pavyzdžiui standartinė paklaida, taip pat gaunami neiškreipti.

6) Daugkartinio užpildymo metodas.

Pakeičia trūkstamas reikšmes aibe tikėtinų reikšmių, kuri atspindi neapibrėžtumą tinkamos reikšmės pasirinkime. Tada statistinė analizė atliekama šiai užpildytų duomenų aibei ir gaunami rezultatai sujungiami. Pagal duomenų trūkumo prielaidą MAR, šiuo metodu taip pat gaunami efektyvūs ir neiškreipti parametrų įverčiai.

1.1.6. ANALITINĖS DALIES IŠVADOS

Analitinėje dalyje buvo aptartas GRID tinklų naudojimo privalumai, pagrindiniai GRID tinklo darbo apkrovos prognozavimo metodai, o kadangi GRID tinklo duomenys dėl techninių priežasčių gaunami nepilni, dėl to buvo aptartos ir pagrindinės duomenų atstatymo metodikos, bei pateiktas scenarijų medžio naudojimo pavyzdžiai spręsti optimizavimo problemomis su duomenų neapibrėžtumu. Kadangi laiko eilučių metodas gali pateikti ilgalaikes prognozes, skirtingai nei dauguma kitų GRID darbo apkrovos prognozavimo metodų, be to šis metodas yra daugumoje programinių paketų, todėl jį patogu naudoti. Duomenų atstatymui naudoti tinkamiausi metodai yra didžiausio tikėtinumo ir daugkartinio užpildymo duomenų atstatymo metodai, kadangi jie pateikia neiškreiptus ir efektyvius parametrų įverčius.

1.2. METODOLOGINĖ DALIS

Šioje dalyje pateikiama magistrinio baigiamojo darbo problemos sprendimui taikomų matematinių, statistinių ir ekonometrinių metodų aprašymas.

1.2.1. DIDŽIAUSIO TIKĖTINUMO DUOMENŲ ATSTATYMO METODAS

Didžiausio tikėtinumo metodas įrodė, kad gali būt puikus metodas apdorojant nepilnus duomenis daugybėje situacijų. Didžiausio tikėtinumo metodas trūkstamiems duomenims pateikia įverčius, kurie turi norimas savybes : nuoseklumą, asimptotinį efektyvumą ir asimptotinį normalumą. Nuoseklumo savybe norima pasakyti, kad parametrai bus apytikriai neiškreipti. Asimptotinis normalumas reiškia, kad parametrai bus apytikriai pilnai efektyvūs, pavyzdžiui turės mažiausias standartines paklaidas. asimptotinis normalumas yra svarbus, kadangi galima naudoti normalaus pasiskirstymo aproksimavimą skaičiuojant pasikliautinus intervalus ir statistikos p-reikšmes.

Pagal duomenų daugiamačio normaliojo pasiskirstymo modelį, tikėtinumo funkcija gali būti maksimizuojama naudojant tikėtinumo-maksimizavimo algoritmą (angl. *Expectation-Maximization* – toliau EM). Šio metodo principą aprašysime.

Pažymėkime stebėtus duomenis $\mathrm{Y_{obs}}$, o trūkstamus duomenis $\mathrm{Y_{miss}}$. Pilni duomenys nusakomi sujungta šitų duomenų aibe $\mathrm{Y} = (\mathrm{Y_{obs}}, \mathrm{Y_{miss}})$. Tarkime, kad egzistuoja jungtinė pasiskirstymo tankio funkcija:

$$\mathrm{p(y|\theta)} = \mathrm{p(y_{obs}, y_{miss}|\theta)} = \mathrm{p(y_{miss}|y_{obs}, \theta)} * \mathrm{p(y_{obs}|\theta)}$$

kur θ yra pasiskirstymo parametrų aibė.

Su tankio funkcija galima apibrėžti pilnų duomenų tikėtinumo funkciją :

$$L(\theta|Y) = L(\theta|\mathrm{Y_{obs}}, \mathrm{Y_{miss}}) = p(\mathrm{Y_{obs}}, \mathrm{Y_{miss}}|\theta)$$

Funkcija $L(\theta|\mathrm{Y_{obs}})$ vadinama nepilnų duomenų tikėtinumo funkcija. Kadangi trūkstami duomenys pagal prielaidą pasiskirstę pagal tam tikrą pasiskirstymą, funkciją $L(\theta|\mathrm{Y_{obs}}, \mathrm{Y_{miss}})$ galima laikyti kaip funkciją nuo atsitiktinio kintamojo $\mathrm{Y_{miss}}$ su konstantomis $\mathrm{Y_{obs}}$ ir θ :

$$L(\theta|\mathrm{Y_{obs}}, \mathrm{Y_{miss}}) = f_{(\mathrm{Y_{obs}},\theta)}(\mathrm{Y_{obs}})$$

EM algoritmas vykdomas dviem žingsniais:

1) Tikėtinumo žingsnis.

   Pilnų duomenų logtikėtinumo funkcijos tikėtinumas apibrėžiamas taip:

$$Q\left(\theta, \theta^{(t)}\right) = \mathrm{E}_{\theta^{(t)}}\left[\log \mathrm{p}\left(\mathrm{Y_{mis}}, \mathrm{Y_{obs}}|\theta\right)|\mathrm{Y_{obs}}, \theta^{(t)}\right]$$

   kur , $\theta^{(t)}$ yra esama tam žingsnyje parametrų aibė, kurią naudojama apskaičiuoti tikėtinumą. Čia $\mathrm{Y_{obs}}$ ir $\theta^{(t)}$ yra žinomos konstantos ir θ yra ieškomas naujas parametrų rinkinys. Kadangi Y_{miss} yra atsitiktinis kintamasis pagal tam tikrą pasiskirstymą $f\left(y_{miss}|Y_{obs}, \theta^{(t)}\right.$. Tada tikėtinumą galima užrašyti ir taip :

$$\mathrm{E}_{\theta^{(t)}}\left[\log \mathrm{p}\left(\mathrm{Y_{mis}}, \mathrm{Y_{obs}}|\theta\right)|\mathrm{Y_{obs}}, \theta^{(t)}\right] = \int_{\mathrm{y_{miss}} \in \Omega} \log \mathrm{p}\left(\mathrm{y_{miss}}, \mathrm{Y_{obs}}|\theta\right) * f\left(\mathrm{y_{miss}}|\mathrm{Y_{obs}}, \theta^{(t)}\right)\mathrm{dy_{miss}}$$

   čia Ω yra galimų reikšmių erdvė, kurią gali įgyti y_{miss}.

2) Maksimizavimo žingsnis.

   Šiame žingsnyje naudojant didžiausio tikėtinumo metodą, randami parametrų įverčiai, kurie maksimizuoja pilnų duomenų logtikėtinumo funkciją:

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, Q\left(\theta, \theta^{(t)}\right)$$

Kartojant tikėtinumo ir maksimizavimo žingsnius, kiekvienoje iteracijoje didėjant logtikėtinumui, algoritmas garantuotai konverguoja į tikėtinumo funkcijos lokalų maksimumą. [3,26]

Trumpai apibendrinant, tikėtinumo žingsnyje trūkstami duomenys yra užpildomi pagal esamas parametrų ir esamų duomenų reikšmes, o maksimizavimo žingsnyje naudojant didžiausio tikėtinumo metodą, randami nauji parametrų įverčiai.

1.2.2. DAUGKARTINIO UŽPILDYMO METODAS

Kitas metodas, daugkartinis užpildymas, pakeičia trūkstamas reikšmes aibe tikėtinų reikšmių, kuri atspindi neapibrėžtumą tinkamos reikšmės pasirinkime. Tada statistinė analizė atliekama šiai užpildytų

duomenų aibei ir gaunami rezultatai sujungiami. Šia procedūra gaunamos patikimos statistinės išvados, kurios deramai atspindi dėl trūkstamų reikšmių gaunamą neapibrėžtumą, kaip, pavyzdžiui, pasikliautinus parametrų intervalus.

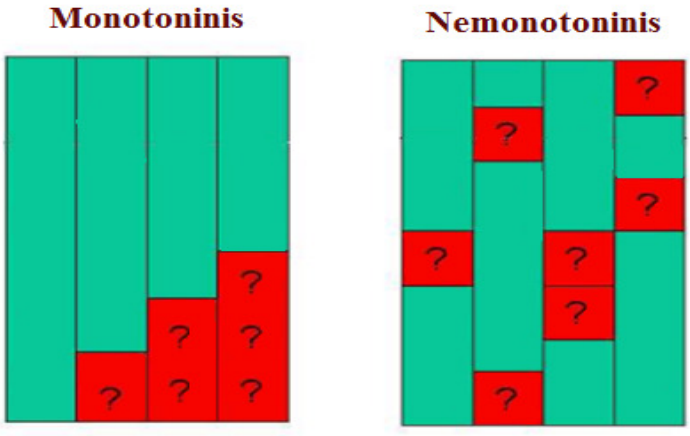
Daugkartinį užpildymą sudaro trys atskiri etapai :

1) Trūkstami duomenys yra užpildomi m kartų, taip sugeneruojant m pilnų duomenų aibių;
2) m pilnų duomenų aibių yra analizuojamos standartinėm statistinėmis procedūromis;
3) Rezultatai iš m pilnų duomenų aibių yra sujungiami ir padaromos išvados.

Pagal trūkstamų duomenų išsidėstymą, daugkartinio užpildymo metodai yra skirstomi į dvi klases. Duomenų aibė turi monotoninį duomenų išsidėstymą, jei kai trūksta kintamojo Y_j duomenų reikšmės, tai trūksta to paties kintamojo visų reikšmių Y_k (kai $k > j$) . Monotoninis išsidėstymas pavaizduotas 1.2.2 paveiksle kairėje. Tokiam duomenų išsidėstymui naudojami tokie metodai:

1) Regresijos metodas
2) Polinkio rezultatas (angl. *Propensity score*)

Nemonotoninis trūkstamų duomenų išsidėstymui (pavaizduotas 1.2.2 paveiksle dešinėje) naudojamas Markovo grandinių Monte Carlo metodas. Kadangi GRID tinklo duomenų klasterių trūkstamų duomenų išsidėstymas yra nemonotoninis, plačiau aptarsime tik Markovo grandinių Monte Carlo metodą.[11]



1.2.2 pav. Monotoninis (kairėje) ir nemonotoninis (dešinėje) trūkstamų duomenų išsidėstymas

1.2.3. MARKOVO GRANDINIŲ MONTE CARLO METODAS

Remiantis Bajeso teorema, informacija apie ieškomus parametrus θ yra išreiškiama aposteriorinio tikimybinio pasiskirstymo forma:

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{\int P(Y|\theta)P(\theta)d\theta}$$

Daugelyje nepilnų duomenų atvejų, pilnų duomenų aposteriorinis pasiskirstymas $P(\theta|Y_{obs})$ yra sunkiai gaunamas, jo negalima lengvai apibendrinti ar imituoti. Pirmuoju daugkartinio užpildymo žingsniu iš šio skirstinio turi būti atsitiktinai imitaciniu būdu sugeneruotos parametrų reikšmės , kad būtų galima imitaciniu būdu generuoti Y_{miss} reikšmes iš jo prognozuojamo pasiskirstymo $p(Y_{miss}|Y_{obs})$. Pasirodo, jeigu stebimi duomenys yra sujungiami su užpildytomis Y_{miss} reikšmėmis, gaunamas pilnų duomenų aposteriorinis pasiskirstymas $P(\theta|Y_{obs}, Y_{miss})$ yra daug lengviau apdorojamas. Dėl to buvo sukurtas duomenų sujungimo algoritmas (Tanner, Wong).

Kitaip negu standartiniai Monte Carlo metodai, kurie sukuria aibę nepriklausomų imitaciniu būdu modeliuotų reikšmių iš norimo tikimybinio pasiskirstymo, Markovo grandinių Monte Carlo metodas sukuria grandines, kuriose imitaciniu būdu modeliuotos reikšmės priklauso tik nuo ankstesnės reikšmės.

Pagrindinis principas MCMC yra toks, kad jei Markovo grandinė kuriama pakankamą iteracijų skaičių, ji artėja į norimą pasiskirstymą. Taip grandinė gali sukurti šio skirstinio imtis. Daugkartinio užpildymo kontekste, ieškomas pasiskirstymas yra jungtinis sąlyginis pasiskirstymas $P(Y_{miss}, \theta|Y_{obs})$.

