



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Matas Kiškūnas

**AUTOMATIZUOTOS PREKYBOS SISTEMOS, PAGRĪSTOS HARMONINĒMIS KAINŪ
KONFIGŪRACIJOMIS, SUKŪRIMAS IR TYRIMAS**

Baigiamasis magistro projekts

Vadovas

Doc. dr. Vygandas Vaitkus

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA

**AUTOMATIZUOTOS PREKYBOS SISTEMOS, PAGRĮSTOS HARMONINĖMIS KAINŲ
KONFIGŪRACIJOMIS, SUKŪRIMAS IR TYRIMAS**

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

Vadovas

(parašas) Doc. dr. Vygandas Vaitkus
(data)

Recenzentas

(parašas) Doc. dr. Vidas Raudonis
(data)

Projektą atliko

(parašas) Matas Kiškūnas
(data)

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos

(Fakultetas)

Matas Kiškūnas

(Studento vardas, pavardė)

Valdymo technologijos, 621H66001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Automatizuotos prekybos sistemos, pagrįstos harmoninėmis kainų konfigūracijomis, sukūrimas ir tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. Gegužės 17 d.
Kaunas

Patvirtinu, kad mano **Mato Kiškūno** baigiamasis projektas tema „Automatizuotos prekybos sistemos, pagrįstos harmoninėmis kainų konfigūracijomis, sukūrimas ir tyrimas“ 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.

(vardą ir pavardę įrašyti ranka)

(parašas)

Kiškūnas, Matas. Automatizuotos prekybos sistemos, pagrįstos harmoninėmis kainų konfigūracijomis, sukūrimas ir tyrimas. *Valdymo sistemų magistro* baigiamasis projektas / vadovas doc. dr. Vygandas Vaitkus; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, automatikos katedra.

Mokslo kryptis ir sritis: Elektros ir elektronikos inžinerija, Technologiniai mokslai

Reikšminiai žodžiai: *automatinė, prekyba, harmoninės, konfigūracijos*

Kaunas, 2017. 68 p.

SANTRAUKA

Darbo tikslas – sukurti ir ištirti harmoninėmis kainų konfigūracijomis paremtą automatinę prekiavimo sistemą, kuri prekiautų valiutų poromis. Darbe buvo naudojama Metatrader4 prekiavimo platforma ir MQL4 programavimo kalba sukurta sistema. Automatinėje prekiavimo sistemoje buvo optimizuojami trys minimumų ir maksimumų paieškos parametrai. Ištirtos ir palygintos sistemos su skirtingais optimizavimo ir darbo laikotarpiais. Ištirtas sistemos darbas su Euro/Svaro pora ir sudaryta prekiavimo strategija, kuria galima siekti pelno.

Kiškūnas, Matas. Development and Investigation of Automated Trading System Based on Harmonic Configurations of Price *Master's* thesis in *Control Systems* / supervisor assoc. prof. Vygandas Vaitkus. Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of automation.

Research area and field: Electrical and Electronics Engineering, Technological Sciences

Key words: automated, forex, harmonics, pattern

Kaunas, 2017. 68 p.

SUMMARY

The aim of this study is to create and analyze automated trading system based on harmonic patterns. This system was made to work in Foreign Exchange Market. System was made on Metatrader4 platform using MQL4 programming language. Three system parameters were optimized to make biggest profits with Euro/Pound pair. Automated trading system was optimized and tested on different timeframes. And on the basis of these tests were made successful trading strategy.

Turinys

Įvadas	3
1. Techninė analizė.....	5
1.1. Techniniai indikatoriai.....	5
1.1.1. Slenkantys vidurkiai	6
1.1.2. Slankiųjų vidurkių konvergencija/divergencija (MACD).....	6
1.1.3. Bolingerio juostos	7
1.2. Geometrinės kainų konfigūracijos.....	7
1.2.1. Simetriškas trikampis	8
1.2.2. Vėliavos ir gairės.....	8
1.2.3. Puodelis su rankenėle	9
1.2.4. Harmoninės kainų konfigūracijos	11
1.3. Ankstesni tyrimai.....	12
2. Programinė įranga ir prekyavimo strategija.....	13
2.1. Programinė įranga ir jos galimybės	13
2.1.1. MetaTrader4 platformos ypatumai.....	13
2.1.2. MQL4 programavimo kalba.....	14
2.1.3. Strategijos testavimas ir optimizavimas.....	14
2.2. Strategijos sudarymas	18
2.2.1. Harmoninėmis kainų konfigūracijomis paremta strategija	18
2.2.2. Programavimo eiga ir programos algoritmas	22
3. Tiriamoji dalis	26
3.1. Sistema, apmokyta su 9 metų duomenimis	27
3.1.1. Keturių valandų grafikas (H4).....	27
3.1.2. Keturių valandų grafikas su papildoma įėjimo į rinką sąlyga	29
3.1.3. Valandinis grafikas (H1)	31
3.1.4. Trisdešimties minučių grafikas (M30).....	32
3.2. Sistema, kurios parametrai po tam tikro laikotarpio perrenkami iš naujo.....	34

3.2.1.	2 metų laikotarpio optimizavimas	34
3.2.2.	3 metų laikotarpio optimizavimas	39
3.2.3.	4 metų laikotarpio optimizavimas	44
3.2.4.	5 metų laikotarpio optimizavimas	49
3.2.5.	Ištirtų sistemų palyginimas.....	54
išvados ir pasiūlymai.....		56
Literatūros sąrašas		58
Priedai.....		60
1 Priedas		60
2 Priedas		65

ĮVADAS

Vertybinių popierių techninė analizė jau seniai yra naudojama praktikoje, tačiau mokslininkai nesutaria ar remtis vien tik technine analize yra tesinga. Techninė analizė yra paremta tuo, kad vadovaujamosi vien tik kainos ir prekybos apimtys kitimu, nesigilinant dėl ko tai vyksta. Kai kurių kainos judėjimų atpažinimas yra reikšminis - jie prisideda prie specifinių modelių formacijos –, o kiti paprasčiausiai yra triukšmas, į kurį nereikėtų kreipti dėmesio. Daugeliu atveju prekiaujama tik žmogaus akimis įvertinus modelio susidarymą. Tačiau šiuolaikinės technologijos leidžia tai padaryti naudojantis kompiuteriniais algoritmais.

Taigi, kaip vieną iš svarbiausių techninės analizės sričių galima išskirti automatinę techninės analizės modelių, kuriuos atpažįsta investuotojai, suradimą. Tam yra kuriamos ekspertinės sistemos, kurių prekavimo algoritmą ir taisykles aprašo patyrę prekiautojai. Šiuose algoritmuose nurodoma kokiems parametrams esant galima spręsti kad modelis susidarė ir reikia įeiti į rinką. Taip pat nurodoma kaip elgtis susidarius kainų konfigūracijoms.

Pasinaudojant ekspertų žiniomis galima ne tik rasti kainų modelius, bet ir juos radus automatiškai prekiauti. Tokios sistemos vadinamos automatinėmis prekavimo sistemomis. Šios sistemos pašalina žmogiškąjį faktorių iš prekavimo strategijos. Taip pat žmogui nebereikia gaišti laiko ir daryti sandorius rankiniu būdu. Prieš pradėdant eksploatuoti automatines prekavimo sistemas, jas galima ištestuoti su istoriniais duomenimis. Dėl šių priežasčių tokių sistemų paklausa nuolat auga.

Darbo tikslas: ištirti ar strategijos, paremtos harmoninių kainų konfigūracijų principais, gali nešti pelną. Taip pat sukurti automatinę prekavimo sistemą.

Buvo iškelti tokie darbo uždaviniai:

- Išanalizuoti vertybinių popierių techninės analizės metodus.
- Išanalizuoti rinkoje naudojamą programinę įrangą ir jos galimybes.
- Suformuoti harmoniniais kainų modeliais praemtą prekavimo strategiją.
- Sukurti imitacinę automatinę prekavimo sistemą panaudojant sukurtą strategiją.
- Rasti optimalų parametų rinkinį sukurtai sistemai.
- Testuoti sistemą su istoriniais duomenimis

Iškelta hipotezė: prekiaujant harmoniniais modeliais paremta strategija galima siekti pelno. Remiantis eksperimentiniais tyrimais siekiama šią hipotezę patvirtinti arba paneigti.

Darbą sudaro trys dalys. Teorinės darbo dalies pirmame skyriuje apibrėžiama techninė analizė, supažindinama su techniniais indikatoriais bei žinomiausiais techninės analizės grafinais

modeliais. Antrame teorinės dalies skyriuje analizuojama programinė įranga ir jos galimybės, traip pat aprašoma pasirinkta prekiavimo strategija. Trečioje dalyje pristatomi tyrimo rezultatai.

1. TECHNINĖ ANALIZĖ

Techninė analizė - visuma finansinės analizės technikų, metodų, kuriomis siekiama nuspėti biržos kursus remiantis kursų kitimo (pavyzdžiui, tendencijų) duomenimis, istorija. Tokios analizės tikslas – nustatyti palankių finansinių priemonių (vertybinių popierių, akcijų, valiutų porų) pirkimo ar pardavimo laikotarpį, momentą. Techninė analizė dažniausiai naudojama valiutų porų prekyboje (angl. Foreign exchange market). Ji remiasi įvairiais indikatoriais[1]. Pagrindiniai indikatoriai bus apžvelgti šiame skyriuje.

Techninėje analizėje, priešingai nei fundamentalioje, neatsižvelgiama į makroekonominius, ūkio šakų ar atskirų įmonių (mikroekonomika) duomenis. Daroma prielaida, kad visa sprendimui priimti reikalinga informacija atsispindi kainos ir apimčių kreivėje. Taip pat teigiama, kad geometriniai kainų modeliai, konfigūracijos yra buvę istorijoje ir linkę kartotis.

Šiuolaikinės techninės analizės pagrindą sudaro trys aksiomos:

1. Kaina visapusiškai atspindi visas rinkos jėgas. Visa rinkos informacija ir jėgos bet kuriuo metu atsispindi kainose.
2. Kainos kinta pagal tam tikras tendencijas, kurias galima atpažinti ir taip susikurti progas pasipelnyti.
3. Kainos dinamika yra istoriškai pasikartojanti[2].

Techninė analizė, lyginant su fundamentalia, turi ir trūkumų ir privalumų. Pagrindinis privalumas, kad techninei analizei nereikia daug duomenų. Ji paremta tik kainos ir prekybos apimtys grafiku. Dar vienas privalumas - techninė analizė gali tapti savaime išsipildančia pranašyste. Kai daugelis panašius įrankius naudojančių ir tais pačiais principais besiremiančių investuotojų kartu keičia pasiūlą ir paklausą, kainos gali imti judėti numatyta kryptimi. Pagrindinis trūkumas – techninėje analizėje yra remiamasi daugybe indikatorių ir instrumentų. Dažnai nutinka taip, kad instrumentai duoda skirtingus signalus ir reikia nuspręsti, kuriuo pasitikėti.

1.1. Techniniai indikatoriai

Techniniai indikatoriai, tai tam tikri duomenys/signalai, kurie apskaičiuojami iš prekybos apimčių ir kainos grafiku, pritaikius formules. Per ilgametę vertybinių popierių rinkos istoriją yra sukurta daugybė techninių indikatorių, kuriais bandoma prognozuoti

akcijų, valiutų porų kainų kitimo kryptį. Šiame skyriuje pateikiami pagrindiniai techniniai indikatoriai naudojami „forex“ (angl. Foreign exchange market) rinkoje[3].

1.1.1. Slenkantys vidurkiai

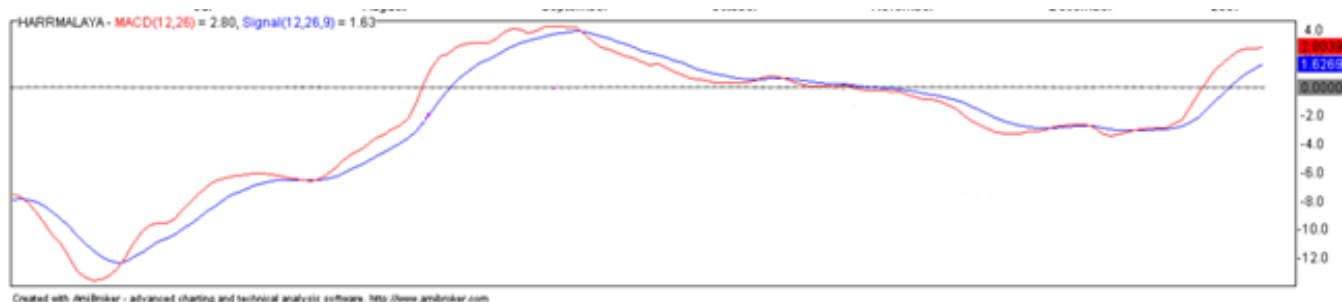
Vieni iš populiariausių indikatorių – slenkantys vidurkiai – padeda prekybininkams patikrinti vyraujančias tendencijas, nustatyti naujas tendencijas ir peržiūrėti užsitęsusias tendencijas, kurios netrukus gali keistis. Kaip rodo pats pavadinimas, grafiko linijos, atitinkančios „vidutinius“ trumpalaikius kainų svyravimus, yra vyraujančios, ir todėl jos parodo ilgalaikę kainų tendenciją. Išskiriami trijų tipų slenkantys vidurkiai:

Paprastasis slenkantis vidurkis parodo kiekvieną kainos punktą per nurodytą laikotarpį. Prekybininkas nustato, ar naudojama aukščiausia, žemiausia, užsidarymo ar atsidarymo kaina, tada paskaičiuojamas vidurkis vidurkis, kuris sudaro liniją.

Eksponentinis slenkantis vidurkis labiau remiasi naujausiais kainų duomenimis kitu būdu. Eksponentinis slenkantis vidurkis apskaičiuojamas naujausios kainos procentinę išraišką dauginant iš ankstesniojo laikotarpio vidutinės kainos[4].

1.1.2. Slankiųjų vidurkių konvergencija/divergencija (MACD)

Techninis slankiųjų vidurkių konvergencijos/divergencijos indikatorius (angl. Moving Average Convergence/Divergence, sutrumpintai MACD) — tai techninis indikatorius, parodantis dviejų slankiųjų vidurkių sąryšį. Dažniausiai naudojamas su 12 ir 26 dienų periodais. Tame pačiame grafike atvaizduojama ir 9 dienų slenkančio vidurkio linija. Priklausomai nuo šių dviejų linijų susikirtimų daromi sprendimai (1.1 paveikslas). MACD efektyviausias, kai rinkos svyravimo amplitudė yra didelė.



1.1 pav. Galimi pirkimo ir pardavimo momentai, remiantis MACD[5]

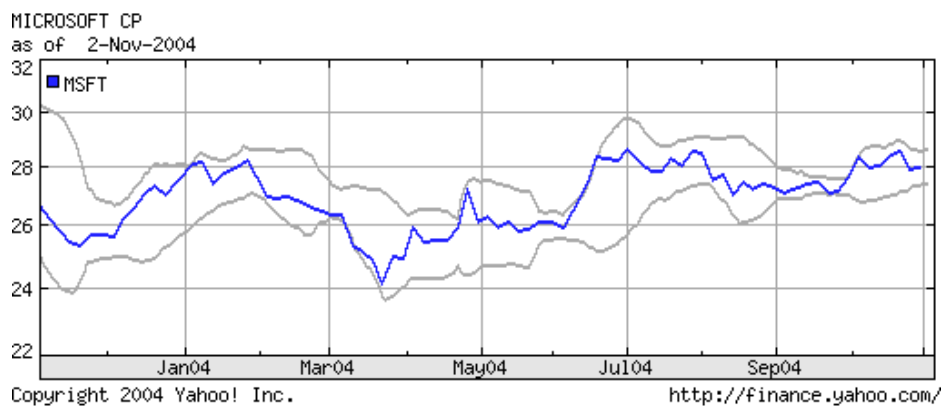
Pagrindinė prekiavimo su MACD taisyklė paremta indikatoriaus susikirtimu su signaline linija: kai MACD nusileidžia žemiau signalinės linijos – reikia parduoti, o kai kyla aukščiau jos – pirkti[5].

1.1.3. Bolingerio juostos

Kai kada akcijų kainos svyruoja tam tikrose ribose gana ilgą laiką. Šiuo atveju naudinga informacija gali būti akcijos kainos svyravimų ribos. Paprasčiausias būdas skaičiuoti apatinę ir viršutinę ribą yra naudojant tokią formulę: prie suskaičiuoto slenkamojo vidurkio pridedami 5% kainos (skaičiuodami viršutinę ribą) ir atimami 5% (skaičiuodami apatinę ribą). Tačiau šis metodas kai kuriais atvejais netinka. Kartais kainos svyravimas yra didelis, skirtumas tarp viršutinės ir apatinės ribos pasidaro per siauras, o kai kainos svyravimas nedidelis, skirtumas per didelis.

Geresnį sprendimą pasiūlė Džonas Bolingeris (John Bollinger). Jis pasiūlė viršutinę ribą skaičiuoti prie slenkamojo vidurkio pridedant dvigubą akcijos kainos kitimo standartinį nuokrypį. Skaičiuojant apatinę ribą iš slenkamojo vidurkio atimama dviguba standartinio nuokrypio reikšmė.

Bollingerio juostą sudaro 3 linijos: viršutinė, centrinė ir apatinė (1.2 paveikslas). Centrinė linija – tai slenkamasis akcijos kainos vidurkis. Viršutinė linija - centrinės linijos ir dvigubo standartinio nuokrypio suma. Apatinė linija - centrinės linijos ir dvigubo standartinio nuokrypio suma[6].



1.2 pav. Bolingerio juostos

1.2. Geometrinės kainų konfigūracijos

Tęstinės konfigūracijos nurodo tendencijos pauzę ir reiškia, kad ankstesnė kainų kryptis po tam tikro laiko pasikartos. Šiame skyriuje bus apžvelgti kainų kanalai, simetriški trikampiai ir vėliavos ir gairėlės.

1.2.1. Simetriškas trikampis

Simetriškasis trikampis – tai tęstinė konfigūracija, susidariusi tose rinkose, kuriose nesimato aiškios krypties. Konfigūracijoje yra bent du žemesni maksimumai ir du aukštesni minimumai, kurie atrodo suartėjantys. Kai šiuos taškus jungiančios tiesės pratęsimas, jos susikerta ir sudaro simetrišką trikampį.

Remiantis simetriškuoju trikampiu galima prognozuoti ir kainos pokyčio mastą, ir laiką. Kai konfigūracija yra baigta, kaina ir apimtis mažėja, kol kerta trikampio kraštinę (1.3 paveikslas). Kraštinę kirtusios kainos yra linkusios pakisti tokiu intervalu, kuris prilygsta trikampio pagrindui arba yra už jį didesnis (žr. toliau pateiktą pavyzdį). Laiko atžvilgiu trikampis turėtų būti kertamas tuo momentu, kuris maždaug atitinka pusiaukelę arba du trečdalius atstumo nuo pagrindo iki viršūnės, t. y. trikampio aukščio [9, 58 psl.].

Proveržis galimas bet kurioje trikampio pusėje. „Bulių“ rinkoje simetriškasis trikampis kertamas ta pačia kryptimi, kokia buvo ankstesnės „bulių“ tendencijos kryptis. „Meškų“ rinkoje simetriškasis trikampis kertamas ta pačia kryptimi, kokia buvo ankstesnės „meškų“ tendencijos kryptis. Simetriškojo trikampio „bulių“ rinkoje pavyzdys:



1.3 pav. Simetriškas trikampis

1.2.2. Vėliavos ir gairės

Šios dvi panašios tęstinės konfigūracijos paprastai susidaro ryškesnio kainų judėjimo trajektorijos vidurio taške ir yra tik trumpos pauzės dinamiškoje rinkoje. Jas galima atpažinti

ir atskirti pagal jų „korpuso“ pavidalą; vėliavos atveju tai stačiakampis, šiek tiek pakrypęs priešinga tendencijai kryptimi, o gairėlės atveju – trikampis.

Vėliavos ir gairėlės panašios ir savo forma, ir pobūdžiu. Abi konfigūracijos žymi nedidelę kainų judėjimo konsolidaciją, nors tam, kad jos būtų laikomos tikrai tęstinėmis konfigūracijomis, reikia, kad jos patvirtintų ankstesnę tendenciją.

Prieš vėliavas ir gaireles paprastai būna ryškus tendencijos krypties šuolis ar nuopuolis, kuris grafike atrodo kaip vėliavos ar gairės „kotas“ (1.4 paveikslas). Po konfigūracijos proveržio kainos nueis minimalų atstumą, kuris turėtų prilygti koto aukščiui.



1.4 pav. Vėliavos ir gairėlės

1.2.3. Puodelis su rankenėle

„Puodelio su rankenėle“ modeliu (1.5 paveikslas) naudojami nemaža dalis prekyautojų, kurie remiasi technine analize. Skiriamos dvi šios konfigūracijos dalys: puodelis ir rankenėlė. Po akcijos kainos kilimo pirmiausia susiformuoja puodelis. Po to formuojasi rankenėlė, po kurios seka persilaužimas, signalizuojantis tolesnį kainos kilimą.

Toliau pateikiamos pagrindinės „puodelio su rankenėle“ formacijos charakteristikos:

- Tendencija. Vykstant formacijai turėtų išlikti ankstesnė tendencija. Ji turėtų būti bent jau kelių mėnesių senumo. Kuo senesnė tendencija, tuo didesnė tikimybė, kad formacija bus teisinga.

- Puodelis. Puodelis turėtų būti „U“ raidės formos ir panašus į dubenį apvalia apačia. Jei puodelio apačia yra „V“ raidės formos, jis laikomas per aštrus. Švelnesnė „U“ forma užtikrina, kad formacija yra teisinga ir „U“ apatiniame taške pasiekiami atraminė riba, t.y. kai daugelis investuotojų mano, jog akcijos kaina kils dėl padidėjusios paklausos. Ideali formacija turi vienodo lygio viršūnes abiejuose puodelio šonuose, tačiau tai ne visada būtina.
- Puodelio gylis. Idealiu atveju puodelio gylis turėtų atitikti 1/3 arba mažesnę dalį buvusio pakilimo. Tačiau nepastoviose ir greitai besikeičiančiose rinkose ši riba gali svyruoti nuo 1/3 iki 1/2. Ekstremaliu atveju, maksimali riba yra 2/3.



5 pav. Puodelis su rankenėle

- Rankenėlė. Po to kai dešiniam puodelio šone susiformuoja viršūnė, įvyksta kainos kritimas ir kilimas iki buvusio lygio, kas suformuoja rankenėlę. Kartais rankenėlė būna vėliavos arba riešuto formos. Rankenėlė – tai paskutinis formacijos etapas prieš didįjį persilaužimą. Rankenėlė paprastai sudaro iki 1/3 puodelio gylio. Kuo mažesnė rankenėlė, tuo stipresni rinkoje yra pirkėjai ir tuo didesnis būna persilaužimas. Kartais naudinga palaukti kol persilaužimas įvyks virš pasipriešinimo ribos, kuri yra tarp puodelio viršūnių.
- Trukmė. Puodelio formacija trunka nuo 1 iki 6 mėnesių, kartais gali trukti ilgiau, jei pasirenkamas savaitinis grafikas. Rankenėlė idealiu atveju susiformuoja per 1-4 savaites.
- Apyvarta. Persilaužimo taške apyvarta turi stipriai padidėti. Reikšmingas yra apyvartos pakilimas apie 50% lyginant su 50 dienų vidutine apyvarta. Idealiu atveju apyvarta pakyla 100% ir daugiau.
- Tikslas. Po persilaužimo planuojamas kilimas matuojamas atstumu tarp dešinėsios puodelio viršūnės ir puodelio dugno (minimumo).

