

**VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS**

INFORMATIKOS KATEDRA

Verslo informacijos sistemų studijų programa
Kodas 62103S138

VILMA VASILIAUSKAITĖ

MAGISTRO BAIGIAMASIS DARBAS

**SPIEČIAUS INTELEKTO TAIKYMO FINANSŲ RINKOSE ANALIZĖ IR
OPTIMIZAVIMAS**

Kaunas 2008

**VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS**

INFORMATIKOS KATEDRA

VILMA VASILIAUSKAITĖ

MAGISTRO BAIGIAMASIS DARBAS

**SPIEČIAUS INTELEKTO TAIKYMO FINANSŲ RINKOSE ANALIZĖ IR
OPTIMIZAVIMAS**

Leidžiama ginti _____

Magistrantas _____
(parašas)

Darbo vadovas _____
(parašas)

(darbo vadovo mokslo laipsnis, mokslo
pedagoginis vardas, vardas, pavardė)

Darbo įteikimo data _____

Registracijos Nr. _____

TURINYS

SANTRUMPŲ SĄRAŠAS	4
PAVEIKSLŲ SĄRAŠAS	5
LENTELIŲ SĄRAŠAS	6
SANTRAUKA.....	7
ĮVADAS.....	8
1. FINANSINIŲ RINKŲ IR JOS PROGNOZAVIMUI NAUDOJAMŲ SPRENDIMŲ TEORINIAI ASPEKTAI	11
1.1. Akcijų rinkos apžvalga.....	11
1.1.1. S&P500 Indeksas.....	13
1.2. Finansinių rinkų prognozavimui naudojamų sprendimų samprata ir apibrėžimai	14
1.2.1. Metodai prognozavimui.....	15
1.2.2. Neuroninių tinklų ir evoliucinių algoritmų apžvalga	16
1.2.2.1. Neuroniniai tinklai	16
1.2.2.2. Biologinis neurono modelis	17
1.2.2.3. Dirbtinis neurono modelis.....	18
1.2.2.4. Neuroninių tinklų mokymas.....	20
1.2.2.5. Evoliuciniai algoritmai	21
1.2.2.6. Genetiniai algoritmai ir spiečiaus optimizavimas	22
1.2.2.6.1. Genetiniai algoritmai.....	23
1.2.2.6.2. Spiečiaus optimizavimas	23
1.2.2.6.3. Genetinių algoritmų ir spiečiaus optimizavimo palyginimas	25
2. SIŪLOMO SPRENDIMO PRIĖMIMO METODAS	27
2.1 Sprendimo priėmimo metodas.....	27
2.2 Sprendimo priėmimo algoritmas	31
2.3. Ekspertinės sistemos.....	38
2.3.1. GA/GP trading rule optimizer	38
2.3.2. Swarm 19.6.....	40
2.3.2.1. Sistemos architektūra	40
2.3.2.2. Konceptinis objekto modelis (ER diagramos)	41
2.3.2.3. Duomenų bazės projektas.....	42
3. EKSPERIMENTINIAI TYRIMAI.....	44
3.1. Duomenų rinkimas	44
3.2. Duomenų pasirinkimas	44
3.3. Duomenų apdorojimas ir analizė.....	45
3.4. Bandomieji tyrimai.....	46
3.5. Rezultatų patikimumo tikrinimas	50
3.6. Metodų panašumai ir skirtumai	504
IŠVADOS IR PASIŪLYMAI.....	55
LITERATŪRA	58
PRIEDAI.....	62

SANTRUMPŲ SĄRAŠAS

1. S&P500 - Standard & Poor's 500 indeksas;
2. GA – genetinis algoritmas;
3. PSO – spiečiaus optimizavimas;
4. VP – vertybiniai popieriai;
5. DB – duomenų bazė;
6. ER - esybių – ryšių diagrama;
7. Pav. – paveikslėlis;
8. P. – puslapis;
9. Žr. – žiūrėti;
10. Proc. – procentai;
11. m. – metai;
12. d. – diena.

PAVEIKSLŲ SĄRAŠAS

1 pav. S&P500 analizė	14
2 pav. Fundamentinių ir techninių veiksnių įtakos v_p portfelio formavimui įvertinimo modelis.....	15
3 Pav. Biologinis neuronas.....	17
4 Pav. Dirbtinio neuroninio tinklas	18
5 Pav. Tiesinės sistemos parametrų adaptavimas.....	21
6 pav. Sprendimo priėmimo idėja	29
7 pav. Sprendimo priėmimo algoritmas	32
8 pav. Architektūra.....	39
9 pav. Sistemos pagrindinis langas.....	39
10 pav. Sistemos architektūra.....	40
11 pav. ER diagrama	41
12 pav. Duomenų bazė	42
13 pav. Pelno įvertinimas	45
14 pav. Prisijungimo langas	46
15 pav. Duomenų atnaujinimo langas	47
16 pav. Portfelio sukūrimo langas.....	48
17 pav. Pagrindinis sistemos langas.....	49
18 pav. Prognozės langas	49
19 pav. Prognozavimo duomenų pasirinkimo procesas	50
20 pav. Istoriniai duomenys	50
21 pav. Istoriniai duomenys	51
22 pav. Pagrindinis sistemos langas.....	52
23 pav. Neuroniniai tinklai.....	53

LENTELIŲ SĄRAŠAS

1 lentelė Indeksai.....	12
2 lentelė GA ir PSO palyginimas	26
3 lentelė Algoritmų palyginimas	34
4 lentelė DB lentelių ir laukų aprašymas.....	42

SANTRAUKA

VASILIAUSKAITĖ, Vilma. (2008) *Analysis and Optimization of Swarm Intelligent in Financial Markets*. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 67 p.

SUMMARY

The theme of the Master's degree paper is "Analysis and Optimization of Swarm Intelligent in Financial Markets".

The object of the Master's degree paper is decision making method for financial markets, re neural network and swarm intelligence.

The length of this paper is 69 pages, there are 23 pictures and 4 tables in this paper.

IVADAS

Technologijų pasaulis keičiasi ir tobulėja dideliu tempu. Ir tai, kas buvo nauja vakar, šiandien jau yra sena. Kuriasi informacinė visuomenė, kurioje įsitvirtinti reikalingas nuolatinis mokymasis ir tobulėjimas. Jos dėka kuriamos valdymo sistemos, kurios padeda išvengti nereikalingų veiksmų, palengvina ir pagreitina valdymo procesus. Todėl šiais tobulėjančių technologijų laikais daugelis procesų tampa vis spartesni, paprastesni ir efektyvesni, sutaupomi ne tik laiko, bet ir materialiniai resursai, atsiranda naujos galimybės.

Modernios valdymo sistemos ir adaptyvus procesų valdymas bei intelektinės valdymo sistemos, dirbtinio intelekto technologijos naudojant kompiuterius, sparčiai populiarėja. Atliekant išsamius tyrimus dirbtinio intelekto srityje paaiškėjo, kad šių metodų panaudojimas galimas ne tik technologinių procesų valdymo srityje, bet ir finansų sektoriuje naudojant neraišką (fuzzy) logiką, dirbtinius neuroninius tinklus, ekspertines sistemas ir t.t.

Viena iš svarbiausių visuomenės ir technologijų progreso sąlyga - gerai sureguliuoti finansiniai procesai ir investicijų mechanizmai. Bet kuriai ekonomikai būtinos investicijos.

Taigi prekiaujant vertybiniais popieriais, svarbiausia yra priimti teisingą sprendimą: pirkti arba parduoti. Daugelis investuotojų prieš priimdami sprendimą atkreipia dėmesį į pasirinktos akcijos kainos kitimo grafiką ir vadovaujasi juo. Tačiau ne kiekvienas investuotojas galėtų tiksliai apibūdinti savo pasirinktą grafinį modelį.

Problemos aktualumas - Prognozuoti rinkas yra pakankamai sudėtinga, pastebimas žymus akcijų kursų svyravimas. Ženklius akcijų kursų pasikeitimai skaičiuojami ne per metus ar mėnesius, o dienomis ar net valandomis. Investitoriams, finansų analitikams finansinėse rinkose sunku dirbti. Spekuliavimas akcijomis aktyviose akcijų rinkose yra labai rizikingas, bet pelningas užsiėmimas. Pasiūlius sprendimo priėmimo metodą investavimo procesas techniniu požiūriu supaprastės ir nereikalaus didelių sąnaudų, bei gilių žinių, leis platesniam ratui žmonių įeiti į akcijų rinką.

Problema – Sudėtingas akcijų rinkų prognozavimas, kadangi pastebimas žymus akcijų kursų svyravimas, todėl rizikinga spekuliuoti akcijomis aktyviose akcijų rinkose.

Baigiamojo darbo objektas – sprendimo priėmimo metodas finansinių rinkų prognozėms atlikti, remiantis neuroniniais tinklais ir spiečiaus algoritmu.

Baigiamojo darbo tikslas – Spiečiaus intelekto taikymo finansų rinkose analizė ir optimizavimas.

Baigiamojo darbo uždaviniai:

1. Atlikti akcijų rinkos ir rinkų prognozavimui naudojamų metodų analizę;
2. Remiantis teorine medžiaga išnagrinėti neuroninių tinklų klasifikaciją, jų panaudojimo procesą;

3. Atlikti evoliucinių algoritmų apžvalgą;
4. Atlikti lyginamąją analizę pateikiant PSO ir genetinių algoritmų skirtumus;
5. Pasiūlyti sprendimo priėmimo metodą ir pateikti modelį, paremtą neuroninių tinklų pritaikomumu ir spiečiaus optimaliu algoritmu (PSO);
6. Optimizuoti pasiūlytą sprendimo priėmimo algoritmą;
7. Atlikti eksperimentinius tyrimus naudojant Swarm 19.6 ekspertinę sistemą sprendimo priėmimo metodui patikrinti;
8. Patvirtinti arba paneigti ginamuosius teiginius (darbinę tyrimo hipotezę).

Iškelta hipotezė: Remiantis algoritmo pritaikymo idėja, galima parinkti neuroninius tinklus ateities investicijų sprendimams, remiantis trijų geriausių rezultatų žadančių tinklų rezultatų vidurkiu. Remiantis eksperimentiniais tyrimais siekiama šią hipotezę patvirtinti arba paneigti.

Baigiamąjį darbą sudaro trys dalys: teorinė/analitinė dalis, siūlomo sprendimo dalis ir eksperimentinė dalis.

Teoriniame/analitiniame skyriuje pateikiama akcijų rinkos apžvalga bei baigiamajame darbe metodo realizavimui naudojamų S&P500 indekso duomenų analizė. Išanalizuojama mokslinėje literatūroje, empiriniuose tyrimuose aptariami modeliai, metodai, kurie įvairiais požiūriais tiria evoliucinius ir genetinius algoritmus, neuroninius tinklus, spiečiaus ir neuroninių tinklų panaudojimo galimybes finansinių rinkų prognozėms. Skyriaus gale pateikiama lyginamoji šių metodų analizė.

Siūlomo sprendimo metodikos skyriuje pateikiamas siūlomas sprendimo priėmimo metodas. Pateikiamas sprendimo priėmimui naudojamo pakoreguoto algoritmo detalus aprašymas, bei jo skirtumai su jau anksčiau pasiūlytu algoritmu. Išanalizuojama ekspertinė sistema, pasiūlyto sprendimo metodo pritaikomumui patikrinti, bei kitos sistemos, naudojamos finansinių rinkų prognozėms atlikti.

Eksperimentiniame skyriuje pristatyti pasiūlyto sprendimo metodo bandomieji tyrimai ir analizės ypatybės. Pateikta trumpa sistemos vartotojo instrukcija. Paskutiniuoju šio skyriaus dalyje suformuotos ir pateiktos gautų rezultatų išvados.

Eksperimentinis tyrimas bus vykdomas ir aprašomas darbe tokia seka:

1. Duomenų rinkimas;
2. Duomenų apdorojimas ir analizė;
3. Bandomieji tyrimai;
4. Rezultatai;
5. Išvados.

Darbe naudota įvairi literatūra, pateikiama literatūros sąrašė, t.y. knygos, internetiniuose puslapiuose pateikiama informacija, paskaitų konspektai ir autoriaus visų mokslo metų eigoje sukaupta informacija. Baigiamojo darbo apimtis – 62 puslapiai be priedų, su priedais – 69 puslapiai. Darbe pateiktos 4 lentelės, 23 paveikslėliai ir 5 priedai.

Baigiamajame darbe naudoti metodai:

1. Dar prieš rašant tyrimą buvo panaudotas mintinės analizės metodas, atliekamas abstrakčiomis sąvokomis ir sprendimais. (Reikėjo nuspręsti kokie terminai, sąvokos bus naudojami, koks tyrimas bus atliekamas);
2. Renkant medžiagą darbui, analizuojant duomenis naudoju duomenų analizės metodu;
3. Visuotinio pažinimo metodas buvo panaudotas darbo tikslo nustatymui, uždavinių formavimui, duomenų apibendrinimui;
4. Apibendrinimo metodas panaudotas medžiagai grupuoti ir išdėstyti pagal konkrečią jos reikšmę;
5. Palyginimo metodas (metodas panaudotas lyginant metodų charakteristikas, panašumus ir skirtumus);
6. Siekiant susikontretizuoti ties esminiais darbo bruožais, atskleisti esmę naudojamas abstrakcijos metodas;
7. Indukcijos metodas - juo remiantis buvo suformuluotos išvados;
8. Naudoti procesų modeliavimo grafiniai metodai diagramoms atvaizduoti.

Darbo teorinė ir praktinė reikšmė – atlikus finansinių rinkų prognozavimui naudojamų sprendimų analizę pastebėta, jog pirminė tyrėjų užduotis - sukurti tokį įrankį, kuris galėtų būti naudojamas vertybinių popierių rinkų svyravimų prognozei. Taigi buvo pasiūlyta sprendimo priėmimui naudoti metodu, pagrįstu neuroninių tinklų ir spiečiaus algoritmo panaudojimu finansinių rinkų prognozėm atlikti, prognozę atlikti remiantis trijų geriausių rezultatą žadančių tinklų rezultatų vidurkiu. Ateityje naudojant šį metodą gauti pelną bus lengviau.

Panaudojus Swarm 19.6 sistemą buvo atlikti bandomieji tyrimai šiam metodui patikrinti. Metodas pasiteisino ir galima teigti, jog ateities sprendimams galima naudoti pasiūlytą sprendimo priėmimo metodą.

Darbo sunkumai – atliekant pasiūlyto sprendimo metodo bandomuosius tyrimus su Swarm 19.6 sistema buvo sunku ją pritaikyti, dėl sistemos aprašymų metodinės medžiagos trūkumo. Sistema sukurta MATLAB ir MYSQL įrankių pagalba, tačiau programinis kodas yra labai sudėtingas. Įžvelgtas automatizavimo trūkumas, programa nepatogu naudoti. Programą dar būtų galima tobulinti.

1. FINANSINIŲ RINKŲ IR JOS PROGNOZAVIMUI NAUDOJAMŲ SPRENDIMŲ TEORINIAI ASPEKTAI

Šiame skyriuje pateikiama akcijų rinkos apžvalga bei baigiamajame darbe metodo realizavimui naudojamų S&P500 indekso duomenų analizė. Išanalizuojama mokslinėje literatūroje, empiriniuose tyrimuose aptariami modeliai, metodai, kurie įvairiais požiūriais tiria evoliucinius ir genetinius algoritmus, neuroninius tinklus, spiečiaus ir neuroninių tinklų panaudojimo galimybes finansinių rinkų prognozėms. Skyriaus gale pateikiama lyginamoji šių metodų analizė.

1.1. Akcijų rinkos apžvalga

Finansų rinkos yra svarbi ekonomikos dalis, per jas vysta finansavimo ir investavimo procesai. Rinka savo ruožtu įtakoja investuotojus ir jo sprendimus.

Pagal tyrimus, nuo 30 proc. iki 50 proc. atskirų akcijų kainos pokyčių sukelia bendrosios rinkos efektai. Gerai diversifikuoto vertybinių popierių portfelio pelningumo variantiškumą apie 90 proc. lemia rinka (Kancerevyčius, g. , 2004, p. 281).

Situacija pasaulio akcijų rinkose vis kinta, investuotojai stengiasi nuspėti rinką interpretuodami, atsižvelgdami į išorinius parametrus, tokius kaip ekonominiai rodikliai, viešieji opcionai, dabartinis politinis klimatas. “Jie laukia, kol pavojaus varpai apie rinkos bėdas nustos skambėti ir bus pranešta, kad nuosmukis baigėsi – prasideda naujas „bulių“, tai yra augimo, periodas.”¹ Tačiau dažnai situacija tampa aiški tik tada, kai jau per vėlu ką nors daryti.

Dažniausiai pasitaikančias investuotojų klaidos:

1. Investicinio tikslo neapibrėžtumas: investuotojai nežino, kodėl investuoja, kokio pelningumo tikisi, kuriam laikui įdeda į akcijas savo pinigus;
2. Investicijas investuotojai parduoda pasitarę su draugu ar pažįstamu arba remdamiesi žiniasklaidos pranešimais, o ne pasikonsultavę su profesionaliu investuotoju ar investicijų konsultantu;
3. Vertybinius popierius investuotojai parduoda tuomet, kai jų kaina pasiekia žemiausią ribą;
4. Investicinį portfelį jie sudaro neatsižvelgdami į ilgalaikes investicines tiesas – tai yra nepaiso diversifikavimo ir ilgalaikio turto paskirstymo galimybių;

¹ Beresnevičienė (2008) *Investavimo psichologijos labirintai*[Interaktyvus] prieiga per Internetą: < http://www.bs-b.lt/lt/straipsniai/article_detail.php?id=28393 >

5. Investuotojai investuoja tuomet, kai akcijų kainos aukštos – jie laukia, kol rinkos atsigaus (pvz.: kol rinka pakils 10–15 proc.), ir tik tuomet ryžtasi investuoti.¹

Kad būtų išvengtos šios dažnai pasitaikančios klaidos dauguma tyrėjų, kurie dirba su akcijų kursų prognozėmis, bando modeliuoti ir pritaikyti techninę analizę, naudotis tam tikslui sukurtomis sistemomis vertybinių popierių prognozei.

Finansinių rinkų prognozės idėja nėra nauja. Didelis dėmesys skiriamas sprendimo priėmimui, spėjimams ir prognozės technikai. Šita technika tampa vis svarbesnė daugelyje sričių. Tokių sistemų poreikis įtakoja įvairius faktorius ekonomikoje, greitus pakeitimus kompiuterių technologijose, ir taip toliau. Pradedant nuo mažų įmonių ar net individų ir baigiant su didelėmis gamyklomis, visi nori žinoti, kaip rinkos situacija ketina pasikeisti, kokios bus būsimos kainos ir taip toliau.

Galimybė numatyti šituos pakeitimus galėtų garantuoti pasisekimą verslo aplinkoje. Visos šitos svarstomos problemos įtakoja tyrėjus, kad ieškotų įvairių sprendimo priėmimo įrankių, kurie galėtų padėti pasiekti efektyvumą gamyboje, versle ir neabejotinai įgyti pelno. Verslo pasaulyje svarbu priimti teisingus sprendimus dėl stokos.

Kalbant apie akcijų rinką reikėtų paminėti, jog finansų rinkose naudojama begalės indeksų, kurie atspindi konkrečios rinkos dalies ar sektoriaus vertybinių popierių kitimo tendencijas. Indeksą galima traktuoti kaip tam tikro finansinių instrumentų portfelio su tam tikra struktūra vertės pokyčius, fiksuojamus per laiką. Norint jog indeksas būtų sėkmingas, jis turi būti naudingas ir apskaičiuotas pagal realius instrumentus.

Indeksus skaičiuoja daugelis pasaulyje žinomų organizacijų, fondų ir kitų kompanijų. Populiariausi indeksai pateikiami žemiau pirmoje lentelėje.

1 lentelė

Indeksai

Kompanija	Šalis	Apibūdinimas	Indeksai
Dow Jones & Company	JAV	Leidybos ir finansinės informacijos kompanija, skaičiuojanti net 3600 indeksų.	<ul style="list-style-type: none"> • Dow Jones Composite Average • Dow Jones Global Titans • Dow Jones Industrial Average (DJIA, „Dow Jones 30,“ arba dažnai tiesiog „the Dow“) • Dow Jones Transportation Average • Dow Jones Utility Average • Dow Jones U.S. Large Cap Growth • Dow Jones U.S. Large Cap Value • Dow Jones U.S. Small Cap Growth • Dow Jones U.S. Small Cap Value • Dow Jones Wilshire 5000 Total Market Index

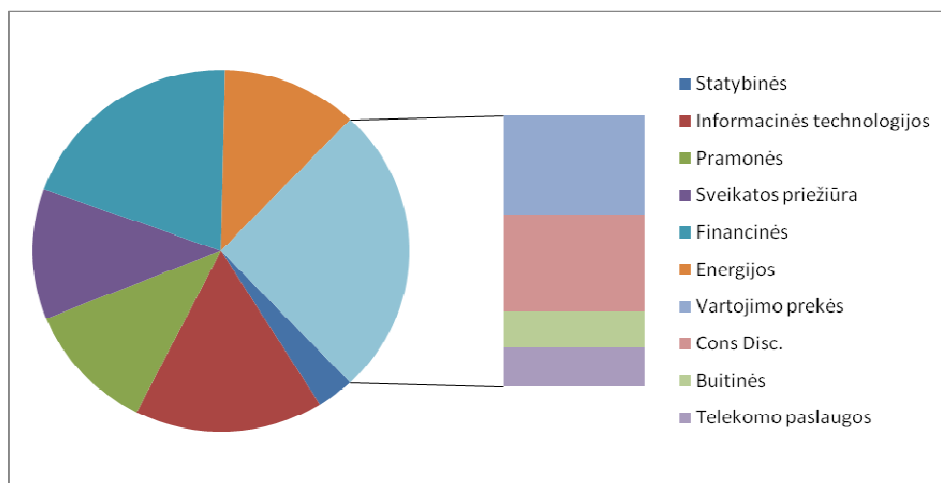
Kompanija	Šalis	Apibūdinimas	Indeksai
Standard & Poor's	JAV	Firma turinti daugiausiai finansinių analitikų - tai yra 1,250 žmonių, dirbančių visame pasaulyje ir atliekančių kreditų rizikos analizes. Didžiausia dalis investuotojų naudoja S&P 500 indeksą, negu kuri nors kita pasaulyje (Akcijų investicinę grąžą nustatyti panaudotas S&P 500 indeksas. Šį indeksą sudaro 500 JAV kompanijų, kurias atrenka "Standart&Poors" agentūra.	<ul style="list-style-type: none"> • S&P100 • S&P500 • S&P MidCap400 • S&P SmallCap600 • S&P900 • S&P1000 • S&P Composite 1500
NASDAQ	JAV	Nacionalinė vertybinių popierių dilerių asociacija. Nuo 1961 metų atsakinga už NASDAQ akcijų rinkos ir užbiržinės rinkos funkcionavimo priežiūrą.	
LITIN	Lietuva	Apimantis visas Oficialiojo sąrašo akcijas. (Lietuvos akcijų rinkoje buvo 5 oficialūs indeksai)	
LITIN - A	Lietuva	Apimantis visas Einamojo sąrašo akcijas. Šio indekso nuo 2001 metų vasario 1 d. birža nebeskaičiuoja.	
LITIN-G	Lietuva	Apimantis visas NVPB listinguojamas akcijas. Indekse dominuoja: Lietuvos Telekomas, Lietuvos Energija, Mažeikių nafta ir Lietuvos dujos.	
LITIN-10	Lietuva	Pradėtas skaičiuoti 1999 metais. Šis indeksas apima 10 likvidžiausių akcijų. Kad patektų į indeksą, matuojamą akcijos apyvarta ir sandorių skaičiaus per paskutines 52 dienas. Indeksas skaičiuojamas nepertraukiamos prekybos realiu laiku.	

