



**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS**

**Džeineta Vailionytė**

**DINAMINIO LAIKO SKALĖS IŠKRAIPYMO METODO  
TAIKYMAS FINANSINĖMS LAIKO EILUTĖMS TIRTI**

Baigiamasis magistro projektas

**Vadovas**

doc. dr. Kristina Šutienė

**KAUNAS, 2016**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS**

**DINAMINIO LAIKO SKALĖS IŠKRAIPYMO METODO**  
**TAIKYMAS FINANSINĖMS LAIKO EILUTĖMS TIRTI**

Baigiamasis magistro projektas

**Taikomoji matematika (621G10003)**

**Vadovas**

doc.dr. Kristina Šutienė

2016 05 30

**Rezendentas**

doc.dr. Liepa Bikulčienė

2016 05 30

**Projektą atliko**

Džeineta Vailionytė

2016 05 30

**KAUNAS, 2016**

## TURINYS

Įvadas.....	7
1. Literatūros apžvalga.....	8
1.1. Dinaminio laiko skalės iškraipymo metodo plėtojimas .....	8
1.2. Finansinių laiko eilučių tyrimo atvejai .....	13
1.3. Dinaminio laiko skalės iškraipymo algoritmo taikymas finansinėms laiko eilutėms tirti ..	14
1.4. Literatūros šaltinių analizės apibendrinimas.....	16
2. Tyrimų metodai .....	17
2.1. Atstumų metrikos .....	17
2.2. Dinaminis laiko skalės iškraipymas .....	18
2.2.1. Sukauptų skirtingumo įverčių (atstumų) matrica.....	19
2.2.2. Trajektorija (sutapatinimo kelias).....	20
2.2.3. Globalaus išlyginimo apribojimas .....	22
2.2.4. Kaštų ir sutapatinimo kelio statistikos .....	23
2.3. Spirmeno koreliacijos koeficientas .....	26
2.4. Naudota programinė įranga .....	26
2.4.1. Naudotų programų bibliotekos ir procedūros .....	27
2.5. Apibendrinimas .....	28
3. Tiriamaoji dalis .....	29
3.1. Duomenys .....	30
3.2. Vienmatis atvejis .....	30
3.2.1. Sintetinės laiko eilutės tyrimas: $\sin(x/4\pi)$ eilutė.....	30
3.2.2. Realios laiko eilutės tyrimas: OMX indeksas.....	31
3.3. Dvimatis atvejis.....	37
3.3.1. Sintetinių laiko eilučių: $\sin(x/8\pi)$ eilutės ir $\sin((x/8\pi) + 4)$ eilutės su triukšmu tyrimas .....	37
3.3.2. Realių laiko eilučių tyrimas: OMX Vilnius ir OMX Ryga kainų indeksai .....	43
3.4. Daugiamatis atvejis.....	47
Diskusijos .....	53
Išvados.....	54
Literatūros sąrašas .....	55
1 priedas. OMX gražos indekso istoriniai kainų elgsenos pasikartojimai.....	57
2 priedas. OMX gražos indekso pradiniai duomenys klasterizavimui.....	58
3 priedas. OMX Vilnius ir OMX Ryga trajektorijų grafikai .....	59
4 priedas. OMX Vilnius ir OMX Ryga dinaminio laiko skalės iškraipymo grafikai.....	60
5 priedas. OMX Vilnius ir OMX Ryga tankio grafikai .....	63
6 priedas. Programos tekstas .....	67



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Matematikos ir gamtos mokslų fakultetas

Džeineta Vailionytė

Taikomoji matematika, 621G10003

„Dinaminio laiko skalės iškraipymo metodo taikymas finansinėms laiko eilutėms tirti“

**AKADEMINIO SAŽININGUMO DEKLARACIJA**

2016 m. gegužės 30d.

Kaunas

Patvirtinu, kad mano Džeinetos Vailionytės baigiamasis projektas tema „Dinaminio laiko skalės iškraipymo metodo taikymas finansinėms laiko eilutėms tirti“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

Vailionytė, Džeineta. Dinaminio laiko skalės iškraipymo metodo taikymas finansinėms laiko eilutėms tirti. Magistro baigiamasis projektas / vadovė doc. dr. Kristina Šutienė; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Mokslo kryptis ir sritis: Fiziniai mokslai, matematika.

Reikšminiai žodžiai: dinaminis laiko skalės iškraipymas, laiko eilutės, finansinių laiko eilučių tyrimas.

Kaunas, 2016. -73 psl.

## **SANTRAUKA**

Darbe nagrinėjamas dinaminio laiko skalės iškraipymo algoritmas ir jo taikymo galimybės finansinėms laiko eilutėms tirti. Atlikta publikuotų mokslinių tyrimų apžvalga parodė, kad šis algoritmas finansiniuose uždaviniuose nėra plačiai taikomas. Šiuolaikiniame pasaulyje finansinės rinkos tampa vis sunkiau analizuojamos klasikiniiais metodais, todėl ieškoma naujų metodų arba taikomi metodai iš kitų sričių, kurie leistų tirti finansines laiko eilutes ir jose atpažinti tam tikrus dėsningumus. Todėl šiame darbe tiriama tikėtina dinaminio laiko skalės algoritmo taikymo pridedamoji vertė analizuojant finansines laiko eilutes, ypač kai joms būdingas nevienodas kitimo greitis ar skiriasi laiko skalės.

Siekiant užtikrinti korektišką algoritmo veikimą reikėjo patikrinti ar sudarytos trumpiausios trajektorijos tenkina tris pagrindinius kriterijus, taip pat priklausomai nuo turimų duomenų tinkamai parinkti lokalius arba globalius apribojimus. Atlikus tris tyrimo atvejus buvo pastebėta, kad paprastiems asimetriniams ir simetriniams apribojimams būdingas vieno taško į vieną tašką sutapatinimas, o parinkus tam tikrą slopinimo koeficientą gaunamas vieno taško į keletą taškų sujungimas. Viename iš praktinių taikymų buvo surasti panašios formos OMX gražos indekso pasikartojimai istoriniuose 2010 – 2013 metų laikotarpio duomenyse. Taip pat šio tyrimo atveju buvo suklasterizuoti 5 panašūs istoriniai laiko eilutės fragmentai, kurie buvo padalinti į 3 pagrindines grupes pagal jiems būdingą laiko eilutės formą. Daugiamačių laiko eilučių atveju, panaudojant dinaminį laiko skalės algoritmą, ištyrėme 2014 – 2015 metų OMX kainų indeksus ir nustatėme, kad OMX Baltic kainų indekso laiko eilutės forma yra panašiausia su OMX Ryga kainų indekso eilutės forma.

Remiantis baigiamojo projekto teorinės dalies analize ir gautais tyrimų rezultatais, šio darbo pabaigoje pateikiamos išvados ir diskusijos.

Vailionytė, Džeineta. Application of Dynamic Time Warping to Analyze Financial Time Series. Master's thesis in applied mathematics / supervisor assoc.prof. Kristina Štutienė; The Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Research area and field: Physical sciences, Mathematics.

Key words: dynamic time warping, time series, financial time series analysis.

Kaunas, 2016. –73 psl.

## **SUMMARY**

In this paper dynamic time warping algorithm is discussed and its application opportunities for financial time series. The overview of published researches has shown that this algorithm in financial studies is not widely applicable. In this world the financial markets are becoming increasingly difficult to analyze using traditional methods, so there is a need of developing new techniques or using methods from the other areas that would let to investigate the financial time series and recognize their certain shape patterns. Therefore, in this work we investigate the application of dynamic time warping algorithm and its added value to analyse financial time series, especially when they are characterized in different speed or different time scales.

In order to ensure the correct operation of the algorithm it was needed to verify if established path satisfied the main three criteria, also properly select local or global constraints depending from the data structure. After the three cases of experiments, we observed that the ordinary assymetric and symmetrical restrictions are characterized by one point to one mapping, selection of a slope coefficient gave us one point to many points mapping. In the one practical application we found similar form of OMX return index behavior in the historical 2010 – 2013 year period. Also in this study case, 5 similar patterns of historical times series data were clustered and divided into 3 main groups according to their inherent time series pattern. Multidimensional time series case, where OMX price indices in the period of 2014 – 2015 years have benn analysed, has shown that OMX Baltic index time series is most similar to the price index of OMX Ryga time series.

According to the final work of the theoretical analysis and the research results, conclusions and discussions are provided at the end of this work.

## ĮVADAS

Laiko eilučių panašumo tyrimai yra ir buvo vienas iš pagrindinių mokslinių tyrimų laiko eilučių duomenų gavyboje daugiau nei vieną dešimtmetį. Egzistuoja dviejų tipų panašumai: formos (angl. *shape – based*) ir struktūros (angl. *structure based*). Tyrėjas nustato dviejų rinkinių panašumą lygindamas jų lokalius modelius, kadangi pastarieji lemia panašumus norint apibūdinti rinkinių globalias struktūras. Daugiausia moksliniuose darbuose yra susitelkiama į panašios formos ieškojimą. Turint dvi sekas A ir B panašumo forma apibūdina kaip tiksliai šios dvi sekos yra panašios tam tikruose lokaliuose taškuose. Duomenų tyryboje plačiausiai žinomas atstumo matas yra Euklido metrika. Tačiau ji nėra tinkama duomenims, kurių laiko ašys nėra vienodo ilgio arba kai jos būna paslinktos. Dirbant su tokiais duomenimis patogiau ir tiksliau yra naudoti dinaminį laiko skalės iškraipymą. Panaudojus dinaminio programavimo techniką randamas geriausias optimalus atstumas tarp dviejų laiko sekų.

Magistriniame darbe nagrinėjamas dinaminio laiko skalės iškraipymo algoritmas, analizuojami teoriniai bei praktiniai aspektai, pateikiamas pritaikymas finansinių laiko eilučių panašumui tirti. Nagrinėjami ne tik Lietuvos, bet ir kitų šalių kainų indeksai. Kiekvienu tiriamu atveju siekiama gauti patikimus rezultatus, todėl dinaminio laiko skalės iškraipymo algoritmo veikimas iš pradžių patikrinamas sintetinei laiko eilutei. Pagrindinis magistrinio darbo tyrimų objektas – skirtingose laiko skalėse išmatuotos arba nevienodo greičio laiko eilutės.

Dinaminio laiko skalės iškraipymo algoritmo taikymas finansinėms laiko eilutėms yra tik neseniai pradėtas nagrinėti, todėl atliktų tyrimų nėra daug. Dažniausiai praktikoje finansinės laiko eilutės yra nagrinėjamos klasikiniais metodais: ieškant stacionarumo, koreliacijos koeficiento, trendo, sezoninių svyravimų ar atliekant prognozavimą. Tačiau norint išspręsti šiuos uždavinius reikia vienaip ar kitaip pertvarkyti turimas laiko eilutes. Dinaminio laiko skalės iškraipymo algoritmo vienas iš pagrindinių privalumų yra skirtingo greičio – laiko sekų sutapatinimas. Be to, jo veikimui nedaro įtakos įvairūs vėlinimai, sezoniniai svyravimai ar trendai.

Pirmoje darbo dalyje apžvelgiamas dinaminio laiko skalės iškraipymo algoritmo aktualumas ir tyrimų rezultatai visame pasaulyje. Aprašomos skirtingos pritaikymo sritys bei išskiriamas ribotas algoritmo taikymas finansiniuose tyrimuose. Antroje dalyje aprašomi pagrindiniai dinaminio laiko skalės iškraipymo metodo teoriniai aspektai. Trečioje dalyje pateikiami atlikti tyrimai vienmačiu, dvimačiu ir daugiamačiu laiko eilučių atvejais. Taip pat ieškoma formos panašumų tarp nagrinėjamų indeksų kainų eilučių.

Dalis baigiamojo magistro projekto rezultatų buvo paskelbti konferencijoje „Matematika ir matematikos dėstymas – 2016“.

## 1. LITERATŪROS APŽVALGA

Dažniausiai dinaminis laiko skalės iškraipymo (DLSI) metodas taikomas žodžių atpažinimo sistemose. Šiuo atveju dėl nevienodo kalbos greičio jis padeda išvengti vienkartinį laiko nesutapimų. Tačiau be kalbos atpažinimo sistemų, šis algoritmas taip pat plačiai taikomas tokiose srityse kaip duomenų gavyba, gestų kalba, robotikos anomalijos atpažinimai, gamyba, finansai, medicina ir kitos. Pirminio algoritmo variante buvo siekiama susidoroti su specifinėmis problemomis, atsiradusioms dėl laiko sekų nesutapimo. Laikui bėgant algoritmas buvo vis tobulinamas, taip pat buvo ieškomos ir kuriamos įvairios jo modifikacijos.

### 1.1. DINAMINIO LAIKO SKALĖS IŠKRAIPYMO METODO PLĖTOJIMAS

Nomikos ir MacGregor (1994) paskelbė publikaciją apie šio algoritmo pritaikymą periodinių procesų stebėjimuose [1]. Li ir kiti (2004) sujungė dinaminį laiko skalės iškraipymo algoritmą su dekompozicija, kuris sinchronizuoja tam tikrų periodų sutapatinimo kelius. Panaudojant DLSI originalūs signalai buvo išskaidyti į aproksimacijas, išanalizuoti skirtingose skalėse ir sutapatinti kiekvienoje iš šių skalių atskirai. Sujungti panašūs signalai buvo panaudojami norint išgauti sinchronizuotą signalą [2].

Rodriguez ir Alonso (2004) pristatė sprendimo medžių skirtų eilučių klasifikavimui kūrimo metodą. Sudaryti dviejų tipų medžiai, vienas iš jų paremtas intervaliniu pagrindu, kitas – dinaminio laiko skalės iškraipymu. Kiekvienas vidinis medžio mazgas apskaičiuoja funkciją (vidurkį, nuokrypį) intervale arba atstumą tarp pavyzdžio klasifikavimui ir etalono (angl. *reference*). Palyginus eksperimento rezultatus su pradiniais duomenimis kaip ir tikėtasi buvo gauta, kad medžio suprantamumas (angl. *comprehensibility*) turi savo kainą. Taip pat sprendimo medžio tikslumas priklauso nuo to, kad vienas dydis gaunamas padidinus kitą. Šis autorių pristatytas metodas yra adekvatus tuo atveju, kai pagrindiniu tikslu laikoma aiški sudaryto sprendimų medžio struktūra ir lengvas interpretavimas [3].

Li ir kiti (2015) pasiūlė naują priežastingumo analizės įrankį, paremtą dinaminio laiko skalės iškraipymu su dinaminio latentinių kintamųjų modeliu. Iš pradžių dinaminių kintamųjų modelis aptinka klaidą, po to daugiakryptis RBC (angl. *reconstruction based contributions*) metodas pateikia diagnozę apie pirminius kintamuosius susijusius su klaida (angl. *fault*). Išskleidus pirminių klaidų duomenų eilutes pagal trendo signalą, susiję kintamieji suklasterizuojami į grupes pagal  $k$  vidurkių metodą. Paskutiniame žingsnyje priežastingumo kryptis buvo parodyta naujai sukurtu indeksu, kuris remiasi dinaminio laiko skalės iškraipymo



algoritmo pagrindu. Taigi šis tyrimas parodė, kad autorių sukurtas DLSI indeksas gali būti sėkmingai pritaikomas nestacionarioms klaidoms aptikti [4].

Jorge (2014) pateikė dinaminio laiko skalės iškraipymo algoritmo pritaikymą savo programinei aplinkai „Pure Data“. Autorius pasinaudojo DLSI savybe palyginti ir atpažinti duomenis, kurie gali skirtis netiesiškai laike. Tokiu būdu šiek tiek patobulino pačią algoritmo idėją žiūrint iš dviejų perspektyvų: vykdymo ir adekvatumo atpažįstant gestus. Veiklos testai parodė, kad sukurtas būdas yra tinkamas naudoti realiame laike, kai algoritmo efektyvumas tampa vienu iš svarbiausių prioritetų. Šią sukurtą aplikaciją galima naudoti daugelyje platformų, kuriose aplinka „Pure Data“ yra prieinama [5].

Taylor, Zhou, Roupail ir Porter (2015) pritaikė dinaminio laiko skalės iškraipymo algoritmą trajektoriniams duomenims, t.y. sukalibruojamas tam tikras imitacinis modelis pagal sintetinius vairuotojo trajektorijos duomenis. Pasinaudojant DLSI unikaliomis savybėmis siekiama iširti vairuotojo heterogeniškumą pagal sekamos mašinos elgesį, taip pat įvairias vairuotojo situacijas priklausančias nuo elgesio kelionės metu, paremtas reagavimo laiku ir buvimu kamščiuose. Bendru atveju DLSI algoritmas yra pajėgus išspręsti tokius specifinius skaičiavimus. Todėl įrodyta, kad DLSI yra tinkamas metodas norint apdoroti didelius transporto trajektorijų duomenis. Tačiau reikalaujamas pastebimas duomenų apkarpymas siekiant gauti suprantamus ir aiškius rezultatus, kai dirbama su didelės raiškos transporto trajektorijų duomenimis [6].

Bhanu ir Zhou (2004) pristatė kreivę pagrįsto žmogaus profilio sutapatavimo metodą, kurio siekis yra panaudoti visus turimus veido duomenis. Erdvės skalės filtravimas buvo panaudojamas norint išlyginti profilį ir apskaičiuoti išfiltruotą profilio kreivę. Panaudojant kreivės reikšmę, atskaitos taškus, žmogaus kaukolės sankirtą tarp priekinio kaulo ir dviejų nosies kaulų (lot. *nasion*) bei kaklą galima greitai ir patikimai išgauti paprastą atpažinimo būdą. Dinaminis laiko skalės iškraipymo metodas su apskaičiuota kreivės reikšme buvo panaudotas atpažinti veido profilio proporcijas nuo žmogaus kaukolės sankirtos tarp priekinio kaulo ir dviejų nosies kaulų iki kaklo. Šis eksperimentas buvo atliekamas su dviejų profilių duomenimis. Buvo pastebėta, kad panaudojus kreivės reikšmę profilio atpažinimui gauti rezultatai yra daug žadantys. Netgi veido profiliui su akivaizdžia nekokybiška raiška, t.y. skaičiai ir atskaitos taškų vieta yra visiškai skirtingo ilgio, metodas veikė taip pat lanksčiai ir korektiškai. Nors metodo veikimas šiek tiek sulėtėja dėl jautrumo turimų duomenų išskirtims, tačiau jis vis tiek veikia sklandžiai kaip ir normaliems duomenims [7].

Celebi, Aydin, Temiz ir Arici (2013) pristatė svertinį DLSI algoritmą naudojantį padidinto svorio atstumus kaštų skaičiavime. Tyrimas skirtas žmogaus veiksmų atpažinimui panaudojant jo skeleto duomenis. Svoriai parenkami siekiant maksimaliai padidinti DLSI algoritmu pagrįstą diskriminantinį santykį. Patys svoriai išgaunami iš parametrinio modelio, kuris priklauso nuo to kaip aktyviai naudojami sujungimai gestų rinkinyje. Taigi modelio parametrai yra optimizuojami didinant diskriminantinį santykį. Tokiu būdu vienu sujungimų svoriai padidinami, kitų sumažinami tam, kad padidinti variaciją išorėje tarp grupių ir sumažinti variaciją jų viduje. Pavyzdžiui, gestų grupėje nesusiję sujungimai (dalys neįtrauktos į gestų grupę) prisideda prie DLSI kaštų mažesniu mastu, bet tuo pačiu padidina variaciją grupių viduje. Sistemos veikimas pradamas nuo kūno sujungimo bruožų išskyrimo iš skeleto duomenų. Duomenys susideda iš šešių sujungimo pozicijų, kurios yra kairė ir dešinė rankos, riešai ir kulkšnys. Pastebėta, kad mokomajame duomenų rinkinyje, kuris turi gana skirtingus judesio modelius, yra reikalaujama panaudoti visus šešis sujungimus arba tik jų poaibius. Šios savybės paimtos iš skeleto griaučių yra kaupiamos bėgant laikui ir panaudojamas gestų sutapatinimui su anksčiau įrašytomis sekomis. Sutapatinimas vyksta priskiriant įvesties eilutę užklausiai su minimaliais DLSI metodo kaštais. Dinaminis laiko skalės iškraipymo algoritmas iškraipo dvi sekas padidindamas arba sumažindamas jų sekų greičius laike [8].

Lei ir Sun (2007) tyrinėjo dinaminio laiko skalės iškraipymo branduolį ir kartu užbaigė teorinį įrodymą apie sąryšį tarp teigiamai apibrėžto simetrinio branduolio (angl. *PDS*) ir atkuriamojo Hilbert erdvės branduolio (angl. *RKHS*). DLSI metodas yra pranašesnis už Euklido atstumą net ir taikant branduolio metodą, nors dauguma realizacinių pavyzdžių pateikia, kad DLSI nėra teigiamai apibrėžtas ir simetrinis, kartu nepriimtinas atraminių vektorių metodui. Tačiau teorine prasme gana mažai žinoma, kodėl jis negali būti PDS. Analizė parodė, kad visos Hilberto metrikos gali būti paversto į Gauso formos PDS branduolį. Tačiau atvirkščiai to padaryti negalima. Taip pat autoriai įrodė, kad DLSI branduolys visgi nėra teigiamai apibrėžtas ir simetrinis. Šis įrodymas gali leisti padaryti išvadą, kad elastingas sutapatinimo atstumas nėra tinkamas konstruoti PDS branduolį. Eksperimento metu taip pat buvo parodyta, kad atraminių vektorių klasifikatoriaus metodu palyginus RBF branduolį ir DLSI branduolį, Euklido metrika yra geresnė. Nors elastingas sutapatinimo dydis yra pranašesnis paprastoje įėjimo erdvėje, tačiau reikėtų būti atsargiems naudojant tokius dydžius branduolio metoduose [9].

Forestier ir kiti (2012) ištyrė chirurginių procesų modelių automatinę klasifikaciją remiantis dinaminio laiko skalės iškraipymo algoritmu. Dėl vis atsinaujinančių kompiuterių intervencijos

kūrimo sistemų chirurginių procesų modeliai yra labiausiai kylanti koncepcija. Jie naudojami analizuojant ir priskiriant chirurgines intervencijas. Šie modeliai formaliai vaizduoja simboliškai struktūrizuotus chirurginių intervencijų aprašymus panaudojant iš anksto apibrėžtą reikalaujamą detalumo lygį ir priskirtą terminologiją. Šiame kontekste vienas iš pagrindinių iššūkių yra naujų metrikų sukūrimas palyginimui ir chirurginių procesų kūrimo įvertinimui. Koreliacija tarp šių metrikų ir prieš tai stebimų duomenų yra dažniausiai panaudojama klasifikuojant operacijas ir išryškinant specifinę informaciją apie pačią operaciją ir jos atlikėją. Taigi DLSI algoritmas naudojamas apskaičiuoti panašumą tarp dviejų skirtingų chirurginių procesų veiklų per pačią operaciją ir jų seką, t.y. siekiama sumažinti laiko skirtumus tarp kintamųjų. Eksperimentas buvo vykdomas su 24 juosmens disko išvaržos operacijomis siekiant suskirstyti chirurgus ir jų kompetencijos lygius pagal išankstinę chirurginių procesų klasifikaciją. Klasifikavimo su mokytoju ir be mokytojo rezultatai parodė, kad šis klasifikavimas galėjo būti automatizuotas. Grupės buvo identifikuotos remiantis chirurgų kompetencija/ patirtimi (vyresnis ar jaunesnis). Taigi DLSI algoritmo sukurta panašumo metrika gali išskirti chirurginių procesų grupes, o koreliacija tarp šių grupių ir iš anksto stebėtų duomenų leido išryškinti tam tikrą specifinę informaciją [10].

Pierobon, Marcon, Sarti ir Tubaro (2005) pateikė žmogaus veiksmų klasterizavimo metodą, paremtą 3D kūno atvaizdavimą iš tūrinių koordinačių. Požymiai atvaizduojantys kūno pozas buvo išgauti tiesiogiai iš 3D duomenų. Todėl sisteminė prigimtis yra nejautri priklausomybės požičiūriū, judesio dviprasmybei ir savarankiškam užpildymui. Nekintamos žmogaus kūno formos apibūdinimas yra gaunamas siekiant užfiksuoti tik laikysenai priklausančias charakteristikas. Nepaisant galimų perkėlimų, orientacijos, mastelio nustatymo ir kūno dydžio. Sudaryto rėmo aprašymai sugeneruojami iš gestų sekos, tuomet laikomi kartu matricose. Po to atliekamas veiksmų matricos grupavimas ir kol skaičiuojamos atstumų matricos paremtos DLSI, gaunama nepriklausomybė nuo galimai laikinų netiesinių iškraipymų tarp daugybės skirtingų to pačio gesto atvejų. Taigi tyrimas parodė, kad šios sistemos pagrįsto formos apibūdinimu ne tik atpažįsta tam tikras padėtis, bet ir gali būti interpretuotos dinaminiam procese. Atliktas imitacinis modeliavimas parodė, kad autorių pasiūlytas metodas turi galimybę klasifikuoti skirtingus veiksmus. Dar daugiau, algoritmas gali būti sugretintas su optimizavimo procedūra ir naudojamas realaus laiko veiksmams [11].

Dai, Wan ir Jing (2014) pasiūlė stebėjimo metodą statistinio proceso kontrolei (SPK), paremtą DLSI algoritmu. Tradicinis SPK negali būti tiesiogiai pritaikomas realaus laiko

duomenims, kadangi stebimi dydžiai nėra vienodo ilgio. Pateiktame metode pagrindinis profilis apskaičiuojamas iš suvienodintų istorinių duomenų. Realiu stebėjimo laikotarpiu, neužbaigtų profilių trajektorijos yra sulyginamos su pagrindiniu profiliu. Tuomet apibendrinto tikėtimumo santykio kriterijaus statistika, kilusi iš besikeičiančių taškų teorijos, gali aptikti kitokias, nebūdingas situacijas. Palyginus modifikuotą AEWMA (angl. *adaptive exponentially weighted moving average*) diagramą su autorių pasiūlyta DLSI diagrama matoma, kad naujoji yra mažiau jautri staigiems vidutiniams poslinkiams. Pagrindinė to priežastis būtų ta, kad vidutinis poslinkis negali būti tinkamai priskiriamas DLSI sulyginimo algoritmui. Tačiau DLSI diagrama pranašesnė aptinkant laipsninius pokyčius arba ciklinius dinامينius pokyčius, kurie yra dažnai stebimi sudėtinguose inžinieriniuose procesuose [12].

Llanos ir kiti (2003) panaudojo dinaminį laiko skalės iškraipymą susidoroti su konkrečių atvejų motyvais (angl. *Case Based Reasoning (CBR)*). CBR metodika nurodo keturių etapų ciklą: gauti, pakartoti, peržiūrėti ir išsaugoti. Iš esmės saugomi tam tikri atvejai tolimesniems pakartojimams. Patys atvejai yra registruojami kartu su jų simptomų ir diagnostikos aprašymais. Tikslas – pakartoti šiuos atvejus naujų analogiškų problemų sprendimui. Todėl atsiradus naujai problemai pagrindinė procedūra susideda iš analogiškų atvejų peržiūrėjimo pagal jų aprašymą (atributais apibrėžti simptomai) ir rastų atvejų sprendimo pakartojimo. Dinaminis laiko skalės iškraipymas čia naudojamas kaip panašumo kriterijus siekiant įgyvendinti sprendžiamą atvejį ir taip sumažinti laiko nesutapimo daromą įtaką. Autorių sukurtas metodas pritaikomas registruojant 25 kW pastotės įtampos gedimus ir jų diagnozei [13]. Bregón ir kiti, 2006 atliko gana panašų darbą susijusį su išankstiniu klasifikavimu apie industriniame procese naudojamų rezervuarų klaidas [14].

Kassidas ir kiti (1998) panaudojo DLSI sinchronizuoti procesų trajektorijas, kurias sujungtų su daugiakrypte vyriausiųjų komponentų analize (angl. *Multiway Principal Component Analysis (MPCA)*) ir daugiakrypte projekcija į latentines struktūras (angl. *Multiway Projection to Latent Structures (MPLS)*). Iš pradžių turimas duomenų rinkinys kiekvienam kintamajam priklausančiam skirtingos grupės procesui. Šios trajektorijos privalo būti sinchronizuotos ir autorių pasiūlymu panaudojamas iteracinis metodas pagrįstas DLSI veikimu. Po sinchronizacijos visos trajektorijos privalo būti tokios pačios trukmės ir taip pat turi būti apibrėžta vidutinė trajektorija. Panaudojus šias trajektorijas kaip pirminius šaltinius, apskaičiuojama normalizuota svorių matrica pagal kiekvieno kintamojo svorį. Galiausiai panaudojus sinchronizuotas

trajektorijas buvo galima sukurti MPCA/ MPLS modelį skirtą stebėti produkto kokybę naujiems procesams [15].

Tie patys autoriai Kassidas ir kiti (1998) vėl panaudojo DLSI metodą norėdami gauti atstumą tarp trijų etaloninių kintamųjų, kurie vaizdavo skirtingus gedimus ir keletą išbandytų šablonų prieš tai išfiltruotų ir suvienodintu masteliu. Šablonai susidėjo iš 26 kintamųjų, taigi vyriausiųjų komponentų analizė buvo panaudota atskirti didelį skaičių kintamųjų, kurie buvo tarpusavyje koreliuoti. Kai naujas šablonas iš nežinomų klaidų tampa prieinamas, tuomet jis suprojektuojamas į apibrėžtą poaibį pagal vyriausiųjų komponentų analizę. Šie pagrindiniai komponentai leido autoriams palyginti trajektorijas gautas iš DLSI algoritmo [16].

## 1.2. FINANSINIŲ LAIKO EILUČIŲ TYRIMO ATVEJAI

Dažniausiai finansinių laiko eilučių analizėje yra naudojami tokie finansiniai duomenys kaip valiutų keitimo kursai, akcijų arba obligacijų kainos, įvairių indeksų reikšmės ir kiti. Vienas iš svarbiausių faktorių jų tyryboje yra periodiškumas, t.y. savaitiniai, mėnesiniai ar metiniai duomenys. Esama ypač aukšto valandinių ir dieninių duomenų arba netgi ultra aukšto dažnio, t.y. keletos sekundžių laiko intervalų. Paprastai finansinis rinkos tyrimas paaiškina stebimų duomenų savybes, priklausančias nuo tam tikrų politinių, ekonominių ar kitokių aplinkybių. Iš pirmo žvilgsnio atrodo, kad naftos pardavimo sandoriai, valstybės obligacijos ar tam tikros valiutos kursas savo kainos ypatybėmis yra visiškai skirtingos. Visgi paskutinio šimtmečio empiriniai kainų pasikeitimo tyrimai atskleidė, kad žiūrint iš statistinio požiūrio tokie finansiniai duomenys turi nemažai bendrų savybių.