Kaip ir kitiems grafiniams techninės analizės modeliams, svarbiausia yra esminės modelio charakteristikos, o ne smulkios detalės. Puodelis turi būti dubens formos, o rankenėlė yra trumpas kainos kritimas bei kilimas, po kurio seka persilaužimas su išaugusia apyvarta [10].

1.2.4. Harmoninės kainų konfigūracijos

Harmoninės kainų konfigūracijos yra geometrinės kainų konfigūracijos, kurių kainos krypties keitimo taškus apibrėžia Fibonači (Fibonacci) lygiai (Fibonacci). Harmoninėmis konfigūracijomis siekiama nuspėti kainų kitimą. Šios formacijos yra labai tikslios, reikalaujančios kelių tikslų kainų judesių iš eilės. Iš pirmo žvilgsnio gali pasirodyti, kad harmoninės konfigūracijos kainų kitimo grafike kartojasi labai dažnai, tačiau išanalizavus atidžiau, paaiškėja, kad krypties keitimo taškai dažnai nesutampa su fibonači lygiais. Tokiu atveju kainų konfigūracijos yra negaliojančios. Tai gali būti strategijos privalumas, nes kainų pasikeitimas tiksluose fibonači lygiuose reikalauja prekiautoja kantrybės ir išlaukimo iki idalaus momento. Dažnai harmoninės kainų konfigūracijos sutampa su kitais būdais randamomis kainų konfigūracijomis. Būtina pabrėžti, kad harmoninės kainų konfigūracijos, kaip ir visos kitos, gali privesti prie klaidingo pirkimo ar pardavimo, todėl būtina taikyti rizikos valdymą.

Harmoninių kainų konfigūracijų yra gana nemažai. Pačios populiariausios yra Gartley, drugio, šikšnosparnio ir krabo konfigūracijos[11]. Jomis paremta prekybos strategija ir buvo tiriama šiame darbe. Plačiau šios konfigūracijos išnagrinėtos 2 skyriuje.

Fibonači seka

Fibonači skaičiai – tai esminis harmoninių konfigūracijų elementas. Šie skaičiai turi seką, kuri dažnai sutinkama gamtoje vykstančiuose procesuose. Prekyboje ji veikia dėl to, kad yra populiarai, ir naudojama daugybės prekiautojų. Fibonači skaičių seka gaunama labai paprastai – sudedant du praėjusius skaičius, t.y., 0, 1, 1, 2, 3, 5, 8, 13 ir t.t.. Laikoma, kad esminiai Fibonači (Fibonacci) lygiai yra 23,6%, 38,2%, 61,8% ir 100% tiksliau tariant – 0.236, 0.382, 0.618, 1. Jeigu praleistume kelis skaičius sekos pradžioje ir pamatuotume santykį tarp greta esančių skaičių, visuomet gautume santykį: **0,618** (pavyzdžiui, $34/55=0,618$). O santykis tarp kas antrinių skaičių sekos narių visuomet yra **0,382** (pavyzdžiui, $13/34=0,382$). Tokiu pačiu principu gaunami ir kiti fibonači lygiai. Šie ryšiai tarp konkrečių sekos skaičių yra pagrindas apibrėžiant bendruosius santykius, naudojamus kainų grįžimo ir kainų pratęsimo judesiams tendencijoje nustatyti[12].

1.3. Ankstesni tyrimai

Gene I. Sher sukūrė ir ištyrė kainos prognozavimo sistemą, kuri yra paremta neuroniniais tinklais, tačiau atsižvelgia ir į tai ar susidariusi geometrinė konfigūracija. Buvo iškelta ir patvirtinta hipotezė, atsižvelgiant į geometrines konfigūracijas ir naudojant dirbtinį neuroninį tinklą galima pasiekti didesnio pelningumo nei vien tik naudojant neuroninį tinklą. Buvo pasiektas 52% pelnas per 15 dienų laikotarpį su 1:50 svertu[19].

J. Nedzveckas ir R. Dapkus ištyrė dažniausiai rinkoje naudojamus instrumentus ir metodus naudojamus valiutų kursų analizei ir prognozavimui. Duomenų apdorojimui bet pertvarkymui dažnai naudojami MS Excel, MS Access ir Global Server Desktop Edition prakaminiai paketai. Investavimo strategijų pelningumui ir patikimumui tirti gali būti naudojama Omega Prosuite 2000i, Tradestation 2000i programinė įranga. Techninės analizės grafinių metodų tyrimui – Omega Tradestation 2000i, Web Trader Charts programinė įranga. Taip pat šiame straipsnyje J. Nedzveckas palygino trijų, technine analize paremtų, strategijų pelningumą: slankiaisiais vidurkiais paremta strategija, MACD paremta strategija, RSI paremta strategija. Naudojant RSI indikatorių buvo pasiektas geriausias pelningumas (18,46% per pusę metų)[20].

Murat Ozturk ir kiti sukūrė valiutų porų prekybai pritaikytą sistemą, kuri buvo paremta techniniais indikatoriais paremtomis taisyklėmis. Buvo išvestos matematinės formulės, paremtos bolingerio juostomis, divergencija ir kitais indikatoriais, kurios aptinka specialias konfigūracijas kainos grafike. Ši Sistema buvo įgyvendinta trimis etapais. Pirmame etape, kiekvienas indikatorius buvo testuojamas atskirai kvalifikacijai. Tuomet iš visų kvalifikaciją praėjusių indikatorių, paskirstant reikšmingumo svorius yra kuriama bendra Sistema. Geriau kvalifikacijos etapą praėję indikatoriai gauna didesnę svorį bendroje sistemoje. Svoriai buvo skirstomi dviem būdais: panaudojant genetinį algoritmą ir panaudojant euristinės paieškos metodą. Paskutiniame etape testuojama sukurtas algoritmas. Ši Sistema buvo testuota su EUR/USD ir GBP/USD poromis, trijuose skirtingų laikų (angl. Timeframe) grafikuose. Ten kur buvo naudojamas euristinės paieškos metodas, rezultatai neigiami. Tačiau naudojant genetinį algoritmą buvo gauti teigiami rezultatai. Naudojant bendrai sukurtą taisyklę (panaudojant kelis indikatorius su skirtingais svoriais), pelnas buvo visuomet didesnis nei naudojant tik vieną geriausią indikatorių. Ištirtoje prekyvimo sistemoje nebuvo atsižvelgta į transakcijos kainą (angl. Spread), nebuvo naudojamos “take profit” ir “stop loss” ribos[21].

Šiame darbe, kitaip nei senesniuose bandymuose, tiriama harmoninėmis formacijomis paremta prekiavimo sistema, kai harmoninių konfigūracijų taškai susidaro ZigZag indikatoriaus ekstremumuose. Taip pat ieškomi optimalūs ekstremumų paieškos parametrai bei bandoma pagerinti pelną į transakcijas įeinant ne iškart pasitvirtinus konfigūracijai, bet dar ir palaukiant kainos judėjimo tinkama linkme.

2. PROGRAMINĖ ĮRANGA IR PREKIAVIMO STRATEGIJA

Automatinės prekiavimo sistemos yra dar kitaip vadinamos mechaninėmis prekiavimo sistemomis arba algoritmais paremtu prekiavimu. Šios sistemos leidžia prekiautojams susikurti specifines taisykles, kuriomis vadovaujantis automatiškai atliekami pirkimo ir pardavimo sandoriai. Tokia sistema turi daugybę privalumų lyginant su neautomatizuotu prekiavimu. Vienas iš privalumų, kad strategijos efektyvumą galima nesunkiai patikrinti su istoriniais duomenimis ir sukurtą sistemą optimizuoti. Taip pat yra padidinamas transakcijų tikslumas ir įėjimo į rinką ir išėjimo iš rinkos greitis. Dar vienas privalumas - panaikinamos žmogaus emocijos. Deja automatinės prekiavimo sistemos turi ir trūkumų. Dėl kompiuterio gedimų ar nutrūkusio interneto ryšio, gali būti neatlikti reikiami sandoriai. Todėl sistemas rekomenduojama nuolat stebėti. Taip pat yra peroptimizavimo pavojus. Sistemos parametrai parenkami tokie, kad testuojant su istoriniais duomenimis nešūtų didelį pelną, tačiau tokia sistema gali būti labai rizikinga[13].

2.1. Programinė įranga ir jos galimybės

Rinkoje yra nemažai prekiavimui skirtų platformų. Viena populiariausių yra MetaTrader4. Ši platforma leidžia analizuoti rinką, prekiauti valiutų poromis rankiniu ir automatiniu būdu. Vienas iš platformos įrankių yra MQL4 programavimo kalbos editorius. Šia kalba galima kurti automatinio prekiavimo sistemas ir jas testuoti, optimizuoti ir naudoti MetaTrader4 platformoje.

2.1.1. MetaTrader4 platformos ypatumai

Metatrader4 (MT4) programa yra populiari prekiavimo platforma, kuria naudojasi daugelis brokerių. Šios programos pagalba galima:

- Gauti naujienas online režimu;
- Atlikti pirkimo/pardavimo operacijas;

- Kontroliuoti atidarytas pozicijas ir laukiančius užsakymus;
- Atlikti techninę analizę;
- Kurti ekspertines sistemas, indikatorius, skriptus, pasinaudojant MQL4 kalbos editoriumi;
- Testuoti sukurtas programas;
- Optimizuoti ekspertinių sistemų parametrus.

Magistrinio darbo užduočių atlikimui buvo pasirinkta ši platforma, nes joje yra integruoti visi reikiami įrankiai sukurtos prekiavimo strategijos įgyvendinimui ir optimizavimui, automatinės prekiavimo sistemos įgyvendinimui[15].

2.1.2. MQL4 programavimo kalba

MetaQuotes (MQL4) kalba yra programavimo kalba pritaikyta prekiavimo strategijoms įgyvendinti. Šią kalbą sukūrė MetaQuotes Software kompanija. Pasinaudojant šia kalba galima kurti programas, kurios prekiauja automatiškai pagal sukurtą prekybos strategiją. MQL4 turi daug funkcijų ir plačią indikatorių biblioteką, kurių pagalba galima kurti ekspertines sistemas, naujus indikatorius, skriptus.

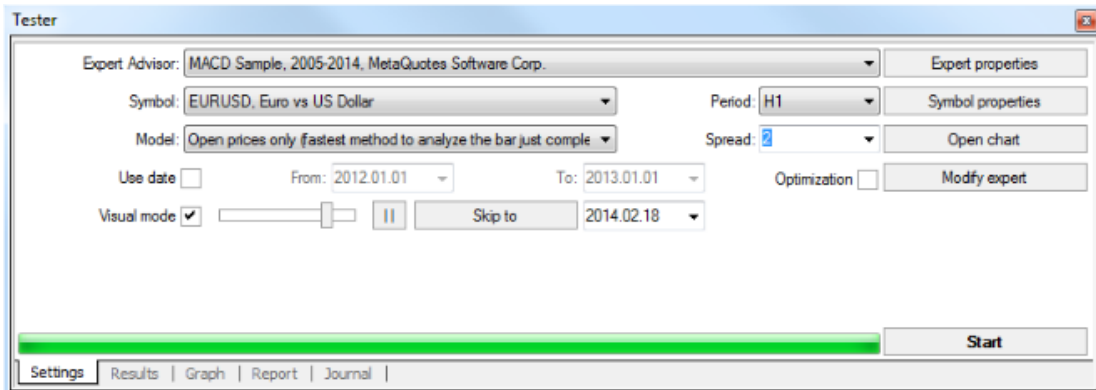
Skriptas yra pati paprasčiausia programa, kurią paleidus veiksmai įvykdomi tik vieną kartą. Šioje programoje yra būtina tik OnStart funkcija, kuri įvykdoma iškart po skripto paleidimo. Šiek tiek sudėtingesnė yra indikatoriaus programa. Ji skirta kainos kitimo analizavimui. Paleidus indikatoriaus programą veiksmai atliekami iškart po paleidimo (suveikia OnInit funkcija), ir programa cikliška kartojama po kiekvieno iš serverio gauto signalo. Indikatoriaus programoje yra būtinos trys funkcijos OnInit, OnCalculate ir OnDeinit. OnInit funkcija įvykdo reikiamus veiksmus paleidimo metu, OnCalculate funkcija vykdoma kiekvino ciklo metu, o OnDeinit funkcija įvykdoma indikatoriaus išjungimo metu.

Sudėtingiausia MQL4 programa yra ekspertinė sistema. Tai sistema kuri automatiškai gali atlikti pirkimo ir pardavimo veiksmus pagal sukurtą prekiavimo strategiją. Ši programa turi tas pačias būtinas funkcijas kaip ir indikatoriaus programa, tačiau skirtumas tas, kad OnInit funkcija yra įvykdoma tik tada kai gaunamas pirmas kainos signalas iš serverio ir ekspertinė sistema gali vykdyti transakcijas [16].

2.1.3. Strategijos testavimas ir optimizavimas

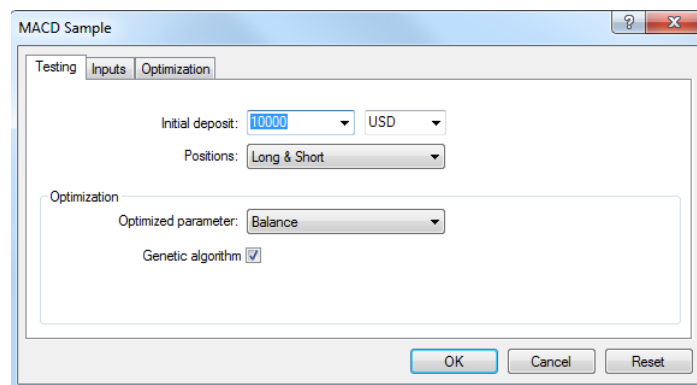
MetaTrader4 programa leidžia netik kurti ekspertines sistemas, bet ir jas testuoti prieš naudojant. Pasinaudojant istoriniais duomenimis, galima patikrinti prekiavimo strategijos efektyvumą. Testuojant ekspertinę sistemą testavimo lange reikia atlikti šiuos veiksmus:

- Pasirinkti ekspertinę sistemą ir nustatyti jos įėjimus
- Pasirinkti valiutų porą ir periodą
- Pasirinkti vieną iš trijų modeliavimo metodų, bei pasirinkti kainos paplitimą (“spread”)
- Pasirinkti testavimo laikotarpį



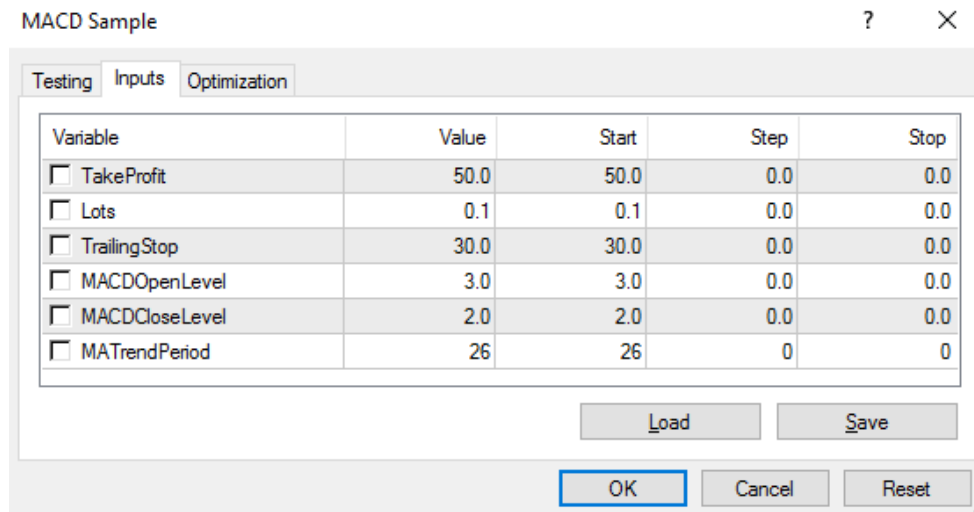
2.1 pav. Tester langas

Kai ekspertinė Sistema jau pasirinkta ir parametrai nustatyti, reikia nustatyti papildomus parametrus Expert properties lange. Paspaudus mygtuką Expert properties atsiranda langas su trimis kortelėmis: Testing, Inputs ir Optimization.



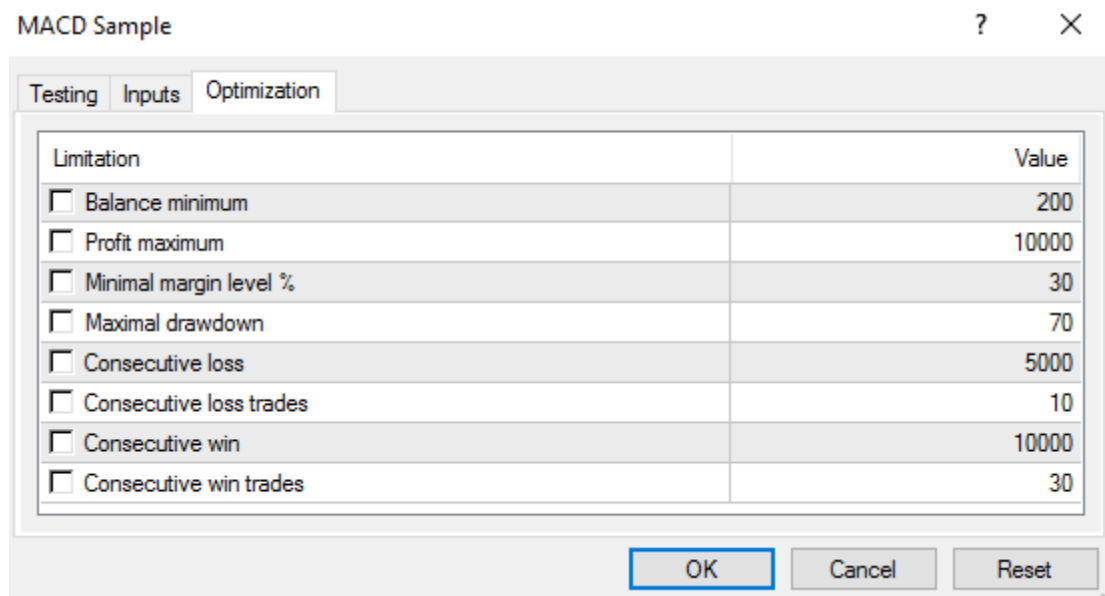
2.2 pav. Testing skiltis

Testing lange pasirenkame valiutą, depozitą, leistinus veiksmus (pirkti/parduoti). Taip pat galima įjungti optimizavimą ir pasirinkti tikslo funkciją.



2.3 pav. Inputs skiltis

Inputs lange lentelėje pateikiami visi įėjimai. Vertės galima keisti lentelėje. Ekspertinės programos kodo keisti nebūtina. Start, Step ir Stop stulpeliai yra naudojami tik optimizavime. Testavime jie nereikalingi.



2.4 pav. Optimizavimo apribojimai

Optimizavimo lange galima pasirinkti ir nustatyti parametrus, kuriuos pasiekus būtų nutrauktas optimizavimas. Šis langas testavimo operacijai įtakos neturi, naudojamas tik optimizavime.

Modeliavimo metodai

Istoriniai duomenys yra laikomi terminale kaip stulpeliai HST formatu. Duomenys gali būti panaudoti testuojant ekspertines sistemas. Kaikuriais atvejais informacijos gali būti negana,

pavyzdžiui norint testuoti valandiniame grafike. Tais atvejais duomenys gali būti importuoti iš kitų šaltinių. Testavimui gali būti panaudoti trys būdai:

- **Naudojant tik atsidarymo kainą.**
Tai greičiausias iš visų metodų. Imamos tik pasibaigusių žvakių atsidarymo kainos. Žvakė skaitoma pasibaigusia, kai jau yra atsiradusi nauja žvakė.
- **Naudojant kontrolinius taškus.**
Tai dažniausiai naudojamas metodas testuojant ekspertines sistemas. Šiame metode modeliuojant yra panaudojamas artimiausias mažesnis laikotarpis (“timeframe”). Kad šį metodą būtų galima pritaikyti, būtina turėti mažesnio laikotarpio duomenis. Veiksmai atliekami ne po žvakės užsidarymo, o po mažesnio laikotarpio žvakės užsidarymo.
- **Modeliuojant per kiekvieną tiką (“tick”).**
Šis metodas naudoja visus galimus mažesnius laikotarpius. Tai pats tiksliausias metodas, tačiau reikalaujantis daugiausiai duomenų ir kompiuterio resursų. Jei mažesnių laikotarpių duomenų nėra, tai naudoti šį metodą nerekomenduojama, nes rezultatai nebus tikslūs [17].

Optimalaus sistemos parametrų rinkinio radimas

Optimizavimas, tai tos pačios ekspertinės sistemos su skirtingais įėjimo parametrais testavimas, panaudojant vienodus istorinius duomenis siekiant didžiausio sistemos efektyvumo. Norint įjungti optimizavimo funkciją, nustatomi visi parametrai kaip ir testavimo funkcijoje ir papildomai reikia pagrindiniame testavimo lange aktyvuoti „optimization“ lauką. Galimi tokie optimizavimo kriterijai:

- Didžiausia balanso reikšmė („Balance“)
- Didžiausias pelno faktorius („Profit factor“)
- Didžiausia išmokėjimo reikšmė („Payoff“)
- Mažiausias praradimas („Drawdown“)
- Mažiausias praradimas procentine išraiška („Relative drawdown“)
- Kitas, vartotojo pasirinktas kuriant programą.

Optimizavimas gali būti atliekamas dviem būdais: optimizuojamus parametrus keičiant pasirinktu žingsniu ir pereinant per visus galimus parametrų rinkinius arba panaudojant genetinį algoritmą. Pirmasis būdas gali būti panaudojamas ten, kur optimizuojamas vienas ar du parametrai siaurose ribose. Jei optimizuojami daugiau nei du parametrai, tai iteracijų

skaičius būna didelis ir skaičiavimai ilgai užtrunka. Daug mažiau skaičiavimo laiko reikalauja optimizavimas panaudojant genetinį algoritmą.

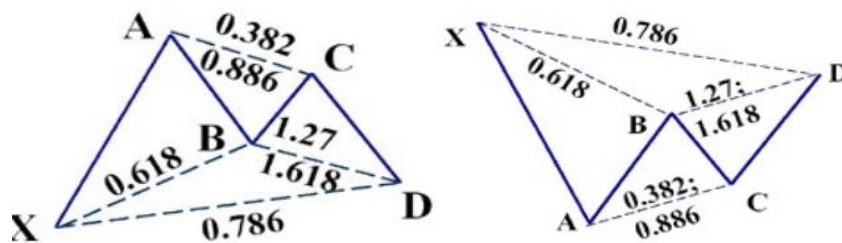
2.2. Strategijos sudarymas

Šiame darbe sandorio įėjimo ir išėjimo taisyklės yra paremtos harmoninių kainų konfigūracijų susidarymo principais.

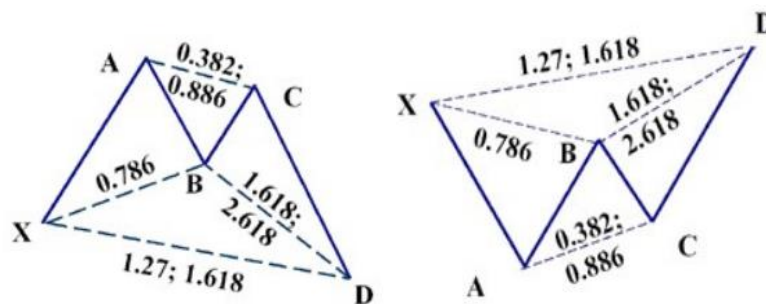
2.2.1. Harmoninėmis kainų konfigūracijomis paremta strategija

Prekiavimas remiantis harmoninėmis konfigūracijomis yra metodas, paremtas tam tikrų geometrinių struktūrų susidarymu, kurių kainos krypties keitimo taškai turi atitikti Fibonači lygius. Pirmiausiai yra ieškoma lokalių minimumų ir maksimumų, o po to tikrinama ar minimumai ir maksimumai yra reikiamuose fibonači lygiuose.