Šis metodas atliekamas dviem žingsniais:

1) Užpildymo žingsnis. Su esamu parametrų rinkiniu $\theta^{(t)}$ (vidurkių vektoriumi ir kovariacijos matrica) ir stebėtais duomenimis Y_{obs} , imitacinio modeliavimo būdu užpildomi trūkstami duomenys iš sąlyginio prognozuojamo Y_{miss} (angl. *predictive*) pasiskirstymo :

$$Y_{MISS}^{(t+1)} \sim P\left(Y_{MISS}\middle|Y_{OBS}, \theta^{(t)}\right).$$

2) Aposteriorinis žingsnis. Tada panaudojant sujungtus duomenis (Y_{obs} ir Y_{miss}), iš aposteriorinio pasiskirstymo sugeneruojama nauja $\theta^{(t+1)}$ reikšmė:

$$\theta^{(t+1)} \sim P\left(\theta\middle|Y_{OBS}, Y_{MISS}^{(t+1)}\right).$$

Tai yra Markovo grandinių procedūra, kadangi kiekviena žingsnis priklauso nuo praėjusio, ir taip pat Monte Carlo procedūra, kadangi pradedama su atsitiktiniu duomenų užpildymu.

Atliekant šitą procesą nuo pradinių parametrų reikšmių $\theta^{(0)}$ gaunama Markovo grandinė $\{\theta^{(t)}, Y_{MISS}^{(t)}: t = 1,2, \dots\}$ kuri konverguoja į ieškomą pasiskirstymą $P(Y_{MISS}, \theta | Y_{OBS})$. Tada galima iš šio skirstinio imitaciškai modeliuoti trūkstamas reikšmes. [8]

1.2.4. SCENARIJŲ MEDIS

Stochastiniame optimizavime norint rasti optimalius sprendimus, reikia įvertinti parametrų neapibrėžtumą. Kelių stadijų stochastinio optimizavimo problema gali būti suformuluota taip:

minimizuoti

$$\min_{x \epsilon X} E_P \{f(\xi, x)\} = \min_{x \epsilon X} \int_{\Omega} f(\xi, x) \cdot d\, P(\xi)$$

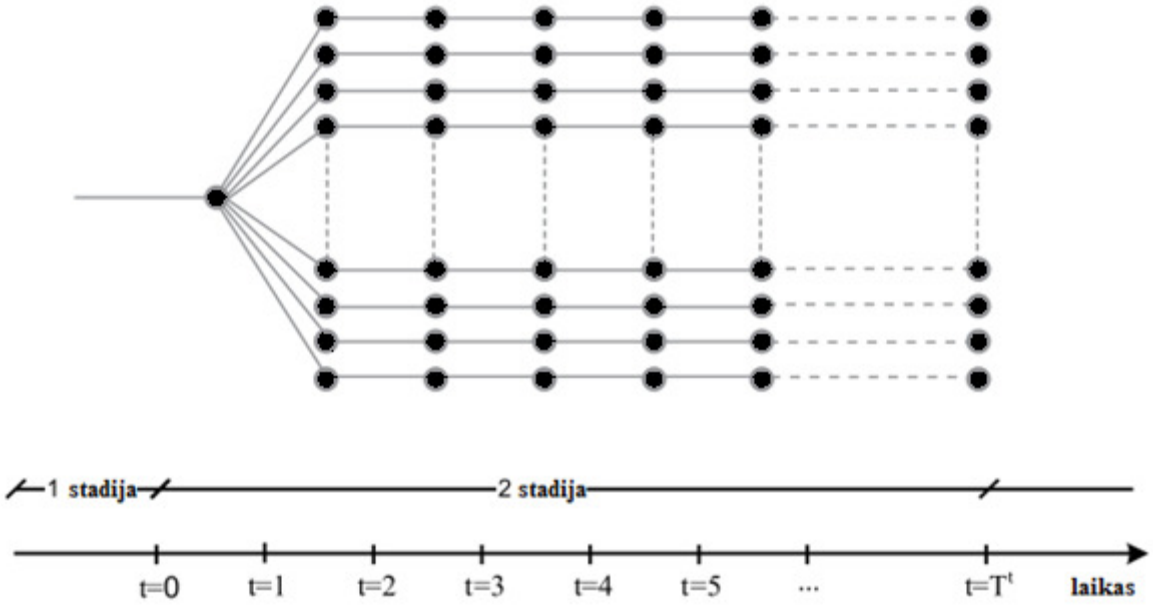
kur

- $x = \{x_t\}$ yra sprendimų aibė visoms stadijoms $t = 1,2, \dots, T$.
- X yra galimų sprendimų aibė.
- ξ yra atsitiktinis procesas.
- Ω yra visų galimų įvykių aibė.
- P yra tikimybinė atsitiktinio proceso funkcija.
- E_P yra tikėtina vertė pagal tikimybinę funkciją P.
- $f(\xi, x) -$ minimizuojama funkcija priklausianti nuo atsitiktinių parametrų.

Neapibrėžtumų išreiškimas stochastiniame programavime yra vienas iš svarbiausių uždavinių. Vienas iš dažniausiai aproksimavimas tikimybinio pasiskirstymo P, taip sugeneruojant scenarijų medį.

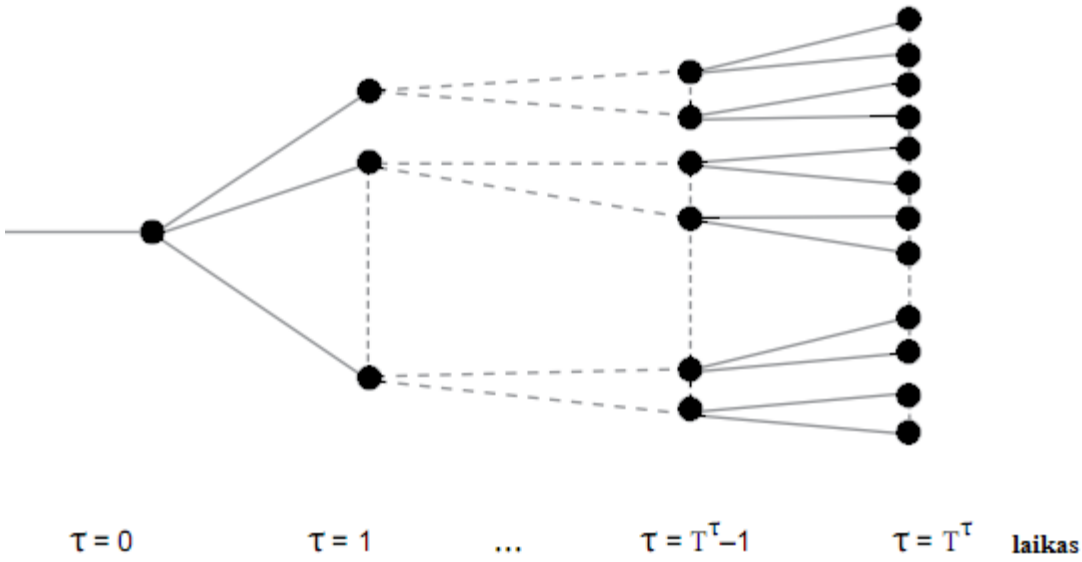
Pirmiausia apibrėšime laiko diskretizacijos indeksą t, kurio visų diskretizavimo momentų aibė žymima taip : $t \in \{0, \dots, T^t\}$, kur $t = T^t$ yra laiko horizontas, $t = 0$ yra pradinis laiko momentas. Laiko žingsnis Δt yra laiko tarpas nuo laiko $t - 1$ iki laiko t. Tarkime stochastinis procesas $\xi = \{\xi_t\}_{t=1}^{T^t}$ yra apibrėžiamas pagal filtruotą tikimybinę erdvę $(\Omega, S, \mathcal{F}, P)$. Imties erdvė Ω yra baigtinio matavimo, σ-algebra S yra aibė įvykių su priskirtom tikimybėmis pagal tikimybinį skirstinį P ir $\{\mathcal{F}_t\}_{t=1}^{T^t}$ yra S filtracija. Scenarijais paremtiems modeliams priimta, kad tikimybinis pasiskirstymas yra diskretus.

Apibrėšime dvi scenarijų struktūras. Laiko momentu $t = 0$ stochastinio proceso reikšmė ξ_0 yra žinoma ir neatsitiktinė. Tada visi scenarijai imitaciniu būdu sumodeliuojami iš pagrindinio mazgo ξ_0. Šis mazgas ir reprezentuoja pirmąją stadiją. Antroje stadijoje struktūra išsišakoja į individualius scenarijus, kaip pavaizduota 1.2.4.1 paveiksle. Tokia struktūra vadinama scenarijų vėduokle (angl. *scenario fan*).



1.2.4.1 pav. Scenarijų vėduoklės pavyzdys

Antroji scenarijų struktūra yra kelių stadijų scenarijų medis, kuris leidžia atspindėti tarp stadijinę priklausomybę ir sumažinti mazgų skaičių. Laiko stadijų indeksas $t \in \{0, \dots, T^t\}$, kur $t = T^t$ yra laiko horizontas, $t = 0$ yra pradinio sprendimo momentas yra asocijuotas su laiko momentais, kada priimamai sprendimai. Stadija yra laiko tarpas $\Delta\tau$ nuo laiko $\tau - 1$ iki laiko τ. Struktūra taip pat prasideda pagrindiniame mazge $\tau = 0$ ir prasideda pagrindiniame mazge, kuris šakojasi į baigtinį skaičių scenarijų, kaip pailiustruota 1.2.4.2 pav. Kelias per medį nuo pagrindinio mazgo iki paskutinio mazgo laiko momentu $\tau = T^t$ vadinamas scenarijumi . [6,7]



1.2.4.2 pav. Scenarijų medžio pavyzdys

Toliau aptarsime scenarijų medžio generavimo pagrindines metodikas.

Pirmas žingsnis generavime scenarijų medžio yra aprašymas atsitiktinių kintamųjų statistinių savybių. Jei atsitiktinai kintamieji diskretūs su keliomis kombinacijom, medžio generacija yra tiesmuka ir gali būti atlikta rankiniu būdu. Bet visais kitais atvejais to atlikti rankiniu būdu praktiškai neįmanoma [13] .

Dažniausiai scenarijų generavimo procedūrą sudaro keli arba visi sekantys žingsniai:

1) Modelio prielaidos apie elgseną atsitiktinių parametrų;
2) Įvertinimas pasirinkto modelio parametrų, kurie naudoja praeities duomenis;
3) Generacija duomenų trajektorijų kelių remiantis pasirinktu modeliu ar skirstinių diskretizacija naudojant statistinių savybių aproksimaciją.
4) Sąlyginė trajektorijų atranka, su kurios pagalba sukonstruojamas scenarijų medis su norimomis savybėmis.

Daugeliu atvejų taip pat atliekama gauto scenarijaus medžio redukcija, kurios pagalba pateikiami modelio atskiri atvejai, kurie gali būti realistiškai optimizuoti. 1.2.4 lentelėje išvardintos dažniausiai taikomos technikos atlikti scenarijaus generacijos žingsnius [6].

1.2.4 lentelė. Metodika naudojama scenarijų generavime

Tikslas	Metodas
Duomenų trajektorijų generavimas	Ekonometriniai modeliai ir laiko eilutės: • Autoregresiniai modeliai: Ar(p) • Slenkančio vidurkio modeliai: MA(q) • Autoregresiniai slenkančio vidurkio metodai: ARMA(p,q) • Apibendrintas autoregresijos sąlyginio heteroskedastiškumo modelis (p,q) • Vektorinis Auto Regresiniai modeliai: VAR • Bajeso VAR • Sumažinto rango regresija (angl. *Reduced Rank Regression*) Difuzijos procesai: • Vynerio procesas (Browno judėjimas) • Apibendrintas Vynerio procesas (Browno judėjimas su dreifu)
Diskretizacija	Statistinė aproksimacija: • Savybių atitikimo (Property matching) • Momentų atitikimo (Moment matching) • Ne parametriniai metodai Atrankos metodai (Sampling) : • Atsitiktinė atranka (Random sampling) • Stratifikuota atranka (Stratified sampling) • Bootstrapping
Medžio konstravimas ir sąlyginė atranka	• Optimali diskretizacija (Optimzal discretisation) • Baricentrinė (angl. Barycentric) aproksimacija • Nuosekli klasterizacija (Sequential clustering)
Redukcija	• Scenarijaus redukcijos
Vidaus atranka	• Stochastinė dekompozicija • Stochastinis kvazi-gradiento metodas (angl. *Quasi-gradient*)

1.2.5. SCENARIJŲ MEDŽIO GENERAVIMAS KLASTERIZAVIMO BŪDU

Scenarijų medžio sudarymas kelių stadijų klasterizavimo būdu buvo pasiūlytas Dupačova [7]. Bendrai šie metodai dažniausiai susideda iš dviejų fazių – imitacinio modeliavimo ir klasterizavimo fazės.

Imitacinio modeliavimo fazėje, daug skirtingų duomenų trajektorijų yra sumodeliuojama remiantis tam tikru duomenis atitinkančiu laiko eilučių modeliu, kur kiekviena imitaciniu būdu sumodeliuota trajektorija reprezentuoja vieną galima ateities prognozę. Visos sumodeliuotos trajektorijos sudaro scenarijų vėduoklę. Antroje fazėje iš scenarijų vėduoklės, sujungiant panašias duomenų trajektorijas į klasterius keliais laiko momentais, sukuriamas scenarijų medis.