Šaltinis: sudaryta autoriaus pagal KANCEREVYČIUS, G. Finansai ir investicijos, 2004, p. 45.

1.1.1. S&P500 Indeksas

Finansų rinkose naudojami indeksai, kurie atspindi konkrečios rinkos dalies ar sektoriaus vertybinių popierių kitimo tendencijas. Baigiamajame darbe metodo realizavimui bus pasirinkti būtent S&P500 indekso duomenys.

S&P500 įvertintas kaip vienas iš geriausių JAV vertybinių popierių rinkos matas, tai JAV vertybinių popierių rinkos dalis, užimanti maždaug 75 % JAV vertybinių popierių rinkos. Šis įžymus, pasaulinis indeksas apima 500 JAV ekonomikos kompanijų, kurios veikia 10-yje pramonės šakų.



Šaltinis: sudaryta autoriaus pagal Standard&Poor's (2008) S&P 500 [Interaktyvus] prieiga per Internetą: www2.standardandpoors.com/spf/pdf/index/500factsheet.pdf >

1 pav. S&P500 analizė

S&P 500 yra S&P JAV indeksų dalis, kuri gali būti panaudota kaip statybinis blokas portfelio statybai, kuris apskaičiuojamas atsižvelgiant į 500 JAV įmonių akcijų kainas. Į sąrašą atrenkamos tos įmonės, kurios geriausiai atitinka atstovaujamo sektoriaus rinkos dydį, likvidumą ir augimo tempus.

Remiantis Standard&Poor's² duomenimis S&P 500 indeksas pradėtas apskaičiuoti 1923 metais. Tuo metu jis buvo apskaičiuojamas atsižvelgiant į 223 įmonių akcijų kainas, tačiau 1957 metais įmonių skaičius pasipildė iki 500. Dabar S&P 500 apskaičiuojamas pagal konkretų metodą, t.y. akcijų vertės nustatymas, arba įvertinimas pagal kapitalizaciją. Šio metodo esmė: akcijų kainos dauginamos iš akcijų skaičiaus esančio apyvartoje ir visos taip gautos kapitalizacijos sumuojamos. Po to šis skaičius dalinamas iš bendros kapitalizacijos, buvusios nulinę indekso skaičiavimo dieną ir dauginamas iš nustatytos pradinės indekso reikšmės.

1.2. Finansinių rinkų prognozavimui naudojamų sprendimų samprata ir apibrėžimai

Prekiaujant vertybiniais popieriais, svarbiausia yra priimti teisingą sprendimą: pirkti arba parduoti. Todėl svarbiausia investuotojui pasirinkti tinkamą metodą prognozei atlikti. Keletą metodų, tokių kaip, techninės analizės ir fundamentalūs analizės metodai, dirbtinių neuroninių tinklų panaudojimas, evoliuciniai skaičiavimai, skirtų prognozavimui, pabandyčiau aptarti savo darbe, tolimesniuose skyriuose. Taip pat pateiksiu savąjį sprendimo priėmimo metodą.

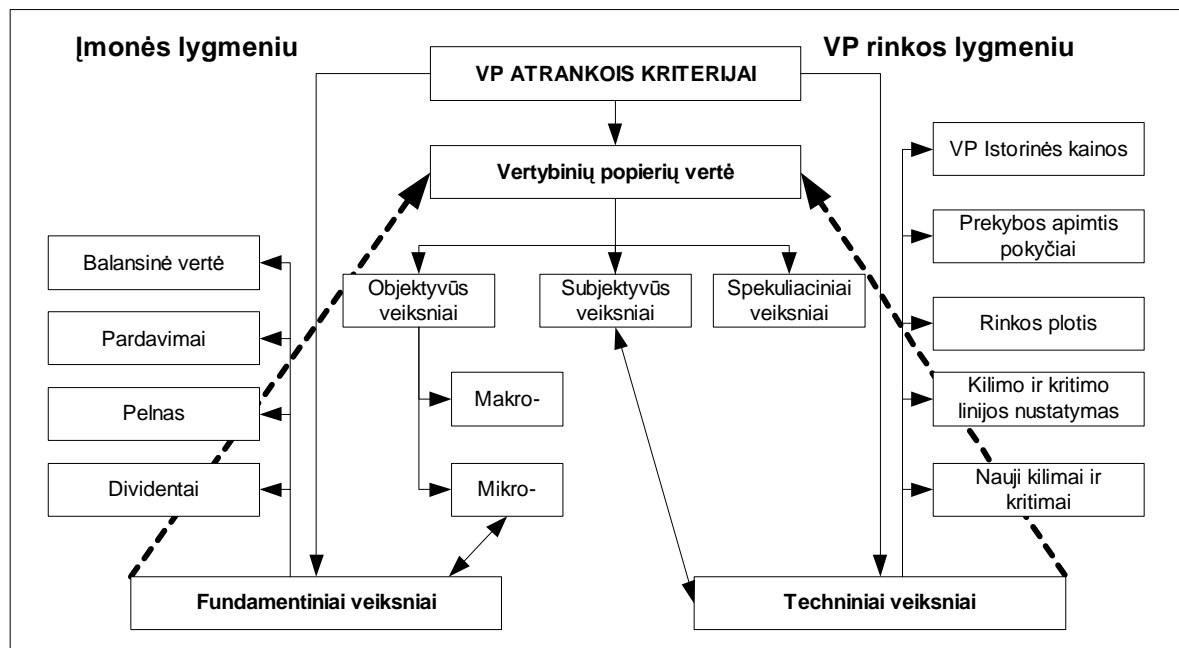
² Standard&Poor's (2008) S&P 500 [interaktyvus] prieiga per internetą: < www2.standardandpoors.com/ >

1.2.1. Metodai prognozavimui

Kyla daug diskusijų dėl finansinių rinkų prognozavimui skirtų sprendimų. Gausybė literatūros, kurioje mokslininkai siūlo naudoti vienus ar kitus sprendimus.

Dažniausiai rinkos analizei atlikti taikomi techninės analizės ir fundamentinės analizės metodai. Fundamentinės analizės metodas grindžiamas šalies / pasaulio makroekonominiais rodikliais ir įmonių finansiniais duomenimis. Išties apima rodiklius, susijusius su pačia įmone, t. y. dividendai, pardavimai, pelnas ir t. t.

Techninės analizės metodas pagrįstas būsimų finansinių kainų pranašavimu, naudojant tikrai rinkos kainų istorinius duomenis. Techninė analizė remiasi įvairiais indikatoriais, tačiau neatsižvelgia į makroekonominius, ūkio šakų ar atskirų įmonių (mikroekonomika) duomenis. (Blanchet-Scalliet Christophe, Diop Awa, Gibson Rajna, Talay Denis ir Tanr´e Etienne, 2004)



Šaltinis: CIBULSKIENĖ, D., GRIGALIŪNIENĖ, Ž. Fundamentinių ir techninių veiksnių įtaka vertybinių popierių portfelio formavimui, 2006, p. 69.

2 pav. Fundamentinių ir techninių veiksnių įtakos vertybinių popierių portfelio formavimui įvertinimo modelis

1.2.2. Neuroninių tinklų ir evoliucinių algoritmų apžvalga

Norint teisingai įvertinti finansines rinkas, bei sužinoti kokie pakitimai akcijų rinkose, nepakanka pritaikyti tradicinius įrankius prognozei atlikti. Todėl atliekant išsamius tyrimus paaiškėjo, jog dirbtino intelekto (artificial intelligent), dirbtinių neuroninių tinklų (artificial neural networks), ekspertinių sistemų ir t.t panaudojimas daro didelę įtaką prognozei ir investiciniam sprendimui. Šis darbas bei sprendimo sukūrimo modelis finansinėms rinkoms yra siejamas su dirbtiniais neuroniniais tinklais. Viena iš sprendimų priėmimo modelio dalių, atsakingų už modelio mokymą, yra dirbtiniai neuroniniai tinklai. Taip pat pasiūlytame sprendimų priėmimo modelyje didelis dėmesys suteikiamas ir evoliuciniams skaičiavimo algoritmams. Šiame skyriuje pastangos buvo priverstos pristatyti pagrindinius dirbtinius nervų tinklus taip pat kaip mintį apie evoliucinį skaičiavimo algoritmą.

1.2.2.1. Neuroniniai tinklai

Pirmasis neuroninių tinklų modeliavimas buvo pristatytas McCulloch ir Pitts 1943 metais pristatant "nervų veiklos" apskaičiavimo modelį. Įrodyta, kad naudojant dirbtinį neuroną su slenkstinėmis aktyvavimo funkcijomis galima sukurti įrenginį vykdančią bet kokią loginę funkciją. Prie šio modelio teko dirbti ir Neumann, Marvin Minsky, Frank Rosenblatt, ir daugeliui kitų mokslininkų.

1949 metais Pirmą kartą suformuluotas dirbtinių neuronų mokymo metodas, vėliau gavęs Hebb mokymo pavadinimą. Hebb pristatė klasikinę knygą „*the Organization of behavior*“. Jis rašė teigė jog neuronai yra atitinkamai sujungti pagal savo organizaciją ir „egzistuoja trajektorija sustiprinanti ryšius tarp neuronų“. Hebb teigė, kad smegenų ryšys nuolat kinta, taip kaip organizmas mokosi skirtingų funkcijų ir kad elementų (ląstelių) sanbūriai sukuriama pagal tokius kitimus.

1959 metais Rosenblatt pateikė Perceptrono matematinį modelį. Išleidžiamas pirmas solidus veikalas dirbtinių neuroninių tinklų srityje skirtas nuodugniam perceptronų tinklo analizei. Tais pačiais metais taip pat sukuriama adalina bei madalina - dirbtinių neuronų tinklai vėliau pradėti labai plačiai naudoti adaptyviame signalų apdorojime.

1964 - Grossberg (the centre of adaptive systems). Studijuojami psichologiniai bei biologiniai procesai vykstantys žmoguje jam apdorojant informaciją. Didžiausias dėmesys skiriamas save-organizuojantiems, save-stabilizuojantiems ir save-adaptuojantiems procesams, kurie įgalina betarpiškai priimti informaciją tuo metu kai sistema veikia realiame laike. Sukuriama minimalaus dydžio instar, outstar bei keleto pavyzdžių įsiminimo - srauto (angl. K. *Avalanche*)

tinklai. Vėliau sukuriama art1, art2, art3 modeliai, kurių pagrindu veikia nauji dirbtiniai neuroniniai tinklai.

1968 – Anderson. Sukuriama tiesinės asociatyviosios atminties (angl. K. *Linear associative memory*) bei smegenų būsenos dėžėje (angl. K. *Brain-state-in-a-box*) tinklai.

1971 – Kohonen. Sukuriama efektyvesnis optimalios tiesinės asociatyviosios atminties (angl. K. *Optimal linear associative memory*) tinklas. Pirmą kartą naudojama varžybų mokantis sąvoka ir jos pagrindu sukuriama vektorinio kvantavimo mokymo (lvq1, lvq2, lvq3) tinklai.

1977 - McClelland, Rumelhart. Dirbtinių neuroninių tinklų modeliai naudojami žmogaus smegenų psichologinių funkcijų tyrinėjimui. Išsivysto nauja nuoseklus paskirstyto apdorojimo (angl. K. *Parallel distributed processing*) sritis.

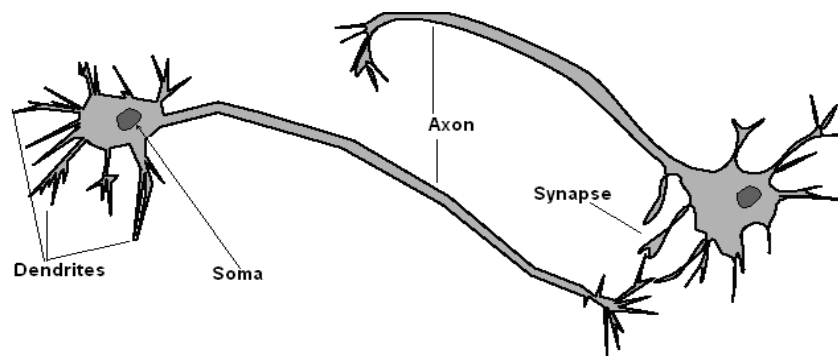
1982 - Hecht-Nielsen. Sukuriama kontroliuojamojo sklidimo (angl. K. *Counterpropagation*) tinklas. Vienas pirmųjų dirbtinių neuroninių tinklų teorijos įgyvendintas praktikoje - sukuriama mark iii neurokompiuteris. Įkuriama viena iš pirmųjų ANN gamybos kompanijų (hnc).

1990 - Hinton, Waibel, Lang. Sukuriama dinaminis neuronų tinklas (angl. K. *Time-delay neural network*), kuris apdoroja signalus įvertindamas ir jų buvusias reikšmes (Martavičius R., Urbanavičius V., Navakauskas D., Paulikas Š., 2008).

1.2.2.2. Biologinis neurono modelis

Žmogaus smegenys susideda iš daugiau nei 100 bilijonų neuronų, sujungtų vieni su kitais. Neuronas vidutiniškai turi apie 10 000 jungčių su kitais neuronais, jungčių skaičius yra kintantis dydis. Kiekvienas neuronas yra speciali ląstelė, galinti išduoti elektrocheminį signalą. Pavyzdys pateiktas 3 paveikslėlyje.

Pagrindinis biologinio neurono tinklo tikslas yra vystyti dirbtinius elementus hipotezių patvirtinimui apie biologines sistemas.



Šaltinis: ANDREAS Zell., *Simulation Neuronaler Netze*, Addison-Wesley, 1994, p. 23.

3 Pav. Biologinis neuronas

Biologinis neuronas gali būti sąlyginai suskirstytas į tris dalis: įėjimo, laidumo bei išėjimo sluoksnius. Įėjimo sluoksnyje esančios *dendritės* surenka iš aplinkinių neuronų „signalus“ į *somą* (ląstelės kūną), laidumo sluoksnyje signalai praėję *aksonos* kalnelį perduodami *aksona* toliau per išėjimo sluoksnyje esančias *aksonos* atšakas - *sinapsinius* ryšius (Simutis, R., 2007), (Engelbrecht, A. D., 2002).

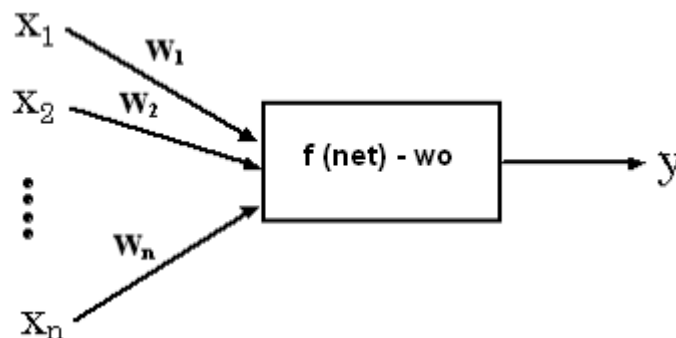
Sinapsėmis vadinamos aksono jungtys su kitų neuronų dendritais. Šios jungtys, perduodančios elektrocheminį signalą, yra įvairaus stiprumo. Kai sužadinama pakankamas kiekis neuronų, prijungtų prie neurono dendritų, neuronas sužadinamas, ir išilgai aksono jis išduoda elektrocheminį impulsą. Signalas per sinapses perduodamas kitiems neuronams, kurie taip pat gali būti sužadinami. Neuronas sužadinamas tik tuo atveju, kai bendras dendritais gautas signalas viršija tam tikrą lygį (sužadinimo slenkstį).

Priimamo signalo stiprumas ir neurono tikimybė būti sužadintam priklauso nuo sinapsių perdavimo efektyvumo. Šiuo metu manoma, kad mokymosi procesas vyksta keičiantis sinapsių efektyvumui.

Taigi, turint didžiulį skaičių visiškai paprastų elementų (kiekvienas elementas skaičiuoja svorinę įėjimo signalų sumą ir išduoda binarinį signalą, jei suminis signalas viršija tam tikrą lygį), įmanoma atlikti be galo sudėtingas užduotis. Šis paprastas pavyzdys toli gražu neparodo viso smegenų sudėtingumo, tačiau naudojant dirbtinius neuroninius tinklus, sudarytus iš daugelio analogiškų nesudėtingų elementų, galima pasiekti puikių rezultatų (Simutis, R., 2007), (Engelbrecht, A. D., 2002).

1.2.2.3. Dirbtinis neurono modelis

Dirbtinis neuroninis tinklas – tarpusavyje sujungtų dirbtinių neuronų grupė. Ši technologija mėgdžioja žmogaus galvos smegenų darbą – tiksliau neuronų veikimą.



Šaltinis: ENGELBRECHT, A. D., Computational intelligence (an introduction), 2002, p. 17.

4 Pav. Dirbtinio neuroninio tinklas

Dirbtinis neuronas - biologinio neurono abstrakcija, pagrindinis dirbtinių neuroninių tinklų komponentas.

Dirbtinis neuronas turi:

$x_1 \dots x_n$ - neurono įėjimus;

$w_1 \dots w_n$; w_0 – įėjimo svorius;

$f()$ – perdavimo funkcija;

Y – neurono išėjimą. Išėjimo reikšmė gaunama pagal formulę:

$$net = \sum_{i=1}^n x_i w_i$$

1. Neuronas gauna keletą įėjimo reikšmių. Tai gali būti viso neuroninio tinklo įėjimo reikšmės arba kitų tinklo neuronų išėjimo reikšmės. Kiekviena įėjimo jungtis turi savo perdavimo koeficientą (svorį), šie svoriai atitinka biologinio neurono sinapsių efektyvumą. Kiekvienas neuronas turi savo slenksčio reikšmę. Neurono sužadinimo reikšmė formuojama skaičiuojant svorinę įėjimo signalų sumą ir atimant slenksčio reikšmę.

2. Pagal sužadinimo signalą, naudojant neurono perdavimo funkciją skaičiuojama neurono išėjimo reikšmė (Simutis, R., 2007), (Engelbrecht, A. D., 2002).

Nygren, K., (2004) teigė, jog dirbtinių tinklų neuroniniame modelyje naudojamos aktyvavimo funkcijos išėjimo signalų apskaičiavimui. Aktyvavimo funkcija priima įėjimo signalus ir suskaičiuoja išėjimo. Gali būti naudojamos tam tikros aktyvavimo funkcijos:

1. Linear funkcija:

$$f(net - w_0) = \beta (net - w_0)$$

Linear funkcija pateikia linery moduliuotą išėjimą, kur β yra konstanta.

2. Step funkcija:

$$f(net - w_0) = \begin{cases} \beta_1 & \text{if } net \geq w_0 \\ \beta_2 & \text{if } net < w_0 \end{cases}$$

Ši funkcija pateikia dvi skaliarines reikšmes, kurios priklauso nuo w_0 .

3. Ramp funkcija

$$f(net - w_0) = \begin{cases} \beta & \text{if } net - w_0 \geq \beta \\ net - w_0 & \text{if } |net - w_0| < \beta \\ -\beta & \text{if } net - w_0 \leq -\beta \end{cases}$$

Ši funkcija jungia Step ir Linear funkcijas.

4. Sigmoid funkcija:

$$f(net - w_0) = \frac{1}{1 + e^{-\lambda(net - w_0)}}$$

Sigmoid funkcijos statusas pakeičiamas keičiant reikšmės λ parametru.

5. Hyperbolic funkcija:

$$f(\text{net} - w_o) = \frac{e^{\lambda(\text{net}-w_o)} - e^{-\lambda(\text{net}-w_o)}}{e^{\lambda(\text{net}-w_o)} + e^{-\lambda(\text{net}-w_o)}}$$

Hyperbolinio tangento išėjimas yra apibrėžiamas intervale (-1; 1).

6. Gaussian funkcija:

$$f(\text{net} - w_o) = e^{\lambda(\text{net}-w_o)^2 / \sigma^2}$$

σ^2 yra Gaussian distribucijos laipsnis. (Engelbrecht, A. D., 2002)

1.2.2.4. Neuroninių tinklų mokymas

Neuroninio tinklo apmokymo tikslas nėra tikslai atstovauti mokymo duomenų imtį. Mokant siekiama įvertinti ir sudaryti statistikinį proceso modelį, pagal kurį, labiausiai tikėtina, buvo gauti mokymo duomenys. Siekiama sudaryti proceso modelį, pagal kurį generuoti duomenys savo savybėmis būtų kiek įmanoma labiau panašūs į mokymo duomenis. Neuroninis tinklas pagal mokymo imties duomenis turi “išmokti” ir padaryti išvadas apie visus galimus ir nematytus duomenis.

Skiriami du neuroninių tinklų mokymo tipai – **prižiūrimas ir neprižiūrimas**.