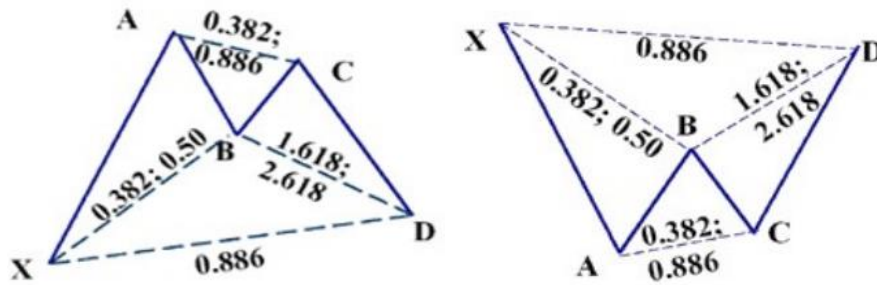
Harmoninių kainų konfigūracijų yra gana nemažai. Pačios populiariausios yra Gartley, drugio, šikšnosparnio ir krabo konfigūracijos.



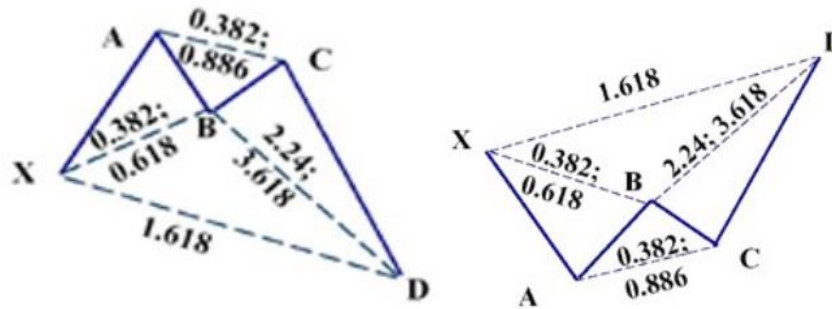
2.5 pav. Bulių ir lokių rinkos gartley konfigūracija[18]



2.6 pav. Bulių ir lokių rinkos drugio konfigūracija[18]



2.7 pav. Bulių ir lokių rinkos šikšnosparnio konfigūracija[18]



2.8 pav. Bulių ir lokių rinkos krabo konfigūracija[18]

Visos šios konfigūracijos yra panašios, skiriasi tik Fibonači lygiai, kuriuose įvyksta krypties pasikeitimas einant iš vieno taško į kitą. Turint pirmąjį minimumą ir maksimumą galima pasiskaičiuoti sekančius reikiamus taškus pagal formules:

$$UR = H - ((H - L) * P)$$

$$DR = L + ((H - L) * P)$$

Čia P-procentai(Fibonači skaičius);

H-Aukštesnė reikšmė;

L-Žemesnė reikšmė;

UR- tendencijos aukštyn keitimasis (Uptrend retracement);

DR- tendencijos žemyn keitimasis (Downtrend retracement).

Panaudojus šias formules, gauname konfigūracijų leistinas ribas (2.1.1, 2.1.2 lentelės), kuriose turėtų susidaryti B,C ir D taškai, kad konfigūracija būtų patvirtinta. Nors vienam minimumui arba maksimumui susidarius už leistinų ribų, konfigūracija laikoma nepasitvirtinusia.

2.1 lentelė. Bulių konfiguracijų taškų susidarymo sąlygos

Taškas	Žemutinė riba	Aukštesnė riba
Gartley		
B	$A - ((A - X) * 0,786)$	$A - ((A - X) * 0,618)$
C	$B + ((A - B) * 0,382)$	$B + ((A - B) * 0,886)$
D	$C - ((C - B) * 1,618)$	$C - ((C - B) * 1,27)$
A-D	$A - ((A - X) * 0,5)$	$A - ((A - X) * 0,786)$
Drugio		
B	$A - ((A - X) * 0,786)$	$A - ((A - X) * 0,618)$
C	$B + ((A - B) * 0,382)$	$B + ((A - B) * 0,886)$
D	$C - ((C - B) * 2,618)$	$C - ((C - B) * 1,618)$
A-D	$A - ((A - X) * 1,618)$	$A - ((A - X) * 1,27)$
Šikšnosparnio		
B	$A - ((A - X) * 0,5)$	$A - ((A - X) * 0,382)$
C	$B + ((A - B) * 0,382)$	$B + ((A - B) * 0,886)$
D	$C - ((C - B) * 2,618)$	$C - ((C - B) * 1,618)$
A-D	$A - ((A - X) * 0,886)$	$A - ((A - X) * 0,786)$
Krabo		
B	$A - ((A - X) * 0,618)$	$A - ((A - X) * 0,382)$
C	$B + ((A - B) * 0,382)$	$B + ((A - B) * 0,886)$
D	$C - ((C - B) * 3,618)$	$C - ((C - B) * 2,224)$
A-D	$A - ((A - X) * 1,618)$	$A - ((A - X) * 1,5)$

2.2 lentelė. Lokių konfiguracijų taškų susidarymo leistinos ribos

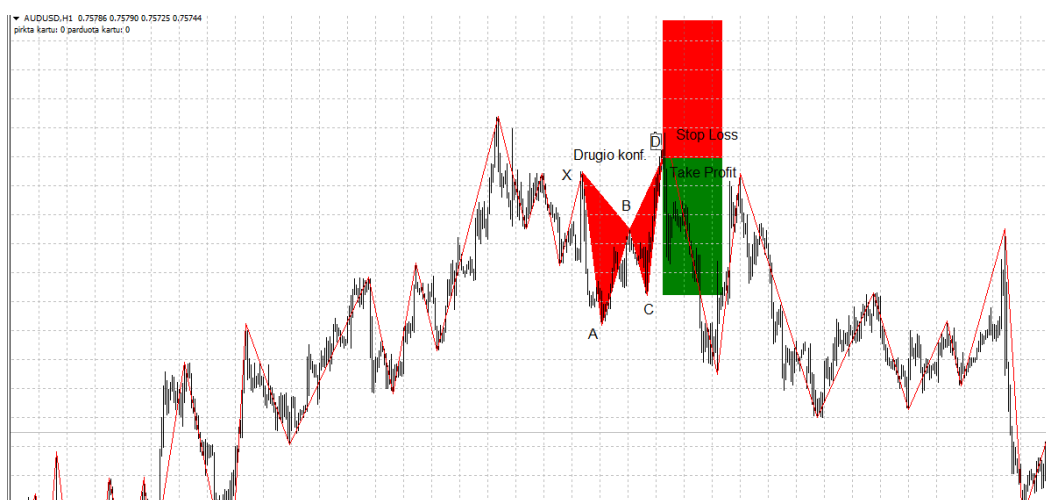
Taškas	Žemutinė riba	Aukštesnė riba
Gartley		
B	$A + ((X - A) * 0,618)$	$A + ((X - A) * 0,786)$
C	$B - ((B - A) * 0,886)$	$B - ((B - A) * 0,382)$
D	$C + ((B - C) * 1,27)$	$C + ((B - C) * 1,618)$
D	$A + ((X - A) * 0,5)$	$A + ((X - A) * 0,786)$
Drugio		
B	$A + ((X - A) * 0,618)$	$A + ((X - A) * 0,786)$

C	$B - ((B - A) * 0,886)$	$B - ((B - A) * 0,382)$
D	$C + ((B - C) * 1,618)$	$C + ((B - C) * 2,618)$
D	$A + ((X - A) * 1,27)$	$A + ((X - A) * 1,618)$
Šikšnosparnio		
B	$A + ((X - A) * 0,382)$	$A + ((X - A) * 0,5)$
C	$B - ((B - A) * 0,886)$	$B - ((B - A) * 0,382)$
D	$C + ((B - C) * 1,618)$	$C + ((B - C) * 2,618)$
D	$A + ((X - A) * 0,786)$	$A + ((X - A) * 0,886)$
Krabo		
B	$A + ((X - A) * 0,382)$	$A + ((X - A) * 0,618)$
C	$B - ((B - A) * 0,886)$	$B - ((B - A) * 0,382)$
D	$C + ((B - C) * 3,618)$	$C + ((B - C) * 2,224)$
D	$A + ((X - A) * 1,5)$	$A + ((X - A) * 1,618)$

Bulių konfigūracijos atveju, pasitvirtinus D taškui yra perkama valiutų pora. Lokių konfigūracijos atveju valiutų pora parduodama.

Toleruojamo nuostolio riba (stop limit) ir pelno riba (take profit)

Kad strategija būtų sėkminga, būtina nusistatyti toleruojamo nuostolio ir pelno ribas. Kainai pasiekus bet kurią iš šių ribų, pozicija yra uždaroma. Tik vienu atveju pozicija uždaroma pelningai (kai pasiekiami “take profit” riba), o kitu – nuostolingai.



2.9 pav. Drugio lokių konfigūracija su “stop limit” ir “take profit” ribomis



2.10 pav. Gartley bulių konfiguracija su „stop limit“ ir „take profit“ ribomis

Prekiautojai naudoja kelis „take profit“ ir „stop limit“ nusistatymo būdus. Vienas iš jų, „take profit“ ribą nusistatyti ties tašku C, o „stop loss“ – keliais pipais (priklausomai nuo konfiguracijos dydžio) žemiau nei įėjimo į rinką taškas. Tokiu atveju didžioji dalis transakcijų būna nuostolingos, tačiau tas nuostolis lyginant su galimu pelnu, yra labai nedidelis. Šiuo atveju buvo naudojamas kitas būdas. „Take profit“ ir „stop loss“ ribos nustatomos atstumui lygiam tarp taškų C ir D. Tokiu būdu kur kas didesnė transakcijų dalis būna sėkminga. Tačiau galimi praradimai didesni. „Take profit“ ir „stop loss“ ribos nebuvo optimizuojamos. Laikoma, kad sėkmingam strategijos veikimui jos turi būti lygios atstumui tarp taškų C ir D.

2.2.2. Programavimo eiga ir programos algoritmas

Ekstremumų paieška

Pirmiausia norint rasti bet kokią harmoninę konfiguraciją reikia rasti minimumus ir maksimumus kainos grafike. Metatrader4 savo bibliotekoje jau turi įdiegtą ekstremumų paieškos algoritmą, kurį galima rasti indikatorių skiltyje. Šis indikatorius yra vadinamas „ZigZag“. Tokį pavadinimą gavo dėl savo formos (2.10 pav. raudona linija).

Indikatorius yra sudarytas iš trijų buferių – ExtLowBuffer, ExtHighBuffer, ExtZigZagBuffer. Pirmajame fiksuojami visi kainos istorijoje buvę minimumai, antrajame – maksimumai, o trečiajame minimumai ir maksimumai kartu. Kuriant automatinę prekiavimo sistemą iš ZigZag indikatoriaus buvo importuotas ir naudojamas ExtZigZagBuffer buferis. Šis buferis yra matrica, kurios ne nuliui lygūs nariai yra ekstremumai.

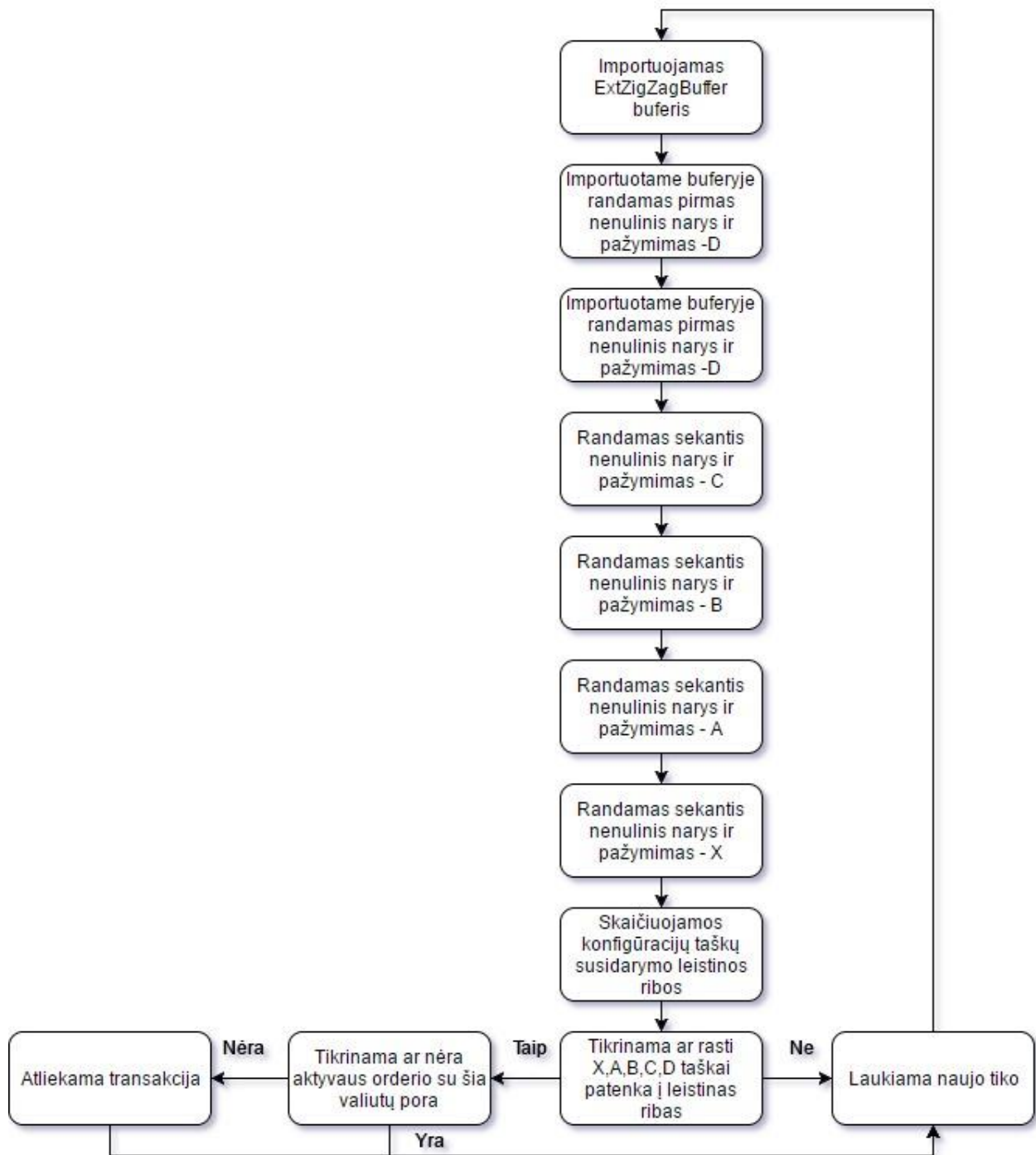
Šis indikatorius turi tris įėjimo parametrus: InpDepth, InpDeviation ir InpBackstep. InpDepth parametras nurodo minimalų žvakių skaičių, kuriame negali būti kito ekstremumo didesnio arba mažesnio už InpDeviation skaičių. InpBackstep parametras nurodo minimalų žvakių skaičių tarp minimumo ir maksimumo. Keičiant šių parametrų derinį galima nufiltruoti triukšmus norimu lygiu. Taip pat nustatyti optimalūs ZigZag indikatoriaus įėjimai, kad būtų maksimizuotas pelnas.

Programos algoritmas

Automatinėje prekyavimo sistemoje yra naudojamas jau aptartas ZigZag indikatorius. iCustom funkcijos dėka, galima pasiimti ExtZigZagBuffer matricą, kurioje yra minimumų ir maksimumų taškai. Nulinis matricos narys yra pats naujausias narys. Kai jau turime ekstremumų matricą, tai joje ieškome penkių ne nuliui lygių narių ir juos pažymime D,C,B,A,X. Tuomet paskaičiuojame pagal 2.1.1 ir 2.1.2 lenteles leistinas taškų susidarymo ribas ir tikriname ar rasti D,C,B,A,X taškai tenkina šias ribas. Jei tenkina ir nėra aktyvaus orderio, tai atliekamas pirkimas/pardavimas. Jei netenkina – ciklas kartojamas iš naujo.

Ciklas kartojamas tuomet, kai gaunamas naujas signalas iš serverio (naujas tikas). Gavus naują signalą, ExtZigZagBuffer matrica sukuriama iš naujo ir aprašytas ciklas (pav. 2.2) kartojamas nuo pradžių. ZigZag indikatoriaus kodas pateiktas 1 priede, o automatinės prekyavimo sistemos kodas – 2 priede. Svarbiausios panaudotos funkcijos pateiktos 2.2 lentelėje.

Jei algoritme nebūtų apibrėžta kada pasitraukti iš rinkos, tai tokia sistema būtų labai pavojinga. Jos nebegalėtų vadinti automatine, nes prekiautojas turėtų sekti ir nuspręsti, kada laikas pasitraukti. Taip pat atsirastų žmogiškos klaidos galimybė. Apibrėžus „take profit“ ir „stop loss“ ribų nustatymo taisykles, sistema tampa pilnai automatine. Išvengiama prekiautojo klaidų ir emocijomis paremtų sprendimų.



2.11 pav. programos algoritmas

2.2 lentelė. Panaudotos funkcijos

Nr.	Funkcija	Įėjimo kintamieji	Aprašymas	Pastabos
1	Print	Norimas pranešimas	Parašo pranešimą	Naudojama tikrinant ar

			Expert Advisor registre.	programa veikia korektiškai ir kintamųjų stebėjimui
2	Comment	Norimas pranešimas	Parašo pranešimą lango kairiame viršutiniame krašte	Naudojama kintamųjų stebėjimui
3	Bars	Nėra	Žvakių skaičius grafike	
4	RefreshRates	Nėra	Tikrina ar duomenys atnaujinti	Jeigu duomenys teisingi, funkcijos išėjimas TRUE
5	iCustom	Simbolis, laikotarpis, indikatoriaus pavadinimas, indikatoriaus įėjimai, indikatoriaus buferio numeris	Grąžina betkurio indikatoriaus iš bibliotekos reikiamą reikšmę	Naudojama ZigZag indikatoriaus buferio gavimui
6	OrdersTotal	Nėra	Grąžina skaičių aktyvių orderių	
7	AccountFreeMargin	Nėra	Grąžina laisvą pinigų sumą	
8	OrderSend	Simbolis, operacijos tipas, apimtis, kaina, toleruojamas praslydimas, stoploss riba, take profit riba, komentaras, magiškas skaičius, orderio nebegaliojimo data, žymėjimo spalva	Pagrindinė funkcija naudojama orderio atidarymui	

9	OrderSelect	Orderio numeris	Pažymi orderį tolesniam naudojimui	
10	OrderOpenPrice	Nėra	Grąžina pasirinkto orderio atidarymo kainą	
11	GetLastError	Nėra	Grąžina paskutinės klaidos numerį	Pagal numerį galima gauti klaidos aprašymą
12	ObjectCreate	Objekto vardas, tipas, lango nr., objekto kampų koordinatės.	Sukuria grafinį objektą	Naudojama konfigūracijos trikampių kūrimui
13	ObjectSet	Objekto vardas, objekto savybės tipas, savybės nustatymas	Objekto savybių nustatymas	Naudojama trikampio spalvų parinkimui

3. TIRIAMOJI DALIS

Sukūrus sistemą aprašytą antrame skyriuje, buvo atliekami testavimai ir optimizavimas. Tyrimai buvo atliekami su keturių valandų (H4), vienos valandos (H1), 30 minučių (M30) ir 5 minučių (M5) grafikais. Dėl didelių skaičiavimų apimčių tyrimai atlikti tik su euro/svaro pora. Šiais tyrimais siekiama nustatyti ar sistema gali generuoti pelną ilgame laikotarpyje (8 metų laikotarpis) ir kaip dažnai ir kokie pakeitimai sistemoje yra būtini, siekiant optimalaus veikimo. Pasirinktas optimizavimo kriterijus (galimi optimizavimo kriterijai aprašyti 2.1.3 skyriuje) - maksimalus pelnas. Optimizuoti trys ekstremumų paieškos parametrai (InpDepth, InpDeviation ir InpBackstep). Tikslo funkcija:

$$J = \text{Pelnas}(\text{InpDepth}, \text{InpDeviation}, \text{InpBackstep}) \rightarrow \max \quad (3.1 \text{ formulė})$$

Geriausiam rastam variantui optimizuotos ir „Take profit“ ir „Stop loss“ ribos.

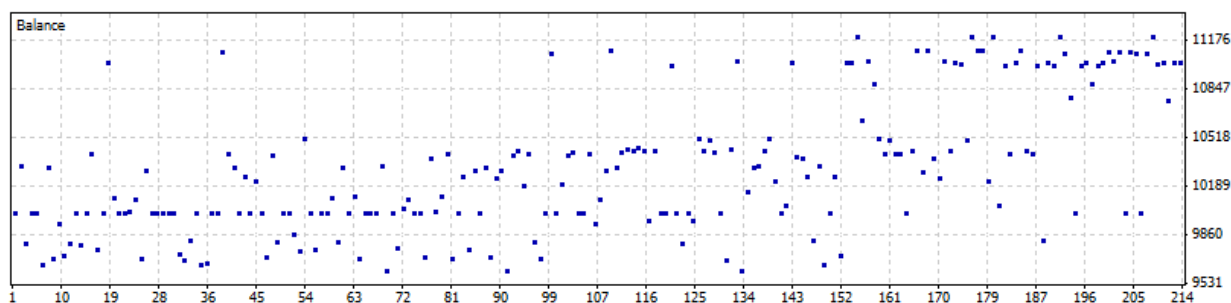
Visų transakcijų dydis buvo 0.1 loto. Tai reiškia, kad norint prekiauti nenaudojant svertų, reikia turėti 10000 dolerių. Tačiau panaudojus svertą, galima to paties dydžio transakciją atlikti su kur kas mažesniu depozitu. Kai kurie brokeriai leidžia naudotis 1:300 ir didesniais svertais. Jei būtų naudojamas 1:100 svertas, tai norint atlikti 0.1 loto transakciją, sąskaitoje užtektų turėti 100 dolerių. Tačiau svertų naudojimas padidina riziką. Šiame darbe visi pirkimai ir pardavimai buvo atliekami be svertų, o jų dydis buvo fiksuotas – 0.1 loto. Tokiu atveju 1 pipo pasikeitimas generuoja 1,3 dolerio pasikeitimą. Taigi, jei konfigūracijos take profit riba yra per 100 pipų nutolusi nuo transakcijos atlikimo kainos, tai sėkmės atveju bus uždirbta 130 dolerių.

3.1. Sistema, apmokyta su 9 metų duomenimis

Turimus istorinius duomenis, nuo 2000 metų iki dabar, padaliname į dvi dalis: 2000-2009 ir 2009-2017. Pirmoji dalis buvo skirta sistemos apmokymui. Tai reiškia, kad buvo atliktas optimizavimas su 2000-2009 ir pasirinktas didžiausią pelną nešęs parametų rinkinys. Su šiuo parametų rinkiniu buvo ištirta, kokį pelną būtų atnešusi sistema, jei būtų dirbusi 2009-2017 laikotarpiu.