Nagrinėjant finansines laiko eilutes dažnai susiduriama su tam tikromis savybėmis, kurios nevisuomet yra pageidaujamos ir apsunkina pačią tyrimo eigą bei iškreipia rezultatus. Viena iš jų vadinama sunkiomis uodegomis, t.y. finansinės grąžos logaritminio skirstinio požymis. Jis apibūdina didelių kainų pokyčių pasirodymą dažniau nei tuomet, kai skirstinys normalusis. Kita savybė – asimetrija arba svarto efektas (angl. *leverage effect*), t.y. po kainos kritimo egzistuoja didesnis kintamumas tarp to paties absoliutinio dydžio grąžų. Paskutinis požymis yra ilgalaikė atmintis. Yra žinoma keletas hipotezių, kurios remiasi įvairiais nestacionariais procesais kaip trendas ar šuoliai [17]. Būtent jomis bandoma pagrįsti šį ilgalaikės atminties efektą finansinių grąžų absoliutiniuose arba kvadratinuose dydžiuose. Visgi išlieka daugybė neaiškumų ir tai tampa aktuali finansiniu uždaviniu.

Pagrindiniai finansinių grąžų svyravimų modeliai vadinami sąlyginio heteroskedastiškumo. Plačiausiai taikomi yra du šios klasės modeliai, vienas iš jų vadinamas autoregresijos sąlyginio heteroskedastiškumo (angl. *ARCH*) modeliu, kitas – apibendrintas autoregresijos sąlyginio

heteroskedastiškumo (angl. *GARCH*). Šie modeliai populiarūs statistikos specialistų tarpe dėl aukščiau aptartų rinkos savybių adekvatumo ir panaudojimo paprastumo. Tačiau klasikiniai *GARCH* ir *ARCH* modeliai turi ir tam tikrų minusų, vienas iš jų yra trumpa atmintis [18]. Taip pat šie modeliai turi trūkumą, kai finansiniuose duomenyse stebima tolimesnių priklausomybė. Taip pat finansiniai duomenys dažniausiai turi pastebimą kintamumo kaitą, t.y. heteroskedastiškumą, dėl ko jų negalima interpretuoti tiesiniais *ARIMA*, *FARIMA* ar *ARMA* tipo modeliais.

Dinaminio laiko skalės iškraipymo algoritmas yra neįprastai laiko skalės nesutapimams, todėl tai tampa jo pranašumu prieš kitus panašumo metodus. Vis dažniau jis bandomas pritaikyti ir finansiniuose duomenyse, kad surastų pasikartojančias laiko eilučių formas reikšmes. Apskaičiuotus atstumus pagal *DLSI* metodą tarp laiko eilučių galima panaudoti tolimesniems tyrimams, t.y. prognoziniais, klasteriniams ir analiziniams modeliams. Taigi *DLSI* metodo panaudojimas yra gana novatoriškas finansinių laiko eilučių tyrime. Visgi šiuo metu realizuotų finansinių modelių pasiremiant *DLSI* metodu nėra daug, dar mažiau atlikta panašumo tyrimų tokio tipo duomenims. Kitame skyriuje bus aprašomi atlikti eksperimentai su finansinėmis laiko eilutėmis.

### **1.3. DINAMINIO LAIKO SKALĖS IŠKRAIPYMO ALGORITMO TAIKYMAS FINANSINĖMS LAIKO EILUTĖMS TIRTI**

Tsinaslanidis ir kiti (2014) pritaikė dinaminio laiko skalės iškraipymo metodą 6 finansų rinkos indeksams. Tyrimas buvo atliekamas diachroniškai, atskirais etapais, pasiremiant *DLSI* kaip panašumo matu ir dubliuotu *DLSI* koreliacijos koeficientu  $\rho_{DLSI}$ . Taip pat buvo atsižvelgiama į kintamumo ir Pirsono koreliacijos koeficientą  $\rho_P$ . Gauti rezultatai patvirtino išankstinius spėjimus, kad didžiuose finansinių rodiklių nepastovumo laikotarpiuose egzistuoja stipresnis panašumas. Tiek tiesinis sąryšis paremtas  $\rho_P$ , tiek netiesinis  $\rho_{DLSI}$  rodė šiuos rastus panašumus. Vėliau buvo atliktas imitacinis modeliavimas tarp koeficiento  $\rho_{DLSI}$  ir glaudžiai su juo susijusių kitų dviejų dydžių  $\rho_P$ ,  $\rho_S$  (Spirmeno koreliacijos koeficientas). Jie parodė, kad koeficientas  $\rho_{DLSI}$  artėja prie nulio, kai  $\rho_P$  ir  $\rho_S$  įgyja didesnes absoliučias reikšmes. Dydis  $\rho_{DLSI}$  įgyja didžiausią reikšmę tuomet, kai koreliacija artėja prie nulio. Taigi dinaminio laiko skalės iškraipymo metodo privalumas šiame tyrime yra jo, kaip panašumo mato gebėjimas, palyginti skirtingo laiko ar ilgio duomenų eilutes, kadangi su tradiciniais koreliacijos koeficientais, tai įgyvendinti neįmanoma. Iš finansinės pusės šio metodo pritaikymas yra reikšmingas tuo, kad *DLSI* gali būti panaudotas tyrinėjant rinkos sezoniškumą. Pavyzdžiui, galima palyginti kelių mėnesių grįžtamųjų verčių reikšmių pokyčius, kurie gali skirtis trukme. Vėliau gautus rezultatus

galima panaudoti kuriant prognozinius modelius pagrįstus dinaminio laiko skalės iškraipymo metodu [19].

Kia, Haratizadeh ir Zare (2013) tyrinėjo valiutos kurso prognozavimo problemą, kuri vis dar išlieka gana aktuali per pastarąjį dešimtmetį. Jų tikslas buvo ištirti artimiausių kaimynų algoritmo ir DLSI metodiką, taip pat pagerinti svyravimo prognozavimą ir pateikti geresnius vertinimo kriterijus finansinės rinkos prognozavimui. Tyrimas buvo atliktas su USD/ JPY (angl. *United States dollar/ Japanese yen*) valiutos kursų laiko eilutėmis. Pasirinktas laikotarpis nuo 1971 iki 2012 metų, kuris buvo išskaidytas į 30 elementų segmentus dėl cikliško mėnesinio laiko eilučių pobūdžio. Tuomet du skirtingi duomenų rinkiniai sudaryti iš šių 30 segmentų buvo išskaidyti santykiu 7:3. Artimiausių kaimynų metodas buvo panaudotas surasti 3 artimiausius kaimynus, o atstumai buvo pasirinkti pagal panašumo funkciją pagrįstą dinaminio laiko skalės iškraipymo metodu. Gauti rezultatai parodė, jog buvo pagerinti dažniausiai naudojami laiko eilučių prognozavimo kriterijai [20].

Bagheri ir Peyhani (2014) sukūrė prekybos platformos Forex vedlį, t.y. tokią sistemą, kuri naudoja buvusias valiutų kursų vertes, sulygina esamųjų ir buvusių verčių diagramas bei galiausiai padeda priimti sprendimą prekiautojui. Ši sistema atspindi visiškai naują tikėtinos vertės metodą, kurio prognozė apima vieną žingsnį į prieky panaudojant ANFIS (angl. *adaptive network based fuzzy interference system*), QPSO (angl. *quantum behaved particle swarm optimization*), WT (angl. *wavelet transform*) modelius. Pasinaudojus įrankiu galima sudaryti akcijų modelių diagramas panaudojant WT ir DLSI metodus tuo pačiu metu. Pasiūlyto vedlio pranašumas prieš kitas iki tol sukurtas sistemas įvardijamas kaip prekybos akcijomis patarimai, o ne valiutų kursų verčių prognozavimas. Eksperimentas buvo atliekamas ir su sintetinėmis duomenų imtimis, ir realiomis. Sukurtos sistemos gebėjimas priimti teisingus prekybinius sprendimus yra 69 procentai, todėl autorių pasiūlytas metodas tikrai korektiškai atlieka finansinius sprendimus [21].

Lee ir Suk Jun (2011) analizavo investicinių sprendimų skirtų prekybos sistemos rinkai panaudojimą ir pritaikė DLSI metodą, kuris leido nustatyti panašių modelių pasikartojimą laiko eilutėje bei parinkti tinkamą prekybos pradžios laiką. Tyrime buvo išnagrinėti keletas akcijų rinkų pavyzdžių ir panaudotas DLSI algoritmas pagal turimos eilutės dažnį (10, 30, 60 minučių, diena). Po to buvo surasti panašiausi šių akcijų modeliai ir taip pat patikrinti vykdant imitacinį prekybos modelį. Gautoms pelningiausioms strategijoms dar kartą buvo pritaikytas DLSI

metodas pasikartojančių intervalų radimui. Eksperimento rezultatas – optimalus prekybos laikas panaudojant DLSI metodą priklausomą nuo turimų duomenų dažnio [22].

#### **1.4. LITERATŪROS ŠALTINIŲ ANALIZĖS APIBENDRINIMAS**

Šiame skyriuje buvo apžvelgta, plačiau susipažinta ir pateikta įvairių pasaulyje atliktų tyrimų tiesiogiai susijusių su dinaminio laiko skalės iškraipymo metodu. Pirmą kartą jis buvo panaudotas 1950 metais automatiniam kalbos atpažinimo uždaviniui spręsti, o Lietuvoje pabandytas pritaikyti analogiškam uždaviniui tik 1970 metais. Nors šis metodas jau seniai žinomas bei naudojamas įvairiose kryptyse, tačiau jo taikymas finansiniuose tyrimuose nėra labai dažnas. Visgi šiuolaikiniame pasaulyje finansinės rinkos vis sunkiau analizuojamos klasikiniiais metodais, todėl reikia ieškoti naujų metodų arba taikyti metodus iš kitų sričių, kurie leistų tirti finansines laiko eilutes ir jose atpažinti tam tikrus dėsningumus.



## 2. TYRIMŲ METODAI

Šioje darbo dalyje pristatomi atstumų metrikų ir dinaminio laiko skalės iškraipymo metodai, kuriais remiantis buvo atliktas tolimesnis tyrimas. Tyrimų metodų darbo dalis pradedama pristatant paprastas atstumų metrikas. Po to supažindinama su dinaminio laiko skalės iškraipymo metodu. Apibūdinamos šio metodo sudedamosios dalys: kaštų matrica, sutapatavimo kelias, globalūs apribojimai, kaštų ir trajektorijos statistikos. Pristatomi pagrindiniai metodo reikalavimai, kaip jais remiantis gaunamos tinkamiausios trajektorijos, parametrų suradimo formulės.

### 2.1. ATSTUMŲ METRIKOS

Egzistuoja daugybė atstumo mato variantų, kuriuos panaudojant galima apskaičiuoti atstumą tarp dviejų sekų reikšmių. Taigi turime atstumo matą tarp  $x_i$  ir  $y_i$ , kurį pažymėsime  $D_{i,j}$ . Dažniausiai norint apskaičiuoti atstumą tarp dviejų vienodo ilgio vektorių yra naudojamas tokios atstumo metrikos kaip Euklido, Manheteno (angl. *Manhattan*), Minkovskio. Jos apskaičiuoja tik tiesioginių sekų taškų atstumus, t.y. horizontalius arba vertikalius. Taigi Euklido atstumas apskaičiuojamas tarp dviejų vienodo ilgio sekų pagal formulę [23]:

$$D_{i,j} = \sqrt{(x_i - y_j)^2} \quad (1)$$

Manheteno atstumas yra paprasta dviejų sekų horizontalių ir vertikalų komponentų suma, kuri apskaičiuojama taip [23]:

$$D_{i,j} = \sum_{i=1}^n |x_i - y_j| \quad (2)$$

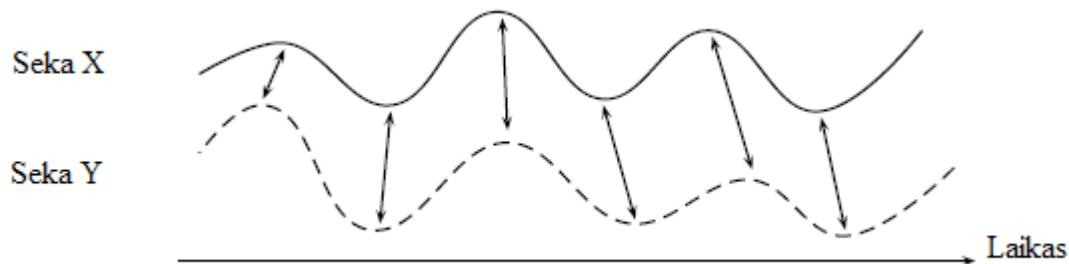
Minkovskio atstumas tarp dviejų vektorių su eile  $p$  randamas pagal formulę [23]:

$$D_{i,j} = \sqrt[p]{(x_i - y_j)^p} \quad (3)$$

Tačiau šios metrikos nėra tinkamos tuo atveju, kai turimų sekų ilgiai yra skirtingi. Taip pat nėra tikslinga jas naudoti jeigu skiriasi laiko eilučių greičiai, t.y. egzistuoja vėlavimai. Todėl tiksliau ir naudingiau naudoti dinaminio laiko skalės iškraipymo metodą siekiant surasti atstumą tarp dviejų sekų. Sekančiame skyriuje ir bus pristatytas šis metodas.

## 2.2. DINAMINIS LAIKO SKALĖS IŠKRAIPYMAS

Dinaminis laiko skalės iškraipymas (angl. *dynamic time warping*) yra plačiai žinomas metodas skirtas tarp dviejų nuo laiko priklausomų sekų rasti optimalų panašumą prie tam tikrų apribojimų. Šis algoritmas įvertina panašumą tarp dviejų sekų, kurios tarpusavyje gali skirtis trukme arba greičiu. Intuityviai šios sekos iškreipiamos netiesiškai taip, kad atitiktų viena kitą. Tai iliustruojama 2.1 paveikslėlyje, čia sutapatinti taškai pažymėti rodyklėmis:



2.1.pav. Dviejų nuo laiko priklausomų sekų sutapatinimas.

Pagal Rabiner ir Juang (1993) turime dvi diskretaus laiko eilutes  $X$  ir  $Y$ , kurias norime palyginti. Taigi  $X$  yra tikslas ( angl. *query*), o  $Y$  – įvestis (angl. *reference*). Jas žymėsime taip:

$$\begin{aligned} X &= (x_1, x_2, \dots, x_{t_x}), \\ Y &= (y_1, y_2, \dots, y_{t_y}), \end{aligned} \quad (4)$$

$t_x, t_y$  –  $X$  ir  $Y$  trukmės.

Apibrėšime šių  $X$  ir  $Y$  eilučių laiko indeksus atitinkamai,  $i = 1, 2, \dots, t_x$  ir  $j = 1, 2, \dots, t_y$ . Po to konstruojame skirtumo arba atstumų matricą  $D$ , kurios formatas  $t_x \times t_y$ .

$$\begin{aligned} D &= (D_{i,j})_{t_x \times t_y}, \\ D_{i,j} &= d(x_i, y_j) = (x_i - y_j)^2, \\ i &= 1, 2, \dots, t_x; j = 1, 2, \dots, t_y. \end{aligned} \quad (5)$$

Pažymėsime, kad  $D_{i,j} = d(x_i, y_j) = d(i, j)$  [2].

Pagrindinis metodo tikslas – laiko skalę pakeisti taip, kad naujai sudarytos laiko eilutės  $X'$  ir  $Y'$  būtų kaip galima panašesnės. Tokiu tikslu ieškoma funkcijų pora:

$$\begin{aligned} \phi(k) &= (\phi_x(k), \phi_y(k)), \text{ čia} \\ \phi_x(k) &\in (1, \dots, t_x), \end{aligned} \quad (6)$$

$$\begin{aligned}\phi_y(k) &\in (1, \dots, t_y), \\ k &= 1, T.\end{aligned}$$

Optimalaus išlyginimo funkcijos  $\phi_x$  ir  $\phi_y$  pereina visus laiko indeksus atsižvelgdamos į pradines sekas  $X$  ir  $Y$ . Taigi ši funkcijų porą turi atitikti optimalų išlyginimą pagal apskaičiuotą:

$$d_\phi(X, Y) = \sum_{k=1}^t d(\phi_x(k), \phi_y(k)) m_\phi(k) / M_\phi, \quad (7)$$

čia  $m_\phi$ - žingsnio svoris,  $M_\phi$  – normalizuojanti konstanta.

Tuomet optimalus išlyginimas  $\phi$  būtų apskaičiuojamas pagal tokią sąlygą:

$$D(X, Y) = \min_{\phi} d_\phi(X, Y) \quad (8)$$

Dalinis sukaupėtasis minimumas randamas tokiu būdu:

$$\begin{aligned}D(i, j) &= \min_{\phi, t'} \sum_{k=1}^{t'} \sum_{k=1}^t d(\phi_x(k), \phi_y(k)) m_\phi(k) / M_\phi, \\ \phi_x(t') &= i, \phi_y(t') = j,\end{aligned} \quad (9)$$

čia  $m_\phi$ - žingsnio svoris,  $M_\phi$  – normalizuojanti konstanta

### 2.2.1. SUKAUPTŲ SKIRTINGUMO ĮVERČIŲ (ATSTUMŲ) MATRICA

Tegul  $D$  žymi matricą  $t_x \times t_y$  su visais galimais atstumo matais tarp  $x$  ir  $y$  sekų, kur  $D_{i,j}$  atspindi  $(i, j)$ -ąjį elementą iš šios matricos, čia  $j$  yra matricos eilutės indeksas, o  $i$  – stulpelio indeksas. Jeigu atstumo matrica  $D$  turi daugybę elementų, tuomet panašumo dydis privalo sutraukti šią matricą į tokį vieną skaičių:  $sim(X, Y) = norm(D)$  [24, 25].

Skaičiuojant trumpiausią trajektoriją  $p^*$  galima surasti visas įmanomas trajektorijas tarp  $X$  ir  $Y$  sekų, visgi tokio siekiamo skaičiavimo sudėtingumas išauga eksponentiškai nuo  $N$  ir  $M$ . Siekiant sumažinti skaičiavimo kaštus yra naudojamas  $O(NM)$  algoritmas, pagrįstas dinaminio programavimu. Iš pradžių apibrėžiama dalinė seka  $X = (x_1, \dots, x_n)$ ,  $\forall n \in (1, N)$  ir  $Y = (y_1, \dots, y_m)$ ,  $\forall m \in (1, M)$  ir rinkinys [26] :

$$D(n, m) = DLSI(X, Y) \quad (10)$$

Čia  $D(n, m)$  reikšmės apibrėžia  $N \times M$  matricą  $D$ . Ši matrica dar vadinama atstumų arba sukaupėtų skirtingumo įverčių matrica (angl. *accumulated cost matrix*). Norint efektyviai apskaičiuoti matricą  $D$ ., ji turi tenkinti tapatybes [26] :

$$1. D(n, 1) = \sum_{k=1}^n c(x_k, y_1), \forall n \in (1, N),$$

$$\begin{aligned}
2. \quad & D(1, m) = \sum_{k=1}^m c(x_1, y_k), \forall m \in (1, M), \\
3. \quad & D(n, m) = \min\{D(n-1, m-1), D(n-1, m), D(n, m-1)\} + c(x_n, y_m), \\
& \text{kiekvienam } 1 < n \leq N \text{ ir } 1 < m \leq M.
\end{aligned} \tag{11}$$

Taigi norint rasti sutapatinimo kelią  $p^*$  reikia apskaičiuoti visą matricą  $D$  [26].

### 2.2.2. TRAJEKTORIJA (SUTAPATINIMO KELIAS)

Trajektorija braižoma per visą elementų atstumų matricą, pradedant nuo elemento (1,1) ir baigiant ties elementu  $(t_x, t_y)$ . Žingsniai per visą šią trajektoriją saugomi laiko sekos atžvilgiu, negrįžtant prie tų pačių matricos elementų ir nežingsniuojant laiku atgal. Ieškant panašumo mato, pagrindinis tikslas surasti tokį sutapatinimo kelią, kuris būtų gautas maksimaliai minimizavus kaštus. Vadinasi ieškoma trajektorija privalo turėti mažiausius atstumus tarp įvesties ir tikslinės sekų reikšmių [25].

Tegul  $\Pi$  yra visos galimos trajektorijos per atstumų matricą  $D$ . Tuomet  $\pi \in \Pi$  bus vienas iš šių kelių. Pati trajektorijos funkcija priskiria įvesties sekos indeksą ir tikslinės sekos indeksą duotos trajektorijos indeksui, susietam su trajektorija  $\pi \in \Pi$ . Kai turimas toks sutapatinimo kelio indeksas  $p = 1, \dots, P$  ir funkcija  $(i_p, j_p) = \text{path}(p)$ , tuomet grąžinamas įvesties sekos indeksas  $i_p \in (1, \dots, t_x)$  ir tikslinės sekos laiko indeksas  $j_p \in (1, \dots, t_y)$ , susietas su ieškomo kelio indeksu. Trajektorijos ilgis per atstumų matricą yra  $P$ , čia  $\max(t_x, t_y) \leq P < t_x + t_y$ . Dėl paprastumo pažymėsime, kad bet kokia galima trajektorijos funkcija yra  $\text{path}(p)$ .

Siekiant išsaugoti dviejų tyrinėjamų sekų laiko indeksą, tinkamos trajektorijos funkcija turi tenkinti tam tikrus apribojimus [24].

1. Monotoniškumo apribojimu siekiama išlaikyti laiko eiliškumą, todėl uždraudžiami žingsniai atgal.

$$\begin{aligned}
\phi_x(k+1) &\geq \phi_x(k), \\
\phi_y(k+1) &\geq \phi_y(k).
\end{aligned} \tag{12}$$

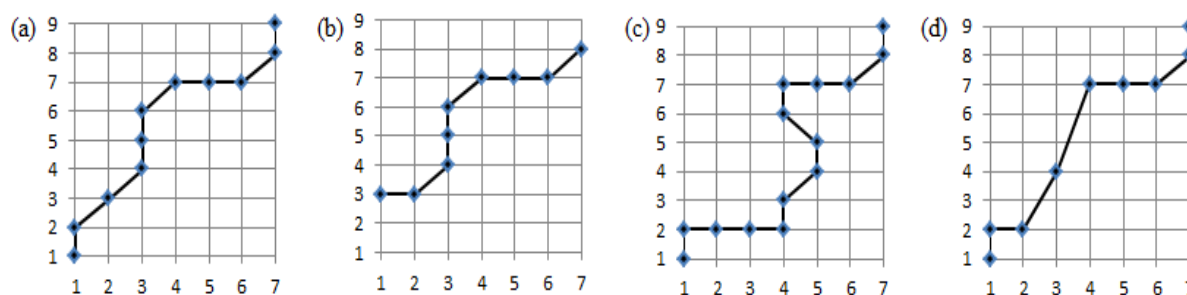
2. Tęstinumo apribojimas uždraudžia peršokti laiko skalę daugiau nei per 1 laiko vienetą:

$$\begin{aligned}
|\phi_x(k+1) - \phi_x(k)| &\leq 1, \\
|\phi_y(k+1) - \phi_y(k)| &\leq 1.
\end{aligned} \tag{13}$$

3. Globalaus išlyginimo apribojimas nurodo, kad abiejų laiko eilučių pradžios ir pabaigos taškai turi sutapti [25].

$$\begin{aligned}\phi_x(1) &= \phi_y(1) = 1, \\ \phi_x(t) &= t_x, \\ \phi_y(t) &= t_y.\end{aligned}\tag{14}$$

Šias aptartas sąlygas iliustruosime konkrečiu pavyzdžiu, kai turimos dvi sekos  $X$  ir  $Y$ . Pirmosios ilgis yra  $N = 9$ , o antrosios ilgis  $M = 7$ . Paveikslėlis (a) vaizduoja tokią sudarytą trajektoriją, kuri tenkina visus 3 anksčiau paminėtus apribojimus. Paveikslėlyje (b) pažeidžiama globalaus išlyginimo savybė, t.y. taškas (1; 3) nesutampa su  $N$  ir  $M$  pradžios tašku (1; 1), o trajektorijos taškas (7; 8) nesutampa su šių sekų pabaigos tašku (7; 9). Trečiasis paveikslėlis (c) parodo monotoniškumo apribojimo netenkinimą, t.y. trajektorijos taškai (4; 6) ir (4; 7) yra žingsniai atgal. Paskutinis paveikslėlis (d) vaizduoja tęstinumo savybės pažeidimą, kadangi yra praleidžiami  $X$  sekos 5 ir 6 elementai, t.y. iš trajektorijos taško (3; 4) pereinama į tašką (4; 7) ir taip laiko skalė persokama daugiau nei per vieną vienetą.



2.2.pav. Iškraipymo trajektorijos apribojimų savybių pavyzdžiai

Egzistuoja daugybė galimų trajektorijų iš kurių galima apskaičiuoti panašumą, tuo atveju kai norima teisingai nustatyti jį tarp įvesties ir tikslinės sekos. Kaštų statistika (bus plačiau aprašyta sekančiame skyriuje) gali pateikti informaciją tik apie vieną dydį, bet trajektorija optimizuojanti šiuos kaštus suteikia galimybę apskaičiuoti norimą panašumą. Taigi pasinaudojus trajektorijos funkcija  $(i_p, j_p) = path(p)$ , jos sekos gali būti suformuotos pasiremiant trajektorijos indeksu. Tegul  $\tilde{x}_p = x_{i_p}$  žymi laike iškraipytą įvesties seką, o  $\tilde{y}_p = y_{j_p}$  žymi laike iškraipytą tikslinę seką visiems  $p = 1, \dots, P$ , kur simbolis  $\sim$  žymi iškraipymą. Kitaip tariant, abi sekos yra išplėtos priklausomai nuo kelio indekso  $p$ . Galimybė palyginti trajektorijos sekas leidžia apskaičiuoti jos atstumus. Taip pat panaudojus kartu ir atstumo matą  $D_{i,j} = d(x_i, y_j)$ , galima apskaičiuoti kelio atstumą tarp įvesties sekos ir tikslinės sekos reikšmių per visą turimą sutapatinimo kelią. Taigi pačio kelio atstumus pažymėsime  $D_p = D_{i_p, j_p} = d(x_{i_p}, y_{j_p}) =$

$d(\tilde{x}_p, \tilde{y}_p)$ . Tegul  $\Pi$  yra visi galimi atstumų matricos sutapatavimo keliai, o  $\pi \in \Pi$  žymi vieną iš kelių ir  $cost_\pi(X, Y)$  yra šio kelio perėjimo kaštai [24, 25]. Taigi tikslas yra tenkinti šią lygybę:

$$sim(X, Y) = \min_{\pi \in \Pi} (cost_\pi(X, Y)) \quad (15)$$

Optimalus trajektorijos algoritmas pagal M. Muller [27] gali būti aprašomas tokiu būdu.

**Ivestis:** Sukauptų skirtingumo įverčių (atstumų) matrica  $D$

**Išeitis:** Optimali trajektorija  $p^*$

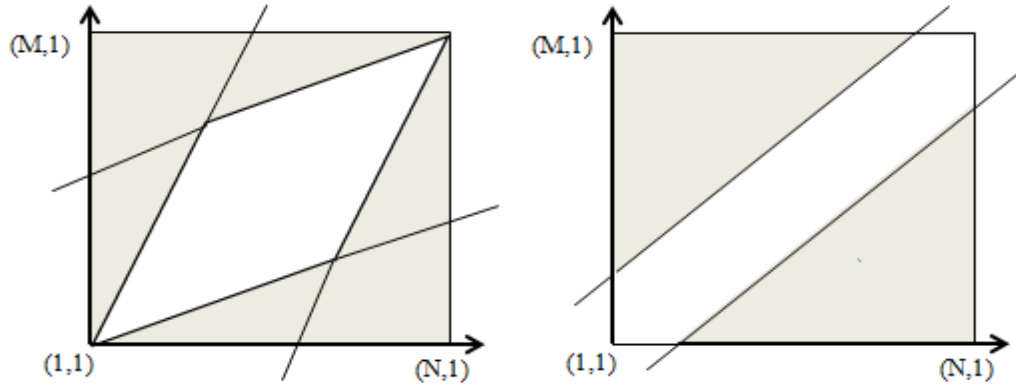
**Vykdomo eiga:** Optimali trajektorija  $p^* = (p_1, \dots, p_L)$  randama skaičiuojant atvirkštine indeksų tvarka pradėdant nuo  $p_L = (N, M)$ . Tarkime, kad turime apskaičiuoti dydį  $p_l = (n, m)$ . Kai  $(n, m) = (1, 1)$ , tai bent vienas  $l = 1$  ir tuomet algoritmas nustoja vykti. Kitu atveju,

$$p_{l-1} := \begin{cases} (1, m-1), & \text{jei } n = 1 \\ (n-1, 1), & \text{jei } m = 1 \\ \operatorname{argmin} \begin{cases} D(n-1, m-1), \\ D(n-1, m), \\ D(n, m-1). \end{cases} & \text{kitu atveju} \end{cases}$$

Taigi trajektorija – tai seka, sudaryta iš tiesioginių (diagonalių), suspaustų (vertikalių) ir išplėstų (horizontalių) priskyrimų nuo apatinio kairiojo kampo (elemento (1,1)), iki viršutinio kairiojo kampo (elemento  $(t_x, t_y)$ ) skirtingumų matricoje.

### 2.2.3. GLOBALAUS IŠLYGINIMO APRIBOJIMAS

Dauguma specialistų (Itakura, 1975; Sakoe and Chiba 1978; Keogh ir Pazzani 2000) naudojančių dinaminį laiko skalės iškraipymo algoritmą savo praktikoje siekia apriboti pačios skalės iškraipymo trajektoriją, kad ši kaip galima mažiau nuklystų nuo pagrindinės atstumų matricos įstrižainės. Iškraipymo langų vadinsime tokį atstumų matricos poaibį, kurį iškraipytai trajektorijai leidžiama aplankyti. Dažniausiai naudojami ir plačiausiai realizuoti globalaus išlyginimo apribojimai vadinami Sakoe Chiba juosta [28] ir Itakura lygiagretainiu [29].



2.3.pav. Globalaus išlyginimo apribojimai

Sakoe Chiba juosta veikia išilgai pagrindinės įstrižainės ir turi fiksuotą (horizontalų ir vertikalų) plotą  $T \in N$ . Šis apribojimas reiškia, kad elementas  $x_n$  negali būti priskiriamas tik į vieną iš elementų  $y_m$  su tokia  $m \in \left[ \frac{M-T}{N-T} \cdot (n - T), \frac{M-T}{N-T} \cdot (n + T) \right] \cap (1: M)$ . Itakura lygiagretais apibūdina sritį, kuri konstruoja slopinimą iškraipymo trajektorijai [27].

Egzistuoja keletas svarbių priešasčių dėl kurių šie apribojimai yra taikomi, viena iš jų yra siekis pagreitinti algoritmo atstumo skaičiavimą. Tačiau pagrindinė priežastis yra užkirsti kelią patologiniams iškraipymams, kur santykinai maža vienos sekos grupės dalis būtų atvaizduota į gana didelę antrosios sekos dalį. Globalaus išlyginimo apribojimų svarbumas buvo pagrįstas pagrindinių šio dinaminio laiko skalės iškraipymo iniciatorių Sakoe ir Chiba. Jie labiausiai buvo įsigilinę į kalbos sekų rikiavimo modelius. Visgi empiriškai buvo patvirtinta, kad jie taip pat tinkami ir dirbant su kitais parametrais turinčiais duomenų rinkiniais. Pavyzdžiui, finansiniais, medicininiais, biometriniais duomenimis ir dauguma kitų [30].

