

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Ronald Ruusa 155553IAPB

**VÄÄRTPABERITEHINGUTE AJAS
HAJUTAMISE ALGORITMID JA NENDE
RAKENDAMINE**

bakalaureusetöö

Juhendaja: Tõnn Talpsepp
PhD

Tallinn 2021

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Ronald Ruusa

05.01.2021

Annotatsioon

Selle bakalaureusetöö eesmärgiks on koostada väärtpaberitehingute ajas hajutamise algoritm, mis tükeldab suure tehingukorralduste väiksemateks tükideks ning luua programm, mis rakendab algoritmi. Algoritmi efektiivsuse testimiseks saadetakse simuleeritud aktsiaturule ostukorraldused, kus kasutatakse algoritmi ning ostukorraldused, kus algoritmi ei kasutata ning võrreldakse kahe meetodi tulemusi.

Töö eesmärgi täitmiseks loodud programmis kasutatakse algoritmi realiseerimiseks tuntud maakleri pakutavat APIt. API abil luuakse ühendus turuga, kuhu on võimalik algoritmi töö käigus loodud tükeldatud tehingukorraldused turule saata ning need teostada. Korralduse teostamisel saadud andmed salvestatakse programmi abil faili, mille abil töödeldakse ning analüüsitakse saadud tulemusi.

Töös loodud programmi on võimalik integreerida mitmesugustesse süsteemidesse, kus on vaja automatiseerida tehingukorralduste turule saatmist või implementeerida korralduse ajas hajutamist. Samuti on võimalik loodud programmi edasi arendada, lisades sellele lisaväärtusi ning -funktsionaalsusi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 5 peatükki, 7 joonist, 7 tabelit.

Abstract

Implementation and execution of order execution algorithms for security transactions

The first purpose of this thesis is to create an order execution algorithm that splits a stock order into pieces as effective as possible. The second purpose is to create a program that executes said algorithm on a simulated market. To evaluate the effectiveness of the algorithm, tests are carried out. In the tests, two types of orders are sent to a simulated market: orders that use the algorithm and orders which do not use the algorithm, and the results will be compared.

To fulfil the purpose of this thesis, the program will use an API from a known broker to execute the algorithm. With the help of the API, a connection is created with the stock market, which makes it possible to execute the algorithm and send its outcome to the market. Order data that is received from the market is saved into a file, which is used to analyse the data.

The program that is created in this thesis can be integrated into numerous systems, where it is necessary to optimise order placement or to automatize order placements. In addition, the program can be further developed to add new features or functionalities.

The thesis is in Estonian and contains 25 pages of text, 5 chapters, 7 figures, 7 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
CSV	<i>Comma-separated values</i> , failitüüp
IB	<i>Interactive Brokers</i> , firma
POV	<i>Percent of volume</i> , kauplemissalgoritmi tüüp
TWAP	<i>Time weighted average price</i> , kauplemissalgoritmi tüüp
VWAP	<i>Volume weighted average price</i> , kauplemissalgoritmi tüüp
TWS	<i>Trading Workstation</i> , programm

Sisukord

1 Sissejuhatus	10
2 Teoreetiline ülevaade.....	11
2.1 Väärtpaber	11
2.2 Aktsiaorder ja selle liigid.....	12
2.3 Algoritmiline kauplemine.....	13
2.3.1 Ülevaade algoritmilisest kauplemisest	13
2.3.2 Latentsus.....	14
2.3.3 Kõrgsagedusega kauplemine	14
2.3.4 Tehingunimekiri ja kauplemissüsteemi etapid	15
2.3.5 Andmed, nende puhtus ning analüüs.....	15
2.3.6 Korralduse täitmise algoritmid	16
2.3.7 Ajas hajutamise algoritmid.....	16
3 Tehniline lahendus.....	19
3.1 Ülevaade	19
3.2 Kasutatud tehnoloogiad	19
3.2.1 TWS.....	19
3.2.2 IB API.....	20
3.3 Programmi ülesehitus	21
3.3.1 Stock	21
3.3.2 Algorithm	22
3.3.3 FileHandler	22
3.3.4 ConnectionHandler.....	22
3.3.5 Main.....	23
3.4 Programmi töötamise skeem	23
3.5 Ajas hajutamise algoritm	24
3.6 Algoritmi tööprotsessi käigus kasutatavad parameetrid.....	25
3.6.1 Kasutatavad arvulised parameetrid.....	25
3.6.2 Kasutatavad arvutatavad parameetrid.....	27
4 Katsed ja analüüs	28

4.1 Katsete ülesehitus	28
4.2 Katsete tulemused.....	29
4.2.1 Microsofti aktsiate ostu tulemused.....	29
4.2.2 Nvidia aktsiate ostu tulemused.....	31
4.3 IB demo konto piirangud ja nende mõju katsete tulemustele.....	32
4.4 Algoritmi integreerimine	34
5 Kokkuvõte	35
Kasutatud kirjandus	36
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	38

Jooniste loetelu

Joonis 1. TWAP kauplemisalgoritmi kauplemismaht päeva jooksul [11].	17
Joonis 2. VWAP kauplemisalgoritmi kauplemismaht päeva jooksul [11].	17
Joonis 3. POV kauplemisalgoritmi kauplemismaht päeva jooksul [11].	18
Joonis 4. TWS vaade aktiivse ostukorraldusega.	20
Joonis 5. Programmi töötamise skeem.	23
Joonis 6. Kasumi jaotus protsentuaalselt MSFT aktsia puhul.	30
Joonis 7. Kasumi jaotus protsentuaalselt NVDA aktsia puhul.	32

Tabelite loetelu

Tabel 1. TWSis kuvatavad andmed.....	20
Tabel 2. Algoritmi töös kasutatavad eelnevalt fikseeritud arvulised parameetrid.	26
Tabel 3. Algoritmi töös kasutatavad kauplemismahud ning volatiilsused.....	27
Tabel 5. Algoritmi töös kasutatavad arvutatavad parameetrid.....	27
Tabel 6. Microsofti aktsiate ostu tulemused.....	30
Tabel 7. Nvidia aktsiate ostu tulemused.....	31

1 Sissejuhatus

Kui mõned kümnendid tagasi toimus kogu aktsiaturul kauplemine vahendajate kaudu ning elektrooniliselt teostatud lahendusi oli vähe, siis tänapäeval toimub valdav enamus kauplemismahust elektrooniliselt. Populaarseks on saanud algoritmiline kauplemine, mis kujutab endast spetsiaalsete kauplemisalgoritmide kasutamist kauplemisprotsessis. Kauplemisel kasutatavaid algoritme on erinevaid ning nende keerukus varieerub tohutult.

Ajas hajutamise algoritm on üks kauplemisalgoritmide liike ning seda on optimaalsete kauplemistulemuste saamiseks mõistlik kasutada. Ajas hajutamise algoritm kujutab endast algoritmi, mis tükeldab suure tehingukorralduse väiksemateks tükideks. Kui saata suuremahuline ostukorraldus börsile ühes tükis, mõjutab korralduse suurus turu hetkeseisu ning aktsiad võidakse saada kätte halvema hinnaga. Kuna ajas hajutamise algoritm tükeldab suure korralduse väiksemateks osadeks, ei mõjuta nende börsile saatmine turu tasakaalu, mille tulemusena võidakse tehing sooritada parema hinnaga.

Antud töö eesmärgiks on koostada väärtpaberitehingute ajas hajutamise algoritm ning luua programm, millega on võimalik algoritmi rakendada. Algoritmi efektiivsuse testimiseks viiakse läbi katsed, kus esitatakse korraldused simuleeritud aktsiaturule nii algoritmi realiseerides, kui ka algoritmi realiseerimata ehk ühes tükis. Katsete tulemusena leitakse algoritmi efektiivsus, mida võrreldakse tulemustega, kus algoritmi ei kasutatud.

Käesoleva töö käigus käsitletakse väärtpaberitest aktsiaid ning realiseeritakse ostukorraldust. Lõputöös on kasutusel Interactive Brokersi börsiprogramm Trading Workstation ning ka firma poolt pakutav API, mille abil on võimalik algoritmi börsil rakendada.

Töö jaguneb kolmeks osaks. Esimeses osas antakse ülevaade erinevatest väärtpaberitest ning nende tehingukorralduste erinevatest tüüpidest. Samuti antakse ülevaade algoritmilisest kauplemisest ning kauplemisalgoritmidest. Töö teises osas antakse ülevaade töö tehnilisest ülesehitusest ning kasutatavatest lahendustest. Töö kolmandas osas realiseeritakse algoritmi katsetega ning analüüsitakse saadud katsete tulemusi.

2 Teoreetiline ülevaade

2.1 Väärtpaber

Väärtpaber on finantsiline instrument, millel on mingisugune rahaline väärtus ning mida saab osta, müüa või vahetada. Selle omamine annab õiguse omada osa ettevõtte varast. Tuntuimad väärtpaberi liigid on aktsia ning võlakiri.

Aktsia on omandiõigust tõendav väärtpaber, mis väljendab teatud osa ettevõttest. Aktsiad ei ole fikseeritud tuluga, kuid aktsia omanikul ehk aktsionäril on õigus saada dividende ehk osa ettevõtte kasumist. Aktsia on võrreldes võlakirjadega palju riskantsem, kuna aktsia väärtus ning dividendide suurus sõltub firma majandlikust edukusest. Aktsiad jaotuvad lihtaktsiaks ja eelisaktsiaks, kus eelisaktsias on lihtaktsia omanikega võrreldes erinevad õigused. Eelisaktsia omanik saab kasumi jaotumise või ettevõtte pankrotistumise puhul väljamaksed enne tavalisi aktsionäre. Eelisaktsia omanikul on tavaliselt fikseeritud dividendid, mistõttu on neil väiksem teenimisvõimalus, kuid ka väiksem risk [1].

Võlakiri on võlakohustust tõendav fikseeritud tuluga finantsinstrument, mis kohustab laenuvõtjal võlg kokkulepitud tähtajal koos intressidega tagasi maksta. Võlakirjad on madala riskiga ning garanteerivad kindla fikseeritud tulu, kuid võlakiri ei anna erinevalt aktsiale õigust otsustada emitendi ehk väärtpaberi väljalaskja tegevuse üle. Võlakirjade peamised liigid on valitsuse võlakirjad ja ettevõtte võlakirjad, kus ettevõtte võlakirjad on üldjuhul kõrgema intressiga, kuna nendega kaasnev risk on suurem. Optsioon on ostu- või müügiõigust tõendav väärtpaber, mis annab omanikule õiguse osta või müüa kindlaks määratud ajal kindel kogus finantsvara kindlaks määratud hinnaga [1].