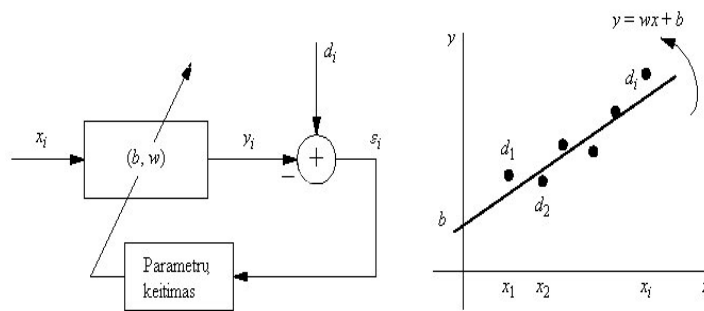
Prižiūrimam mokymui reikalingas išorinis mokytojas, valdantis mokymosi procesą ir teikiantis informaciją. Neuroninis tinklas apmokomas naudojant mokymo imtį. Mokyme su mokytoju mokymo imtis sudaroma iš įėjimo reikšmių kartu su atitinkamomis išėjimo (užduoties) reikšmėmis. Jei tinklas tinkamai apmokytas, jis gali modeliuoti funkciją, siejančią įėjimo bei išėjimo kintamuosius. Tokiu būdu, neuroninis tinklas pagal įėjimo reikšmes gali išduoti nežinomas išėjimo reikšmes. Svarbus veiksnys – mokymo duomenų aibė, kuri turi būti suprantama ir privalo aprėpti visas praktines tinklo taikymo sritis.

Neprižiūrimas mokymas neturi išorinio mokytojo. Remdamasi vidiniais kriterijais ir tinklo informacija, sistema pati save turi suderinti.

Tokiems neuroniniams tinklams pateikiami tik įvesčių pavyzdžiai, o sistema pati pagal požymius turi suklasifikuoti įvestis. Neprižiūrimo mokymo pavyzdys – kohoneno tinklai.

Dirbtinio neuroninio tinklo mokymo procedūra vadinama *mokymo algoritmu*. Tai funkcija, kuri modifikuoja tinklo sinapsinius svorius taip, kad tinklas galėtų pasiekti norimą tikslą.

Neuroninio tinklo kūrimas grindžiamas mokymu ar adaptacija



Šaltinis: SIMUTIS, R. Intelektualios sistemos finansų rinkose, paskaitų konspektai, 2007.

5 Pav. Tiesinės sistemos parametru adaptavimas

Ši tiesinė sistema turi tik du adaptuojamus parametrus b ir w . Sakykime, sistemos uždavinys – pagal x reikšmę išmokti prognozuoti, kokia bus d reikšmė. Mokymo tikslas yra pritaikyti sistemos parametrus (reikšmes b ir w) taip, kad minimizuoti skirtumus tarp sistemos išėjimo reikšmės y_i ir norimo atsako d_i .

Sistemos daroma klaida naudojama sistemos parametru keitimui. Suskaičiuota klaida

$\epsilon_i = d_i - y_i = d_i - (wx_i + b)$ gražinama sistemai, ir netiesiogiai vėl įtakoja išėjimą, keisdama parametrus. Pradžioje parametrai būna netinkami, ir sistema daro dideles klaidas. Mokymo metu parametrai derinami taip, kad klaida mažėtų. Sistema “susipažindama” su duomenimis suranda geriausią parametru rinkinį. Adaptyvios sistemos pakankamai sudėtingos. Jos turi vykdyti reikalingas funkcijas, be to, jose papildomai naudojamos posistemės parametru keitimui. Reikia pastebėti, jog pasikeitus duomenims ši metodika leidžia modifikuoti sistemos parametrus optimaliam uždavinio sprendimui. Ta pati sistema gali būti taikoma įvairių problemų sprendimui (Cichocki, A., Unbehauen, R., 1993), (Simutis, R., 2007).

1.2.2.5. Evoliuciniai algoritmai

Evoliucines strategijas išvystė Rechenberg (1973), naudodamas individų parinkimą, mutacijas ir populiaciją sudarytą iš vieno individo. H. Shwefel (1981) panaudojo kryžminimą ir populiacijas sudarytas iš didesnio individų kiekio. Taip pat evoliucines strategijas sulygino su kitomis, labiau tradicinėmis optimizavimo technologijomis.

Pasak Engelbrecht, A. D., (2002) evoliucinis skaičiavimas turi tikslą, tokį kaip gamtos evoliucijos modelis, pagrindinė sąvoka yra prisitaikytojų išlikimas: silpnieji miršta. Gamtos evoliucijoje išlikimas pasiekiamas per dauginimąsi. Palikuoniai ir dauginimasis priklauso nuo dviejų tėvų (kartais daugiau negu dviejų), daug įtakos turi genetinės abiejų tėvų medžiagos – tikėtina jog paveldės kiekvieno tėvo geriausius bruožus. Tie individai, kuriems būdingi blogi bruožai yra silpni ir jie praranda norą kovoti dėl išlikimo.

Evoliuciniame skaičiavime modeliuojama individų populiacija, nurodoma kaip *chromosoma*. Chromosoma apibūdina individų bruožus populiacijoje. Kiekvienas bruožas vadinamas *genu*. Geno reikšmė vadinama *aleliniu genu*. Individas sukuriamas sujungiant tėvų dalis, šis procesas vadinamas paveldėjimu.. Kiekvienas individas gali patirti mutaciją, kuri keičia kai kurias chromosomos aleles. Individų išlikimo tvirtumas matuojamas naudojant *fitness* funkciją, kuri atspindi problemos tikslus jai spręsti. Po kiekvieno kūrimo, individai gali būti sunaikinami arba gali išgyventi iki kito kūrimo (vadinamo *elitarizmu*).

Skirtingos Evoliucinių algoritmų išvystytos klasės:

- Genetinis algoritmas kuris modeliuoja genetinę evoliuciją.
- Genetinis programavimas, paremtas genetiniais algoritmais, bet individai yra programos (pristatomos kaip medžiai)

- Evoliucinis programavimas, paremtas individų panaudojimu, kurie savo genomu yra susiję su sprendimo priėmimu. Individų parinkimas yra konkurencija pagrįsta tikimybinė funkcija. Naujų individų sukuriama tiek pat kiek ir buvo senų. Vėliau geriausi atrenkami iš bendro naujų ir senųjų individų sąrašo ir patenka į sekančią kartą. Naujas individas sukuriamas iš senojo pritaikant jam mutaciją. Mutacijos operacijos tipas labai priklauso nuo uždavinio specifikos ir panašiai kaip evoliuciniuose skaičiavimuose dažnai yra pritaikantis (Fogel, L. J., Owens, A. J., Walsh, M. J., 1966).

- Evoliucijos strategijos
- Diferencinė evoliucija, kuri yra panaši į genetinius algoritmus.
- Kultūrinė evoliucija, modeliuoja populiacijos kultūros evoliuciją.
- Ko – evoliucija.

1.2.2.6. Genetiniai algoritmai ir spiečiaus optimizavimas

Pasak Yuhui SHI ir Eberhart, R. C. (1998) Evoliucinis skaičiavimas apima kelias paradigmas:

- Genetinis algoritmas. Paieškos technika naudojama kompiuterių moksluose rasti apytikrius sprendimus kombinatorinėms optimizavimo problemoms. Genetinis algoritmas yra populiacija, paremta algoritmu, kuris taiko operatorių atrinkimą ir susijungimą norint sukurti naują pavyzdį ieškomoje erdvėje.

- Spiečiaus intelektas (Swarm Intelligence). Tai yra dirbtinio intelekto technika, paremta kolektyvine elgsena decentralizacijos, saviorganizacijos sistemose.

1.2.2.6.1. Genetiniai algoritmai

Studijuojant ląstelių mechanizmą ir atliekant tyrimus, kuriuos vykdė John Holland buvo pradėti naudoti genetiniai algoritmai. 1975 metais John Holland parašė publikaciją *Adaptacijos Natūralioje ir Dirbtinėje Sistemoje*. Tai buvo jo pirmasis pasiekimas rašant apie genetinius algoritmus (Holland, J., 1975).

Genetinių algoritmų vystymosi procedūra prasideda chromosomų populiacijos generacija. Paprastai tokia populiacija yra atsitiktinė. Chromosomos, kurios pateikia geresnį sprendimą planuojamai problemai (vadinamos "geros" chromosomos) turi daugiau galimybių išgyventi ir "veistis". Chromosomas, kurios pateikia blogesnius sprendimus, vadinamos "blogomis" chromosomomis, ir jų galimybė išlikti yra labai maža ar net apskritai neegzistuoja. Sprendimo "sėkmė" yra tipiškai apibrėžiama atsižvelgiant į populiaciją (Whitley, D., 1994).

Genetiniai algoritmai remiasi biologijos žiniomis apie gyvybės evoliuciją. Naudojanti gamtoje egzistuojančius gyvybės evoliucinius mechanizmus: paveldėjimą, mutaciją, natūraliąją atranką.

Genetiniai algoritmai yra metodas analizuoti duomenis, jų dėka galima rasti apytikslių užduoties sprendimą. Sprendimas randamas naudojant evoliucinį ciklą, veikiantį gamtoje. Genetiniai algoritmai panaudojimui ieškant užduoties sprendimo reikalingi du pagrindiniai elementai: tinkama pradinė duomenų struktūra ir atrankos kriterijus. algoritmui paprastai reikia jog uždavinio sprendimas galėtų būti pateikiamas kaip duomenų struktūra, kad būtų galima nesunkiai kurti šios struktūros pakeistas kopijas. antrasis elementas tam tikras atrankos metodas – funkcija, aprašanti atrankos kriterijų. jos dėka galima taikyti kiekybinius atrankos kriterijus, atrenkant geresnius ir blogesnius sprendinius (Fentress, Sam W., 2005).

1.2.2.6.2. Spiečiaus optimizavimas

Spiečiaus optimizavimas (PSO) yra vienas iš spiečiaus intelekto (swarm intelligence) algoritmų, kuris buvo išvystytas James Kennedy ir Russell Eberhart 1990 metų viduryje (Hassan, R., Cohanim, B., Weck, O., 2004).

PSO yra visuotinis optimizavimo būdas, sumodeliuotas pagal paukščio būrių socialinę elgseną. Tai yra dirbtinio intelekto technika, paremta kolektyvine elgsena decentralizacijos, saviorganizacijos sistemose. Siejasi su dirbtiniu intelektu ir paukščio būriavimusi bei žuvies plaukiojimu būriais ir konkrečiai su spiečiaus teorija. PSO yra populiacija paremta ieškojimo procedūra, kur individai, nurodomi kaip dalelės sugrupuotos į spiečių. Kiekviena dalelė spiečiuje simbolizuoja kandidato sprendimą optimizavimo problemai. PSO sistemoje kiekviena dalelė

„skraido“ per multidimensines paieškos erdves, nustatomos jos pozicijos ieškojimo erdvėje priklausomai nuo kiekvienos patirties ir kaimynų dalelių (Engelbrecht, A. D., 2002, P. 66).

Miller, R., (2007) analizuodamas spiečių elgseną, pasitelkia fizikos kryptimi, vadinama - statistine mechanika, idėjinis aparatas. Tai tarsi paprastas paukščių judėjimo būriuose modelis, pasitelkiant 3 esmines taisykles:

1. stenkis nesusidurti su savo kaimynais;
2. stenkis laikytis kaimynų kuo arčiau;
3. skrisk ten, kur skrenda dauguma.

James Kennedy ir Russell Eberhart (1995) savo straipsniuose pristato netiesinių funkcijų optimizavimo sąvoką naudojant PSO metodologiją. Jie teigia, jog optimizavimo metodas buvo atrastas iš supaprastinto socialinio modelio modeliavimų. Apibūdina paradigmos testavimą ir pritaikomumą įskaitant ir netiesinės funkcijos optimizavimą ir neuroninių tinklų mokymą.

PSO pasak autorių susidaro labai paprastai ir paradigmos gali būti įgyvendintos pritaikius keletą kompiuterinio kodo eilučių. Reikalingos tik primityvios matematinės operacijos.

PSO esmę galima apibūdinti susiejant dvi pagrindines komponentų metodologijas. Jos siejasi su dirbtiniu intelektu ir paukščio būriavimusi bei žuvies plaukiojimu būriais ir konkrečiai su spiečiaus teorija. Tai taip pat susiję su evoliuciniu skaičiavimu, genetiniais algoritmais ir evoliuciniu programavimu.

Reynolds ir Heppner ir Grenander (1990) pristatė paukščio būriavimosi modeliavimus. Reynolds sudomino paukščio būriavimosi choreografijos estetika, zoologas Heppner sudomino atradamas pagrindines taisykles, kurios suteikė galimybę dideliame skaičiui paukščių būriuotis sinchroniškai, daug kartų netikėtai, staigiai keičiant kryptį, išsiskaidant ir persigrupuojant ir t.t. Abu mokslininkai turėjo nuovoką, jog lokaliniai procesai, tokie kaip šie, sumodeliuoti pagal tinklinį automata, gali sudaryti nenumatomą paukščio socialinės elgsenos grupės dinamiką. Abu modeliai sunkiai pasiklovi tarpasmenine distancijų manipuliacija. Tai yra būriavimosi elgsenos sinchronija, paukščių ketinimas išsilaikyti optimalioje distancijoje tarp savęs ir savo kaimynų.

Tai neatrodo per didelis logikos šuolis manyti, kad kai kurios taisyklės sukelia gyvūnų socialinę elgseną, įskaitant avis, galvijus ir būrius. Kaip sociobiologas Wilson, E., (1975) rašė minėdamas žuvų plaukiojimą būriais „mažiausiai teorijoje, individualūs būrių nariai gali pasimokyti iš kitų visų būrių narių atradimų ir ankstesnės patirties ieškant maisto.“ Taigi šis sakinytis reiškia socialinį informacijos pasidalinimą tarp pasiūlymų evoliuciniam pranašumui: ši hipotezė buvo pagrindinis principas PSO vystyme.

PSO yra paprastas algoritmas, efektyvus optimalioms plačios sferos funkcijoms. Jis apžvelgiamas kaip vidutinio lygio dirbtinio intelekto forma ar biologiškai pateiktas algoritmas,

užgrobiant vietą gamtoje, tarp vystymosi tyrinėjimų, kuris reikalauja amžinybės ir neuroninio proceso, kuris įvyksta milisekundžių tikslumu. Socialinis optimizavimas yra paprasta patirtis. Jo ryšiai su dirbtiniu intelektu ir PSO turi akivaizdžius ryšius su evoliuciniu skaičiavimu. Abstrakčiai, tai atrodo kaip apgavystė tarp genetinių algoritmų ir evoliucinio programavimo. Tai priklauso nuo stochastinių procesų, tokių kaip evoliucinis programavimas. *Pbest* ir *gbest* reikšmių pritaikymas pagal PSO yra paprastai *perėjimo* operacijai panaudojant genetinius algoritmus. Naudojama *fitness* sąvoka, įvykdant visas evoliucinio skaičiavimo paradigmas.

Pbest ir *gbest* reikšmės ir jų didėjimas yra ypač svarbios. Abstrakčiai *pbest* yra panaši į autobiografinę atmintį, kaip kiekvienas individas prisimena savo patirtį ir spartos suderinimas asocijuojasi su *pbest* vadinama „paprasta nostalgija“, tokiu būdu individas turi tendenciją grįžti į vietą, kuri daugiausiai patenkino jį praeityje. Kitu atveju *gbest* yra abstrakčiai panašus į žinias ar grupės normą ar standartą, kuriuos individualiai stengiamasi pasiekti (Kennedy, J., ir Eberhart, R., 1995).

1.2.2.6.3. Genetinių algoritmų ir spiečiaus optimizavimo palyginimas

PSO susijęs su evoliuciniu skaičiavimu ir „dirbtiniu gyvenimu“. Bendrai tariant tai yra tarp genetinių algoritmų ir evoliucinio programavimo. (J. Kennedy ir W..M. Spears, 2003).

PSO dalelė yra analogiška genetinio algoritmo (GA) populiacijos nariui (chromosomai). Taip pat, kaip ir genetinio algoritmo chromosoma, kiekviena spiečiaus dalelė įtakoja problemos sprendimą. PSO sistemoje kiekviena dalelė „skraido“ per multidimensines paieškos erdves, taip nustatomos jos pozicijos ieškojimo erdvėje priklausomai nuo kiekvienos patirties ir kaimynų dalelių (Engelbrecht, A. D., 2002).

Čia mes iškeliamė hipotezę, kad finansinėje rinkoje sprendimo priėmimui, iš visų kitų evoliucinių skaičiavimo paradigmu, PSO algoritmas yra tinkamiausias, kadangi PSO jis yra vienintelis evoliucinis algoritmas, kuris neapjungia natūraliosios atrankos, kuriai būdingas kai kurių populiacijos narių pašalinimas. Pasiūlytame sprendimo priėmimo modelyje šis PSO mokymo proceso bruožas yra labai svarbus, kadangi dirbama su dinamiškai besikeičiančiomis aplinkomis ir netgi „blogos“ dalelės gali duoti teisingą sprendimą. Genetinio algoritmo atveju, kuris yra panašiausias į PSO, mokymasis paremtas paveldėjimu ir mutacija. Daugiau GA ir PSO skirtumų aprašyta antrojoje lentelėje.

GA ir PSO palyginimas

Parametrai / Funkcijos		GA	PSO
Istorija		John Holland, 1975m.	James Kennedy ir Russell Eberhart 1990 m.
Bruožai	Populiacija	m individų sprendimų grupė (pvz.: m = 100); individų populiacija, nurodoma kaip chromosoma, kuri apibūdina individų bruožus populiacijoje. O kiekvienas bruožas vadinamas genu.	individų sprendimų grupė, susideda iš m dalelių (pvz.: m = 20). Populiacija paremta ieškojimo procedūra, kur individai, nurodomi kaip dalelės sugrupuotos į spiečių. Kiekviena dalelė spiečiuje simbolizuoja kandidato sprendimą optimizavimo problema.
	Fitness funkcija	kiekvieno individo sprendimo efektyvumo įvertinimas	tam tikro individo sprendimo efektyvumo įvertinimas per gyvavimo laikotarpį.
	Įvertinimo trukmė	Kiekvieno individo sprendimo įvertinimas per generaciją. Įvertinimo trukmės laikas labai skiriasi, jis priklauso nuo eksperimento scenarijaus: pavyzdžiui, tiesiog skaičiuojant fitness funkciją pritaikius kompiuterinę sistemą arba gali apimti bandomąjį periodą (pavyzdžiui, įvertinant kontrolės parametrus)	Kiekvieno individo sprendimo įvertinimas per vieną tarpsnį.
	Gyvavimo trukmė	Prisitaikiusieji išlieka, o silpnieji miršta.	
	Principai	paveldėjimas, mutacija, natūralioji atranka.	imitavimas, įvertinimas, palyginimas
Individuali atmintis		Ne	Taip
Individualios funkcijos		Mutacija	Asmeniškai geriausiai pozicija istorijoje, greita inercija
Socialinės funkcijos		Paveldėjimas	Priklausomumas nuo kaimynų ir kiekvienos dalelės patirties.

Šaltinis: sudaryta autoriaus

2. SIŪLOMO SPRENDIMO PRIĖMIMO METODAS

Šiame skyriuje bus pateikiamas siūlomas sprendimo priėmimo metodas. Pateikiamas sprendimo priėmimui naudojamo pakoreguoto algoritmo detalus aprašymas, bei jo skirtumai su jau anksčiau pasiūlytu algoritmu. Išanalizuojama ekspertinė sistema, pasiūlyto sprendimo metodo pritaikomumui patikrinti, bei kitos sistemos, naudojamos finansinių rinkų prognozėms atlikti.

2.1 Sprendimo priėmimo metodas

Prekiaujant vertybiniais popieriais, svarbiausia priimti teisingą sprendimą: pirkti arba parduoti. Norint palengvinti investuotojui priimti sprendimą, siūloma naudotis sprendimo priėmimo metodu, *pagrįstu neuroninių tinklų ir spiečiaus algoritmo panaudojimu finansinių rinkų prognozėm atlikti, prognozę atlikti remiantis trijų geriausių rezultatų žadančių tinklų rezultatų vidurkiu.*

Neuroninių tinklų panaudojimas grindžiamas tuo, kad neuroniniai tinklai apsimokymo procese turi galimybę įvertinti ryšius tarp įvairių rinkos veiksnių. Jų privalumas, kad čia ryšiai tarp įvairių faktorių nėra fiksuojami a priori bet formuojami mokymo metu pasitelkus eksperimentinius duomenis. Taigi jų išėjimai yra dalinai apsaugoti nuo "žmogiškojo veiksnio", kai gaunama ne tai kas slypi duomenyse, o tai ko trokštama.

Spiečiaus intelekto panaudojimas pagrįstas tuo, kad finansinėje rinkoje sprendimo priėmimui, iš visų kitų evoliucinių skaičiavimo paradigmu, PSO algoritmas yra tinkamiausias, kadangi PSO jis yra vienintelis evoliucinis algoritmas, kuris neapjungia natūraliosios atrankos, kuriai būdingas kai kurių populiacijos narių pašalinimas. Pasiūlytame sprendimo priėmimo modelyje šis PSO mokymo proceso bruožas yra labai svarbus, kadangi dirbama su dinamiškai besikeičiančiomis aplinkomis ir netgi „blogos“ dalelės gali duoti teisingą sprendimą. Daugiau skaityti skyriuje “žr. 1.2.2.6.3. PSO ir GA palyginimas”.

Priešingai nei kiti tyrėjai sprendimo priėmimui siūloma remtis šių kelių įrankių pagalba, būtent neuroninių tinklų ir PSO algoritmu. Svarbiausias šio algoritmo tikslas - gali padėti priimti sprendimą aplinkose su pastoviai besikeičiančiais parametrais.

