

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Aaron Avaste IAIB185325

**Maakonnaliini bussiliikluse optimeerimine  
Tallinn-Jüri suuna näitel lokaalse otsingu  
põhimõttel**

Bakalaureusetöö

Juhendaja: Marko Kääramees  
PhD

Tallinn 2023

## **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: Aaron Avaste

22.05.2023

## Annotatsioon

Maakonnaliinidel sõidavad igapäevaselt tuhanded inimesed. Graafikuid muudetakse pidevalt ning mõlemale osapoolale, teenuse pakkujale ja kasutajatele, sobiliku graafiku koostamine on keerukas probleem.

Sarnast probleemi on lahendatud linna tingimustes eeldusega, et inimeste nõudlus bussi järgi on ühtlane või allub ette määratud jaotusele, kuid antud töö raames uuritakse maakonnaliine, kus nõudlus busside järgi on ebahürtlase jaotusega. Autor viis läbi küsimustiku, et paremini aru saada inimeste liikumisharjumustest maakonnaliinidel ning paljundas andmeid, et tekitada reaalsusele sarnane testimissüsteem.

Antud töö eesmärk on välja töötada optimeeriv algoritm, mis põhinedes inimeste liikumisharjumustel koostaks sõiduplaani, mis oleks optimaalsem nii bussifirmale teenitava kasumi kui sõitjate rahulolu poolest töö kirjutamise hetkel eksisteerivast graafikust, pidades silmas eelkõige inimeste liikumisharjumustest tulenevat nõudlust. Töö jaoks püstitati 2 hüpoteesi:

- 1) On võimalik koostada optimaalsem bussigraafik kaotamata märgatavat osa nõudlusest.
- 2) Kõige kasumlikum sõiduplaan ei teeninda maksimaalset hulka reisijaid.

Töö koosneb kolmest osast: taust, teostus ja tulemuse valideerimine. Taust kirjeldab probleemi olemust, toob välja praegu kättesaadavad lahendused sarnastele probleemidele ning hõlmab kirjeldust kasutatavast tarkvarast. Teostuse all on ülevaade andmete kogumisest, nende analüüsimisest ja optimeerivast algoritmist. Tulemuse valideerimise osas näidatakse, miks lokaalse otsingu põhimõttel saadakse kasumlikum sõiduplaan kui 20.05.2023 seisuga kasutuses olev sõiduplaan.

Koostatud optimeeriv algoritm oli võimeline koostama optimaalsema algoritmi, kui töö kirjutamise ajal kasutuses olnud sõiduplaan. Kasutuses olev sõiduplaan antud töös kogutud lähteandmete puhul tõestas, et ei ole kasumlik, samas kui optimeeritud

algoritmiga on mitmeid erinevaid valikuid, kui suurte busside arvudega on võimalik kasumit teenida. Samuti võis koostatud graafik olla kasumlikum töö eelduseks võetud lihtsustuste või lähteandmete kogumisviisi tõttu.

Ka teine hüpotees osutus tõeseks, sest maksimaalse soovitud reisidega tõuseb kulu ja tulu suhe. Põhjusteks on busside liisingumaksud ning väikese mahutavusega sõitude korraldamine, et ka üksikud reisijad jõuaksid oma soovitud sihtpunkti soovitud ajal.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 19 leheküljel, 4 peatükki, 15 joonist.

## **Abstract**

### **Bus schedule optimization of the county line using the example of the Tallinn-Jüri route based on the principle of local search**

Thousands of people in Estonia use county lines daily. Schedules are modified often and it is difficult to create Schedule to benefit both – the service provider and user.

Similar problems have been solved with assumptions of more evenly set passenger arrival times. The author of this work created a survey to collect initial data to better understand the habits of movement using the county bus lines with a simple data multiplication in order to create a more realistic testing system.

The main goal of the thesis is to create an optimizing algorithm based on local search, which would be more profitable for the service provider and more suitable for possible passengers.

The first subtask was to prove it is possible to create a more optimal bus schedule without losing a big portion of passengers.

The second subtask was to prove that the most profitable schedule does not serve the maximum number of passengers.

Although these subtasks were proven, it is not possible to be entirely sure about the result of this thesis as there were problems with the code, but the logic itself is still accurate in theory.

The thesis consists of 3 sections – background, realization and validation. The background specifies solutions to similar problems and reasons why they cannot be entirely applied to this specific case. Realization explains how data was collected and worked through to create an initial bus schedule. The validation section specifies how the result schedule of optimized and greedy algorithm was more profitable than schedule that was in use at the time of the writing.