Väärtpaberitesse investeerimine on kasulik, kuna neid on kerge osta ja müüa ning üldjuhul suudab investor vältida inflatsiooni mõju ning teenida ka täiendavat kasumit. Küll aga kaasnevad väärtpaberitesse investeerimise suured riskid ning õige investeeringu tegemine nõuab põhjalikku uurimistööd [1].

2.2 Aktsiaorder ja selle liigid

Aktsiaorder on korraldus osta või müüa mingis koguses kindlaks määratud aktsiat. Aktsiaorderite tüüpi on mitmeid erinevaid ning nende valik sõltub investeerija nõudmistest ja vajadustest.

Market order ehk turuhinnaga order on käsk teostada tehing parima turul oleva hinnaga. Üldiselt teostatakse turuhinnaga orderid koheselt. Turuhinnaga orderi puhul pole määratud ja garanteeritud hind, millega aktsiad soetatakse. Investorid peavad selle orderiga arvestama, et tehingu sooritamise hind turule saates ei pruugi tulla sama kui viimati teostatud tehingu hind. Kuna orderiga pole ette antud hinda, võib ühes tehingus olevate aktsiate hind olla erinev [2].

Limit order ehk limiitorder on order, millega on kaasa antud piirhind ning mille eesmärk on teostada tehing piirhinnaga võrdse või sellest parema hinnaga. Ostukorraldust saab teostada ainult piirhinnaga võrdse või sellest väiksema hinanga ning müügi korraldust saab teostada vaid piirhinnaga võrdse või sellest suurema hinnaga. Limiitorderi puhul pole garanteeritud, et tehing teostud, kuna aktsia turuhind ei pruugi jõuda tehingu piirhinnani. Küll aga on limiitorderiga investoril suurem kontroll selle üle, mis hinnaga ta tehingu sooritab [2].

Stop order ehk stopporder on order, mis teostatakse, kui aktsia hind jõuab stopp-hinnani, mis on eelnevalt määratud. Peale stopp-hinnani jõudmist sooritatakse korraldus turuhinnaga orderina. Seda tüüpi orderit kasutatakse siis, kui positsioon on soetatud ning soovetakse piirata kahjumit. Stopporderi alaliik on stopp-limiitorder, mille puhul stopp-hinnani jõudmisel sooritatakse tehing eelnevalt ette määratud piirhinnaga limiitorderina [2].

Hidden order ehk peidetud order on order, mille abil on võimalik maskeerida suurt tehingukorraldust turu eest. See orderi liik on lihtne võimalus, kuidas jääda turul anonüümseks, mis suure korralduse esitamisel ei pruugi võimalik olla [3].

Volatility order ehk volatiilsusorder on order, kus hinna asemel sisestakse orderi parameetriks volatiilsus. Volatiilsusorderit kasutatakse, kui aktsia volatiilsuse muutumise ajaloo põhjal saab väita, et mingil teatud kuul võib aktsia olla volatiilsem. Sellisel juhul on aktsia volatiilsuse kasvu juures võimalik kauplemisest loobuda [3].

Fill or kill ehk täida või tapa order on order, kus antakse soov kaubelda vaid siis, kui order teostatakse silmapilkselt. Kaupleja võib näha head võimalust osta või müüa aktsiat, kuid see võimalus võib kaduda silmapilkselt. Sellises olukorras on *fill or kill* orderi kasutamine eelistatuim [3].

2.3 Algoritmiline kauplemine

2.3.1 Ülevaade algoritmilisest kauplemisest

Tehnoloogia areng on avaldanud suurt mõju sellele, kuidas kauplemine tänapäeval toimub. Iga kauplemise samm on nüüdseks suurel määrel automatiseeritud, mis vähendab dramaatiliselt vahenduskulusid. Tehnoloogia võimaldab meil vähendada kauplemise riske, suurendada turu likviidsust ning suurendada kauplemise efektiivsust. Algoritmiline kauplemine on tehnoloogiliste uuenduste suurepärase näide. Paljudel turgudel kasutatakse kauplemisel spetsiaalselt välja töötatud algoritme, mis teevad inimese eest otsuseid, saadavad teele ostu- või müügi korraldused ning haldavad neid peale tehingut. Algoritmide kasutamine kauplemisturgudel algas 1990date keskel ning 2009. aastaks oli algoritmilise kauplemise osakaal USA aktsiaturgudel kuni 73%. Algoritmid, mida turul kasutatakse võivad olla väga erineva eesmärgiga ning olla kasutusel erinevate turul osalejate poolt. Statistilised arbitraažfondid kasutavad algoritme tohutu hulga turuandmete läbi töötamiseks, et oleks võimalik kiirelt analüüsida hinna kõikumise mustreid. Riskifondid kasutavad algoritme, et suurendada turu likviidsust ning on kasuks teiste turul osalejatega konkureerimiseks. Algoritme võib kasutada ka selleks, et arvutada, kuhu on kõige mõistlikum tehingukorraldus saata [4]. Kauplemise automatiseerimine on viinud olukorrani, kus finantsinformatsiooni andmevahetuse kiirus on ületanud kõik prognoosid ning on suuresti muutunud, kuidas turg toimib, seega on kauplemisalgoritmide pidev uuendamine ning efektiivsuse suurendamine konkurentsipüsimeks ülimalt oluline. Lihtsamat sorti algoritmidega saab arvutada näitaks mahukaaluga keskmine hinna, ajakaaluga keskmine hinna, saabumishinna erinevusi, tehingu mahu ning optimaalseima tehingukorralduse saatmise teekonna. Keerulisemad algoritmid võivad korraga koguda mitme erineva turu andmed, teha nende põhjal ulatuslikud hinnangud, luua optimaalsed tehingustrateegiad ning sooritada tehingud ning seda kõike sekundi murdosa jooksul [5].

Algoritmilist kauplemist kasutavad peamiselt institutsioonilised investorid ja suured maaklerifirmad, et kauplemisega seonduvaid kulusid vähendada. Algoritmiline

kauplemise eeliseks on ka turu likviidsuse suurenemine. Algoritmid on tehingute lihtsustamise ning kiiruse tõttu kasutusel ka börsidel. Tehingute üli kiire sooritamine võib olla ka probleemiks ning 2010 aasta *flash crashi* põhjuseks peetakse algoritmilist kauplemist. Algoritmide kasutamine üldplaanis küll suurendab turu likviidsust, kuid kuna tehingud toimuvad nii kiiresti, võib likviidsus hetkega tõusta, kui ka kardinaalselt langeda [6].

2.3.2 Latentsus

Latentsus ja selle võimalikult väike väärtus on elektrooniliste turgude puhul äärmiselt tähtis. Viimaste kümnendite jooksul on aktsiatehingu töötlemisaeg vähenenud minutite pealt millisekunditeni. Latentsust peetakse turu mõistes madalaks, kui see on alla 10 millisekundi ning ülimadalaks, kui see on alla ühe millisekundi. Võrdluseks võib tuua, et keskmise inimese reaktsiooniaeg on sadades millisekundites, seega on elektroonilistel turgudel tehingu töötlemisaeg äärmiselt madal. Teatud turgudel, näiteks kõrgsagedusega kauplemise puhul on latentsus nii oluline, et millisekundine eelis võib konkureerivale maaklerifirmale teatud turuolukorras tuua 100 miljoni dollari suuruse kasumi. Kõrgsagedusega kauplemise puhul on vajadus latentsus madal hoida on ülimalt tähtis, näiteks, võidakse latentsuse vähendamiseks sageli võimalusel ka arvutitehnika börsiga samasse hoonesse kolida [7].

2.3.3 Kõrgsagedusega kauplemine

Kõrgsagedusega kauplemise puhul tehakse tehinguid turul kuni sadu korde sekundis. Kõrgsagedusega kauplemise puhul panustatakse väga suuri ressursse võimalikult efektiivsete ning kiirete algoritmide loomisesse, hooldusesse ning edasi arendamisesse. Sageli on need algoritmid ka ülimalt keerukad. Kõrgsagedusega kauplejaid arvatakse osalevat enam kui pooltes kõigi USA väärtpaberitehingutes [7]. Lisaks kiiretele tehingutele iseloomustavad kõrgsagedusega kauplemist ka suured käibed ning kõrge suhe saadetud tehingukorralduste ja sooritatud tehingute vahel. Kõrgsagedusega kauplemine sai populaarseks, kui börsid hakkasid ajendama firmasid turu likviidsust suurendama. Peamiseks kõrgsagedusega kauplemise puuduseks peetakse inimese, tema otsuste ning interaktiivsuse eemaldamine kauplemisest. Kauplemiseks ning otsuste tegemiseks kasutatakse peamiselt algoritme ning matemaatiliste mudelite asemel tehakse otsuseid tehakse väga kiirelt, mis võib turu teatud olukorras ka tasakaalust välja viia [8].

2.3.4 Tehingunimekiri ja kauplemissüsteemi etapid

Algoritmilise kauplemise puhul teostavad üldjuhul kauplejad oma tehingukorraldused kasutades ühist tsentraliseeritud tehingunimekirja, mis sorteerib korraldused hinna ja saabumisaja järgi, kasutades lihtjärjekorra põhimõtet. Tehingunimekirja alusel püüab süsteem pidevalt sobitada müügi- ning ostukorraldusi. Ideaalselt on algoritmilise kauplemise eeldusteks eelnevalt mainitud tehingunimekiri, likviidne turg ning finantsilise informatsiooni protokollide olemasolu andmevahetuse sujumiseks [9].

Algoritmilise kauplemissüsteemi protsessi võib jaotada viieks etapiks. Esimeseks etapiks on finants-, majandus ning muude aktsia hinnale mõju avaldavate andmete hankimine ning sorteerimine. Teiseks etapiks on tehingueelne andmete analüüs, mille käigus analüüsitakse aktsia omadusi, et tuvastada erinevaid võimalusi, kuidas tehinguid enda osapoolle võimalikult soodsalt sooritada. Kolmandaks etapiks on kauplemisajendi tekitamine ehk analüüsijärgne teadvustamine, millega kaubelda ning kui suures mahus. Neljandaks etapiks on tehingu sooritamine ning viiendaks etapiks on tehingujärgne analüüs, kus analüüsitakse tehingu tulemusi. Tehingujärgse analüüsi käigus analüüsitakse näiteks, kui suur on ostuhinna erinevus tehingukorralduse esitamise ning ostu sooritamise vahel. Algoritmilise kauplemise protsessi puhul on automatiseerimine pühendatud pigem esimesele kahele etapile ehk andmete hankimisele ja analüüsile ning viimased etapid sooritatakse inimeste järelvalve all [9].