3.1.1. Keturių valandų grafikas (H4)

2000-2009.06 optimizavimo rezultatai:



3.1 pav. H4 optimizavimo rezultatai 2000.01-2009.06 laikotarpiu

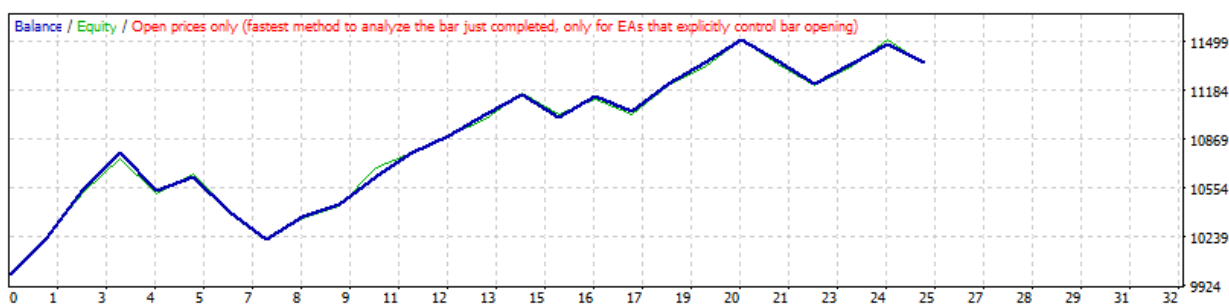
3.1 lentelė. Geriausi optimizavimo rezultatai su H4 grafiku

Iteracija	Pelnas, \$	Transakcijų sk.	Pelno faktorius	Nuosmukis \$	Nuosmukis %
155	1192.92	69	1.63	415.36	3.82
209	1187.92	69	1.63	413.65	3.80
192	1187.67	69	1.63	413.49	3.80
180	1187.64	69	1.63	413.47	3.80

176	1187.52	69	1.63	413.48	3.80
110	1103.65	73	1.52	565.44	5.23
185	1098.72	73	1.52	562.93	5.20
178	1098.58	73	1.52	562.92	5.20
177	1098.56	73	1.52	562.91	5.20
168	1098.18	73	1.52	562.70	5.20

2000.01 – 2009.01 laikotarpio geriausias parametru rinkinys uždirbo 1192.92 dolerius. Per šį laikotarpį buvo padarytos 69 transakcijos. Testuojame kaip ši Sistema būtų dirbusi 2009.01 – 2017.05 laikotarpiu (3.2 paveikslas).

Testuojama geriausia Sistema:



3.2 pav. Darbas su H4 grafiku ir geriausiu parametru rinkiniu (2009.01-2017.05)

3.2 lentelė. Testavimo rezultatai su H4 grafiku

Parametrai		InpDepth=11; InpDeviation=2; InpBackstep=8; Lots=0.1; sl=1; tp=1;			
Pradinis depozitas	10000.00			Spreadas	10
Bendras pelnas/nuostolis	1365.14	Bruto pelnas	2668.15	Bruto nuostolis	-1303.01
Pelno faktorius	2.05				
Maksimalus nuosmukis	653.37 (6.04%)	Realiatyvus nuosmukis %(\$)	6.04% (653.37)		
Transakcijų skaičius	25	Pardavimo pozicijos (laimėtos %)	9 (88.89%)	Pirkimo pozicijos (laimėtos %)	16 (56.25%)
		Pelningų transakcijų(% iš visų transakcijų)	17 (68.00%)	Nuostolingų transakcijų (% iš visų transakcijų)	8 (32.00%)
Didžiausia		Pelninga transakcija	318.33	Nuostolingos transakcijos	-248.41
Vidutinė		Pelninga transakcija	156.95	Nuostolingos transakcijos	-162.88
Didžiausias		Pelningų transakcijų sk. Iš eilės(pelnas)	7 (935.62)	Nuostolingų transakcijų sk. Iš eilės(pelnas)	2 (-406.00)
Didžiausias		Nuolatinis pelnas (Transakcijų sk.)	935.62 (7)	Nuolatinis nuostolis (Transakcijų sk.)	-406.00 (2)

Vidutinis	Pelningų transakcijų sk. Iš eilės	3	Nuostolingų transakcijų sk. Iš eilės	1
-----------	-----------------------------------	---	--------------------------------------	---

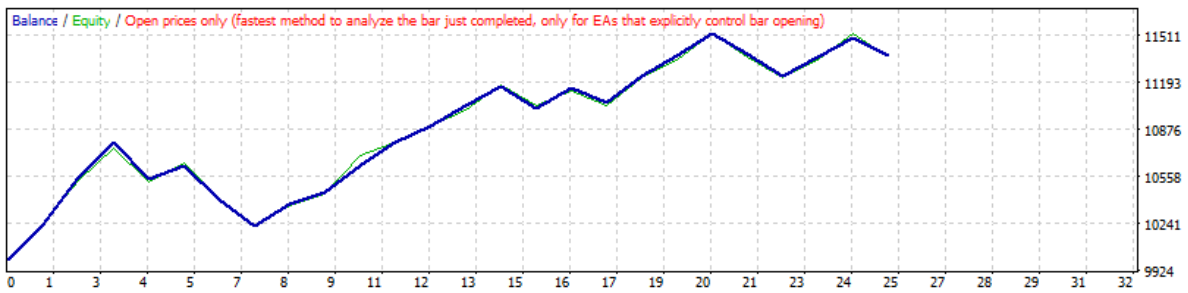
Dirbant su keturių valandų grafiku per beveik 8 metų laikotarpį su šiuo parametų rinkiniu buvo pasiektas 1365.14 dolerių pelnas (13.65% pelningumas). Sistema padarė 25 transakcijas. 68% transakcijų buvo pelningos.

3.1.2. Keturių valandų grafikas su papildoma įėjimo į rinką sąlyga

Siekiant išvengti tokių situacijų (3.3 paveikslas) kai konfigūracija yra susidariusi, tačiau kaina krypties dar nekeičia, buvo įvesta papildoma sąlyga. Į rinką neįeinama tol, kol kaina nepradėjo judėti reikiama kryptimi. Tai reiškia, kad neužtenka vient tik konfigūracijos susidarymo, taip pat reikia, kad dviejų iš eilės žvakių atsidarymo kaina būtų pakeitusi kryptį lyginant su prieš tai buvusiomis (bulių konfigūracijos atveju, dviejų iš eilės žvakių atsidarymo kainos turi būti didesnės už prieš tai buvusią, o lokių konfigūracijos atveju – mažesnės). Sistemoje padarius tokius pakeitimus, pelningų transakcijų skaičius turėtų padidėti. Tačiau į rinką įeinama pavėluotai, dėl to gali sumažėti teisingų transakcijų pelningumas.



3.3 pav. Atvejis, kai konfigūracija susidariusi, tačiau kaina juda netinkama kryptimi. Sistema testuota su tuo pačiu parametų rinkiniu kaip ir 3.1.1 skyriuje (2009.01-2017.06 laikorapiu):



3.4 pav. Darbas su H4 grafiku ir geriausiu parametų rinkiniu, įvedus papildomą sąlygą (2009.01-2017.05 laikotarpiu)

3.2 lentelė. Testavimo rezultatai su H4 grafiku

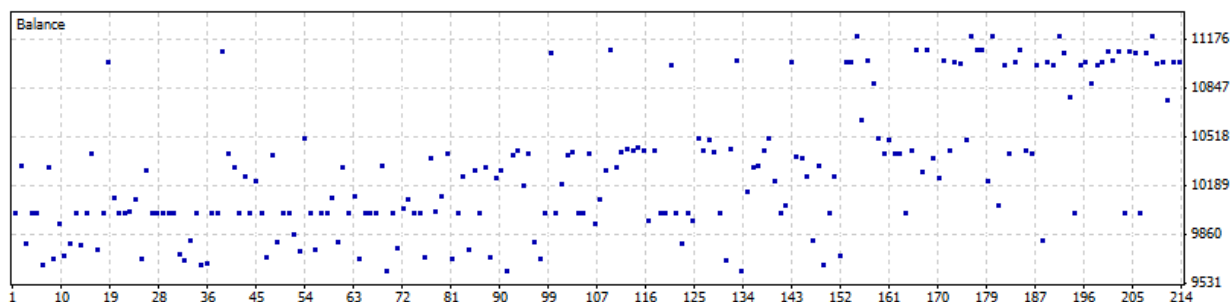
Parametrai		InpDepth=11; InpDeviation=2; InpBackstep=8; Lots=0.1; sl=1; tp=1;			
Pradinis depozitas	10000.00			Spreadas	10
Bendras pelnas/nuostolis	1375.74	Bruto pelnas	2688.29	Bruto nuostolis	-1312.55
Pelno faktorius	2.05				
Maksimalus nuosmukis	658.02 (6.08%)	Realiatyvus nuosmukis %(\$)	6.08% (658.02)		
Transakcijų skaičius	25	Pardavimo pozicijos (laimėtos %)	9 (88.89%)	Pirkimo pozicijos (laimėtos %)	16 (56.25%)
		Pelningų transakcijų(% iš visų transakcijų)	17 (68.00%)	Nuostolingų transakcijų (% iš visų transakcijų)	8 (32.00%)
Didžiausia		Pelninga transakcija	320.85	Nuostolingos transakcijos	-250.21
Vidutinė		Pelninga transakcija	158.13	Nuostolingos transakcijos	-164.07
Didžiausias		Pelningų transakcijų sk. Iš eilės(pelnas)	7 (942.66)	Nuostolingų transakcijų sk. Iš eilės(pelnas)	2 (-408.93)
Didžiausias		Nuolatinis pelnas (Transakcijų sk.)	942.66 (7)	Nuolatinis nuostolis (Transakcijų sk.)	-408.93 (2)
Vidutinis		Pelningų transakcijų sk. Iš eilės	3	Nuostolingų transakcijų sk. Iš eilės	1

Dirbant su keturių valandų grafiku per beveik 8 metų laikotarpį su efektyviausiu parametų rinkiniu buvo pasiektas 1375.74 dolerių pelnas (13.76% pelningumas). Sistema padarė 25 transakcijas. 68% transakcijų buvo pelningos.

Lyginant su 3.1.1 skyriuje tirta sistema, ši padarė tiek pat ir tas pačias transakcijas, bet įėjimais į rinką buvo tikslesni, todėl pasiektas šiek tiek didesnis pelningumas. Tačiau didžiausias patirtas nuosmūkis buvo šiek tiek didesnis.

3.1.3. Valandinis grafikas (H1)

2000.01-2009.06 optimizavimo rezultatai:

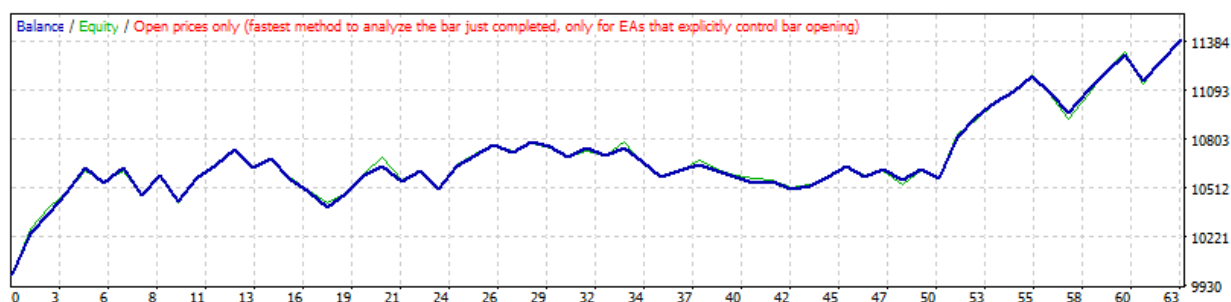


3.4 pav. H1 optimizavimo rezultatai 2000.01-2009.06 laikotarpiu

3.3 lentelė. Geriausi optimizavimo rezultatai su H1 grafiku

Iteracija	Pelnas, \$	Transakcijų sk.	Pelno faktorius	Nuosmukis \$	Nuosmukis %
155	1192.92	69	1.63	415.36	3.82
209	1187.92	69	1.63	413.65	3.80
192	1187.67	69	1.63	413.49	3.80
180	1187.64	69	1.63	413.47	3.80
176	1187.52	69	1.63	413.48	3.80
110	1103.65	73	1.52	565.44	5.23
185	1098.72	73	1.52	562.93	5.20
178	1098.58	73	1.52	562.92	5.20
177	1098.56	73	1.52	562.91	5.20
168	1098.18	73	1.52	562.70	5.20

Testuojama geriausia sistema:



3.5 pav. H1 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu

2000.01 – 2009.06 laikotarpio geriausias parametrų rinkinys pasiekė 1192.92 dolerių pelną (11.92% pelningumas). Per šį laikotarpį buvo padarytos 69 transakcijos. Testuojame kaip ši Sistema būtų dirbusi 2009.01 – 2017.05 laikotarpiu (20 paveikslas).

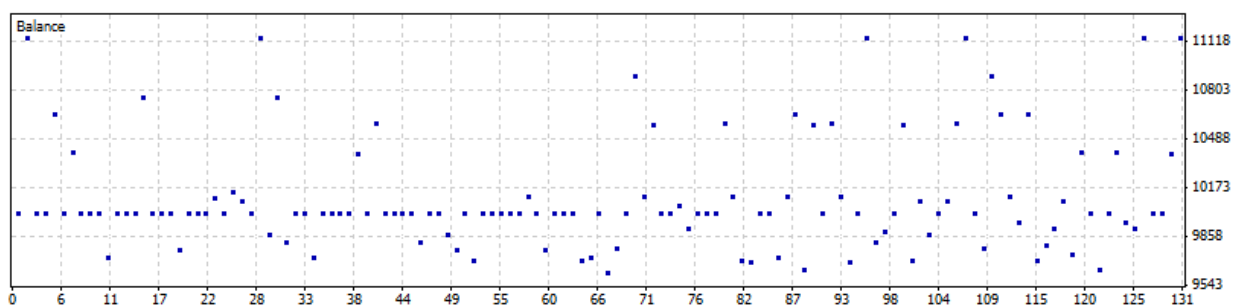
3.4 lentelė. Testavimo rezultatai su H1 grafiku

Parametrai		InpDepth=17; InpDeviation=10; InpBackstep=12; Lots=0.1; sl=1; tp=1;			
Pradinis depozitas	10000.00			Spreadas	10
Bendras pelnas/nuostolis	1398.92	Bruto pelnas	3463.53	Bruto nuostolis	-2064.61
Pelno faktorius	1.68				
Maksimalus nuosmukis		451.20 (4.18%)	Realiatyvus nuosmukis %(\$)	4.18% (451.20)	
Transakcijų skaičius	63	Pardavimo pozicijos (laimėtos %)	31 (67.74%)	Pirkimo pozicijos (laimėtos %)	32 (53.13%)
		Pelningų transakcijų(% iš visų transakcijų)	38 (60.32%)	Nuostolingų transakcijų (% iš visų transakcijų)	25 (39.68%)
Didžiausia	Pelninga transakcija	244.78	Nuostolingos transakcijos	-162.32	
Vidutinė	Pelninga transakcija	91.15	Nuostolingos transakcijos	-82.58	
Didžiausias	Pelningų transakcijų sk. Iš eilės(pelnas)	5 (608.02)	Nuostolingų transakcijų sk. Iš eilės(pelnas)	3 (-291.81)	
Didžiausias	Nuolatinis pelnas (Transakcijų sk.)	637.75 (4)	Nuolatinis nuostolis (Transakcijų sk.)	-291.81 (3)	
Vidutinis	Pelningų transakcijų sk. Iš eilės	2	Nuostolingų transakcijų sk. Iš eilės	1	

Dirbant su valandiniu grafiku per beveik 8 metų laikotarpį su šiuo parametų rinkiniu buvo pasiektas 1398.92 dolerių pelnas (13.98 %). Sistema padarė 63 transakcijas. 60,32% transakcijų buvo pelningos.

3.1.4. Trisdešimties minučių grafikas (M30)

2000-2009 optimizavimo rezultatai:

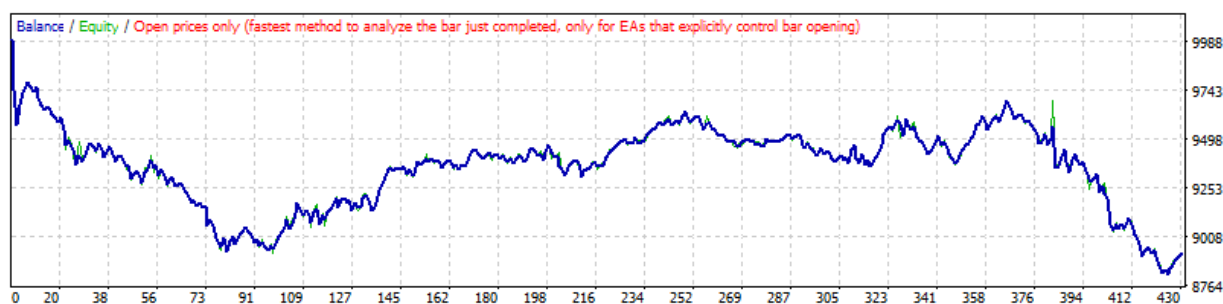


3.6 pav. M30 sistemos optimizavimo rezultatai rezultatai 2000.01-2009.05 laikotarpiu

3.5 lentelė. Optimizavimo rezultatai su M30 grafiku

Iteracija	Pelnas, \$	Transakcijų sk.	Pelno faktorius	Nuosmukis \$	Nuosmukis %
96	1133.80	424	1.23	487.10	4.52
107	1133.72	424	1.23	487.12	4.52
131	1133.66	424	1.23	486.98	4.51
127	1133.58	424	1.23	486.93	4.51
28	1133.19	424	1.23	487.28	4.52
2	1133.19	424	1.23	487.28	4.52
70	889.27	341	1.18	474.35	4.49
110	889.15	341	1.18	474.22	4.49
30	750.45	407	1.15	515.49	5.03
15	750.45	407	1.15	515.49	5.03

Testuojama geriausia Sistema:



3.7 pav. M30 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu

2000 – 2009 laikotarpio geriausias parametrų rinkinys atnešė 1069.71 dolerius nuostolio (-10.69% pelningumas). Per šį laikotarpį buvo padarytos 429 transakcijos. Testuojame kaip ši Sistema būtų dirbusi 2009 – 2017.05 laikotarpiu (20 paveikslas).

3.6 lentelė. Testavimo rezultatai su M30 grafiku

Parameters		InpDepth=5; InpDeviation=10; InpBackstep=1; Lots=0.1; sl=1; tp=1;			
Pradinis depozitas	10000.00			Spreadas	10
Bendras pelnas/nuostolis	-1069.71	Gross profit	5336.91	Bruto nuostolis	-6406.62
Pelno faktorius	0.83				
Maksimalus nuosmukis	1179.15 (11.79%)	Realiyvus nuosmukis %(\$)	11.79% (1179.15)		
Transakcijų skaičius	429	Pardavimo pozicijos (laimėtos %)	219 (48.40%)	Pirkimo pozicijos (laimėtos %)	210 (63.33%)

	Pelningų transakcijų(% iš visų transakcijų)	239 (55.71%)	Nuostolingų transakcijų (% iš visų transakcijų)	190 (44.29%)
Didžiausia	Pelninga transakcija	99.25	Nuostolingos transakcijos	-255.18
Vidutinė	Pelninga transakcija	22.33	Nuostolingos transakcijos	-33.72
Didžiausias	Pelningų transakcijų sk. Iš eilės(pelnas)	10 (239.10)	Nuostolingų transakcijų sk. Iš eilės(pelnas)	5 (-187.54)
Didžiausias	Nuolatinis pelnas (Transakcijų sk.)	239.10 (10)	Nuolatinis nuostolis (Transakcijų sk.)	-430.48 (2)
Vidutinis	Pelningų transakcijų sk. Iš eilės	2	Nuostolingų transakcijų sk. Iš eilės	2

Dirbant trisdešimt minučių grafiku su geriausiu parametų rinkiniu per beveik 8 metų laikotarpį buvo patirtas -1069,71 dolerių nuostolis (-10.69% pelningumas). Sistema padarė 429 transakcijas. 48.40% transakcijų buvo pelningos

3.2. Sistema, kurios parametrai po tam tikro laikotarpio perrenkami iš naujo

Buvo tiriama apibrėžto veikimo laiko sistema, kuri apmokoma su istoriniais duomenimis. Skirtumas nuo prieš tai buvusiame skyriuje tirtos sistemos yra toks, kad šios sistemos parametrai yra keičiami apibrėžtam laiko tarpui. Buvo tirta kaip sistema veikia, jei optimizavimas atliekamas su 5, 4, 3, 2 metų duomenimis. Taip pat ištirta po kiek laiko geriausia keisti sistemos parametrus. Kadangi skaičiavimų apimtys didelės ir reikalauja daug laiko, tai sistema buvo tiriama tik su keturių valandų grafiku. Šiuo tyrimu buvo siekiama sužinoti, kokiems optimizavimo ir darbo laikotarpiams esant, galima pasiekti geriausius rezultatus.

3.2.1. 2 metų laikotarpio optimizavimas

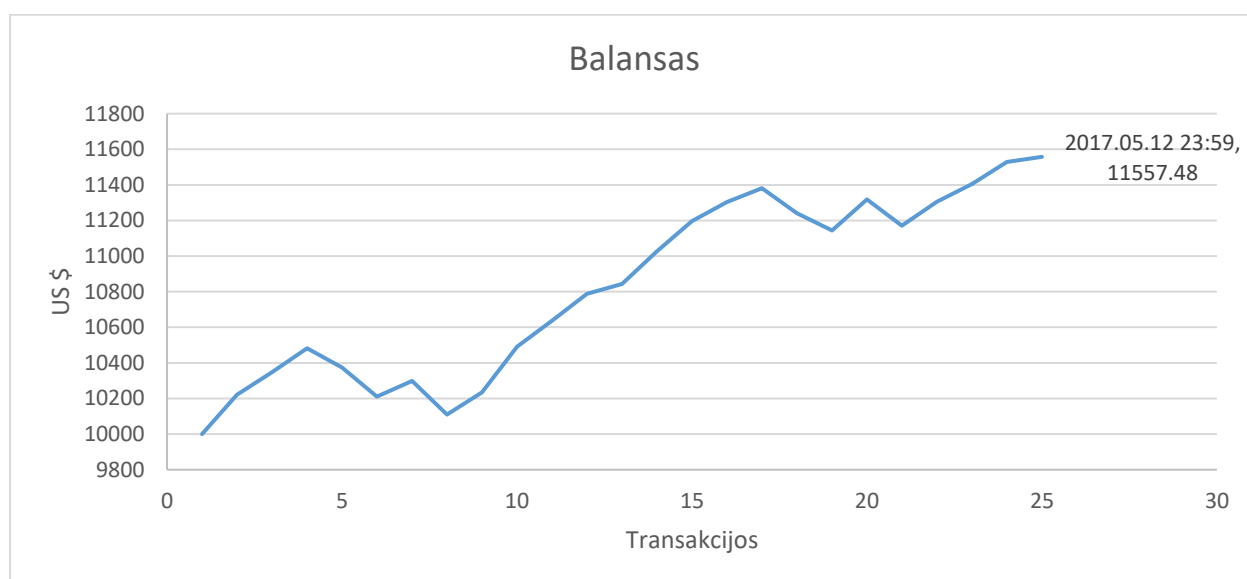
Sistemos apmokymui skiriami 2 metai. Buvo tiriama, koks darbo laikotarpis (1,2,3,4 ar 5 metai) atneša geriausius rezultatus.

Sistemos darbui skiriami 1 metai

Ieškomas optimalus parametų rinkinys dviejų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2007.01.