The thesis is in Estonian and contains 19 pages of text, 4 chapters, 15 figures.

## Lühendite ja mõistete sõnastik

CSV	<i>Comma separated values</i> , failitüüp, mis sisaldab komaga eraldatud väärtuseid
ZIP	Faili formaat, mis toetab kadudeta andmepakkimist
VRPTW	Vehicle Routing Problem with Time Windows, sõidukite teekonna planeerimise probleem koos aja vahemikega
SOS	Schedule Optimization System, graafiku optimeerimise süsteem
EWT	Excess Wait Time, liigne ootamise aeg

## Sisukord

1 Sissejuhatus .....	9
2 Taust .....	10
2.1 Probleemi kirjeldus .....	10
2.2 Eelnev käsitlus .....	11
2.3 Kasutatav tarkvara .....	13
3 Teostus .....	14
3.1 Andmete kogumine .....	14
3.1.1 Bussipeatused .....	14
3.1.2 Reisiandmed .....	15
3.2 Andmete töötlemine ja salvestamine .....	17
3.3 Ahne algoritm .....	20
3.4 Optimeerimine lokaalse otsingu põhimõttel .....	21
3.5 Tulemuse valideerimine .....	23
4 Kokkuvõte .....	29
Kasutatud kirjandus .....	30
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	31



# 1 Sissejuhatus

Maakonnaliinidel sõidavad igapäevaselt tuhanded inimesed. 2018. aastal, kui tasuta sõiduõigused laienesid ka maakonnaliinidele, suurenes tehtud sõitude arv 10000 võrra [1]. Graafikuid muudetakse pidevalt ning mõlemale osapoolale, teenuse pakkujale ja kasutajatele, sobiliku graafiku koostamine on siiani autorile teadaolevalt lahendamata probleem maakonnaliinidel.

Sarnast probleemi on lahendatud linna tingimustes eeldusega, et inimeste nõudlus bussi järgi on ühtlane või allub ette määratud jaotusele, kuid antud töö raames uuritakse maakonnaliine, kus nõudlus busside järgi on ebaühtlane. Autor viis läbi küsimustiku, et paremini aru saada inimeste liikumisharjumustest maakonnaliinidel ning paljudas andmeid, et tekitada reaalsusele sarnane testimissüsteem.

Antud töö eesmärk on välja töötada optimeeriv algoritm, mis põhinedes inimeste liikumisharjumustel koostaks sõiduplaani, mis oleks optimaalsem nii bussifirmale teenitava kasumi kui sõitjate rahulolu poolest töö kirjutamise hetkel eksisteerivast graafikust, pidades silmas eelkõige inimeste liikumisharjumustest tulenevat nõudlikkust. Töö jaoks püstitati 2 hüpoteesi:

- 1) On võimalik koostada optimaalsem bussigraafik kaotamata märgatavat osa nõudlusest.
- 2) Kõige kasumlikum sõiduplaan ei teeninda maksimaalset hulka reisijaid.

Töö koosneb kolmest osast: taust, teostus ja tulemuse valideerimine. Taust kirjeldab probleemi olemust, toob välja praegu kättesaadavad lahendused sarnastele probleemidele ning hõlmab kirjeldust kasutatavast tarkvarast. Teostuse all on ülevaade andmete kogumisest, nende analüüsimisest ja optimeerivast algoritmist. Tulemuse valideerimise osas võrreldakse olemasoleva graafiku kasumlikkust ahnelt ja optimeeritult koostatud graafikute kasumlikkusega.

## 2 Taust

Antud peatükis uuritakse, millised asjaolud mõjutavad Eesti maakonnaliinide sõidugraafiku koostamist. Kuidas hinnata kulu ja tulu ning milliste piirangutega arvestada maakonnaliinide kasutamise osas. Lisaks arutletakse, miks olemasolevad programmid ei sobi antud tingimustes sõidugraafiku koostamiseks.

### 2.1 Probleemi kirjeldus

Eesti üks suuremaid transpordisüsteeme on maakonnaliinid, mis seovad suuremaid asustusi väiksematega. Seda kasutatakse igapäevaselt, et käia tööl, koolis või kasvõi poes sisseoste tegemas. Teenusepakkuja jaoks on see suur kulu allikas, kuid kui sõidugraafik on õigesti koostatud, saab sellest tulus ettevõtmine. Antud peatükis analüüsitakse täpsemalt probleemi olemust, millest sõltub kasumi teenimine sõiduteenuse pakkuja jaoks.