Sprendimo priėmimo metodas sudarytas remiantis dr. Jovitos Nenortaitės disertacija. Savo darbe ji teigė, jog galutinis sprendimas "pirkti“, “parduoti“ ar visgi “laikyti“ priimamas remiantis neuroninio tinklo "global best" (geriausia) rekomendacija. Atliekamas palyginimas tarp neuroninių tinklų ir pasirenkamas geriausia rezultatą žadantis tinklas. Šis tinklas naudojamas tolimesniems bandomiesiems tyrimams (kitos dienos pelno skaičiavimui). Kai žinomas "global best" (geriausias)

tinklas, atliekamas kitų neuroninių tinklų mokymas. Neuroniniai tinklai apmokami keičiant svorius. Jų svorių perskaičiavimas atliekamas pagal šią lygtį:

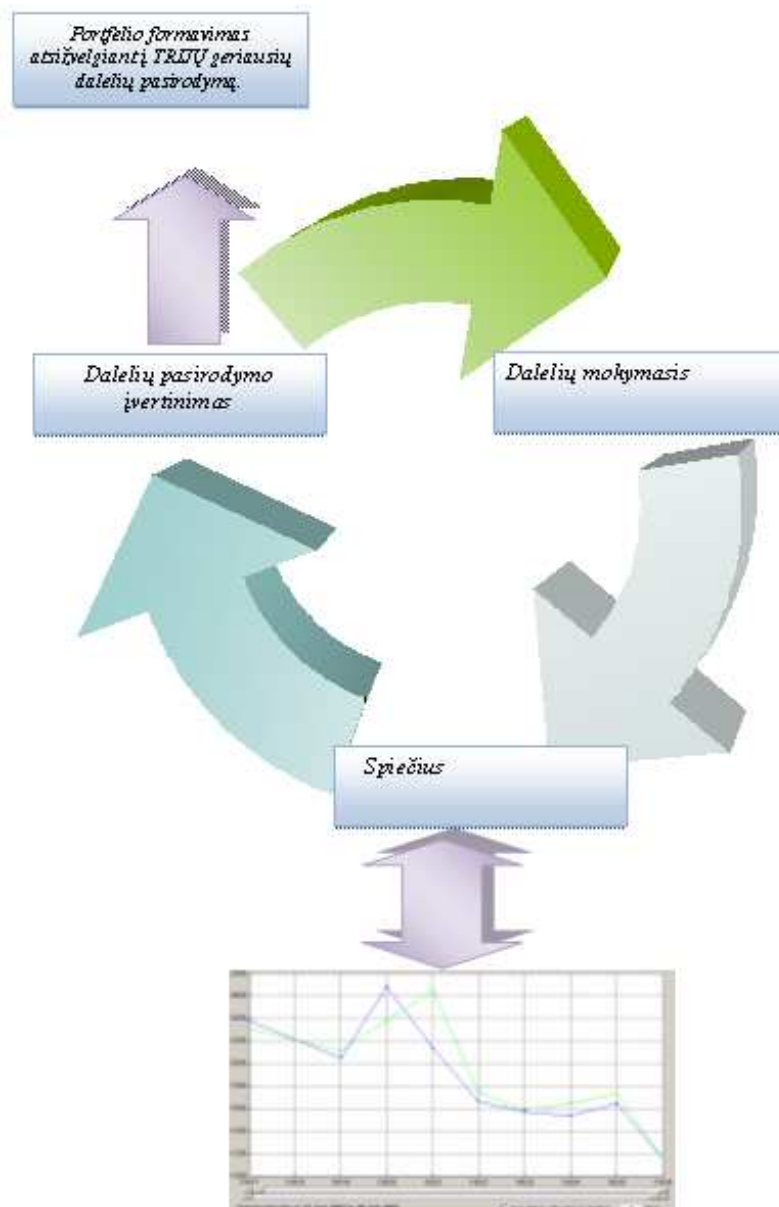
$$W_{\text{new}} = W_{\text{old}} + C * (W_{\text{best}} - W_{\text{old}})$$

W_{old} – ištirto neuroninio tinklo svoris;

W_{best} – geriausio neuroninio tinklo svoris;

C- Mokymo akcijos kursas.

Priešingai nei dr. Jovita Nenortaitė mano siūlomas priėmimo sprendimas pagrįstas neuroninių tinklų ir PSO algoritmu, tačiau galutinis sprendimas "pirkti", "parduoti" ar "laikyti" priimamas remiantis trijų geriausių neuroninių tinklų rekomendacijomis. Atliekamas palyginimas tarp neuroninių tinklų ir pasirenkami trys geriausių rezultatą žadantys tinklai. Prognozei atlikti remiamasi trijų geriausių rezultatą žadančių tinklų rezultatų vidurkiu. Manau, jog ateityje naudojant šį metodą gauti pelną bus lengviau. Mano siūlomo sprendimo priėmimo modelis pateikiamas šeštajame paveikslėlyje.



Šaltinis: Sudaryta autoriaus pagal NENORTAITĖ, J. Stock trading system based on application of swarm intelligence and artificial neural networks, 2006, p. 52.

6 pav. Sprendimo priėmimo idėja

Kaip matote 6 paveikslėlyje yra 5 svarbiausios sprendimo priėmimo metodo dalys: praeities akcijų duomenys, spiečius, dalelių pasirodymo įvertinimas, dalelių mokymasis ir portfelio sudarymas. Sprendimo priėmimas metodas naudoja praeities akcijų duomenis (S&P500). Akcijų duomenys perduodami į spiečių. Dalelės atlieka duomenų analizę ir prieina prie išvadų apie akcijų pirkimą, priklausomai nuo analizės rezultatų. Kiekviena dalelė atlieka savo savą analizę ir priima savus sprendimus. Kai dalelių sprendimai yra žinomi, tuomet kiekvienos dalelės pasirodymo įvertinimas vadinamas atliktu. Dalelės pasirodymo įvertinimas atliekamas per dalelės spiečiaus optimizavimo algoritmo paraišką.

Siūlomame sprendimo priėmimo metode neuroniniai tinklai yra dalelės, o spiečiaus optimizavimo algoritmas siūlomas tam kad, surastų neuroninio tinklo įvesties svorių vertes. Idėja pavaizduota paveikslėlyje „idėja sprendimo priėmimui“.

Yra trys pagrindiniai principai spiečiaus optimizavimo algoritme:

1. Dalelių pasirodymo įvertinimas;
2. Dalelių pasirodymo palyginimas;
3. Imitacija.

Dalelių pasirodymo įvertinimas atliekamas tam, kad būtų žinoma, ar dalelė yra "gera" ar "bloga". Dalelių pasirodymo palyginimas būtinas, kad būtų galima palyginti su kitomis egzistuojančiomis dalelėmis (kaimynais) ir pamėgdžioti tikrai tuos kaimynus, kurie yra geresni už lyginamą dalelę. Imitavimo procesas šiuo atveju yra svarbiausias išmokimui ir protinių gebėjimų palaikymui (Kennedy, J. ir Eberhart, R., 2001).

Siūlomame sprendimo priėmimo idėjos modelyje, kiekvienos dalelės pasirodymo įvertinimas atliekamas tokiais žingsniais: žinojimas, koks sprendimas buvo priimtas pagal dalelės sprendimo analizę, lyginant su realiais duomenimis. Jei kurios nors dalelės sprendimas buvo artimas tikrajai rinkos situacijai, tokiu atveju manoma, jog ir didžiausias pelnas galėtų būti pasiektas pagal tos dalelės sprendimą, vadinasi ši dalelė pasirodė geriausiai tą dieną. Geriausias pasirodymas reiškia, jog dalelė sugeba atlikti sprendimus, artimus realiai situacijai. Kai dalelė, parodžiusi geriausią rezultatą yra žinoma, visos kitos dalelės stumiamos link to geriausio rezultato. Skirtingai nuo genetinių algoritmų, spiečiaus optimizavimo algoritmo atveju, dalelės, kurios parodė blogiausią rezultatą nepašalinamos iš populiacijos, kadangi manoma, jog dažnai besikeičiančiose aplinkose, šios dalelės kada nors parodys geriausią rezultatą ir bus priimti sprendimai, kurie leis ateityje išvengti nuostolių.

Kaip pristatyta paveikslėlyje yra galimybės panaudoti skirtingas (x) modelio įvestis. Tarkime, kad turime $y = f(x; w)$, kur y yra rekomendacija, x - pastebimos kažkokio reiškinio reikšmės ir w metodo parametrų vertės. Sprendimo priėmimo metodo užduotis - surasti teisingą funkciją f , kuri galėtų duoti geriausią rezultatą, priimant sprendimus. Idėja apie algoritmą, skirta geriausios funkcijos f pasirinkimui, gali būti padalyta į kitus svarbiausius žingsnius:

1. Apskaičiuoti investicinę kiekvieno "eksperto" rekomendaciją ($y_i = f_i(x; w)$);
2. Įvertinti kiekvieno eksperto pasirodymą (galima nauda eksperto atveju, kai ekspertas būtų pasirinkęs praėjusios dienos galimybę), kai $p_i(y_i(t))$ yra nauda laiku t eksperto i ;
3. Surasti nugalėtoją ekspertą w_{in} su geriausiu pasirodymu $w_{in} = \text{argmax}_i(j_i)$;
4. Informuoti visus ekspertus apie geriausią ekspertą w_{in} - dalijasi patirtimi vienas su kitu;

5. Investuoti pagal laimėtoją ekspertą (Nenortaitė., J.,2006).

Šiame darbe manome, kad f = neroniam tinklui, x - pastebimų reiškinių verčių delta ir w – neuroninių tinklų svoriai. Žinant x ir w vertes lengva apskaičiuoti y rekomendacijas. Kai rekomendacija yra žinoma, funkcija (dirbtinis neurono tinklas), kuris duoda geriausią rezultatą einamuoju laiku f_{win} yra pažymima. Ši funkcija yra naudojama sprendimo priėmimui.

Lyginant su kitais egzistuojančiais sprendimais šis sprendimų priėmimo metodas yra geresnis dėl savo lankstumo. Jis gali būti pritaikomas daugelyje kitų sričių, tokių kaip telekomunikacija, logistika ir taip toliau. Svarbiausia, jog pasiūlytu sprendimų priėmimo metodu gali naudotis skirtingi vartotojai (investiciniai fondai, nepriklausomi vartotojai ar brokeriai), priklausomai nuo jų finansinių galimybių ir prekybos mastų, sumų. Vartotojas gali išrinkti prekybos strategiją, gali keiti akcijas ar indeksus.

2.2 Sprendimo priėmimo algoritmas

Kad pateikti išsamesnį pasiūlyto sprendimo priėmimo metodo paaiškinimą, nuspręsta naudoti trumpą algoritmo paaiškinimą. Iš pradžių pateiksiu Nenortaitės J., pasiūlytą sprendimo priėmimo algoritmą ir detalų jo aprašymą, o vėliau bus aprašomas ir pateikiamas optimizuotas sprendimo priėmimo algoritmas.

Algorithm *Reccomend*(*day*)

1. **for each** *net* **do**
2. **for each** *S* **do**
3. **for each** *k* **do**
4. $\Delta_{\langle day, S, k \rangle} \leftarrow 100 \cdot \frac{price_{\langle day, S \rangle} - price_{\langle day - k, S \rangle}}{price_{\langle day - k, S \rangle}}$
5. **end**
6. $rec_{\langle day, net, S \rangle} \leftarrow sigmod\left(\sum_k \Delta_{\langle day, S, k \rangle} \cdot w_{\langle net, k \rangle}\right)$
7. **end**
8. $rec'' \leftarrow rec$
9. **for** *no* = 1 **to** *invest* **do**
10. $beststk_{\langle no \rangle} \leftarrow argmax_S(rec''_{\langle day - 1, net, S \rangle})$
11. $rec'' \leftarrow rec'' \setminus rec''_{\langle day - 1, net, beststk_{\langle no \rangle} \rangle}$
12. **end**
13. $sumrek_{\langle day, net \rangle} \leftarrow mean\left(\sum_{no} \Delta_{\langle day, beststk_{\langle no \rangle}, 1 \rangle}\right)$
14. $maxint_{\langle day, net \rangle} \leftarrow \sum_{d=day-window}^{day} sumrek_{\langle d, net \rangle}$
15. **end**
16. $bestnet \leftarrow argmax_{net}(maxint_{\langle day, net \rangle})$
17. **for each** *net, k* **do**
18. $w_{\langle net, k \rangle} \leftarrow w_{\langle net, k \rangle} + \eta \cdot (w_{\langle bestnet, k \rangle} - w_{\langle net, k \rangle})$
19. **end**
20. **for** *no* = 1 **to** *invest* **do**
21. $reccomend_{\langle day, no \rangle} \leftarrow argmax_S(rec_{\langle day, bestnet, S \rangle})$
22. $rec \leftarrow rec_{\langle day, bestnet, reccomend_{\langle no \rangle} \rangle}$
23. **end**
24. **return** *reccomend*

Šaltinis: sudaryta NENORTAITĖ, J. Stock trading system based on application of swarm intelligence and artificial neural networks, 2006. p. 52.

7 pav. Sprendimo priėmimo algoritmas

Nuo pirmosios algoritmo eilutės pradedami analizuoti visi tinklai. Taigi 1 - 15 algoritmo eilutėse įvertinami tinklai. 3-5 eilutėse suskaičiuojamos *k* delta reikšmės.

$$\Delta_{\langle day, S, k \rangle} \leftarrow 100 * \frac{price_{\langle day, S \rangle} - price_{\langle day - k, S \rangle}}{price_{\langle day - k, S \rangle}}$$

Čia *price* sudaro *day* dienos akcijų kainos *S*. Tuomet 6 eilutėje kiekvienai akcijai suskaičiuojama rekomendacija. Po to algoritmas ieško akcijų su geriausiomis rekomendacijomis. Geriausia akcija randama naudojant *argmaxS* funkciją, kuri suranda akciją *S* su aukščiausiomis rekomendacijomis (10 eilutė). Tam kad surasti kitą geriausią akciją, surasta rekomendacija ištrinama iš išanalizuoto rekomendacijų rinkinio (11 eilutė). Po to kai surandamos geriausios

akcijos, algoritmas suranda geriausių rekomenduojamų vakar dienos ($k=1$) akcijų kainų pokyčių vidurkį. (13 eilutė). Toliau algoritmas sumuoja geriausių rekomenduojamų akcijas.

Tokiu būdu tinklai yra apdorojami ir po to nustatomas geriausias tinklas (16 eilutė). Geriausių tinklu laikomas tas tinklas, kuris parodo geriausią *maxint* (aukščiausias *maxint* reikšmes). Likusių tinklų svoriai pritaikomi pagal geriausią tinklą (17 – 19 eilutės). Tada pasirenkamos geriausios rekomenduojamos akcijos pagal geriausią tinklą (20-23 eilutės), čia pasirenkamos akcijos, kurios turi aukščiausias rekomendacijų reikšmes (*rec*) naudojant geriausią tinklą.

Kadangi mano pasiūlytas metodas pagrįstas ne geriausio tinklo (*bestnet*) rezultatu, o trijų geriausių tinklų rezultatų vidurkiu, todėl 16 algoritmo eilutė, 7 paveikslėlyje pažymėta raudonai, bus koreguojama. Tuo tikslu bus optimizuojamas algoritmas rekomendacijos skaičiavimui, kuris pristatytas apačioje. Raudona spalva pažymėti algoritmo koregavimai.

Algoritmų palyginimas

Totalnets=30;	Tiriamų tinklų skaičius.	Totalnets=30;	---“---
K_dienu=5;	Naudojamų analizei k_dienų atgal duomenys.	K_dienu=5;	---“---
Tiriamos akcijos		Tiriamos akcijos	
Yahoolist = loadstockfile('stocklist.txt'); Stockyahood = [yahoolist.indexlist;yahoolist.stocklist]; Akc = size(stockyahood,1); Stockyahood = [stockyahood; {'^gspc'}];	Tiriamos akcijos.	Yahoolist = loadstockfile('stocklist.txt'); Stockyahood = [yahoolist.indexlist;yahoolist.stocklist]; Akc = size(stockyahood,1); Stockyahood = [stockyahood; {'^gspc'}];	---“---
I_cumm=50; Deltawindow = i_cumm; Stockwindow = i_cumm+k_dienu; Fromdate = datenum('03-jan-2000'); Todate = datenum('05-dec-2005');	Dienų skaičius naudojamas kumuliatyviniam <i>swarm</i> elemento <i>fitness</i> skaičiavimui. Tikrinimas, ar gerai deltas paskaičiuoja.	I_cumm=50; Deltawindow = i_cumm; Stockwindow = i_cumm+k_dienu; Fromdate = datenum('03-jan-2000'); Todate = datenum('05-dec-2005');	---“---
Deltų palyginimas		Deltų palyginimas	
Idienos=size(stockdata.data,2); Duom(:,:,1) = stockdata.data; Startas=1+k_dienu;	Galime pradėti dirbti tik nuo tos dienos kadangi reikalingi k_dienu pokyčiai.	Idienos=size(stockdata.data,2); Duom(:,:,1) = stockdata.data; Startas=1+k_dienu;	---“---
For k=1:5 for i=startas:idienos for j=1:akc+1	Skaičiuojame pokyčius visoms akcijoms ir <i>s&p500</i> indeksui.	For k=1:5 for i=startas:idienos for j=1:akc+1	---“---
delta(j,i-k_dienu,k)=(duom(j,i,1)-duom(j,i-k,1))/duom(j,i-k,1)*100; end end End	Pokyčių skaičiavimas.	delta(j,i-k_dienu,k)=(duom(j,i,1)-duom(j,i-k,1))/duom(j,i-k,1)*100; end end End	---“---
A = round(delta*10000) - round(deltadata.delta*10000); Display('kiek sutampa'); Size(find(abs(a) == 0)) Display('kiek nesutampa'); Size(find(abs(a) > 0)) Display('kiek nesutampa paklaida > 0.0001'); Size(find(abs(a) > 1)) % kiek nesutampa paklaida > 0.0001	Deltų palyginimas.	A = round(delta*10000) - round(deltadata.delta*10000); Display('kiek sutampa'); Size(find(abs(a) == 0)) Display('kiek nesutampa'); Size(find(abs(a) > 0)) Display('kiek nesutampa paklaida > 0.0001'); Size(find(abs(a) > 1)) % kiek nesutampa paklaida > 0.0001	---“---

Delta = deltadata.delta; Idienos=size(delta,2); Clear m; Id=idienos-k_dienu	Analizuojamų dienų skaičius. Id -dienų skaičius, kurioms Turime pilnus pokyčių vektoriaus duomenis.	Delta = deltadata.delta; Idienos=size(delta,2); Clear m; Id=idienos-k_dienu	----“---
Neuroniniai tinklai		Neuroniniai tinklai	
Tmp_w=rand(6,totalnets)-0.5; Suma=zeros(8,3027,2); Rekomendacija=zeros(8,3022); Ir=0;	Rnd svoriu nustatymas Parametras reikalingas realioms kapitalo kaupimo dienoms suskaičiuoti.	Tmp_w=rand(6,totalnets)-0.5; Suma=zeros(8,3027,2); Rekomendacija=zeros(8,3022); Ir=0;	----“---
For i=1:id-1 for net=1:totalnets for j=1:akc suma(j,i,net)=0; for k=1:5		For i=1:id-1 for net=1:totalnets for j=1:akc suma(j,i,net)=0; for k=1:5	----“---
suma(j,i,net)=suma(j,i,net)+delta(j,i,k)*tmp_w(k,net); End	Bendra suma padauginta iš w	suma(j,i,net)=suma(j,i,net)+delta(j,i,k)*tmp_w(k,net); End	
rekomendacija(j,i,net)=(2./(1+2.71828^(-suma(j,i,net))))-1; end end	Rekomendacijos[-1,+1] skaičiavimas.	rekomendacija(j,i,net)=(2./(1+2.71828^(- suma(j,i,net))))-1; end end	----“---
for net=1:totalnets [y,i]=sort(rekomendacija(:,i,net)); y=flipud(y); i=flipud(i); for j=1:akc yy(j,i,net)=y(j); ii(j,i,net)=i(j); end end	Akcijų rekomendacijų rūšiavimas.	for net=1:totalnets [y,i]=sort(rekomendacija(:,i,net)); y=flipud(y); i=flipud(i); for j=1:akc yy(j,i,net)=y(j); ii(j,i,net)=i(j); end end	----“---
Pelno skaičiavimas kiekvienai dienai, pasirinkus 3 rekomenduotinas akcijas ir žiūrint ką jos duos kitą dieną.		Pelno skaičiavimas kiekvienai dienai, pasirinkus 3 rekomenduotinas akcijas ir žiūrint ką jos duos kitą dieną.	
k=1; for net=1:totalnets sumrek(1,net)=0;	Pirmą dieną pelno neturime	k=1; for net=1:totalnets sumrek(1,net)=0; %	----“---

For no=1:3 p=0; %k=1;	Atrenkamos pirmos trys šiandien rekomenduotinos akcijos ir žiūrimas jų pelnas rytojaus dienai.	For no=1:3 p=0; %k=1;	---“---
P=p+delta(ii(no,i,net),i+1,1); end sumrek(i+1,net)=p/3; end	Sumuojami 3 geriausių akcijų pokyčiai	P=p+delta(ii(no,i,net),i+1,1); end sumrek(i+1,net)=p/3; end	---“---
Kumuliatyvines slenkančios sumos skaičiavimas		Kumuliatyvines slenkančios sumos skaičiavimas	
in=i; delta_tomorrow(1)=0; sp_tomorrow(1)=0; tmp=0; if in < i_cumm+1 for net=1:totalnets	Kol mažiau negalime skaičiuoti	in=i; delta_tomorrow(1)=0; sp_tomorrow(1)=0; tmp=0; if in < i_cumm+1 for net=1:totalnets	---“---
Maxint(in,net)=0; end else for net=1:totalnets		Maxint(in,net)=0; end else for net=1:totalnets	---“---
Maxint(in,net)=sum(sumrek((in-i_cumm):in,net));	Kiekvienam tinklui skaičiuojame slenkančio lango <i>fitness</i> .	Maxint(in,net)=sum(sumrek((in-i_cumm):in,net));	---“---
tmp(net)=maxint(in,net); end end	Randame tinklą, kuris yra geriausias pasirinktame laiko intervalui.	tmp(net)=maxint(in,net); end end	---“---
Geriausio tinklo radimas		Geriausio tinklo radimas	
if i >= i_cumm+1 ir=ir+1; [y1,i1]=sort(tmp); y1=fliplr(y1);	Rūšiavimas pagal kumuliatyvių sumų max, nes reikės geriausios.	if i >= i_cumm+1 ir=ir+1; [y1,i1]=sort(tmp); y1=fliplr(y1);	---“---

<pre> i1=flipr(i1); bestnet(ir)=i1(1); for net=1:totalnets if net ~= bestnet for k=1:5 tmp_w(k,net) = tmp_w(k,net) + 0.05*(tmp_w(k,bestnet(ir))- tmp_w(k,net)); end end end end </pre>	Geriausio tinklo indeksas yra i1(1).	<pre> i1=flipr(i1); bestnet(ir)=i1(1); for net=1:totalnets if net ~= bestnet for k=1:5 tmp_w(k,net) = tmp_w(k,net) + 0.05*(tmp_w(k,bestnet(ir))-tmp_w(k,net)); end end end end </pre>	---“---
<pre> Delta_tomorrow(ir+1)=mean(delta(ii(1:3,i,bestnet(ir)),i+1,1)) -0.15; Sp_tomorrow(ir+1)=delta(akc+1,i+1,1); end </pre>	Geriausios akcijos, paskaičiuotos pagal tinklą 'bestnet' diena i yra ii(1:3,i,bestnet).	Geriausių trijų tinklų vidurkių skaičiavimas	
<pre> End Clear sumrek; Clear rekomendacija; Clear maxint; </pre>	Visų dienų ciklo pabaiga	<pre> delta_sum_t=0; sp_sum_t=0; for loop=1:3 bestnet(ir)=i1(loop); Delta_sum_t=delta_sum_t + mean(delta(ii(1:3,i,bestnet(ir)),i+1,1))-0.15; Sp_sum_t=sp_sum_t + delta(akc,i+1,1); end delta_tomorrow(ir+1)=delta_sum_t/3; sp_tomorrow(ir+1)=sp_sum_t/3; end End Clear sumrek; Clear rekomendacija; Clear maxint; </pre>	<p>Vykdomė ciklą tris kartus</p> <p>Imame tinklus iš eilės pagal gerumą ir sumuojame delta tomorrow ir sp tomorrow</p> <p>Priskiriame vidurkį</p> <p>Visų dienų ciklo pabaiga.</p>

Šaltinis: sudaryta autoriaus pagal NENORTAITĖ, J. Stock trading system based on application of swarm intelligence and artificial neural networks, 2006, p. 52.