3.7 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 2 metai, darbui – 1 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2007-2009	2009-2010	8	234.52	5	9	3
2008-2010	2010-2011	1	255.98	12	9	11
2009-2011	2011-2012	1	145.68	12	6	11
2010-2012	2012-2013	2	207.63	18	4	12
2011-2013	2013-2014	3	459.98	18	10	8
2012-2014	2014-2015	2	-63.37	18	10	4
2013-2015	2015-2016	2	78.06	13	4	6
2014-2016	2016-2017	2	-13.28	11	10	7
2015-2017	2017-2017.05	3	252.28	5	2	2
	Viso:	24	1557.48			



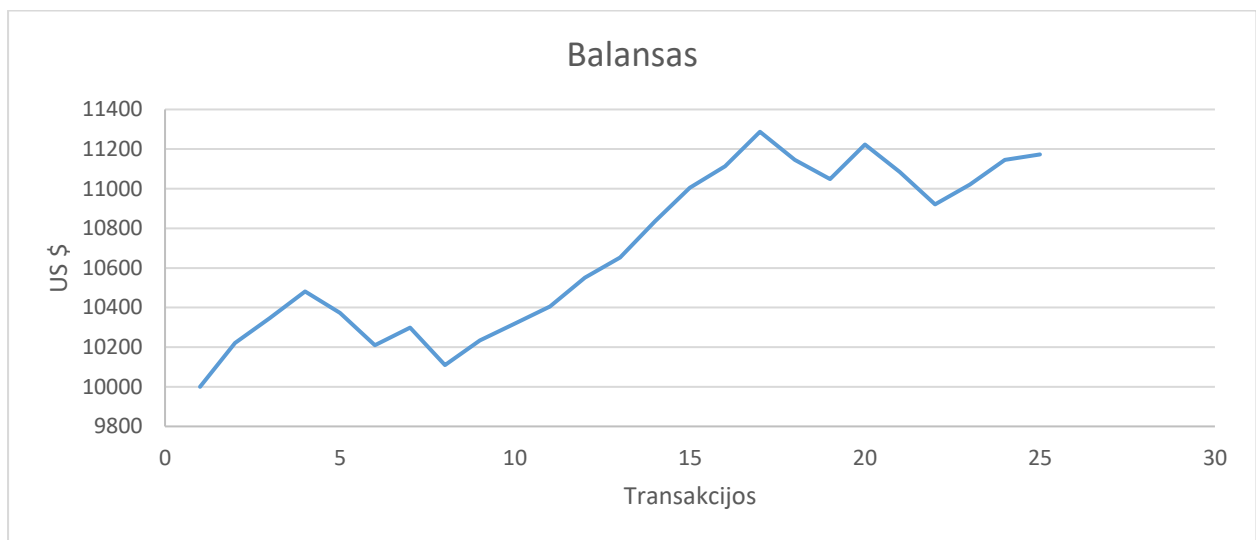
3.8 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 2 metai, o darbui – 1 metai.

Sistemos darbui skiriami 2 metai

Ieškomas optimalus parametų rinkinys dviejų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2007.01.

3.8 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 2 metai, darbui – 2 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametrų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2007-2009	2009-2011	10	404.84	5	9	3
2009-2011	2011-2013	2	247.22	12	6	11
2011-2013	2013-2015	5	493.52	18	10	8
2013-2015	2015-2017	4	-224.70	13	4	6
2015-2017	2017-2017.05	3	252.28	5	2	2
	Viso:	24	1173.16			



3.9 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 2 metai, o darbui – 2 metai.

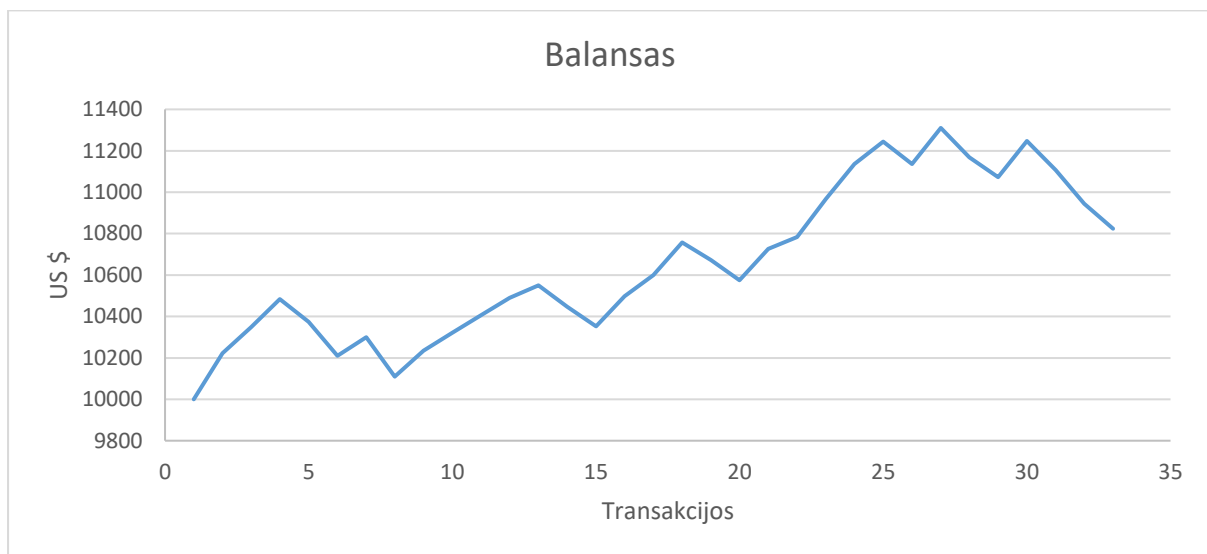
Sistemą optimizuojant kas 1 metus, o optimizavimui skiriant 2 metų laikotarpio duomenis su geriausiais parametrų rinkiniais buvo pasiektas 1173.16 dolerių pelnas (11.73% pelningumas). Sistema padarė 24 transakcijas.

Sistemos darbui skiriami 3 metai

Ieškomas optimalus parametrų rinkinys dviejų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2007.01.

3.9 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 2 metai, darbui – 3 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2007-2009	2009-2012	19	574.81	5	9	3
2010-2012	2012-2015	8	593.62	18	4	12
2013-2015	2015-2017.05	5	-344.68	13	4	6
	Viso:	32	823.75			



3.10 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 2 metai, o darbui – 3 metai.

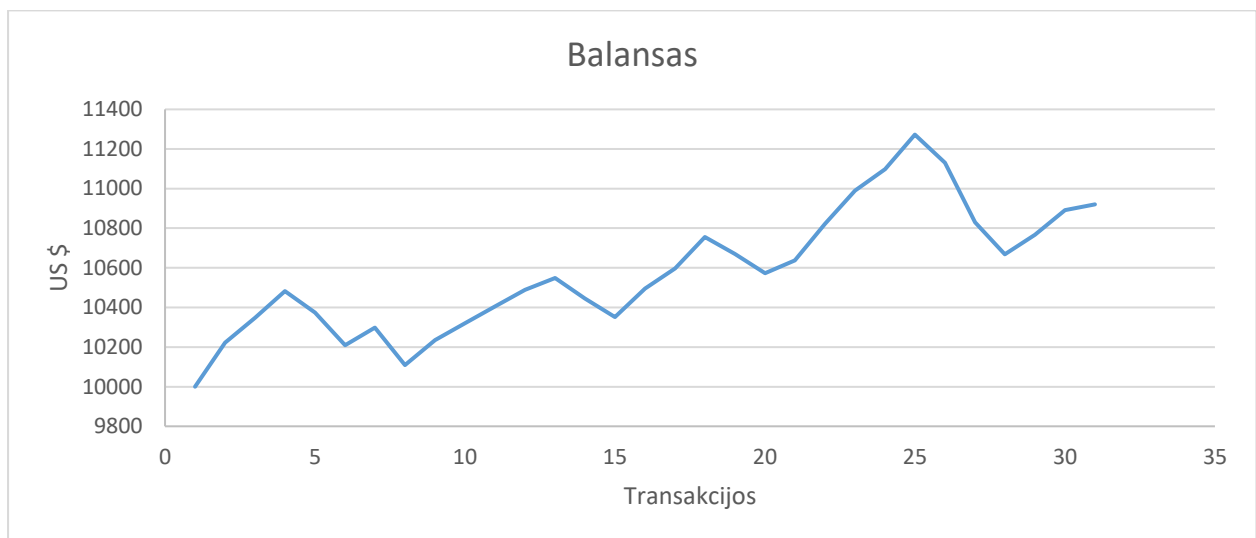
Sistemą optimizuojant kas 3 metus, o optimizavimui skiriant 2 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 823.75 dolerių pelnas (8.23% pelningumas). Sistema padarė 32 transakcijas.

Sistemos darbui skiriami 4 metai

Ieškomas optimalus parametų rinkinys dviejų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2007.01.

3.10 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 2 metai, darbui – 4 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametrų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2007-2009	2009-2013	20	637.25	5	9	3
2011-2013	2013-2017	7	30.26	18	10	8
2015-2017	2017-2017.05	3	252.28	5	2	2
	Viso:	30	919.79			



3.11 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 2 metai, o darbui – 4 metai.

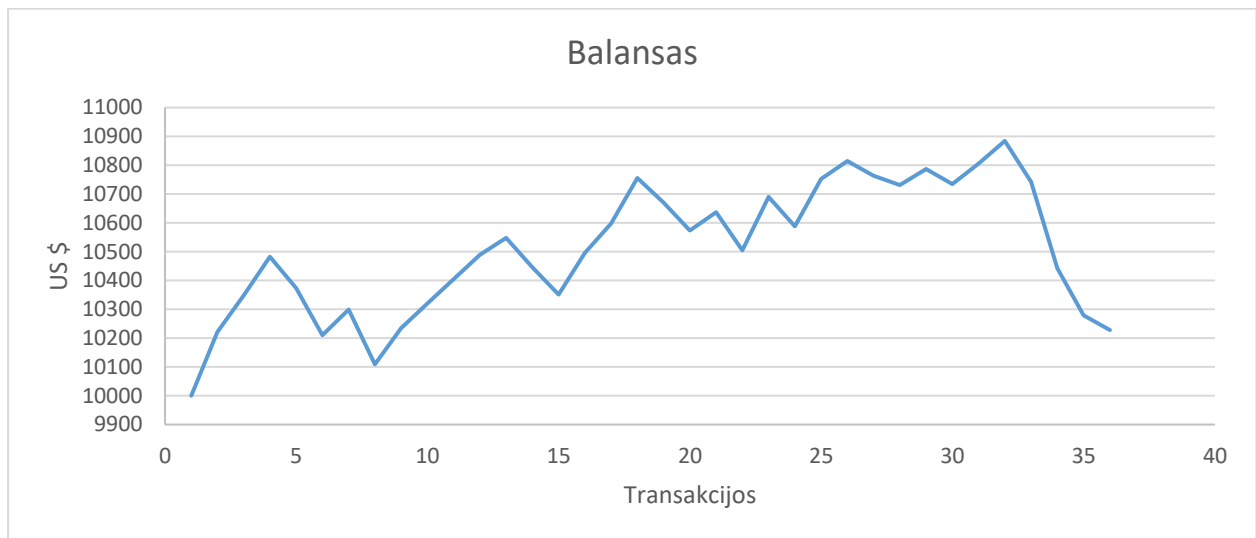
Sistemą optimizuojant kas 4 metus, o optimizavimui skiriant 2 metų laikotarpio duomenis su geriausiais parametrų rinkiniais buvo pasiektas 919.79 dolerių pelnas (9.20% pelningumas). Sistema padarė 30 transakcijas.

Sistemos darbui skiriami 5 metai

Ieškomas optimalus parametrų rinkinys dviejų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2007.01.

3.11 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 2 metai, darbui – 5 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2007-2009	2009-2014	30	806.05	5	9	3
2012-2014	2014-2017.05	5	-577.75	18	10	4
	Viso:	35	228.3			



3.12 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 2 metai, o darbui – 5 metai.

Sistemą optimizuojant kas 5 metus, o optimizavimui skiriant 2 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 228.3 dolerių pelnas (2.28 % pelningumas). Sistema padarė 35 transakcijas.

3.2.2. 3 metų laikotarpio optimizavimas

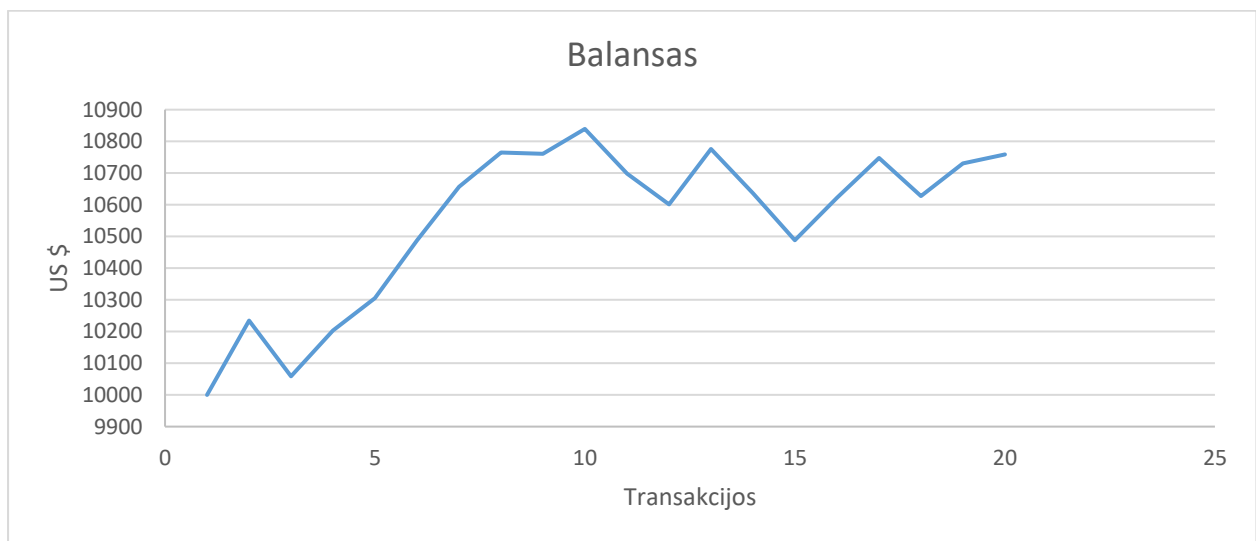
Sistemos apmokymui skiriami 3 metai. Buvo tiriama, koks darbo laikotarpis (1,2,3,4 ar 5 metai) atneša geriausius rezultatus.

Sistemos darbui skiriami 1 metai

Ieškomas optimalus parametų rinkinys trijų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus trijų metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2006.01.

3.12 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 3 metai, darbui – 1 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametrų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2006-2009	2009-2010	1	234.42	17	5	12
2007-2010	2010-2011	1	-175.95	12	10	8
2008-2011	2011-2012	1	145.03	12	4	11
2009-2012	2012-2013	1	101.58	12	3	11
2010-2013	2013-2014	4	456.12	18	3	13
2011-2014	2014-2015	2	-63.37	18	7	1
2012-2015	2015-2016	2	78.06	12	9	8
2013-2016	2016-2017	3	-153.16	11	9	8
2014-2017	2017-2017.05	4	136.06	10	8	7
	Viso:	19	758.79			



3.13 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 3 metai, o darbui – 1 metai.

Sistemą optimizuojant kas 1 metus, o optimizavimui skiriant 3 metų laikotarpio duomenis su geriausiais parametrų rinkiniais buvo pasiektas 758.79 dolerių pelnas (7.59% pelningumas).

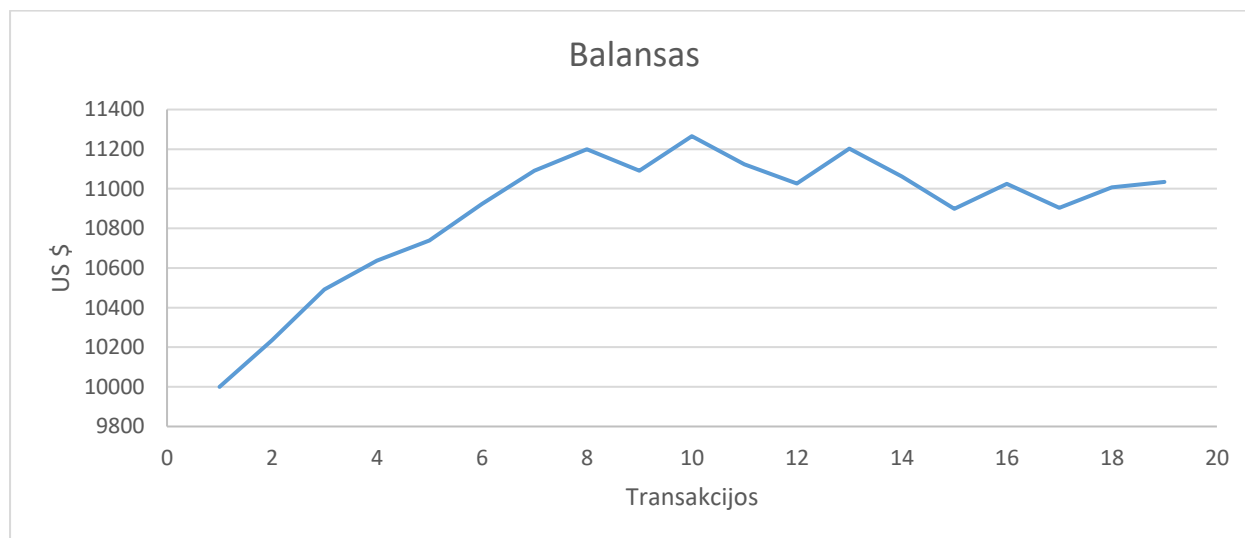
Sistema padarė 19 transakcijas.

Sistemos darbui skiriami 2 metai

Ieškomas optimalus parametų rinkinys trijų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 2 metus, o po to ieškomas naujas optimalus trijų metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2006.01.

3.13 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 3 metai, darbui – 2 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2006-2009	2009-2011	2	491.60	17	5	12
2008-2011	2011-2013	2	247.08	12	4	11
2010-2013	2013-2015	6	385.06	18	3	13
2012-2015	2015-2017	4	-224.49	12	9	8
2014-2017	2017-2017.05	4	136.06	10	8	7
	Viso:	18	1035.31			



3.14 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 3 metai, o darbui – 2 metai.

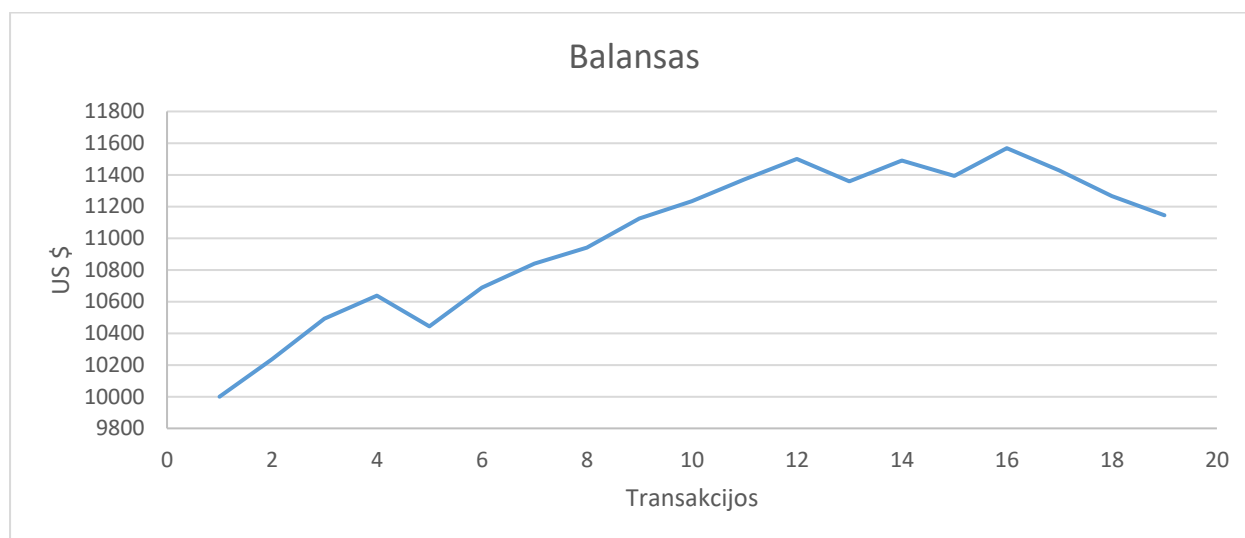
Sistemą optimizuojant kas 2 metus, o optimizavimui skiriant 3 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 1035.31 dolerių pelnas (10.35 % pelningumas). Sistema padarė 18 transakcijas.

Sistemos darbui skiriami 3 metai

Ieškomas optimalus parametrų rinkinys trijų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 3 metus, o po to ieškomas naujas optimalus trijų metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2006.01.

3.14 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 3 metai, darbui – 3 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametrų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2006-2009	2009-2012	6	839.45	17	5	12
2009-2012	2012-2015	7	650.68	12	3	11
2012-2015	2015-2017.05	5	-344.57	12	9	8
	Viso:	18	1145.56			



3.15 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 3 metai, o darbui – 3 metai.

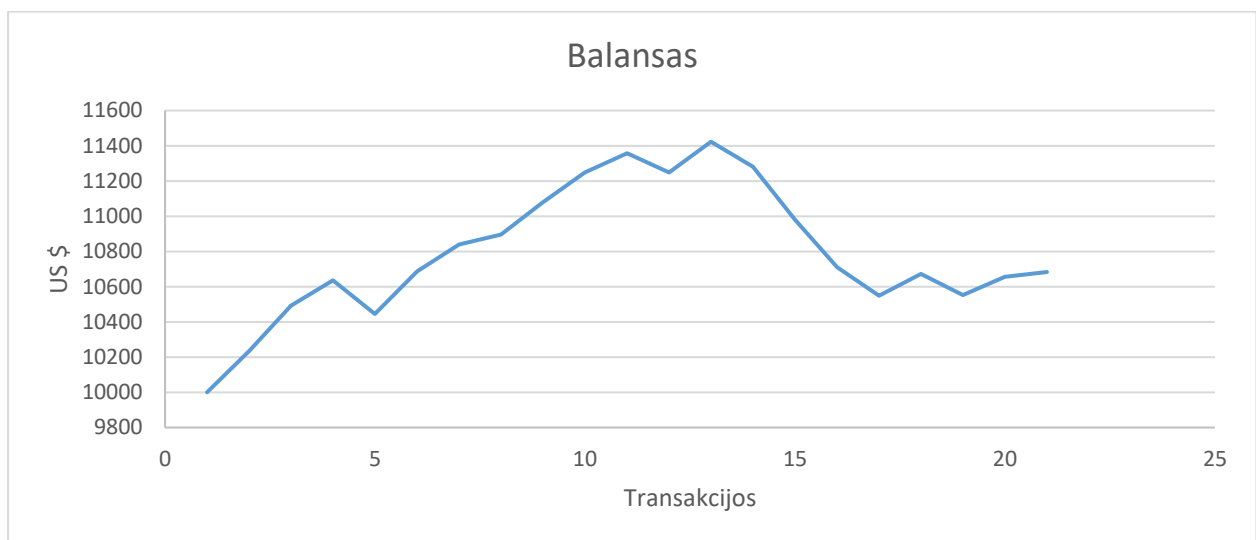
Sistemą optimizuojant kas 3 metus, o optimizavimui skiriant 3 metų laikotarpio duomenis su geriausiais parametrų rinkiniais buvo pasiektas 1145.56 dolerių pelnas (11.45% pelningumas). Sistema padarė 18 transakcijas.

Sistemos darbui skiriami 4 metai

Ieškomas optimalus parametrų rinkinys trijų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 4 metus, o po to ieškomas naujas optimalus trijų metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2006.01.

3.15 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 3 metai, darbui – 4 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2006-2009	2009-2013	7	896.28	17	5	12
2010-2013	2013-2017	9	-348.53	18	3	13
2014-2017	2017-2017.05	4	136.06	10	8	7
	Viso:	20	683.81			



3.16 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 3 metai, o darbui – 4 metai.