Maakonnaliinide kasutusharjumus erineb märkimisväärselt tiheasustuse sisesest harjumusest. Maakonna bussiliinide asutusotstarbeks on üsna võrdselt tuttavate/sugulaste külastamine, vaba aja veetmine, sisseostude tegemine ning tööl/koolis käimine. Teenuse kasutajad ei ole rahul auto ja ning jalgratta parkimise võimalustega peatustes, maakonnaliinide ühenduste arvuga, marsruutide ja ümberistumise võimalustega, sõiduplaanide ja nende ühilduvusega rongide ja teiste transpordiliikidega ning lapsevankri ja suurema pagasiga reisimisega. [2]

Bussigraafiku genereerimisel on määravaks 3 peamist muutujat – püsikulu, muutuvkulu ja transporditeenuse piletimüügist saadav tulu. Bussifirma muutuvkulu sõltub kasutuses olevate busside arvust ja liinide pikkusest. Kui suurendada busside arvu, suureneb bussijuhtide arv ning kütusekulu.

Bussifirma püsikulu arvutatakse kasutades bussijuhi brutopalka, mis antud töö raames arvestatakse kui 2000 eurot bussijuhi kohta.

Antud töö raames arvutatakse muutuvkulu kilomeetri kohta arvestusega, et teenuse pakkumiseks kasutatakse uuemaid gaasibusse, mille kulu teenusepakkujale 1 kilomeetri kohta on 0.77 eurosentit aastal 2023. See kulu hõlmab endas nii kütust kui remondi ja hoolduskulusid. [3]

Samas suureneb sobivus bussiga sõitjate jaoks, mistõttu on klientuur laiem ning piletitulu suurem. Seega tuleb leida moodus, kuidas vähendada bussifirma kulu samal ajal maksimeerides teenitavat tulu.

Bussigraafikus on esimene sõidu algusaeg Tallinn-Jüri-Tallinn suunal kasutuses olevas sõidugraafikus kell 5:20 ning viimane saabub 22:51. Küll aga jõuab maakonnaliinide busse kõige hilisemalt veel 23:30, mistõttu antud töö raames arvestatakse ajalise piiranguga busside liikumise osas 05:00-23:59. Üks buss mahutab kuni 50 inimest. Buss alustab liikumist alati Tallinnast Balti Jaamast, ja sõidab läbi mõlemad suunad jõudes tagasi Balti jaama, suundade vahel on lubatud ooteaeg.

Läbi viidud küsimustikust selgunud andmetest kasutatakse vaid 1 bussipeatust ning bussiajad töötatakse välja eeldusel, et 1 kuu on täpselt 4 nädalat ehk 28 päeva pikk. Kõik intervallide algusajad, mis lähevad üle kesköö teisendatakse ümber 23:45 ning lõpuajad 23:59. Kui vastus bussiga sõitmise korduste arvu kohta on negatiivne arv, siis on eeldatud, et on mõeldud kuni määratud arvuni käimiskordi.

Arvestades kõiki piiranguid, tuleb koostada sõidugraafik, milles kasum oleks võimalikult palju suurem kui kulu, et teenusepakkuja oleks võimeline tagama ühistranspordi teenuse, millest võidavad ka teenuse kasutajad.

## **2.2 Eelnev käsitlus**

Ühistranspordi optimeerimine on olnud pikaaegne probleem. Võimalikult sidusa või tuluka sõiduplaani koostamiseks on lähenemisvariante mitmeid.

Sarnase temaatikaga lahendatud probleem on VRPTW (*Vehicle Routing Problem with Time Windows*). VRPTW lahendab teekonnaplaneerimise probleemi kasutades ajavahemikke punktide vahel liikumiseks. Antud töös käsitletav probleem proovib aga optimeerida ajavahemiku valimist konstantse marsruudi ja sõidule kuluva ajaga.

NEC Group pakub busside ajakava optimeerimiseks graafiku optimeerimise süsteemi SOS (*Schedule Optimization System*). Nende lahendus eeldab suuremas koguses busse ning proovib optimeerida väiksemat ooteaega inimestele, kuid see ka tähendab, et busse paigutatakse väikse ajavahemiku peale palju kokku. Antud metoodikat kasutatakse näiteks Ahmedabadis. [4]

Qun Chen avalikustas teadusartikli teemal globaalne optimeerimine bussiliinide ajatabeli koostamise probleemile. Kogu probleem on lahendatud eeldusel, et reisijad ja bussid kogunevad ja sõidavad võrdsete aegade tagant. Lahenduses optimeeritakse ooteaegu kui on teada, mitu bussi sõidab. [5]