Tačiau globalaus išlyginimo apribojimų naudojimas gali būti ir probleminis tuo atžvilgiu, jog optimali iškraipyta trajektorija gali kirsti nurodytos srities ribas. Kitaip sakant gautas suvaržytas sutapatavimo kelias  $p$  dažniausiai nesutampa su nesuvaržytu sutapatavimo keliu. Šis faktas gali sukelti nepageidautinus ar net visiškai nenaudingus sutapatavimo rezultatus [27].

#### 2.2.4. KAŠTŲ IR SUTAPATINIMO KELIO STATISTIKOS

Duota funkcija  $(i_p, j_p) = path(p)$ , iš kurios trajektorijos statistikos gali būti apskaičiuojamos apibūdinant tiesioginius, suspaustus ar išplėstus braižymus per visą trajektoriją. Vadinasi trajektorijos statistikos bendru požiūriu apibūdina tikslinės sekos visumą suspaudimų ir išplėtimų susietų su įvesties seka [24, 25].

Tiesioginio braižymo statistika gali būti apskaičiuojama iš kelio susieto su tiesiais braižymais tokiu būdu [24, 25]:

- Tiesioginis numeris:  $P^D = \|\{path(p+1) - path(p) = (1,1)\}_{p=1}^{p-1}\|$

- Tiesioginės trajektorijos procentas:  $\frac{P^D}{P}$
- Tiesioginės įvesties procentas:  $\frac{P^D}{t_x}$
- Tiesioginio tikslo procentas:  $\frac{P^D}{t_y}$
- Tiesioginis minimumas arba mažiausias gretimų įstrižainių judėjimo skaičius:

$$P_{min}^D = \min_m \{m: path(p) - (0,0) = \dots = path(p+m) - (m,m), 1 \leq p \leq P\} \quad (16)$$

- Tiesioginis maksimumas arba didžiausias gretimų įstrižainių judėjimo skaičius:

$$P_{max}^D = \max_m \{m: path(p) - (0,0) = \dots = path(p+m) - (m,m), 1 \leq p \leq P\} \quad (17)$$

Tegul optimizavimo kaštai  $cost(X, Y)$  žymi vieną kaštų statistiką susietą su trajektorija tarp tikslinės ir įvesties sekų. Tuomet ši statistika yra susieta su tam tikra trajektorijos  $\pi \in \Pi$ , gaunamas toks pažymėjimas  $cost_\pi(X, Y)$ . Suspausto braižymo statistika gali būti apskaičiuojama iš trajektorijos susietos su suspaustais braižymais tokiu būdu [24, 25]:

- Suspaustas numeris:  $P^C = \|\{path(p+1) - path(p) = (1,1)\}_{p=1}^{p-1}\|$
- Suspaustos trajektorijos procentas:  $\frac{P^C}{P}$
- Suspaustos įvesties procentas:  $\frac{P^C}{t_x}$
- Suspausto tikslo procentas:  $\frac{P^C}{t_y}$
- Suspaustas minimumas arba mažiausias gretimų vertikalių judėjimų skaičius:

$$P_{min}^C = \min_m \{m: path(p) - (0,0) = \dots = path(p+m) - (m,m), 1 \leq p \leq P\} \quad (18)$$

- Suspaustas maksimumas arba didžiausias gretimų vertikalių judėjimų skaičius:

$$P_{max}^C = \max_m \{m: path(p) - (0,0) = \dots = path(p+m) - (m,m), 1 \leq p \leq P\} \quad (19)$$

Išplėsto braižymo statistika gali būti apskaičiuojama iš trajektorijos susietos su išplėstais braižymais tokiu būdu [24, 25]:

- Išplėstinis numeris:  $P^E = \|\{path(p+1) - path(p) = (1,1)\}_{p=1}^{p-1}\|$
- Išplėstos trajektorijos procentas:  $\frac{P^E}{P}$
- Išplėstos įvesties procentas:  $\frac{P^E}{t_x}$



- Išplėsto tikslo procentas:  $\frac{P^E}{t_y}$
- Išplėstas minimumas arba mažiausias gretimų horizontalių judėjimų skaičius:

$$P_{min}^E = \min_m \{m: path(p) - (0,0) = \dots = path(p+m) - (m,m), 1 \leq p \leq P\} \quad (20)$$

- Suspaustas maksimumas arba didžiausias gretimų horizontalių judėjimų skaičius:

$$P_{max}^E = \max_m \{m: path(p) - (0,0) = \dots = path(p+m) - (m,m), 1 \leq p \leq P\} \quad (21)$$

Taigi iš šių visų aptartų trajektorijų statistikų galima įvertinti visumą reikalingų iškraipymų, norint pereiti per visą nubraižytą trajektoriją. Nors panašumo matas susietas su dvejomis sekomis gali būti per mažas, tačiau perkrauta kraipymų visuma gali parodyti, jog šios dvi sekos nėra panašios. Dėl šios priežasties trajektorijos statistikos gali padėti aptikti laiko anomalijas įvesties sekoje, tuo atveju, kai vyksta lyginimas su tiksline seka [24, 25].

Kaštų statistikos gali būti apskaičiuotos apibūdinant visą kaštų keliavimą per nagrinėjamą trajektoriją. Taigi duotas trajektorijos atstumo vektorius  $\vec{D} = \{D_p\}_{p=1}^P$  ir/ arba susijęs trajektorijos vektorius  $\vec{r} = \{r_p\}_{p=1}^P$ . Galimos tokios kaštų statistikos: [24, 25]

1. Skaičius	$P$	$P$
2. Iš viso	$\sum_{p=1}^P D_p$	$\sum_{p=1}^P r_p$
3. Vidurkis	$\frac{1}{P} \sum_{p=1}^P D_p$	$\frac{1}{P} \sum_{p=1}^P r_p$
4. Standartinis nuokrypis	$std\{D_p\}_{p=1}^P$	$std\{r_p\}_{p=1}^P$
5. Minimumas	$min\{D_p\}_{p=1}^P$	$min\{r_p\}_{p=1}^P$
6. Maksimumas	$max\{D_p\}_{p=1}^P$	$max\{r_p\}_{p=1}^P$

Be to šios aukščiau apibrėžtos kaštų statistikos gali įgauti baudą už per ilgą trajektoriją per atstumų matricą [24, 25]:

- Įvesties ilgio vidurkis  $\frac{1}{t_x} \sum_{p=1}^P D_p$   $\frac{1}{t_x} \sum_{p=1}^P r_p$
- Tikslo ilgio vidurkis  $\frac{1}{t_y} \sum_{p=1}^P D_p$   $\frac{1}{t_y} \sum_{p=1}^P r_p$
- Mažiausios trajektorijos vidurkis  $\frac{1}{max(t_x, t_y)} \sum_{p=1}^P D_p$   $\frac{1}{max(t_x, t_y)} \sum_{p=1}^P r_p$
- Didžiausios trajektorijos vidurkis  $\frac{1}{(t_x + t_y - 1)} \sum_{p=1}^P D_p$   $\frac{1}{(t_x + t_y - 1)} \sum_{p=1}^P r_p$

Jeigu atstumų sumą padalinsime iš kažko kito, o ne trajektorijos ilgio, tuomet kaštų statistika susieta su skirtingomis sekų poromis gali būti baudžiama dėl didelio porų ilgio. Taigi iš

šių aptartų kaštų statistikų galima įvertinti visumą reikalingų kaštų, norint pereiti visą trajektoriją per atstumų matricą [24, 25].

### 2.3. SPIRMENO KORELIACIJOS KOEFICIENTAS

Dažniausiai tyrinėjant kiekybinių kintamųjų reikšmių kitimą siekiama išsiaiškinti ar šie kintamieji yra tarpusavyje priklausomi bei kokia yra jų ryšio tendencija. Siekiant atsakyti į šį klausimą yra panaudojama koreliacinė analizė. Praktikoje analizuojant realius reiškinius dažnai susiduriame su tarpusavyje susietais kintamaisiais, t. y. kai vieno kintamojo reikšmės kitimas priklauso nuo kito kintamojo kitimo. Ši kintamiesiems būdinga ryšio tendencija vadinama monotone arba nemonotone. Taigi koreliacijos koeficientai (kintantys nuo -1 iki +1 arba nuo 0 iki +1) naudojami ryšio tarp kintamųjų stiprumui įvertinti. Pagal koreliacijos koeficiento dydį yra daromos išvados apie nagrinėjamų kintamųjų koreliacinio ryšio stiprumą [31].

Spirmeno koreliacijos koeficientas naudojamas įvertinti ryšio stiprumui tarp kintamųjų, kurie yra išmatuoti santykių, tvarkos ir intervalinėje skalėse. Hipotezė apie Spirmeno ranginės koreliacijos koeficiento reikšmingumą [31] :

$$H_0: \rho_S = 0 \quad (22)$$

$$H_1: \rho_S \neq 0$$

Taigi Spirmeno koreliacijos koeficientas  $\rho_S$  naudojamas ryšio tarp kintamųjų X ir Y stiprumo nustatymui monotoniškumo prasme. Vadinasi kintamajam X didėjant, kintamasis Y irgi monotoniškai didėja (nebūtinai tiesiškai), t.y. didėja jei  $\rho_S > 0$  , mažėja jei  $\rho_S < 0$  [31].

### 2.4. NAUDOTA PROGRAMINĖ ĮRANGA

Šiame skyrelyje bus pristatyti keletas įvairių programinės įrangos paketų, kurie yra specializuoti duomenų analizei. Kadangi dauguma skaičiavimų be šiuolaikinių technologijų pagalbos užimtų gana daug laiko, todėl šiam darbui palengvinti naudojami tokie programiniai paketai:

- **R Studio**

R Studio yra kiekvienam vartotojui prieinama atviro kodo R programos plėtinio aplinka. Ši programa parašyta pasinaudojant C++ programavimo kalba ir naudoja Qt sistemą grafinei vartotojo sąsajai. R Studio naudojama tiesioginiam vartotojo programos kodui vykdyti, duomenų ir gautų rezultatų vizualiniam atvaizdavimui. Vienas iš didžiausių šios programos privalumų, kad ji lengvai tobulinama, papildoma naujomis bibliotekomis ir funkcijomis.

- **SAS**

Programinės įrangos paketas sukurtas labiau patyrusiems specialistams, taip pat verslo rizikos faktorių veiksmų, duomenų valdymo analizėms. Su šiuo paketu galima atlikti tokias užduotis kaip duomenų įvedimo, paieškos ir valdymo sprendimus, grafinis vaizdavimas, statistinė ir matematinė analizė, programų kūrimas ir dar daugybę kitų naudingų funkcijų.

- **MS Office Excel**

Universali skaičiuoklė, kurios pagalba nesudėtingai galima atlikti daug ir įvairių paskaičiavimų, grafinių atvaizdavimų. Galima šią programą apjungti su MS Access duomenų bazėmis, tuomet lengviau ir patogiau tvarkoma informacija.

#### **2.4.1. NAUDOTŲ PROGRAMŲ BIBLIOTEKOS IR PROCEDŪROS**

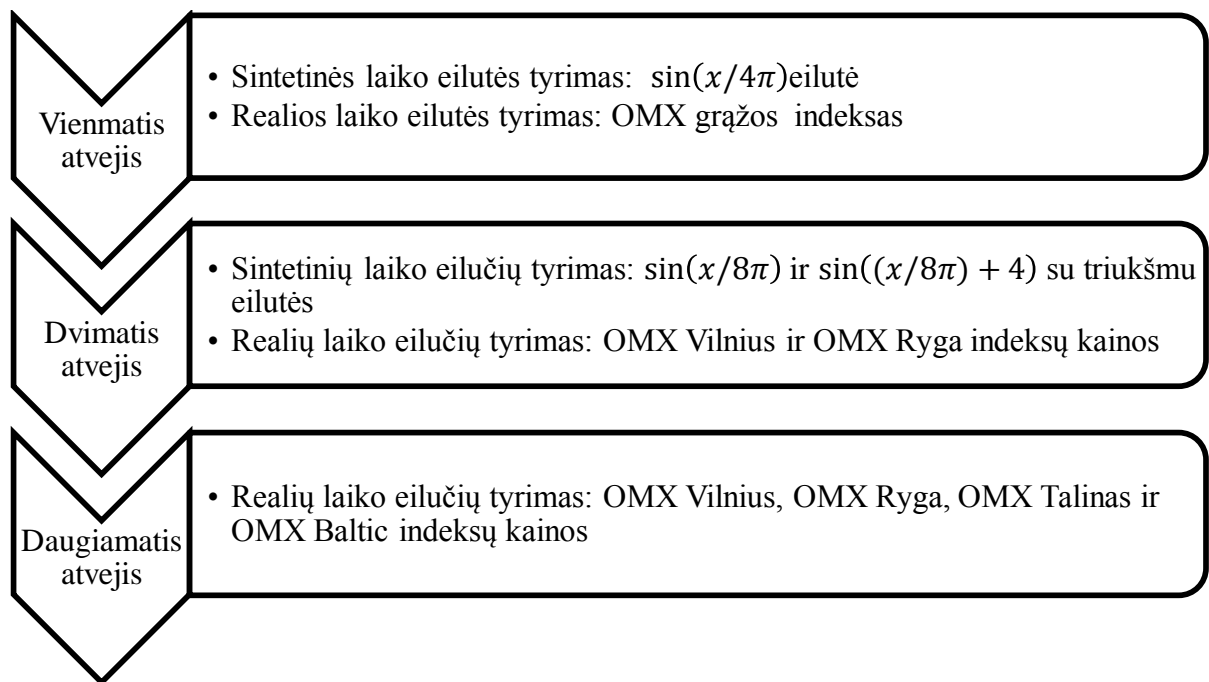
- `library(dtw)` – susideda iš visapusiško dinaminio laiko skalės iškraipymo algoritmo įgyvendinimo R Studio programoje. Ši biblioteka leidžia apskaičiuoti optimalų atstumą tarp dviejų laiko eilučių pasinaudojant sukauptą įverčių matricą. Taip pat šiai bibliotekai priklauso populiariausių lokalių ir globalių apribojimų parinktys, slopinimo koeficiento pritaikymas, iškraipymo funkcijos atvaizdavimas (trimatis, dvimatis ir tankio grafikai).
- `library(quantmod)` – kiekybinio finansinio modeliavimo stuktūros biblioteka R Studio programoje. Ją sudaro greitų finansinių prototipų aplinka su įrankiais skirtais duomenų apdorojimui, vaizdavimui. Taip pat ši aplinka padeda aiškiai ištyrinėti ir sukurti galimus prekybinius modelius.
- `library(xlsx)` – šis paketas suteikia programinę galimybę nuskaityti MS Excel tipo dokumentus ir tokiu būdu sukurti duomenų struktūrą programoje R Studio. Taip pat pasinaudojant šia biblioteka galima įrašyti naujai susikurtas duomenų struktūras ar rezultatų išeitį į MS Excel dokumentą.
- `library(zoo)` – šis paketas skirtas nagrinėti nereguliaraus laiko eilutes (angl. *irregular time series*). Paketas priklauso S3 metodų klasei programoje R Studio ir plačiai naudojamas indeksų priskyrimui ar sutapatinimui laiko eilutėse arba jų vektoriuose, matricose.
- `proc_simmilarity` – statistinės programos SAS procedūra leidžianti apskaičiuoti panašumus susijusius su laiko eilutėmis ar duomenimis išmatuotais laike. Pagal įvesties seką procedūra su pasirinktais įrankiais apskaičiuoja panašumo metriką su tiksline seka, pateikia skaitinius ir grafinius rezultatus. Taip pat ši procedūra tinkama apdoroti ir dirbti su dideliais duomenų kiekiais, todėl analizės rezultatai atliekami pakankamai greitai ir efektyviai.

## **2.5. APIBENDRINIMAS**

Šiame skyriuje buvo pristatyti svarbiausi dinaminio laiko skalės iškraipymo algoritmo teoriniai aspektai, kurie bus panaudojami tiriamojoje dalyje. Svarbiausiu prioritetu laikysime trumpiausią sutapatavimo kelią, tenkinantį pagrindinius apribojimus. Taip pat bus atsižvelgta ir į trajektorijos ir kaštų statistikas, atliktą sekų sutapatinimą.

### 3. TIRIAMOJI DALIS

Pagrindinis darbo tikslas – ištirti ir pritaikyti dinaminio laiko skalės iškraipymo algoritmą sintetinėms ir finansinėms laiko eilutėms. Šiame skyriuje bus pristatyta atlikto tyrimo analizė ir jos eiga, pagal kurią buvo ištirtos sintetinės bei realios duomenų imtys, pritaikyti tinkamiausi metodo nustatymai. Patį tyrimą ir jo vykdymą galima suskirstyti kaip pavaizduota 3.1 paveikslėlyje.



3.1.pav. Tiriomojo darbo eigos schema

3.1 skyriuje pristatomi duomenys, jų apibūdinimai ir reikšmės. 3.2.1 skyriuje aprašomas atliktas vienmačio atvejo tyrimas sintetinei  $\sin(x/4\pi)$  eilutei siekiant išsiaiškinti dinaminio laikos skalės iškraipymo algoritmo pritaikymą. Po to šis metodas panaudojamas OMX gražos indekso tyrimui. 3.3.1 skyriuje siekiama išsiaiškinti dviejų laiko eilučių  $\sin(x/8\pi)$  ir  $\sin((x/8\pi) + 4)$  panašumą bei tokiu būdu nustatyti, kurie lokalūs ir globalūs apribojimai yra tinkamiausi tokio pobūdžio duomenims. Patikrinama, ar turimi duomenys tenkina tris pagrindines apribojimo sąlygas, surandamas tinkamiausias atstumas tarp dviejų sekų. Tokia pačia schema DLSI algoritmas pritaikomas OMX Ryga ir OMX Vilnius indeksų kainos. Skyriuje 3.4 aprašomas daugiamačių laiko eilučių tyrimas su indeksu OMX Vilnius, OMX Ryga, OMX Talinas ir OMX Vilnius uždaromosiomis dienos kainomis. Siekiama išsiaiškinti, ar DLSI algoritmas gali būti korektiškai pritaikytas keletos eilučių panašumo tyrimams.

### 3.1. DUOMENYS

Sintetinės duomenų eilutės buvo sugeneruotos su programa MS Excel ir jos funkcija  $\sin()$ . Realūs laiko eilučių duomenys buvo paimti iš Yahoo finance ir Nasdaqbaltic puslapių. Nagrinėjamų šalių indekso kainų skaičiavimas remiasi įtrauktų bendrovių akcijų pokyčiais per tam tikrą laikotarpį, todėl indeksų reikšmių apskaičiavimui yra naudojamos praėjusios sesijos akcijų kaina. Bendrasis Baltijos šalių akcijų lyginamasis kainų indeksas žymimas OMX Baltic. Jis yra apskaičiuojamas iš visų Baltijos šalių bendrovių, esančių prekybinių biržų sąrašuose. Šis indeksą sudaro pačios likvidžiausios ir turinčios didžiausią kapitalizaciją bendrovių akcijos, kurios klasifikuojamos pagal tam tikrus veiklos sektorius. Bendrai OMX Baltic kainos indeksas parodo ir vertybinių popierių rinkos tendenciją. Pagal jį investuotojai turi galimybę sudaryti savo investicinį portfelį. Šiame indekse bendrovių akcijų svoris priklauso nuo kiekvienos bendrovės laisvų akcijų rinkoje kapitalizacijos. Likusiems OMX Vilnius, OMX Ryga ir OMX Talinas kainų indeksams taip pat priklauso analogiškų kriterijų bendrovių akcijos, tačiau įsikūrusios būtent toje šalyje.

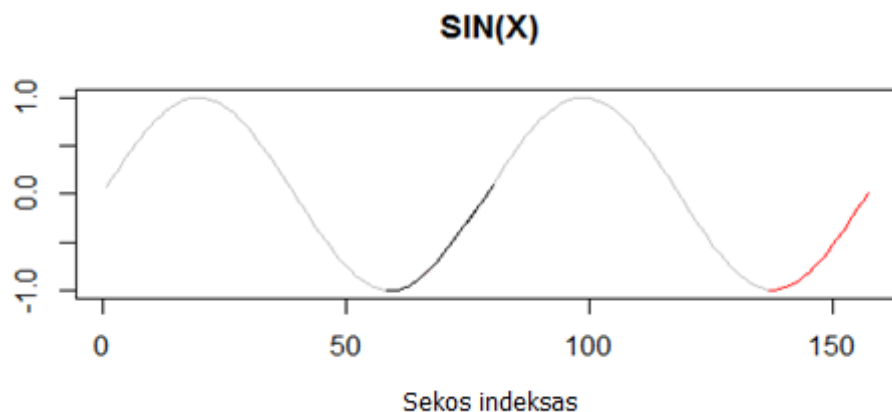
Šiuo metu OMX indeksui priklauso tokios didžiosios Lietuvos bendrovės kaip AB „Šiaulių bankas“, APB „Apranga“, AB „TEO LT“, AB „Klaipėdos nafta“, AB „Grigeo Grigiškės“, AB „Linas Agro Group“, AB „Rokiškio sūris“, AB „Pieno žvaigždės“, AB „Vilkyškių pieninė“ ir kitos.

### 3.2.VIENMATIS ATVEJIS

Šiame skyriuje nagrinėsime pasikartojančius panašumus vienoje laiko eilutėje. Iš pradžių DLSI algoritmą pritaikysime sintetinei laiko eilutei  $\sin(x/4\pi)$ , o vėliau pereisime prie realios laiko eilutės duomenų, t.y. OMX grąžos indekso.

#### 3.2.1. SINTETINĖS LAIKO EILUTĖS TYRIMAS: $SIN(X/4\pi)$ EILUTĖ

Turime  $\sin(x/4\pi)$  eilutę su periodu  $50\pi$ , o sekos ilgis lygus 158. Pati funkcija yra nesudėtinga, daugeliui žinoma ir lengvai panaudojama. Pabandykite surasti šios eilutės paskutinių 20 narių pasikartojimą per visą turimą laiko eilutę. Kadangi turimos eilutės periodas yra  $50\pi$ , vadinasi pradiniai imami nariai priklausys intervalui  $[43\pi; 50\pi]$ . Šių paskutinių 20 reikšmių pasikartojimas randamas prieš tai esančiame intervale  $[19\pi; 25\pi]$ . Tai iliustruojama žemiau esančiame grafike (**3.2 pav.**)

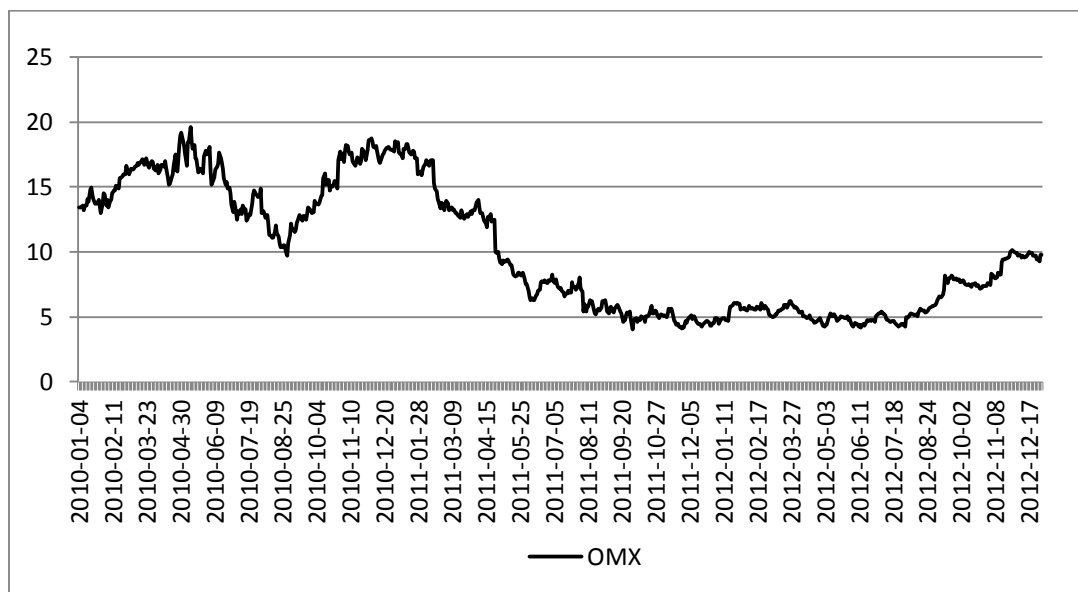


3.2.pav.  $\sin(x/4\pi)$  eilutės pasikartojimo radimas

Taigi iš gauto grafinio vaizdo galima teigti, kad DLSI algoritmas tinkamai susidorojo su sintetiniais duomenimis ir korektiškai surado panašių laiko eilutės reikšmių intervalus. Todėl galime pereiti prie realių duomenų laiko eilutės vienmačio tyrimo.

### 3.2.2. REALIOS LAIKO EILUTĖS TYRIMAS: OMX INDEKSAS

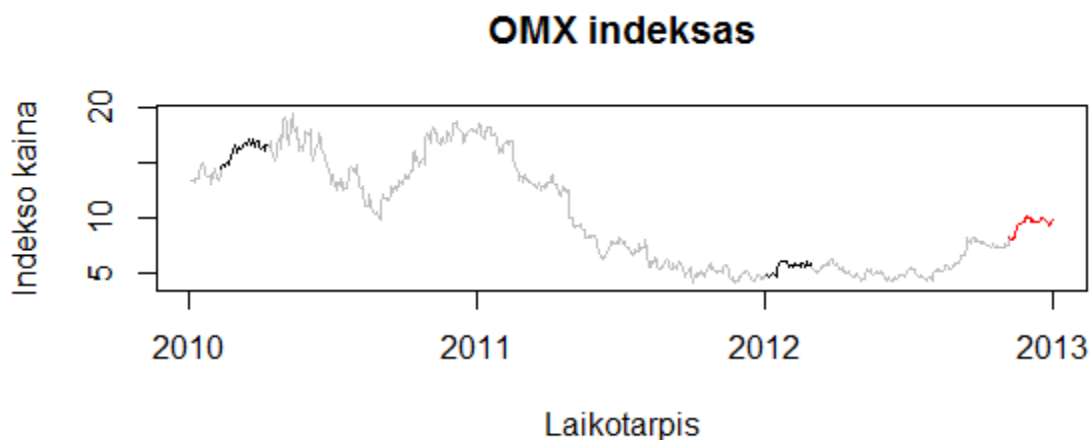
Nagrinėsime OMX grąžos indekso laiko eilutę, kurios tiriamas laikotarpis yra nuo 2010.01.01 iki 2013.01.01, iš viso 768 reikšmės. Grafiškai duomenys atrodo taip:



3.3.pav. OMX grąžos indekso laiko eilutės pradiniai duomenys

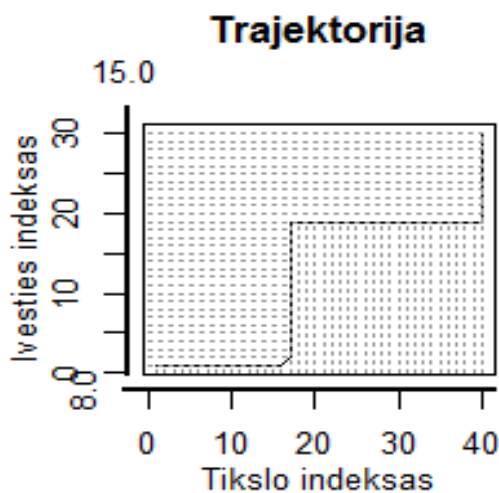
Pabandydysime surasti šios eilutės paskutinių 40 dienų iš 2012 – 2013 metų keletą panašių pasikartojimų per visą turimą 3 metų laikotarpį. Panašaus indekso istorija buvo rasta 2012 (2012.01.05 – 2012.02.29) ir 2010 (2010.02.11 – 2010.04.09) metų pradžiose (rezultatų lentelę galima rasti 1 priede). Raudona spalva grafike žymi ieškomą fragmentą, paryškintos juodos kreivės žymi rastus panašumus. Ieškomo fragmento indekso reikšmės svyravo nuo 8 iki 10, pirmojo rasto pasikartojimo 2010 metais indekso reikšmė buvo tarp 15 ir 17, o antrojo 2012

metais indekso reikšmė kito nuo 4 iki 6. Visų nagrinėtų laikotarpių indekso grąžai būdingas pakilimas per 2 vienetus.



3.4.pav. OMX grąžos indekso pasikartojimo radimas

Vieną iš šių surastų panašumų išnagrinėsime atskirai, t.y. paskutines 2012 metų 40 dienų su pirmomis 30 dienų. Pasirinktas skirtingas laiko eilučių ilgis, norint išbandyti vieną iš dinaminio laiko skalės iškraipymo algoritmo privalumų – skirtingų trukmių laiko eilučių sutapatinimą.



3.5.pav. OMX grąžos indekso 40 ir 30 dienų trajektorijos grafikas

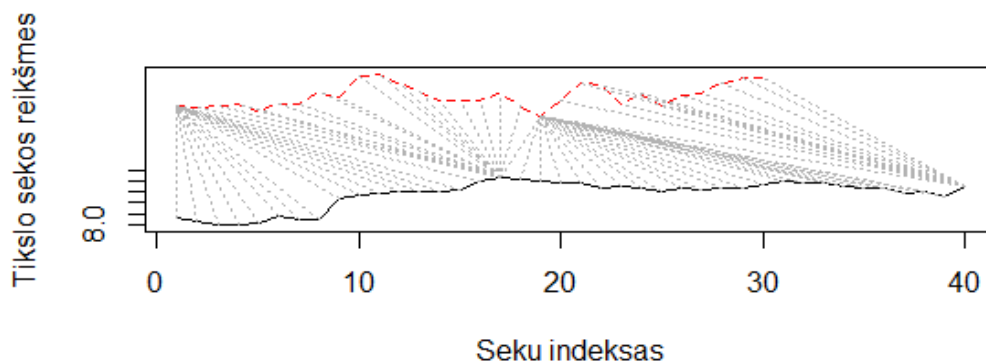
Sudaryta trajektorija turi tenkinti 3 pagrindines sutapatinimo kelio savybes: monotoniškumą, tęstinumą ir globalų išlyginimą (žr. 2.3 skyrių). Žingsnių atgal, peršokimų daugiau nei per vieną laiko skalę nėra. Trajektorijos pradžios ir pabaigos taškai sutampa su sutapatintų sekų taškais. Apskaičiuoti trajektorijos atstumai pagal kuriuos matoma, kad trajektorijos ilgis yra 265 su 67 žingsniais.