2.3.5 Andmed, nende puhtus ning analüüs

Andmeid, mida üks kauplemisalgoritm oma arvutuste ning hinnangu tegemiseks kasutab on palju erinevaid. Finantsandmed on finantsinstrumentide hinda iseloomustavad andmed börsilt või elektrooniliselt kommunikatsioonikanalilt. Majandusandmed on fundamentaalsed majandusandmed, näiteks mingi riigi töötuseäär, sisemajanduslik kodutoodang või riiklikud poliitikad. Kauplemises kasutatavaks andmeteks on ka uudistest pärinevad andmed ning sotsiaalmeediast pärinevad andmed. Kasutatavad andmed jaotatakse kaheks: ajaloolised andmed, mis on varem kogutud või arhiveeritud; või reaajas andmed, näiteks uudiste otseülekanded või värked sotsiaalmeedia postitused. Andmed on kas töötlemata, ehk otse andmeallikast ning vigadest puhastamata; puhastatud, ehk vigadest või vastuoludest puhastatud või analüüsitud andmed. Analüüsitud andmed on juba täpse eesmärgiga kontekstipõhised andmed, mis on fundamentaalsed kauplemisalgoritmide disainimisele, testimisele ning

edasiarendamisele. Analüüsitud andmed juhivad kõiki algoritmilise kauplemise süsteemi komponente ning seetõttu on nende puhtus esmatähtis kogu süsteemi sujuvale tööle, küll aga on puhaste andmete hankimine kallis ning andmete puhastamine väga ajamahukas. Andmete puhtus seisneb eelkõige andmete kasuliku info taustamüra osakaalust, vigadest, vastuoludest ning tervikust puuduoleva informatsiooni hulgast [9].

Tehingueelne andmete analüüs on algoritmilise kauplemise protsessi üks olulisemaid etappe. Tehingueelse analüüsi käigus analüüsitakse reaalaja ning ajalooliseid andmed, et tuvastada parimaid võimalusi tehinguks. Tehingueelse analüüsi üks tehnikatest on fundamentaalne analüüs, mille käigus hinnatakse muutujaid, mis võivad muuta väärtpaberi hinda. Need muutujad võivad olla nii makroökonomilised faktorid, näiteks üleüldine majanduskulg, kui ka firmapõhised faktorid, näiteks firma finantsraportid või juhatuse koosseis. Teiseks tehnikaks on tehniline analüüs, mis on peamiselt seotud väärtpaberi ajalooliste trendide analüüsiga ning mingisuguste mustrite avastamise ning interpreteerimisega. Kolmandaks tehnikaks on kvantitatiivne analüüs, mille käigus andmete peal kasutatakse erinevaid valemeid, arvutuslikke mōõdikuid või masinõpet, et ennustada ja ekspluateerida kauplemiskäiku [9].

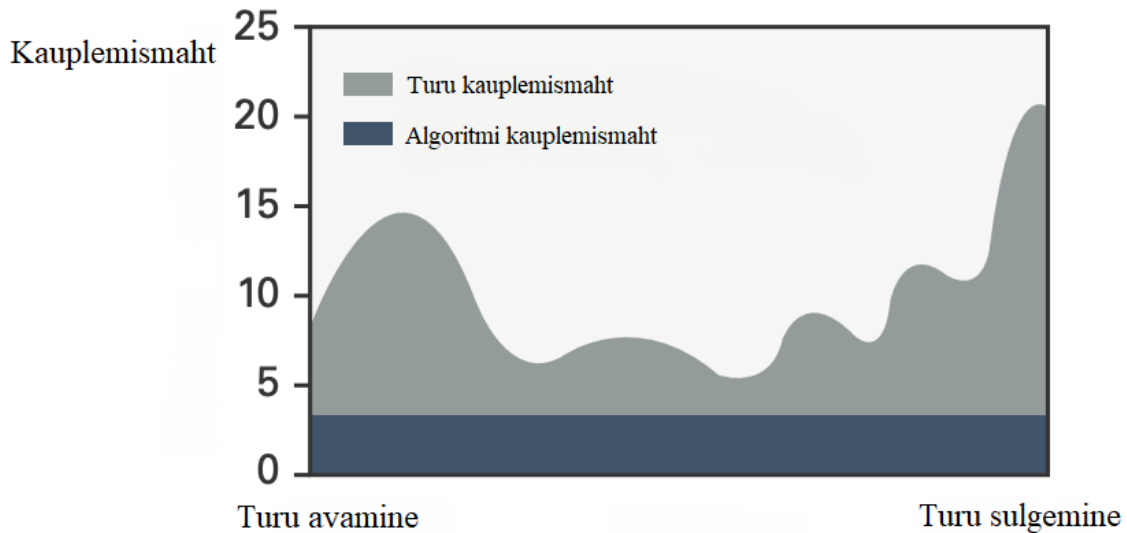
2.3.6 Korralduse täitmise algoritmid

Korralduse täitmise algoritmide kasutamine on üks paljudest moodustest, kuidas muuta kauplemist kasumlikumaks. Seda liiki algoritme on kasulik kasutada, kui on soov kaubelda suurte aktsiakogustega, kuna nad pakuvad osalist kaitset olukorras, kus tehingu sooritamise hind on mõjutatud vähesest likviidsusest. Samuti aitavad need algoritmid ära hoida suure korralduse tõttu tulenevat turutasakaalu muutust, mis tõttu realiseeritakse väärtpaberitehing ebasoodsa hinnaga. Korralduse täitmise algoritmide põhiidee on üritada korraldust realiseerida võimalikult hea hinnaga ning efektiivselt, mida on võimalik saavutada, kui tükeldada suur korraldus väiksemateks osadeks [10].

2.3.7 Ajas hajutamise algoritmid

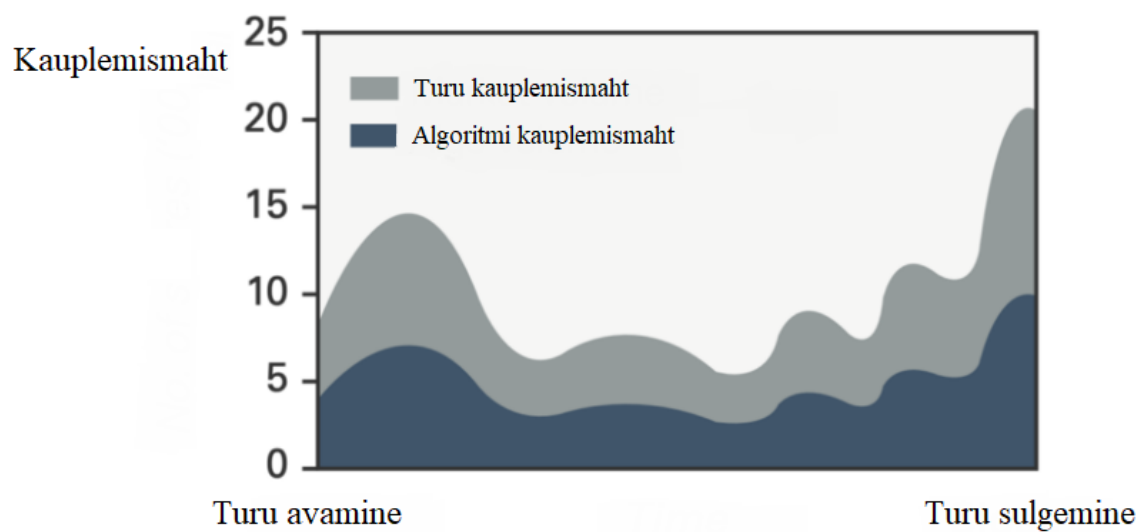
Üks lihtsamaid ajas hajutamise algoritme on *Time Weighted Average Price* ehk TWAP algoritm. See algoritm jaotab tehingukorralduse samasuurusteks tükkideks, arvestades aja kaaluga keskmist hinda. Peale tükeldamist saadetakse korraldus tükkidena konstantse aja tagant turule. Sageli saadetakse iga korralduse tükk turule iga 5 minuti tagant. TWAP algoritmi peamised puudused on see, et see algoritm ei arvesta turul oleva

kauplemismahu muutust päeva jooksul, mis võib viia olukorrani, kus turg on üle ujutatud või hoopis olukorrani, kus kauplemine toimub suuremahuliselt, kuid korralduste hulk on liiga väike, et sellest kasu võtta. Selle probleemi lahendab VWAP algoritm.



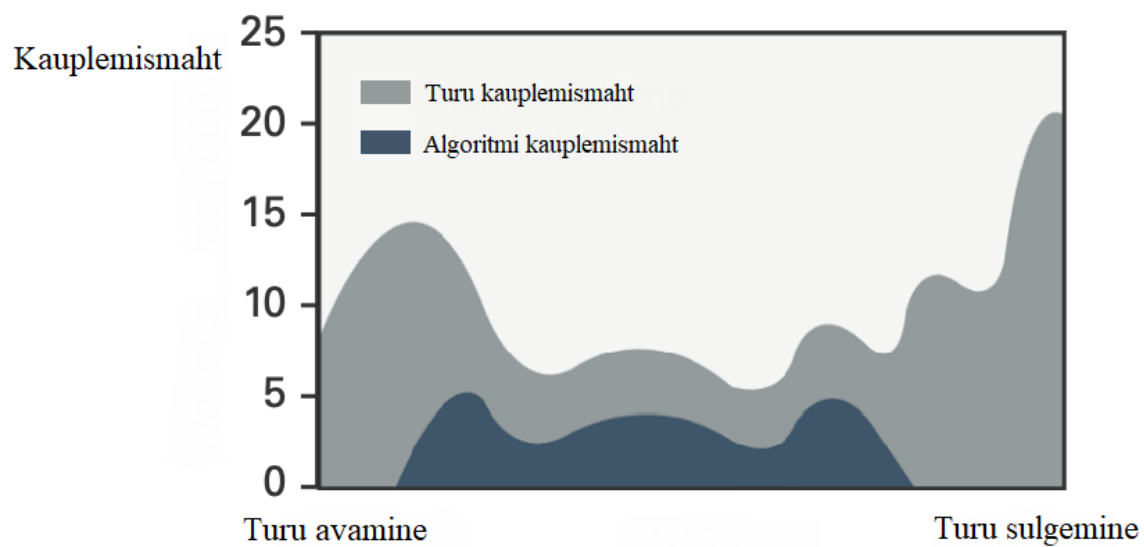
Joonis 1. TWAP kauplemisalgoritmi kauplemismaht päeva jooksul [11].