Xiao-yue Yang, Xin-yu Li, Jun Liang ja Cheng Peng on välja töötanud mitme eesmärgiga optimeeriva algoritmi SPEA2, millest arendati välja ISPEA (*Improved Strength Pareto Evolutionary Algorithm*). Sellega maksimeeritakse bussifirma tulu ja minimeeritakse sõitja rahulolematust ja ootamiskulu. Selle tugevuseks on mitme erineva variandi leidmine, nii et bussifirma saab ise valida sobivaima. Küll aga töötab see eeldusel, et inimesed, kes sisenevad valitud peatusest määratud hetkel vastavad Poissoni jaotusele. [6]

Kuna maakonnaliinidel ei esine nii ühtlast reisijate jõudmist peatusesse päeva läbilõikes, vaid moodustuvad ajahetked, kus nõudlus on suurem, siis ei sobi SOS, globaalne optimeerimine ega SPEA2 maakonnaliinide bussigraafiku koostamiseks.

F. D. Wihartiko, A. Buono ja B. P. Silalahi lahendasid bussiaegade planeerimise probleemi kasutades *integer programming model*'it, mida täiustati modifitseeritud geneetilise algoritmiga. Modifitseerimine toimus kromosoomide disainis ning saavutas optimaalse tulemuse täpsusega 99,1%. Lahenduse töötamiseks peab aga bussifirma ise teadma, kui mitu bussi oleks optimaalne kasutada antud liinil. [7]

Maakonnaliinide jaoks eelnevalt välja toodud valikutest sobib *integer programming model* kõige paremini. Küll aga eeldab see väga detailseid lähteandmeid.

Antud töös käsitletakse konkreetselt maakonnaliinide bussigraafikuid. Kuna tegu ei ole linnaga ehk tihedalt asustatud alaga, siis reisijate arv on tunduvalt ebahütlasem. Maakonna liinidel on tihtilugu mitu suuremat asustust ühel liinil, seega toimub pidevalt ka nende asustuste vaheline liikumine ning suurem liikumine keskusesse, Harju

maakonna näitel Tallinnasse. Eelnevalt välja toodud uurimused keskenduvad eelkõige tihedalt asustatud linna transpordi optimeerimisele ja inimeste nõudlikkuse ebäühtluse tõttu ei rakendu optimaalselt maakonnaliinidele.

Varasemalt on uuritud rohkem tiheasustuse ehk linnaliinide optimeerimist või marsruudi parimat võimalikku valimist. See eeldab inimeste ühtlasemat peatustesse jaotumist ning läbi päeva ringi liikumist. Antud tööga otsitakse aga kõige paremaid ajahetki olemasolevate liinide jaoks eeldusel, et transporditeenuse kasutajad ei jagune võrdselt, vaid on aktiivsemad peatustes kindlatel ajaperioodidel.

### **2.3 Kasutatav tarkvara**

Töö eesmärgiks on välja töötada kasumlikum sõidugraafik ja seetõttu kirjutatakse töö kasutades Pythoni <sup>1</sup>programmeerimiskeelt. Tänu oma lihtsusele on Python kõige populaarsem programmeerimiskeel masinõppe ja tehisintellekti arendamiseks ning seetõttu sobib ka hästi optimeerimisülesande täitmiseks.

---

<sup>1</sup> <https://www.python.org/>

## **3 Teostus**

Teostus koosneb kahest osast, andmete kogumine ja töötlemine ning bussigraafikut koostav ja optimeeriv algoritm. Andmete kogumine hõlmab endas küsimustikku ning avalikust veebist kättesaadavaid andmeid. Bussigraafiku koostamine toimub kahes osas. Kõige pealt koostatakse ahne algoritmiga maksimaalne graafik, peale mida proovitakse seda optimeerida kasutades lokaalse otsingu põhimõtet.

### **3.1 Andmete kogumine**

Autor proovis saada busside täitumise andmeid otse lokaalselt teenusepakkujalt GoBus, lisaks sai uuritud andmete kohta ka Transpordiametilt ja Transpordiameti andmetöötlusega tegelevalt ettevõttelt Ridango, kuid andmetele ligipääsu ei saanud. Alternatiivina koostas autor andmete kogumiseks küsimustiku inimeste liikumisharjumuste uurimiseks.

Küsimustik põhines bussiliinidel, mis läbivad Jüri alevikku. Autor tegi just sellise valiku, sest antud bussiliinid läbivad nii ühte või mitut suuremat asulat kui ka väiksemaid vähemasustatud külasid.