Sistemą optimizuojant kas 4 metus, o optimizavimui skiriant 3 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 683.81 dolerių pelnas (6.83% pelningumas).

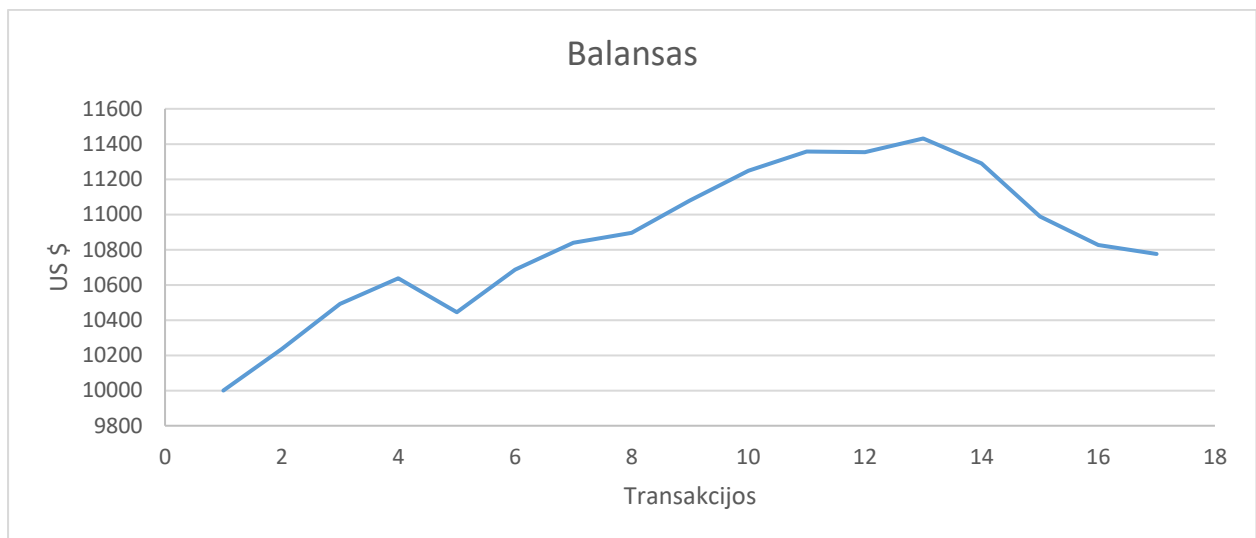
Sistema padarė 20 transakcijas.

Sistemos darbui skiriami 5 metai

Ieškomas optimalus parametų rinkinys trijų metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus trijų metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2006.01.

3.16 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 3 metai, darbui – 5 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2006-2009	2009-2014	11	1353.17	17	5	12
2011-2014	2014-2017.05	5	-577.73	18	7	1
Viso:		16	775.44			



3.17 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 3 metai, o darbui – 5 metai.

Sistemą optimizuojant kas 5 metus, o optimizavimui skiriant 3 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 775.44 dolerių pelnas (7.75% pelningumas). Sistema padarė 16 transakcijas.

3.2.3. 4 metų laikotarpio optimizavimas

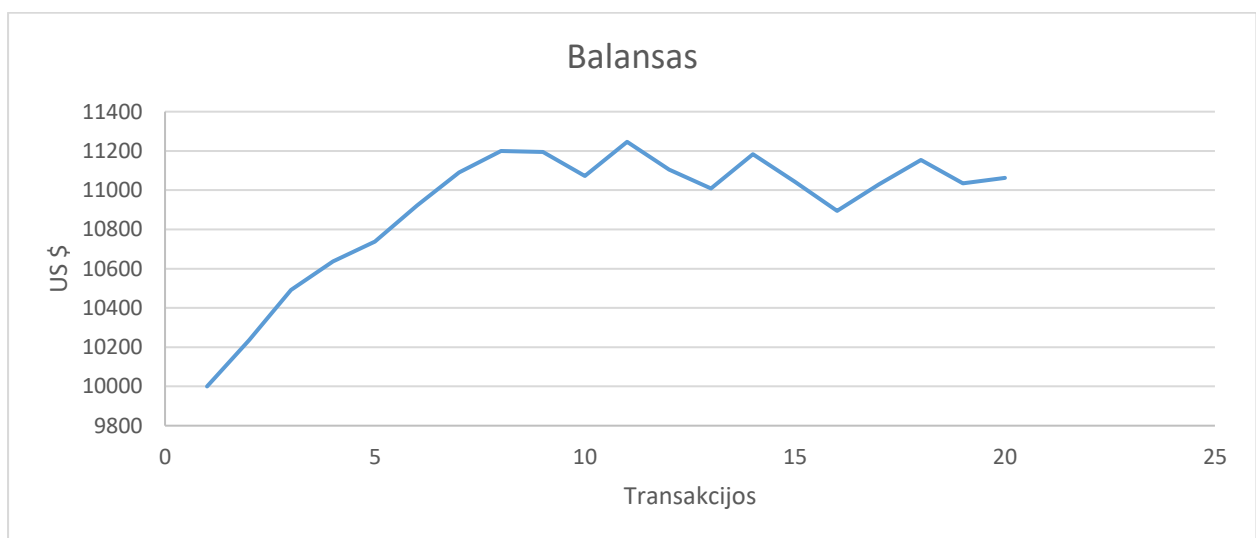
Sistemos apmokymui skiriami 4 metai. Buvo tiriama, koks darbo laikotarpis (1,2,3,4 ar 5 metai) atneša geriausius rezultatus.

Sistemos darbui skiriami 1 metai

Ieškomas optimalus parametų rinkinys keturių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2005.01.

3.17 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 4 metai, darbui – 1 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametrų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2005-2009	2009-2010	1	234.85	17	2	6
2006-2010	2010-2011	1	256.44	12	6	11
2007-2011	2011-2012	1	145.30	12	2	11
2008-2012	2012-2013	1	101.7	12	10	11
2009-2013	2013-2014	4	456.72	18	6	11
2010-2014	2014-2015	3	-90.30	18	5	12
2011-2015	2015-2016	2	78.20	13	7	6
2012-2016	2016-2017	3	-153.41	11	3	8
2013-2017	2017-2017.05	3	33.08	11	7	6
	Viso:	19	1062.58			



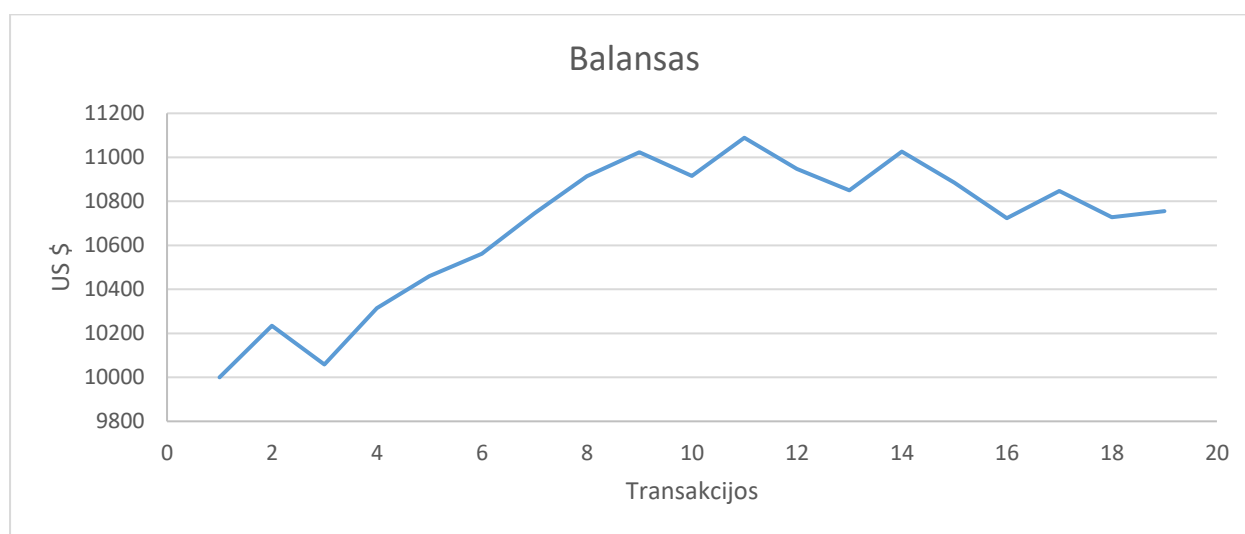
3.18 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 4 metai, o darbui – 1 metai.

Sistemą optimizuojant kas 1 metus, o optimizavimui skiriant 4 metų duomenis su geriausiais parametrų rinkiniais buvo pasiektas 1062.58 dolerių pelnas (10.63% pelningumas). Sistema padarė 19 transakcijas.

Sistemos darbui skiriami 2 metai

Ieškomas optimalus parametų rinkinys keturių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 2 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2005.01. 3.18 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 4 metai, darbui – 2 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2005-2009	2009-2011	3	315.08	17	2	6
2007-2011	2011-2013	2	247.05	12	2	11
2009-2013	2013-2015	6	385.08	18	6	11
2011-2015	2015-2017	4	-224.45	13	7	6
2013-2017	2017-2017.05	3	33.08	11	7	6
Viso:		18	755.84			



3.19 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 4 metai, o darbui – 2 metai.

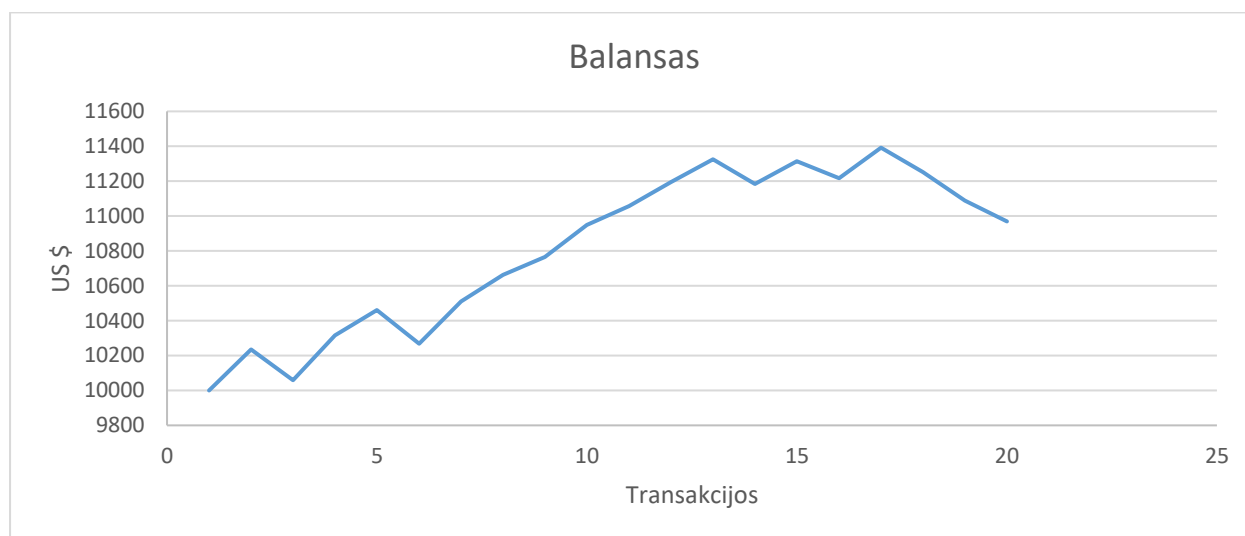
Sistemą optimizuojant kas 2 metus, o optimizavimui skiriant 4 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 755.84 dolerių pelnas (7.55% pelningumas). Sistema padarė 18 transakcijas.

Sistemos darbui skiriami 3 metai

Ieškomas optimalus parametų rinkinys keturių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 3 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2005.01.

3.19 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 4 metai, darbui – 3 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2005-2009	2009-2012	7	663.07	17	2	6
2008-2012	2012-2015	7	650.56	12	10	11
2011-2015	2015-2017.05	5	-344.48	13	7	6
	Viso:	19	969.15			



3.20 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 4 metai, o darbui – 3 metai.

Sistemą optimizuojant kas 3 metus, o optimizavimui skiriant 4 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 969.15 dolerių pelnas(9.69% pelningumas). Sistema padarė 19 transakcijas.

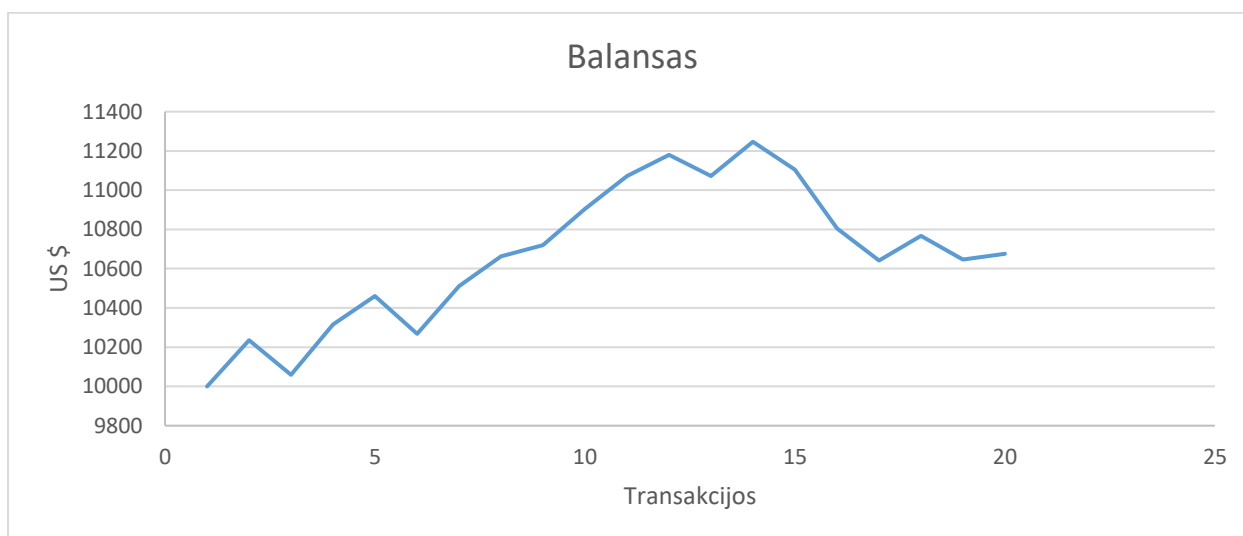
Sistemos darbui skiriami 4 metai

Ieškomas optimalus parametų rinkinys keturių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 4 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2005.01.

3.20 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 4 metai, darbui – 4 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2005-2009	2009-2013	8	719.85	17	2	6
2009-2013	2013-2017	8	-77.82	18	6	11

2013-2017	2017-2017.05	3	33.08	11	7	6
	Viso:	19	675.11			



3.21 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 4 metai, o darbui – 4 metai.

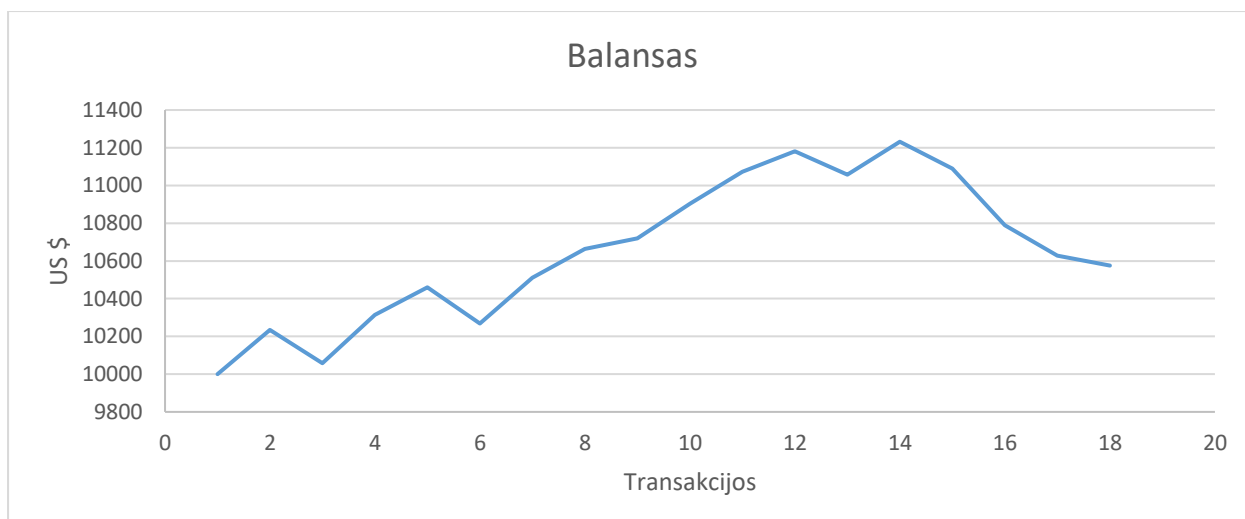
Sistemą optimizuojant kas 1 metus, o optimizavimui skiriant 4 metų duomenis su geriausiai parametų rinkiniais buvo pasiektas 675.11 dolerių pelnas (6.75% pelningumas). Sistema padarė 19 transakcijas.

Sistemos darbui skiriami 5 metai

Ieškomas optimalus parametų rinkinys keturių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 5 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2005.01.

3.21 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 4 metai, darbui – 5 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2005-2009	2009-2014	11	1180.52	17	2	6
2010-2014	2014-2017.05	6	-604.34	18	5	12
	Viso:	17	576.18			



3.22 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 4 metai, o darbui – 5 metai.

Sistemą optimizuojant kas 5 metus, o optimizavimui skiriant 4 metų laikotarpio duomenis su geriausiais parametru rinkiniais buvo pasiektas 576.18 dolerių pelnas (5.76% pelningumas). Sistema padarė 17 transakcijas.

3.2.4. 5 metų laikotarpio optimizavimas

Sistemos apmokymui skiriami 5 metai. Buvo tiriama, koks darbo laikotarpis (1,2,3,4 ar 5 metai) atneša geriausius rezultatus.

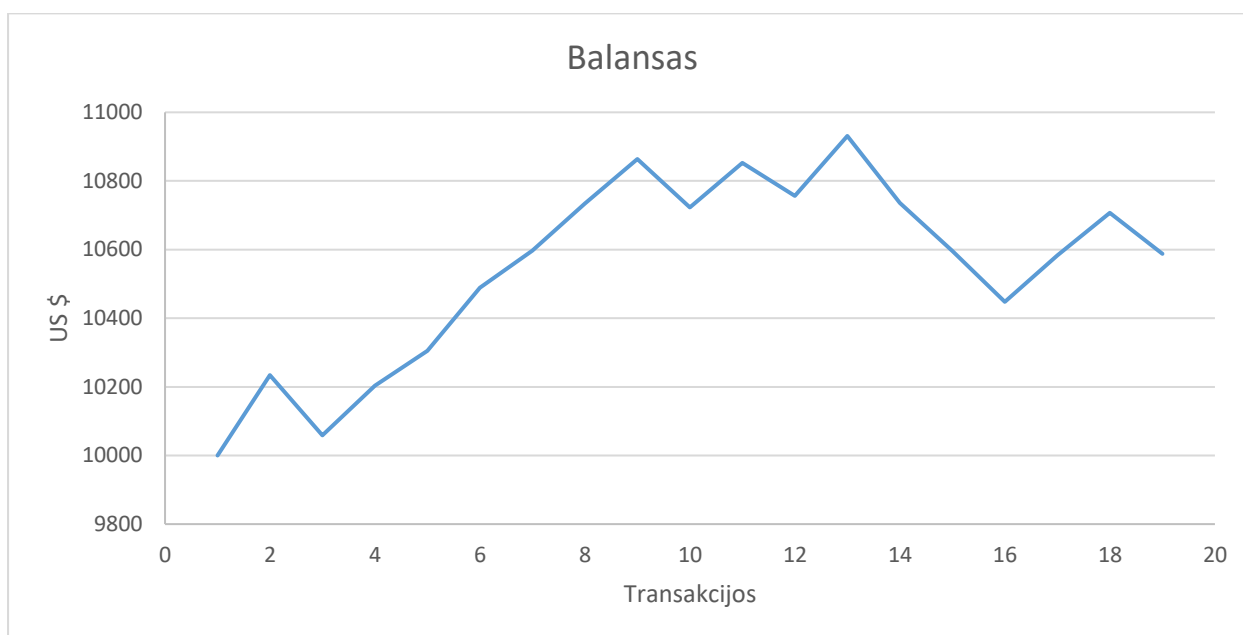
Sistemos darbui skiriami 1 metai

Ieškomas optimalus parametru rinkinys penkių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Iširtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2004.01.

3.22 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 5 metai, darbui – 1 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis \$	Optimalus parametru rinkinys		
				InpDepth	InpDeviation	InpBackstep
2004-2009	2009-2010	1	234.42	17	9	6
2005-2010	2010-2011	1	-175.95	12	10	8
2006-2011	2011-2012	1	145.03	12	2	11
2007-2012	2012-2013	1	101.58	12	9	11
2008-2013	2013-2014	2	291.75	12	10	11

2009-2014	2014-2015	4	256.17	12	2	11
2010-2015	2015-2016	3	-117.11	12	4	11
2011-2016	2016-2017	2	-153.16	11	9	8
2012-2017	2017-2017.05	2	5.01	11	2	8
Viso:		17	587.74			



3.23 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 5 metai, o darbui – 1 metai.

Sistemą optimizuojant kas 1 metus, o optimizavimui skiriant 5 metų duomenis su geriausiais parametų rinkiniais buvo pasiektas 587.74 dolerių pelnas (5.87% pelningumas). Sistema padarė 17 transakcijas.

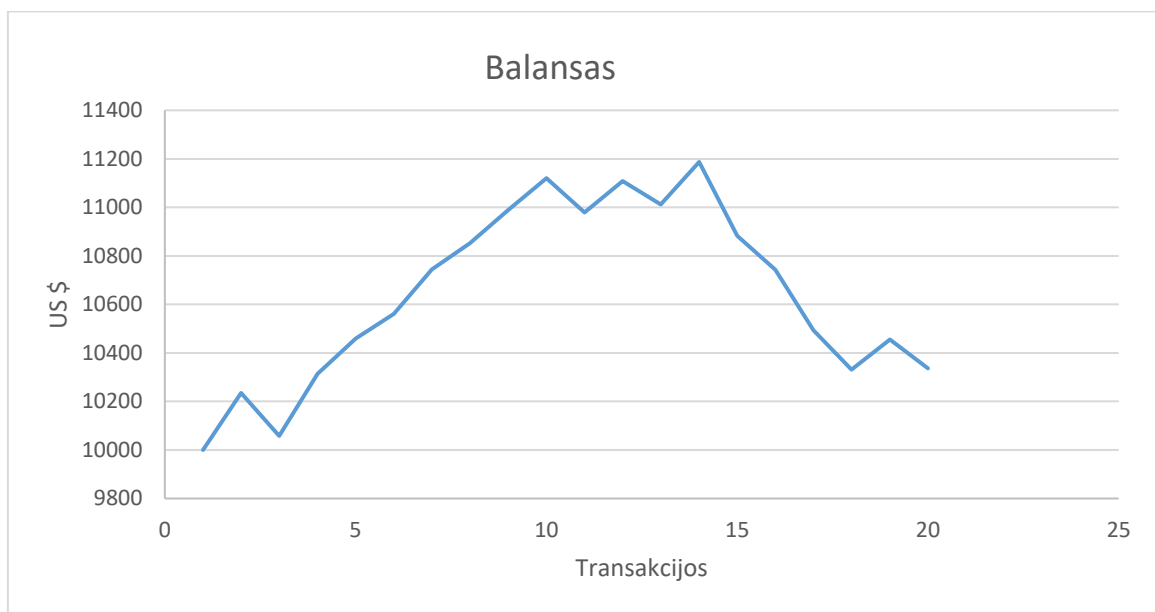
Sistemos darbui skiriami 2 metai

Ieškomas optimalus parametų rinkinys penkių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 2 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2004.01.