Volume Weighted Average Price ehk VWAP algoritm jaotab korralduse tükkideks, arvestades väärtpaberi kauplemismahtu andmete põhjal, proovides ennustada järgmise ajahetke kauplemismahtu. Tükkideks jaotatud korraldus saadetakse päeva jooksul turule, kuid ühes tükis olevate aktsiate arv on võrdelises seoses kauplemismahuga. Seega, see algoritm kaupleb rohkem siis, kui turu kauplemismaht on suur ning vähem siis, kui kauplemismaht on väiksem.



Joonis 2. VWAP kauplemisalgoritmi kauplemismaht päeva jooksul [11].

Percent of Volume ehk POV algoritm on sarnane VWAP algoritmile, kuid selle asemel, et teha otsus ajalooliste andmete põhjal teeb see algoritm otsuse päeva hetkelise kauplemissahu põhjal. See algoritm arvutab tükide suuruse arvestades turul osalemise protsenti [10].



Joonis 3. POV kauplemissalgoritmi kauplemissaht päeva jooksul [11].

3 Tehniline lahendus

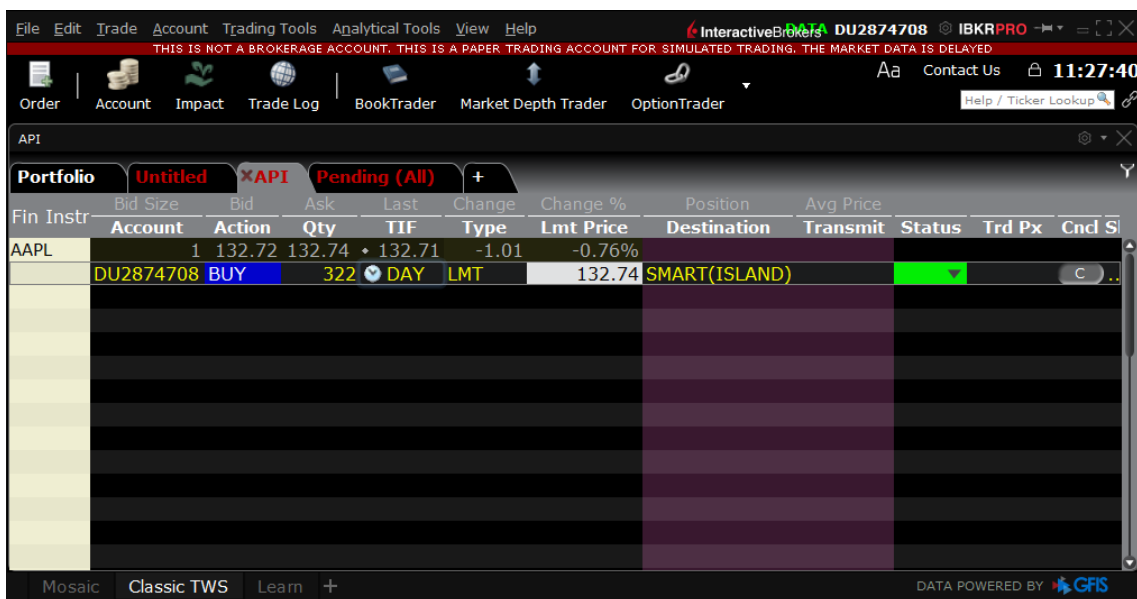
3.1 Ülevaade

Lõputöö praktiline osa põhineb ajas hajutamise algoritmi loomisel, aktsia ostukorralduste saatmisel börsisüsteemi, realiseerides loodud algoritmi ning sellejärgsete tulemuste analüüsil. Kasutatavaks börsisüsteemiks on Interactive Brokersi börsisüsteem/programm TWS ehk Trading Workstation. TWS'iga ühenduse saamiseks on kasutusel IB pakutud API. API abil luuakse börsisüsteemiga ühendus, saadetakse sinna korraldused ning saadakse sealt tagasi andmed korralduse täitmise kohta, millele järgneb andmete töötlemine ja tulemuste analüüs.

3.2 Kasutatud tehnoloogiad

3.2.1 TWS

Interactive Brokersi Trading Workstation on globaalne kauplemissüsteem, mille abil saavad kliendid esitada ning monitoorida tehingukorraldusi; vaadata ja analüüsida enda portfelli; näha suuremate aktsiate kohta *live* uudiseid ning palju muud. TWS'iga on *Java Runtime Environmenti* abil IB API olemasolul võimalik luua ühendus turuga [12]. Lõputöö raames oli TWS'i roll tehingute staatuse jälgimisel. TWS'i abil sai kinnitada, kas tehing jõudis börsini, millised on hetkeseisul aktsia küsimis- ning pakkumishinnad, milline on tehingu status ning mitu aktiivset tehingut hetkel börsil veel olemas on.



Joonis 4. TWS vaade aktiivse ostukorraldusega.

Kuvatõmmisel TWS programmist on näha aktiivset programmi poolt börsile saadetud korraldust. Tähtsamad väljad, mida TWS kuvab ja mida töö käigus kasutati on:

Parameetri nimi	Tähendus
Financial instrument	Aksia nimi
Action	Korralduse liik, kas ostukorraldus või müügi korraldus
Bid	Aksia pakkumishind
Ask	Aksia küsimishind
Quantity	Korralduses olevate aktsiate arv
Type	Tehingu tüüp, peamiselt kas <i>market order</i> (MKT) ehk turuhinnaga order või <i>limit order</i> (LMT) ehk limiitorder
Limit price	Näitab tehingukorralduse piirhinda, kui tegemist on limiitorderiga
Destination	Näitab, millisele börsile on tehing suunatud
Transmit	Nupp, mille vajutamine võimaldab tehingut manuaalselt sooritada
Status	Tehingu staatus
Cancel	Nupp, mille vajutamisel tehing tühistatakse
Trade price	Tehingu hind

Tabel 1. TWSis kuvatavad andmed.

3.2.2 IB API

IB pakub lisaks kauplemissüsteemile ka API. API abil on võimalik luua unikaalsed kauplemissüsteeme ning integreerida kauplemist juba olemasolevatesse

infosüsteemidesse ning *front-end* lahendustesse. API abil on võimalik automatiseerida kauplemissstrateegiaid, küsida turgu iseloomustavaid andmeid ning monitoorida kauplemissportfelli. Samuti on API abil võimalik ka kauplemissstrateegia esitada läbi algoritmi kasutuse, mille kasutust näeb ka antud lõputöös. API pakub ligi 100 olemasolevat klassi, mida saab kasutada TWS'iga suhtlemiseks. API on saadaval kasutamiseks neljas programmeerimiskeeles: Java, C++, Python ning C# [13]. Antud töös on kasutusel API programmeerimiskeeles Java. Java sai valitud, kuna autoril on kõige suurem kokkupuude just Java keelega ning tunneb ennast selle keelega töötamisel kõige mugavamalt.

3.3 Programmi ülesehitus

Programmis on kasutusel 5 klassi, millest üks implementeerib API *interface*'i. Kahele klassile, FileHandler ning Algorithm on loodud ka testid. Testide eesmärk on avastada peamiselt sisendparameetrite anomaaliaga seotud vigu. Järgnevalt annan ülevaate klasside eesmärkidest.

3.3.1 Stock

Stock klassi ülesandeks on luua aktsia objekt. Samuti luuakse *Stock* klassi abil finantsinstrumendi objekt *Contract*, mille olemasolu on oluline osa tehingukorralduse saatmisel börsile. Finantsinstrumendi loomisel määratakse neli tähtsat atribuuti:

- SecType – *Security Type*, määrab väärtpaberi tüübi. Antud töös „STK“ ehk *Stock*.
- Currency – valuuta, antud töös on valitud valuutaks „USD“
- Exchange – valib börsi, kuhu tehingukorraldus saadetakse. Antud töös on valitud börsiks „SMART“. Smart exchange kasutab *Smart Routing* tehnoloogiat, mis pidevalt otsib konkureerivaid turgusid, et leida parim turg, kuhu korraldus saata. Lõplik otsus tehakse hinna põhjal [14].
- PrimaryExch – *Primary Exchange* ehk primaarne börs. Määrab börsi, kuhu korraldus saata, kui Smart Routing tehnoloogiaga on mitu börsi samavääriliste pakkumistega. Antud töös on primaarseks börsiks „NASDAQ“.

3.3.2 Algorithm

Algorithm klass kujutab endast ajas hajutamise algoritmi realiseerivat klassi. Klassi konstruktori loomisel tehakse valemite järgi arvutused, mille tulemusena tekib nimekiri aktsia tükkidest, mis tuleb ajas hajutades börsile saata. Samuti arvutatakse klassis aeg, mille jooksul tuleb tehingukorraldus teele saata. Kogu korralduse saatmise aeg jagatakse läbi aktsia tükkidega ning saadakse aeg, mis tuleb oodata iga korralduse saatmise vahel. Klassi sisendiks on aktsia nimi, hind, aktsiate kogus ning tehingukorralduse liik, s.t kas osta või müüa.

3.3.3 FileHandler

FileHandler klass kujutab endast faili kirjutajat. Fail luuakse .csv formaadis, mis on tekstifail, kus iga rida on üks kirje ning igal real on andmed eraldatud komaga. Failitüüp valiti eelistatuna nt .txt formaadi ees, kuna antud formaadiga failist on lihtne luua .xlsx ehk Exceli formaadis fail, mille abil luuakse korralduste ajaloo kohta statistikat ning mille põhjal tehakse töö katsete analüüs. Faili loomisel on sisendiks aktsia nimi, tehingukorralduse liik ning kas kasutati algoritmi või mitte. Antud sisendparameetrite põhjal koostatakse ka faili nimi. Näiteks, kui tahetakse kirjutada faili korralduse kohta, kus aktsia nimi on MSFT, tehingukorraldus on BUY ning ei kasutatud algoritmi, siis faili nimeks saab MSFT_BUY_NO_ALGO.csv. Antud töö käigus oli FileHandler klass kasutusel algoritmi iga tsükli töös, et salvestada turult saadud informatsioon tehingukorralduse täituvuse, tehingu hinna ning staatuse kohta.