Küsitlus viidi läbi Facebooki Rae valla grupi, sest see oli kõige paremini kättesaadav valim. Küsimustikule said vastata kõik Rae valla elanikud, kes kasutavad või sooviksid kasutada antud teenust. Küsitluse käigus ei kogutud lisainformatsiooni vastajate kohta, kuna kätte saadav ei ole ka vastav informatsioon teenuse reaalse kasutajaskonna kohta.

#### **3.1.1 Bussipeatused**

Bussiliikluse optimeerimiseks on vaja teada, millist liini paigutada vastavalt nõudele ja selleks on küsimustikku vaja vastajale ette anda loetelu peatustest, sest erinevatel liinidel

on erinevad peatused. Peatuste loetelu saamiseks kasutan veebilehe [peatus.ee](http://peatus.ee) teadmusbasi, mis on saadaval peatuse leheküljel <sup>1</sup> ZIP failina.

Bussipeatuste nimede saamiseks tuleb kõige pealt kätte GoBus AS vastav agentuuri identifikaator saada tekstifailist *agency*. Selle identifikaatori alusel sai autor teada teekonna identifikaatori tekstifailist *routes*. Teekonna id saamisel tuleb välja filtreerida liin, mille nimi sisaldab sõna Jüri ning välja filtreerida liinid, mida veel ei eksisteeri või on kunagi eksisteerinud. Lisaks tuleb üle ka kontrollida, et bussiliinid mitte ainult ei mainiks Jürist läbi sõitmist, vaid sõidaksid ka päriselt. Sinna kategooriasse langeb liin 134, mis mainib Jürist läbi sõitmist kuid tegelikkuses läbib vaid 1 peatust. Teekonna identifikaatori läbi saab leida sõidu identifikaatori tekstifailist *trips*, millega on seotud peatuste identifikaatorid failis *stop\_times*. Tekstifailist *stops* saab peatuste identifikaatorite läbi kätte peatuste nimed, millest eemaldades duplikaadid, sorteerides ja vormindades saab kätte kogu peatuste loetelu küsimustiku jaoks.

### **3.1.2 Reisiandmed**

Küsitlus põhines 6 küsimusel, mis moodustasid 2 küsimuste blokki. Mõlemad küsimuste blokid andsid vastuse kasutatavale peatusele ja sobivatele ajavahemikele nii nädala sees kui nädalavahetusel nagu on näha jooniselt 1. Blokid erinesid sõidusuuna poolest, esimene blokk uuris teekonda Tallinna poole ning teine osa käis vastupidise sõidusuuna kohta.

---

<sup>1</sup> <http://peatus.ee/gtfs/>

Palun valida kõige tavapärasemad peatused, kust sõidate bussiga suunaga Tallinna poole \*

Peatuse valik \*

Vali peatus

+

Lisage siia kõik ajavahemikud nädala sees, millal sõidaksite bussiga suunal Tallinna poole \*

Vahemiku algus (E-R) \*    Ideaalne kellaeg (E-R)    Vahemiku lõpp (E-R) \*    Mitu korda kuus käiksite (E-R) \*

▼ : ▼

▼ : ▼

▼ : ▼

0

+ lisa bussisõidu aeg

Lisage siia kõik ajavahemikud nädalavahetusel, millal sõidaksite bussiga suunal Tallinna poole \*

Vahemiku algus (L-P) \*    Ideaalne kellaeg (E-R)    Vahemiku lõpp (L-P) \*    Mitu korda kuus käiksite (L-P) \*

▼ : ▼

▼ : ▼

▼ : ▼

0

+ lisa bussisõidu aeg

Joonis 1: Ühe küsimuste bloki näide küsimustikust.

Küsimustik põhines bussiliinidel 120, 125, 132, 132A, 134, 135, 135C, 214, 244 ja 259. Peale küsimustiku laiali saatmist muutus ka bussigraafik, millega kaotati endised kommertsliinid 214, 244 ja 259. Graafikusse lisati bussiliinid 120A, 125A, 135A ja 164. Küsimustiku vastused, mis käisid peatuste kohta, mis asetsesid liinidel 214, 244 ja 259 asendati lähimate peatustega, kust liiguvad töö kirjutamise ajal eksisteerivad bussiliinid.

Küsimustik koostati keskkonnas JotForms, sest Google Forms keskkond ei suutnud pakkuda piisavat tehnilist tuge sobiliku küsimustiku koostamiseks. JotFormsi tasuta versioon võimaldab saada kuni 100 vastust, millest piisab inimeste liikumisharjumuste uurimiseks etteantud piirkonnas. JotForms võimaldab andmeid kätte saada CSV failina.