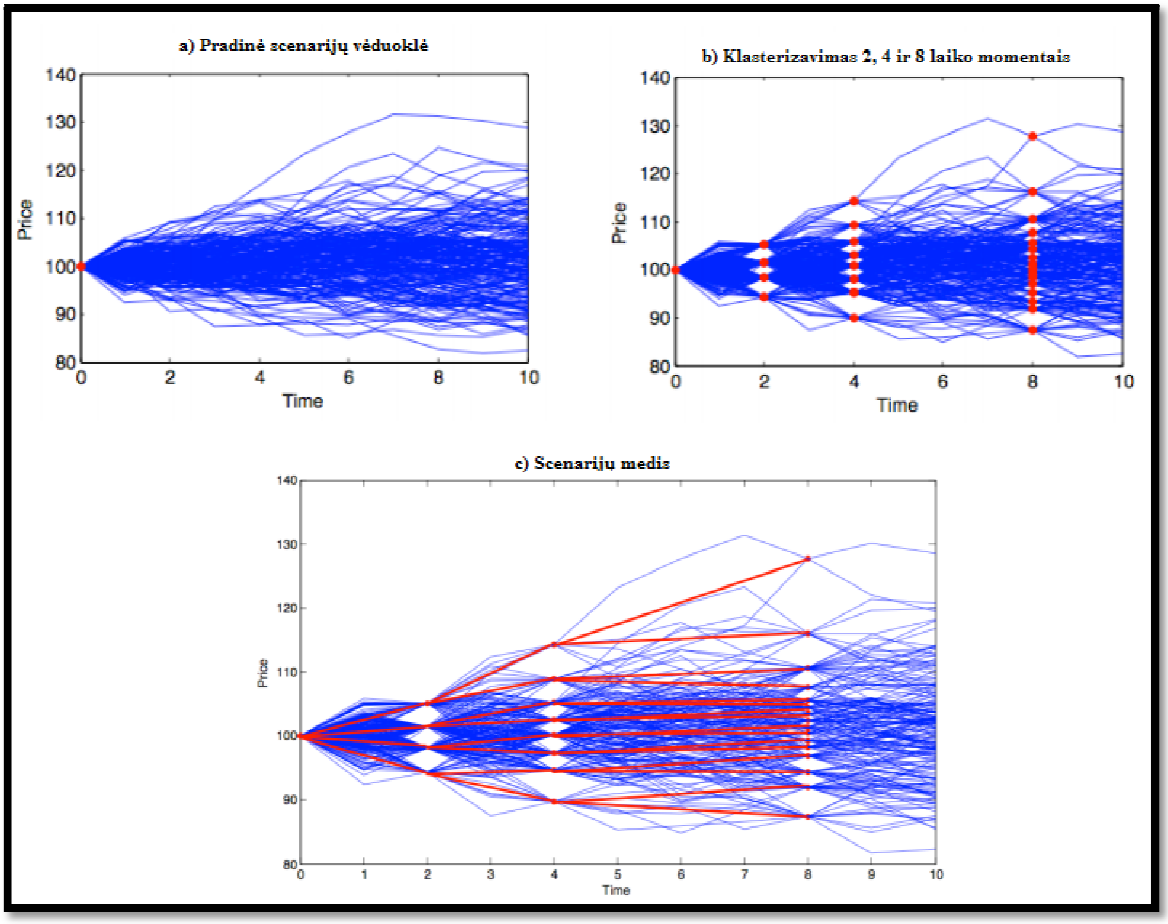
Imitacinio modeliavimo fazei dažniausiai naudojami du metodai :

1) Lygiagretus imitacinis modeliavimas – šituo būdu visos duomenų trajektorijos yra imitaciniu būdu sumodeliuojamos prieš klasterizavimą;

2) Nuoseklus imitacinis modeliavimas – imitacinio modeliavimo ir klasterizavimo fazės yra atliekama paeiliui.

Paraleliniu metodu gaunamos duomenų trajektorijos yra platesnės, su daugiau ekstremalių kintamųjų reikšmių, negu naudojant nuoseklų imitacinį modeliavimą. [15,22]

Pateiksime paralelinio klasterizavimo paprastą iliustracinį pavyzdį (Reynisson [22]) .Iš pradinės scenarijų vėduoklės (1.2.5 pav. a)), klasterizuojama 2,4 ir 8 laiko momentais (1.2.5 pav. b)), o atitinkamus duomenų trajektorijų klasterius sujungus skirtingais laiko momentais gaunamas scenarijų medis (1.2.5 pav. c)).



1.2.5 pav. Scenarijų medžio sudarymo schemos pavyzdys

1.2.6. KLASTERINĖ ANALIZĖ

Taikydami klasterinę analizę, nustatome objektų panašumą ir suskirstome juos į klasterius. Klasteris – panašių objektų grupė. Klasterinės analizės tikslas – suskirstyti objektus taip, kad skirtumai klasterių viduje būtų kuo mažesni, o tarp klasterių – kuo didesni.

Klasterizuodami turime pareiti 5 etapus:

1) Pasirinkti klasterizuojamus objektus.
2) Nuspręsti, pagal kokius požymius klasterizuosime.
3) Pasirinkti kiekybinį matą, kuriuo matuosime objektų panašumą.
4) Vienu ar kitu metodu suskirstyti objektus į klasterius.
5) Peržiūrėti gautus rezultatus.

Dažniausiai naudojami panašumo matai:

1) Metriniai atstumo matai.
2) Koreliacijos koeficientai.
3) Asociatyvumo koeficientai.

Euklido atstumas tarp objektų X ir Y aprašomas tokia formule:

$$\|X - Y\| = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Skiriamos dvi pagrindinės klasterizavimo analizės metodų klasės – hierarchiniai ir nehierarchiniai metodai. [4]

1.2.7. HIERARCHINIS KLASTERIZAVIMAS

Hierarchinių metodų rezultatai nusako klasterių tarpusavio hierarchiją, t.y. visi objektai laikomi vienu dideliu klasteriu, kurį sudaro mažesni klasteriai, šiuos savo ruožtu dar mažesni ir t.t. .

Taikydami hierarchinius metodus, nustatome bendrą visų klasterių skaičių ir tik po to sprendžiame, koks klasterių skaičius optimalus. Hierarchiniai metodai skirstomi į jungimo ir skaidymo metodus. Jungimo metodai smulkius klasterius jungia vis į stambesnius, kol galų gale lieka vienas. Hierarchinių jungimo metodų strategiją galima apibrėžti taip:

1) Turime N klasterių po 1 objektą ir $N \times N$ simetrinę atstumų matricą $(d_{ij})_{i,j}$.
2) Pagal atstumų matricą nustatome du klasterius, tarp kurių atstumas yra mažiausias. Tarkime, kad tai klasteriai U ir V.
3) Sujungiame klasterius U ir V. Naująjį klasterį pavadiname (UV). Tada atstumų matricą pakeičiame taip:
   a) Išbraukiame stulpelius ir eilutes, atitinkančius klasterius U ir V,
   b) pridedame eilutę ir stulpelį su atstumais tarp (UV) ir likusiųjų klasterių.

4) Kartojame 2 ir 3 žingsnius $(N - 1)$ kartų. Procesą baigiame, kai visi objektai yra viename klasteryje.

Šio proceso schema vaizduojama grafiku, vadinamu dendograma. [4]

Dažniausiai naudojami klasterių panašumo matai pateikti 1.2.7 lentelėje.

1.2.7 lentelė. Klasterių panašumo matai

Atstumas	d(U, V)
Vienetinės jungties (artimiausio kaimyno)	$\mathrm{d(U,V)} = \min_{\mathrm{X_i}\in\mathrm{U},\mathrm{Y_j}\in\mathrm{V}} \mathrm{d}(\mathrm{X_i},\mathrm{Y_j})$, $\mathrm{X_i}$ – i-asis U objektas, $\mathrm{Y_j}$ – j-asis V objektas,
Pilnosios jungties (tolimiausio kaimyno)	$d(U,V) = \max_{X_i\in U, Y_j\in V} d\left(X_i, Y_j\right)$
Vidutinės jungties	$d(U,V) = \sum_{X_i\in U}\sum_{Y_j\in V} d(X_i,Y_j)/(\mathrm{n_U n_V})$ n_U, n_V- klasterio objektų skaičius
Centrų	$d(U,V) = d(\bar{U},\bar{V})$, $\bar{U}$ ir $\bar{V}$ – klasterius sudarančių objektų požymių vidurkiai
Vordo	$d(U,V) = \|\bar{U}-\bar{V}\|^2/(1/n_U + 1/n_V)$

1.2.8. KOPHENETINIS KORELIACIJOS KOEFICIENTAS

Kophenetinis (angl. *cophenetic*) koreliacijos koeficientas naudojamas kaip dydis, parodantis kaip tiksliai dendograma išlaiko atstumą tarp duomenų taškų, jis naudojamas įvertinti naudojamą panašumo matą. Naudojant pradinius duomenis X_{ij} ir gautą dendogramą T_{ij}, jis apskaičiuojamas taip:

$$c = \frac{\sum_{i<j}(x(i,j)-x)(t(i,j)-t)}{\sqrt{\left[\sum_{i<j}(x(i,j)-x)^2\right]\left[\sum_{i<j}(t(i,j)-t)^2\right]}}$$

čia $x(i,j) = \left|X_i - X_j\right|$ - įprastas Euklido atstumas tarp i-ojo ir j-ojo stebėjimo, x- vidurkis $x(i,j)$, o $t(i,j)$- dendogramos atstumas tarp sumodeliuotų taškų T_i ir T_j, t- vidurkis $t(i,j)$. [25]

1.2.9. SILUETO INDEKSAS

Nustatant optimalų klasterių skaičių naudosime Silueto (angl. silhouette) validavimo indeksą. Jis apskaičiuojamas pagal tokią formulę:

$$S(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}}$$

kur $a(i)$ yra i-ojo objekto nepanašumo palyginimo su visais kitais objektais tame pačiame klasteryje vidurkis, $b(i)$ – minimalus nepanašumo vidurkis lyginant i-ąjį objektą su visais objektais artimiausiame klasteryje. Kuo šio indekso reikšmė artimesnė 1, tuo parinktas klasterių skaičius yra optimalesnis. [1]

1.2.10. GARCH MODELIS

Apibendrintas autoregresijos sąlyginio heteroskedastiškumo (angl. *Generalized AutoRegressive Conditional Heteroskedasticity*) GARCH (p,q) modelis buvo pasiūlytas Bollerslev(1986) darbuose.

GARCH modelis buvo pasiūlytas norint sumodeliuoti laiko eilutes pagal duomenis su „sunkiomis uodegomis“, didelėmis asimetrijos ir eksceso koeficiento reikšmėmis. Šitas modelis į prognozavimą įtraukia sąlyginę variaciją priklausančios nuo praeitų eilutės laiko tarpų, kas leidžia modeliuoti eilutės duomenų ryškų kintamumą. Procesas tariama, kad varijuoja aplink savo pastovų vidurkį c :

$$y_t = c + \varepsilon_t$$

kur ε_t yra baltasis triukšmas su vidurkiu 0.

Sąlyginė variacija apibrėžiama taip:

$$\sigma_t^2 = \kappa + \sum_{i=1}^{P} G_i \cdot \sigma_{t-i}^2 + \sum_{j=1}^{Q} A_j \cdot \varepsilon_{t-j}^2$$

su apribojimais:

$$\sum_{i=1}^{P} G_i + \sum_{j=1}^{Q} A_j < 1$$

$$\kappa > 0, G_i \geq 0, A_j \geq 0 .$$

Čia κ −konstanta, G ir A parametrai.

Ganėtinai paprastas GARCH(1,1) modelis aprašomas taip:

$$\sigma_t^2 = \kappa + G \cdot \sigma_{t-1}^2 + A \cdot \varepsilon_{t-1}^2$$

dažnai atitinka empiriniuose bandymuose modeliuojamas eilutes.

Parametrai G ir A yra įvertinami naudojant didžiausio maksimalaus tikėtinumo metodus. [17]

1.2.11. PROGNOZAVIMO PAKLAIDOS MATAI

Dažniausiai prognozavimo tikslumui nustatyti naudojamas matai [22]:

1) RMSE – vidurkinė kvadratinė paklaida (angl. *root mean squared error*)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - p_i)^2}$$

Kuo RMSE reikšmė mažesnė, tuo tikslesnis prognozavimas.

2) MAE – klaidos absoliutinis vidurkis (angl. *mean absoliute error*)

$$MAE = \sum_{i=1}^{n}\frac{|r_i - p_i|}{n}$$

Analogiškai, kuo MAE reikšmė mažesnė, tuo tikslesnis prognozavimas.

3) Santykinė paklaida:

$$SP = \frac{p - r}{r} \cdot 100\%$$

4) Vidutinė santykinė paklaida:

$$VSP = \frac{SP_1 + SP_2 + \ldots + SP_n}{n}$$

Visose paklaidose r_i yra stebima reikšmė momentu i ir p_i - prognozuota reikšmė, $i = 1, \ldots, n$.