3.3.4 ConnectionHandler

ConnectionHandler on klass, mis implementeerib APIs olevat interface'i nimega Ewrapper. Antud klass vastutab kogu ühenduse eest TWS'iga. Klassi ülesanded on:

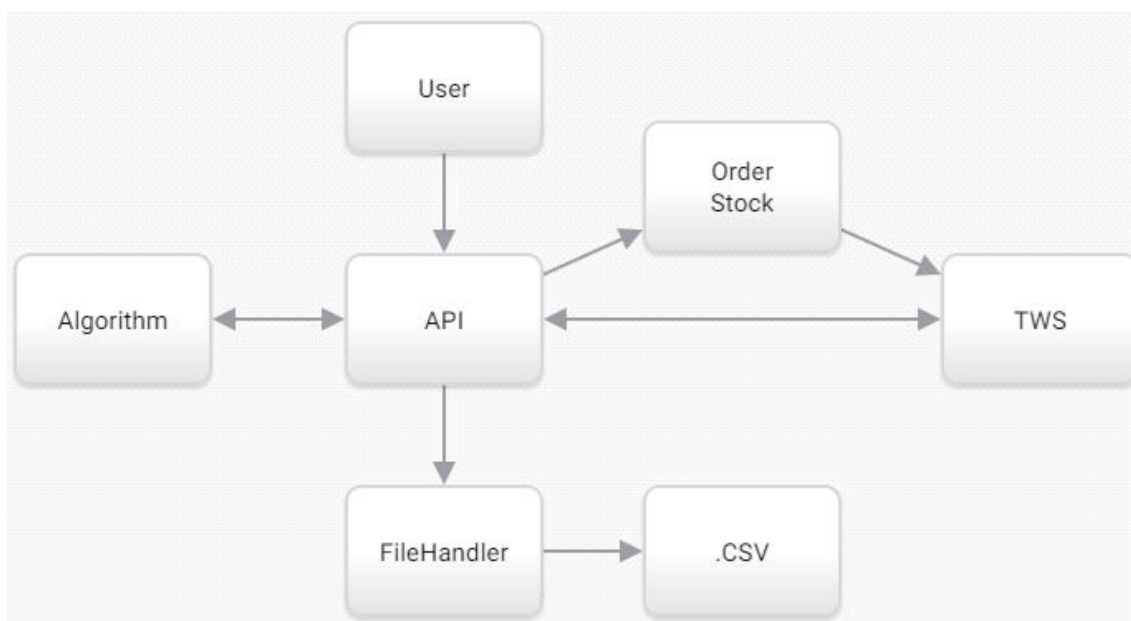
- Luua ning katkestada ühendus TWS'iga.
- Saada TWS'ist aktsia turuandmeid.
- Luua Stock ning Order objektid ning saata need börsile.
- Saada pidevalt andmeid tehingukorralduste täituvuse, hinna ja muude korraldust iseloomustavate andmete kohta.

3.3.5 Main

Main klass vastutab programmi töötamise eest. Main klassis määratakse ka aktsia nimi, aktsiate koguarv ning aktsiatükkide arv. Peale koodi käivitamist tellitakse turult andmed aktsia väärtuste kohta ning kui turg on suletud, programmi töö lõppeb. Kui turg ei ole suletud, saadetakse Algorithm klassi ning sealt saadakse aktsiatükkidega nimekiri. Seejärel saadetakse tükeldatud korraldused turule. Main klassis töötab *for-loop* senikaua, kuni kõik tehingukorraldused on turule saadetud.

3.4 Programmi töötamise skeem

Töös loodud programm töötab järgmiselt:



Joonis 5. Programmi töötamise skeem.

1. Kasutaja sisestab soovitud aktsia nime ning hulga.
2. Programm käivitatakse.
3. API loob ühenduse TWS'iga ning küsib sealt aktsia küsimis- ja pakkumishinda ning salvestab need muutujatesse. Kui turg on suletud, programmi töö lõppeb.
4. Algorithm klass loob fikseeritud andmete ning äsja saadud turuandmete põhjal nimekirja aktsiatükkidega ning arvutab aja, mille jooksul tuleb korraldused teele saata.

5. API loob Stock ja Order objekti ning saadab objektid börsiturule.
6. Tehingukorralduse täitumisel saadab TWS APIsse andmed korralduse täituvuse kohta.
7. API salvestab FileHandler klassi abil saadud andmed korralduse kohta .CSV faili.
8. Kõikide korralduste täitumisel ühendus TWSiga katkestatakse ning programmi töö lõppeb.

3.5 Ajas hajutamise algoritm

Lõputöös kasutatakse tehingukorralduste ajas hajutamiseks kasutatakse algoritmi, mis põhineb Almgren ja Chrissi kirjeldatud valemil [15]. Valem kasutab hüperboolfunktsioone erinevate kogutud andmetega, mille väljundiks on ajaperioodil t ning tervikliku korralduse n osaks jaotamise korral loend $X = [x_0, x_n]$, kus x on teatud ajahetkel olemasolevate aktsiate arv. Kui $n = 1000$, siis ajahetkel t_1 võib x_1 väärtus olla näiteks 800.

$$x_j = \frac{\sinh(\kappa(T - t_j))}{\sinh(\kappa T)} X, \quad j = 0, \dots, N,$$

T – tehingu sooritamise kogu aeg

T_j – ajahetkeks j möödunud aeg

X – aktsiate koguarv

κ – aja pöördväärtus

Peamised valemis otseselt kasutatavad parameetrid on korralduses olevate aktsiate koguhulk, kogu korralduse esitamise aeg ning tehingu poolestusaeg ehk aja pöördväärtus. Lisaks eelnevalt nimetatud parameetritele on algoritmi töös kasutusel veel palju erinevaid fikseeritud ja fikseerimata parameetreid.

Lisaks tehingu tükideks jagamise algoritmile on programmis arvutatud ka kogu korralduse teele saatmise aeg T . Korralduse teele saatmise aeg jagatakse omakorda läbi

aktsia tükide arvuga n , mille tulemusena saadakse aeg t ehk aeg, mille järel iga tükk saadetakse turule [16].

Ajale lisatakse juhuslikkuse faktor, mille tulemusena kõigub aeg 10% võrra. Juhuslikkuse eesmärgiks on mitte jätta muljet automatiseeritud süsteemist, kus iga korraldus tuleb sisse sama ajavahemiku tagant. Kui juhuslikkuse faktor välja jätta, võib süsteem eelnevate korralduste põhjal ennustada, millal uus korraldus sisse tuleb.

$$T = \sqrt{\frac{2X}{A}} \qquad t = \frac{T}{n}$$

X – aktsiate kogu arv

A – riski maksumuse parameeter

N – aktsia tükide arv

3.6 Algoritmi tööprotsessi käigus kasutatavad parameetrid

Algoritmis on kasutusel kahte erinevat tüüpi parameetreid, mille põhjal tehakse otsus. Esimeseks tüübiks on eelnevalt ette määratud ehk fikseeritud parameetrid, mis ei muutu algoritmi töö ning valideerimise jooksul. Fikseeritud parameetri näideteks on riskiparameeter ning kauplemispäevade arv aastas. Teiseks tüübiks on fikseerimata parameetrid, mis pidevalt muutuvad algoritmi rakendamise käigus. Fikseerimata parameetrite näideteks on tehingukorralduses olevate aktsiate koguarv ning aktsia piirhind. Samuti on parameetreid kahte tüüpi: arvulised parameetrid ning arvutatavad parameetrid, mis tuleb eelnevate arvuliste parameetrite põhjal välja arvutada.

Ühe parameetri väärtus on võetud Almgren ja Chrissi teadustööst, kuna selle väärtus on kõige optimaalsem antud algoritmi jaoks. Ülejäänud parameetrite väärtused määras autor kas ise manuaalselt või saadi parameetrite väärtused turult peale turuga ühenduse loomist.

3.6.1 Kasutatavad arvulised parameetrid

Järgnevalt loetlen algoritmis kasutatavaid arvulisi parameetreid, nende väärtusi ning tähendusi. Mõnede parameetrite väärtused olid määratud enne programmi käivitamist, kuna nendele parameetritele polnud võimalik saada väärtuseid aktsiaturult.

Parameetri nimi	Tähendus
Riskiparameeter	Iseloomustab, kui suurt riski on kaupleja tehingu tegemiseks nõus kasutama. Mida suurem on parameetri väärtus, seda vähem riskialdim on tehingu korraldus. Antud lõputöö algoritmis on parameetri väärtuseks $2 * 10^6$, mis põhineb Almgren ja Chrissi teadustööst. Väärtus iseloomustab riskikartlikuse poole kalduvat kauplemisstrateegiat.
Kauplemispäevade arv aastas	Kauplemispäevade arv aastas on kõik päevad aastas, kus kauplemine on võimalik. Selle aasta kauplemispäevade arv ehk lõputöös kasutatav parameetri väärtus on 253.
Aktsiate koguarv	Tehingukorralduses olevate kogu aktsiate arv on kasutaja poolt määratud enne koodi käivitamist.
Aksia pakkumishind	Aksia pakkumishind on maksimaalne hind, mida investor on nõus aksia eest välja käima. Aktsia pakkumishinna keskmine väärtus on igas kauplemissüsteemis olemas ning see väärtus saadakse API abil TWS'ist, mis iseloomustab hetkel turuv olevat keskmist väärtust.
Aksia küsimishind	Aksia küsimishind on minimaalne hind, mida aksia müüja on nõus aksia eest vastu võtma. Aktsia küsimishinna keskmine väärtus on igas kauplemissüsteemis olemas ning see väärtus saadakse API abil TWS'ist, mis iseloomustab hetkel turuv olevat keskmist väärtust.
Pakkumise-nõudmise hinnavahe	Parameeter, mis iseloomustab, kui palju erinevad aktsia pakkumishind ning küsimishind. Parameetri väärtus saadakse aktsia küsimishinna ning pakkumishinna vahest ning mõlema hinna väärtused saadakse API abil TWS'ist. Keskmine pakkumise-nõudmise hinnavahe väärtus on 0.01 ja 0.5 vahel.
Volatiilsus	Aksia volatiilsus näitab, kui palju kõigub aktsia hind teatud ajaperioodi jooksul. Aktsia volatiilsus määratakse kasutaja poolt enne koodi käivitamist. Aktsia volatiilsus on fikseeritud, kuna demokasutajal puuduvad õigused TWS'i abil reaallaja turu volatiilsuse saamiseks.
Päevane aktsia kauplemismaht	Näitab, kui palju teatud aktsiaga ühe päeva jooksul kaubeldakse. Väärtus on määratud kasutaja poolt enne koodi käivitamist, kuna demokasutajal puuduvad õigused TWS'i abil reaallaja kauplemismahu saamiseks.