**1 lentelė.** Trumpiausios OMX indekso trajektorijos rezultatai su simetriniais apribojimais

Trajektorijos atstumas	Normalizuotas trajektorijos atstumas	Atlikti žingsniai
265	3.79	67

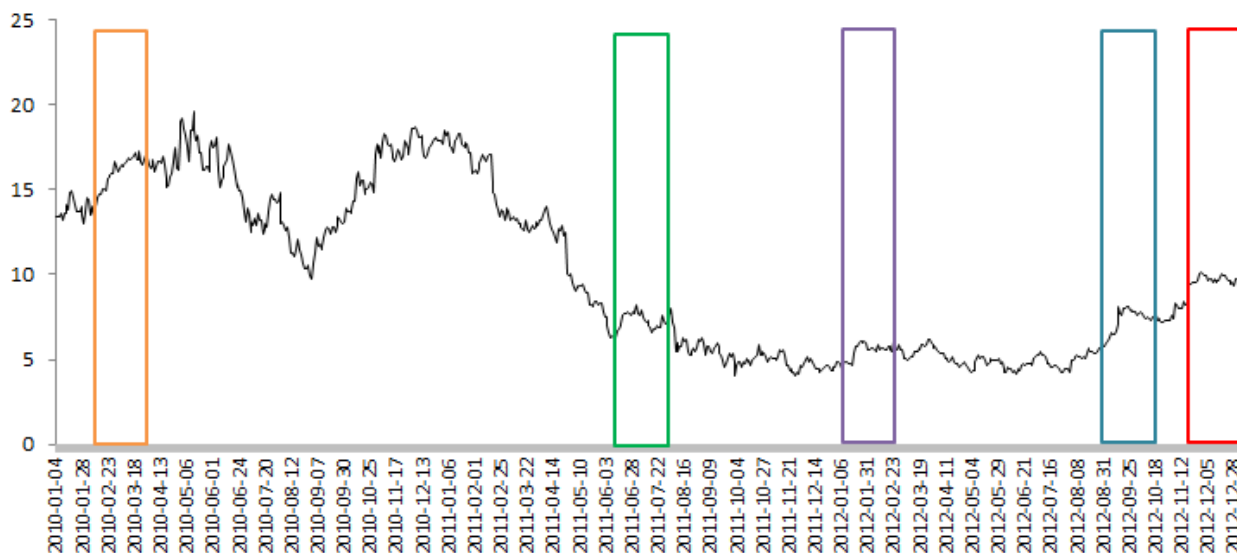
Dinaminio laiko skalės iškraipymo grafikas tarp tiriamų OMX gražos indekso sekos reikšmių (2012 metų paskutinių 40 dienų ir 2012 metų pirmųjų 30 dienų) vaizduojamas 3.6 paveikslėlyje. Horizontali ašis žymi įvesties (30 pirmųjų 2012 metų dienų) ir tikslo (40 paskutinių 2012 metų dienų) sekos indeksą  $k = \overline{1, N}$ , kur  $N = 40$ , kadangi  $N = \max(t_x, t_y)$ . Vertikali ašis žymi tikslo sekos reikšmes. Raudona trūki kreivė vaizduoja 30 pirmųjų 2012 metų dienas, t.y. seką  $y_j$ , kur  $j = 1, 30$ . Juoda ištisa kreivė vaizduoja 40 paskutinių 2012 metų dienų, t.y. seką  $x_i$ , kur  $i = 1, 40$ . Linijos sujungiančios šių dviejų sekų reikšmes, žymi sutapimus tarp jų per visą trajektoriją:  $(i_p, j_p) = \text{path}(p)$ , kai  $p = 1, 40$ .



**3.6.pav.** OMX gražos indekso dinaminiai laiko skalės iškraipymo grafikas

Palyginus trajektorijos grafiką su dinaminio laiko skalės iškraipymo grafiku galima pastebėti, kad šioms sekoms yra būdingas vieno taško su keletu taškų jungimas (angl. *one to many mapping*).

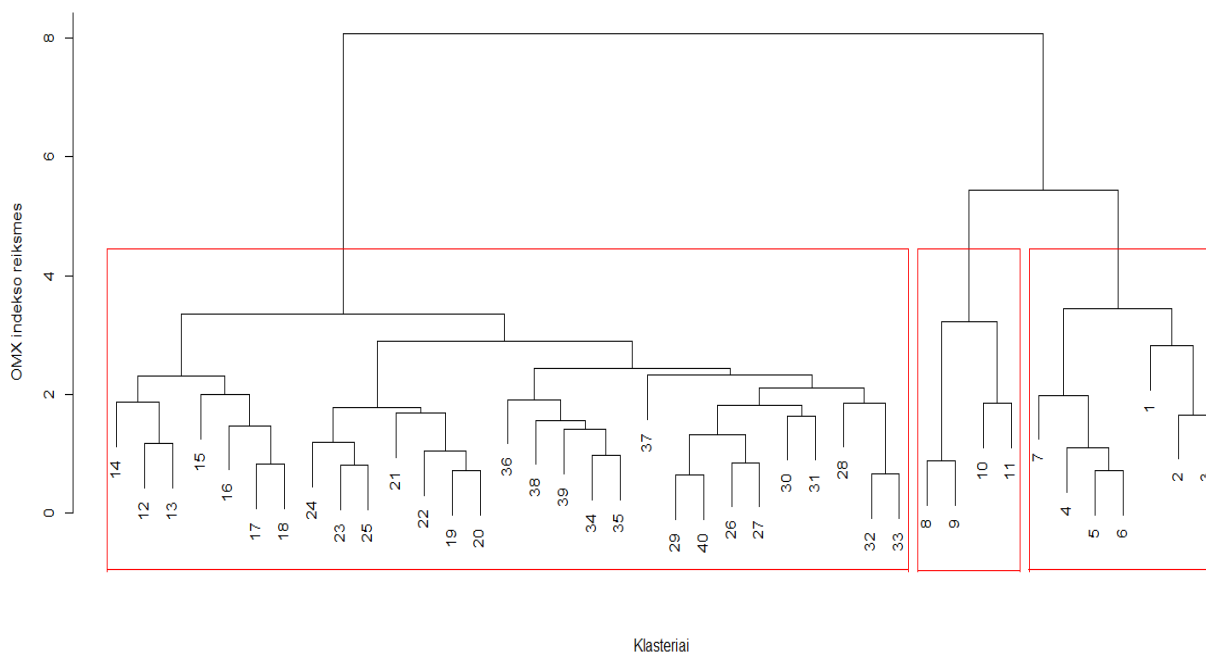
Tolimesniam tyrimui pabandykime surasti keletą papildomų savo forma panašių istorinių pasikartojimų pagal paskutines 40 dienų 2012 metais. Šiam tikslui buvo surasti dar trys pasikartojimai istoriniuose duomenyse, kuriuos sujungsime su anksčiau rasta 2 panašumais ir iš viso turėsime 5 panašius savo forma laiko eilutės fragmentus (detalesius duomenis galima rasti 2 priede).



**3.7.pav.** OMX grąžos indekso 5 pasikartojimų radimas

Aukščiau pavaizduotame grafike vaizduojami surasti penki panašūs OMX grąžos indekso laiko eilutės fragmentai, čia horizontali ašis žymi laiko trukmę, o vertikali grąžos indekso reikšmes. Matome, kad rasti panašūs laiko eilutės pasikartojimai apima gana skirtingus laikotarpius, t.y. 2010 metų pradžią, 2011 metų vidurį ir 2012 metų pradžią bei pabaigą. Be to šiems fragmentams matomas ir gana skirtingos OMX grąžos indekso reikšmės, kurios svyruoja nuo 5 iki 15. Norint detaliau išsiaiškinti šių rastų sutapatinimų panašumus panaudosime hierarchinį klasterizavimą. Klasterizavimo metodas padės mums suskirstyti turimus objektus į tam tikras grupes, kadangi jokios papildomos informacijos apie galimus klasterius neturime, todėl gauti rezultatai padės detaliau išsiaiškinti kokių dar formos panašumų jos turi.

Panaudojus hierarchinio klasterizavimo metodą, paremtą dinaminio laiko skalės iškraipymo algoritmo kaštų matrica tarp 5 laiko eilučių, buvo gauti trys pagrindiniai klasteriai. Gauti grafiniai rezultatai vaizduojami 3.8 paveikslėlyje, čia horizontali ašis žymi visų laiko eilučių indeksus (40 dienų), o vertikali – OMX grąžos indekso reikšmes. Matome, kad pirmojo klasterio trukmė 7 dienos, antrojo – 4 dienos, o trečiojo – 29 dienos.



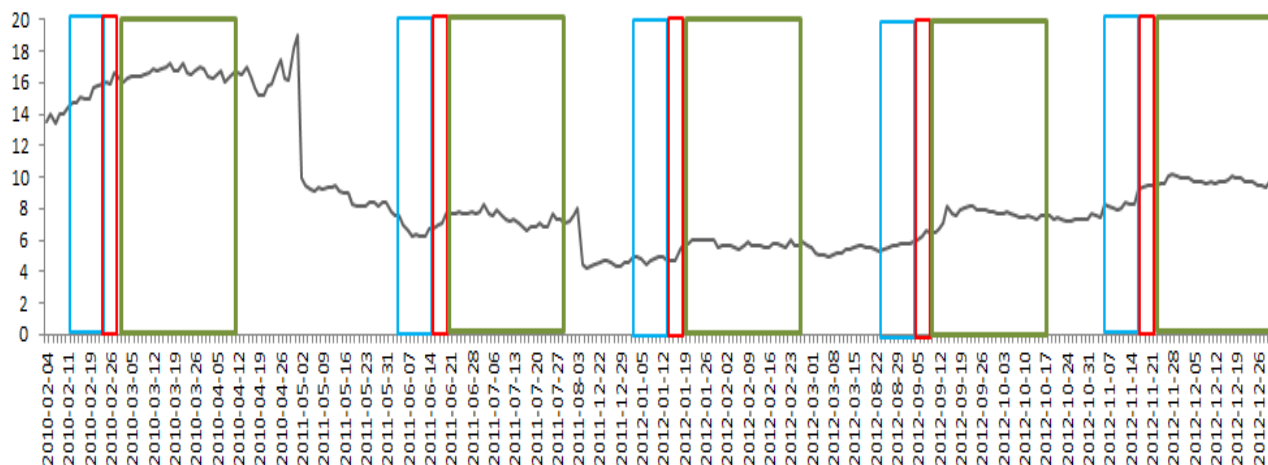
**3.8.pav.** OMX grąžos indekso istorinių pasikartojimų klasterių dendograma

Vadinasi visų nagrinėjamų laiko eilučių pasikartojimų pradžia būdingas 7 dienų nežymus padidėjimas arba sumažėjimas. Grąžos indekso skirtumas tarp visų pirmojo klasterio eilučių pradžios ir pabaigos svyruoja nuo -1.2 iki 1.16. Antrajam klasteriui, kurį sudaro 4 dienos, priskiriami vien tik OMX grąžos indekso reikšmės augimai, indekso reikšmės pakyla nuo 0.19 iki 1.07. Trečiąjį ilgiausios trukmės (29 dienos) klasterį sudaro tokie laikotarpiai, kuriems būdingas grąžos indekso pakilimas ir po to grįžimas į panašią pradinę reikšmę.

**2 lentelė.** OMX grąžos indekso istorinių duomenų klasterių parametrų reikšmės

Fragmentai	Klasteriai								
	1			2			3		
	Laikotapis	Pradinė indekso reikšmė	Paskutinė indekso reikšmė	Laikotapis	Pradinė indekso reikšmė	Paskutinė indekso reikšmė	Laikotapis	Pradinė indekso reikšmė	Paskutinė indekso reikšmė
F1	2010.02.11 – 2010.02.22	14.52	15.68	2010.02.23 – 2010.02.26	15.78	15.97	2010.03.01 – 2010.04.09	16.63	16.69
F2	2011.06.03 – 2011.06.13	7.5	6.3	2011.06.14 – 2011.06.17	6.67	7.11	2011.06.20 – 2011.07.29	7.64	7.08
F3	2012.01.05 – 2012.01.13	4.79	4.73	2012.01.16 – 2012.01.19	4.73	5.8	2012.01.20 – 2012.02.29	5.81	5.6
F4	2012.11.07 – 2012.11.15	8.17	8.23	2012.11.16 – 2012.11.21	9.22	9.49	2012.11.22 – 2013.01.01	9.49	9.76
F5	2012.08.24 – 2012.09.03	5.36	5.81	2012.09.04 – 2012.09.07	5.93	6.57	2012.09.10 – 2012.10.18	6.49	7.58

Žemiau esančiame grafike vaizduojama OMX grąžos indekso eilutė su sudarytais 3 klasteriais, kurie yra būdingi 5 savo forma panašiams istoriniams fragmentams. Horizontali ašis žymi laikotarpį, vertikali grąžos indekso reikšmes. Taip pat skirtingomis spalvomis vaizduojami ir klasteriai, čia mėlyna spalva pažymėtos sritys priklauso pirmajam klasteriui, raudona spalva priklauso antrajam, o žalia – trečiajam.



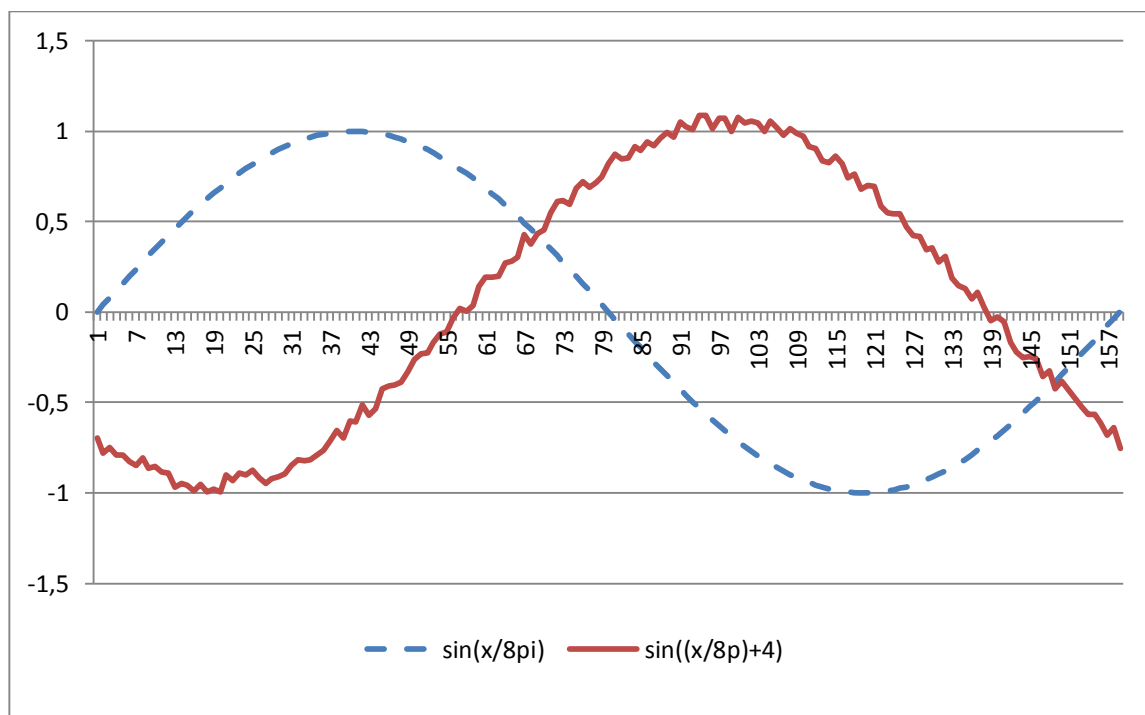
**3.9.pav.** OMX grąžos indekso istorinių pasikartojimų klasteriai

Išnagrinėjus OMX grąžos indekso laiko eilutę suradome kelis panašius pasikartojimus turimame 3 metų laikotarpyje, t.y. pasirinkus paskutines 40 dienų 2012 metais, surasti indekso istorijos panašumai 2012 ir 2010 metų pradžiose. Detaliau išnagrinėjus 2012 metų paskutines 40 dienų ir 2012 metų pirmąsias 30 dienų gautas vieno taško į kelis taškus atvaizdavimas. Trajektorijos ilgis lygus 265, jai sudaryti reikėjo atlikti 67 žingsnius. Šiame tyrime DLSI algoritmas parodė vieną iš savo privalumų, skirtingos trukmės laiko eilučių korektišką sutapatinimą. Žiūrint iš finansinės pusės OMX grąžos indeksui per visą turimą laikotarpį nėra būdinga panaši elgsena laike. Iki 2011 metų jis ypač svyravo, o nuo šių metų pradžios pastebimas ir grąžos kritimo atsiradimas, todėl kintamumas tarp 2012 metų pradžios ir pabaigos yra taip pat padidėjęs. Šiuo laikotarpiu investuotojai pradeda jautriau reaguoti į neigiamą informaciją, nei į teigiamą, todėl grąžos indeksas pradeda didėti tik nusistovėjus akcijų kainoms rinkoje. Surastuose indekso panašumų laikotarpiuose būdingas grąžos padidėjimas per 2 vienetus.

Pasinaudojant hierarchinio klasterizavimo ir dinaminio laiko skalės iškraipymo algoritmo kaštų matrica buvo suklastertizuoti 5 panašūs laiko eilutės fragmentai. Pagal šių eilučių fragmentus vėliau sudaryti 3 klasteriai, kuriems būdinga panaši OMX grąžos indekso elgsena arba laiko eilutės forma.

### 3.3.DVIMATIS ATVEJIS

Šiame skyriuje nagrinėsime panašumus tarp dviejų laiko eilučių. Iš pradžių DLSI algoritmą pritaikysime žinomoms imtims, t.y.  $\sin(x/8\pi)$  ir  $\sin((x/8\pi) + 4)$  su triukšmu laiko eilutėms, kurių sekos ilgis lygus 158. Vėliau pereisime prie realių duomenų nagrinėjimo, t.y. OMX indekso. Pradiniai sintetinės imties duomenys pavaizduoti **3.7 pav.**, čia tikslo ( $\sin(x/8\pi)$ ) eilutė vaizduojama mėlyna (trūkia) linija, o įvesties ( $\sin((x/8\pi) + 4)$ ) eilutė – raudona (ištisa) linija.



**3.10.pav.**  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  laiko eilutės

#### 3.3.1. SINTETINIŲ LAIKO EILUČIŲ: $\sin(x/8\pi)$ EILUTĖS IR $\sin((x/8\pi) + 4)$ EILUTĖS SU TRIUKŠMU TYRIMAS

Taigi turime dvi laiko eilutes  $\sin(x/8\pi)$  ir  $\sin((x/8\pi) + 4)$ , pabandysime surasti panašumus tarp jų pritaikant dinaminio laikos skalės iškraipymo algoritmą. Iš pradžių norint suprasti šio algoritmo globalių ir lokalių apribojimų veikimą, trajektoriją tarp turimų sekų pabandysime surasti pritaikant asimetrinius ir simetrinius apribojimus. Žingsnio parinkimas yra gana svarbus DLSI algoritme, kadangi tokiu būdu mes užduodame leidžiamus perėjimus per kaštų matricią ieškant minimalaus atstumo tarp dviejų sekų.

Pradėsime nuo asimetrinių apribojimų, leidžiamas trajektorijos nukrypimas yra suvaržytas slopinimu tarp 0 ir 2. Parametras  $P$  nurodo reikšmės  $x$  slopinimo koeficientą: 0 – slopinimo nėra, 0.5, 1, 2 – atitinkamo dydžio slopinimas pagal Sakoe ir Chiba. Sutapatinimai tarp dviejų sekų turėtų nurodyti, kad įvesties sekos laiko eilutės elementai bus atvaizduoti tik į vieną tikslinės

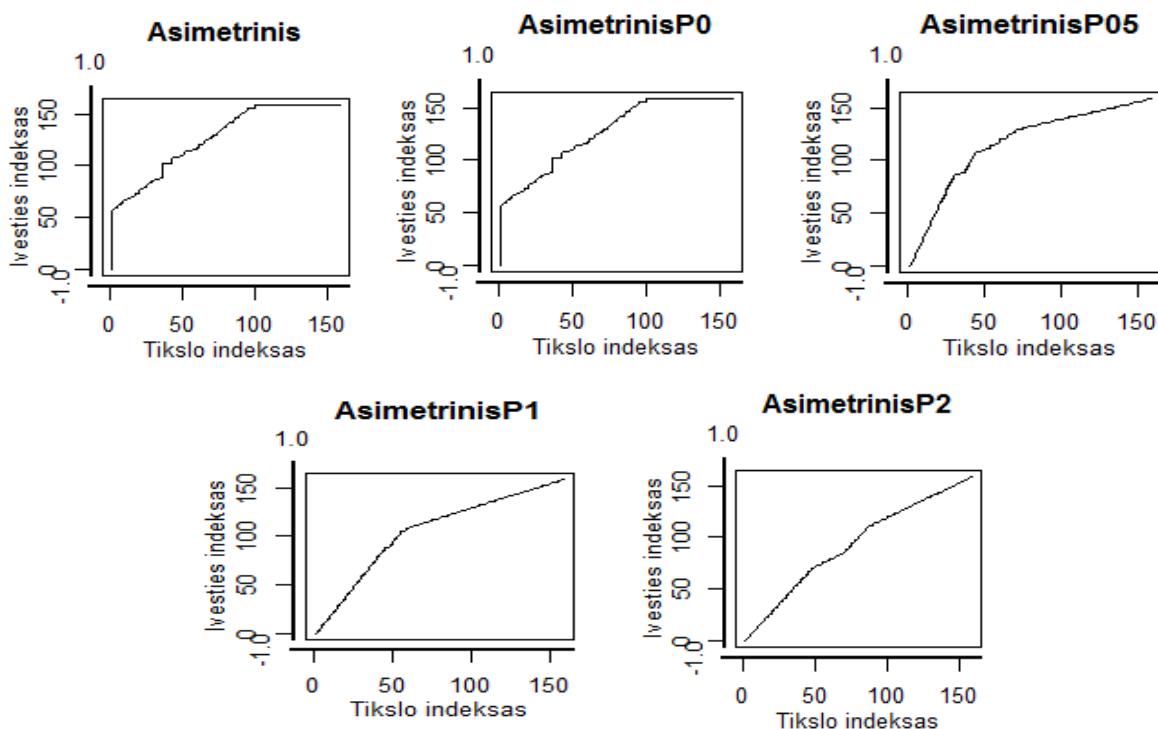
sekos elementą. Todėl sutapatinimo kelias turėtų bus būti vienareikšmis, normalizuotas pagal turimų sekų ilgį  $N = 158$ . Taigi buvo gauti tokie rezultatai:

**2 lentelė.**  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  eilučių trajektorijos rezultatai su asimetriniais apribojimais

Apribojimai	Trajektorijos atstumas	Normalizuotas trajektorijos atstumas	Atlikti žingsniai
Asimetrinis	57	0.361	157
<b>AsimetrinisP0</b>	<b>16</b>	<b>0.102</b>	<b>256</b>
AsimetrinisP05	64.9	0.411	84
AsimetrinisP1	110	0.699	105
AsimetrinisP2	145	0.917	64

Iš gautų rezultatų matome, kad mažiausias trajektorijos atstumas yra prie *AsimetrinisP0* apribojimo, jai gauti buvo atlikti 256 žingsniai. Ilgiausias atstumas turimas prie *AsimetrinisP2* apribojimo, kur buvo atlikti 64 žingsniai. Taigi galima teigti, kad trumpiausiai trajektorijai reikia atlikti kuo daugiau žingsnių ir tokiu būdu panaudoti kaip galima daugiau turimų sekų elementų porų.

Sudarytoms trajektorijoms prie asimetrinių apribojimų turi galioti ir pagrindinės 3 savybės: monotoniškumas, tęstinumas, globalus išlyginimas. Apžvelgus asimetrinių apribojimų sutapatinimo kelius (3.8 pav.) tarp dviejų sintetinių sekų, galima teigti, kad nei vienu atveju nebuvo pažeistas globalaus išlyginimo ir monotoniškumo apribojimas. Vadinasi abiejų sekų pradžios ir pabaigos taškai sutampa, nei viena trajektorija neturi žingsnio atgal. Taip pat tenkinamas ir tęstinumo apribojimas, t.y. neegzistuoja žingsnių peršokusių laiko skalę daugiau nei per 1 žingsnį.



**3.11.pav.**  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  eilučių trajektorijų su asimetrisiais apribojimais grafikai

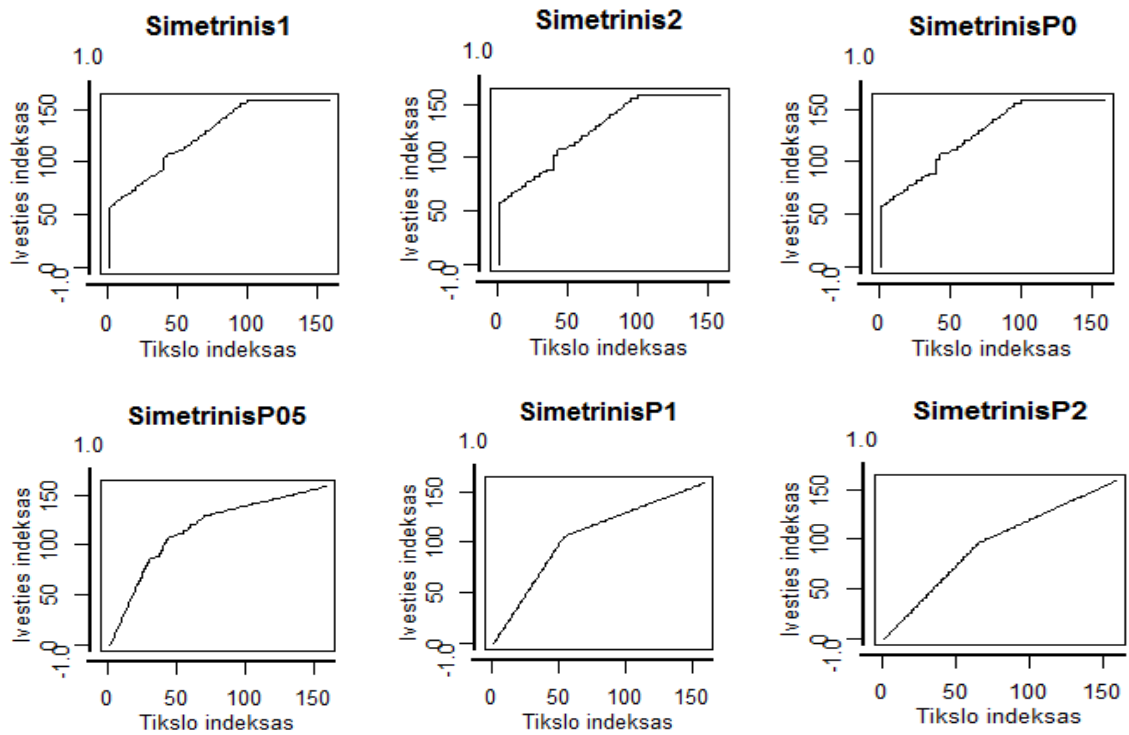
Toliau pereisime prie simetrinių apribojimų, čia *Simetrinis1* yra dažniausiai naudojamas pusiau simetrinis apribojimas be lokalių apribojimų ir be normalizavimo. *Simetrinis2* apribojimas yra normalizuotas, taip pat be lokalių apribojimų. Sekantys simetriniai apribojimai turi slopinimo parametą  $P$ , kuris nurodo laiko eilutės reikšmės  $x$  slopinimo koeficientą: 0 – slopinimo nėra, 0.5, 1, 2 – atitinkamo dydžio slopinimas pagal Sakoe ir Chiba. Trajektorijų su simetrisiais apribojimais rezultatai yra tokie:

**3 lentelė.**  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  eilučių trajektorijos rezultatai su simetrisiais apribojimais

Apribojimai	Trajektorijos atstumas	Normalizuotas trajektorijos atstumas	Atlikti žingsniai
<b>Simetrinis1</b>	<b>57.6</b>	<b>nėra</b>	<b>239</b>
Simetrinis2	58.5	0.185	261
SimetrinisP0	58.5	0.185	261
SimetrinisP05	148	0.469	83
SimetrinisP1	228	0.72	105
SimetrinisP2	294	0.929	64

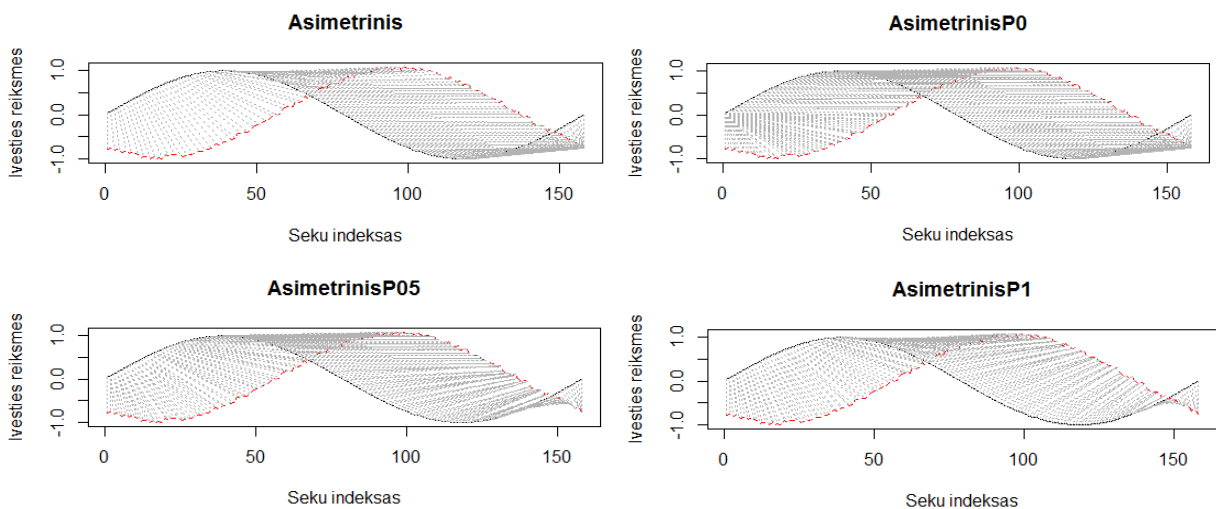
Trumpiausias sutapatavimo kelias lygus 57.6 buvo gautas su *Simetrinis1* apribojimu, šiam keliui sudaryti buvo atlikti 239 žingsniai. Ilgiausias sutapatavimo kelio atstumas sudarytas prie *SimetrinisP2* apribojimo ir buvo padaryti 64 žingsniai. Kaip ir asimetrinių apribojimų atveju, trumpiausia trajektorija turi beveik daugiausia atliktų žingsnių tarp dviejų sekų.

Simetrinių apribojimų trajektorijos grafikai (3.9 pav.) rodo, kad nebuvo pažeistos trys pagrindinės sąlygos. Todėl sudarytos trajektorijos korektiškos ir teisingos.