1.2.12. SCENARIJŲ GENERAVIMO ALGORITMAS

Scenarijų generavimo algoritmas buvo realizuotas Matlab matematinio paketo pagalba. Naudojamas paralelinis imitacinis modeliavimas, tai yra visos duomenų trajektorijos yra sugeneruojamos prieš klasterizavimą.

Algoritmas atlieka 5 stadijų scenarijų medžio generavimą, laiko tarpas tarp stadijų yra pusę dienos.

```
Nuskaitomi sugeneruotų trajektorijų duomenys 12 valandą
Duomenys klasterizuojami naudojant hierarchinio klasterizavimo metodą
Naudojant Silueto indeksą nustatomas optimalus klasterių skaičius n
Apskaičiuojami 1 stadijos klasterių centrai (mazgai) ir šakų tikimybės
    Nuo i= 1 iki  i= n
        Nuskaitomi 24 valandos duomenys
        i-asis klasteris yra klasterizuojmas  pagal 24 valandos duomenis
        Naudojant Silueto indeksą nustatomas optimalus klasterių skaičius i24
        Apskaičiuojami 2 stadijos i-ojo klasterių centrai (mazgai) ir šakų tikimybės
            Nuo j=1  iki  j=i24
                Nuskaitomi 36 valandos duomenys
                j-asis klasteris yra klasterizuojmas  pagal 36 valandos duomenis
                Nustatomas optimalus klasterių skaičius j36
                Apskaičiuojami 3 stadijos j klasterių centrai (mazgai) ir šakų tikimybės
                    Analogiškai 4 stadijai
                        Analogiškai 5 stadijai
                        Ciklo pabaiga
                    Ciklo pabaiga
            Ciklo pabaiga
    Ciklo pabaiga
```

Algoritmas pateikia kiekvieną sugeneruotą scenarijų, jo mazgų reikšmes ir šakų tikimybes kiekvienoje stadijoje. Matlab kodas pateiktas 1 priede.

2. TIRIAMOJI DALIS

Tiriamoji dalis suskaidyta į dvi dalis – pirmojoje dalyje bus palyginti duomenų atstatymo metodai GRID tinklo klasteriams, o antrojoje dalyje naudojant atstatytus duomenis bus sudarytas klasterio scenarijų medis.

2.1. DUOMENŲ ATSTATYMAS GRID TINKLO KLASTERIAMS

Naudojami BalticGrid klasterių duomenys buvo nepilni dėl to kad neveikė arba pats klasteris arba klasterio monitoringo sistema. Remiantis šia prielaida duomenų trūkumo tipas yra MAR(1.1.5. skyrius), todėl duomenis bus galima analizuoti SAS pakete esančiais MCMC ir EM metodais, kurie remiasi prielaida, kad duomenų trūkumas yra MAR ir duomenys yra pasiskirstę pagal normalųjį pasiskirstymą. Duomenų trūkumo pavyzdys pateiktas 2.1.1 lentele.

2.1.1 lentelė. Klasterių duomenų trūkumo pavyzdys

	Klasterio pavadinimas ir darbo apkrovos duomenys						
Laikas	1.grid.etf.r	pupa.elen.ktu.lt	grid.lumii.lv	grid.marko.lt	id.akolegija	.grid.eenet	grid.fi.lt
2010-10-30 12	25	660	1	160	193		211
2010-10-30 13	23	650	12	208	236	5286	256
2010-10-30 14	7		20	632	69		80
2010-10-30 15	19	629	11	177	207		226
2010-10-30 16	26	619	320	300	334	4758	945
2010-10-30 17	8	609	297	112	149		163
2010-10-30 18	15	599	270	236	275		287
2010-10-30 19	10	589	244	72	105		119
2010-10-30 20	26		217	200	238		249
2010-10-30 21	30	578					
2010-10-30 22	11	568	190	421	358	4112	70
2010-10-30 23	14	558	162	144	185		194
2010-10-31 00		549	137	266	301		314
2010-10-31 01	19	539	120	288	722		432
2010-10-31 02	66	529	95	119	155		166
2010-10-31 03	15	519	50	150	190	4603	197
2010-10-31 04	21	500	4	198	226	4313	249
2010-10-31 05	13	490	1	292	418		637
2010-10-31 06	507	480	32	1366	1396		1343
2010-10-31 07	69		2	161	189		212
2010-10-31 08	5		1	280	305		424

Paprastai trūkstami duomenys gali būti užpildomi ir naudojant laiko eilučių prognozėmis. Kodėl toks metodas netinka GRID tinklo klasteriams galima pademonstruoti pagal 2.1.2 lentelę.

2.1.2 lentelė. Duomenų trūkumo fragmentas

Laikas	Klasterių pavadinimai ir jų darbo apkrovos valandiniai duomenys				
	ce01.grid.etf.rtu.lv	pupa.elen.ktu.lt	grid.lumii.lv	grid.marko.lt	grid.akolegija.lt
2010-10-18 12	25	89624	255	224	256
2010-10-18 13	10077	85847	1032	40631	32665
	Nėra duomenų				
2010-10-19 11	43	240	300	310	632
2010-10-19 12			94		138
2010-10-19 13	70	26		103	140
2010-10-19 14		21	193	211	237
2010-10-19 15	32	21	284	723	648

Kadangi visi klasteriai „nulūžo“ laiko momentu 2010-10-18 13 valandą ir neveikė beveik dieną, klasterių ce01.grid.etf.rtu.lv, pupa.elen.ktu.lt ir grid.marko.lt 2010-10-19 12 valandai nebūtų galima atstatyti naudojant laiko eilutėmis, nes joms reikia daugiau praeities stebėjimų duomenų, bet šiems duomenims atstatyti tinka MCMC ir EM metodai naudojantys koreliaciją tarp klasterių.

Kadangi paros 5 klasterių pilnų duomenų eilučių nebuvo, šių metodų tinkamumo ir palyginimo tyrimui naudosime 18 valandinių duomenų, kurie pateikti 2.1.3 lentelėje.

2.1.3 lentelė. Pradiniai duomenys

Data\Klasteris	ce01.grid.etf.rtu.lv	pupa.elen.ktu.lt	grid.marko.lt	grid.akolegija.lt	grid.fi.lt
2010-10-29 06	55	32	305	631	649
2010-10-29 07	50	22	115	149	166
2010-10-29 08	43	12	227	260	279
2010-10-29 09	61	2	627	358	78
2010-10-29 10	53	244	115	154	163
2010-10-29 11	67	234	219	254	268
2010-10-29 12	56	224	308	643	655
2010-10-29 13	43	213	98	134	151
2010-10-29 14	68	203	207	246	260
2010-10-29 15	64	193	415	650	261
2010-10-29 16	118	183	118	158	168
2010-10-29 17	19	173	231	272	279
2010-10-29 18	16	163	641	75	88
2010-10-29 19	26	153	168	207	216
2010-10-29 20	16	143	287	422	634
2010-10-29 21	72	133	105	147	154
2010-10-29 22	14	123	216	255	265
2010-10-29 23	15	113	632	73	78

Atsitiktine tvarka buvo panaikinta 20 % duomenų (kiekviename klasteryje duomenų trūkumas buvo nuo 16,667 % iki 22,22 %) :

2.1.4 lentelė. Pradiniai duomenys po atsitiktinio duomenų pašalinimo

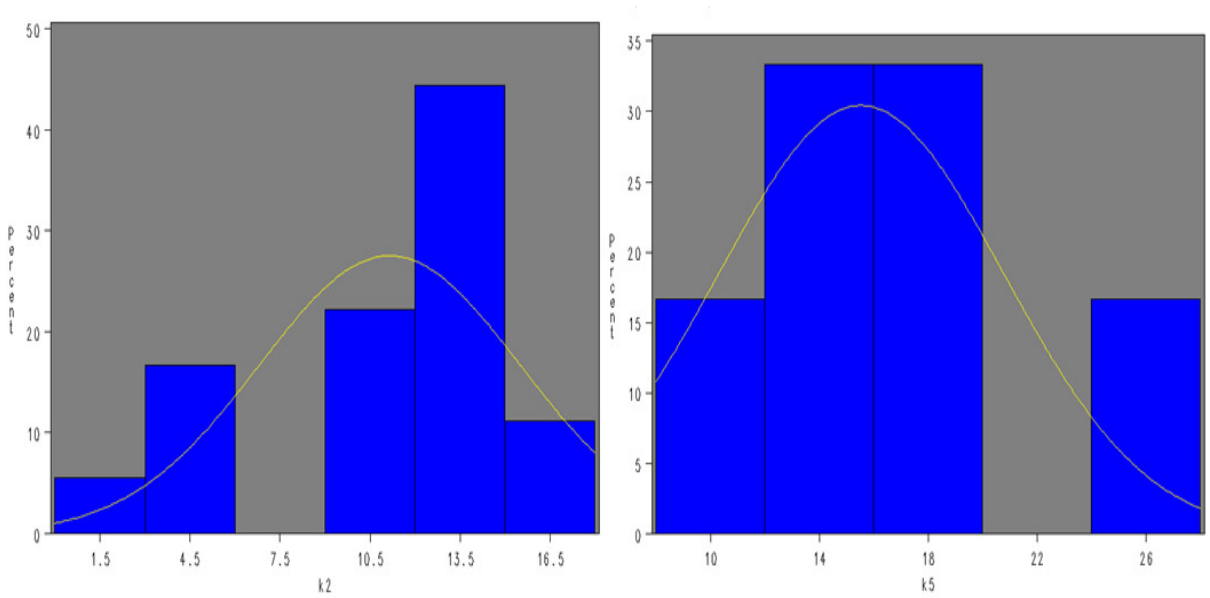
Data\Klasteris	ce01.grid.etf.rtu.lv	pupa.elen.ktu.lt	grid.marko.lt	grid.akolegija.lt	grid.fi.lt
2010-10-29 06	55	32	305	631	649
2010-10-29 07	50		115	149	166
2010-10-29 08	43	12		260	279
2010-10-29 09	61		627	358	
2010-10-29 10	53	244	115	154	163
2010-10-29 11		234	219	254	268
2010-10-29 12	56	224	308		655
2010-10-29 13	43		98	134	151
2010-10-29 14	68	203	207	246	
2010-10-29 15	64	193	415	650	261
2010-10-29 16	118	183			168
2010-10-29 17	19		231	272	279
2010-10-29 18	16	163	641		88
2010-10-29 19		153		207	216
2010-10-29 20	16	143	287	422	634
2010-10-29 21	72	133		147	
2010-10-29 22		123	216	255	265
2010-10-29 23	15	113	632		78

Duomenų atkūrimui buvo panaudoti MCMC (1.2.3. skyrius) ir EM (1.2.1 skyrius) metodai statistinio paketo SAS pagalba. Kadangi duomenys klasteriuose buvo pasiskirstę ne pagal normalųjį skirstinį, o abu šie metodai daro prielaidą, kad duomenys yra pasiskirstę pagal normalųjį skirstinį, todėl klasterių duomenys buvo transformuoti logaritmine ir šaknies traukimo transformacija. Hipotezei apie normalumą patikrinti buvo panaudotas Shapiro-Wilk testas. Jo rezultatai pateikti 2.1.5 lentelėje :

2.1.5 lentelė. Hipotezės apie pasiskirstymą pagal normalųjį dėsnį tikrinimo rezultatai.

Klasteris	t statistikos reikšmė
ce01.grid.etf.rtu.lv	0.1024
pupa.elen.ktu.lt	0.0095
grid.marko.lt	0.1421
grid.akolegija.lt	0.0846
grid.fi.lt	0.0144

Testas apie duomenų normalumą buvo atmestas „pupa.elen.ktu.lt“ ir „grid.fi.lt“. Šių dviejų klasterių histogramos pavaizduotos 2.1.1 paveiksle.