CSV faili esimene rida kirjeldab ära esitatud küsimused ning ülejäänud read iseloomustavad esitatud vastuseid struktuuril, mis on näha jooniselt 2.



```

"Peatuse valik: Jüri kirik
Peatuse valik: Ehituse
", "Vahemiku algus (E-R): 06:55, Ideaalne kellaaeg (E-R): 07:00, Vahemiku lõpp (E-R): 07:05, Mitu korda kuus käiksite (E-R): 18
Vahemiku algus (E-R): 07:50, Ideaalne kellaaeg (E-R): 07:55, Vahemiku lõpp (E-R): 08:00, Mitu korda kuus käiksite (E-R): 4
", "Vahemiku algus (L-P): 10:13, Ideaalne kellaaeg (E-R): , Vahemiku lõpp (L-P): 11:11, Mitu korda kuus käiksite (L-P): 0
", "Peatuse valik: Kivisilla
", "Vahemiku algus (E-R): 13:05, Ideaalne kellaaeg (E-R): 13:10, Vahemiku lõpp (E-R): 13:15, Mitu korda kuus käiksite (E-R): 9
Vahemiku algus (E-R): 14:00, Ideaalne kellaaeg (E-R): 14:05, Vahemiku lõpp (E-R): 14:10, Mitu korda kuus käiksite (E-R): 4
Vahemiku algus (E-R): 15:50, Ideaalne kellaaeg (E-R): 15:55, Vahemiku lõpp (E-R): 16:00, Mitu korda kuus käiksite (E-R): 9
", "Vahemiku algus (L-P): 14:13, Ideaalne kellaaeg (L-P): , Vahemiku lõpp (L-P): 17:10, Mitu korda kuus käiksite (L-P): 0
"

```

Joonis 2. CSV faili struktuuri näide.

### 3.2 Andmete töötlemine ja salvestamine

Küsitluse vastuste seas oli ka vastuseid, mille peatuse nimed ei vastanud päris liinide kasutamisele, mistõttu lammutati need vastused laiali, et oleks võimalik koostada sõit ümberistumisega.