Tabel 2. Algoritmi töös kasutatavad eelnevalt fikseeritud arvilised parameetrid.

Kahe viimase parameetri ehk volatiilsuse ja päevase aktsia kauplemismahu väärtused võeti katsetes kasutatud aktsiatest sõltuvalt. Katseteks valiti MSFT ja NVDA aktsiad ning valikute põhjendus on kirjeldatud peatükis 4.1. Katseteks valitud NVDA ja MSFT aktsiate antud väljade arvilised väärtused on:

Parameeter	Parameetri väärtus [17]
NVDA eelmise kuu keskmine kauplemismaht	5 600 000 aktsiat/päevas
NVDA viimase kuue kuu volatiilsus	43,2%
MSFT eelmise kuu keskmine kauplemismaht	28 000 000 aktsiat/päevas
MSFT viimase kuue kuu volatiilsus	30,8%

Tabel 3. Algoritmi töös kasutatavad kauplemismahud ning volatiilsused.

3.6.2 Kasutatavad arvutatavad parameetrid

Järgnevalt loetlen ning selgitan algoritmis kasutatavaid parameetreid, mille väärtused tuli arvutada valemite põhjal.

Parameetri nimi	Tähendus
Aksia tükkide arv	Aksia tükkide arv näitab, mitmeks tükiks terviklik tehingukorraldus tükeldatakse. Tükkide arv saadakse jagades kogu aktsiate arv 250'ga.
Turumõju	Turumõju iseloomustab mõju turu tasakaalule peale programmi poolt esitatud korralduse esitamist. Turumõju arvutamiseks jagatakse tehingukorralduse aktsiate koguarv päevase kauplemismahu ning pakkumise-nõudmise hinnavahe korrutisega.
Riski maksumuse parameeter	Riski maksumuse parameeter iseloomustab riski ja volatiilsuse mõju tehingule. Riski maksumuse parameetri arvutamiseks jagatakse riskiparameetri ja volatiilsuse korrutis kahekordse turumõju korrutisega.
Tehingu edastamise aeg	Tehingu edastamise aeg näitab, mis aja jooksul kogu tehingukorraldus esitatakse. Tehingu edastamise aeg saadakse, kui võtta ruutjuur kahekordse aktsiate koguhulga ja riski maksumuse parameetri jagatisest.
Tehingu poolestusaeg	Tehingu poolestusaeg on tehingu edastamise aja pöördväärtus.
Tehingukorralduste esitamise vaheline aeg	Tehingukorralduste esitamise vaheline aeg on aeg, mis iseloomustab kui pikk on ajavahemik kahe korralduse turule saatmise vahel. See saadakse jagades tehingu edastamise aeg aktsia tükkide arvuga.

Tabel 4. Algoritmi töös kasutatavad arvutatavad parameetrid.

4 Katsed ja analüüs

4.1 Katsete ülesehitus

Katsete tegemiseks esitatakse börsile kahte tüüpi ostukorraldust: üks on algoritmiga tükki jaoks jaotatud ostukorraldus ning teine on ühes tükis esitatud korraldus. Ostukorralduses olevate aktsiate arvuks on mõlemal juhul valitud 5000, mis on piisavalt suur arv, et demonstreerida tükeldamisalgoritmi tööd. Ühes tükis saatmise korral saadetakse korraldus koguarvuga 5000 aktsiat korraga turule. Tükeldamise korral jagatakse aktsia x tükiks, kus x saadakse jagades kogu aktsiate arv 250'ga, mis tõttu tuleb ühe tüki suurus piisavalt suur, kuid samal ajal ei avalda suurt mõju turuolukorrale. Ostukorraldust viiakse katsete käigus läbi mõlema meetodiga 25 korda, et saadud andmete hulk oleks piisavalt suur analüüsimiseks.

Ostukorraldused saadetakse börsile limiitorderiga. Limiitorderi puhul määratakse aktsia piirhind piirangukäsuga enne korralduse edastamist turule. Piirhind saadakse küsides API meetoditega TWS'ist turu viimast aktsiivset küsimis- ja pakkumishinda. Ostukorralduse puhul määratakse piirhinnaks küsimishind ning müügikorralduse puhul pakkumishind. Algoritmiga börsile saadetud ostukorralduse saatmise jooksul küsitakse tehingu sooritamise vältel pidevalt turuandmeid, et muuta reaalajas korralduse pakkumis- ja küsimishinda.

Peale aktsia nime ja ostukorralduse tüübi valimist läbib korraldus algoritmi, mis andmeid analüüsides tükeldab selle osadeks. Samuti arvutatakse algoritmi abil ka aeg, mille tagant tuleb saata tehingukorraldused teele. Seejärel saadetakse aktsia arvutatud tükkidena arvutatud ajavahemiku järel börsile teele.

Tehingu sooritamise korral saadakse TWS'ist tagasi informatsioon aktsia müügihinna kohta mida seejärel võrreldakse aktsia turule saatmise hinnaga, seega leitakse aktsia ostukorralduse saatmishinna ja realiseeritud tehingu hinnavahe ehk kasum. Need andmed

kirjutatakse FileHandler klassi abil CSV faili, mis võimaldab turult saadud andmeid edasi töödelda.

Katseid tehakse kahe erineva aktsiaga ning realiseeritakse ostukorraldust. Valiti kaks aktsiat, millel on mõõdukalt suur kauplemissaht, et vältida olukorda, kus katsete tegemise käigus on aktsiate nõudlus liiga väike ning tehingukorraldust pole võimalik sooritada.

Esimeseks aktsiaks valiti Microsoft'i aktsia (MSFT), mis iseloomustab suure kauplemissahuga ning kõrge turuhinnaga aktsiat. MSFT'i keskmine päevane kauplemissaht on 28 miljonit ning aktsia hind katsete käigus oli ligikaudu 220\$. Teiseks aktsiaks valiti Nvidia aktsia (NVDA), mis iseloomustab võrreldes MSFT'iga väiksema kauplemissahuga, kuid kõrgema hinnaga aktsiat. NVDA keskmine päevane kauplemissaht on 5,4 miljonit aktsiat ning aktsia hind katsete käigus oli ligikaudu 550\$ [17].

4.2 Katsete tulemused

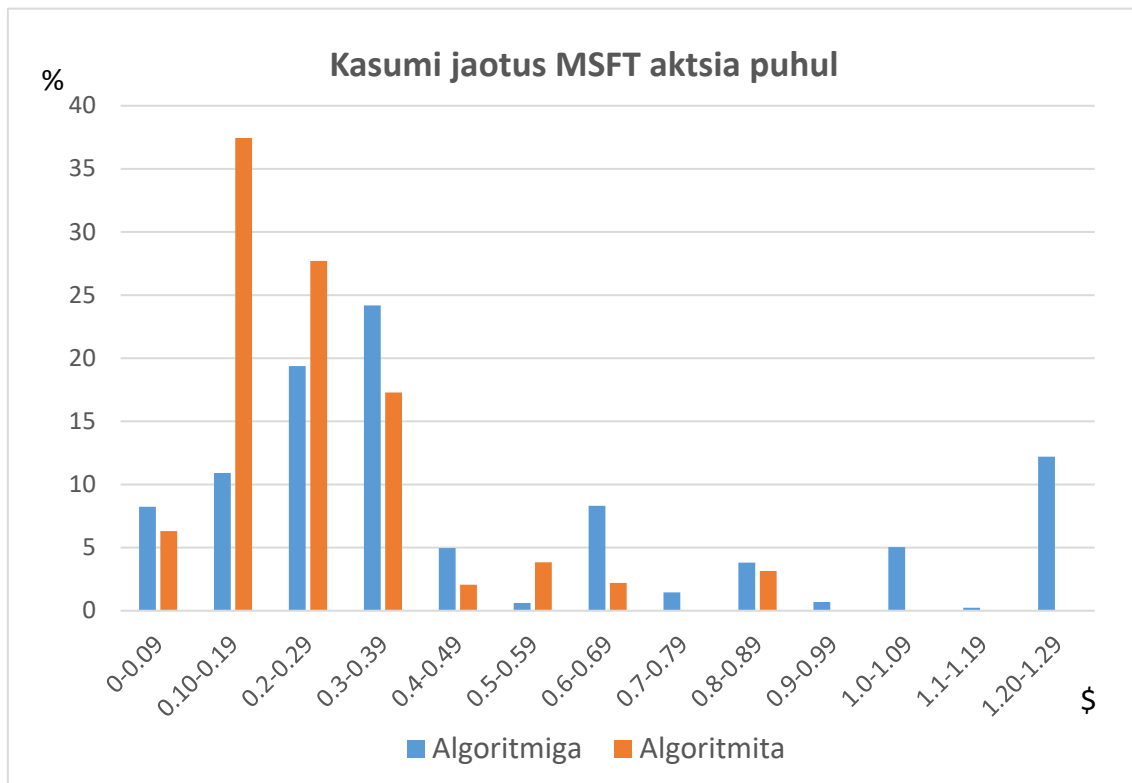
4.2.1 Microsofti aktsiate ostu tulemused

Microsofti aktsia tulemused näitavad, et algoritmi kasutades saab aktsiad parema hinnaga kätte. Algoritmiga ostukorralduse puhul on suuremad nii maksimaalne kui ka keskmine kasum. Minimaalne kasum on mõlemal puhul 0 ehk aktsiad saadi kätte piirhinnaga. Suur erinevus kahe meetodi vahel on ka dispersioonil ja standardhälvel. Algoritmiga ostukorraldust saates on dispersioon kuus korda suurem, mis tähendab seda, et kasumi suuruse varieeruvus on tunduvalt suurem. Varieeruvust illustreerib hästi ka minimaalse ja maksimaalse tehingu hinna vahe erinevus kahe meetodi vahel. Samuti on algoritmiga sooritatud ostukorralduse puhul suurem standardhälve, mis on algoritmiga sooritatud korraldusest ligi kaks korda suurem. See tähendab seda, et kasumi erinevad väärtused on algoritmiga ostukorralduse puhul lähemal keskväärtusele kui algoritmiga korralduse puhul.