2.3. Ekspertinės sistemos

Dažnai literatūroje cituojama garsi eksperto N.Idelson iš Midland Global Markets kompanijos frazė, kad “naudojant neuroninius tinklus gauti pelną finansų rinkose gali gauti tik tie žmonės, kurie sugeba gauti pelną ir be neuroninių tinklų”. Čia turima omenyje, kad norint tinkamai apmokyti neuroninius tinklus, reikia turėti pakankamai žinių apie analizuojamą finansinę sistemą, nes vien formalus duomenų paruošimas ir jų apdorojimas gali daugeliu atveju būti neefektyvus.

Pastaraisiais metais vis svarbesnės tampa intelektinės sistemos, kurios plačiąją prasme remiasi programiniais skaičiavimais. Tradicinio aparatinio skaičiavimo operacijoms būdingas tikslumas ir apibrėžtumas, tuo tarpu programiškai skaičiuojama su tam tikrais tikslumo nuostoliais.

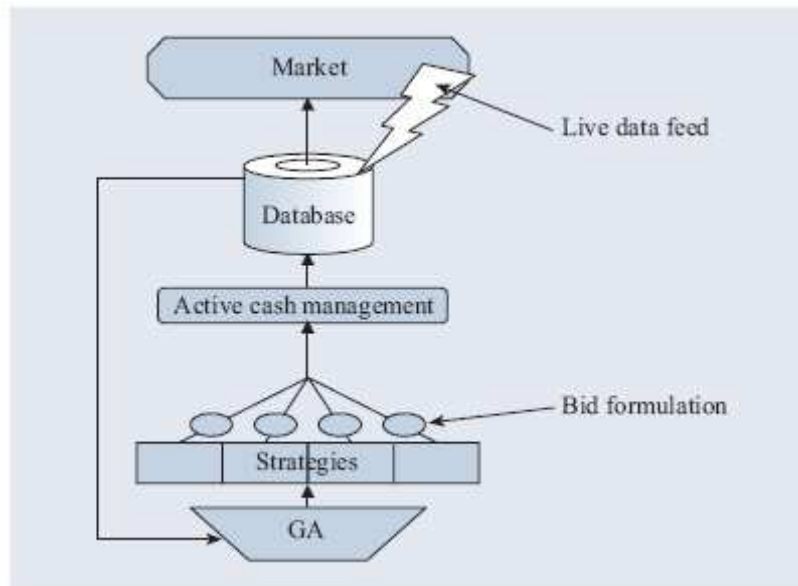
Dažniausiai tokios sistemos remiasi skaičiavimais neuroniniais tinklais, bendriniais algoritmais ir tikimybinio pagrindimu.

Taigi šiame skyriuje bus apibūdinamos kelios tokios intelektinės sistemos, kurių pagalba gali būti gaunamas pelnas ateityje. Taip pat bu pateikiamas pasiūlyto sprendimo metodo patikrinimui naudojamos sistemos detalus aprašymas.

2.3.1. GA/GP trading rule optimizer

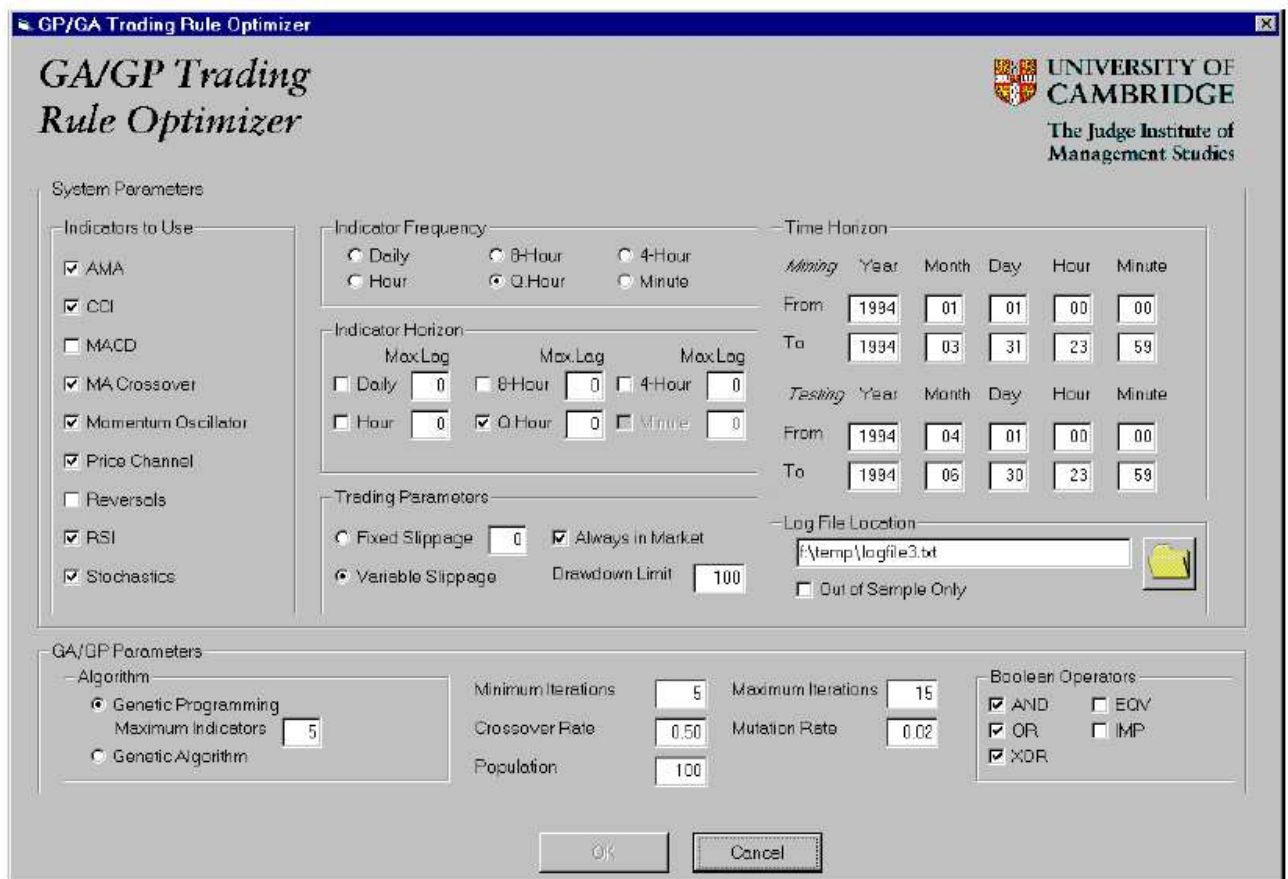
Sistema susideda iš genetiniu programavimu paremtų taisyklių, yra galimybė pasirinkti prekybos strategijas, pagrįstas skirtingų indikatorių kombinacijomis ir taisyklėmis. Genetiniai algoritmai panaudoti pradiniam portfelio taisyklių pasirinkimui. Sistema turi gebėjimą panaudoti pasirinktas strategijas, kad galėtų vėliau įvykdyti nebenaudojamų duomenų analizę arba realaus laiko duomenų analizę. Pasirinktų strategijų rekomendacijos paremtos grynujų pinigų valdymu.

Dempster ir Jones (2000) pasiūlytos prognozavimo sistemos analizei atlikti buvo naudojami FX tick British Pound/US Dollar valiutos kurso duomenys.



Šaltinis: DEMPSTER, M.; JONES, M. (2000) *A real-time adaptive trading system using genetic programming* [Prieiga per internetą: <mahd-pc.jbs.cam.ac.uk/archive/papers/2000/geneticprogramming.pdf>].

8 pav. Architektūra



Šaltinis: DEMPSTER, M.; JONES, M. (2000) *A real-time adaptive trading system using genetic programming* [Prieiga per internetą: <mahd-pc.jbs.cam.ac.uk/archive/papers/2000/geneticprogramming.pdf>].

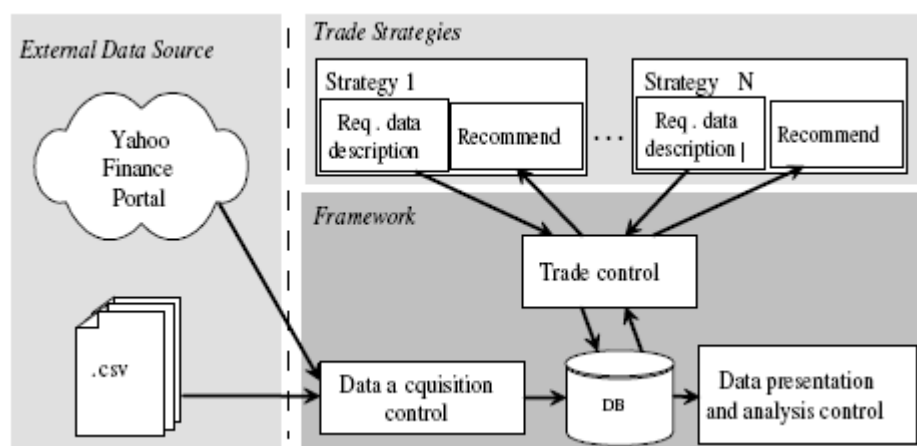
9 pav. Sistemos pagrindinis langas

2.3.2. Swarm 19.6

Swarm 19.6 sistema bus naudojama pasiūlyto sprendimo priėmimo metodo patikrinimui atlikti. Detalesnis naudojamos sistemos aprašymas bus pateikiamas skyriuje (žr. 3. Eksperimentinis skyrius).

2.3.2.1. Sistemos architektūra

Swarm 19.6 sistema bus naudojama pasiūlyto sprendimo priėmimo metodo patikrinimui atlikti. Jos architektūra ir pagrindiniai išvystytos prognozavimo sistemos komponentai yra pristatyti devintajame paveikslėlyje.



Šaltinis: NENORTAITĖ, J. Stock trading system based on application of swarm intelligence and artificial neural networks: doctoral dissertation, 2006, p. 80.

10 pav. Sistemos architektūra

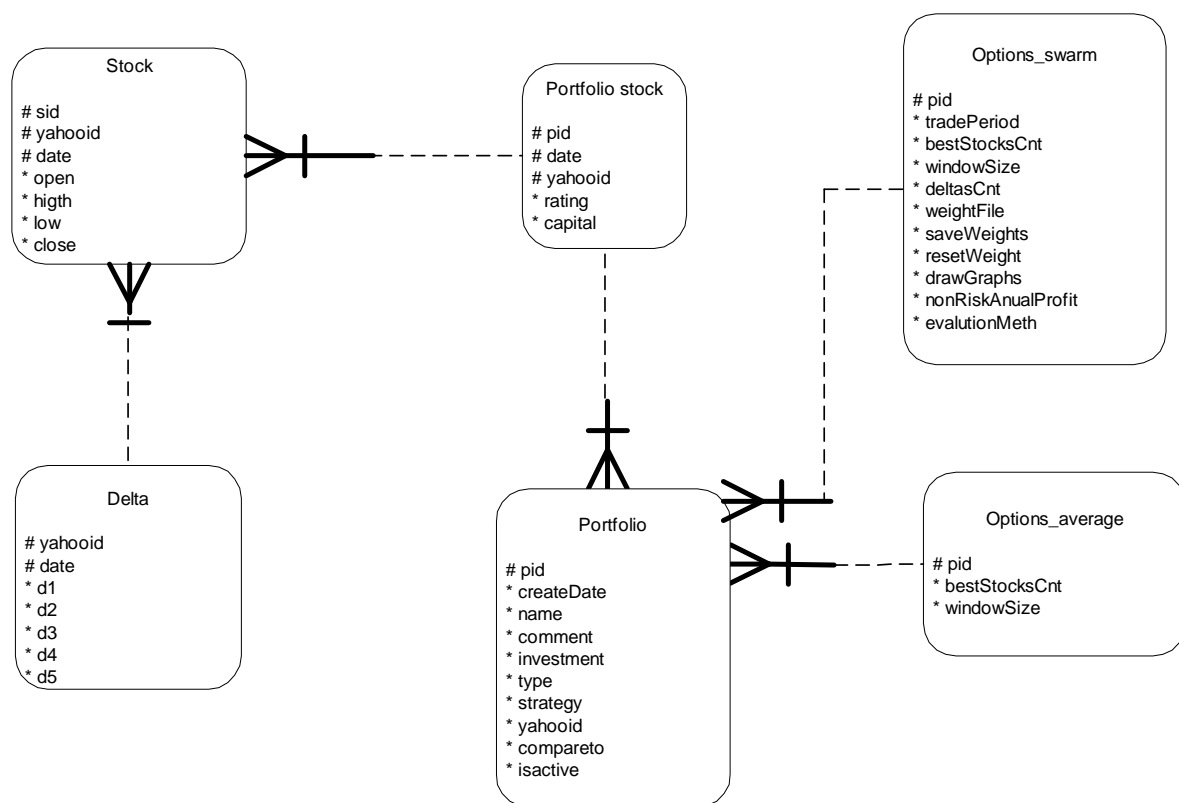
Paveikslėlyje vaizduojama sistemos architektūra padalinta į tris dalis: išorinis duomenų šaltinis ir dvi vidaus dalys (sandara ir prekybos strategijos). Išorinis duomenų šaltinis naudojamas išrinkti realaus laiko ar istorinius akcijų duomenis. Svarbiausia sistemos dalis yra sandara (framework), ji susideda iš šių dalių .

- Duomenų parsisiuntimas iš išorinio duomenų šaltinio;
- Duomenų sukėlimas į duomenų bazę;
- Duomenų paruošimas pagal prekybinę strategijų pasirinkimą;
- Įvykdymas;
- Grafinio vaizdo pateikimas;
- Pateikti duomenų analizės reikšmes.

2.3.2.2. Konceptinis objekto modelis (ER diagramos)

Suprojektuotas DB konceptualus modelis orientuotas į loginį reliacinį modelį. Bendresnio pavidalo DB konceptualiam modeliui, kuris naudojamas ankstyvosiose DB projektavimo stadijose, sudaryti ir smulkiai aprašyti dažnai naudojamos ER (angl. *Entity-relation diagram*, lietuviškai - *esybių - ryšių*) diagramos.

ER modelis - jo komponentai: esybė, atributas, ryšys. Esybė – tai vieno tipo informacinių objektų aibė (klasė); realiai egzistuojantys ar įsivaizduojami modeliuojamos DB taikymo srities prasmieniai objektai, daiktai, sąvokos. Kiekvienai esybei suteikimas unikalus tapatumo vardas. Šiuo atveju esybės yra: stock, stock portfolio, delta, portfolio, options_swarm, options_avegare. Eybės yra sujungtos ryšiais tarpusavyje (Garšva, G., Sekliuckis, V., Gudas, S., 2003).



Šaltinis: sudaryta autoriaus pagal NENORTAITĖ, J. Stock trading system based on application of swarm intelligence and artificial neural networks. 2006, p. 80.

11 pav. ER diagrama

Ryšys ER modelyje apibrėžiamas kaip dvinaris junginys (asociacija), kuris nusako esybių tarpusavio santykį arba sąveiką.

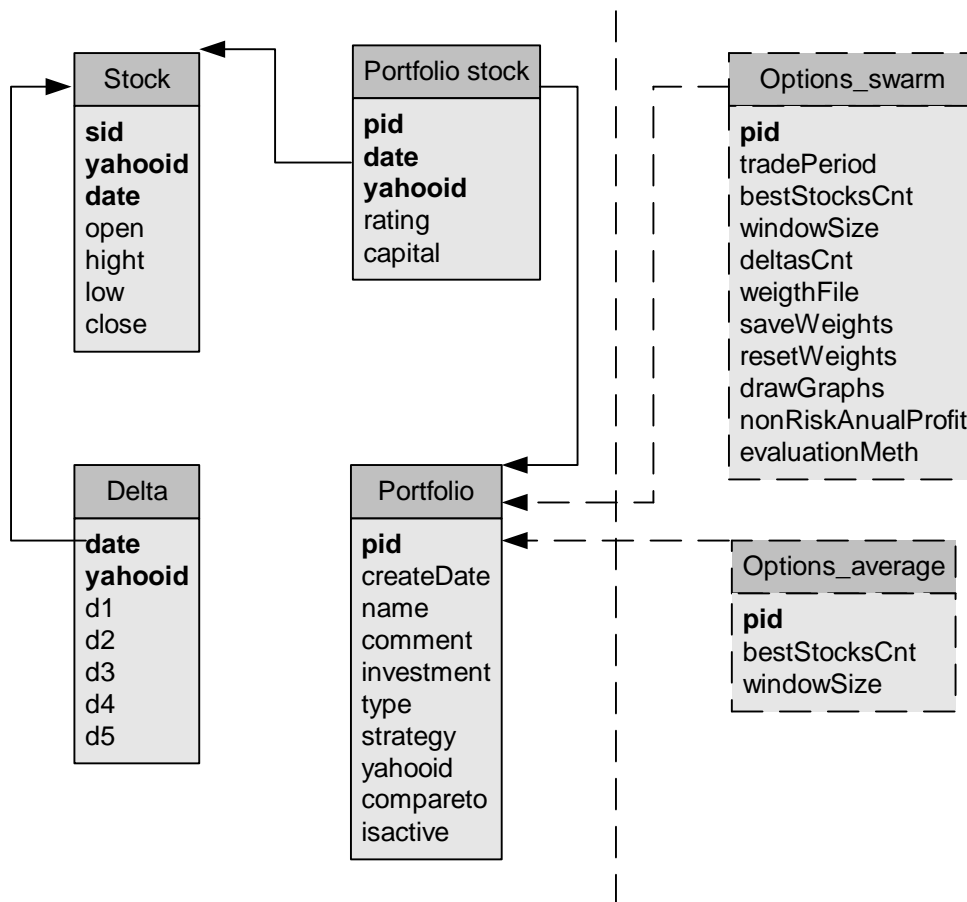
Diagramoje naudojamas atributų žymėjimas:

– identifikatorius;

* – privalomi atributai.

2.3.2.3. Duomenų bazės projektas

Duomenų bazė – įvairių tipų įrašų ir santykių tarp įrašų, duomenų agregatų ir elementų sistema. Taigi atsižvelgiant į suprojektuotą DB konceptualų modelį, suprojektuota siūlomo sprendimo priėmimo sistemos DB loginė schema, pagal kurią ir buvo kuriama Swarm 19.6 sistema.



Šaltinis: sudaryta autoriaus pagal NENORTAITĖ, J. Stock trading system based on application of swarm intelligence and artificial neural networks. 2006, p. 80.

12 pav. Duomenų bazė

Kaip matome iš DB loginės schemas, duomenų bazėje yra 6 lentelės, į kurių laukus duomenys suvedami rankiniu būdu arba juos importuojant MYSQL duomenų bazės įrankių pagalba.

4 lentelė

DB lentelių ir laukų aprašymas

Eil.Nr	(Lentelės vardas) lauko vardas	Tipas	Ilgis	Būtinasis	Pastaba
1.	(stock) sid	int	11	Taip	Raktas.
2.	(stock) date	date		Taip	Data
3.	(stock) yahood	varchar	10	Taip	Akcijos id.
4.	(stock) open	real		Taip	Tam tikros dienos akcijų kaina.
5.	(stock) high	real		Taip	Tam tikros dienos akcijų kaina.