2.1.1 pav. Klasterių ,,pupa.elen.ktu.lt“ (kairėje) ir ,,grid.fi.lt“ (dešinėje) duomenų histogramos

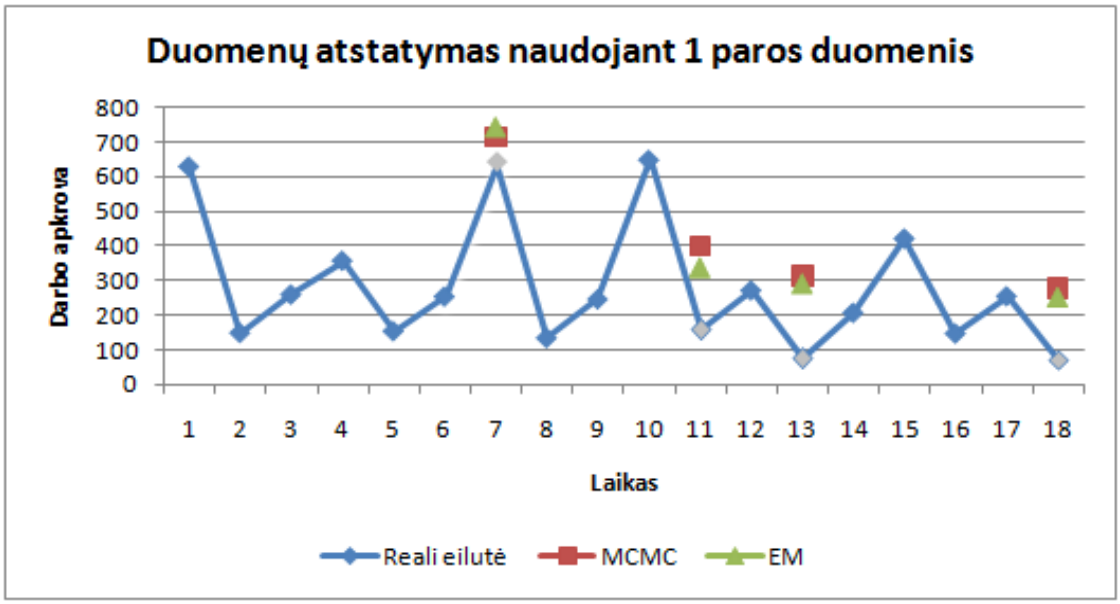
Nors ir atmetama šių dviejų klasterių hipotezė apie normalumą, remiantis Schafer(1987 [24]), kuris naudojant atliktus imitacinius modeliavimus ir praktinius rezultatus teigė, kad šie metodai ganėtinai gerai dirba netgi tada kai duomenys yra susiskaldę į kelias dalis. Dėl to tyrimui naudosime ir šių klasterių duomenis ir patikrinsime kaip duomenų nepasiskirstymas pagal normalųjį dėsnį lemia duomenų atstatymą.

Atlikus duomenų atstatymą, buvo apskaičiuoti pirmieji šių klasterių statistiniai momentai su atstatytais duomenimis, rezultatai palyginti su realiais duomenimis. Šio palyginimo rezultatai yra pateikti 2.1.6 lentelėje.

2.1.6 lentelė. Duomenų atstatymo metodų paklaidos pradiniams duomenims

Metodas	Santykinė paklaida	Klasterio pavadinimas					Vidutinė santykinė (1.2.11 skyrius) paklaida
		ce01.grid.etf.rtu.lv	pupa.elen.ktu.lt	grid.marko.lt	grid.akolegija.lt	grid.fi.lt	
MCMC	Vidurkio	2,72%	43,56%	31,09%	14,86%	2,59%	18,96%
	Variacijos	6,06%	136,28%	5,44%	8,07%	0,94%	31,36%
	Asimetrijos koeficiento	55,17%	520,07%	13,40%	12,10%	7,40%	121,63%
EM	Vidurkio	3,49%	21,28%	0,95%	13,37%	3,65%	8,55%
	Variacijos	2,35%	38,76%	1,16%	3,54%	2,61%	9,69%
	Asimetrijos koeficiento	40,85%	273,62%	1,85%	18,82%	0,76%	67,18%

Iš lentelės matome, kad didžiausios paklaidos gaunamos asimetrijos koeficiento, taip pat naudojant EM metodą, gaunamos ženkliai mažesnės paklaidos nei naudojant MCMC metodą. Ryškios ir labai didelės paklaidos gaunamos ne pagal normalųjį skirstinį pasiskirsčiusio klasterio „pupa.elen.ktu.lt“, tačiau kito ne pagal normalųjį skirstinį pasiskirsčiusio klasterio „grid.fi.lt“ gaunamos mažiausios iš klasterių paklaidos naudojant MCMC metodą ir dar mažesnės paklaidos naudojant EM metodą. Žemiau pateiktas klasterio „grid.akolegija.lt“ atstatytų ir realių duomenų palyginimo grafikas.



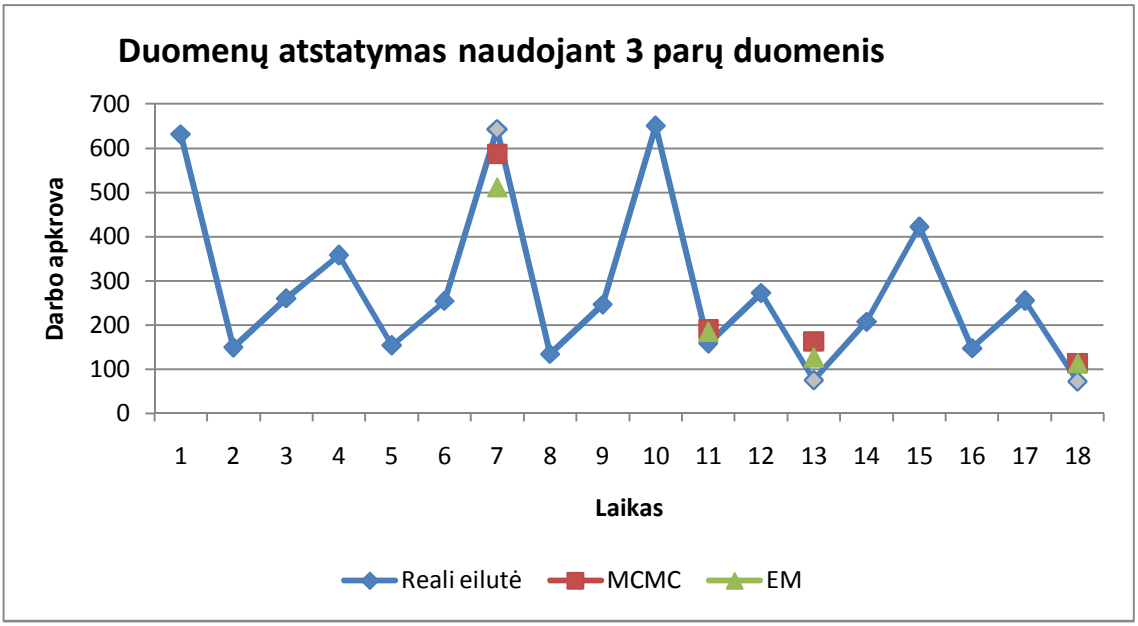
2.1.2 pav. Realių ir atstatytų duomenų palyginimas taikant MCMC ir EM

Toliau patikrinsime kaip šių metodų veikimas priklauso nuo duomenų kiekio naudodami papildomai dar dviejų parų duomenis. Atlikti analogiški palyginimo skaičiavimai pateikti 2.1.7 lentelėje.

2.1.7 lentelė. Duomenų atstatymo metodų paklaidos naudojant 3 parų duomenis

Metodas	Santykinė paklaida	Klasterio pavadinimas					Vidutinė santykinė paklaida
		ce01.grid.etf.rtu.lv	pupa.elen.ktu.lt	grid.marko.lt	grid.akolegija.lt	grid.fi.lt	
MCMC	Vidurkio	0,13%	6,78%	1,56%	2,05%	6,00%	3,30%
	Variacijos	5,55%	15,68%	1,49%	7,75%	2,74%	6,64%
	Asimetrijos koeficiento	35,75%	198,16%	0,89%	10,36%	6,07%	50,25%
EM	Vidurkio	1,74%	1,07%	4,65%	0,26%	6,43%	2,83%
	Variacijos	4,64%	14,58%	4,67%	10,30%	2,90%	7,42%
	Asimetrijos koeficiento	44,17%	60,79%	3,48%	9,58%	7,63%	25,13%

Naudojant daugiau duomenų, ženkliai sumažėjo MCMC metodo paklaidos, taip pat sumažėjo klasterio „pupa.elen.ktu.lt“ klasterio santykinė vidurkio ir variacijos paklaida. Klasterio „grid.akolegija.lt“ atstatytų ir realių duomenų palyginimo grafikas naudojant 3 dienų duomenis pateiktas 2.1.3 paveiksle.



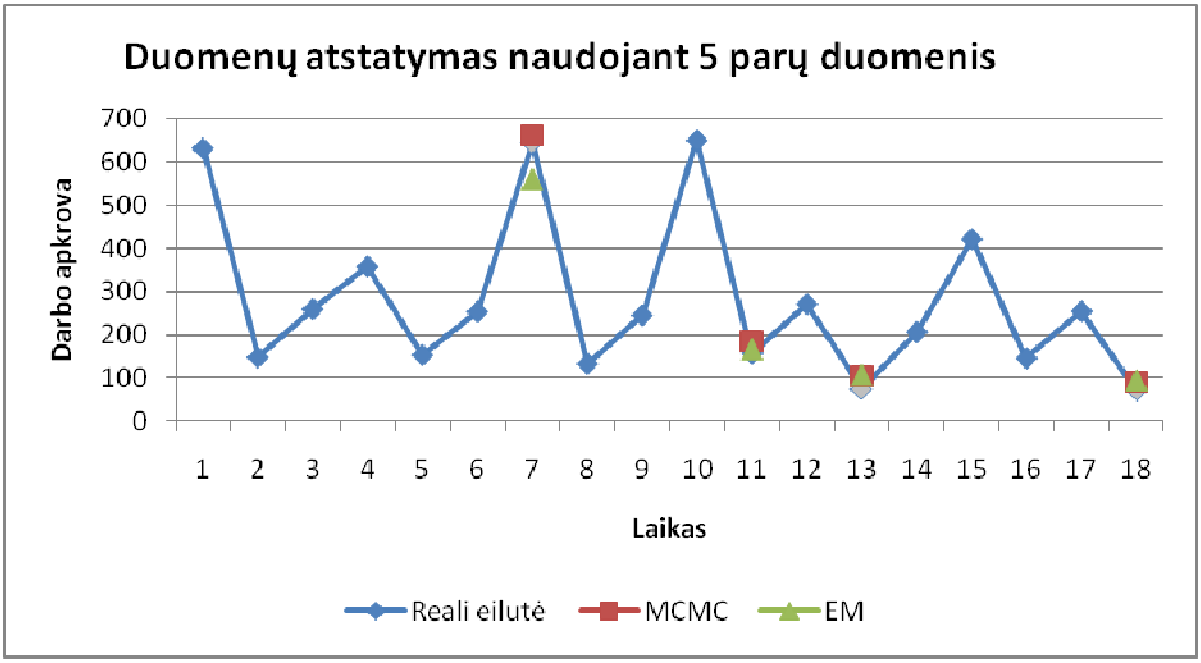
2.1.3 pav. Realių ir atstatytų duomenų palyginimas taikant MCMC ir EM, naudojant 3 parų duomenis

Kadangi asimetrijos paklaidos pirmiesiems dviem klasteriams gaunamos gana didelės, buvo atliktas toks pat tyrimas naudojant iš viso 5 dienų duomenis. Rezultatai pateikti 2.1.7 lentelėje.

2.1.8 lentelė. Duomenų atstatymo metodų paklaidos naudojant 5 parų duomenis

Metodas	Santykinė paklaida	Klasterio pavadinimas					Vidutinė santykinė paklaida
		ce01.grid.etf.rtu.lv	pupa.elen.ktu.lt	grid.marko.lt	grid.akolegija.lt	grid.fi.lt	
MCMC	Vidurkio	3,65%	16,00%	6,16%	1,84%	5,82%	6,70%
	Variacijos	4,49%	7,49%	4,86%	1,05%	2,96%	4,17%
	Asimetrijos koeficiento	9,12%	54,82%	3,16%	7,54%	5,72%	16,07%
EM	Vidurkio	1,71%	5,72%	4,84%	0,44%	5,94%	3,73%
	Variacijos	10,26%	20,41%	7,74%	9,29%	5,76%	10,69%
	Asimetrijos koeficiento	35,37%	8,30%	4,72%	3,63%	5,48%	11,50%

Didesnis duomenų kiekis gerokai sumažino šių klasterių asimetrijos koeficiento santykinę paklaidą naudojant abejus metodus.



2.1.4 pav. Realių ir atstatytų duomenų palyginimas taikant MCMC ir EM, naudojant 5 parų duomenis

Apibendrinant atliktus bandymus pateiksime vidutinės santykinės paklaidos (2.1.8 lentelė) ir vidurkinės kvadratinės paklaidos lenteles naudojant įvairų duomenų kiekį (2.1.9 lentelė).

2.1.9 lentelė. Pagrindinių momentų vidutinė santykinė paklaida naudojant skirtingą duomenų kiekį

	Vidutinė santykinė paklaida		
	Vidurkio	Variacijos	Asimetrijos
Naudojant 1 paros duomenis			
MCMC	18,96%	31,36%	121,63%
EM	8,55%	9,69%	67,18%
Naudojant 3 parų duomenis			
MCMC	3,30%	6,64%	50,25%
EM	2,83%	7,42%	25,13%
Naudojant 5 parų duomenis			
MCMC	6,70%	4,17%	16,07%
EM	3,73%	10,69%	11,50%

2.1.10 lentelė. Vidurkinė kvadratinė paklaida, naudojant skirtingą duomenų kiekį.