3.23 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 5 metai, darbui – 2 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2004-2009	2009-2011	3	314.49	17	9	6

2006-2011	2011-2013	2	246.64	12	2	11
2008-2013	2013-2015	6	547.88	12	10	11
2010-2015	2015-2017	6	-778.04	12	4	11
2012-2017	2017-2017.05	2	5.01	11	2	8
Viso:		19	335.98			



3.24 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 5 metai, o darbui – 2 metai.

Sistemą optimizuojant kas 2 metus, o optimizavimui skiriant 5 metų duomenis su geriausiais parametru rinkiniais buvo pasiektas 335.98 dolerių pelnas(3.35 % pelningumas). Sistema padarė 19 transakcijas.

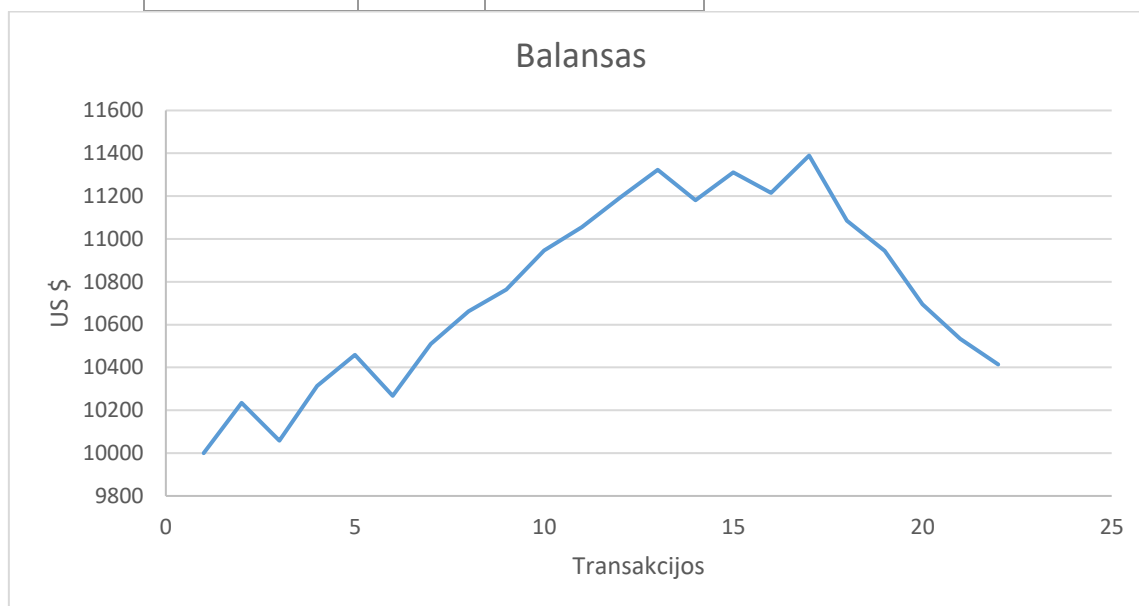
Sistemos darbui skiriami 3 metai

Ieškomas optimalus parametru rinkinys penkių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 3 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2004.01.

3.24 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 5 metai, darbui – 3 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametru rinkinys		
				InpDepth	InpDeviation	InpBackstep
2004-2009	2009-2012	7	661.85	17	9	6
2007-2012	2012-2015	7	649.41	12	9	11

2010-2015	2015-2017.05	7	-897.87	12	4	11
	Viso:	21	413.39			



3.25 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 5 metai, o darbui – 3 metai.

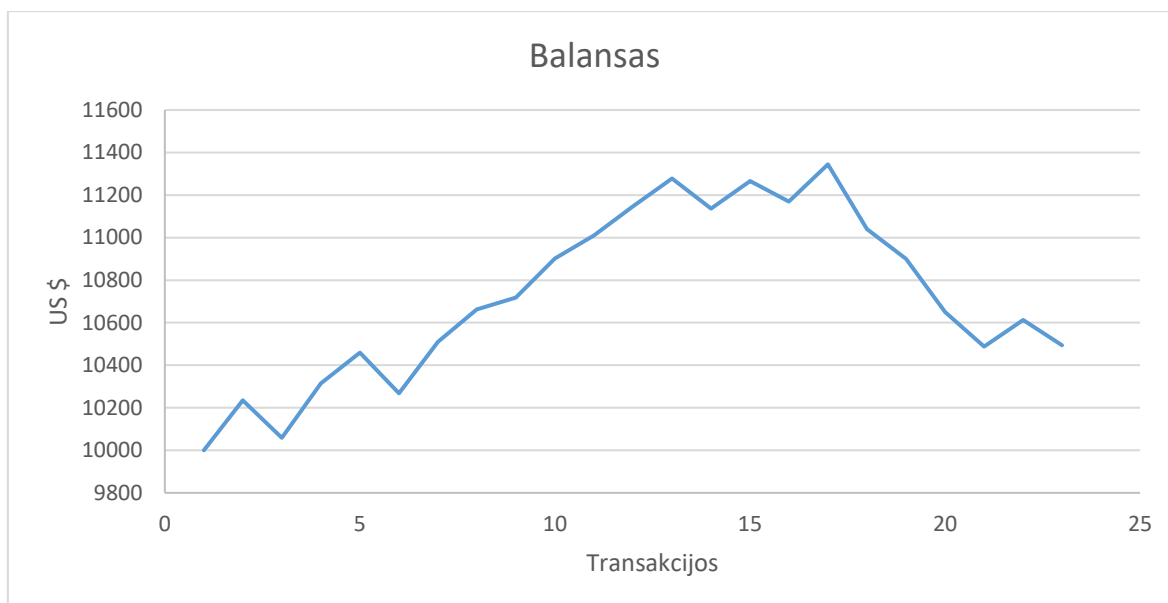
Sistemą optimizuojant kas 3 metus, o optimizavimui skiriant 5 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 413.39 dolerių pelnas (4.13% pelningumas). Sistema padarė 21 transakcijas.

Sistemos darbui skiriami 4 metai

Ieškomas optimalus parametų rinkinys penkių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 4 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2004.01.

3.25 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 5 metai, darbui – 4 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2004-2009	2009-2013	8	718.52	17	9	6
2008-2013	2013-2017	12	-230.17	12	10	11
2012-2017	2017-2017.05	2	5.01	11	2	8
	Viso:	22	493.36			



3.26 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 5 metai, o darbui – 4 metai.

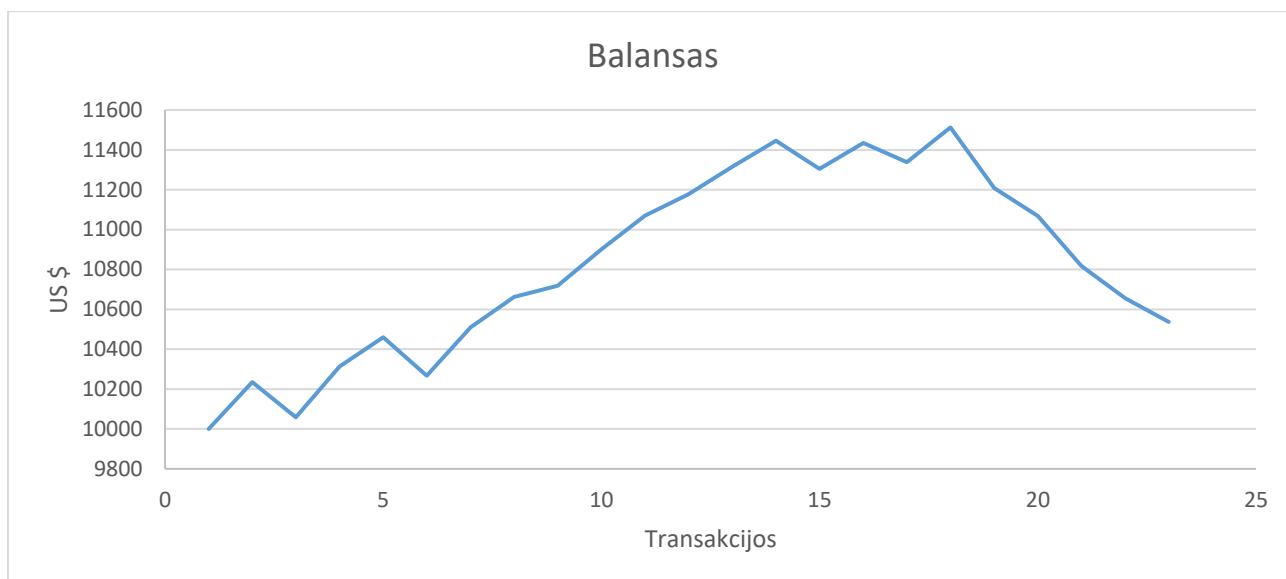
Sistemą optimizuojant kas 4 metus, o optimizavimui skiriant 5 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 493.36 dolerių pelnas (4.93% pelningumas). Sistema padarė 22 transakcijas.

Sistemos darbui skiriami 5 metai

Ieškomas optimalus parametų rinkinys penkių metų laikotarpyje. Su šiuo rinkiniu Sistema dirba 5 metus, o po to ieškomas naujas optimalus penkių metų rinkinys. Ištirtas tokios sistemos pelningumas 2009.01-2017.05 laikotarpiu. Pirmo optimizavimo pradžia 2004.01.

3.26 lentelė. Testavimo rezultatai, kai optimizavimui skiriami 5 metai, darbui – 5 metai.

Optimizavimo laikotarpis	Prekiavimo laikotarpis	Atlikta transakcijų	Pelnas/Nuostolis	Optimalus parametų rinkinys		
				InpDepth	InpDeviation	InpBackstep
2004-2009	2009-2014	11	1178.38	17	9	6
2009-2014	2014-2017.05	11	-641.72	12	2	11
Viso:		22	536.66			



3.27 pav. H4 sistemos darbo rezultatai 2009.01-2017.05 laikotarpiu, kai sistemos apmokymui skiriami 5 metai, o darbui – 5 metai.

Sistemą optimizuojant kas 5 metus, o optimizavimui skiriant 5 metų laikotarpio duomenis su geriausiais parametų rinkiniais buvo pasiektas 536.66 dolerių pelnas (5.37% pelningumas). Sistema padarė 22 transakcijas.

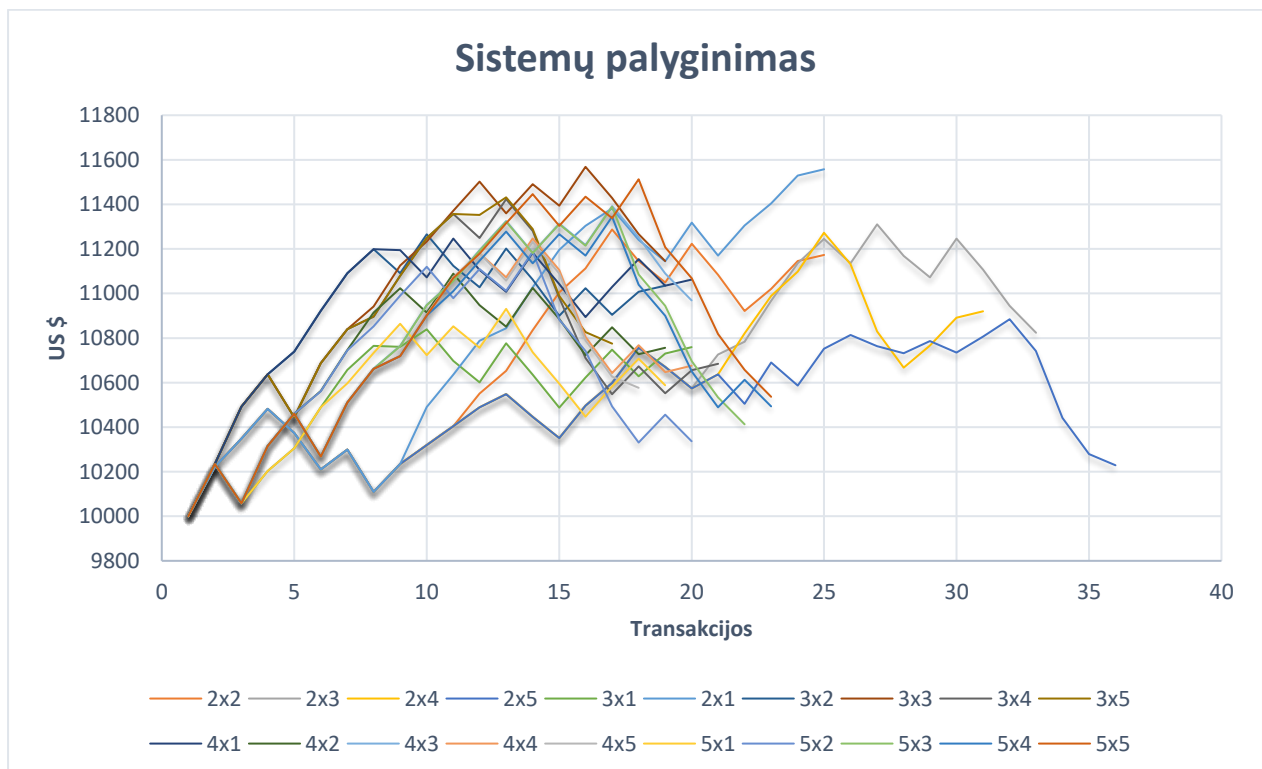
3.2.5. Iširtų sistemų palyginimas

3.1 skyriuje buvo iširtos sistemos, kurių darbui su vienu parametų rinkiniu buvo skiriama 1,2,3,4,5 metai, o optimizacija atliekama su 2,3,4,5 metais. Šiame skyriuje palyginama, kuri sistema buvo pelningiausia. Visos sistemos dirbo su keturių valandų grafiku (H4) nuo 2009 metų iki 2017 metų gegužės.

3.27 lentelė. Sistemų palyginimo lentelė.

Optimizavimui skirta metų	Prekiavimui skirta metų	Transakcijų skaičius	Pelnas/Nuostolis
2	1	24	1557.48
2	2	24	1173.16
2	3	32	823.75
2	4	30	919.79
2	5	35	228.3
3	1	19	758.79
3	2	18	1035.31

3	3	18	1145.56
3	4	20	683.81
3	5	16	775.44
4	1	19	1062.58
4	2	18	755.84
4	3	19	969.15
4	4	19	675.11
4	5	17	576.18
5	1	17	587.74
5	2	19	335.98
5	3	21	413.39
5	4	22	493.36
5	5	22	536.66



3.28 pav. Sistemų palyginimas (pirmas kreivės pavadinimo skaičius reiškia optimizavimo metų skaičių, antras- darbo metų skaičių)

Palyginus visas 3.2 skyriuje tirtas sistemas, matome, kad geriausi rezultatai pasiekti, kai parametų perrinkimo dažnis didžiausias ir parametų perrinkimui naudojami 2 praėjusių

metų duomenys. Didinant Optimizavimui skirtų metų skaičių ir/arba optimizavimo dažnį, rezultatai prastėja.

IŠVADOS IR PASIŪLYMAI

1. Ištyrus rinkoje naudojamus indikatorius ir konfigūracijas, bei apžvelgus seniau darytus tyrimus, buvo pasirinkta dirbti su harmoninėmis kainų konfigūracijomis. Sukurtas šiais kainų modeliais paremtas prekiaavimo algoritmas.
2. Panaudojant MetaTrader4 platformą ir MQL4 programavimo kalbą, buvo sukurta automatinė prekiaavimo sistema, leidžianti prekiautojui tiksliai apibrėžti prekiaavimo taisykles, saugiai įeiti ir išeiti iš rinkos, bei sutaupyti prekiautojo laiko. Sistema prekiaavimas yra paremtas harmoninėmis kainų konfigūracijomis.
3. Buvo ištirta, kaip dirbtų sistemos, jei visas turimas istorinių duomenų diapazonas (2000.01-2017.05) būtų padalintas į dvi dalis. Pirmoji dalis skirta optimalių sistemos parametų radimui. O antroji – sistemos testavimui su rasta parametrais. Buvo tiriami keturių valandų, valandos ir 30 minčių grafikai. Geriausi rezultatai buvo pasiekti su keturių valandų grafiku (1398.92 doleriai) (13.99% viso laikotarpio pelningumas), o su 30 minučių grafiku patirtas nuostolis (-1069,71) (-10.70% viso laikotarpio pelningumas). Vienos valandos grafikas būtų atnešęs 1365.14 dolerių pelną (13.65% viso laikotarpio pelningumas). Sistemos dirbančios ant valandinio ir keturių valandų grafiko galėtų sudaryti dalį portfelio.
4. Patobulinta ir ištirta 3.1.1 skyriuje analizuota sistema, įvedant papildomą patvirtinimo sąlyga. Tokiu būdu buvo tiksliau įeinama į rinką. Pasiiektas 0.11% didesnis pelningumas.
5. Taip pat buvo atliktas tyrimas, kuriuo buvo siekiama rasti koku dažniu geriausia atlikti optimizavimus ir kiek metų istorinių duomenų skirti optimalių parametų radimui. Tyrimas parodė, kad geriausi rezultatai gaunami, kai optimizuojama kas metus, o optimizavimui skiriami 2 metų istoriniai duomenys (1557.48 dolerių pelnas, 15.57% viso laikotarpio pelningumas). Didinant optimizavimui skirtų metų skaičių, rezultatai blogėja. Tai rodo, kad geriausiam sistemos veikimui didžiausią įtaką daro duomenų pagal kuriuos buvo nustatyti sistemos parametrai naujumas. Taip pat blogėja sistemos kokybė, kai didinamas darbo metų skaičius.

6. Tolimesni tyrimai galėtų būti atlikti su skirtingomis valiutų poromis. Taip pat būtų galima iširti kaip veiktų 3.2 skyriuje aprašyta sistema, tik su trumpesnio laiko grafikais (H1, M30), tačiau tam reikia didelių kompiuterinių resursų.
7. Taip pat gali būti atlikti tyrimai derinant harmonines kainų konfigūracijas su kitais indikatoriais.
8. Ištyrus sistemos veikimą su kitomis valiutų poromis ir pritaikius rizikos valdymą, būtų galima sudaryti portfelį iš efektyviausiai veikiančių sistemų.

LITERATŪROS SĄRAŠAS

1. Techninės analizės sąvoka [Interaktyvus] žiūrėta 2017.05.02-
https://lt.wikipedia.org/wiki/Technin%C4%97_analiz%C4%97
2. Martin Sewell (Department of Computer Science, University College London)
„Behavioural Finance“ [Interaktyvus] žiūrėta 2017.05.02
<https://fernandonogueiracosta.files.wordpress.com/2015/08/sewell-martin-behavioural-finance-2008.pdf>
3. Hill A. Technical Analysis. [Interaktyvus] žiūrėta 2017.05.02
<http://www.stockcharts.com/education/TAindex.html>
4. Slenkantys vidurkiai. [Interaktyvus] žiūrėta 2017.05.02
<http://www.investopedia.com/university/technical/techanalysis9.asp>
5. Karthik Marar A Guide to technical analysis – MACD. [Interaktyvus] žiūrėta
2017.05.03
http://cdn3.traderslaboratory.com/forums/attachments/104/8466d1225312919-discussions-macd-usage-trading_the_macd.pdf
6. Glossary of technical indicators Telemet America, Inc. . [Interaktyvus] žiūrėta
2017.05.05 <http://www.taquote.com/public/download/techdefs.pdf>
7. Martin J. Pring „Technical analysis explained“
8. Charting made easy. John Murphy. [Interaktyvus] žiūrėta 2017.05.05
http://stockcharts.com/school/doku.php?id=chart_school:john_murphy_s_charti
9. John Magee. Technical analysis [Interaktyvus] žiūrėta 2016.09.03
http://www.mrao.cam.ac.uk/~mph/Technical_Analysis.pdf
10. Cup with handle pattern. [Interaktyvus] žiūrėta 2016.09.05
http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:chart_patterns:cup_with_handle_continuation
11. Scott Carney. Harmonic Trading [Interaktyvus] žiūrėta 2016.12.05
<https://www.mta.org/wp-content/uploads/2015/11/scottcarney-090810.pdf>
12. Fawad Razaqzada. The Ultimate Fibonacci Guide. [Interaktyvus] žiūrėta 2016.12.05
<http://www.forex.com/uk/pdf/Fibonacci-guide.pdf>
13. Edward Leshik, Jane Cralle .An introduction to algorithmic trading. Basic to advanced
strategies. [Interaktyvus] žiūrėta 2016.12.05

http://financebooks.org/media/pdfs/algorithmic_trading_basic_to_advanced_strategies.pdf

14. Automated trading. [Interaktyvus] žiūrėta 2017.05.02
<http://www.investopedia.com/articles/trading/11/automated-trading-systems.asp>
15. Metatrader platformos tinklapis. [Interaktyvus] žiūrėta 2017.05.02
<https://www.metatrader4.com/en/trading-platform>
16. MQL4 kalbos programos dokumentacija [Interaktyvus] žiūrėta 2017.04.05
<https://www.mql5.com/en/docs>
17. Modeliavimo metodai [Interaktyvus] žiūrėta 2017.04.05
<https://www.mql5.com/en/articles/1511>
18. Sunil Mangwani. Fibonacci ratios and Harmonic patterns. [Interaktyvus] žiūrėta 2017.04.05 http://www.traders.lt/datas/users/10784-fibs_and_harmonic_patterns.pdf
19. Gene I. Sher. University of Central Florida. Evolving Chart Pattern Sensitive Neural Network Based Forex Trading Agents. [Interaktyvus] žiūrėta 2016.04.05 :
<https://arxiv.org/ftp/arxiv/papers/1111/1111.5892.pdf>
20. Journal of management (2013, Nr.1(22)). Valiutų kursų analizei ir prognozavimui naudojamų metodų testavimas. Jonas Nedzveckas, Rimantas Dapkus. [Interaktyvus] žiūrėta 2016.04.05 : http://www.ltvk.lt/file/manual/Vadyba/Vadyba_2013_22.pdf
21. Murat Ozturk, Ismail Hakki Toroslu, Guven Fidan. Heuristic based trading system on Forex data using technical indicator rules. [Interaktyvus] žiūrėta 2016.04.05 :
http://ac.els-cdn.com/S1568494616300369/1-s2.0-S1568494616300369-main.pdf?tid=890bd6ae-3e55-11e7-9c4c-00000aacb361&acdnat=1495392455_d2fdbd31668a12b646ee4d83da2ad1fa