Kogu andmestik loetakse klassis `TravelingTimes` andmestikku `data`, mida on näidatud joonisel 3.

```

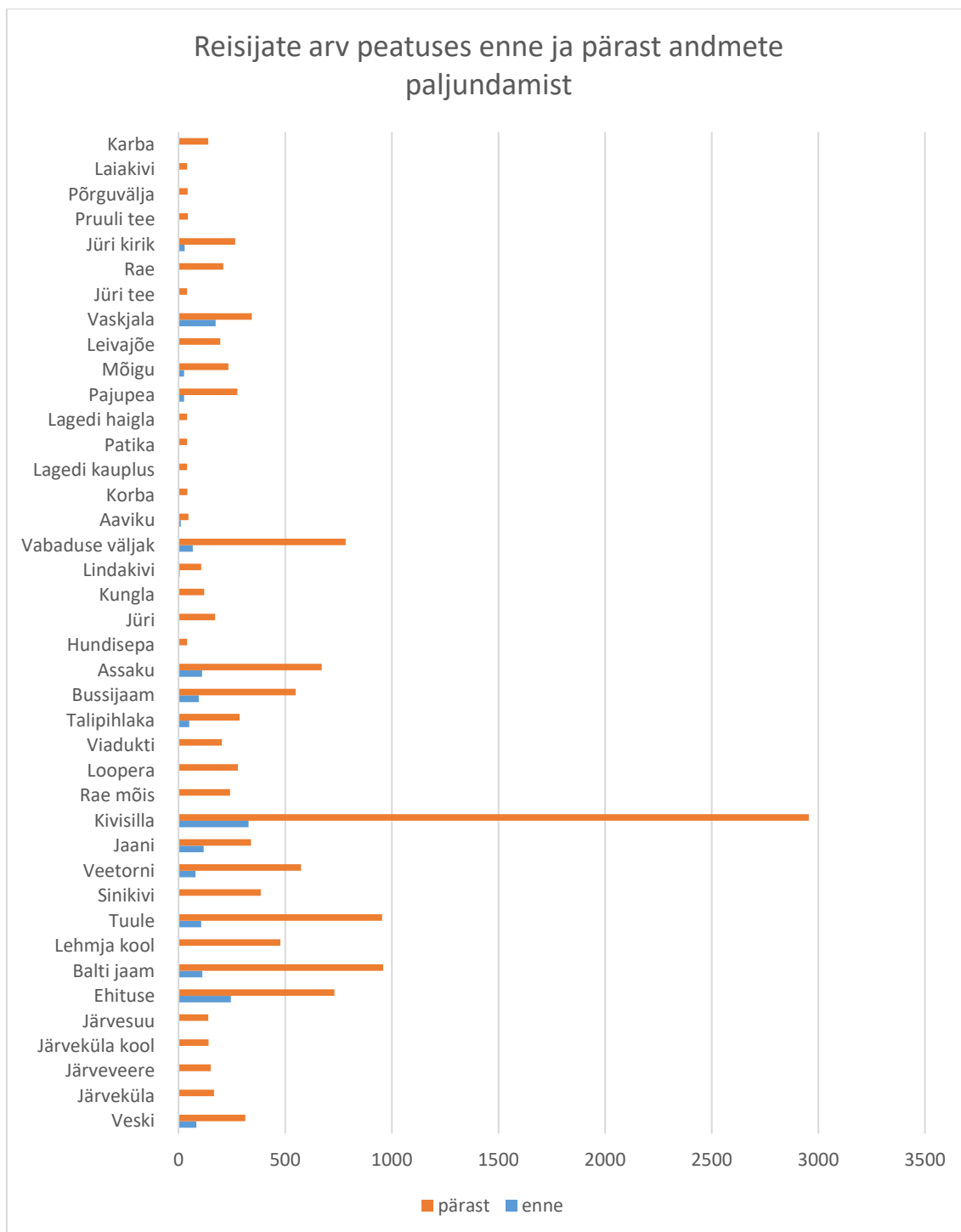
data = [
    peatused1 - [],
    ajad1(E-R) - [
        vahemiku algus, ideaalne aeg, vahemiku lõpp, sõidukordade arv -
        []],
    ajad1(L-P) - [
        vahemiku algus, ideaalne aeg, vahemiku lõpp, sõidukordade arv -
        []],
    peatused2 - [],
    ajad2(E-R) - [
        vahemiku algus, ideaalne aeg, vahemiku lõpp, sõidukordade arv -
        []],
    ajad2(L-P) - [
        vahemiku algus, ideaalne aeg, vahemiku lõpp, sõidukordade arv -
        []]
]

```

Joonis 3. Sisse lugemisel salvestatav andmestruktuur

Kuna küsimustikule vastajaid oli palju vähem kui tegelikkuses sõitjaid, tuli leida moodus, kuidas andmeid juurde genereerida. Eeldusel, et 2 järjestikuse peatuse kasutajaskond omab sarnaseid huvisid, saab tekitada samadel aegadel kasutajaskonda ka lähipeatustesse. Töös eeldatakse, et suurema reisijaskonnaga peatustest oli suurem tõenäosus saada ka vastus küsitluses. Järelikult peatused, mida küsitluse tulemused sisaldavad, omavad suuremat reisijaskonda kui lähipeatused.

Genereerimisel kasutatakse algpeatuse nõudlusest poolt peatustel, mis asetsevad enne või peale algpeatust ning kolmandikku nõudlusest peatustel, mis asetsevad 1 võrra veel lähemal või kaugemal lõpppeatusele. Sedasi tekib iga vastanu kohta nõudluse „püramiid“, mis võimaldab suurendada ja ühtlustada andmehulka kaotamata esialgsete andmete õigsust.



Joonis 4: Reisijate arv peatuses enne ja pärast andmete paljundamist

Antud andmete põhjal genereeritakse listid, kus indeksiga on seotud soovitud reisiga (muutuja soovitud\_reis) seonduvad andmed: soovitava reisi ajavahemiku algus-, parim ja lõppaeg, soovitud reisi kasutamiskordade arv (muutuja *necessity*), alg- ja lõpp-peatuse nimi, reisija ID ja sõidugraafiku variant.

Erinevaid sõidugraafiku variante (edaspidi *mode*) on kokku 4: nädalasisesed ja nädalavahetused sõidud Tallinnast Jüri poole ja Jürist Tallinna poole. *Mode* on justkui id, mille järgi soovitud reise põhjal koostatakse sidus graafik.