	Min tehingu hind	Max tehingu hind	Min kasum	Max kasum	Keskmine kasum	Dispersioon	Standardhälve
Algoritmiga	221,68	222,88	0,00	1,20	0,410	0,150	0,387
Algoritmita	220,21	222,41	0,00	0,73	0,170	0,025	0,160

Tabel 5. Microsofti aktsiate ostu tulemused.

Keskmise kasumi väärtus 0,41\$ tähendab seda, et keskmiselt säästab algoritmi kasutades aktsiate ostul $0,41 / 221,68 * 100\% = 0,19\%$ aktsia turuväärtusest. Ostes 5000 aktsiat, säästab algoritmi kasutades kogu korralduse peale keskmiselt $0,410 * 221,68 = 2050\$,$ kuid katsete põhjal võib maksimaalse kasumi puhul säästa kuni $1,2 * 221,68 = 6000\$.$



Joonis 6. Kasumi jaotus protsentuaalselt MSFT aktsia puhul.

Tulemuste valideerimiseks leiti ka algoritmi tulemusel tekkiva kasumi jaotus protsentides. Jaotusest tuleneb, et algoritmiga on tulemused kontsentreeritud väiksema kasumi poole, kuna ligikaudu pooled tulemused on vahemikus 0-0.2\$. Edasine kontsentratsioon algoritmiga tulemuse puhul langeb ning viimased väärtused jäävad vahemikku 0.81-0.9\$. Algoritmiga tulemuste põhjal kontsentratsioon kasvab jaotuste algusest kuni vahemikuni 0.31-0.4, kus on protsentuaalselt veerand tulemustest. Edasine jaotustes esinev protsent ei ületa 10% kuni viimase, kõige suurema kasumiga jaotuseni,

kus on veidi üle 10% väärtuste koguhulgast. Kõige suurema kasumiga jaotuse üpris kõrge kontsentratsioon on ka keskmise kasumi kõrge väärtuse põhjuseks.

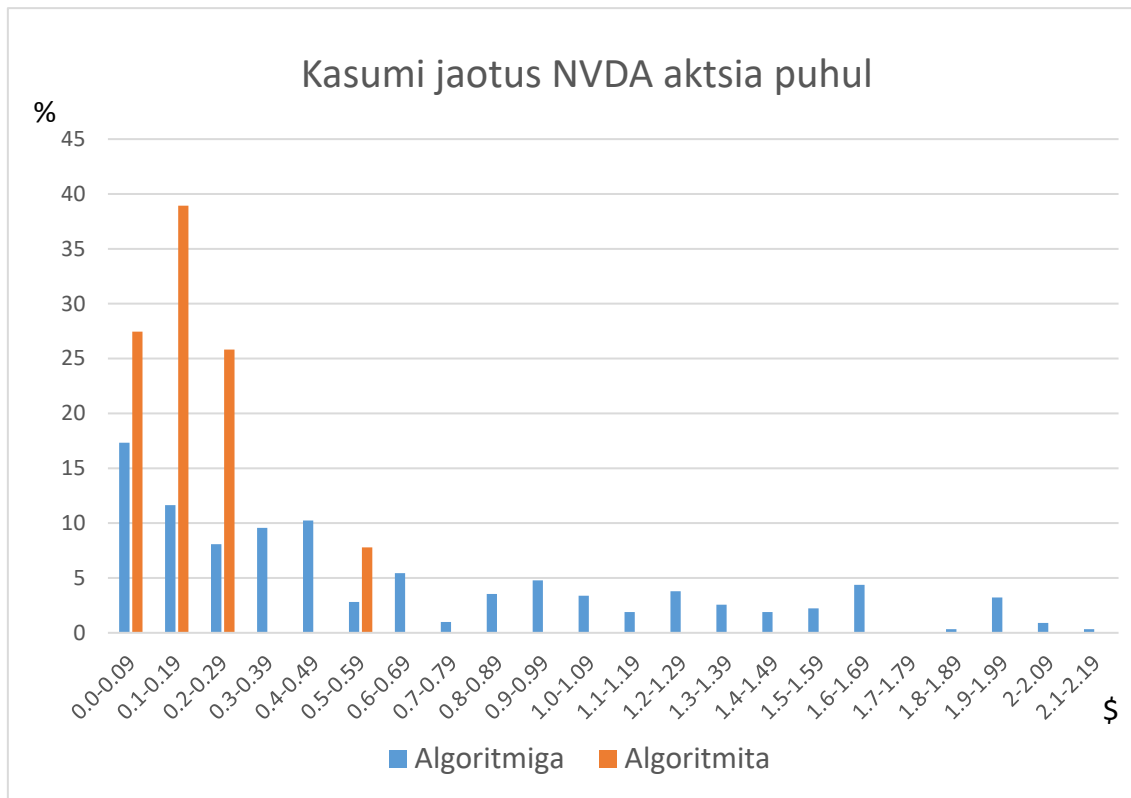
4.2.2 Nvidia aktsiate ostu tulemused

Ka Nvidia aktsia tulemused näitavad, et parema hinnaga saadakse aktsiad kätte kasutades algoritmi. Maksimaalne kasum oli algoritmi kasutades tunduvalt suurem, olles suurem ligikaudu 4 korda võrreldes meetodiga, kus algoritmi ei kasutatud. Ka keskmine kasum on algoritmi kasutades ligikaudu 4 korda suurem. Kuna Nvidia aktsia hind on Microsofti omast ligikaudu 2,5 korda suurem ning ka maksimaalne kasum on ligikaudu 2x suurem viitab sellele, et algoritmi efektiivsus on võrdelises seoses aktsia turuhinnaga, seega on algoritmi efektiivsus suurim kaubeldes aktsiatega, mille turuhind on kõrgem. Nvidia aktsia puhul oli aga minimaalne kasum algoritmiga meetodi puhul väiksem. Järjekordselt on algoritmi kasutades kasumi dispersioon ja standardhälve tunduvalt suuremad võrreldes meetodiga, kus algoritmi ei kasutata. Dispersioon on sel juhul lausa 24 korda suurem, mis tähendab seda, et kasumi varieeruvus on algoritmi kasutades tohutult palju suurem. Samuti on ka standardhälve kordades suurem, mis tähendab jällegi seda, et algoritmiga saadud kasum on keskväärtusest kaugemal võrreldes meetodiga, kus algoritmi ei kasutata.

	Min tehingu hind	Max tehingu hind	Min kasum	Max kasum	Keskmine kasum	Dispersioon	Standard- hälve
Algoritmiga	524,60	526,71	0,00	2,21	0,636	0,339	0,582
Algoritmita	525,30	526,39	0,06	0,51	0,188	0,014	0,120

Tabel 6. Nvidia aktsiate ostu tulemused.

Keskmine kasum Nvidia puhul jääb protsentuaalselt samasse suurusjärku kui Microsofti aktsiatega. Nvidia aktsiaid algoritmi kasutades ostes säästab $0,636 / 524,6 * 100\% = 0,12\%$ aktsia turuväärtusest. Ostes 5000 aktsiat, säästab algoritmi kasutades kogu korralduse pealt keskmiselt $0,636 * 5000 = 3180\text{\$}$, kuid maksimaalset kasumit arvestades võib säästetud rahasumma olla kuni $2,21 * 0,636 = 11050\text{\$}$.



Joonis 7. Kasumi jaotus protsentuaalselt NVDA aktsia puhul.

Võrreldes MSFT'i kasumi jaotusega on NVDA kasumi jaotus palju kaootilisem. Selle peamiseks põhjuseks on NVDA kasumi tulemuste suur dispersioon ehk tulemuste suur varieerumine. Algoritmiga tulemuste kontsentratsioon on pigem keskmistel väärtustel ning vahemik 0.1-0.19\$ moodustab ligikaudu poole tulemuste väärtustest. Algoritmiga tulemuste viimaseks jaotuseks on 0.5-0.59, mis moodustab ligikaudu 10% kogu tulemustest. Algoritmiga tulemuste kontsentratsioon on jaotuste alguses, kuid suurima jaotuse protsentuaalne osakaal on vaid 17%. Algoritmiga tulemuste puhul on jaotuste hulk väga suur ning tulemused on jagunenud jaotuste kahe viimase kolmandiku peale üpris võrdselt. Algoritmiga saadud kasumi maksimaalse väärtuse tõttu on ka keskmine kasum tunduvalt suurem, kui seda saadi algoritmiga meetodiga.

4.3 IB demo konto piirangud ja nende mõju katsete tulemustele

Lõputöös tehtud katsed tehti IB demo ehk *paper account*iga. Antud kasutajaga saab küll kaubelda, kuid tegevus ei toimu päris turul, see tähendab, et ostukorralduse edastades ning aktsiaid ostes ei ole sellel mõju turuhinnale või turutasakaalule. Kuna algoritmi põhimõtteks ongi saata välja tehingukorraldus nii, et sellel poleks väga suurt mõju turuolukorrale, võib demo kontoga saadud kasum väiksemaks jääda võrreldes algoritmiga

saadetud ostukorraldusega. See tähendab seda, et kui saata lõputöö käigus loodud algoritmiga identne tehingukorraldus 5000 aktsiaga päris kasutajaga ja päris börsiturule, võib kasumi erinevus algoritmi kasutades ja kasutamata olla tunduvalt suurem, kui võib see olla demo kasutajaga.

Illustreerimaks eelnevalt selgitatud piirangut võtan näiteks Nasdaq börsi, mis on väärtpaberitehingute mahult teisel kohal maailmas. Nasdaq'i regulaarsed kauplemisajad on argipäevil 9:30'st 16:00'ni, seega on turg avatud 6,5 tundi ehk 390 minutit. Kui oletada, et kõik Nvidia päeva peale ostetud aktsiad ostetakse päeva peale võrdselt ära, tuleb minutis kaubeldud aktsiate hulga ligi 14000 aktsiat. Kui võtta arvesse seda, et lõputöös saadetud ostukorraldus sisaldas 5000 aktsiat, oleks see päris turule saates ligikaudu kolmandik minutisest kauplemismahust. Kui aga saata turule tehingukorraldus, mis on kolmandik minutisest kauplemisest, on garanteeritud, et sellel on suur mõju turuolukorrale ning aktsiad saadakse halvema hinnaga kätte. Sel juhul oleks algoritmi kasutades tulemus palju parem, kui algoritmi kasutamata jättes.