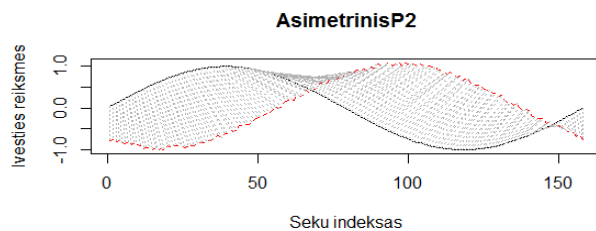


3.12.pav.  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  eilučių trajektorijų su simetrisiais apribojimais grafikai

Laiko skalės iškraipymai tarp tiriamų dviejų  $\sin(x/8\pi)$  sekų su asimetrisiais apribojimais yra vaizduojami 3.10 paveikslėlyje. Horizontali ašis parodo  $(\sin(x/8\pi))$  (tikslas) ir  $\sin((x/8\pi) + 4)$  (įvestis) sekos indeksą  $k = \overline{1, N}$ , kur  $N = 158$ , kadangi  $N = \max(t_x, t_y)$ . Juoda ištisa linija parodo  $\sin(x/8\pi)$  seką  $x_i$ , kur  $j = 1, 158$ . Raudona trūki kreivė parodo  $\sin((x/8\pi) + 4)$  seką  $y_j$ , kur  $i = 1, 158$ . Linijos, kurios sujungia sekų reikšmes, parodo sutapimus tarp  $\sin(x/8\pi)$  ir  $\sin((x/8\pi) + 4)$  sekų per visą trajektoriją:  $(i_p, j_p) = \text{path}(p)$ , kai  $p = 1, 158$ . Šis grafikas parodo laiko indekso iškraipymą atsižvelgiant į  $\sin(x/8\pi)$  ir  $\sin((x/8\pi) + 4)$  sekų vertes.

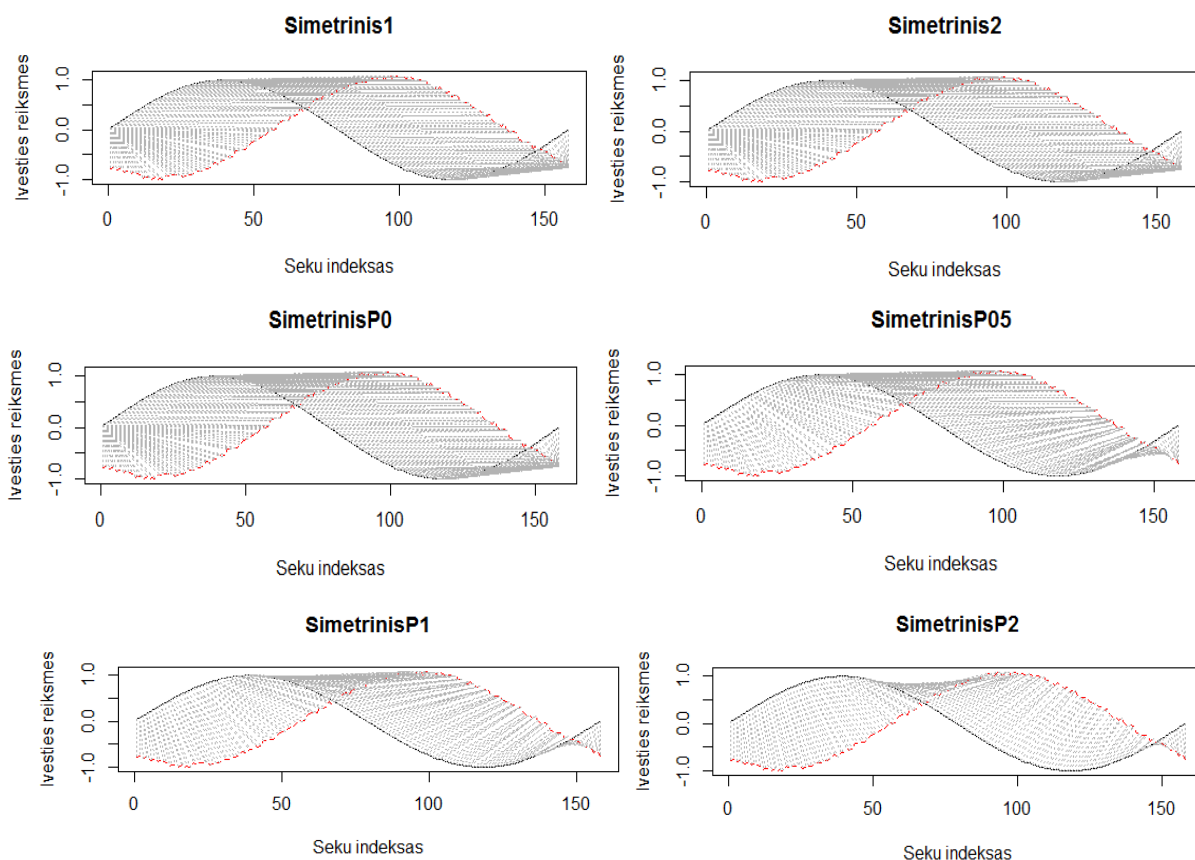






**3.13.pav.**  $\sin(x/8\pi)$  ir  $\sin((x/8\pi) + 4)$  eilučių su asimetrisiais apribojimais laiko skalės iškraipymo grafikai

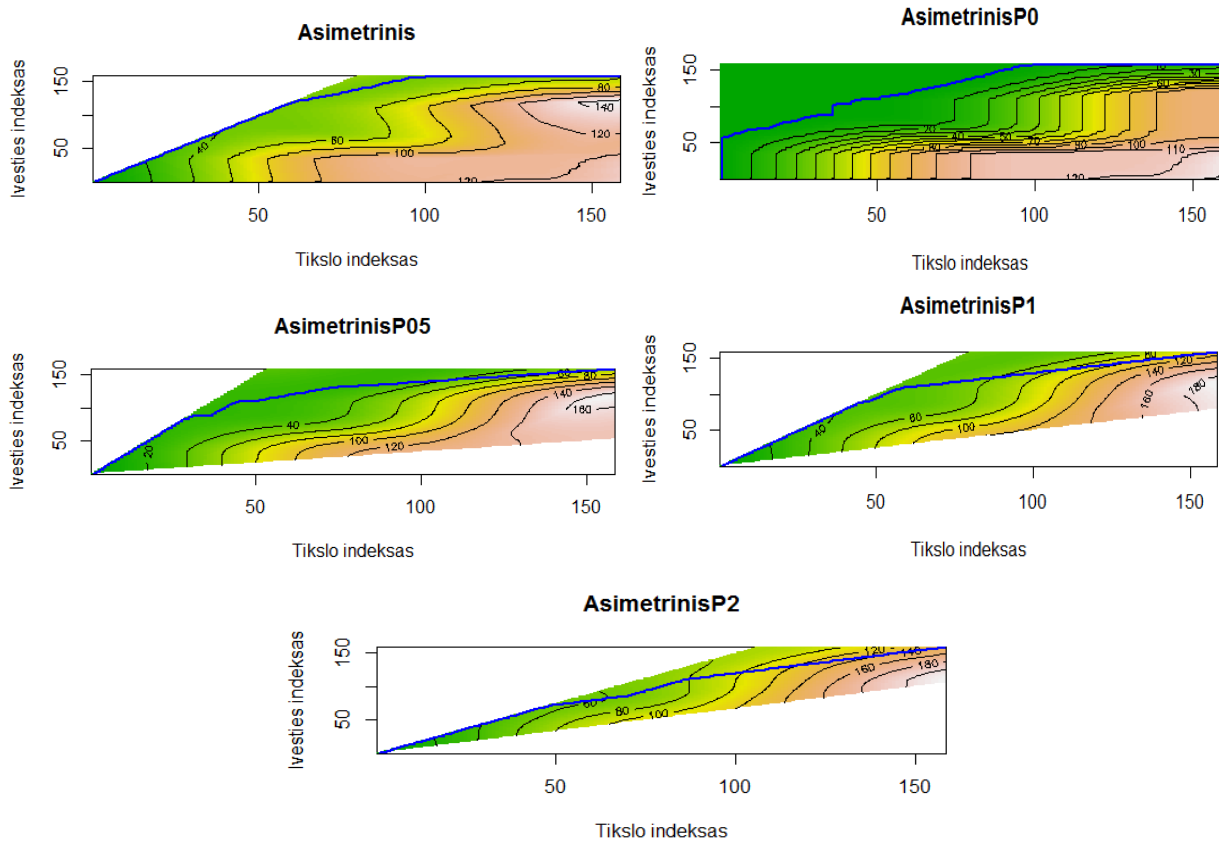
Dinaminio laiko skalės iškraipymai tarp  $\sin(x/8\pi)$  ir  $\sin((x/8\pi) + 4)$  sekų su asimetrisiais apribojimais yra vaizduojami 3.11 paveikslėlyje. Horizontali ašis parodo įvesties sekos ir tikslo sekų indeksą  $k = 1,158$ , vertikali ašis žymi tikslo sekos reikšmes. Juoda ištiesa kreivė parodo  $\sin(x/8\pi)$  seką  $x_i$ , o raudona trūki kreivė –  $\sin((x/8\pi) + 4)$  seką  $y_j$ .



**3.14.pav.**  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  eilučių su simetrisiais apribojimais laiko skalės iškraipymo grafikai

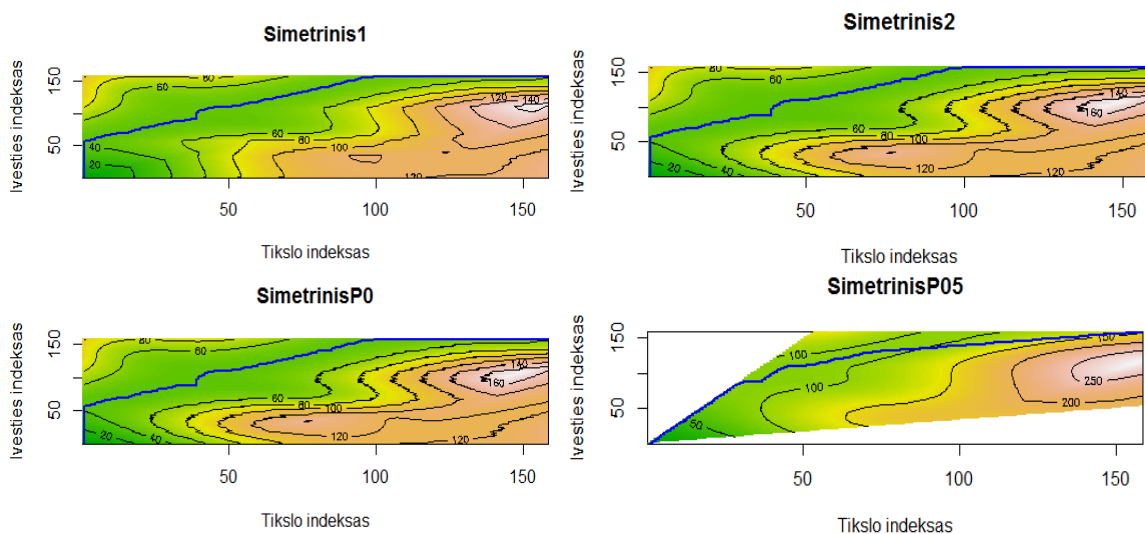
Trajektorijos su asimetrisiais apribojimais tankio grafikai vaizduoja kaštus, t.y. suspaudimus ir išsiplėtimus, su kuriais buvo sudarytas sutapatinimo kelias tarp dviejų sekų. Horizontali ašis žymi tikslo sekos indeksus, vertikali įvesties sekos. Mėlyna linija vaizduoja sudarytą trajektoriją per kaštų matricą, žali plotai atspindi nežymius suspaudimus ar išsiplėtimus, geltoni – vidutinius, o raudoni – didelius. Analizuojant tankio grafikus pagal spalvas galima nustatyti, kuriems apribojimams ir jų sudarytomis trajektorijoms buvo atlikta daugiausia

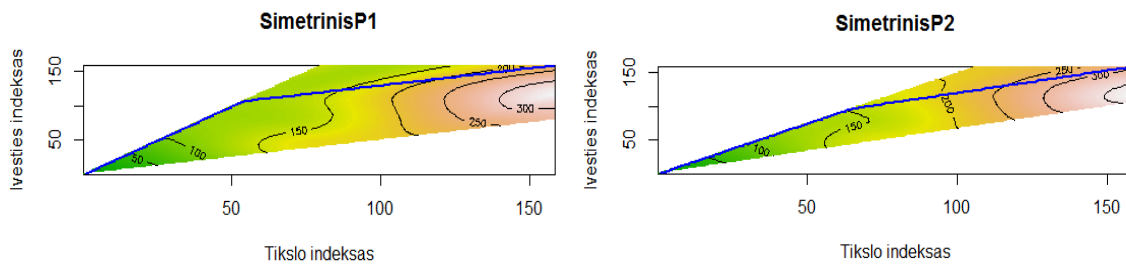
suspaudimų ar išsiplėtimų. Kaip matome daugiausia tiesioginių sujungimų buvo atlikta su apribojimu *AsimetrinisP0*, kadangi šiame grafike matoma mažiausia suspaudimų ar išsiplėtimų.



**3.15.pav.**  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  elučių su asimetriniais apribojimais trajektorijos tankio grafikai

Simetrinių apribojimų tankio grafikai yra gana skirtingi nuo asimetrinių apribojimų. Šiuo atveju prie visų simetrinių apribojimų buvo atlikta nemažai suspaudimų ar išsiplėtimų, tačiau esama ir tiesioginių sekų reikšmių sujungimų. Daugiausia kaštų pareikalavę apribojimai yra *SimetrinisP1* ir *SimetrinisP2*, o mažiausiai *Simetrinis1*.



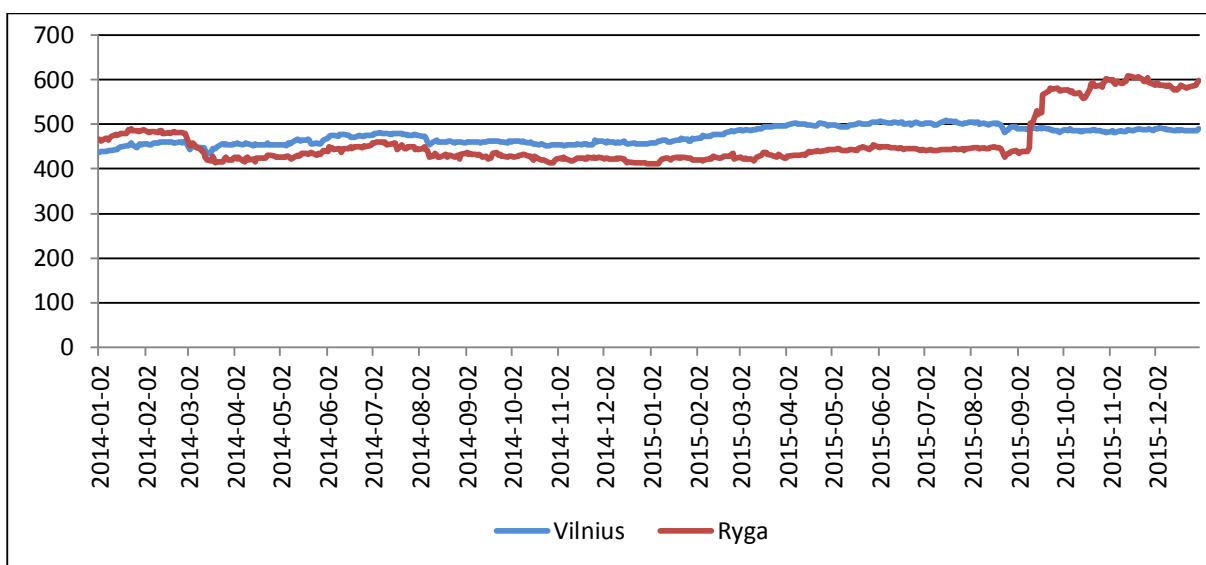


**3.16.pav.**  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  eilučių su simetriniais apribojimais trajektorijos tankio grafikai

Jeigu pagrindinis tikslas būtų surasti trumpiausią, mažiausiai kaštų reikalaujanti sutapatinimo kelią tarp  $\sin(x/8\pi)$  ir triukšmingos  $\sin((x/8\pi) + 4)$  laiko eilučių, neatsižvelgiant į jungiamus taškus, tuomet reikėtų pasirinkti *AsimetrinisP0* apribojimą. Tačiau mažiausi kaštai nevisuomet yra geriausias pasirinkimas. Atidžiau peržvelgus šio apribojimo sudarytą trajektoriją ir dinaminio laiko skalės iškraipymo grafiką galima pamatyti, kad gana daug tikslo sekos taškų yra atvaizduojami į vieną ar kelis įvesties sekos taškus (angl. *one to many mapping*). Todėl tam tikri įvesties sekos taškai nėra sujungiami su tikslo sekos. Jeigu prioritetu laikomas, kiekvienos tikslo sekos reikšmės atvaizdavimas į vieną įvestos sekos reikšmę (angl. *one to one mapping*), tuomet būtų renkamas tarp *AsimetrinisP2* ir *SimetrinisP2* apribojimų. Įvertinus šių apribojimų sutapatinimo kelius, reikėtų pasirinkti *AsimetrinisP2*, nes jo ilgis yra trumpesnis ir lygus 145.

### 3.3.2. REALIŲ LAIKO EILUČIŲ TYRIMAS: OMX VILNIUS IR OMX RYGA KAINŲ INDEKSAI

Nagrinėsime dviejų laiko eilučių OMX Vilnius ir OMX RYGA indeksų uždaromųjų dienos kainų laikotarpį nuo 2014.01.01 iki 2015.12.31, t.y. trukmė 2 metai, iš viso 502 reikšmės. Iš grafinio pirminių duomenų vaizdo galima matyti, kad OMX Vilnius indekso kainų reikšmės yra gana stabilios nagrinėjamame laikotarpyje, tačiau OMX Ryga indekso uždaromosios dienos kainos turi staigų šuolį aukštyn nuo 2015 metų rugsėjo mėnesio. Vis dėl to indekso kainų reikšmė metų pabaigoje nusistovi ir tampa gana stabilia. Vizualiai duomenų eilutės nėra labai skirtingos, tačiau OMX Ryga indekso kainos staigūs ir reikšmingi pasikeitimai 2015 metų pabaigoje tampa didesniu iššūkiu DLSI algoritmui.



3.17.pav. Pradiniai OMX Vilnius ir OMX Ryga indeksų duomenys

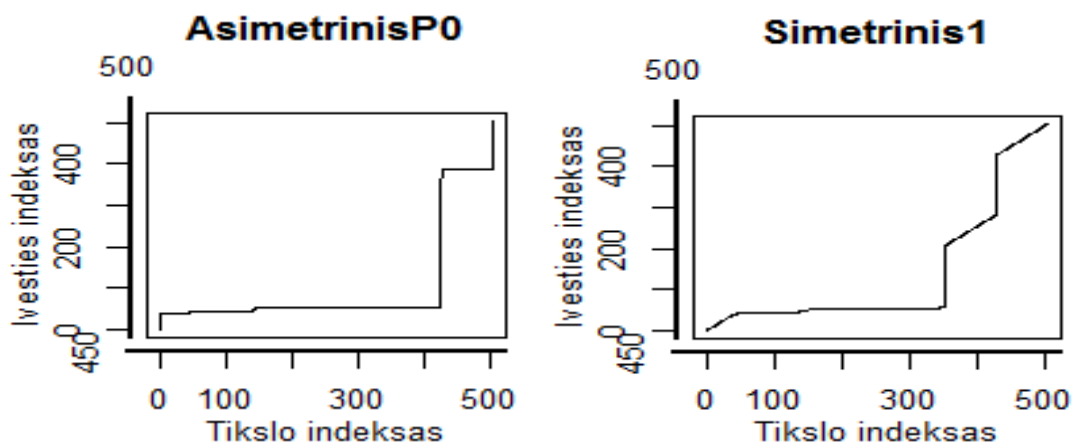
Kaip ir minėjome ankstesniame skyriuje sutapatavimo žingsnio parinkimas yra svarbus, kadangi tokiu būdu nustatomi leidžiami perėjimai per kaštų matricą ieškant minimalaus atstumo. Taip pat remiantis dviejų sintetinių imčių analize, naujai nagrinėjamiems duomenims pritaikysime tuos pačius asimetrinius ir simetrinius apribojimus. Taigi buvo gauti tokie rezultatai:

4 lentelė. OMX Vilnius ir OMX Ryga laiko eilučių trajektorijos rezultatai su apribojimais

Apribojimai	Trajektorijos atstumas	Normalizuotas trajektorijos atstumas	Atlikti žingsniai
Asimetris	14566	29	501
<b>AsimetrisP0</b>	<b>9321</b>	<b>18.6</b>	<b>937</b>
AsimetrisP05	16319	32.5	257
AsimetrisP1	18240	36.7	338
AsimetrisP2	20640	41.1	207
<b>Simetris1</b>	<b>14520</b>	<b>nėra</b>	<b>795</b>
Simetris2	15858	15.8	985
SimetrisP0	15858	15.8	985
SimetrisP05	37045	36.9	259
SimetrisP1	40682	40.5	337
SimetrisP2	43876	43.7	210

Trumpiausios trajektorijos atstumas yra prie *AsimetrisP0* apribojimo, jai gauti buvo atlikti 937 žingsniai. Sekantis mažiausias sutapatavimo kelias buvo gautas su *Simetris1* apribojimu, kelio ilgis 14520 su atliktais 796 žingsniais. Pati ilgiausia trajektorija lygi 43876 sudaryta su *SimetrisP2* apribojimu, kur buvo atlikti 210 žingsnių. Šiame tyrime taip pat matoma tendencija, jog trumpiausiai trajektorijai reikia atlikti kuo daugiau žingsnių.

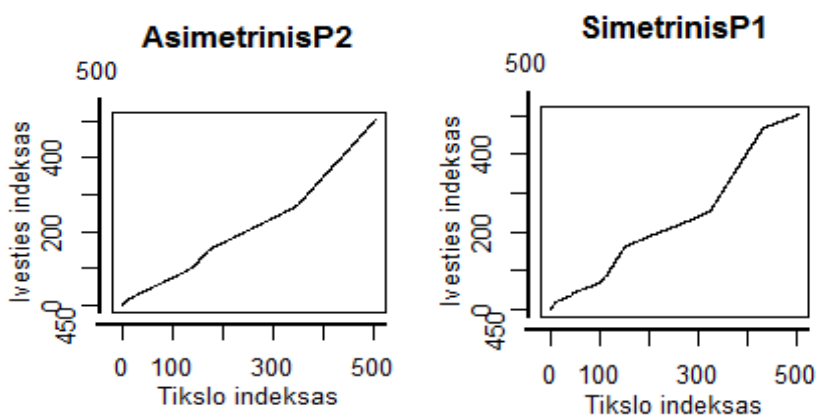
Sudarytomis trajektorijoms prie visų apribojimų turi galioti trys pagrindinės savybės: monotoniškumas, tęstinumas, globalus išlyginimas. Apžvelgus sudarytų apribojimų sutapatavimo kelius (3 priedas) tarp dviejų finansinių sekų nei vienu atveju nebuvo pažeistos apribojimų sąlygos. Taigi abiejų sekų pradžios ir pabaigos taškai sutampa, nėra atliktų žingsnių, kurie būtų peršokę laiko skalę daugiau nei per 1 žingsnį, nei viena trajektorija neturi žingsnio atgal. Dviejų trumpiausių trajektorijų grafikai yra tokie:



3.18.pav. OMX Vilnius ir OMX Ryga trajektorijos

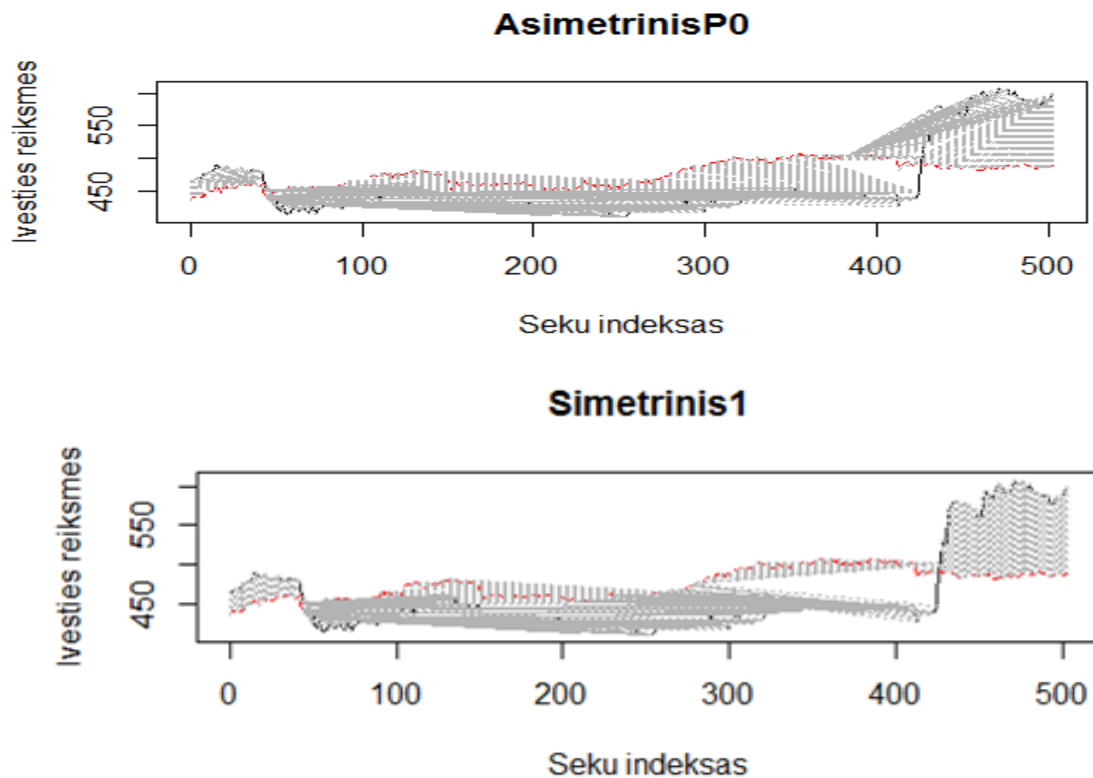
Iš *AsimetrinisP0* apribojimo sudaryto sutapatavimo kelio galima pastebėti, kad nuo 400 tikslinės (OMX Ryga) sekos indekso labai daug įvesties imties (OMX Vilnius) taškų yra atvaizduojama į šį tašką. Taip pat ir nuo 500 įvesties sekos indekso. Panaši situacija turima ir su *Simetrinis1* apribojimu, tik keičiasi įvesties sekos taškai, t.y. 350 indeksas ir 450 indeksas. Taigi šioms trajektorijoms būdingas vieno taško į kelis vaizdavimas.

Pabandykime apžvelgti tas trajektorijas, kurioms labiau būdingas vieno taško į vieną tašką vaizdavimas (3.16 pav.), tai būtų *AsimetrinisP2* ir *SimetrinisP1*. Pirmosios trajektorijos ilgis yra 20640, o antrosios – 40682. Akivaizdu, kad šio vaizdavimo trajektorija yra ilgesnė nei prieš tai aptarto, tačiau šiose trajektorijose yra atlikta daug mažiau vieno taško sutapatavimo į kelis taškus.



3.19.pav. OMX Vilnius ir OMX Ryga trajektorijos

Tolimesniame tyrime pasiliksime prie trumpiausių kelių gavusių apribojimų *AsimetrinisP0* ir *Simetrinis1*. Dinaminio laiko skalės iškraipymai tarp tiriamų OMX Vilnius ir OMX Ryga sekų su trajektorijomis (*AsimetrinisP0* ir *Simetrinis1*) yra vaizduojami 3.17 pav. (detalius grafikus galima rasti 4 priede). Horizontali ašis žymi OMX Ryga ir OMX Vilnius sekos indeksą, čia  $N = 502$ . Raudona trūki kreivė vaizduoja OMX Vilnius seką, juoda ištisa – OMX Ryga seką. Pilkos linijos jungiančios sutampančias sekų reikšmes, parodo sutapimus tarp tiriamų eilučių per visą trajektoriją.



**3.20.pav.** OMX Vilnius ir OMX Ryga dinaminio laiko skalės iškraipymo grafikai

Iš grafinio finansinių laiko eilučių sutapatavimo galima dar aiškiau pamatyti, kad apribojimui *AsimetrinisP0* ypač būdingas įvesties sekos vieno taško sujungimas su keliais tikslo sekos taškais. Siekiant sudaryti trajektoriją su apribojimu *AsimetrinisP0* ir atvaizduoti abiejų sekų laiko skalės iškraipymo grafikus buvo atlikti tokie jungimai:

**5 lentelė.** OMX Vilnius ir OMX Ryga trajektorijos su apribojimu *AsimetrinisP0* statistikos

Trajektorija	Skaičius	Trajektorijos %
Tiesioginiai sujungimai	67	7.150
Suspaudimai	435	46.42
Išplėtimai	435	46.42
Iškraipymai	870	92.85

Buvo atlikti 67 tiesioginiai braižymai, 870 laiko skalės iškraipymai, t.y. 435 suspaudimai ir 435 išsiplėtimai. Iš gautų rezultatų taip pat matyti, kad tiesioginių sujungimų buvo atlikta mažiau negu išsiplėtimų ir suspaudimų. Tiesioginiai sujungimai sudarytoje trajektorijoje sudaro 7% , o laiko skalės iškraipymai net 93%. Kadangi abiejų sekų ilgiai yra vienodi, todėl procentinės dalys sutampa, t.y. jungiami visi abiejų sekų taškai.