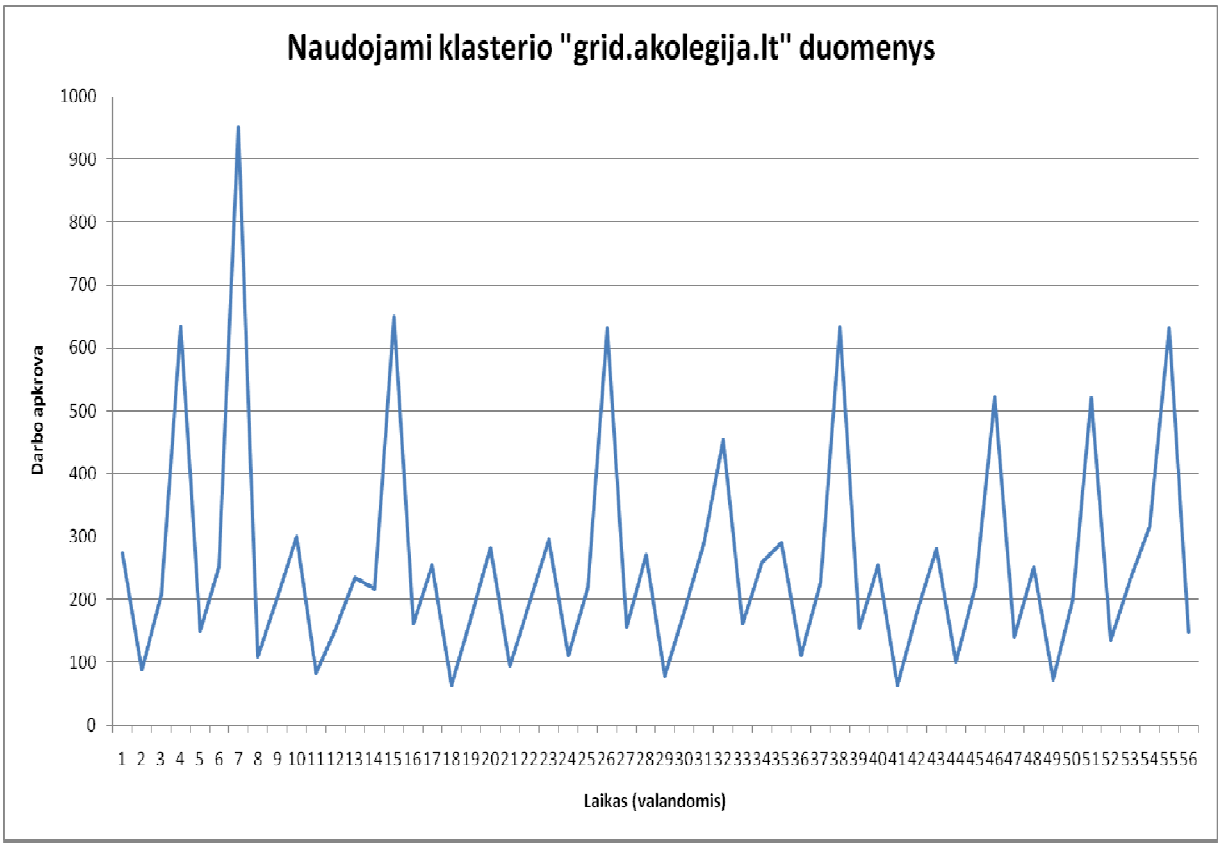
	Metodas	RMSE
Naudojant 1 paros duomenis		
	MCMC	89.23
	EM	52.34
Naudojant 3 parų duomenis		
	MCMC	44.54
	EM	39,56
Naudojant 5 parų duomenis		
	MCMC	42,93
	EM	38,11

Naudojant mažai duomenų, maksimalaus tikėtinumo metodu gaunamos paklaidos žymiai mažesnės už MCMC metodą. Naudojant abu metodus, gaunamos didelės asimetrijos koeficiento paklaidos . Didesnis duomenų skaičius ženkliai pagerina MCMC metodo efektyvumą, taip pat ryškiai sumažina asimetrijos koeficiento paklaidą naudojant abejus metodus.

Palyginus abu šiuos metodus, GRID tinklo klasteriams tinkamesnis maksimalaus tikėtinumo EM duomenų atstatymo metodas, kurį ir naudosime atstatant klasterių duomenis .

2.2. SCENARIJŲ MEDŽIO GENERAVIMAS

Scenarijų medį pirmiausia sugeneruosime klasteriui „grid.akolegija.lt“. Šio klasterio scenarijų medžio generavimui naudosime 2010-10-27 - 2010-10-29 šio klasterio duomenis panaudojant ir praeitame žingsnyje gautas atstatytas reikšmes. Šių duomenų grafikas pavaizduotas 2.2.1 paveiksle.



2.2.1 pav. Naudojamų duomenų grafikas.

Iš pirmo žvilgsnio atrodo, kad duomenų trajektorijos gali būti formuojamos naudojantis ARIMA modeliu, remiantis sezoniškais svyravimus. Tačiau statistiniai testai neaptinka sezoniškumo. Kadangi duomenų pasiskirstymas yra su didesniu už 1 asimetrijos koeficientu ir dideliu ekscesо koeficientu, duomenų trajektorijų imitaciniam modeliavimui naudosime apibendrintą autoregresijos sąlyginio heteroskedastiškumo Garch(1,1) (1.2.10 skyrius) modelį. Modelio eilė buvo nustatyta panaudojant Matalb lratiotest funkciją, kuri atlieka tikėtinumo santykio (angl. *likelihood ratio*) testą, įvertinant ar aukštesnės eilės modelis palyginus su esamu geriau aprašo proceso kintamumą. Rezultatai pateikti 2.2.1 lentelėje, kurioje buvo lyginta su GARCH(1,1) modeliu aukštesnės eilės modeliai, kadangi p-statistikos reikšmė visais atvejais (lentelėje pavaizduoti tik iki p) didesnė už 0.05, todėl hipotezė, kad aukštesnės eilės modelis geriau aprašo kintamumą, atmetama, todėl naudosime GARCH(1,1) modelį.

2.2.1 lentelė. Tikėtinumo santykio testas lyginant su GARCH(1,1) modeliu

Modelis	GARCH(2,1)	GARCH(3,1)	GARCH(4,1)	GARCH(5,1)	GARCH(6,1)
p-statistikos reikšmė	0,6623	0,608	0,5104	0,5394	0,5394

Jo parametrai gaunami panaudojus matematinio programinio paketo Matlab funkciją Garchfit, kur naudojant didžiausio tikėtinumo įverčius, randa parametrų įverčius ir t statistiką. 2.2.2 lentelėje pateikti gauti rezultatai.

2.2.2 lentelė. Parametrų nustatymas

Parametras	Reikšmė	Standartinė paklaida	T statistika
C	260,69	26,532	9,8256
K	2948,9	17712	3,1665
GARCH(1)	0,89221	0,66239	4,347
ARCH(1)	0	0,057247	0

Kadangi ARCH(1) statistikos reikšmė mažesnė už 2, jį atmesime.

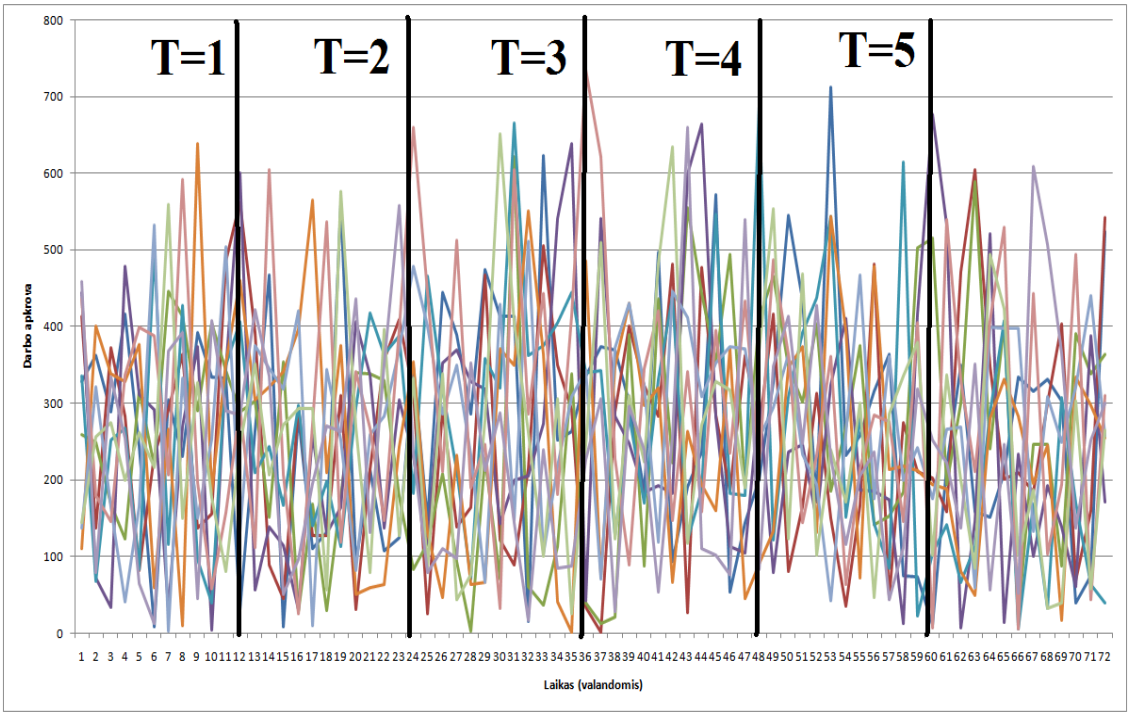
Procesą tada galima aprašyti taip:

$$y_t = 260{,}69 + \varepsilon_t$$

kur ε_t yra baltas triukšmas, o šio proceso variacija aprašoma taip:

$$\sigma_t^2 = 2948{,}9 + 0.89221 \cdot \sigma_{t-1}^2$$

Naudojantis Matlab Garchsim funkciją, buvo imitaciniu būdu sumodeliuota šio proceso 500 atsitiktinių duomenų trajektorijų ateinančiom 72 valandoms kadangi licenzija atlikti skaičiavimus GRID suteikiama būtent tokiam laiko tarpui. Dėl vaizdumo 2.2.2 paveiksle pateikta tik 10 realizacijų grafikas. Šio proceso scenarijų medį sudarysime su 12 valandų laiko tarpu (1.2.4 skyrius), iš viso 5 laiko stadijoms.



2.2.2 pav. 10 sugeneruotų proceso atsitiktinių trajektorijų

Toliau naudojant pirmos stadijos duomenis, ištirsime koks klasterizavimo panašumo matas tinkamiausias naudojant hierarchinį klasterizavimą šių duomenų klasterizavime. Apskaičiuotos kophenetinio (1.2.8 skyrius) koreliacijos koeficiento reikšmės pateiktos 2.2.3 lentelėje.

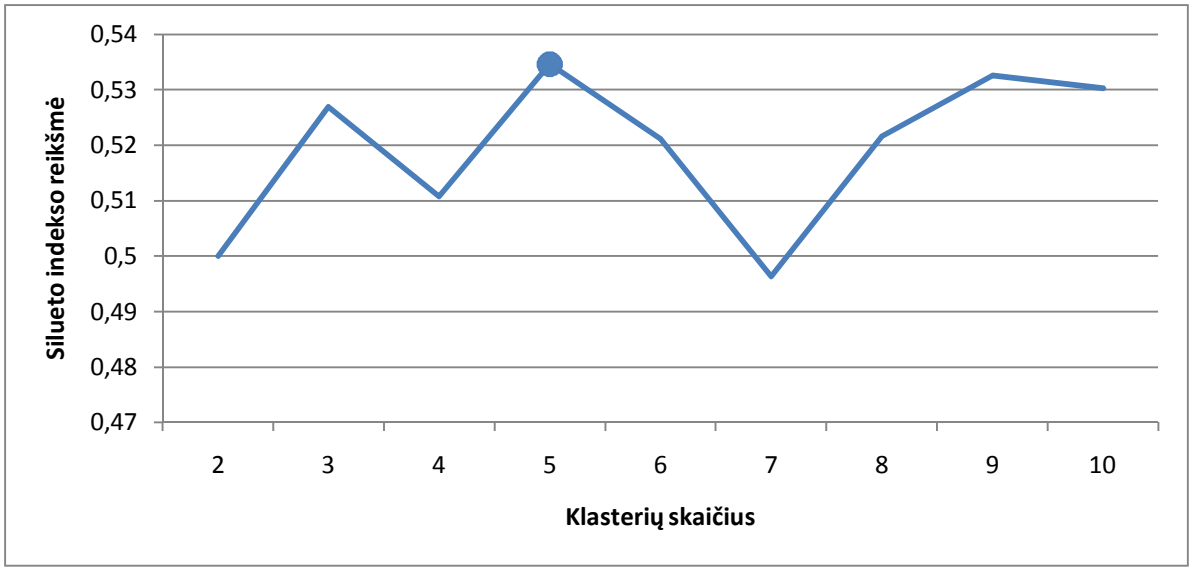
2.2.3 lentelė. Naudojamo panašumo mato nustatymas naudojant kophenetinį koreliacijos koeficientą

Panašumo matas	Kophenetinio koreliacijos koeficiento reikšmė
Vienetinės jungties	0,7019
Pilnosios jungties	0,6955
Centrų	0,7421
Vidutinės jungties	0,7421
Vordo	0,6996

Didžiausios šio koeficiento reikšmės gaunamos naudojant centrų ir vidutinės jungties panašumo matus dendogramos sudaryme. Pasirinksime centrų panašumo matą dendogramos sudaryme.