Eil.Nr	(Lentelės vardas) lauko vardas	Tipas	Ilgis	Būtinai	Pastaba
6.	(stock) low	real		Taip	Tam tikros dienos akcijų kaina.
7.	(stock) close	real		Taip	Tam tikros dienos akcijų kaina.
8.	(delta) date	date		Taip	Data.
9.	(delta) yahooid	varchar	10	Taip	Akcijos id.
10.	(delta) d1	real		Taip	Akcijų kainų pokytis
11.	(delta) d2	real		Taip	Akcijų kainų pokytis
12.	(delta) d3	real		Taip	Akcijų kainų pokytis
13.	(delta) d4	real		Taip	Akcijų kainų pokytis
14.	(delta) d5	real		Taip	Akcijų kainų pokytis
15.	(portfolio) pid	int	11	Taip	Portfelio id.
16.	(portfolio) createdate	date		Taip	Portfelio sukūrimo data.
17.	(portfolio) name	varchar	20	Taip	Portfelio pavadinimas.
18.	(portfolio) comment	varchar	255	Taip	Komentaras.
19.	(portfolio) investment	real		Taip	Pirminė investavimo suma.
20.	(portfolio) type	enum		Taip	Naudojamas, tam, kad galima būtų atskirti indekso ar reguliarių akcijų investicijas.
21.	(portfolio) strategy	int	1	Taip	Strategija.
22.	(portfolio) yahooid	varchar	10	Taip	Akcijos id.
23.	(portfolio) compareto	int	10	Taip	
24.	(portfolio) isactive	int	1	Taip	
25.	(portfoliostock) pid	int	1	Taip	Strategija.
26.	(portfoliostock) date	date		Taip	Data.
27.	(portfoliostock) yahooid	varchar	10	Taip	Akcijos id.
28.	(portfoliostock) rating	int	11	Taip	Skirtas tikrinti pirktų akcijų įvertinimui. Šis laukas įrašomas pagal investavimo strategiją.
29.	(portfoliostock) capital	real		Taip	Investuojama pinigų suma.
30.	(options_swarm) pid	int	11	Taip	Portfelio id.
31.	(options_swarm) tradePeriod	int	6	Taip	Prognozavimo periodas dienomis.
32.	(options_swarm) beststockscout	int	6	Taip	Perkamų akcijų skaičius.
33.	(options_swarm) windowsize	int	6	Taip	Duomenų atvaizdavimo langas.
34.	(options_swarm) deltascount	int	6	Taip	Naudojamų deltų skaičius.
35.	(options_swarm) anncount	int	6	Taip	
36.	(options_swarm) weightfile	varchar	255	Taip	Tinklų svorių failas.
37.	(options_swarm) saveweights	int	1	Taip	
38.	(options_swarm) resetweights	int	1	Taip	
37.	(options_swarm) drawgraphs	int	1	Taip	Laukas , skirtas grafikų atvaizdavimui.
38.	(options_swarm) nonriskannualprofit	real		Taip	
39.	(options_swarm) evaluationMeth	enum		Taip	Pasirenkamas įvertinimo metodas.

Šaltinis: sudaryta autoriaus

3. EKSPERIMENTINIAI TYRIMAI

Šiame skyriuje bus pristatyti pasiūlyto sprendimo metodo bandomieji tyrimai ir analizės ypatybės. Bus pristatyti tyrimams naudojami duomenys, įrankiai. Paskutiniuoju šio skyriaus dalyje bus suformuotos ir pateiktos gautų rezultatų išvados. Eksperimentinis tyrimas bus vykdomas ir aprašomas darbe tokia seka:

6. Duomenų rinkimas;
7. Duomenų apdorojimas ir analizė;
8. Bandomieji tyrimai;
9. Rezultatai;
10. Išvados.

3.1. Duomenų rinkimas

Tyrimams atlikti naudojami S&P500 grupės indekso duomenys. Pasirinkti kasdieniniai akcijų duomenys laikotarpyje nuo 1998 gegužės 1 d. iki 2008 gegužės 1 d. Duomenys atsiųsti iš *yahoo.finance.com* tinklalapio. Akcijų duomenys talpinami *stocklist.txt* faile.

Investuotojo pelnas bus apskaičiuojamas pagal formulę:

$$P_t = \frac{K_t - K_{t-1}}{K_{t-1}} * 100\%$$

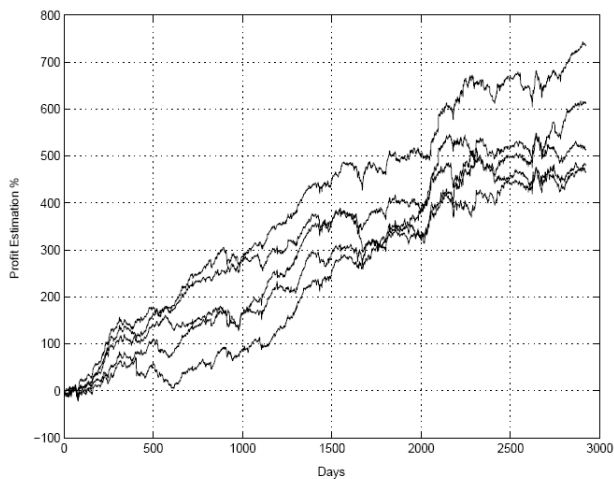
P_t – pelnas per tam tikrą laiko tarpą;

T – laiko tarpas per kurį perkame pasirinktas akcijas už bendrą sumą

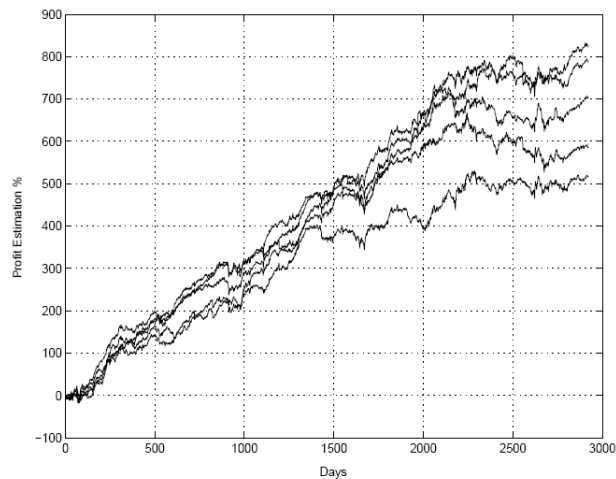
K_{t-1} , tuomet kitą dieną šių akcijų kaina taps K_t ir pelnas bus lygus P_t .

3.2. Duomenų pasirinkimas

Tyrimui atlikti buvo pasirinkti 30 tinklų, kadangi didesnis skaičius neuroninių tinklų su skirtingais svoriais leidžia pasiekti tvirtesnius rezultatus. Tuo tikslu buvo atliktas tyrimas, panaudojant skirtingą kiekį neuroninių tinklų, tas pačias akcijas ir dienas. Tyrimo rezultatai matomi apačioje, tryliktame paveikslėlyje.



a. 5 ANN, sliding window of 100 days



b. 30 ANN, sliding window of 100 days

Šaltinis: NENORTAITĖ, J. Stock trading system based on application of swarm intelligence and artificial neural networks. 2006, p. 80.

13 pav. Pelno įvertinimas

3.3. Duomenų apdorojimas ir analizė

Pasiūlyto sprendimo tinkamumui pagrįsti ir patikrinti buvo sukurta sistema, kuri turi funkcionuoti *Windows XP* aplinkoje ir jai įgyvendinti buvo taikomos šios kompiuterizuotos priemonės:

1. MySQL duomenų bazė;
2. MATLAB duomenų apdorojimo įrankis;
3. MATLAB Datafeed toolbox įrankis;
4. MySQL - MATLAB connector.

Diegimo procesas vyksta tokia tvarka:

1. MySQL duomenų bazės diegimas savo kompiuteryje. (įrašykite `\$MYSQL = the MYSQL`);
2. MATLAB Matlab ir datafeed toolbox diegimas savo kompiuteryje. (įrašykite `\$ MATLAB = the MATLAB`);
3. MySQL - MATLAB connector diegimas;
4. Nukopijuokite `\$MYSQL/lib/opt/libmysql.dll` to `\$WINDOWS/system32` (kur `\$WINDOWS = the WINDOWS`);
5. Sukompiliuokite `mysql.cpp` failą, naudojant MATLAB mex funkciją:
`mex -I"C:/mysql/include" -DWIN32 mysql.cpp "C:/mysql/lib/opt/libmySQL.lib" ;`
6. Atnaujinkite datafeed toolbox įrankį:
 - a. Uždarykite MATLAB programą;

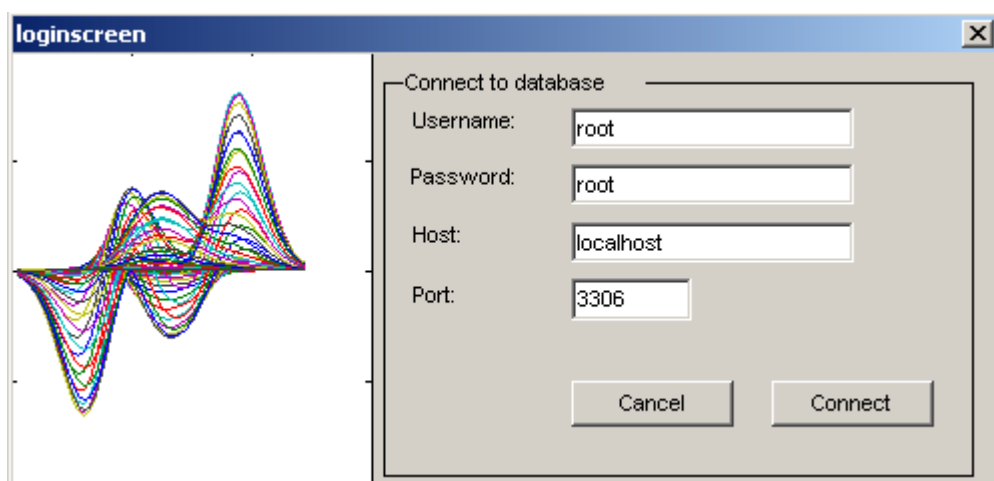
- b. Pervadinkite the following M-failą:
 \\${MATLAB}/toolbox/datafeed/datafeed/@yahoo/fetch.m į fetch.m.old;
 - c. Nukopijuokite iš fetch.m failą į \\${MATLAB}/toolbox/datafeed/datafeed/@yahoo/;
 - d. Paleiskite MATLAB programą;
 - e. Paleidę MATLAB naudokite funkciją: rehash toolboxcache.
7. Sukurkite duomenų bazę;
 8. Atlikę visus anksčiau paminėtus žingsnius, galite paleisti sistemą.

3.4. Bandomieji tyrimai

Bandomieji tyrimai atliekami naudojantis Swarm 19.6 sistema..

Pirminis (žr. 2.2.2. Swarm 19.6) akcijų prekybos sistemos tikslas – pateikti patogią aplinką, pritaikytą vartotojui, norinčiam vystyti ir įvertinti akcijų prekybai naudojamus algoritmus. Sistema gali naudotis ir mažą patirtį naudojimosi kompiuteriu turintys asmenys, kadangi valdymo procesas yra apgalvotas taip, kad būtų paprasta naudotis kiekvienam vartotojui. Svarbu pabrėžti, jog kiekvienai naujai kuriamai sistemai keliami reikalavimai yra orientuoti į vartotojo poreikius, tačiau vis dėlto įsigijus ją, sunku iškart suprasti, kaip naudotis sistema, todėl turi būti pateikiamos naudojimosi taisyklės

1. Bandomieji tyrimai pradedami prisijungimu prie duomenų bazės. Prisijungimas atliekamas įvedant vartotojo vardą : root, slaptažodį: root, host pavadinimą : localhost bei port : 3306.



Šaltinis: sudaryta autoriaus

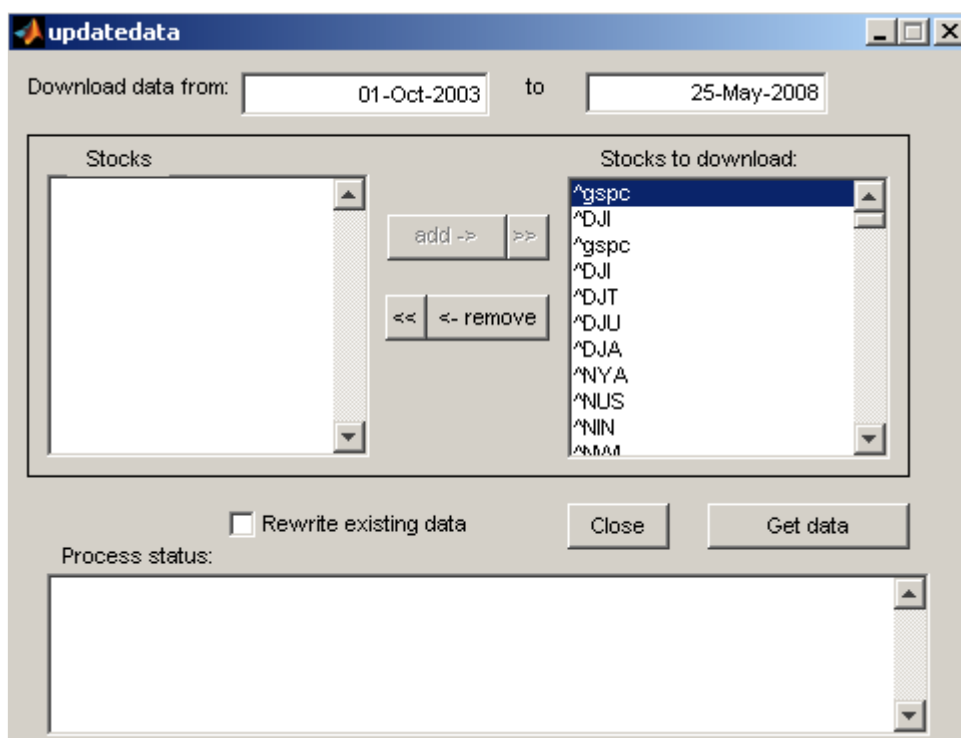
14 pav. Prisijungimo langas

Sėkmingas prisijungimas prie sistemos laikomas tokiu atveju, jei MATLAB darbalaukyje matome panašų tekstą: *Connecting to host=localhost port=3306 user=root password=root Uptime: 51208 Threads: 1 Questions: 1 Slow queries: 0 Opens: 11 Flush tables: 1 Open tables: 0 Queries per second avg: 0.000*

Įsitikinti, jog tikrai pavyko prisijungti galima įvesti matlab funkciją: `mysql ('status')`. Sėkmingai prisijungus matome: `Localhost via tcp/ip server version 4.1.15-nt.`

Prisijungę prie sistemos formuosime portfelius ir atliksime akcijų rinkų prognozes.

2. Antras žingsnis yra duomenų atnaujinimas, kuris atliekamas naudojant meniu pasirinkimus "Data / Get New". Formoje pasirenkamos konkrečios akcijos, kurių duomenis atnaujinsime *Add* ir *Remove* mygtukų pagalba. *Add* mygtukas prideda akciją į sąrašą, o *Remove* – pašalina. Yra galimybė pasirinkti ir pašalinti akcijas visas iškart arba po vieną.



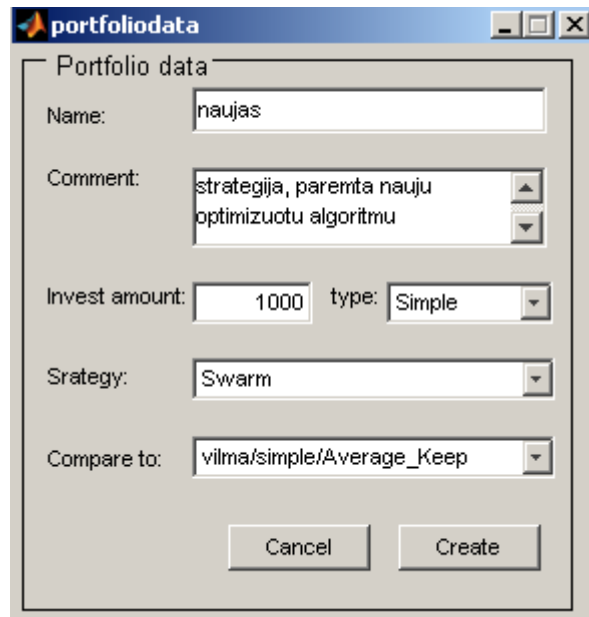
Šaltinis: sudaryta autoriaus

15 pav. Duomenų atnaujinimo langas

3. Trečiasis žingsnis yra naujo portfelio sukūrimas, naudojant meniu "Portfolio / Create new". Atsidariusioje formoje lauke *Strategy* pasirenkame konkrečią strategiją.

- *Conservative* – pagrindinė šios strategijos mintis yra tokia, kad padaryta investicija sulaikoma apibrėžtam laikotarpiui. Ši strategija naudojama atlikti lyginamąją analizę su kitais metodais;

- *Swarm* - prognozės atliekamos naudojant siūlomą sprendimo priėmimo metodą, kuris jau buvo pristatytas ankstesniuose skyriuose;
- *Average* - prognozės atliekamos naudojant slenkančių vidurkių metodą.

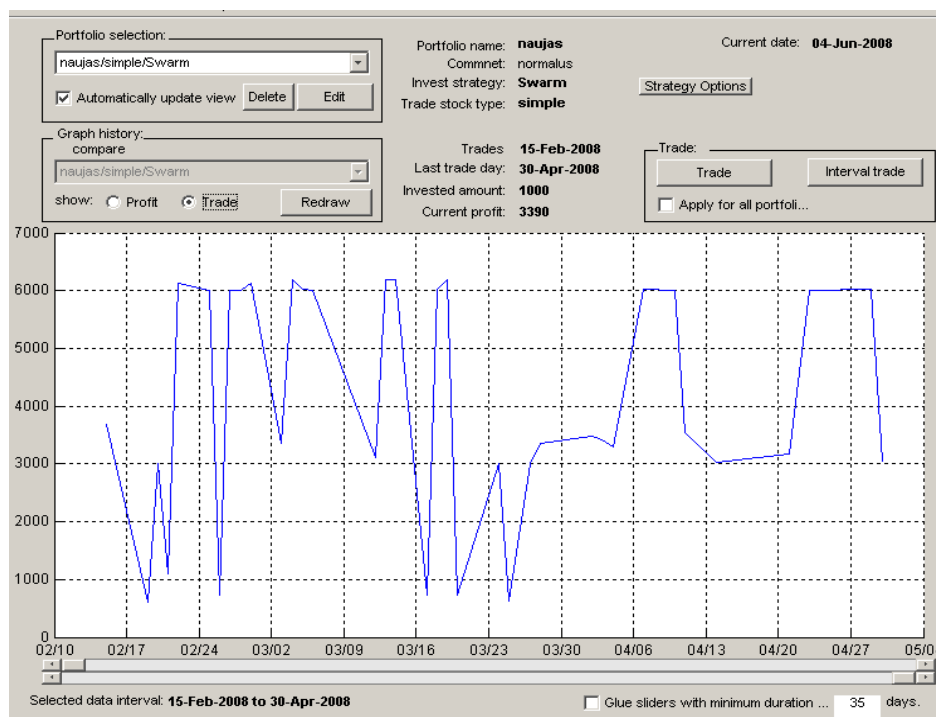


Šaltinis: sudaryta autoriaus

16 pav. Portfelio sukūrimo langas

Mano atveju pasirenku *Swarm* strategiją, kurios algoritmas ir buvo optimizuotas. Pasirenku investuojamą sumą bei lyginamąją strategiją.

4. Ketvirtas žingsnis yra tiesiog prognozavimas. Pagrindinis ir svarbiausias sistemos langas talpina informaciją, kuri yra būtina išvystytų algoritmų analizei ir skirtingų portfelių palyginimui. Sistemos langas pristatytas 17 paveikslėlyje.



Šaltinis: sudaryta autoriaus

17 pav. Pagrindinis sistemos langas

Pagrindiniame lange yra galimybė pasirinkti skirtingus portfelius lyginamajai analizei atlikti. Kiekvieno portfelio rezultatai lyginami su kito portfelio rezultatais (keičiant tam tikrus parametrus). Ši ypatybė leidžia atlikti dviejų skirtingų sprendimų priėmimo strategijų palyginimą. Norint pradėti prognozavimą, reikia paspausti mygtuką *Trade*, esantį pagrindiniame sistemos lange ir pasirinkti periodą prognozei atlikti.



Šaltinis: sudaryta autoriaus

18 pav. Prognozės langas

Kai pasirenkamas periodas yra galimybė matyti duomenų pasirinkimo procesą.



Šaltinis: sudaryta autoriaus

19 pav. Prognozavimo duomenų pasirinkimo procesas

3.5. Rezultatų patikimumo tikrinimas

Optimizavus algoritmą ir atliekant tyrimus, buvo gauti, tokie rezultatai, pavaizduoti sekančiuose paveikslėliuose.



Šaltinis: sudaryta autoriaus

20 pav. Istoriniai duomenys

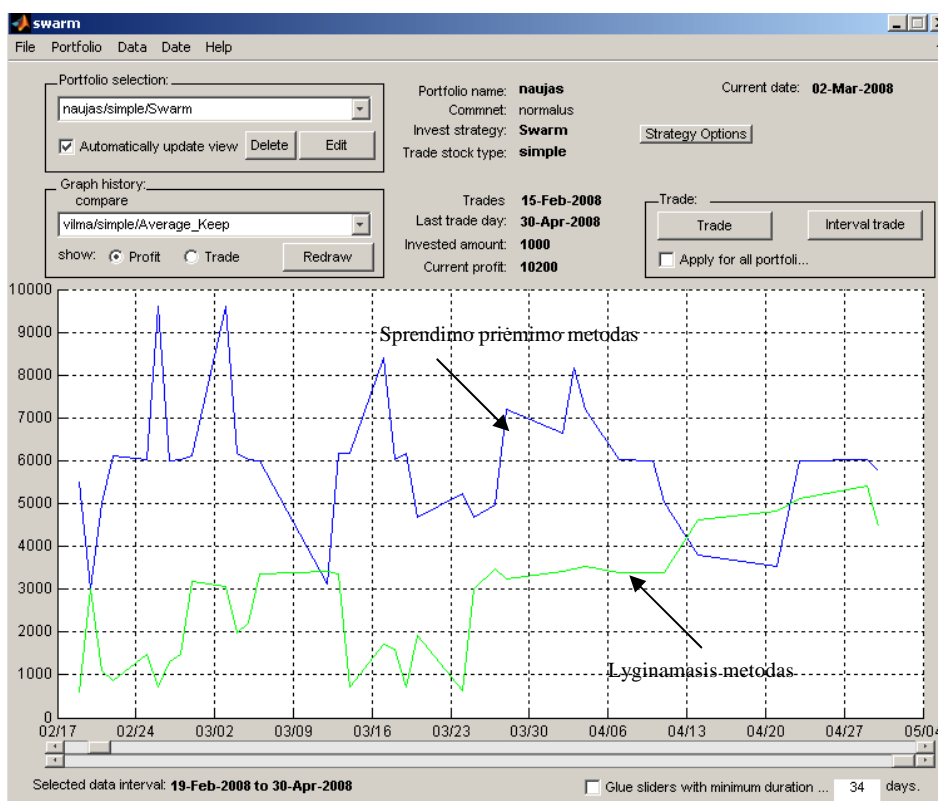


Šaltinis: sudaryta autoriaus

21 pav. Istoriniai duomenys

Buvo pasirinkti istoriniai duomenys prognozei atlikti. Priešingai nei J. Nenortaitė savo darbe buvo pasirinkti tokie duomenys, kuriuose neįžvelgiama akcijų skaidymo (Split). Istorinių duomenų analizės grafikai pateikiami 21 ir 20 paveikslėliuose. Būtent pavaizduoti, jog J. Nenortaitė neįvertino šios dalies prieduose pateikiami disertacijoje naudojamų duomenų analizės grafikai, juose pažymėtos susidariusios akcijų skaidymo vietos.