Teiseks piiranguks demo kasutaja juures on fakt, et kõik turgu iseloomustavad näitajad on 15 minutilise hilinemisega ehk 15 minutit reaalist maas [18]. Kuna algoritmi töö käigus kui ka korralduste turule saatmisel oli oluliseks parameetrikts turult saadud küsimis- ja pakkumishind, muudab hilinenud andmete olemusolu algoritmi efektiivsust. Kui turgu iseloomustavad parameetrid saadakse kätte reaalist, suudab algoritm luua efektiivsema ning turu hetkeseisule paremini orienteeritud lahenduse.

Samuti oli demo kasutaja kasutamisega seotud piiranguks ka see, et aktsiate kauplemismaht ning volatiilsus tuli fikseerida enne algoritmi tööd ning mitte algoritmi töö käigus, kuna demo kasutajal puudusid õigused antud andmete saamiseks. Kuna andmed ei olnud reaalist, võis see ka mõjutada algoritmi tööd ning tulemust.

Kokkuvõttes võib oletada, et lõputöös loodud algoritm ning katsete käigus saadud tulemused ei illustreeri algoritmi maksimaalset võimekust. Kuna kasutusel oli demo kasutaja, jäi korralduse esitamisest välja mõju turule, kuid mõju turule on ajas hajutamise algoritmi loomise peamine põhjus. Kui algoritmiga oleks esitatud korraldus päris börsiturule, oleks selle tulemused olnud tõenäoliselt tunduvalt paremad võrreldes sellega, kui esitada suur korraldus algoritmi kasutamata.

4.4 Algoritmi integreerimine

Antud lõputöö käigus loodud algoritmi on võimalik integreerida mitmetesse süsteemidesse ning selle kasutamisest saab kasu ka lihtkasutaja. Algoritm hajutab ajas tehingukorraldust, mille tulemusena on võimalik aktsiad saada kätte parema hinnaga. Seetõttu on programmi kasutamine kasulik nii inimesele, kes alles alustab kauplemist; „hobikauplejale“ või isegi investeerimisfirmale, kui firmal pole juba parasjagu mingisugune algoritm korralduste saatmiseks või selle ajas hajutamiseks loodud.

Turult võib API abil saada mitmesuguseid andmeid- need võivad olla kas tehingueelsed andmed; tehingu jõustumisel saadud andmed ehk reaalse tehingukorraldusega seotud andmed või ka tehingujärgsed andmed, mille abil on võimalik mõõta korralduse mõju turuolukorrale. Mis suurendab antud programmi väärtust näiteks IB poolt välja pakutud algoritmide ees on fakt, et kõiki eelnevalt mainitud andmeid on võimalik loodud programmiga käsitleda, neid meetodites kasutada ning neid töödelda. Samuti on kogu algoritm ja kõiki selle parameetreid võimalik vajadusel muuta, mis suurendab programmi mitmekülgset ning atraktiivsust. Töö käigus loodud algoritmi ning programmi integreerides on võimalik luua igasuguseid suurema keerukusega süsteeme, näiteks luua veebirakendusi või arendada valminud programmist välja *full-stack* süsteem.

Programmis on kasu ka loodud klassist FileHandler, mis kirjutab turult saadud andmed CSV faili. Kuna CSV failist saab väga lihtsalt luua näiteks Exceli programmis tabeleid, on FileHandler klassi olemasolu ning integreeritus programmi tööprotsessi statistilises mõttes väga kasulik ning võimaldab palju efektiivsemalt ning lihtsamalt käidelda börsil käitumisega seonduvad andmeid.

5 Kokkuvõte

Töö eesmärgiks oli koostada väärtpaperitehingute ajas hajutamise algoritm ning luua programm, mis seda rakendaks. Samuti oli eesmärgiks leida katsete tulemustega loodud algoritmi efektiivsus. Loodud programm saatis turule ostukorraldused kasutades algoritmi ning jättes seda kasutamata ning salvestas mõlema meetodi kohta tulemused.

Katsete tulemused näitasid, et algoritmi abil oli võimalik säästa iga tehinguga 0,1-0,2% aktsia turuväärtusest. Samuti olid algoritmiga edastatud tehingukorraldustega saadud kasumid kordades suuremad võrreldes algoritmita, ühes tükis esitatud korraldustega, seega võib algoritmi lugeda efektiivseks. Algoritmi tulemused viitasid sellele, et algoritmi efektiivsus on võrdelises seoses aktsia turuhinnaga, kuna katsete käigus oli kaks korda suurema turuhinnaga aktsia suurim kasum samuti ligilähedaselt kaks korda suurem.

Põhiraskuseks oli see, et kasutades börsiprogrammi demokasutajat ei ole võimalik anda parimaid ning täpsemaid vaatlustulemusi. Päril kasutajaga tehingukorralduse esitades tekib uueks tingimuseks ka mõju turutasakaalule ning aktsia turuhinnale, mida demokasutajaga polnud võimalik näidata. Turu tasakaalu mõjutamise olemasolu vähendaks tõenäoliselt väga suurel määral algoritmita esitatud tehingu kasumit. Demokasutaja piiranguks oli ka see, et mõningaid turuandmeid ei olnud võimalik kasutaja õiguste tõttu turult saada.

Lõputöö käigus koostatud algoritm ning programm täitsid enda eesmärgi ning katsete tulemused näitasid, et algoritm on efektiivne ning annab eelise võrreldes olukorraga, kus suure aktsiate koguarvuga tehingukorraldus esitatakse börsile ühes tükis. Siiski, kuna kasutusel oli demokasutaja, ei olnud antud lõputöö käigus võimalik näidata loodud ajas hajutamise algoritmi täielikku potentsiaali. Loodud algoritmi on võimalik ka mitmesugustesse süsteemidesse integreerida, mille käigus on võimalik seda kasutada päris turuolukorras. Samuti on võimalik töös loodud programmi edasi arendada, lisades sellele lisaväärtuseid ning suur eelis loodud algoritmi juures on ka see, et kõiki parameetreid on võimalik vastavalt vajadusele muuta, mis muudab selle mitmekülgsemaks.

Kasutatud kirjandus

- [1] LHV Investeerimisõpik, [Võrgumaterjal]. Saadaval: <https://fp.lhv.ee/academy/investmentguide> [Kasutatud 25.10.2020]
- [2] U.S Securities and Exchange Commission, „Investor Bulletin: Understanding Order Types,“ 12.07.2017. [Võrgumaterjal]. Saadaval: <https://www.investor.gov/introduction-investing/general-resources/news-alerts/alerts-bulletins/investor-bulletins-14> [Kasutatud 18.12.2020]
- [3] IBKR Order Types and Algos, Interactive Brokers. [Võrgumaterjal]. Saadaval: <https://www.interactivebrokers.com/en/index.php?f=4985> [Kasutatud 04.01.2021]
- [4] T. Hendershott, C. M. Jones, A. J. Menkveld, “Does Algorithmic Trading Improve Liquidity?,” The Journal of Finance, Veebruar 2011. [Võrgumaterjal]. Saadaval: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.2010.01624.x> [Kasutatud: 25.10.2020]
- [5] K. Kim, “Electronic and Algorithmic Trading Technology, The Complete Guide, 1st edition,” 20.06.2007. [Võrgumaterjal]. Saadaval: <https://books.google.ee/books?id=xYaW3l23h4sC&printsec=frontcover> [Kasutatud: 02.11.2020]
- [6] J. Chen, „Algorithmic Trading,“, Investopedia, 26.10.2020. [Võrgumaterjal]. Saadaval: <https://www.investopedia.com/terms/h/high-frequency-trading.asp> [Kasutatud: 02.11.2020]
- [7] C. C. Moallemi, M. Saglam, „The Cost of Latency,“ 19.07.2011. [Võrgumaterjal]. Saadaval: <http://home.business.utah.edu/finmh/moallemi.pdf> [Kasutatud: 03.11.2020]
- [8] J. Chen, „High-Frequency Trading,“, Investopedia, 10.10.2019. [Võrgumaterjal]. Saadaval: <https://www.investopedia.com/terms/h/high-frequency-trading.asp> [Kasutatud: 03.11.2020]
- [9] P. Treleaven, M. Galas, V. Lalchand, „Algorithmic Trading Review,“ Communications of the ACM, November 2013, 76-79
- [10] „Introduction to trade execution algorithms,“ BlazePortfolio, 25.02.2019. [Võrgumaterjal]. Saadaval: <https://blazeportfolio.com/blog/introduction-to-trade-execution-algorithms-2/> [Kasutatud 26.12.2020]
- [11] A. Kokaz, “Trading Execution Algorithms,” 03.07.2020. [Võrgumaterjal]. https://medium.com/@ali_kokaz/trading-execution-algorithms-17671e315bc8 [Kasutatud 04.01.2021]
- [12] “TWS”, Interactive Brokers. [Võrgumaterjal]. Saadaval: <https://www.interactivebrokers.com/en/index.php?f=14099> [Kasutatud: 28.12.2020]

- [13] Trading API Solutions, Interactive Brokers. [Võrgumaterjal]. Saadaval:
<https://www.interactivebrokers.com/en/index.php?f=5041> [Kasutatud: 28.12.2020]
- [14] Smart routing, Interactive Brokers. [Võrgumaterjal]. Saadaval: <https://ibkr.info/node/728>
[Kasutatud 04.01.2021]
- [15] R. Almgren, N. Chriss, „Optimal Execution of Portfolio Transactions,“ 08.04.1999.
[Võrgumaterjal]. Saadaval:
<https://pdfs.semanticscholar.org/3d2d/773983c5201b58586af463f045befae5bbf2.pdf>
[Kasutatud: 12.12.2020]
- [16] J. Gatheral, „Optimal order execution,“ 04.10.2011. [Võrgumaterjal]. Saadaval:
<http://faculty.baruch.cuny.edu/jgatheral/JOIM2011.pdf> [Kasutatud: 18.12.2020]
- [17] Security Lookup, AlphaQuery. [Võrgumaterjal]. Saadaval:
<https://www.alphaquery.com/lookup> [Kasutatud 22.12.2020]
- [18] Paper Trader Delayed Data Information, Interactive Brokers. [Võrgumaterjal]. Saadaval:
<https://www.interactivebrokers.com/en/index.php?f=17260> [Kasutatud 31.12.2020]

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Ronald Ruusa

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Väärtpaberitehingute ajas hajutamise algoritmid ja nende rakendamine“, mille juhendaja on Tõnn Talpsepp
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

05.01.2021

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.