**6 lentelė. OMX Vilnius ir OMX Ryga trajektorijos su apribojimu Simetrinis1 statistikos**

Trajektorija	Skaičius	Trajektorijos %
Tiesioginiai sujungimai	209	26.29
Suspaudimai	293	36.86
Išplėtimai	293	36.86
Iškraipymai	586	73.71

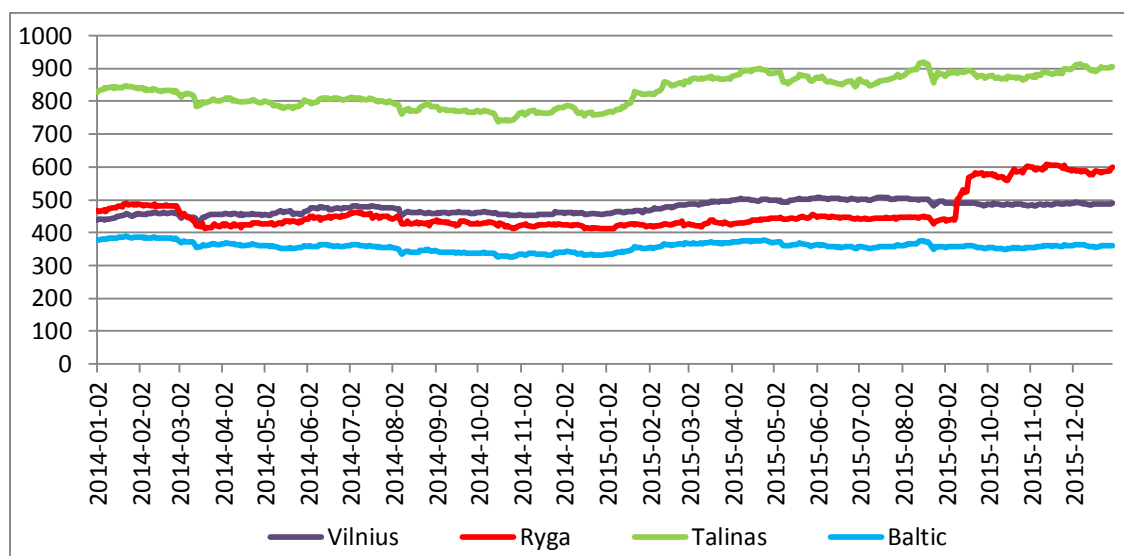
Iš trajektorijos su apribojimu *Simetrinis1* statistinių rezultatų matyti, kad buvo padaryti 209 tiesioginiai sujungimai. Be to trajektoriją sudaro 293 suspaudimai ir 293 išsiplėtimai. Vadinasi buvo atlikti 586 dinaminio laiko skalės iškraipymai, todėl tiesioginių sujungimų buvo atlikta dvigubai mažiau nei iškraipymų. Tačiau šiuo atveju tiesioginių sujungimų atlikta net 19% daugiau negu *AsimetrinisP0* apribojimo atveju.

Palyginus OMX Vilnius ir OMX Ryga uždaramųjų dienos kainų indeksus pasirinkome du tinkamiausius sutapatavimo atstumus. Vienas iš jų buvo pasiektas prie *AsimetrinisP0* apribojimo, kitas prie *Simetrinis1*. Nors pirmojo trajektorijos ilgis buvo trumpesnis, tačiau dėl didesnio atliktų tiesioginių sujungimų skaičiaus, pranašesniu laikysime antrąjį būdą, kurio trajektorijos ilgis yra 14520. Tirtos sekos, kaip ir buvo galima numatyti iš anksto tikrai yra panašios, kadangi vidutinė OMX Vilniaus indekso ir OMX Rygos indekso kaina svyruoja apie 400 – 500. Laikotarpis, kai OMX RYGA kainų indeksas išauga iki 600 nuo 2015 metų rugsėjo mėnesio šiek tiek galėjo padidinti trajektorijos atstumą, t.y. turėjo būti atlikta daugiau suspaudimų ar išsiplėtimų negu tiesioginių sujungimų. Tačiau bendru požiūriu OMX Vilnius ir OMX Ryga kainų indeksų elgsena yra tikrai panaši.

**3.4.DAUGIAMATIS ATVEJIS**

Šioje dalyje tirsime OMX Baltic indekso uždaramųjų dienos kainų panašumą su OMX Vilnius, OMX RYGA ir OMX Talinas indeksų kainomis. Taigi turimos keturios laiko eilutės, kurių nagrinėjamas laikotarpis yra nuo 2014.01.01 iki 2015.12.30, t.y. trukmė 2 metai, iš viso 502 atributai. Iš grafinio pirminių duomenų vaizdo galima matyti, kad OMX Talinas indekso kainų reikšmės yra žymiai didesnės negu likusių indeksų kainos. Vizualiai kitos duomenų eilutės nėra labai skirtingos, taip pat daugumos indeksų kainos yra stabilios nagrinėjamame laikotarpyje, išskyrus OMX Ryga indeksą (nuo 2015 metų rugsėjo mėnesio matomas staigus šuolis aukštyn).

Šiuo metu OMX Baltic indekso pagrindinį bendrovių sąrašą sudaro 30 pačių likvidžiausių Baltijos šalių bendrovių. Didžiausią šio indekso dalį užima Lietuvos ir Estijos bendrovės, kurios sudaro po 43.33 % , likusi 13.33% dalis priklauso Latvijos bendrovėms.



3.21.pav. OMX Vilnius, OMX Ryga, OMX Talinas ir OMX Baltic kainų indeksų pradiniai duomenys

Pirmiausiai patikrinsime ar tiriamos laiko eilutės yra tarpusavyje priklausomos, dėl to panaudosime Spirmeno koreliacijos koeficientą. Iš 7 lentelės matome, kad visi kintamieji yra stipriai tarpusavyje koreliuoti. Nulinė hipotezė, kad Spirmeno koreliacijos koeficientas lygus nuliui yra atmetama, kadangi gauta p reikšmė beveik visais atvejais yra mažesnė už 0.001. Stipriausią koreliaciją OMX Baltic indeksas sudaro su indeksu OMX Talinas, jų koeficientas lygus 0.58905. Silpniausia koreliacija OMX Baltic indekso koreliacija yra su OMX Vilnius indeksu, čia koreliacijos koeficientas lygus 0.20048. Vadinas galime teigti, kad indekso OMX Baltic kaina didėja tuomet, kai atitinkamai didėja likusių indeksų kainos, kadangi visi gauti koeficientai yra teigiami.

7 lentelė. OMX Baltic ir kitų OMX koreliacijos koeficientai

	OMX_Talinas	OMX_Ryga	OMX_Vilnius	OMX_Baltic
OMX_Talinas	1	0.62841	0.74232	0.58905
		<.0001	<.0001	<.0001
OMX_Ryga	0.62841	1	0.38026	0.3808
	<.0001		<.0001	<.0001
OMX_Vilnius	0.74232	0.38026	1	0.20048
	<.0001	<.0001		<.0001
OMX_Baltic	0.58905	0.3808	0.20048	1
	<.0001	<.0001	<.0001	



Pereikime prie dinaminio laiko skalės iškraipymo algoritmo taikymo. Laiko eilučių panašumo nustatymui buvo panaudotas asimetrinis apribojimas su slopinimo koeficientu 1. Taigi kiekvienai galimai porai su įvesties seka OMX Baltic buvo atlikti sutapatinimai per visą nagrinėjamų sekų ilgį  $N = 502$ , sudarytos trajektorijos yra:

**8 lentelė.** OMX Baltic ir kitų OMX indeksų trajektorijų statistikos

Tikslinė seka	Trajektorijos atstumas	Normalizuotas trajektorijos atstumas	Atlikti žingsniai
Vilnius	55624	111	502
<b>Ryga</b>	<b>48269</b>	<b>96.2</b>	<b>557</b>
Talinas	234536	467	336

Iš 8 lentelės matome, kad trumpiausias trajektorijos atstumas yra tarp OMX Ryga ir OMX Baltic indeksų kainų. Jis lygus 48269, kuriam atlikti reikėjo 336 žingsnių. Pats ilgiausias atstumas gautas su OMX Talinas kainų indeksu. Nors visų trajektorijų atliktų žingsnių skaičius gana panašus, tačiau pačių trajektorijų ilgiai tikrai skirtingi.

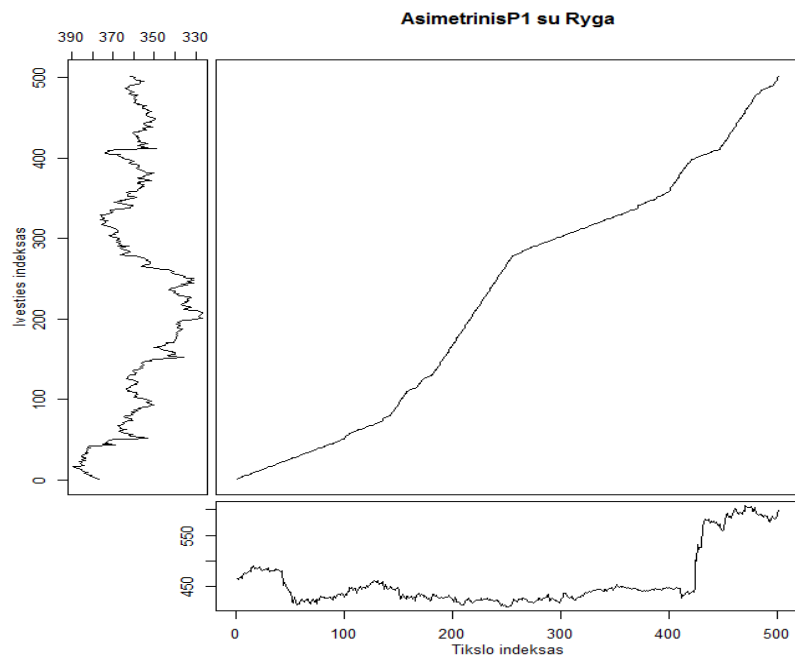
Pabandykime patikrinti nagrinėjamų sekų atstumus pasiremiant kitomis atstumo metrikomis kaip Euklido, Manhattan ir Minkovskio (su parinktimi  $p = 1$ ). Jas galima panaudoti dėl to, kad tiriamų sekų ilgiai yra vienodi.

**9 lentelė.** OMX Baltic ir kitų OMX indeksų atstumo metrikos

Tikslinė seka	Euklido atstumas	Manhattan atstumas	Minkovskio atstumas
Vilnius	2680.677	59170.51	59170.51
<b>Ryga</b>	<b>2609.655</b>	<b>51815.39</b>	<b>51815.39</b>
Talinas	10663.59	238080.3	238080.3

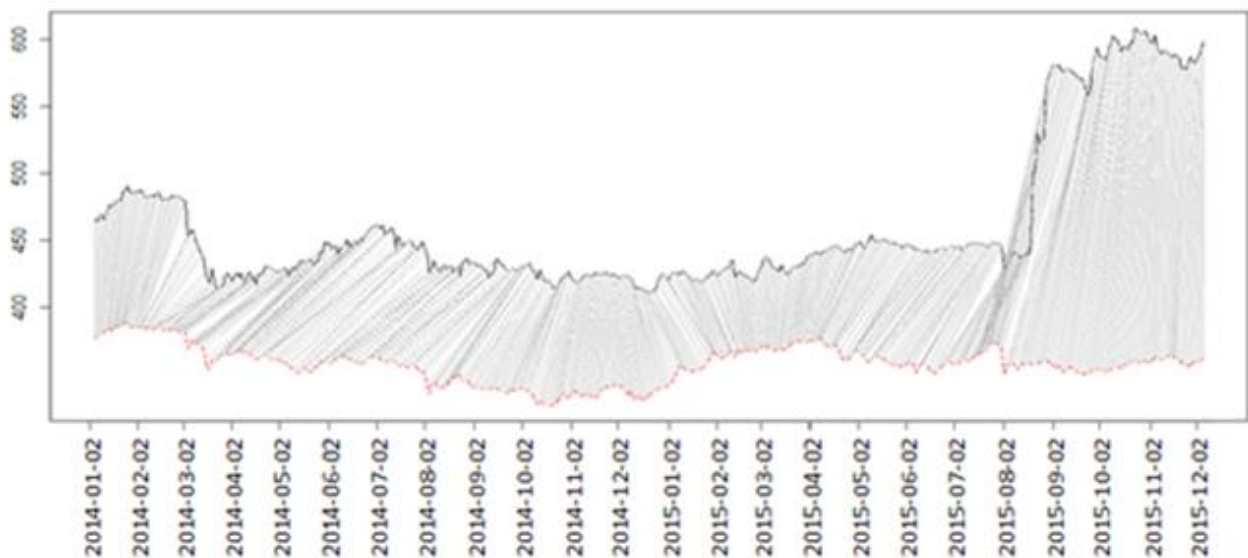
Visų panaudotų atstumo metrikų atveju buvo gauta, kad trumpiausias atstumas yra tarp OMX Baltic ir OMX Ryga kainų indeksų. Taigi DLSI algoritmas tikrai gavo korektišką rezultatą.

Toliau pereisime prie gautos trajektorijos tarp OMX Ryga ir OMX Baltic, kuriai turi galioti trys pagrindinės savybės, t.y. monotoniškumas, tęstinumas ir globalus išlyginimas. Iš žemiau esančio grafiko matyti, kad nebuvo pažeistas globalaus išlyginimo, monotoniškumo ir tęstinumo apribojimai. Todėl abiejų sekų pradžios ir pabaigos taškai sutampa, nebuvo atlikti žingsniai atgal. Be to nematyti ir žingsnių peršokusių laiko skalę daugiau nei per 1 laiko vieneta.



3.22.pav. OMX Ryga ir OMX Baltic trajektorijos grafikas

Apačioje vaizduojamas dinaminio laiko skalės iškraipymo grafikas tarp tiriamų OMX Baltic ir OMX Ryga sekų, čia horizontali ašis parodo įvesties (OMX Baltic) ir tikslinės (OMX RYGA) sekų indeksą  $k = \overline{1,502}$ . Raudona trūki kreivė parodo įvesties seką  $y_j$ , o juoda išsisa kreivė parodo tikslinę seką  $x_i$ . Linijos, kurios sujungia sekų reikšmes, rodo sutapatinimus tarp nagrinėjamų sekų per visą trajektoriją:  $(i_p, j_p) = path(p)$ , kai  $p = 1,502$ .



3.23.pav. OMX Ryga ir OMX Baltic dinaminio laiko skalės iškraipymo grafikas

Iš grafinio finansinių laiko eilučių sutapatinimo galima pamatyti, kad buvo atliktas įvesties sekos vieno taško sujungimas su keliais tikslo sekos taškais. Atidžiau išnagrinėjus viršuje pavaizduotą dinaminio laiko skalės iškraipymo grafiką galima pastebėti, kad iki 2014 metų

rugsėjo mėnesio OMX Baltic (raudona linija) indekso kainos istorija sutapatinama su apie mėnesį vėluojančia OMX Ryga (juoda linija) indekso kainos istorija. Vėlesniame laikotarpyje matoma gana panaši kainų elgsena, išskyrus du 2015 metų laikotarpius, t.y. 1 – 3 mėnesiai ir 5 – 8 mėnesiai.

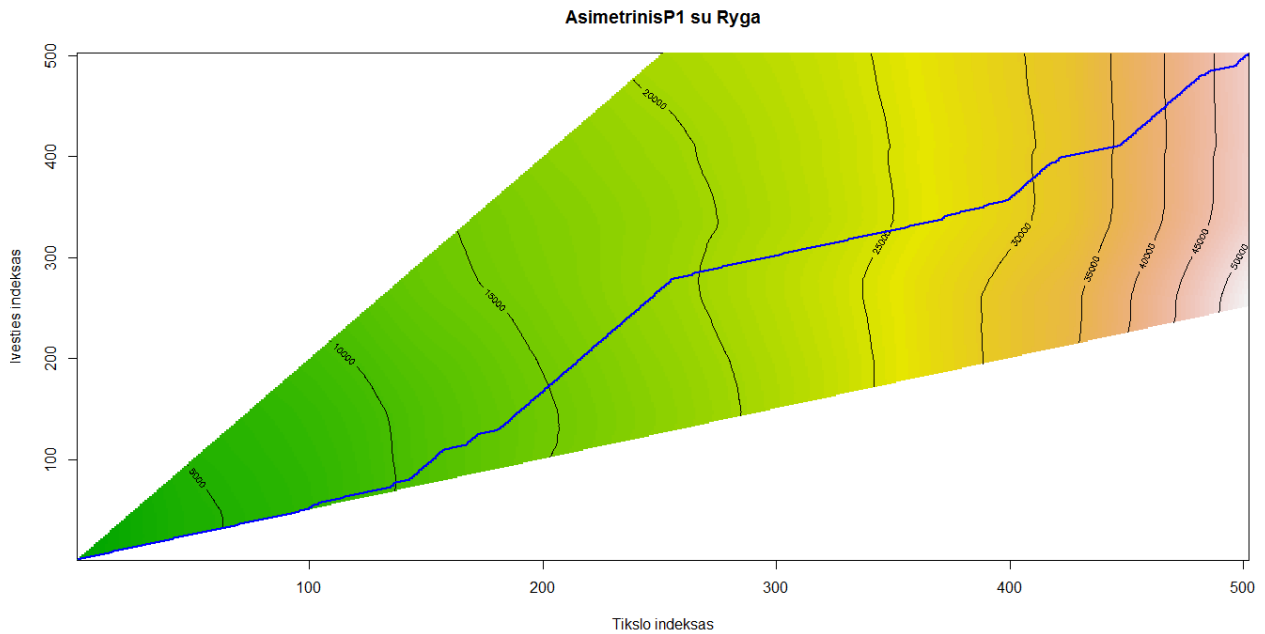
Siekiant sudaryti šią trajektoriją ir atvaizduoti abiejų sekų laiko skalės iškraipymo grafikus buvo atlikti tokie jungimai:

**10 lentelė. OMX Baltic ir OMX Ryga trajektorijos statistikos**

Trajektorija	Skaičius	Trajektorijos %	Įvesties %	Tikslo %
Tiesioginiai sujungimai	445	79.61	88.65	88.65
Suspaudimai	57	10.20	11.35	11.35
Išplėtimai	57	10.20	11.35	11.35
Iškraipymai	114	20.39	22.71	22.71

Buvo atlikti 445 tiesioginiai braižymai, kurie sudaryti iš 87% OMX Baltic sekos duomenų ir 87% OMX Ryga sekos duomenų. Taip pat atlikti 114 dinaminiai laiko skalės iškraipymai, t.y. 57 suspaudimai ir 57 išsiplėtimai. Vadinasi tiesioginių sujungimų buvo atlikta daugiau negu išsiplėtimų ir suspaudimų. Tiesioginiai sujungimai sudarytoje trajektorijoje sudaro apie 80% , o laiko skalės iškraipymai tik 20%. Dėl to, kad abiejų sekų ilgiai yra vienodi procentinės dalys sutampa, t.y. sujungiami visi nagrinėjamų sekų taškai.

Trajektorijos tarp OMX Ryga ir OMX Baltic tankio grafikas parodo atstumų matricos kaštus su kuriais buvo sudarytas sutapatinimo kelias. Horizontali ašis žymi įvesties sekos indeksą, o vertikali ašis – tikslo sekos indeksą. Mėlyna linija parodo sudarytą trajektoriją per kaštų matricą, čia žali plotai atspindi mažus suspaudimus ar išsiplėtimus, geltoni – vidutinius, o raudoni – didelius. Kaip ir tikėtasi dėl OMX Rygos kainų šuolio aukštyn nuo 2015 metų rugsėjo mėnesio, suspaudimo kaštai šiai sekai yra didžiausi pavaizduotos trajektorijos pabaigoje.



**3.24.pav.** OMX Ryga ir OMX Baltic trajektorijos tankio grafikas

Pagrindinis tikslas buvo surasti panašiausias savo forma OMX kainų indeksų eilutes su bendru OMX Baltic kainų indeksu, t.y. rasti trumpiausią, mažiausiai kaštų reikalaujantį sutapatinimo kelią. Nagrinėjamu atveju buvo atsižvelgta ne tik į mažiausią gautą atstumą, bet ir siekiama kuo mažiau tikslo sekos taškų atvaizduoti į vieną ar kelis įvesties sekos taškus (angl. *one to many mapping*). Todėl buvo pasirinktas tinkamiausias *AsimetrisP1* apribojimas. Palyginus gautus rezultatus paaikškėjo, kad labiausiai panašus su OMX Baltic kainų indeksu yra OMX Ryga kainų indeksas, didžiausias skirtumas buvo rastas su OMX Talinas kainų indeksu. Panaudotos Euklido, Manhattan ir Minkovskio atstumo metrikos taip pat parodė, kad OMX Baltic kainų indeksas yra artimiausias atstumo (neatsižvelgiant į skirtingą eilučių greitį) prasme su OMX Ryga kainų indeksu. Nors stipriausias koreliacinis ryšys rastas tarp OMX Baltic ir OMX Talinas indeksų kainų, o OMX Ryga bendrovės sudaro mažiausią dalį bendrajame indekse, tačiau laiko eilučių panašumo formos prasme rezultatai yra priešingi. Dėl nevienodo sekų greičio dinaminio laiko skalės iškraipymo algoritmas surado, kad OMX Baltic ir OMX Ryga kainų indeksų elgsena savo forma nagrinėjamu laikotarpiu yra labiausiai panašios, t.y. tikslo seka sudaro apie 86% formos panašumų su įvesties seka. Nagrinėjant tik šias dvi eilutes pastebimas OMX Ryga indekso kainos vėlavimas pagal OMX Baltic indekso kainą.

## DISKUSIJOS

Darbe buvo nagrinėjamas dinaminio laiko skalės iškraipymo algoritmas: teorinis tyrimas ir praktinis pritaikymas finansinėms laiko eilutėms. Pradiniai tyrimai buvo atliekami sintetiniams duomenims norint patikrinti naudojamo algoritmo korektiškumą, atitinkamų taškų teisingą sutapatinimą, kadangi buvo žinomas greičio skirtumas tarp eilučių. Nors algoritmo veikimas buvo patikrintas su sintetine eilute, tačiau kiekvieno tyrimo atveju gauti rezultatai yra gana skirtingi.

Naudojant šį algoritmą buvo susiduriama su problema kaip patikrinti gautų rezultatų paklaidas, t.y. trūksta statistinių rodiklių, kurie galėtų patvirtinti ar tikrai buvo gautas geriausias rezultatas ne tik atstumo, formos prasme, bet ir teisingu taškų sutapatinimu. Gana didelė gautų rezultatų dalis paliekama nuspęsti ne rodikliniais, bet grafiniais rezultatais.

Pritaikius dinaminio laiko skalės iškraipymo metodą pasirinktoms finansinėms laiko eilutėms buvo gauti jų formos panašumai, pateikti pastebėjimai apie skirtingus eilučių greičius, t.y. matomus vėlinimus. Gauti rezultatai gali būti panaudojami tolimesniems tyrimams, tokiems kaip viso vienos eilutės laikotarpio suskaidymas formos prasme ir tam tikrų dėsningumų ieškojimas jos istorinėje elgsenoje.

## IŠVADOS

Šiame darbe buvo pristatytas dinaminio laiko skalės iškraipymo metodas, išanalizuotas metodo veikimas su sintetinėmis laiko eilutėmis bei atliktas tyrimas su realiomis finansų laiko eilutėmis. Tyrimo metu gautos tokios išvados:

- Pristatyti įvairių sričių dinaminio laiko skalės iškraipymo algoritmo tyrimai visame pasaulyje, pateikti jų rezultatai. Pateikus ribotą finansinį metodo pritaikymą parodyta, kad ši sritis yra mažai tyrinėta.
- Įvaldytas tinkamų apribojimų parinkimas korektiškam dinaminio laiko skalės iškraipymo algoritmo panaudojimui. Ištirtos savybės leidžiančios finansinėse laiko eilutėse rasti minimalius atstumus formos prasme, t.y. panašiausią elgseną vienoje laiko eilutėje arba tarp kelių.
- Vienmačiu realių laiko eilučių tyrimo atveju, t.y. OMX gražos indekso, rasti panašūs gražos elgsenos pasikartojimai istoriniuose duomenyse. Sudarytų panašios formos sutapatinimų laikotarpių indeksui būdingas padidėjimas per 2 vienetus.
- Pirmojo tyrimo atveju pateiktas ir vienas iš metodo privalumų, panaudojant OMX gražos indeksą sutapatintos skirtingos trukmės laiko eilutės. Ištyrus 2012.11.07 – 2013.01.01 laikotarpį su 2012.01.05 – 2012.02.29 laikotarpiu gauta trumpiausia trajektorija, kuriai sudaryti buvo panaudotas vieno taško į kelis taškus atvaizdavimas.
- Taip pat pasinaudojant hierarchinio klasterizavimo metodu ir dinaminio laiko skalės iškraipymo algoritmo kaštų matrica suklastertizuoti 5 panašūs fragmentai iš OMX gražos indekso laiko eilutės. Pagal šių eilučių fragmentus vėliau sudaryti 3 pagrindiniai klasteriai, kuriems būdinga panaši gražos indekso elgsena arba laiko eilutės forma.
- Tyrinėjant dvimatį atvejį su OMX Vilnius ir OMX Ryga laiko eilutėmis rastas mažiausias sutapatinimo kelias su pasirinktu *Simetrinis1* apribojimu, kuris atliko didesnę tiesioginių sujungimų skaičių tarp sekų. Pastebėta, kad indekso OMX Ryga kainos elgsena beveik visame laikotarpyje vėlavo pagal OMX Vilnius indekso kainą.
- Daugiamačio laiko eilučių tyrimo tikslas buvo surasti panašiausias OMX kainų indeksų eilutes su bendru OMX Baltic kainų indeksu. Atsižvelgus į gautą trumpiausią trajektoriją dinaminio laiko skalės iškraipymo metodu nustatyta, kad labiausiai panašus su OMX Baltic kainų indeksu yra OMX Ryga kainų indeksas.
- Dinaminio laiko skalės algoritmo rezultatai daugiamačiu tyrimo atveju parodė, kad laiko eilučių panašumas formos prasme yra priešingas koreliaciniams ryšiams.

## LITERATŪROS SĄRAŠAS

1. NOMIKOS, P. and MACGREGOR, J.F. Monitoring Batch Processes using Multiway Principal Component Analysis. *AIChE Journal*, 1994, vol. 40, no. 8. pp. 1361-1375 ISSN 1547-5905.
2. LI, Y., WEN, C., XIE, Z. and XU, X. *Synchronization of Batch Trajectory Based on Multi-Scale Dynamic Time Warping*. IEEE, 2003.
3. RODRÍGUEZ, J.J. and ALONSO, C.J. *Interval and Dynamic Time Warping-Based Decision Trees*. ACM, 2004.
4. LI, G., YUAN, T., QIN, S.J. and CHAI, T. Dynamic Time Warping Based Causality Analysis for Root-Cause Diagnosis of Nonstationary Fault Processes. *IFAC-PapersOnLine*, 2015, vol. 48, no. 8. pp. 1288-1293.
5. Jorge, Pedro Lopes Joaquim A. Dynamic Time Warping for Pure Data.
6. TAYLOR, J., ZHOU, X., ROUPHAIL, N.M. and PORTER, R.J. Method for Investigating Intradriver Heterogeneity using Vehicle Trajectory Data: A Dynamic Time Warping Approach. *Transportation Research Part B: Methodological*, 2015, vol. 73. pp. 59-80.
7. BHANU, B. and ZHOU, X. *Face Recognition from Face Profile using Dynamic Time Warping*. IEEE, 2004.
8. CELEBI, S., AYDIN, A.S., TEMIZ, T.T. and ARICI, T. *Gesture Recognition using Skeleton Data with Weighted Dynamic Time Warping*. , 2013.
9. LEI, H. and SUN, B. *A Study on the Dynamic Time Warping in Kernel Machines*. IEEE, 2007.
10. FORESTIER, G., et al. Classification of Surgical Processes using Dynamic Time Warping. *Journal of Biomedical Informatics*, 2012, vol. 45, no. 2. pp. 255-264.
11. PIEROBON, M., MARCON, M., SARTI, A. and TUBARO, S. *Clustering of Human Actions using Invariant Body Shape Descriptor and Dynamic Time Warping*. IEEE, 2005.
12. DAI, C., WANG, K. and JIN, R. Monitoring Profile Trajectories with Dynamic Time Warping Alignment. *Quality and Reliability Engineering International*, 2014, vol. 30, no. 6. pp. 815-827.
13. LLANOS, D., et al. *Abstraction of Significant Temporal Features of Voltage Sags Recorded in a 25kV Substation*. , 2003.
14. BREGÓN, A., et al. Current topics in artificial intelligenceSpringer, 2005 *Early Fault Classification in Dynamic Systems using Case-Based Reasoning*, pp. 211-220.
15. KASSIDAS, A., MACGREGOR, J.F. and TAYLOR, P.A. Synchronization of Batch Trajectories using Dynamic Time Warping. *AIChE Journal*, 1998, vol. 44, no. 4. pp. 864-875.

16. KASSIDAS, A., TAYLOR, P.A. and MACGREGOR, J.F. Off-Line Diagnosis of Deterministic Faults in Continuous Dynamic Multivariable Processes using Speech Recognition Methods. *Journal of Process Control*, 1998, vol. 8, no. 5. pp. 381-393.
17. CONT, R. Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues, 2001.
18. HALL, P. and YAO, Q. Inference in ARCH and GARCH Models with Heavy-tailed Errors. *Econometrica*, 2003, vol. 71, no. 1. pp. 285-317.
19. TSINASLANIDIS, P., ALEXANDRIDIS, A., ZAPRANIS, A. and LIVANIS, E. Dynamic Time Warping as a Similarity Measure: Applications in Finance, 2014.
20. KIA, A.N., HARATIZADEH, S. and ZARE, H. Prediction of USD/JPY Exchange Rate Time Series Directional Status by KNN with Dynamic Time Warping AS Distance Function. *Bonfring International Journal of Data Mining*, 2013, vol. 3, no. 2. pp. 12.
21. BAGHERI, A., PEYHANI, H.M. and AKBARI, M. Financial Forecasting using ANFIS Networks with Quantum-Behaved Particle Swarm Optimization. *Expert Systems with Applications*, 2014, vol. 41, no. 14. pp. 6235-6250.
22. LEE, S. and OH, K. Finding the Optimal Frequency for Trade and Development of System Trading Strategies in Futures Market using Dynamic Time Warping. *Journal of the Korean Data and Information Science Society*, 2011, vol. 22, no. 2. pp. 255-267.
23. SINGH, A., YADAV, A. and RANA, A. K-Means with Three Different Distance Metrics. *International Journal of Computer Applications*, 2013, vol. 67, no. 10.
24. LEONARD, M., SLOAN, J., LEE, T. and ELSHEIMER, B. An Introduction to Similarity Analysis using SAS®. *SAS Institute Inc., Cary, NC*, 2010.
25. GIORGINO, T. Computing and Visualizing Dynamic Time Warping Alignments in R: The Dtw Package. *Journal of Statistical Software*, 2009, vol. 31, no. 7. pp. 1-24.
26. SENIN, P. Dynamic Time Warping Algorithm Review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 2008. pp. 1-23.
27. MÜLLER, M. *Information Retrieval for Music and Motion*. Springer, 2007.
28. SAKOE, H. and CHIBA, S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions On*, 1978, vol. 26, no. 1. pp. 43-49.
29. ITAKURA, F. Minimum Prediction Residual Principle Applied to Speech Recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions On*, 1975, vol. 23, no. 1. pp. 67-72.
30. KEOGH, E. and RATANAMAHATANA, C.A. Exact Indexing of Dynamic Time Warping. *Knowledge and Information Systems*, 2005, vol. 7, no. 3. pp. 358-386.
31. PUKĖNAS, K. Kokybinių Duomenų Analizė SPSS Programa. *Kaunas: Lietuvos Kūno Kultūros Akademija*, 2009.