Iš ties akcijų skaidymas neturi jokios įtakos rinkos kapitalizacijai, fundamentaliems rodikliams, pelno ar pajamų didėjimui/mažėjimui, tačiau gali padidinti jos likvidumą, bei tuo pačiu minimaliai kilstelti akcijos kainą. Tačiau naujienos apie akcijų skaidymą neturėtų būti pagrindinis veiksnys, kuris įtakoja sprendimą pirkti būtent šias akcijas (ypač jei kaina po tokios naujienos jau pakilo).



Šaltinis: sudaryta autoriaus

22 pav. Pagrindinis sistemos langas

Naudojant anksčiau pristatytus metodus (strategijas), buvo sukurti du portfeliai su identiškais parametrais. Kaip jau buvo paminėta sistema leidžia pasirinkti prognozavimo periodą ir akcijas, kurios bus naudojamos prognozei atlikti.

Taigi 21 paveikslėlyje vaizduojami dviejų portfelių rezultatai. Kiekvienas portfelis apibūdinamas pagal pavadinimą, naudojamą strategiją, prognozės periodą, akcijas ir investavimo sumą. Sistemoje yra galimybė pakomentuoti kiekvieną portfelį, parinkti, kokia strategija bus naudojama prognozei atlikti bei parinkti prognozės periodiškumą.

Naudojant sprendimo priėmimo metodą yra galimybė pasirinkti rekomenduojamų akcijų skaičių investavimui, lango dydį, deltų skaičių ir neuroninių tinklų skaičių. Rezultatų analizei yra galimybė braižyti grafikus, kurie leidžia atlikti detalizuotą neuroninių tinklų apmokymo analizę, pasirinkti akcijas ar neuroninius tinklus, kurie parodo geriausią rezultatą.

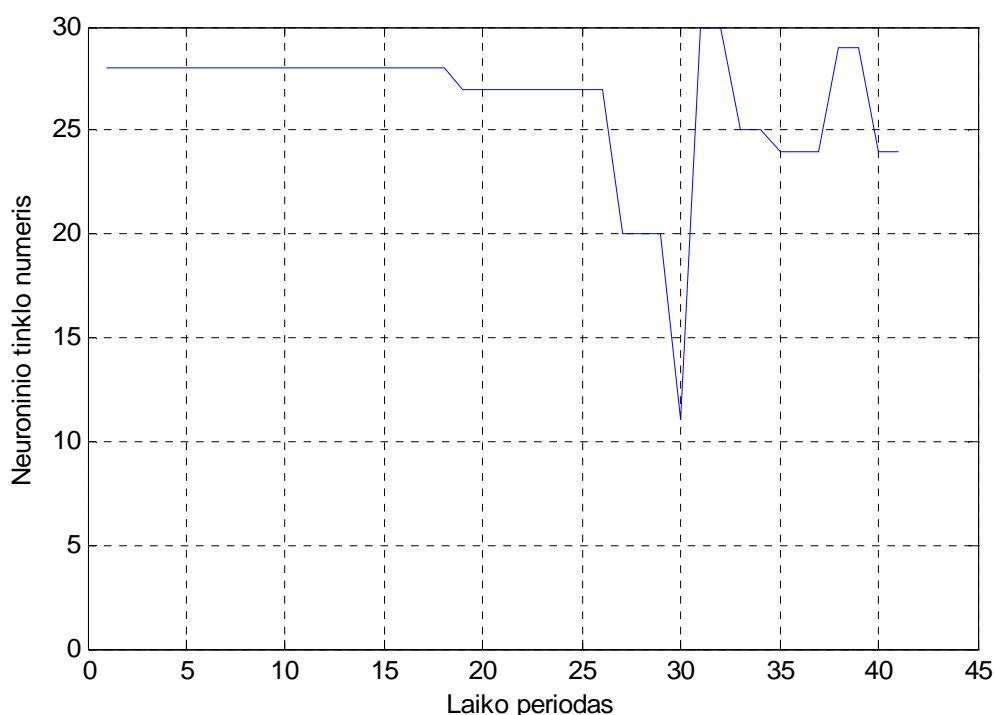
Pagrindiniame lange yra galimybė apkeisti strategijas. Kiekvieno portfelio rezultatai lyginami su kito portfelio rezultatais (keičiant tam tikrus parametrus). Ši ypatybė leidžia atlikti dviejų skirtingų sprendimų priėmimo strategijų palyginimą.

Kaip matome 21 paveikslėlyje, siūlomas sprendimo priėmimo metodas rodo geresnius prognozės rezultatus, kuris pažymėtas mėlyna spalva. Žalioji kreivė yra lyginamojo metodo kreivė.

Kaip matote iš grafiko, jog pasiūlyto sprendimo priėmimo kreivė rodo, jog pranašaujamas kylimas akcijų rinkoje, o lyginamojo metodo kreivė pranašauja žymiai mažesnę kylimą ateityje. Todėl aiškiai matyti, jog priimant sprendimą pirkti ar parduoti remsimės naujojo metodo prognozės grafiku. Taip pat matome, jog pritaikius naujai pasiūlytą metodą, ateityje žadamas didesnis pelnas.

Investavome 1000\$, o jau dabar žadamas pelnas 10200\$.

Sistema leidžia padaryti išsamią kiekvieno portfelio strategijos analizę. Apačioje yra pristatyti grafikai, kurie atstovauja sprendimo priėmimo metodo analizės galimybėms. Kadangi pagrindinė mintis apie pasiūlytą sprendimų priėmimo metodą yra neuroniniai tinklai ir spiečiaus intelektas, vartotojui būtų įdomu pamatyti pradinių svorių susiliejimą, kaip jie svoriai išsidėsto, kad galėtų būti pasirinkti neuroniniai tinklai.



Šaltinis: sudaryta autoriaus

23 pav. Neuroniniai tinklai

Neuroninis tinklas sprendimo priėmimui naudojamas vienai dienai. Kaip matote paveikslėlyje nuo pirmos iki 18 dienos geriausias tinklas buvo 28, nuo 20 dienos iki 28 dienos geriausias tinklas buvo 27 ir nuo 26 iki 29 dienos geriausias tinklas buvo 20. Taigi beveik kiekvienai dienai pasirenkamas naujas geriausias tinklas.

Dirbant su pasiūlytu sprendimo priėmimo metodu svarbu išvengti neuroninių tinklų svorių susiliejimo. Neuroninių tinklų mokymas atliekamas keičiant neuroninių tinklų svorius. Jei tas pats geriausias neuroninis tinklas išlieka ilgesnį laiką, yra galimybė, kad po kažkurio laiko svoriai sueis. Pagrindinis tikslas patikrinti, ar po kažkurio mokant neuroninį tinklą neturėsime atvejo, jog visi

svoriai pasieks tą patį tašką. Jei tai įvyktų, tai spiečiaus optimizavimo dalelės praras jutimą. Taikomo spiečiaus optimizavimo algoritmo privalumas yra toks, kad dalelės su blogu rezultatu nėra pašalinamos, kadangi jos gali parodyti kažkada vertingus rezultatus. Neuroninių tinklų svorių "judėjimas" yra pristatytas 2, 3 ir 4 prieduose. Būtent šiuose prieduose pristatyti trijų geriausių tinklų svoriai, atsižvelgiant į apibrėžtą laiko tarpą.

3.6. Metodų skirtumai ir panašumai

J. Nenortaitės pasiūlytas metodas, pagrįstas neuroninių tinklų ir spiečiaus algoritmo panaudojimu finansinių rinkų prognozėms atlikti. Savo darbe ji teigė ir įrodė, jog galutinis sprendimas "pirkti", "parduoti" ar visgi "laikyti" priimamas remiantis neuroninio tinklo "global best" (geriausia) rekomendacija. Prognozei atlikti buvo pasirinkti istoriniai duomenys nuo 1991 spalio 1 d. iki 2003 spalio 1 d.

J. Nenortaitė savo darbe neišvelgė į akcijų skaidymus (Split). Iš ties akcijų skaidymas neturi jokios įtakos rinkos kapitalizacijai, fundamentaliems rodikliams, pelno ar pajamų didėjimui/mažėjimui, tačiau gali padidinti jos likvidumą, bei tuo pačiu minimaliai kilstelti akcijos kainą. Tačiau naujienos apie akcijų skaidymą neturėtų būti pagrindinis veiksnys, kuris įtakoja sprendimą pirkti būtent šias akcijas (ypač jei kaina po tokios naujienos jau pakilo).

Mano siūlomas metodas, pagrįstas neuroninių tinklų ir spiečiaus algoritmo panaudojimu finansinių rinkų prognozėms atlikti, prognozę atlikti remiantis trijų geriausių rezultatų žadančių tinklų rezultatų vidurkiu. Rezultatų vidurkinimas pasirinktas, kadangi dirbama su dinamiškai besikeičiančiomis aplinkomis, todėl netgi „blogos“ dalelės gali duoti teisingą sprendimą ateityje, todėl nereikėtų pasikliauti tik viena geriausio tinklo rekomendacija. Atlikus analizę, pastebėta, jog finansinėje rinkoje sprendimo priėmimui PSO algoritmas yra tinkamiausias, kadangi PSO jis yra vienintelis evoliucinis algoritmas, kuris neapjungia natūraliosios atrankos, kuriai būdingas kai kurių populiacijos narių pašalinimas.

Prognozei atlikti buvo pasirinkti istoriniai duomenys nuo 1998 gegužės 1 d. iki 2008 gegužės 1 d., renkant duomenis buvo atsižvelgta į akcijų skaidymą (Split).

Atlikus analizę bei bandomuosius tyrimus, galiu teigti, jog mano pasiūlytas metodas taip pat gali būti pritaikomas finansinių rinkų prognozavimui. Hipotezę, jog remiantis algoritmo pritaikymo idėja, galima parinkti neuroninius tinklus ateities investicijų sprendimams, remiantis trijų geriausių rezultatų žadančių tinklų rezultatų vidurkiu galime priimti.

IŠVADOS IR PASIŪLYMAI

Spekuliavimas akcijomis aktyviose akcijų rinkose yra labai rizikingas, tačiau pelningas užsiėmimas. Tačiau prognozuoti rinkas pakankamai sudėtinga, kadangi pastebimi žymūs akcijų kursų svyravimai. Ženklaus akcijų kursų pasikeitimai skaičiuojami ne per metus ar mėnesius, o dienomis ar net valandomis. Investitoriams, finansų analitikams finansinėse rinkose sunku dirbti. O kontroliuoti pavojaus lygmenis yra būtina ir įvairioms kompanijoms ir finansų įstaigoms. Todėl didelis dėmesys baigiamajame darbe buvo skiriamas akcijų rinkų tyrimams ir finansinių rinkų prognozavimui naudojamų metodų ir įrankių analizei. Darbas susitelkia ties sprendimo priėmimo ieškojimu finansinių rinkų prognozavimui ir pasiūlomas galimas metodas, pagrįstas neuroniniais tinklais ir spiečiaus algoritmu.

Pristatytas darbas leidžia daryti tokias išvadas:

- Baigiamajame darbe buvo atlikta finansinių rinkų prognozavimui naudojamų sprendimų analizė. Atlikus analizę, pastebėta, jog pirminė tyrėjų užduotis - sukurti tokį įrankį, kuris galėtų būti naudojamas vertybinių popierių rinkų svyravimų prognozei. Efektyvūs įrankiai ir protingi patarėjai svarbūs efektyviam ir naudingam darbui finansinėse rinkose. Tai skatina kompiuterizuotų sprendimų priėmimo modelių pasirinkimą ir investavimo procesas techniniu požiūriu supaprastėja ir nereikalauja didelių sąnaudų, bei gilių žinių, leidžia platesniam ratui žmonių įeiti į akcijų rinką;
- Skirtingų autorių pristatyti sprendimo priėmimo metodai yra labai skirtingi;
- Atliktas PSO ir GA algoritmų palyginimas parodė, jog PSO algoritmas yra tinkamesnis prognozavimui;
- Buvo pastebėta, jog sudėtinga surasti realius rezultatus, gautus pritaikius tam tikrus sprendimo priėmimo metodus, t.y. jog blogi rezultatai nepublikuojami, pateikiami tik gerus rezultatus vaizduojantys duomenys;
- Atlikus analizę finansinių rinkų prognozei atlikti buvo pasiūlytas sprendimo priėmimo metodas, pagrįstas neuroninių tinklų ir spiečiaus algoritmo panaudojimu finansinių rinkų prognozėm atlikti, prognozę atlikti remiantis trijų geriausių rezultatų žadančių tinklų rezultatų vidurkiu;
- Pateiktas ir eksperimentiniais duomenimis patikrintas optimizuotas algoritmas;
- Metodo patikrinimui atlikti buvo pritaikyta operatyvi sistema ir panaudota skirtingų sprendimų priėmimo modelių ir algoritmų išsivystymui ir analizei. Taip pat buvo pateikta

sistemos architektūra, duomenų bazės paaiškinimas, iliustracijos ir sistemos aprašymas. Pristatyta akcijų rinkos sistema – įrankis, skirtas išsivystymui ir sprendimų priėmimo modelių suderinimui;

- Pasiūlyto sprendimo metodo bandomiesiems tyrimams atlikti buvo pasirinkta Swarm 19.6 sistema. Tačiau ją buvo sunku pritaikyti, dėl sistemos aprašymų metodinės medžiagos trūkumo. Sistema sukurta MATLAB ir MYSQL įrankių pagalba, tačiau programinis kodas yra labai sudėtingas. Įžvelgtas automatizavimo trūkumas, programa nepatogu naudotis. Programą dar būtų galima tobulinti;
- Visada neatmetama galimybė tobulinti tiek sistemą, tiek algoritmą, nes visgi nereikėtų pamiršti, kad laikui bėgant vartotojai tampa reiklesni, todėl norint sukurti tobulesnį produktą, reikia nuolat analizuoti ir orientuotis į vartotojų poreikius, ugdyti bendruosius profesinius bei lyderystės įgūdžius ir intelektualinius sugebėjimus.
- Atliekant eksperimentus su pasiūlytu metodu buvo susidurta su tam tikromis problemomis:
 - Didelis kiekis duomenų ir sudėtingi skaičiavimai neleidžia garantuoti greito sprendimo, kuris būtinas dirbant finansinėse rinkose;
 - Skaičiavimo įtaka. Norint pasiekti priimtinius rezultatus, skaičiavimo procesas turi būti greitas. Didelis kiekis duomenų ir sudėtingas algoritmo skaičiavimas nesuteikia mums garantijos dėl pasiūlyto sprendimo - metodo sukūrimas;
 - Galimybė atlikti kelias metodo realizacijas tuo pačiu metu. Kad garantuotume gerą sprendimą - metodo sukūrimą, turime atlikti bandomuosius tyrinėjimus beveik su kiekvienu kintamuoju kiekvienu atveju;
 - Duomenų pateikimas. Šiuo metu yra grupė tyrėjų, dirbančių ties ta pačia tema: vertybinių popierių analizė, prognozė ir sprendimo priėmimas. Dauguma tyrėjų naudoja tuos pačius duomenis bandomiesiems tyrinėjimams. Galimybė gauti prieigą prie to paties duomenų rinkinio toje pačioje vietoje galėtų garantuoti geresnį skaičiavimo procesą ir galėtų apsaugoti tyrėjus nuo nereikalingų veiksmų, tokių kaip duomenų rinkinio sudarymas, kopijavimas ir taip toliau.
- Lyginant su kitais egzistuojančiais sprendimais šis sprendimų priėmimo metodas yra geresnis dėl savo lankstumo. Jis gali būti pritaikomas daugelyje kitų sričių, tokių kaip telekomunikacija, logistika ir taip toliau. Svarbiausia, jog pasiūlytu sprendimų priėmimo metodu gali naudotis skirtingi vartotojai (investiciniai fondai, nepriklausomi vartotojai ar brokeriai), priklausomai nuo jų finansinių galimybių ir prekybos mastų, sumų. Vartotojas gali išrinkti prekybos strategiją, gali keiti akcijas ar indeksus.
- Atlikus analizę bei bandomuosius tyrimus, galiu teigti, jog mano pasiūlytas metodas gali būti pritaikomas finansinių rinkų prognozavimui. Hipotezę, jog remiantis algoritmo

pritaikymo idėja, galima parinkti neuroninius tinklus ateities investicijų sprendimams, remiantis trijų geriausių rezultatą žadančių tinklų rezultatų vidurkiu galime priimti.

Konkurencija Lietuvos rinkoje tarp panašaus pobūdžio sprendimo priėmimui naudojamų sistemų yra menka, galime teigti, kad jos beveik nėra. Daugelis investuotojų renkasi užsienio šalių rinkos produkciją, galbūt todėl, kad Lietuvoje panašių sistemų beveik nėra. Taigi, atsižvelgiant į pasiūlytos sistemos funkcionalumą, efektyvumą, pritaikyto algoritmo galimybes reikia siūlyti ją Lietuvos rinkai ir ateityje ji turėtų sulaukti didelio pasisekimo.

LITERATŪRA

1. ANDREAS, Z. (1994) *Simulation Neuronaler Netze*, Addison-Wesley, ISBN 3-89319-554-8.
2. BACK, T. (1996) *Evolutionary Algorithms Theory and Practice: Evaluation Strategies, Evolution Programming, Genetic Algorithms*, Oxford University Press.
3. BERESNEVIČIENĖ, R. (2008) *Investavimo psichologijos labirintai* [interaktyvus]. bs-b.lt, [žiūrėta 2007m. gruodžio 20 d.]. Prieiga per internetą: <http://www.bs-b.lt/lt/straipsniai/article_detail.php?id=28393>.
4. BENTLEY, J. ir UJJIN, S. (2002) *Particle swarm optimization recommender system* [interaktyvi duomenų bazė]. IEEE, [žiūrėta 2007 lapkričio 15 d.]. Prieiga per Vilniaus universiteto biblioteką.
5. BLANCHET-SCALLIET, C.; DIOP, A.; GIBSON, R.; TALAY, D. ir TANRĖ, E. (2004) *Technical Analysis Techniques versus Mathematical Models: Boundaries of Their Validity Domains* [interaktyvus]. sop.inria.fr. [žiūrėta 2008m. balandžio 29 d.]. Prieiga per internetą:<www-sop.inria.fr/omega/personnel/Etienne.Tanre/publication/blanchet_all.pdf>.
6. CIBULSKIENĖ, D.; GRIGALIŪNIENĖ, Ž. (2006) *Fundamentinių ir techninių veiksnių įtaka vertybinių popierių portfelio formavimui* [interaktyvus]. smf.su.lt, [žiūrėta 2008m. balandžio 29 d.]. Prieiga per internetą: <[www.smf.su.lt/documents/konferencijos/Galvanauskas/2006/Leidinys%20\(7\)/Cibulskiene_Grigaliuniene.pdf](http://www.smf.su.lt/documents/konferencijos/Galvanauskas/2006/Leidinys%20(7)/Cibulskiene_Grigaliuniene.pdf)>.
7. CICHOCKI, A., UNBEHAUEN, R. (1993) *Neural Networks for optimization and Signal Processing*. NY: John Wiley & Sons. 526 p., ISBN 0-471-930105.
8. DEMPSTER, M. ir JONES, M. (2000) *A real-time adaptive trading system using genetic programming* [interaktyvus]. mahd-pc.jbs.cam.ac.uk [žiūrėta 2008m. gegužės 20 d.]. Prieiga per internetą: <mahd-pc.jbs.cam.ac.uk/archive/papers/2000/geneticprogramming.pdf>.
9. EBERHART, R. C. ir SHI, Y. (1998) *Comparison between genetic algorithms and particle swarm optimization* [interaktyvi duomenų bazė]. IEEE, [žiūrėta 2007 lapkričio 15 d.]. Prieiga per Vilniaus universiteto biblioteką.
10. ENGELBRECHT, A. D. (2002) *Computational intelligence (an introduction)*, London: John Wiley and Sons Inc. 288 P., ISBN 0-470-84870-7.

11. EBERHART, R. C., KENNEDY, (1995) *J. A new optimizer using particle swarm theory*, 6th International Conference on "Micromachine and Human Science" .
12. FENTRESS, S. (2005) *Exaptation as a means of evolving complex solutions*, ma thesis, University of Edinburgh.
13. FOGEL, L. J., OWENS, A. J., IR WALSH, M. J. (1966) *Artificial intelligence through simulated evolution*. John Wiley, New York.
14. GARŠVA, G., SEKLIUCKIS, V., GUDAS, S. (2003) *Informacijos sistemos ir duomenų bazės*. – Kaunas: Technologija. – 338 p. ISBN 9955-09-486-9.
15. HASSAN, R., COHANIM, B., WECK, O. (2004) *A comparison of particle swarm optimization and the genetic algorithm* [interaktyvus].strategic.mit.edu, [žiūrėta 2008 m. balandžio 15 d.]. Prieiga per internetą: <strategic.mit.edu/PDF_archive/3%20Refereed%20Conference/3_50_AIAA-2005-1897.pdf>.
16. HEPPNER, F. ir U. GRENANDER (1990) *A stochastic nonlinear model for coordinated bird flocks* [interaktyvi duomenų bazė]. AAAS [žiūrėta 2008 kovo 15 d.]. Prieiga per Vilniaus universiteto biblioteką.
17. HOLLAND, J. (1975) *Adaptation in natural and artificial system*, Ann Arbor, the university of Michigan press.
18. KANCEREVIČIUS, G. (2004) *Finansai ir investicijos*, Kaunas: Smaltija. 880 p. ISBN 9955-551-40-2.
19. KENNEDY, J. ir SPEARS, W. M. (2003) *Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator*, WCCO.
20. KENNEDY, J. ir EBERHART, R. (2004) *Particle swarm optimization* [interaktyvi duomenų bazė]. IEEE, [žiūrėta 2008 sausio 15 d.]. Prieiga per Vilniaus universiteto biblioteką.
21. KENNEDY, J. ir EBERHART, R. (2001) *Swarm intelligence* [interaktyvi duomenų bazė]. IEEE, [žiūrėta 2008 vasario 25 d.]. Prieiga per Vilniaus universiteto biblioteką.
22. MARTAVIČIUS, R.; URBANAVIČIUS, V.; NAVAKAUSKAS, D.; PAULIKAS, Š. (2008) *Dirbtinių neuronų tinklų (DNT) teorijos raida* [interaktyvus]. vgtu.lt, [žiūrėta 2008m. sausio 15 d.]. Prieiga per internetą: <<http://www2.el.vgtu.lt/ssa/node1.html>>.