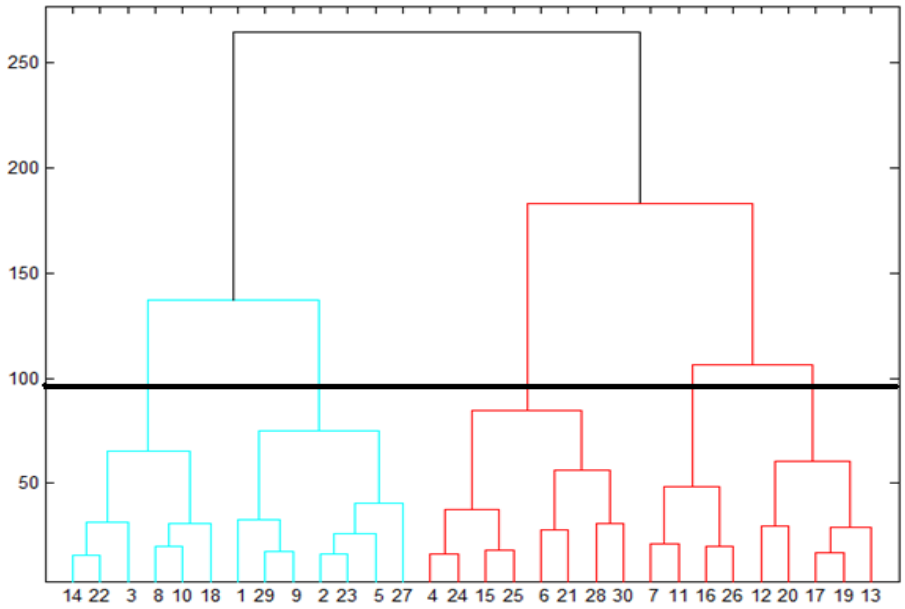
Tada pailiustruosime kaip kiekviename žingsnyje nustatomas klasterių (šakų skaičius). 1-ojoje stadijoje naudojant hierarchinį klasterizavimą (1.2.7. skyrius), apskaičiuojama Silueto indekso (1.2.9.

skyrius) reikšmė skirtingam klasterių skaičių. Šio indekso kaita skirtingam klasterių skaičiui pavaizduota 2.2.3 paveiksle.



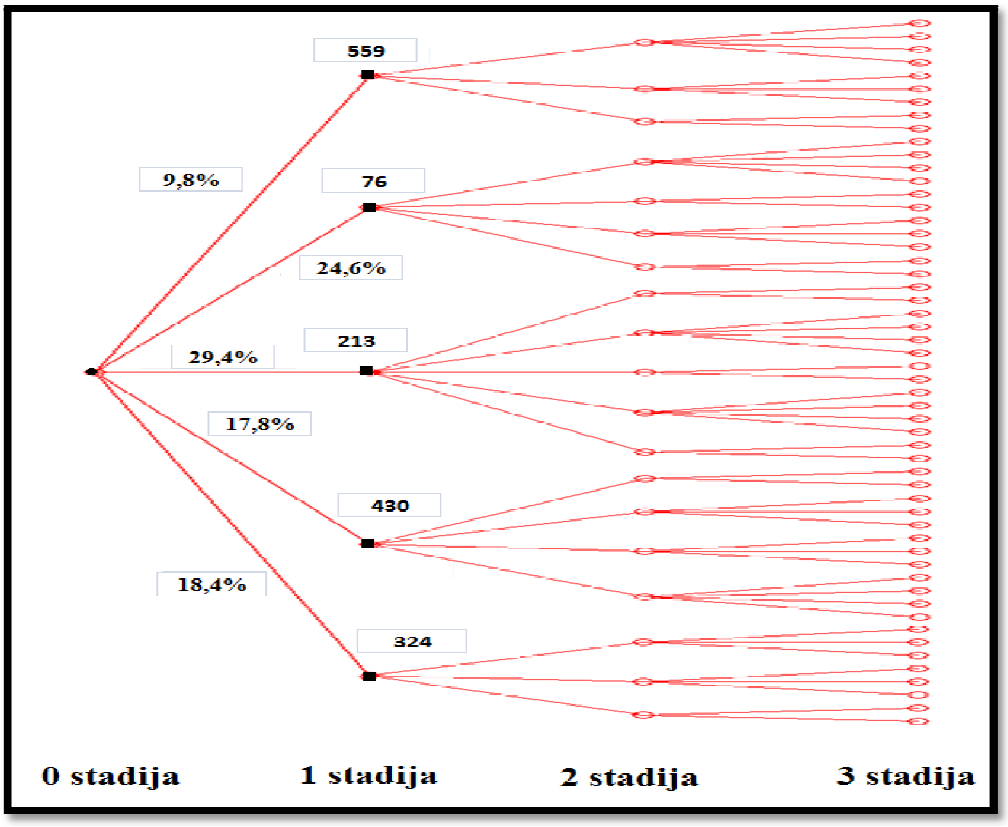
2.2.3 pav. Silueto indekso reikšmės skirtingam klasterių skaičiui

Maksimali indekso reikšmė pažymėta skrituliu. Taigi optimaliausias klasterių skaičius pirmojoje stadijoje yra 5. Optimalus dendogramos skaldymas į 5 klasterius pavaizduotas 2.2.4 paveiksle juoda horizontalia linija.



2.2.4 pav. Pirmos stadijos dendograma

Panaudojant scenarijų medžio generavimo algoritmą (1.2.11 skyrius) buvo sugeneruotas šio klasterio 5 stadijų scenarijų medis. Gautas medis pavaizduotas 2.2.5 paveiksle. Dėl medžio didelio dydžio, pavaizduotos tik pirmos trys stadijos ir pirmos stadijos mazgų reikšmės ir jų šakų tikimybės.



2.2.5 pav. Scenarijų medis pirmoms trims stadijoms.

Galiausiai patikrinta kaip kiekvienos stadijos vidutinė mazgų reikšmė atitinką realią klasterio „grid.akolegija.lt“ reikšmę.

2.2.4 lentelė. Realių ir prognozuotų reikšmių palyginimas.

	1 stadija	2 stadija	3 stadija	4 stadija	5 stadija
Vidutinė reikšmė	272,13	264,40	260,93	271,23	254,79
Reali reikšmė	422	234	105	305	235
Santykinė paklaida	55%	11%	60%	12%	8%
RMSE	99,23				
MAE	77,95				

Naudojant tokį metodą galima gauti ir kitų klasterių scenarijaus medžius.

IŠVADOS

Šiame darbe sudaryta scenarijų medžio generavimo metodika, pritaikyta GRID tinklo klasterių apkrovos prognozavimui. Metodika susideda iš duomenų atstatymo modelio, imitacinio modeliavimo ir klasterizavimo. Darbo metu gautos tokios išvados:

1. Atlikus duomenų atstatymo metodų analizę ir įvertinus metodų patikimumą buvo parinktas didžiausio tikėtinumo duomenų atstatymo metodas, naudojant tikėtinumo-maksimizavimo algoritmą. Tai leido atstatyti trūkstamus GRID klasterių apkrovos laiko eilučių duomenis.

2. Dėl didelės klasterio darbo apkrovos duomenų variacijos šiems duomenims buvo parinktas GARCH modelis, kurio adaptavimas realiems bei atstatytiems duomenims leido generuoti šių duomenų nepriklausomus scenarijus.

3. Klasterių apkrovos scenarijai buvo klasterizuojami naudojant hierarchinio klasterizavimo metodą. Tai leido sugeneruoti daugelio stadijų apkrovos scenarijų medį.

LITERATŪROS SĄRAŠAS

1. AGARWAL, P.; ALAM, M.A.; BISWAS R. *Issues, Challenges and Tools of Clustering Algorithms.* IJCSI,Vol.8, Issue3, 2011.
2. ALLISON, P. D. *Mising data*, Thousand Oaks, 2001.
3. CHANG, S-C.;, HYUNG, J. K.,*EM Algorithm.*2007.
4. ČEKAVIČIUS, V.; MURAUSKAS, G. *Statistika ir jos taikymai*. Vilnius, 2008.
5. DINDA, P.A.; O'HALLARON, D.R. *Host Load Prediction Using Linear Models,Cluster Computing.* vol. 3, pp. 265-280,2000.
6. DOMENICA, N. D.; BIRBILIS, G.; MITRA, G.; VALENTE P. *Stochastic Programming and Scenario Generation within a Simulation Framework: An Information Systems Perspective.* Technical Report. Carisma,Brunel University,UK, 2003.
7. DUPAČOVA, J., CONSIGLI, G.; WALLACE, S.W. *Scenarios for multistage stochastic programas*. Annals of operations research, 2000.
8. FISCHMAN, M.; CUMMINGS, J. M. *Multiple Imputation for Missing Data: Making the Most of What you Know*, Carnegie Mellon Univesity,2003.
9. HEITSCH, H.; ROMISCH W, Generation of Multivariate Scenario Trees to Model Stochasticity in Power Management. Humboldt-University Berlin, Institute of Mathematics
10. HOYLAND, K.; WALLACE, S., *Generating Sceanrio Trees for Multistafe Decision Problems,* Management Science,vol. 47, 2001.
11. YANG, C. Y. *Multiple imputation for missing data: concepts and new development.* SAS Institute Inc., Rockville,2007.
12. YANG, L.; I. FOSTER, SCHOPF, J.M. *Homeostatic and Tendency-based CPU Load Predictions.* Proceedings of IPDPS,2003.
13. KAUT., M; WALLACE, S. *Generating Scenario Trees for Multisafe Decision Problems.* Management Science,vol. 47,2001.
14. KOUWENBERG, R. Scenario *Generation and Stochastic Programming models for Asset Liability Management.* European Journal of Operational Research,2001.
15. LATORRE, J. M.; CERISOLA, S.; RAMOS, A. *Clustering Algorithms for Scenario Tree Generation. Application to Natural Hydro Inflows. Annals of Operations Research*. vol. 166, no. 1, pp. 355-373, 2009.

16. LIANG, J.; NAHRSTED, K.; ZHOU, Y. *Adaptive multi-resource prediction in distributed resource sharing environment.* Proceedings of the IEEE International Symposium on High Performance Distributed Computing,2002,pp 188-196
17. MITRA, L.; MITRA, G.; ROMAN, D., *Scenario generation for financial modelling:Desirable properties and case study.*CARISMA, 2009.
18. MULVEY, JM; THORLACIOUS, AE *The towers Perrin Global Capital Market Scenario Generation System.*1998.
19. PAPPALA, V.S.; ERLICH, I. *Management of Distibuted Generation Units under Stochastic Load Demands using Particle* Swarm Optimization.2007.
20. PORTER, D. *Basel II: Heralding the Rise of operational Risk.* Computer Fraud&Security,2003.
21. REYNISSON, G. M. *Asset Liability Management for Icelandic Pensio Funds- The Stochastic Programming Approach.* Thesis, University of Iceland,2012.
22. SAKALAUSKAS, V. *Duomenų analizė su Statistica*. Vilnius, 2003.
23. SCHAFER, J.L.,Analysis of incomplete multivariate data.,New York, 1997.
24. SHI, L.; GUO, L.; YANG, S.; WU, B. *A Markov Chain Based Resource prediction in Computational Grid.* Proceedings of Fourth International Conference on Frontier of Computer Science and Technology. – Shanghai.2009, 119-124.
25. SOKAL, R.R.; ROHLF, F. J.. *The comparison of dendograms by objective methods*. Taxon, 1962.
26. TRUXILO, C. *Maximum likelihood parameter estimation with incomplete data.* SAS Institute, Cary, NC,2007.
27. ZHANG, Y.; SUN, W.; INOGUCHI, Y. *CPU Load Predictions on the Computational Grid.* Proc. IEEE Sixth International Conference CLuster Computing and Grid,2006.
28. WOLSKI, R. *Dynamically Forecasting Network Performance Using the Network Weather Service.* J. Cluster Computing, vol.1,pp. 119-132,1998.

.