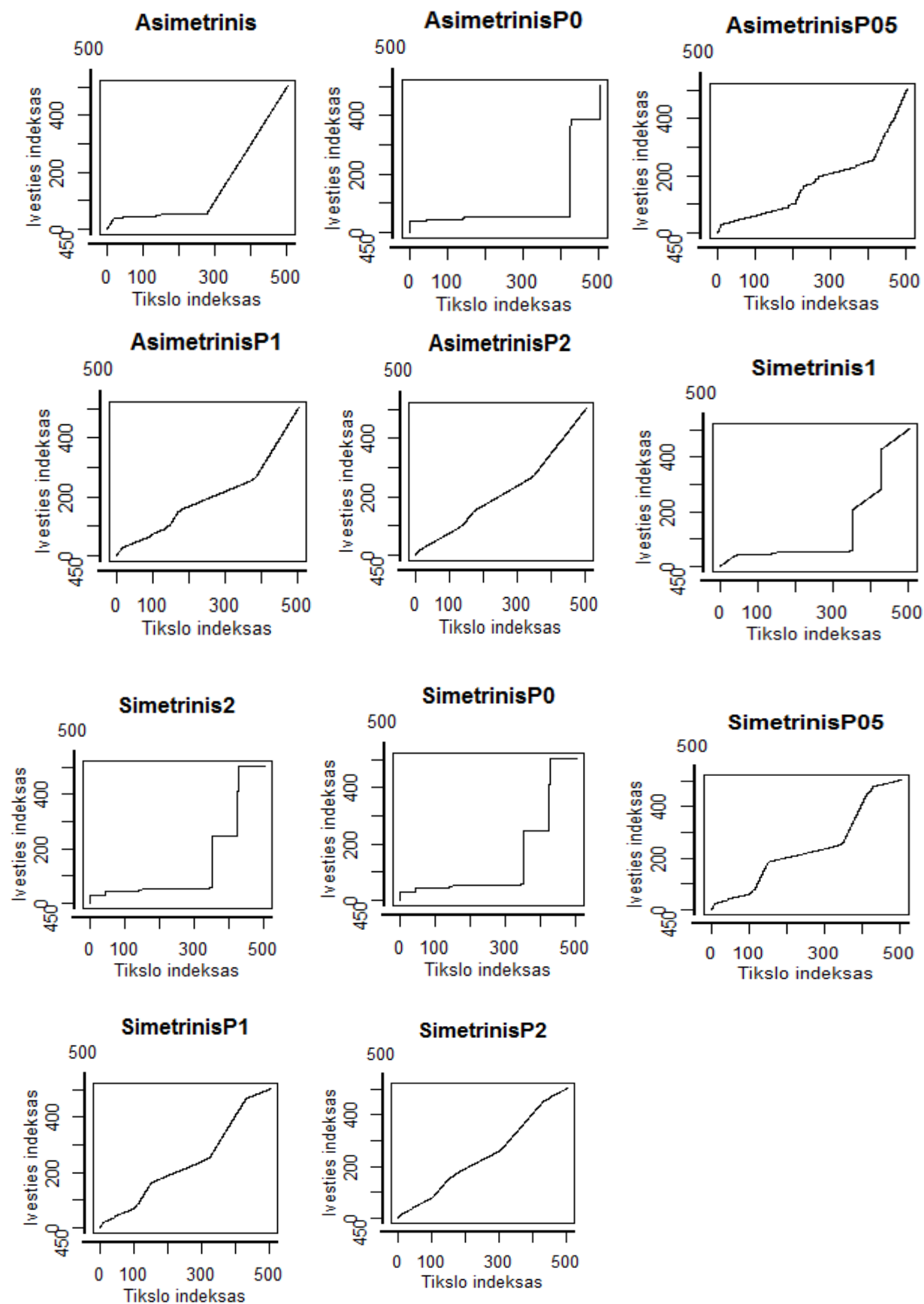
# 1 PRIEDAS. OMX GRAŽOS INDEKSO ISTORINIAI KAINŲ ELGSENOS PASIKARTOJIMAI

Ieškomas fragmentas		Pirmas pasikartojimas		Antras pasikartojimas	
DATA	OMX	DATA	OMX	DATA	OMX
2012-11-07	8.17	2010-02-11	14.52	2012-01-05	4.79
2012-11-08	8	2010-02-12	14.76	2012-01-06	4.48
2012-11-09	7.97	2010-02-16	14.73	2012-01-09	4.74
2012-11-12	8.05	2010-02-17	15.08	2012-01-10	4.81
2012-11-13	8.41	2010-02-18	14.99	2012-01-11	4.91
2012-11-14	8.23	2010-02-19	14.91	2012-01-12	4.89
2012-11-15	8.23	2010-02-22	15.68	2012-01-13	4.73
2012-11-16	9.22	2010-02-23	15.78	2012-01-16	4.73
2012-11-19	9.39	2010-02-24	15.86	2012-01-17	4.65
2012-11-20	9.41	2010-02-25	15.98	2012-01-18	5.37
2012-11-21	9.49	2010-02-26	15.97	2012-01-19	5.8
2012-11-22	9.49	2010-03-01	16.63	2012-01-20	5.81
2012-11-23	9.55	2010-03-02	16.31	2012-01-23	5.95
2012-11-26	9.61	2010-03-03	16	2012-01-24	6.03
2012-11-27	10.03	2010-03-04	16.22	2012-01-25	6.06
2012-11-28	10.16	2010-03-05	16.44	2012-01-26	5.99
2012-11-29	10.07	2010-03-08	16.37	2012-01-27	6.04
2012-11-30	10	2010-03-09	16.41	2012-01-30	5.99
2012-12-03	9.94	2010-03-10	16.55	2012-01-31	5.53
2012-12-04	9.92	2010-03-11	16.61	2012-02-01	5.62
2012-12-05	9.69	2010-03-12	16.87	2012-02-02	5.65
2012-12-06	9.75	2010-03-15	16.73	2012-02-03	5.67
2012-12-07	9.7	2010-03-16	16.89	2012-02-06	5.58
2012-12-10	9.53	2010-03-17	16.96	2012-02-07	5.46
2012-12-11	9.7	2010-03-18	17.17	2012-02-08	5.6
2012-12-12	9.57	2010-03-19	16.73	2012-02-09	5.83
2012-12-13	9.65	2010-03-22	16.75	2012-02-10	5.62
2012-12-14	9.65	2010-03-23	17.25	2012-02-13	5.69
2012-12-17	9.83	2010-03-24	16.61	2012-02-14	5.65
2012-12-18	10.03	2010-03-25	16.5	2012-02-15	5.53
2012-12-19	9.92	2010-03-26	16.8	2012-02-16	5.57
2012-12-20	9.92	2010-03-29	17.03	2012-02-17	5.75
2012-12-21	9.73	2010-03-30	16.91	2012-02-20	5.75
2012-12-24	9.68	2010-03-31	16.42	2012-02-21	5.61
2012-12-25	9.68	2010-04-01	16.24	2012-02-22	5.52
2012-12-26	9.41	2010-04-05	16.46	2012-02-23	6.04
2012-12-27	9.5	2010-04-06	16.72	2012-02-24	5.68
2012-12-28	9.29	2010-04-07	16.08	2012-02-27	5.61
2012-12-31	9.76	2010-04-08	16.36	2012-02-28	5.83
2013-01-01	9.76	2010-04-09	16.69	2012-02-29	5.6

## 2 PRIEDAS. OMX GRAŽOS INDEKSO PRADINIAI DUOMENYS KLASTERIZAVIMUI

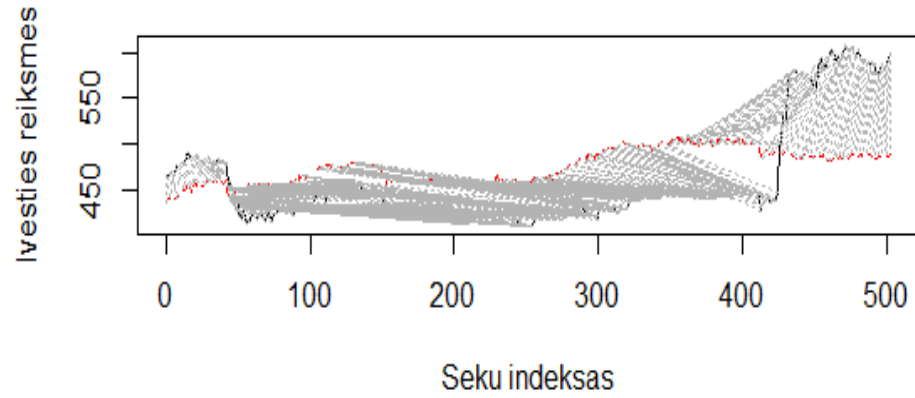
DATA	F1	Data	F2	DATA	F3	Data	F4	DATA	F5
2010-02-11	14.52	2011-06-03	7.5	2012-01-05	4.79	2012-08-24	5.36	2012-11-07	8.17
2010-02-12	14.76	2011-06-06	6.98	2012-01-06	4.48	2012-08-27	5.49	2012-11-08	8
2010-02-16	14.73	2011-06-07	6.61	2012-01-09	4.74	2012-08-28	5.64	2012-11-09	7.97
2010-02-17	15.08	2011-06-08	6.29	2012-01-10	4.81	2012-08-29	5.67	2012-11-12	8.05
2010-02-18	14.99	2011-06-09	6.4	2012-01-11	4.91	2012-08-30	5.78	2012-11-13	8.41
2010-02-19	14.91	2011-06-10	6.27	2012-01-12	4.89	2012-08-31	5.81	2012-11-14	8.23
2010-02-22	15.68	2011-06-13	6.3	2012-01-13	4.73	2012-09-03	5.81	2012-11-15	8.23
2010-02-23	15.78	2011-06-14	6.67	2012-01-16	4.73	2012-09-04	5.93	2012-11-16	9.22
2010-02-24	15.86	2011-06-15	6.75	2012-01-17	4.65	2012-09-05	6.04	2012-11-19	9.39
2010-02-25	15.98	2011-06-16	6.99	2012-01-18	5.37	2012-09-06	6.25	2012-11-20	9.41
2010-02-26	15.97	2011-06-17	7.11	2012-01-19	5.8	2012-09-07	6.57	2012-11-21	9.49
2010-03-01	16.63	2011-06-20	7.64	2012-01-20	5.81	2012-09-10	6.49	2012-11-22	9.49
2010-03-02	16.31	2011-06-21	7.72	2012-01-23	5.95	2012-09-11	6.5	2012-11-23	9.55
2010-03-03	16	2011-06-22	7.69	2012-01-24	6.03	2012-09-12	6.72	2012-11-26	9.61
2010-03-04	16.22	2011-06-23	7.83	2012-01-25	6.06	2012-09-13	7.11	2012-11-27	10.03
2010-03-05	16.44	2011-06-24	7.69	2012-01-26	5.99	2012-09-14	8.15	2012-11-28	10.16
2010-03-08	16.37	2011-06-27	7.63	2012-01-27	6.04	2012-09-17	7.72	2012-11-29	10.07
2010-03-09	16.41	2011-06-28	7.84	2012-01-30	5.99	2012-09-18	7.57	2012-11-30	10
2010-03-10	16.55	2011-06-29	7.71	2012-01-31	5.53	2012-09-19	7.97	2012-12-03	9.94
2010-03-11	16.61	2011-06-30	7.85	2012-02-01	5.62	2012-09-20	8.03	2012-12-04	9.92
2010-03-12	16.87	2011-07-01	8.23	2012-02-02	5.65	2012-09-21	8.16	2012-12-05	9.69
2010-03-15	16.73	2011-07-05	7.73	2012-02-03	5.67	2012-09-24	8.11	2012-12-06	9.75
2010-03-16	16.89	2011-07-06	7.57	2012-02-06	5.58	2012-09-25	7.87	2012-12-07	9.7
2010-03-17	16.96	2011-07-07	7.89	2012-02-07	5.46	2012-09-26	7.86	2012-12-10	9.53
2010-03-18	17.17	2011-07-08	7.55	2012-02-08	5.6	2012-09-27	7.95	2012-12-11	9.7
2010-03-19	16.73	2011-07-11	7.28	2012-02-09	5.83	2012-09-28	7.81	2012-12-12	9.57
2010-03-22	16.75	2011-07-12	7.17	2012-02-10	5.62	2012-10-01	7.85	2012-12-13	9.65
2010-03-23	17.25	2011-07-13	7.26	2012-02-13	5.69	2012-10-02	7.64	2012-12-14	9.65
2010-03-24	16.61	2011-07-14	7.04	2012-02-14	5.65	2012-10-03	7.69	2012-12-17	9.83
2010-03-25	16.5	2011-07-15	6.88	2012-02-15	5.53	2012-10-04	7.83	2012-12-18	10.03
2010-03-26	16.8	2011-07-18	6.58	2012-02-16	5.57	2012-10-05	7.71	2012-12-19	9.92
2010-03-29	17.03	2011-07-19	6.78	2012-02-17	5.75	2012-10-08	7.54	2012-12-20	9.92
2010-03-30	16.91	2011-07-20	6.8	2012-02-20	5.75	2012-10-09	7.45	2012-12-21	9.73
2010-03-31	16.42	2011-07-21	7.02	2012-02-21	5.61	2012-10-10	7.43	2012-12-24	9.68
2010-04-01	16.24	2011-07-22	6.89	2012-02-22	5.52	2012-10-11	7.51	2012-12-25	9.68
2010-04-05	16.46	2011-07-25	6.89	2012-02-23	6.04	2012-10-12	7.41	2012-12-26	9.41
2010-04-06	16.72	2011-07-26	7.64	2012-02-24	5.68	2012-10-15	7.28	2012-12-27	9.5
2010-04-07	16.08	2011-07-27	7.28	2012-02-27	5.61	2012-10-16	7.51	2012-12-28	9.29
2010-04-08	16.36	2011-07-28	7.3	2012-02-28	5.83	2012-10-17	7.52	2012-12-31	9.76
2010-04-09	16.69	2011-07-29	7.08	2012-02-29	5.6	2012-10-18	7.58	2013-01-01	9.76