PRIEDAI

1 Priedas

```
//+-----+
//              ZigZag.mq4 |
//      Copyright 2006-2014, MetaQuotes Software Corp. |
//              http://www.mql4.com |
//+-----+
#property copyright "2006-2014, MetaQuotes Software Corp."
#property link      "http://www.mql4.com"
#property strict

#property indicator_chart_window
#property indicator_buffers 1
#property indicator_color1 Red
//--- indicator parameters
input int InpDepth=12; // Depth
input int InpDeviation=5; // Deviation
input int InpBackstep=3; // Backstep
//--- indicator buffers
double ExtZigzagBuffer[];
double ExtHighBuffer[];
double ExtLowBuffer[];
//--- globals
int ExtLevel=3; // recounting's depth of extremums
//+-----+
//| Custom indicator initialization function |
//+-----+
int OnInit()
{
    if(InpBackstep>=InpDepth)
    {
        Print("Backstep cannot be greater or equal to Depth");
        return(INIT_FAILED);
    }
}
//--- 2 additional buffers
IndicatorBuffers(3);
//--- drawing settings
SetIndexStyle(0,DRAW_SECTION);
//--- indicator buffers
SetIndexBuffer(0,ExtZigzagBuffer);
SetIndexBuffer(1,ExtHighBuffer);
SetIndexBuffer(2,ExtLowBuffer);
SetIndexEmptyValue(0,0.0);
//--- indicator short name
IndicatorShortName("ZigZag("+string(InpDepth)+","+string(InpDeviation)+","+string(InpBackstep)+")");
//--- initialization done
return(INIT_SUCCEEDED);
}
//+-----+
//| |
```

```

//+-----+
int OnCalculate(const int rates_total,
               const int prev_calculated,
               const datetime &time[],
               const double &open[],
               const double &high[],
               const double &low[],
               const double &close[],
               const long& tick_volume[],
               const long& volume[],
               const int& spread[])
{
    int i,limit,counterZ,whatlookfor=0;
    int back,pos,lasthighpos=0,lastlowpos=0;
    double extremum;
    double curlow=0.0,curhigh=0.0,lasthigh=0.0,lastlow=0.0;
//--- check for history and inputs
    if(rates_total<InpDepth || InpBackstep>=InpDepth)
        return(0);
//--- first calculations
    if(prev_calculated==0)
        limit=InitializeAll();
    else
    {
        //--- find first extremum in the depth ExtLevel or 100 last bars
        i=counterZ=0;
        while(counterZ<ExtLevel && i<100)
        {
            if(ExtZigzagBuffer[i]!=0.0)
                counterZ++;
            i++;
        }
        //--- no extremum found - recounting all from begin
        if(counterZ==0)
            limit=InitializeAll();
        else
        {
            //--- set start position to found extremum position
            limit=i-1;
            //--- what kind of extremum?
            if(ExtLowBuffer[i]!=0.0)
            {
                //--- low extremum
                curlow=ExtLowBuffer[i];
                //--- will look for the next high extremum
                whatlookfor=1;
            }
            else
            {
                //--- high extremum
                curhigh=ExtHighBuffer[i];
                //--- will look for the next low extremum
                whatlookfor=-1;
            }
            //--- clear the rest data

```

```

    for(i=limit-1; i>=0; i--)
    {
        ExtZigzagBuffer[i]=0.0;
        ExtLowBuffer[i]=0.0;
        ExtHighBuffer[i]=0.0;
    }
}
}
//--- main loop
for(i=limit; i>=0; i--)
{
    //--- find lowest low in depth of bars
    extremum=low[iLowest(NULL,0,MODE_LOW,InpDepth,i)];
    //--- this lowest has been found previously
    if(extremum==lastlow)
        extremum=0.0;
    else
    {
        //--- new last low
        lastlow=extremum;
        //--- discard extremum if current low is too high
        if(low[i]-extremum>InpDeviation*Point)
            extremum=0.0;
        else
        {
            //--- clear previous extremums in backstep bars
            for(back=1; back<=InpBackstep; back++)
            {
                pos=i+back;
                if(ExtLowBuffer[pos]!=0 && ExtLowBuffer[pos]>extremum)
                    ExtLowBuffer[pos]=0.0;
            }
        }
    }
}
//--- found extremum is current low
if(low[i]==extremum)
    ExtLowBuffer[i]=extremum;
else
    ExtLowBuffer[i]=0.0;
//--- find highest high in depth of bars
extremum=high[iHighest(NULL,0,MODE_HIGH,InpDepth,i)];
//--- this highest has been found previously
if(extremum==lasthigh)
    extremum=0.0;
else
{
    //--- new last high
    lasthigh=extremum;
    //--- discard extremum if current high is too low
    if(extremum-high[i]>InpDeviation*Point)
        extremum=0.0;
    else
    {
        //--- clear previous extremums in backstep bars
        for(back=1; back<=InpBackstep; back++)

```

```

        {
            pos=i+back;
            if(ExtHighBuffer[pos]!=0 && ExtHighBuffer[pos]<extremum)
                ExtHighBuffer[pos]=0.0;
        }
    }
}
//--- found extremum is current high
if(high[i]==extremum)
    ExtHighBuffer[i]=extremum;
else
    ExtHighBuffer[i]=0.0;
}
//--- final cutting
if(whatlookfor==0)
{
    lastlow=0.0;
    lasthigh=0.0;
}
else
{
    lastlow=curlow;
    lasthigh=curhigh;
}
for(i=limit; i>=0; i--)
{
    switch(whatlookfor)
    {
        case 0: // look for peak or lawn
            if(lastlow==0.0 && lasthigh==0.0)
            {
                if(ExtHighBuffer[i]!=0.0)
                {
                    lasthigh=High[i];
                    lasthighpos=i;
                    whatlookfor=-1;
                    ExtZigzagBuffer[i]=lasthigh;
                }
                if(ExtLowBuffer[i]!=0.0)
                {
                    lastlow=Low[i];
                    lastlowpos=i;
                    whatlookfor=1;
                    ExtZigzagBuffer[i]=lastlow;
                }
            }
            break;
        case 1: // look for peak
            if(ExtLowBuffer[i]!=0.0 && ExtLowBuffer[i]<lastlow && ExtHighBuffer[i]==0.0)
            {
                ExtZigzagBuffer[lastlowpos]=0.0;
                lastlowpos=i;
                lastlow=ExtLowBuffer[i];
                ExtZigzagBuffer[i]=lastlow;
            }
    }
}

```

```

    if(ExtHighBuffer[i]!=0.0 && ExtLowBuffer[i]==0.0)
    {
        lasthigh=ExtHighBuffer[i];
        lasthighpos=i;
        ExtZigzagBuffer[i]=lasthigh;
        whatlookfor=-1;
    }
    break;
case -1: // look for lawn
    if(ExtHighBuffer[i]!=0.0 && ExtHighBuffer[i]>lasthigh && ExtLowBuffer[i]==0.0)
    {
        ExtZigzagBuffer[lasthighpos]=0.0;
        lasthighpos=i;
        lasthigh=ExtHighBuffer[i];
        ExtZigzagBuffer[i]=lasthigh;
    }
    if(ExtLowBuffer[i]!=0.0 && ExtHighBuffer[i]==0.0)
    {
        lastlow=ExtLowBuffer[i];
        lastlowpos=i;
        ExtZigzagBuffer[i]=lastlow;
        whatlookfor=1;
    }
    break;
}
}
//--- done
return(rates_total);
}
//+-----+
//|                                     |
//+-----+
int InitializeAll()
{
    ArrayInitialize(ExtZigzagBuffer,0.0);
    ArrayInitialize(ExtHighBuffer,0.0);
    ArrayInitialize(ExtLowBuffer,0.0);
//--- first counting position
return(Bars-InpDepth);
}
//+-----+

```

2 Priedas

```
{
//---
if(Bars<100)
{
Print("bars less than 100");
return(0);
}
//---
return(INIT_SUCCEEDED);
}
//+-----+
//| Expert deinitialization function |
//+-----+
void OnDeinit(const int reason)
{
//---

}
//+-----+
//| Expert tick function |
//+-----+
void OnTick()
{
//---Declarations
RefreshRates();
int iii, ii, buyi, selli, xi, ai, bi, ci, di;
int ticket1, ticket2, ticket3, ticket4, ticket5, ticket6, ticket7, ticket8, total;
double matrica[1000], x, a, b, c, d;
double gartley1, gartley2, gartley3, gartley4, gartley5, gartley6, gartley7, gartley8;
double agartley1, agartley2, agartley3, agartley4, agartley5, agartley6, agartley7, agartley8;
double bat1,bat2,bat3,bat4,bat5,bat6,bat7,bat8,abat1,abat2,abat3,abat4,abat5,abat6,abat7,abat8;
double fly1,fly2,fly3,fly4,fly5,fly6,fly7,fly8,afly1,afly2,afly3,afly4,afly5,afly6,afly7,afly8;
double
crab1,crab2,crab3,crab4,crab5,crab6,crab7,crab8,acrab1,acrab2,acrab3,acrab4,acrab5,acrab6,acrab7,acrab8;
//double xg, xf, xb, xc, xag, xaf, xab, xac;
//Gauname matrica is ZigZag indikatoriaus
for(iii=0;iii<1000;iii++)
{
matrica[iii]=iCustom(NULL,0,"ZigZag",InpDepth,InpDeviation,InpBackstep,0,iii);
}
//Ieskome minimumu ir maksimumu
//d=Close[0]; // paskutinis paterno taskas - uzsidariusios zvakes kaina.
for(ii=0;x==0;ii++)
{
if(matrica[ii]>0 && x==0 && a==0 && b==0 && c==0 && d==0)
{
d=matrica[ii];
di=ii;
}
else if(matrica[ii]>0 && x==0 && a==0 && b==0 && c==0 && d!=0)
{

```

```

    c=matrica[ii];
    ci=ii;
}
else if(matrica[ii]>0 && x==0 && a==0 && b==0 && c!=0 && d!=0) //Randame B taska
{
    b=matrica[ii];
    bi=ii;
}
else if(matrica[ii]>0 && x==0 && a==0 && b!=0 && c!=0 && d!=0) //Randame A taska
{
    a=matrica[ii];
    ai=ii;
}
else if(matrica[ii]>0.0 && x==0.0 && a!=0.0 && b!=0.0 && c!=0.0 && d!=0.0) //Randame X taska ir
skaiciuojame paternu koeficientus
{
    x=matrica[ii];
    xi=ii;
}
}
////////////////////////////////////
// Skaiciuojame paternu fibonaci lygius////////////////////////////////////
////////////////////////////////////
gartley1=a-((a-x)*0.618);////////////////////////////////////
gartley2=a-((a-x)*0.786);///
gartley3=b+((a-b)*0.382);///Gartley paterno fibbonaci lygiai////////
gartley4=b+((a-b)*0.886);///BULIU
gartley5=c-((c-b)*1.270);///
gartley6=c-((c-b)*1.618);////////////////////////////////////
gartley7=a-((a-x)*0.5);////////////////////////////////////
gartley8=a-((a-x)*0.786);///

agartley1=a+((x-a)*0.618);////////////////////////////////////
agartley2=a+((x-a)*0.786);///
agartley3=b-((b-a)*0.382);///Gartley paterno fibbonaci lygiai////////
agartley4=b-((b-a)*0.886);///LOKIU
agartley5=c+((b-c)*1.270);///
agartley6=c+((b-c)*1.618);////////////////////////////////////
agartley7=a+((x-a)*0.5);////////////////////////////////////
agartley8=a+((x-a)*0.786);///

bat1=a-((a-x)*0.382);////////////////////////////////////
bat2=a-((a-x)*0.5);///
bat3=b+((a-b)*0.382);///Bat paterno fibbonaci lygiai////////
bat4=b+((a-b)*0.886);///BULIU
bat5=c-((c-b)*1.618);///
bat6=c-((c-b)*2.618);////////////////////////////////////
bat7=a-((a-x)*0.786);////////////////////////////////////
bat8=a-((a-x)*0.886);///

abat1=a+((x-a)*0.382);////////////////////////////////////
abat2=a+((x-a)*0.5);///
abat3=b-((b-a)*0.382);///Bat paterno fibbonaci lygiai////////
abat4=b-((b-a)*0.886);///LOKIU
abat5=c+((b-c)*1.618);///

```

```

abat6=c+((b-c)*2.618);////////////////////////////////////
abat7=a+((x-a)*0.786);////////////////////////////////////
abat8=a+((x-a)*0.886);///

```

```

fly1=a-((a-x)*0.618);////////////////////////////////////
fly2=a-((a-x)*0.786);///
fly3=b+((a-b)*0.382);///Butterfly paterno fibbonaci lygiai////////
fly4=b+((a-b)*0.886);///BULIU
fly5=c-((c-b)*1.618);///
fly6=c-((c-b)*2.618);////////////////////////////////////
fly7=a-((a-x)*1.27);////////////////////////////////////
fly8=a-((a-x)*1.618);///

```

```

afly1=a+((x-a)*0.618);////////////////////////////////////
afly2=a+((x-a)*0.786);///
afly3=b-((b-a)*0.382);///Butterfly paterno fibbonaci lygiai////////
afly4=b-((b-a)*0.886);///LOKIU
afly5=c+((b-c)*1.618);///
afly6=c+((b-c)*2.618);////////////////////////////////////
afly7=a+((x-a)*1.27);////////////////////////////////////
afly8=a+((x-a)*1.618);///

```

```

crab1=a-((a-x)*0.382);////////////////////////////////////
crab2=a-((a-x)*0.618);///
crab3=b+((a-b)*0.382);///Crab paterno fibbonaci lygiai////////
crab4=b+((a-b)*0.886);///BULIU
crab5=c-((c-b)*2.224);///
crab6=c-((c-b)*3.618);////////////////////////////////////
crab7=a-((a-x)*1.5);////////////////////////////////////
crab8=a-((a-x)*1.618);///

```

```

acrab1=a+((x-a)*0.382);////////////////////////////////////
acrab2=a+((x-a)*0.618);///
acrab3=b-((b-a)*0.382);///Crab paterno fibbonaci lygiai////////
acrab4=b-((b-a)*0.886);///LOKIU
acrab5=c+((b-c)*2.224);///
acrab6=c+((b-c)*3.618);////////////////////////////////////
acrab7=a+((x-a)*1.5);////////////////////////////////////
acrab8=a+((x-a)*1.618);///
////////////////////////////////////

```

```

total=OrdersTotal();
if(total<10)
{
// no opened orders identified
if(AccountFreeMargin())<(1000*Lots)
{
Print("We have no money. Free Margin = ", AccountFreeMargin());
//return(0);
}
// check for long position (BUY) possibility
if(crab2<b && b<crab1 && crab3<c && c<crab4 && crab6<d && d<crab5 && crab8<d && d<crab7 &&
x!=xc) //&& Low[0]>Low[1] && Low[1]>Low[2])
{

```



```

xc=x; // tikrinama ar nera jau tokio pat treido
//Lots=suma/((c-d)/Point);
    string trikampis=IntegerToString(ii)+"CrabBULIU";
    string trikampis2=IntegerToString(ii)+"Crab2BULIU";
    ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],Low[xi],Time[ai],High[ai],Time[bi],Low[bi]);
    ObjectSet(trikampis,OBJPROP_COLOR,Green);
    ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],Low[bi],Time[ci],High[ci],Time[di],Close[di]);
    ObjectSet(trikampis2,OBJPROP_COLOR,Green);
//buyi=buyi+1;
Print("Perkam Crab");//+DoubleToString(x,3));
//ticket=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-
StopLoss*Point,Ask+TakeProfit*Point,"Gartley",16384,0,Green);
ticket1=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-sl*(c-d),tp*(c-d)+d,"Gartley",16384,0,Green);
if(ticket1>0)
{
    if(OrderSelect(ticket1,SELECT_BY_TICKET,MODE_TRADES)) Print("BUY order opened :
",OrderOpenPrice());
}
else Print("Error opening BUY order : ",GetLastError());
//return(0);
}
//////////
if(gartley2<b && b<gartley1 && gartley3<c && c<gartley4 && gartley6<d && d<gartley5 &&
gartley8<d && d<gartley7 && x!=xc)// && Low[0]>Low[1] && Low[1]>Low[2])
{
    xc=x;
//Lots=suma/((c-d)/Point);
    string trikampis=IntegerToString(ii)+"GartleyBULIU";
    string trikampis2=IntegerToString(ii)+"Gartley2BULIU";
    ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],Low[xi],Time[ai],High[ai],Time[bi],Low[bi]);
    ObjectSet(trikampis,OBJPROP_COLOR,Green);
    ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],Low[bi],Time[ci],High[ci],Time[di],Close[di]);
    ObjectSet(trikampis2,OBJPROP_COLOR,Green);
//buyi=buyi+1;
Print("Perkam Gartley");//+DoubleToString(x,3));
//ticket=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-
StopLoss*Point,Ask+TakeProfit*Point,"Gartley",16384,0,Green);
ticket2=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-sl*(c-d),tp*(c-d)+d,"Gartley",16385,0,Green);
if(ticket2>0)
{
    if(OrderSelect(ticket2,SELECT_BY_TICKET,MODE_TRADES)) Print("BUY order opened :
",OrderOpenPrice());
}
else Print("Error opening BUY order : ",GetLastError());
//return(0);
}
//////////
if(bat2<b && b<bat1 && bat3<c && c<bat4 && bat6<d && d<bat5 && bat8<d && d<bat7 && x!=xc)//
&& Low[0]>Low[1] && Low[1]>Low[2])
{
    xc=x;
//Lots=suma/((c-d)/Point);
    string trikampis=IntegerToString(ii)+"BatBULIU";
    string trikampis2=IntegerToString(ii)+"Bat2BULIU";
    ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],Low[xi],Time[ai],High[ai],Time[bi],Low[bi]);

```

```

ObjectSet(trikampis,OBJPROP_COLOR,Green);
ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],Low[bi],Time[ci],High[ci],Time[di],Close[di]);
ObjectSet(trikampis2,OBJPROP_COLOR,Green);
//buyi=buyi+1;
Print("Perkam Bat");//+DoubleToString(x,3));
//ticket=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-
StopLoss*Point,Ask+TakeProfit*Point,"Gartley",16384,0,Green);
ticket3=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-sl*(c-d),tp*(c-d)+d,"Bat",16386,0,Green);
if(ticket3>0)
{
if(OrderSelect(ticket3,SELECT_BY_TICKET,MODE_TRADES)) Print("BUY order opened :
",OrderOpenPrice());
}
else Print("Error opening BUY order : ",GetLastError());
//return(0);
}
//////////
if(fly2<b && b<fly1 && fly3<c && c<fly4 && fly6<d && d<fly5 && fly8<d && d<fly7 && x!=xc)//
&& Low[0]>Low[1] && Low[1]>Low[2])
{
xc=x;
//Lots=suma/((c-d)/Point);
string trikampis=IntegerToString(ii)+"ButterflyBULIU";
string trikampis2=IntegerToString(ii)+"Butterfly2BULIU";
ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],Low[xi],Time[ai],High[ai],Time[bi],Low[bi]);
ObjectSet(trikampis,OBJPROP_COLOR,Green);
ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],Low[bi],Time[ci],High[ci],Time[di],Close[di]);
ObjectSet(trikampis2,OBJPROP_COLOR,Green);
//buyi=buyi+1;
Print("Perkam Butterfly");//+DoubleToString(x,3));
//ticket=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-
StopLoss*Point,Ask+TakeProfit*Point,"Gartley",16384,0,Green);
ticket4=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-sl*(c-d),tp*(c-d)+d,"Fly",16387,0,Green);
if(ticket4>0)
{
if(OrderSelect(ticket4,SELECT_BY_TICKET,MODE_TRADES)) Print("BUY order opened :
",OrderOpenPrice());
}
else Print("Error opening BUY order : ",GetLastError());
//return(0);
}
//////////
// check for short position (SELL) possibility
if(agartley2>b && b>agartley1 && agartley3>c && c>agartley4 && agartley6>d && d>agartley5 &&
agartley8>d && d>agartley7 && x!=xag)// && Low[0]<Low[1] && Low[1]<Low[2])
{
xag=x;
//Lots=suma/((d-c)/Point);
selli=selli+1;
string trikampis=IntegerToString(ii)+"GartleyLOKIU";
string trikampis2=IntegerToString(ii)+"Gartley2LOKIU";
ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],High[xi],Time[ai],Low[ai],Time[bi],High[bi]);
ObjectSet(trikampis,OBJPROP_COLOR,Red);
ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],High[bi],Time[ci],Low[ci],Time[di],Close[di]);
ObjectSet(trikampis2,OBJPROP_COLOR,Red);

```

```

//ticket=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+StopLoss*Point,Bid-TakeProfit*Point,"macd
sample",16384,0,Red);
ticket5=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+s1*(d-c),d-tp*(d-c),"Gartley sell",16388,0,Red);
if(ticket5>0)
{
if(OrderSelect(ticket5,SELECT_BY_TICKET,MODE_TRADES)) Print("SELL order opened :
",OrderOpenPrice());
}
else Print("Error opening SELL order : ",GetLastError());
//return(0);
//return(0);
}
//////////
if(abat2>b && b>abat1 && abat3>c && c>abat4 && abat6>d && d>abat5 && abat8>d && d>abat7 &&
x!=xag)// && Low[0]<Low[1] && Low[1]<Low[2])
{
xag=x;
//Lots=suma/((d-c)/Point);
seli=seli+1;
string trikampis=IntegerToString(ii)+"BatLOKIU";
string trikampis2=IntegerToString(ii)+"Bat2LOKIU";
ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],High[xi],Time[ai],Low[ai],Time[bi],High[bi]);
ObjectSet(trikampis,OBJPROP_COLOR,Red);
ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],High[bi],Time[ci],Low[ci],Time[di],Close[di]);
ObjectSet(trikampis2,OBJPROP_COLOR,Red);
//ticket=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+StopLoss*Point,Bid-TakeProfit*Point,"macd
sample",16384,0,Red);
ticket6=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+s1*(d-c),d-tp*(d-c),"Bat sell",16389,0,Red);
if(ticket6>0)
{
if(OrderSelect(ticket6,SELECT_BY_TICKET,MODE_TRADES)) Print("SELL order opened :
",OrderOpenPrice());
}
else Print("Error opening SELL order : ",GetLastError());
//return(0);
//return(0);
}
//////////
if(afly2>b && b>afly1 && afly3>c && c>afly4 && afly6>d && d>afly5 && afly8>d && d>afly7 &&
x!=xag)// && Low[0]<Low[1] && Low[1]<Low[2])
{
xag=x;
//Lots=suma/((d-c)/Point);
seli=seli+1;
string trikampis=IntegerToString(ii)+"ButterflyLOKIU";
string trikampis2=IntegerToString(ii)+"Butterfly2LOKIU";
ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],High[xi],Time[ai],Low[ai],Time[bi],High[bi]);
ObjectSet(trikampis,OBJPROP_COLOR,Red);
ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],High[bi],Time[ci],Low[ci],Time[di],Close[di]);
ObjectSet(trikampis2,OBJPROP_COLOR,Red);
//ticket=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+StopLoss*Point,Bid-TakeProfit*Point,"macd
sample",16384,0,Red);
ticket7=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+s1*(d-c),d-tp*(d-c),"Butterfly sell",16390,0,Red);
if(ticket7>0)
{

```

```

        if(OrderSelect(ticket7,SELECT_BY_TICKET,MODE_TRADES)) Print("SELL order opened :
",OrderOpenPrice());
    }
    else Print("Error opening SELL order : ",GetLastError());
    //return(0);
//return(0);
}
//////////
if(acrab2>b && b>acrab1 && acrab3>c && c>acrab4 && acrab6>d && d>acrab5 && acrab8>d &&
d>acrab7 && x!=xag)// && Low[0]<Low[1] && Low[1]<Low[2])
{
    xag=x;
    //Lots=suma/((d-c)/Point);
    selli=selli+1;
    string trikampis=IntegerToString(ii)+"CrabLOKIU";
    string trikampis2=IntegerToString(ii)+"Crab2LOKIU";
    ObjectCreate(trikampis,OBJ_TRIANGLE,0,Time[xi],High[xi],Time[ai],Low[ai],Time[bi],High[bi]);
    ObjectSet(trikampis,OBJPROP_COLOR,Red);
    ObjectCreate(trikampis2,OBJ_TRIANGLE,0,Time[bi],High[bi],Time[ci],Low[ci],Time[di],Close[di]);
    ObjectSet(trikampis2,OBJPROP_COLOR,Red);
    //ticket=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+StopLoss*Point,Bid-TakeProfit*Point,"macd
sample",16384,0,Red);
    ticket8=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+s1*(d-c),d-tp*(d-c),"Crab sell",16391,0,Red);
    if(ticket8>0)
    {
        if(OrderSelect(ticket8,SELECT_BY_TICKET,MODE_TRADES)) Print("SELL order opened :
",OrderOpenPrice());
    }
    else Print("Error opening SELL order : ",GetLastError());
    //return(0);
//return(0);
}
}
}

```