PRIEDAI

1 PRIEDAS. SCENARIJŲ GENERAVIMO ALGORITMAS

```
clear all

load duom2.txt   % imitaciskai sumodeliuotu 500 trajektoriju failas 72 valandu momentui
proga=duom2;


for j=1:500
prog(j)=proga(j,12);
end

pro=transpose(prog);
Y = pdist(pro,'euclidean');
Z = linkage(Y,'average');
T = cluster(Z,'maxclust',3:7);

maxT=0;
for i=1:5
s = silhouette(pro,T(:,i),'euclid');
sil(i)=mean(s);
if sil(i)>maxT
 maxT=sil(i);
 Max12=i;
end
end

Max12=Max12+2;
Q12 = cluster(Z,'maxclust',Max12);


Kiekiai=zeros(7,1);
Suma=zeros(7,1);

for i=1:500
  KKK = Q12(i);
  aaa=Kiekiai(KKK);
  Kiekiai(KKK) = aaa + 1;
end

for i=1:500
  KKK = Q12(i);
  aaa=pro(i);
  SumaElementas = Suma(KKK);
  Suma(KKK) = aaa + SumaElementas;
end

for i=1:Max12
  if Kiekiai(i) > 0
    Vidurkis1stadija(i) = Suma(i) / Kiekiai(i);
  end
end

for i=1:Max12
  if Kiekiai(i) > 0
    Tikimybes1stadija(i) = Kiekiai(i) / size(pro,1);
  end
end


 for A=1:Max12
    %  ..................................2stadija

   clear PRO24
   clear MAS24
   clear sil2
   clear kiekis2

   k=1;
   for i=1:500
```

```
 if (Q12(i)==A)
  PRO24(k)=proga(i,24);
  ind24(k,A)=i;
  k=k+1;
 end
end

MAS24=transpose(PRO24);
if size(MAS24,1)>1


  Y = pdist(MAS24,'euclidean');
  Z = linkage(Y,'average');
  T2 = cluster(Z,'maxclust',3:5);

  maxT2=0;
  for i=1:3
   s = silhouette(MAS24,T2(:,i),'euclid');
   sil2(i)=mean(s);
   if sil2(i)>maxT2
    maxT2=sil2(i);
    MaxInd24=i;
   end
  end

  Max24=MaxInd24+2;
  Q24 = cluster(Z,'maxclust',Max24);


  maxas24=0;
      for d=1:size(Q24,1)
         if Q24(d) > maxas24
            maxas24=Q24(d);
         end
      end
      Max24=maxas24;

  Kiekiai2=zeros(7,1);
  Suma2=zeros(7,1);

  for i=1:size(MAS24,1)
   KKK2 = Q24(i);
   aaa2=Kiekiai2(KKK2);
   Kiekiai2(KKK2) = aaa2 + 1;
  end

  for i=1:size(MAS24,1)
   KKK2 = Q24(i);
   aaa2=MAS24(i);
   SumaElementas2 = Suma2(KKK2);
   Suma2(KKK2) = aaa2 + SumaElementas2;
  end
else
  Kiekiai2=zeros(7,1);
  Suma2=zeros(7,1);
  Q24=1;
  MAX24=1;
end

for i=1:Max24
 if Kiekiai2(i) > 0
  Vidurkis2stadija(A,i) = Suma2(i) / Kiekiai2(i);
 end
end

for i=1:Max24
 if Kiekiai2(i) > 0
  Tikimybes2stadija(A,i) = Kiekiai2(i) / size(MAS24,1);
 end
end

% .....................................3stadijos parametras
```

```
for B=1:Max24

 clear PRO36
 clear MAS36
 clear sil3
 clear kiekis3

 benuliuku=0;
 for i=1:size(MAS24,1)
  if MAS24(i) > 0
    benuliuku=benuliuku+1;
  end
 end

 k=1;
 for i=1:benuliuku
  if (Q24(i)==B)
    PRO36(k)=proga(ind24(i,A),36);
    ind36(k,A,B)=ind24(i,A);
    k=k+1;
  end
 end

 MAS36=transpose(PRO36);

 if  size(MAS36,1)>1

    Y = pdist(MAS36,'euclidean');
    Z = linkage(Y,'average');
    T3test = cluster(Z,'maxclust',2:4);

    maxT3=0;
    for i=1:3
      s = silhouette(MAS36,T3test(:,i),'euclid');
      sil3(i)=mean(s);
      if sil3(i)>maxT3
        maxT3=sil3(i);
        MaxInd36=i;
      end
    end

    MaxInd36=MaxInd36+1;
    Q36 = cluster(Z,'maxclust',MaxInd36);


    maxas36=0;
       for d=1:size(Q36,1)
         if Q36(d) > maxas36
             maxas36=Q36(d);
          end
       end
     MaxInd36=maxas36;

    Kiekiai3=zeros(7,1);
    Suma3=zeros(7,1);

    for i=1:size(MAS36,1)
      KKK3 = Q36(i);
      aaa3=Kiekiai3(KKK3);
      Kiekiai3(KKK3) = aaa3 + 1;
    end

    for i=1:size(MAS36,1)
      KKK3 = Q36(i);
      aaa3=MAS36(i);
      SumaElementas3 = Suma3(KKK3);
      Suma3(KKK3) = aaa3 + SumaElementas3;
    end

    for i=1:MaxInd36
      if Kiekiai3(i) > 0
      Vidurkis3stadija(A,B,i) = Suma3(i) / Kiekiai3(i);
              end
```

```
    end

    for i=1:MaxInd36
     if Kiekiai3(i) > 0
     Tikimybes3stadija(A,B,i) = Kiekiai3(i) / size(MAS36,1);
     end
    end

  else

     MAS36=PRO36;
     Q36=1;
     Vidurkis3stadija(A,B,1) =MAS36;
     Tikimybes3stadija(A,B,1) =1;
     MaxInd36=1;

  end

% 4 .....stadijos
    for C=1:MaxInd36


    clear PRO48
    clear MAS48
    clear sil4
    clear kiekis4

    benuliuku=0;
    for i=1:size(MAS36,1)
     if MAS36(i) > 0
       benuliuku=benuliuku+1;
      end
    end


    k=1;
    for i=1:benuliuku
     if (Q36(i)==C)
       PRO48(k)=proga(ind36(i,A,B),48);
       ind48(k,A,B,C)=ind36(i,A,B);
       k=k+1;
      end
    end

    if size(PRO48,2)>1
       MAS48=transpose(PRO48);
       Y = pdist(MAS48,'euclidean');
       Z = linkage(Y,'average');
       T4test = cluster(Z,'maxclust',2:4);

       maxT4=0;
       for i=1:3
        s = silhouette(MAS48,T4test(:,i),'euclid');
        sil4(i)=mean(s);
        if sil4(i)>maxT4
          maxT4=sil4(i);
          MaxInd48=i;
        end
       end


       MaxInd48=MaxInd48+1;
       Q48 = cluster(Z,'maxclust',MaxInd36);

       maxas48=0;
       for d=1:size(Q48,1)
          if Q48(d) > maxas48
             maxas48=Q48(d);
          end
       end
       MaxInd48=maxas48;

       Kiekiai4=zeros(7,1);
```

```
    Suma4=zeros(7,1);

    for i=1:size(MAS48,1)
      KKK4 = Q48(i);
      aaa4=Kiekiai4(KKK4);
      Kiekiai4(KKK4) = aaa4 + 1;

    end

  for i=1:size(MAS48,1)
    KKK4 = Q48(i);
    aaa4=MAS48(i);
    SumaElementas4 = Suma4(KKK4);
    Suma4(KKK4) = aaa4 + SumaElementas4;
  end

  for i=1:MaxInd48
    if Kiekiai4(i) > 0
    Vidurkis4stadija(A,B,C,i) = Suma4(i) / Kiekiai4(i);
    end
  end

  for i=1:MaxInd48
    if Kiekiai4(i) > 0
      Tikimybes4stadija(A,B,C,i) = Kiekiai4(i) / size(MAS48,1);
      end
  end

  else
    MAS48=PRO48;
    Q48=1;
    Vidurkis4stadija(A,B,C,1) =MAS48;
    Tikimybes4stadija(A,B,C,1) =1;
    MaxInd48=1;

  end


%     5stadija.....................................................................
  for D=1:MaxInd48

    clear PRO60
    clear MAS60
    clear sil5
    clear kiekis5

    benuliuku=0;
    for i=1:size(MAS48,1)
      if MAS48(i) > 0
        benuliuku=benuliuku+1;
      end
    end


    k=1;
    for i=1:benuliuku
      if (Q48(i)==D)
        PRO60(k)=proga(ind48(i,A,B,C),60);
        k=k+1;
      end
    end


    if size(PRO60,2)>1
      MAS60=transpose(PRO60);
    else
      MAS60=PRO60 ;
    end

    if size(MAS60,1)>1
      Y = pdist(MAS60,'euclidean');
      Z = linkage(Y,'average');
      T5test = cluster(Z,'maxclust',2:3);
```

```
          maxT5=0;
          for i=1:2
            s = silhouette(MAS60,T5test(:,i),'euclid');
            sil5(i)=mean(s);
            if sil5(i)>maxT5
              maxT5=sil5(i);
              MaxInd60=i;
            end
          end


          MaxInd60=MaxInd60+1;
          Q60 = cluster(Z,'maxclust',MaxInd48);

          maxas60=0;
          for d=1:size(Q60,1)
             if Q60(d) > maxas60
                maxas60=Q60(d);
             end
          end
          MaxInd60=maxas60;

          Kiekiai5=zeros(7,1);
          Suma5=zeros(7,1);

          for i=1:size(MAS60,1)
            KKK5 = Q60(i);
            aaa5=Kiekiai5(KKK5);
            Kiekiai5(KKK5) = aaa5 + 1;
          end


         for i=1:size(MAS60,1)
           KKK5 = Q60(i);
           aaa5=MAS60(i);
           SumaElementas5 = Suma5(KKK5);
           Suma5(KKK5) = aaa5 + SumaElementas5;
         end

         for i=1:MaxInd60
           if Kiekiai5(i) > 0
           Vidurkis5stadija(A,B,C,D,i) = Suma5(i) / Kiekiai5(i);
           end
         end

         for i=1:MaxInd60
           if Kiekiai5(i) > 0
             Tikimybes5stadija(A,B,C,D,i) = Kiekiai5(i) / size(MAS60,1);
             end
         end

         else

               Q60=1;
               MAS60=PRO60;
               Vidurkis5stadija(A,B,C,D,1) =MAS60;
               Tikimybes5stadija(A,B,C,D,1) =1;
               MaxInd60=1;

         end

       end
      end
  end
end

%  isrinkimo etapas

 number=1;
    for  A=1:size(Vidurkis1stadija,2)
      for B=1:1:size(Vidurkis2stadija,2)
        for C=1:size(Vidurkis3stadija,3)
```

```
        for D=1:size(Vidurkis4stadija,4)
           for E=1:size(Vidurkis5stadija,5)
              if Tikimybes5stadija(A,B,C,D,E)>0

                 Scenarijai(1,number)=A;
                 Scenarijai(2,number)=B;
                 Scenarijai(3,number)=C;
                 Scenarijai(4,number)=D;
                 Scenarijai(5,number)=E;

                 Scenarijai(6,number)=Tikimybes1stadija(A);
                 Scenarijai(7,number)=Tikimybes2stadija(A,B);
                 Scenarijai(8,number)=Tikimybes3stadija(A,B,C);
                 Scenarijai(9,number)=Tikimybes4stadija(A,B,C,D);
                 Scenarijai(10,number)=Tikimybes5stadija(A,B,C,D,E);

                 Scenarijai(11,number)=Vidurkis1stadija(A);
                 Scenarijai(12,number)=Vidurkis2stadija(A,B);
                 Scenarijai(13,number)=Vidurkis3stadija(A,B,C);
                 Scenarijai(14,number)=Vidurkis4stadija(A,B,C,D);
                 Scenarijai(15,number)=Vidurkis5stadija(A,B,C,D,E);

                 number=number+1;

              end
           end
        end
     end
  end
end
```

2 PRIEDAS. DUOMENŲ ATKŪRIMO SAS KODAS NAUDOJANT DIDŽIAUSIO TIKĖTINUMO METODĄ

```
filename fail 'C:\duom1.txt';

data duomenys;
  infile fail dlm=',';
  input kl1 kl2 kl3 kl4 kl5;
  k1=sqrt(kl1);
  k2=sqrt(kl2);
  k3=log(kl3);
  k4=log(kl4);
  k5=log(kl5);

  run;
proc mi data=duomenys nimpute=0 ;

        EM out=outras;
    em itprint outem=outem;
        var k1 k2 k3 k4 k5 ;

   run;

data transform;
set outras (keep= k1 k2 k3 k4 k5 );
s1 = k1*k1;
s2 = k2*k2;
s3 = exp(k3);
s4 = exp(k4);
s5 = exp(k5);

proc print data=transform;
  title 'Em estimates' ;
  run;
data atstatytiduom;
set transform(keep= s1 s2 s3 s4 s5 );
proc print data=atstatytiduom;
run;
```