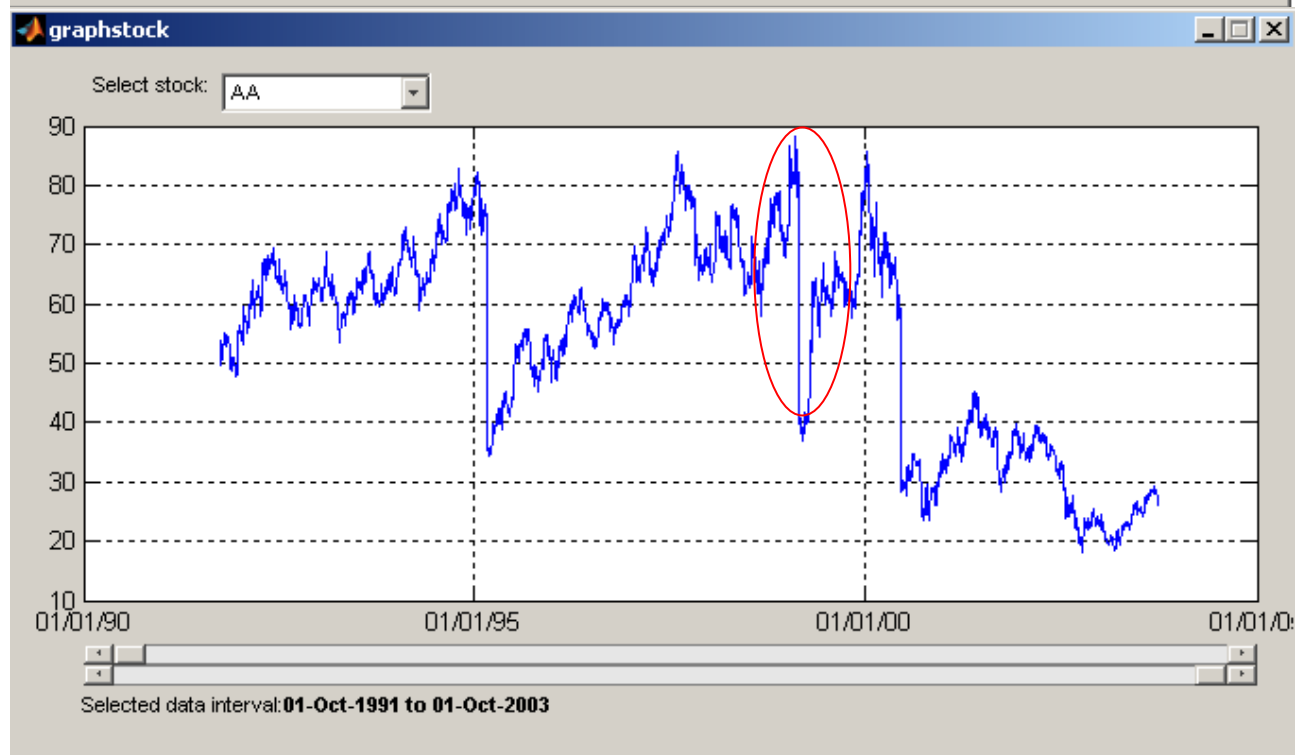
23. MATLAB 7.0.1 (2004) *Technical programming language*, MathWorks.
24. MENDES, M., KENNEDY, J., ir NEVES, J. (1997) *Watch Thy Neighbor or How the Swarm Can Learn from Its Environment* [interaktyvi duomenų bazė]. IEEE, [žiūrėta 2008 kovo 15 d.]. Prieiga per Vilniaus universiteto biblioteką.
25. MILLER, P. (2007) *Swarm Behavior*, [interaktyvus]. cs.iastate.edu, [žiūrėta 2008 m. kovo 15 d.]. Prieiga per internetą: <<http://ngm.nationalgeographic.com/ngm/0707/feature5/text2.html>>.
26. MITCHELL, M. (1997) *An introduction to genetic algorithms*, Cambridge, 209p. ISBN 0-262-13316-4.
27. MYSQL 4.1., The world's most popular open source database [interaktyvus]. mysql.com [žiūrėta 2008m. sausio 22 d.]. Prieiga per internetą: <<http://www.mysql.com/>>.
28. NENORTAITĖ, J. (2006) *Stock trading system based on application of swarm intelligence and artificial neural networks*: doctoral dissertation: physical sciences, informatics. Vilniaus Universitetas, Kauno Humanitarinis fakultetas. 107 p.
29. NYGREN, K. (2004) *Stock Prediction – A Neural Network Approach* [interaktyvi duomenų bazė]. IEEE, [žiūrėta 2008 kovo 15 d.]. Prieiga per Vilniaus universiteto biblioteką.
30. PARSOPOULOS, K. ir VRAHATIS, M. (2001) *Particle swarm optimizer in noisy and continuously changing environments* [interaktyvi duomenų bazė]. AAAS [žiūrėta 2008 kovo 15 d.]. Prieiga per Vilniaus universiteto biblioteką.
31. RECHENBERG, I. (1973) *Evolutionstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart.
32. SCHWEFEL, H. P. (1981) *Numerical Optimization of ComputerModels*, John Wiley, Chichester, U.K.
33. SIMUTIS, R. (2007) *Intelektualios sistemas finansų rinkose*, paskaitų konspektai.
34. STANDARD&POOR‘S [interaktyvus]. [žiūrėta 2007m. gruodžio 29 d.]. Prieiga per internetą: <<http://www2.standardandpoors.com>>.
35. STANDARD&POOR‘S (2008) S&P 500 [interaktyvus]. [žiūrėta 2007m. gruodžio 29 d.]. Prieiga per internetą: <www2.standardandpoors.com/spf/pdf/index/500factsheet.pdf>.

36. WHITLEY, D. (1994) *A genetic algorithm tutorial* [interaktyvus].cs.iastate.edu, [žiūrėta 2007m. vasario 15 d.]. Prieiga per internetą: <www.cs.iastate.edu/~honavar/ga_tutorial.pdf>.
37. WHITE, H. (1989) *Learning in neural networks: A statistical perspective*, Neural Computat 4
38. WILSON, E.O. (1975) *Sociobiology: The new synthesis*. Cambridge, MA: Belknap Press.
39. WINNERSTOCKPICKS, *Daily trading system* [interaktyvus]. WinnerStockPicks.com. [žiūrėta 2007m. gruodžio 29 d.]. Prieiga per internetą: <<http://www.winnerstockpicks.com> >.
40. WIKIPEDIA / The Free Encyclopedia, *Evolutionary algorithm*. [interaktyvus]. wikipedia.org. [žiūrėta 2007m. vasario 15 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Evolutionary_algorithm>.

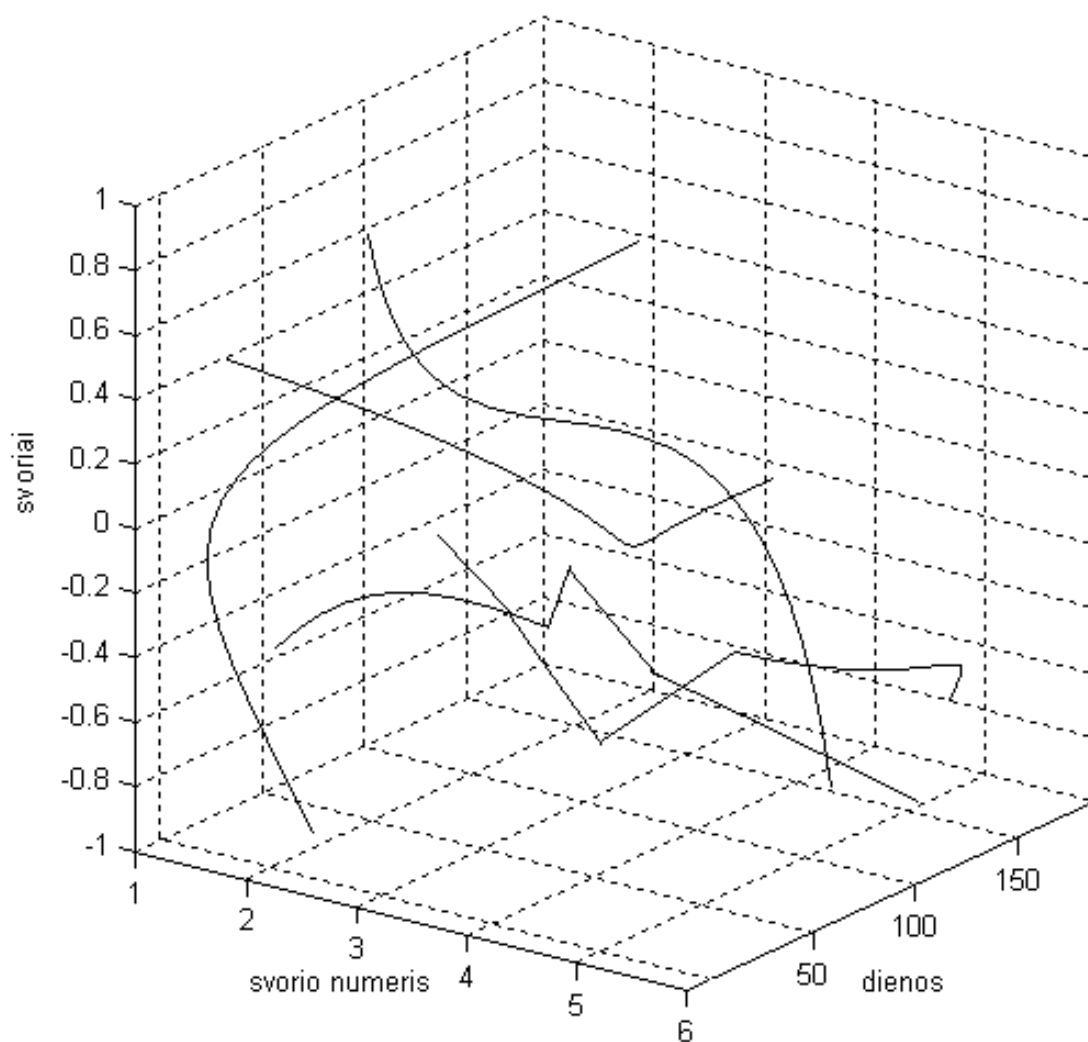
PRIEDAI

1 PRIEDAS PROGNOZAVIMO DUOMENYS	63
2 PRIEDAS 20 TINKLO ANALIZĖ	64
3 PRIEDAS 27 TINKLO ANALIZĖ	65
4 PRIEDAS 28 TINKLO ANALIZĖ	66
5 PRIEDAS OPTIMIZUOTO ALGORITMO PROGRAMINIS KODAS.....	67

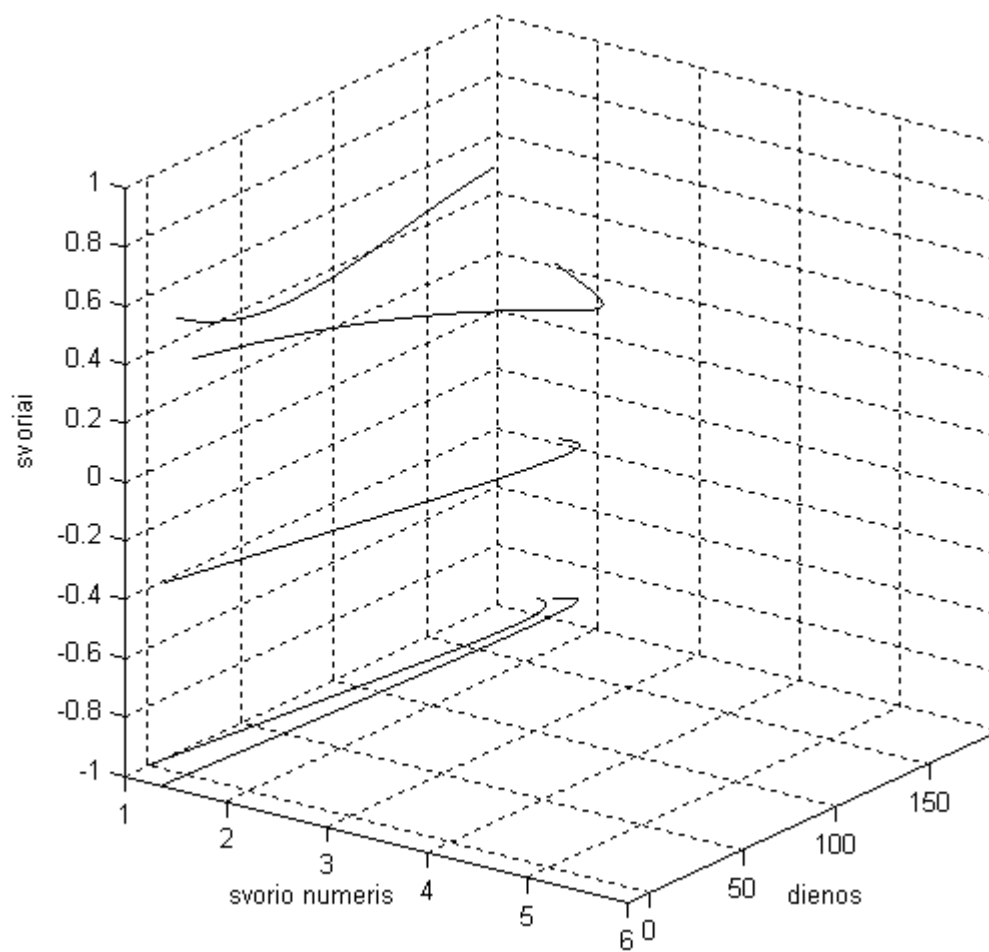
PROGNOZAVIMO DUOMENYS



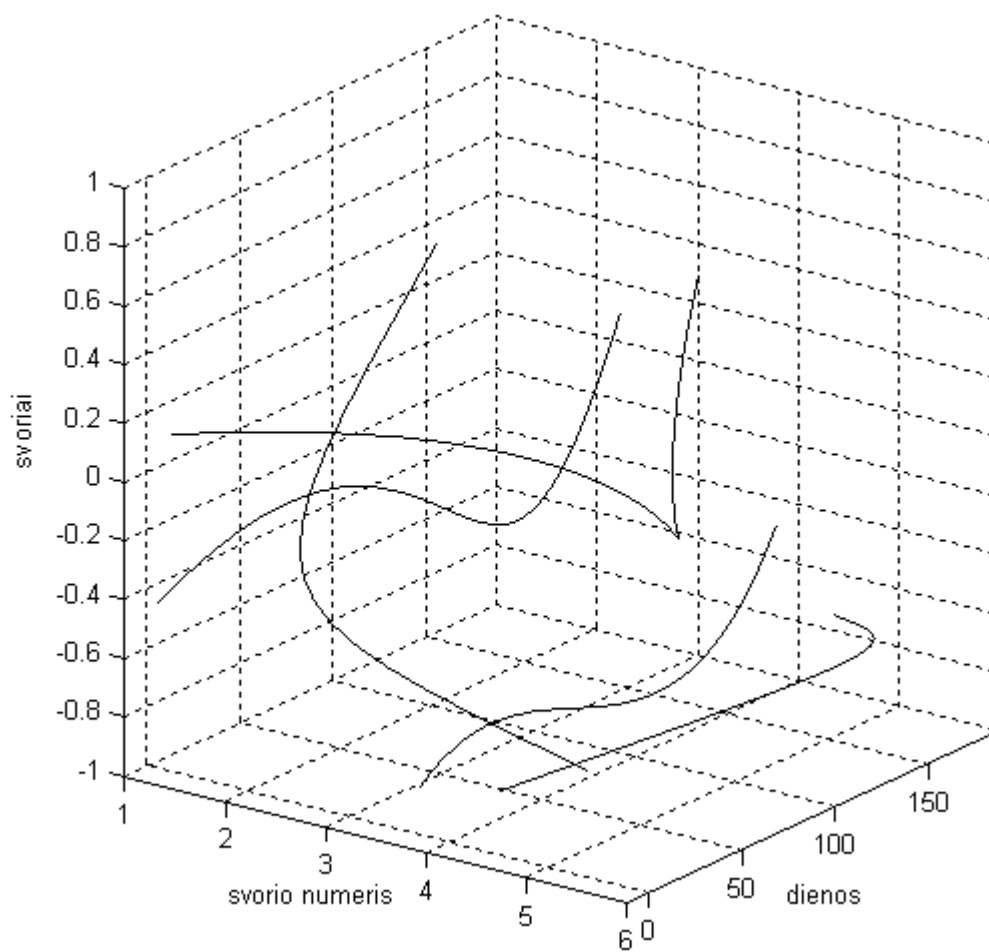
20 TINKLO ANALIZĖ



27 TINKLO ANALIZĖ



28 TINKLO ANALIZĖ



OPTIMIZUOTO ALGORITMO PROGRAMINIS KODAS

```

clear all, close all
t=cputime;

totalnets=30;
k_dienu=5;

i_cumm=5;
deltu_sk=5

yahoolist = loadstockfile('stocklist_test.txt');
stockyahoodid = [yahoolist.indexlist;yahoolist.stocklist];

akc = size(stockyahoodid,1);

deltawindow = i_cumm;
stockwindow = i_cumm+k_dienu;
fromdate = datenum('22-FEB-2008');
todate = datenum('30-APR-2008');

passwd = 'root';
mysql('open','localhost:3306','root',passwd);
mysql('use stockNN');
[stockdata, deltadata] = LoadData(stockyahoodid,stockwindow,deltawindow,fromdate,todate);
mysql('close');

idienos=size(stockdata.data,2);
duom(:,1) = stockdata.data;
startas=1+k_dienu;
for k=1:5
    for i=startas:idienos
        for j=1:akc
            delta(j,i-k_dienu,k)=(duom(j,i,1)-duom(j,i-k,1))/duom(j,i-k,1)*100;
        end
    end
end

delta = deltadata.delta;
a = round(delta*10000) - round(deltadata.delta*10000);

display('kiek sutampa');
size(find(abs(a) == 0))
display('kiek nesutampa');
size(find(abs(a) > 0))
display('kiek nesutampa paklaida > 0.0001');
size(find(abs(a) > 1))

delta = deltadata.delta;
idienos=size(delta,2);

clear M;
id=idienos-k_dienu
akciju_sk=size(stockdata.data,1);

tmp_w=rand(deltu_sk,totalnets)-0.5;
suma=zeros(akciju_sk,3027,2);
rekomendacija=zeros(akciju_sk,3022);

```

```

ir=0;
for i=1:id-1
    for net=1:totalnets
        for j=1:akc
            suma(j,i,net)=0;
            for k=1:5
                suma(j,i,net)=suma(j,i,net)+delta(j,i,k)*tmp_w(k,net);
            end
            rekomendacija(j,i,net)=(2./(1+2.71828^(-suma(j,i,net))))-1;
        end
    end

    for net=1:totalnets
        [Y,I]=sort(rekomendacija(:,i,net));
        Y=flipud(Y);
        I=flipud(I);
        for j=1:akc
            YY(j,i,net)=Y(j);
            II(j,i,net)=I(j);
        end
    end

    k=1;
    for net=1:totalnets
        sumrek(1,net)=0;
        for no=1:3
            p=0;
            p=p+delta(II(no,i,net),i+1,1);
        end
        sumrek(i+1,net)=p/3;
    end
    in=i;
    delta_tomorrow(1)=0;
    SP_tomorrow(1)=0;
    tmp=0;
    if in < i_cumm+1
        for net=1:totalnets
            maxint(in,net)=0;
        end
    else
        for net=1:totalnets
            maxint(in,net)=sum(sumrek((in-i_cumm):in,net));
            tmp(net)=maxint(in,net);
        end
    end

    if i >= i_cumm+1
        ir=ir+1;
        [Y1,I1]=sort(tmp);
        Y1=fliplr(Y1);
        I1=fliplr(I1);
        bestnet(ir)=I1(1);
        for net=1:totalnets
            if net ~= bestnet
                for k=1:5
                    tmp_w(k,net) = tmp_w(k,net) + 0.05*(tmp_w(k,bestnet(ir))-tmp_w(k,net));
                end
            end
        end

        delta_sum_t=0;
        sp_sum_t=0;

```

```

for loop=1:3
    bestnet(ir)=II(loop);
    delta_sum_t=delta_sum_t + mean(delta(II(1:3,i,bestnet(ir)),i+1,1))-0.15;
    sp_sum_t=sp_sum_t + delta(akc,i+1,1);
end

    delta_tomorrow(ir+1)=delta_sum_t/3;
    SP_tomorrow(ir+1)=sp_sum_t/3;
end
end
clear sumrek;
clear rekomendacija;
clear maxint;

KapS(1)=1000;
KapSP(1)=1000;
ir=0;

for i=i_cumm+1:id-1
    ir=ir+1;
    KapS(ir+1)=KapS(ir)+(KapS(ir)*delta_tomorrow(ir+1))/100;
    KapSP(ir+1)=KapSP(ir)+(KapSP(ir)*SP_tomorrow(ir+1))/100;
end;

deltu_cumsuma=cumsum(delta_tomorrow);
SP_cumsuma=cumsum(SP_tomorrow);
fprintf('Kumulatyviniai procentai,Swarm'),deltu_cumsuma(end)
fprintf('Kumulatyviniai procentai,SP500'),SP_cumsuma(end)
bendras=mean(delta_tomorrow);
bendrasSP=mean(SP_tomorrow);
fprintf('Vidutiniai procentai, per diena ,Swarm'),bendras
fprintf('Vidutiniai procentai, per diena ,SP500'),bendrasSP

figure(1)
plot(SP_cumsuma,'r'),grid,hold on
plot(deltu_cumsuma)
figure(2)
plot(KapSP,'r'),grid,hold on
plot(KapS)

cputime-t

```