### 3 PRIEDAS. OMX VILNIUS IR OMX RYGA TRAJEKTORIJŲ GRAFIKAI

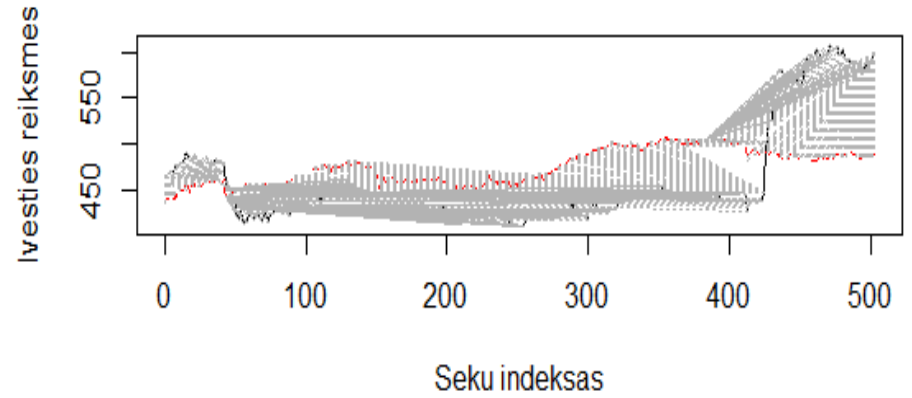


#### 4 PRIEDAS. OMX VILNIUS IR OMX RYGA DINAMINIO LAIKO SKALĖS IŠKRAIPYMO GRAFIKAI

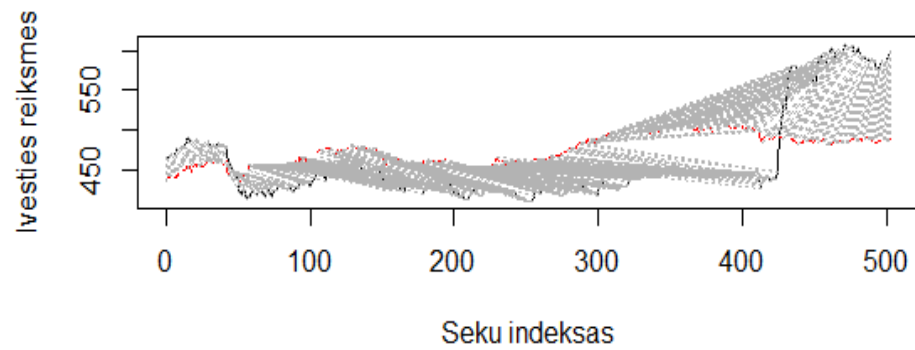
**Asimetrinis**



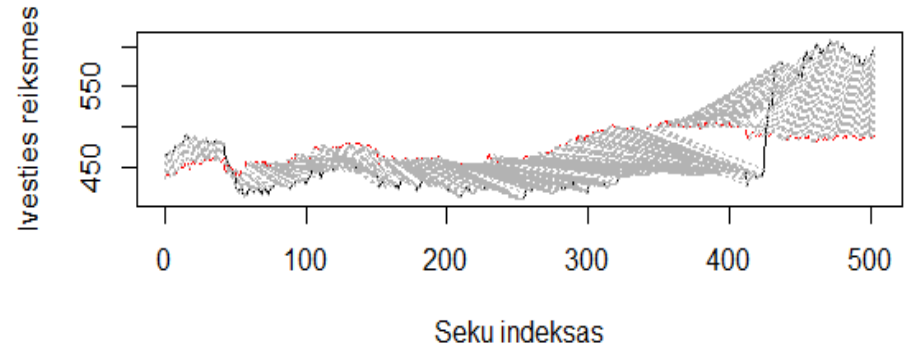
**AsimetrinisP0**



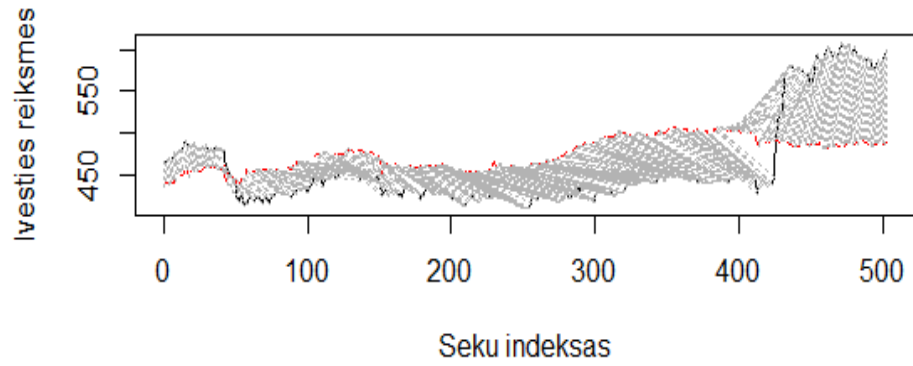
**AsimetrinisP05**



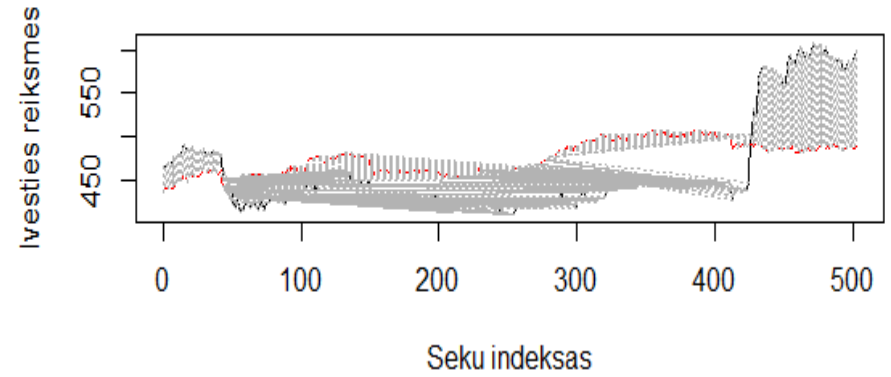
**AsimetrinisP1**



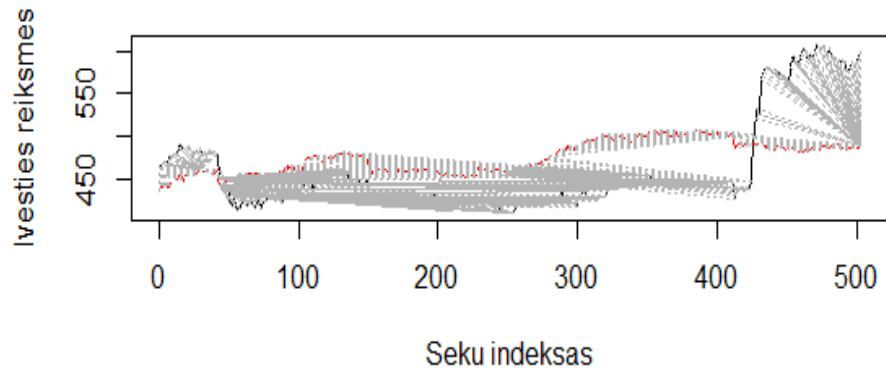
**AsimetrinisP2**



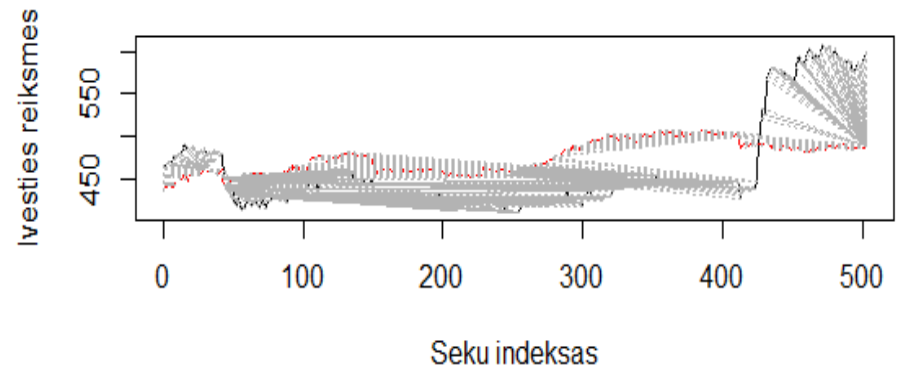
**Simetrinis1**



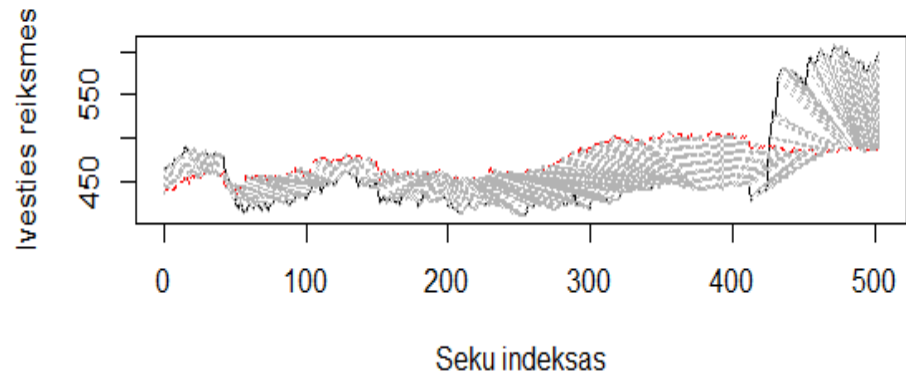
**Simetrinis2**



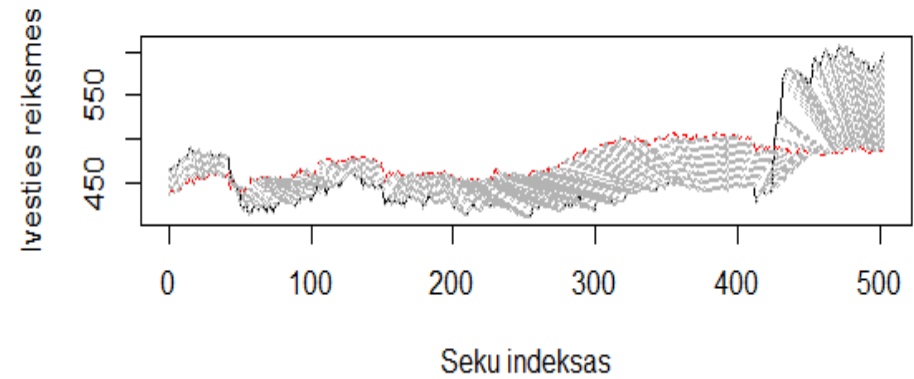
**SimetrinisP0**



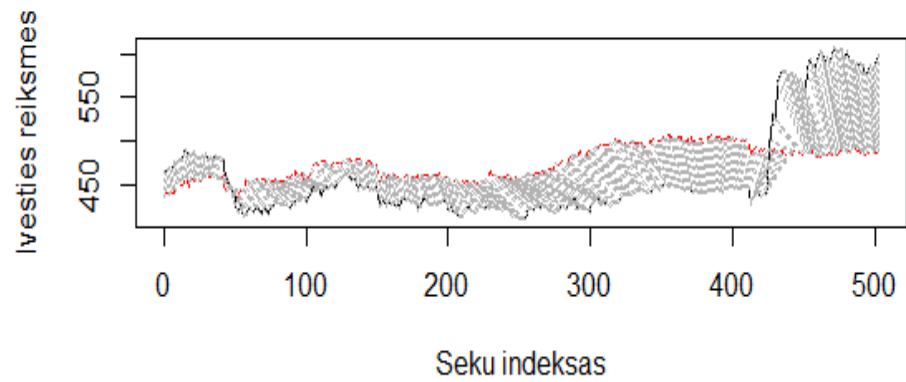
**SimetrinisP05**



**SimetrinisP1**

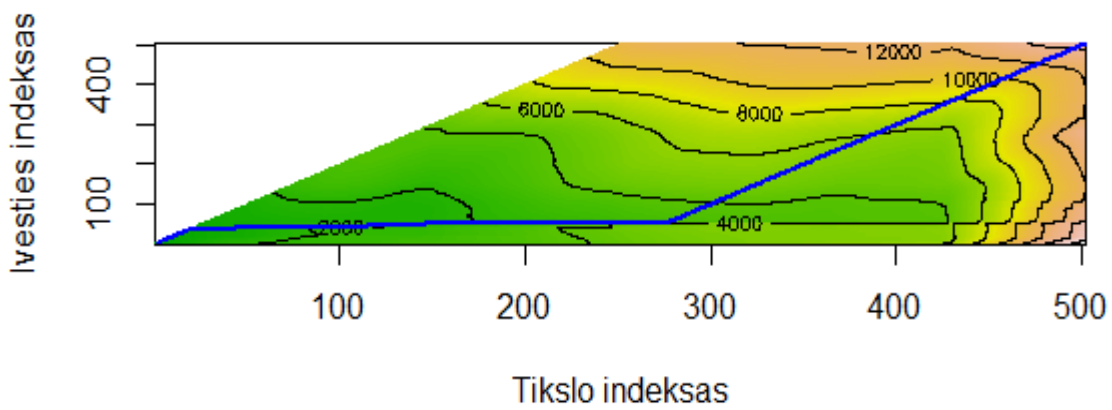


**SimetrinisP2**

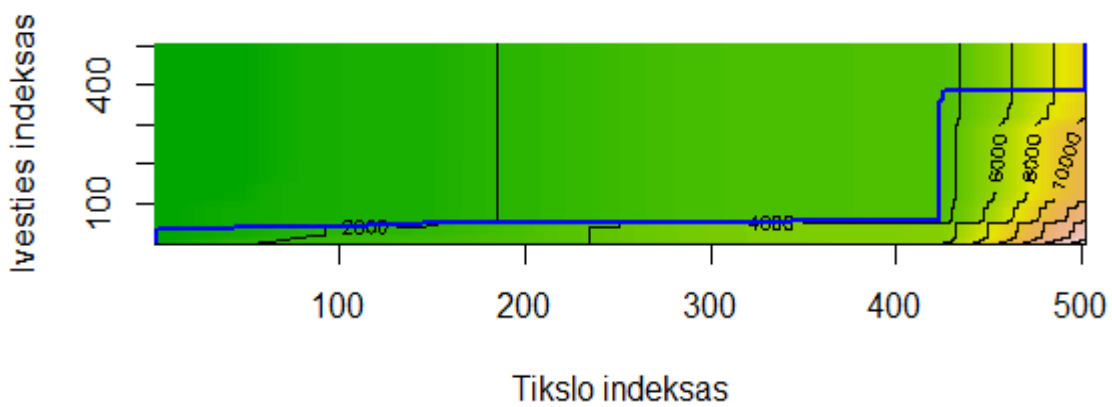


## 5 PRIEDAS. OMX VILNIUS IR OMX RYGA TANKIO GRAFIKAI

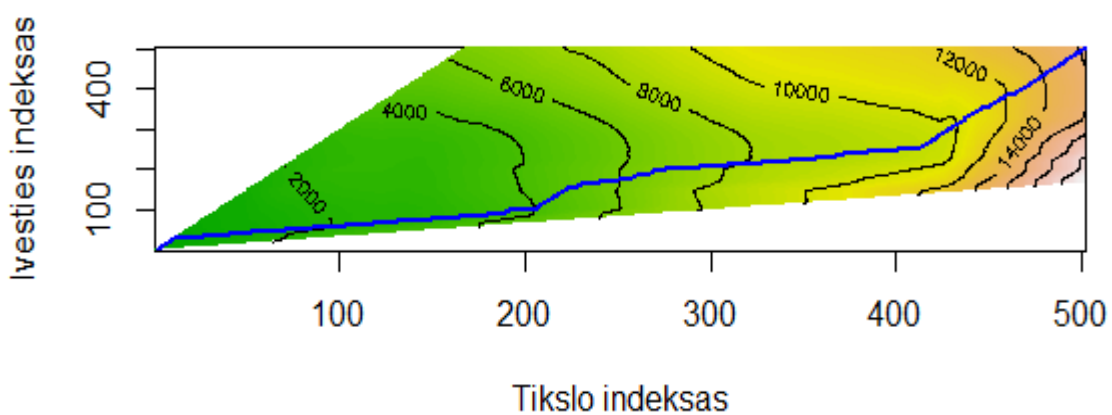
### Asimetrinis1



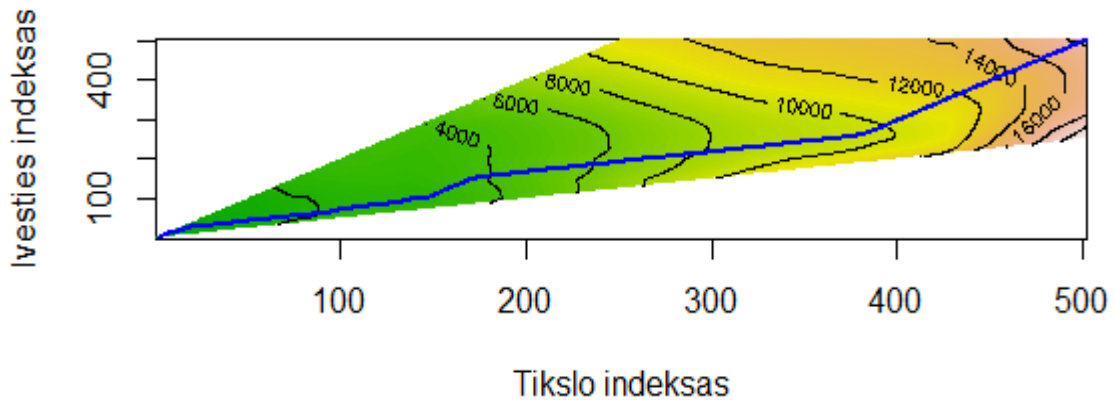
### AsimetrinisP0



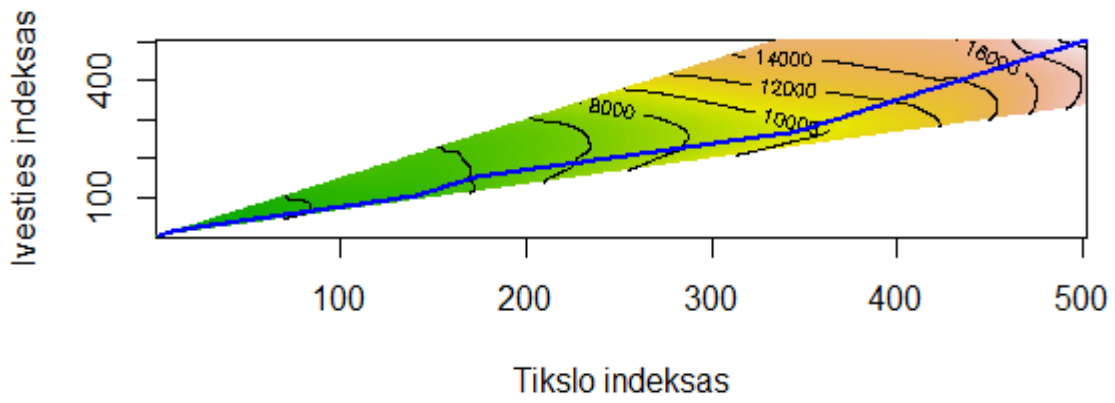
### AsimetrinisP05



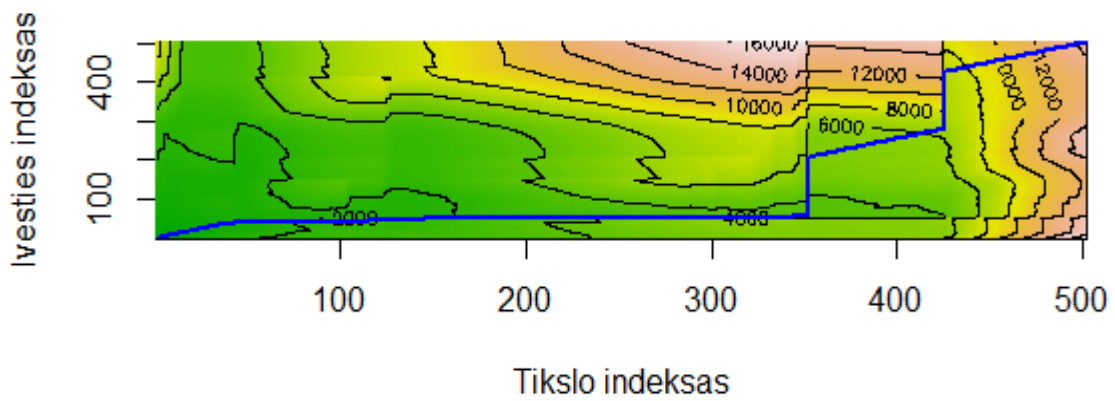
### AsimetrinisP1



### AsimetrinisP2

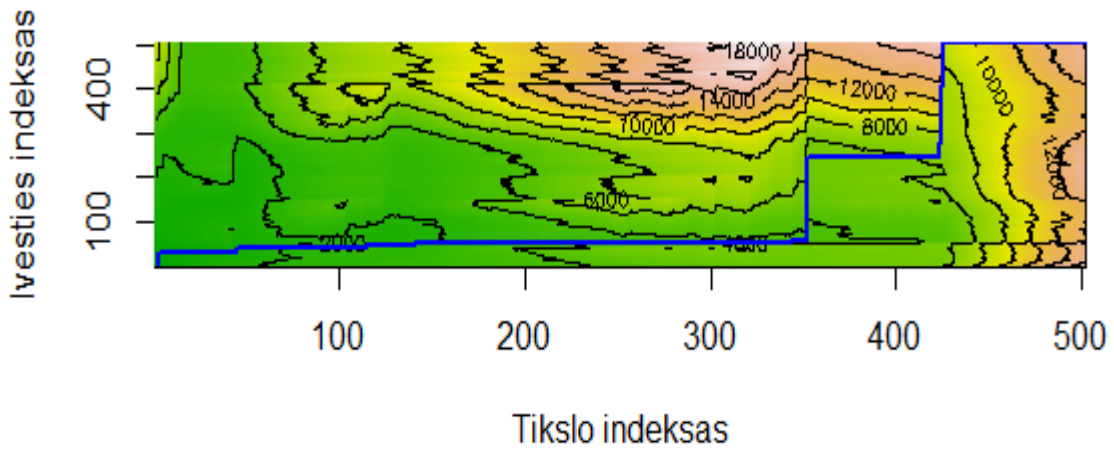


### Simetrinis1

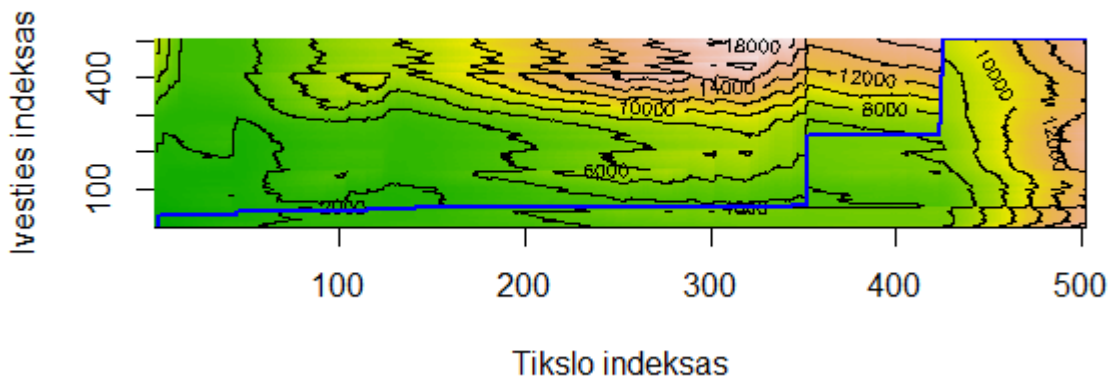




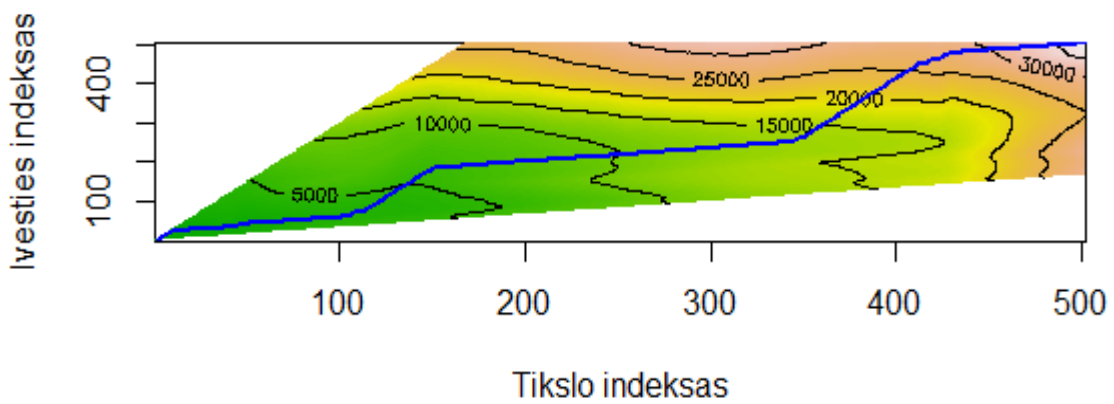
### Simetrinis2



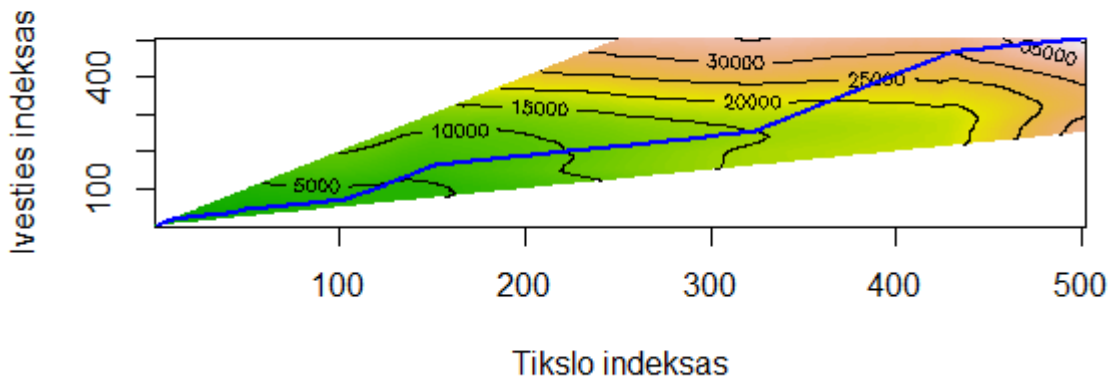
### SimetrinisP0



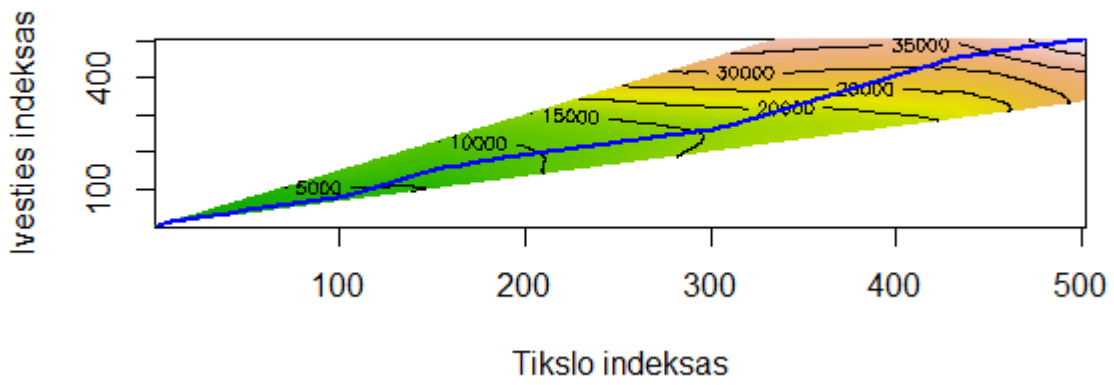
### SimetrinisP05



### SimetrinisP1



### SimetrinisP2



## 6 PRIEDAS. PROGRAMOS TEKSTAS

R studio programos kodas:

```
#vienmatis atvejis
library(dtw)
library(xlsx)
library(quantmod)
library(zoo)
#sin tyrimas: sugeneruojami duomenys
sin_pradiniai=read.xlsx("C:/Users/W7/Desktop/sin.xlsx",1, header=T)
reference_sin<-sin_pradiniai
laikas<-seq(as.Date("2000-01-01"), as.Date("2000-06-06"), by = "1 day")
#sukuriamas duomeniu masyvas
sin_duomenys=data.frame(laikas, reference_sin)
sin_duomenys$laikas<-format(as.POSIXct(sin_duomenys$laikas, format = "%Y-%m-%d"),format = "%Y-%m-%d")
SIN=data.frame(xts(sin_duomenys[,1],order.by=as.POSIXct(sin_duomenys$laikas)))
data.frame<-setNames(data.frame(SIN),c("SIN.Close"))
datas =Cl(data.frame)
#Parenka paskutines nurodytas stebejimo reikšmes kaip fragmenta, nustato jo masteli, paima tik reikšmes
pat <- scale(coredata(last(datas, 20)))
n<-length(pat)
rr<- rollapply(datas, width = n, FUN = function(z) dtw(as.vector(pat),
as.vector(scale(coredata(z))),distance.only=TRUE)$distance,by.column = TRUE, align = "right")
# Ištrina paskutines pasirinktas dienas t.y.20
rr <- rr[1:(length(rr)-length(pat))># rr turi vertimo atstumus
# Ieško pirmuju k lokaciju
ks<-sort(as.vector(rr), index.return=T)
#braizo laiko eilutes grafika
plot(as.zoo(datas), type='l', col='gray', main="SIN(X)")
#pazymi ieskoma fragmenta
lines(as.zoo(last(datas,20)), col='red')
#pazymi panasius fragmentus
for (i in 1:2){
  lines(as.zoo(window(datas, start=time(rr)[ks$ix[i]], end=time(rr)[ks$ix[i]+n-1])), col="black")
}
#OMX indekso nuskaitymas
OMX = getSymbols("OMX", src = 'yahoo', from = '2010-01-01', to= '2013-01-01',auto.assign = F)
data =Cl(OMX)
#irasomi pradiniai duomenys i iseitini faila
write.xlsx(OMX,"C:/Users/W7/Desktop/omx_duomenys1.xlsx")
#Parenka paskutines nurodytas stebejimo reikšmes kaip fragmenta, nustato jo masteli, paima tik reikšmes
pat <- scale(coredata(last(data, 40)))
n<-length(pat)
rr <- rollapply(data, width = n, FUN = function(z) dtw(as.vector(pat),
as.vector(scale(coredata(z))),distance.only=TRUE)$distance,by.column = TRUE, align = "right")
# Ištrina paskutines pasirinktas dienas t.y.40
rr <- rr[1:(length(rr)-length(pat))># rr turi vertimo atstumus
# Ieško pirmuju k lokaciju
ks<-sort(as.vector(rr), index.return=T)
#braizo laiko eilutes grafika
plot(as.zoo(data), type='l', col='gray', main="OMX indeksas", xlab="Laikotarpis", ylab="Indekso kaina")
#pazymi ieskoma fragmenta
lines(as.zoo(last(data,40)), col='red')
```

```

#pazymi panasius fragmentus
for (i in 1:100){
  lines(as.zoo(window(data, start=time(rr)[ks$ix[i]], end=time(rr)[ks$ix[i]+n-1])), col="black")
}
write.xlsx(as.zoo(data),"C:/Users/W7/Desktop/omx_40.xlsx")
write.xlsx((as.zoo(window(data, start=time(rr)[ks$ix[i]], end=time(rr)[ks$ix[i]+n-1]))), "C:/Users/W7/Desktop/omx_1.xlsx")
#OMX lyginimas metams
query_omx = as.vector(coredata(last(CI(data['2012::2012']), 20)))
reference_omx = as.vector(coredata(first(CI(data['2012:2012']), 10)))
#simetriniu apribojimu taikymas
alignment = dtw(query_omx, reference_omx, keep=TRUE)
#trajektorijos grafikai
plot(alignment,main='Trajektorija', type='three', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment1,main='Symetrinis2', type='three', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment2,main='SymetrinisP0', type='three', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment3,main='SymetrinisP05', type='three', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment4,main='SymetrinisP1', type='three', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment5,main='SymetrinisP2', type='three', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
#dinaminio laiko skales iskraipymo grafikai
plot(alignment, type='two', xlab="Sekų indeksas", ylab="Tikslo sekos reikšmės")
plot(alignment1,main='Symetrinis2', type='two', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment2,main='SymetrinisP0', type='two', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment3,main='SymetrinisP05', type='two', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment4,main='SymetrinisP1', type='two', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
plot(alignment5,main='SymetrinisP2', type='two', xlab="Tikslo sekos indeksas", ylab="Įvesties sekos indeksas")
hq=(0:40)/40
hq=round(hq*40)
hw=(alignment$index1 %in% hq)
hi=(1:length(alignment$index1))[hw]
dtwPlotThreeWay(alignment,match.indices=hi, main='Trajektorija', xlab="Tikslo indeksas", ylab="Įvesties indeksas")

#dvimatis atvejis
library(dtw)
library(plotly)
library(SimilarityMeasures)
library(xlsx)
library(quantmod)
library(zoo)
#nuskaitomi pradiniai duomenys
sin_du=read.xlsx("C:/Users/W7/Desktop/sin_du.xlsx",1, header=T)
sin_du<-setNames(data.frame(sin_du),c("SIN(x/8pi)", "SIN((x/8pi)+4)"))
#priskiriamos tikslo ir ivesties eilutes
query = sin_du$`SIN(x/8pi)`
reference=sin_du$`SIN((x/8pi)+4)`
#asimetriniu apribojimu trajektorijos
align1=dtw(query,reference,keep=TRUE, step.pattern = asymmetric)
align11=dtw(query,reference,keep=TRUE, step.pattern = asymmetricP0)
align12=dtw(query,reference,keep=TRUE, step.pattern = asymmetricP05)
align13=dtw(query,reference,keep=TRUE, step.pattern = asymmetricP1)
align14=dtw(query,reference,keep=TRUE, step.pattern = asymmetricP2)
#simetriniu apribojimu trajektorijos
align2=dtw(query,reference,keep=TRUE, step.pattern = symmetric1)
align21=dtw(query,reference,keep=TRUE, step.pattern = symmetric2)

```

```

align22=dtw(query,reference,keep=TRUE, step.pattern = symmetricP0)
align23=dtw(query,reference,keep=TRUE, step.pattern = symmetricP05)
align24=dtw(query,reference,keep=TRUE, step.pattern = symmetricP1)
align25=dtw(query,reference,keep=TRUE, step.pattern = symmetricP2)
#asimetrines trajektorijos kastu matrica
align1$costMatrix
write.xlsx(align1$costMatrix, "C:/Users/W7/Desktop/Rezultatai/sin/asimetriniai/asimetriniai/costMatrixa.xlsx")
#asimetriniu DTW grafikai 2 dimensiju
dtwPlotTwoWay(align1, main = "Asimetrinis", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align11, main = "AsimetrinisP0", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align12, main = "AsimetrinisP05",xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align13, main = "AsimetrinisP1", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align14, main = "AsimetrinisP2", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
#simetriniu DTW grafikai 2 dimensiju
dtwPlotTwoWay(align2, main = "Simetrinis1",xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align21, main = "Simetrinis2", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align22, main = "SimetrinisP0", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align23, main = "SimetrinisP05", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align24, main = "SimetrinisP1", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align25, main = "SimetrinisP2", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
#asimetriniu DTW grafikai 3 dimensiju
dtwPlotThreeWay(align11, main = "Asimetrinis", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align11, main = "AsimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align12, main = "AsimetrinisP05",xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align13, main = "AsimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align14, main = "AsimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
#simetriniu DTW grafikai 3 dimensiju
dtwPlotThreeWay(align2, main = "Simetrinis1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align21, main = "Simetrinis2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align22, main = "SimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align23, main = "SimetrinisP05", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align24, main = "SimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align25, main = "SimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
#asimetriniu density grafikas
dtwPlotDensity(align1, main = "Asimetrinis", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align11, main = "AsimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align12, main = "AsimetrinisP05", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align13, main = "AsimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align14, main = "AsimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
#simetriniu density grafikas
dtwPlotDensity(align2, main = "Simetrinis1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align21, main = "Simetrinis2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align22, main = "SimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align23, main = "SimetrinisP05", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align24, main = "SimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align25, main = "SimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")

wt=warp(align1,index.reference = TRUE)#iskraipo reference
plot(query);
points(reference[wt])
warpArea(align1)
wt1=warp(align1,index.reference = FALSE)#iskraipo query
plot(reference);

```

```

points(query[wt1])
warpArea(algn1)

#realiu eiluciu tyrimas
#nuskaitomi pradiniai duomenys
index=read.xlsx("C:/Users/W7/Desktop/Magistrinis/Programa/index.xlsx",1, header=T)
#isskaidomi indesakai: Riga
indexR=data.frame(date = index$Date[index$INDEX == "Riga"],
  close = index$Close[index$INDEX == "Riga"])
indexR$date<-format(as.POSIXct(indexR$date, format = "%Y-%m-%d"),format = "%Y-%m-%d")
Riga=data.frame(xts(indexR[,-1],order.by=as.POSIXct(indexR$date)))
#isskaidomi indeksai: Vilnius
indexV=data.frame(date = index$Date[index$INDEX == "Vilnius"],
  close = index$Close[index$INDEX == "Vilnius"])
indexV$date<-format(as.POSIXct(indexV$date, format = "%Y-%m-%d"),format = "%Y-%m-%d")
Vilnius=data.frame(xts(indexV[,-1],order.by=as.POSIXct(indexV$date)))
write.xlsx(Vilnius,"C:/Users/W7/Desktop/Vilnius.xlsx")
write.xlsx(Riga,"C:/Users/W7/Desktop/Riga.xlsx")
#asimetriniu apribojimu trajektorijos
align1=dtw(Riga,Vilnius,keep=TRUE, step.pattern = asymmetric)
align11=dtw(Riga,Vilnius,keep=TRUE, step.pattern = asymmetricP0)
align12=dtw(Riga,Vilnius,keep=TRUE, step.pattern = asymmetricP05)
align13=dtw(Riga,Vilnius,keep=TRUE, step.pattern = asymmetricP1)
align14=dtw(Riga,Vilnius,keep=TRUE, step.pattern = asymmetricP2)
#simetriniu apribojimu trajektorijos
align2=dtw(Riga,Vilnius,keep=TRUE, step.pattern = symmetric1)
align21=dtw(Riga,Vilnius,keep=TRUE, step.pattern = symmetric2)
align22=dtw(Riga,Vilnius,keep=TRUE, step.pattern = symmetricP0)
align23=dtw(Riga,Vilnius,keep=TRUE, step.pattern = symmetricP05)
align24=dtw(Riga,Vilnius,keep=TRUE, step.pattern = symmetricP1)
align25=dtw(Riga,Vilnius,keep=TRUE, step.pattern = symmetricP2)
#asimetriniu DTW grafikai 2 dimensiju
dtwPlotTwoWay(align1, main = "Asimetrinis", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align11, main = "AsimetrinisP0", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align12, main = "AsimetrinisP05", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align13, main = "AsimetrinisP1", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align14, main = "AsimetrinisP2", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
#simetriniu DTW grafikai 2 dimensiju
dtwPlotTwoWay(align2, main = "Simetrinis1", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align21, main = "Simetrinis2", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align22, main = "SimetrinisP0", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align23, main = "SimetrinisP05", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align24, main = "SimetrinisP1", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
dtwPlotTwoWay(align25, main = "SimetrinisP2", xlab="Sekų indeksas", ylab="Ivesties reikšmes")
#asimetriniu DTW grafikai 3 dimensiju
dtwPlotThreeWay(align1, main = "Asimetrinis", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align11, main = "AsimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align12, main = "AsimetrinisP05", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align13, main = "AsimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align14, main = "AsimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
#simetriniu DTW grafikai 3 dimensiju
dtwPlotThreeWay(align2, main = "Simetrinis1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align21, main = "Simetrinis2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")

```

```

dtwPlotThreeWay(align22, main = "SimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align23, main = "SimetrinisP05", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align24, main = "SimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotThreeWay(align25, main = "SimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
#asimetriniu density grafikas
dtwPlotDensity(align1, main = "Asimetrinis1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align11, main = "AsimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align12, main = "AsimetrinisP05", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align13, main = "AsimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align14, main = "AsimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
#simetriniu density grafikas
dtwPlotDensity(align2, main = "Simetrinis1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align21, main = "Simetrinis2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align22, main = "SimetrinisP0", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align23, main = "SimetrinisP05", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align24, main = "SimetrinisP1", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
dtwPlotDensity(align25, main = "SimetrinisP2", xlab="Tikslo indeksas", ylab="Ivesties indeksas")
#asimetriniu DTW grafikai 3 dimensiju
hq=(0:158)/158
hq=round(hq*158)
hw=(align12$index1 % in% hq)
hi=(1:length(align12$index1))[hw]
dtwPlotThreeWay(align12,match.indices=hi, main = "Asymmetric1")

#daugiamatis atvejis
library(xlsx)
library(dtw)
library(zoo)
library(xts)
#nuskaitomi pradiniai duomenys
index=read.xlsx("C:/Users/W7/Desktop/Magistrinis/Programa/index.xlsx",1, header=T)
#isskaidomi indeskai: Tallin
indexT=data.frame(date = index$Date[index$INDEX == "Tallin"],close = index$Close[index$INDEX == "Tallin"])
indexT$date<-format(as.POSIXct(indexT$date, format = "%Y-%m-%d"),format = "%Y-%m-%d")
Tallin=data.frame(xts(indexT[,-1],order.by=as.POSIXct(indexT$date)))
Tallin<-setNames(data.frame(Tallin),c("Close"))
#isskaidomi indeskai: Riga
indexR=data.frame(date = index$Date[index$INDEX == "Riga"],close = index$Close[index$INDEX == "Riga"])
indexR$date<-format(as.POSIXct(indexR$date, format = "%Y-%m-%d"),format = "%Y-%m-%d")
Riga=data.frame(xts(indexR[,-1],order.by=as.POSIXct(indexR$date)))
Riga<-setNames(data.frame(Riga),c("Close"))
#isskaidomi indeksai: Vilnius
indexV=data.frame(date = index$Date[index$INDEX == "Vilnius"], close = index$Close[index$INDEX == "Vilnius"])
indexV$date<-format(as.POSIXct(indexV$date, format = "%Y-%m-%d"),format = "%Y-%m-%d")
Vilnius=data.frame(xts(indexV[,-1],order.by=as.POSIXct(indexV$date)))
Vilnius<-setNames(data.frame(Vilnius),c("Close"))
#isskaidomi indeksai: Baltic
indexB=data.frame(date = index$Date[index$INDEX == "Baltic"], close = index$Close[index$INDEX == "Baltic"])
indexB$date<-format(as.POSIXct(indexB$date, format = "%Y-%m-%d"),format = "%Y-%m-%d")
Baltic=data.frame(xts(indexB[,-1],order.by=as.POSIXct(indexB$date)))
Baltic<-setNames(data.frame(Baltic),c("Close"))
#skaiciuojamas atstumas pagal Baltic
#Vilnius, Baltic

```

```

alignB1<-dtw(Vilnius, Baltic, keep=TRUE, step.pattern = asymmetric)
alignB11<-dtw(Vilnius, Baltic, keep=TRUE, step.pattern = asymmetricP0)
alignB12<-dtw(Vilnius, Baltic, keep=TRUE, step.pattern = asymmetricP05)
alignB13<-dtw(Vilnius, Baltic, keep=TRUE, step.pattern = asymmetricP1)
alignB14<-dtw(Vilnius, Baltic, keep=TRUE, step.pattern = asymmetricP2)
dtwPlotThreeWay(alignB3, main = "Asymmetric su Ryga")
dtwPlotThreeWay(alignB31, main = "AsymmetricP0 su Ryga")
dtwPlotThreeWay(alignB32, main = "AsymmetricP05 su Ryga")
dtwPlotThreeWay(alignB33, main = "AsimetrinisP1 su Ryga", xlab="Tikslo indeksas", ylab="Investies indeksas")
dtwPlotThreeWay(alignB34, main = "AsymmetricP2 su Ryga")
dtwPlotTwoWay(alignB3, main = "Asymmetric su Ryga")
dtwPlotTwoWay(alignB31, main = "AsymmetricP0 su Ryga")
dtwPlotTwoWay(alignB32, main = "AsymmetricP05 su Ryga")
dtwPlotTwoWay(alignB33, main = "AsimetrinisP1 su Ryga", xlab="Indeksas", ylab="Tikslo sekos reikšmes")
dtwPlotTwoWay(alignB34, main = "AsymmetricP2 su Ryga")
dtwPlotDensity(alignB33, main = "AsimetrinisP1 su Ryga", xlab="Tikslo indeksas", ylab="Investies indeksas")
library(TSdist)
#euklido atstumai
TSDistances(Riga$Cclose, Baltic$Cclose, distance="euclidean")
TSDistances(Vilnius$Cclose, Baltic$Cclose, distance="euclidean")
TSDistances(Tallin$Cclose, Baltic$Cclose, distance="euclidean")
#manhatan atstumai
TSDistances(Riga$Cclose, Baltic$Cclose, distance="manhattan")
TSDistances(Vilnius$Cclose, Baltic$Cclose, distance="manhattan")
TSDistances(Tallin$Cclose, Baltic$Cclose, distance="manhattan")
#minkovskio atstumai
TSDistances(Riga$Cclose, Baltic$Cclose, distance="minkowski", p=1)
TSDistances(Vilnius$Cclose, Baltic$Cclose, distance="minkowski", p=1)
TSDistances(Tallin$Cclose, Baltic$Cclose, distance="minkowski", p=1)

```

SAS programos kodas:

```

PROC IMPORT OUT= WORK.auto DATAFILE= "C:\Users\W7\Desktop\Riga" DBMS=xlsx REPLACE;
SHEET="Sheet1";
GETNAMES=YES;
format date yymmdd10.;
RUN;
proc print data=WORK.auto;
run;
ods graphics on;
/*symmetric1- 795 žingsniai*/
proc similarity data=WORK.auto out=iseitis
print=all plot=all;
/*id date interval=month accumulate=total;*/
input Ryga /normalize=absolute;
target Vilnius / normalize=absolute measure=msqrdev
compress =(localpct=22)
expand =(localpct=22);
run;
ods graphics off;
ods graphics on;
/*AsymmetricP0- 937 žingsniai*/
proc similarity data=WORK.auto out=iseitis
print=all plot=all;
/*id date interval=month accumulate=total;*/
input Ryga ;
target Vilnius /
compress =(localpct=100)
expand =(localpct=100);
run;
ods graphics off;

```



```
ods graphics on;
/*AsymmetricP2- 207 žingsniai*/
proc similarity data=WORK.auto out=iseitis
print=all plot=all;
/*id date interval=month accumulate=total;*/
input Ryga/ normalize=absolute ;
target Vilnius / normalize=absolute
compress =(localpct=0)
expand =(localpct=0);
run;
ods graphics off;
```

```
PROC IMPORT OUT= WORK.auto DATAFILE= "C:\Users\W7\Desktop\daugiamatis" DBMS=xlsx REPLACE;
SHEET="Sheet1";
GETNAMES=YES;
format date yymmdd10.;
RUN;
```

```
proc print data=WORK.auto;
run;
ods graphics on;
/*symmetric1- 795 žingsniai*/
proc similarity data=WORK.auto out=iseitis
print=all plot=all;
/*id date interval=month accumulate=total;*/
input Baltic /normalize=absolute;
target Ryga / normalize=absolute measure=msqrdev
compress =(localpct=1)
expand =(localpct=1);
run;
ods graphics off;
```

```
PROC IMPORT OUT= WORK.auto DATAFILE= "C:\Users\W7\Desktop\Magistrinis\Programa\index_OMX - Copy" DBMS=xlsx
REPLACE;
SHEET="1";
GETNAMES=YES;
format date yymmdd10.;
RUN;
proc print data=WORK.auto;
run;
proc corr data = WORK.auto spearman kendall;
var OMXT_Value OMXR_Value OMXV_Value OMXBBPI_Value;
title 'Spirmeno koreliacija';
run;
```