### 3.3 Ahne algoritm

Ahne algoritmi koostamiseks tuleb iga *mode* kohta eraldi sõiduplaan, mis kirjeldab konkreetselt selle variandi soovitud reise. Koostatakse list tõeväärtustest *bus\_found*, millega hoitakse meeles, millise soovitud reisi kohta on koostatud juba reis. Kuna listis hoitakse koos 4 erinevat graafiku versiooni, on algväärtuseks list, kus kõik väärtused on tõesed ja vääraks muudetakse vaid elemendid, mis kehtivad ühe *mode* kohta. Nõnda käiakse läbi kõik erinevad *mode*'d, kuni on koostatud sõiduplaan kõikidele soovitud reisidele. Mälus hoitakse soovitud reise arvu *unfound\_count*, mis iseloomustab, kui mitu soovitud reisi vajavad bussireisi. Nõnda saab vältida listi võrdlemist sama suure listiga, kus kõik väärtused oleksid tõesed.

Seejärel genereeritakse bussireisid alustades kõige suurema nõudlusega ajast lõpetades kõige vähem populaarse ajaga kuni kõigile soovitud reisidele on koostatud bussireis, millega reisija saab jõuda algpeatusest sihtkohta. Seda protsessi kirjeldab joonis 5.

```
bus_found = [True, True, ..., True]
unfound_count = 0
for mode in modes:
    for soovitud_reis in reisid:
        if soovitud_reis.mode == mode:
            bus_found[soovitud_reis] = False
            unfound_count += 1
    while unfound_count != 0:
        most_useful_time = get_reis_with_biggest_necessity()
        bus_travel = create_reis(most_useful_time)
        unfound_count -= bus_travel.get_scaled_passenger_count()
```

Joonis 5. Ahne algoritmi baasstruktuur

Selleks, et meetodiga *get\_reis\_with\_biggest\_necessity* leida kõige suurema nõudlusega aeg, kaardistatakse iga kellaaja kohta kogu sobilike reisijate arv. Selleks kasutatakse andmetüüpi sõnastik, kus võtmeväärtuseks on kellaeg ja sellele vastab reisijate arv, kes soovivad reisi sellel kellaajal.

Seejärel koostatakse bussireis *BusTravel* meetodiga *create\_reis*. Bussireis sisaldab endas liini, algus- ja lõppkellaega, soovitud reise ning *mode*. Andmetüüp kasutab soovitud

reise ja *mode* selleks, et arvutada kaaludega bussi täituvus, mida on kirjeldatud joonisel 6. Antud töö käigus kasutatakse kaaludega busside täituvust, sest soovitud reis ei võrdu reisijaga. Kui reisija soovib 3 nädalasisesel päeval transporditeenust kasutada, täidab ta bussist 12/20 kohta. Maksimaalne arv on nädalasisestel sõitudel 20 ja nädalavahetuseti 8, sest sõiduplaan koostatakse kuuajase arvestusega. Kaalutud reisijate arv tähendab seda, et bussireisi klientuur on laiem kui 50 inimest, sest nad satuvad bussile erinevatel aegadel vältides sellega bussi lõhki ajamist.

```
def get_scaled_passenger_count():
    day_count = 20 if self.mode % 2 == 1 else 8
    return sum(list(map(lambda x: round(soovitud_reis.necessity /
day_count, 2), soovitud_reisid)))
```

Joonis 6. Kaaludega reisisoovide kogusumma arvutusmeetod

Optimeerimaks saadud mahukat graafikut koostatakse uued ahned graafikud vähendades korruga kasutuses olevate busside arvu. Selleks käiakse läbi ahnelt saadud bussireisid ning tekitatakse puhver, mis hoiab mälus teatud kellaajal aktiivseid bussireise. Aktiivne reis tähendab bussireisi, mille algusaja ja lõpuaja vahele jääb vaadeldav ajahetk. Kui see puhver ületab oma maksimaalse aktiivsete reiside arvu, siis eemaldatakse väikseima kaaludega reisisoovidega reis puhvrilt. Nõnda filtreerides välja vähem kasutatavamad reisid valmib ahne sõidugraafik erinevate kasutuses olevate busside arvuga nagu on kujutatud joonisel 7.

```
puhver = []
max_busside_arv = N
for kellaag in (05:00-23:55):
    while puhver.aktiivsete_reiside_arv(kellaag) > max_busside_arv:
        puhver.remove_min_kasutusega_buss(puhver, max_busside_arv)
```

Joonis 7. Puhvri tööpõhimõte

Ahne algoritm koostab iga bussireisi kohta kasuteguri, kust joonistub selgelt välja ka kõige kasumlikum punkt. Kui lisada eeldus, et bussid on liisitud ning vastavalt bussile lisada püsikulu, nihkub optimaalne busside arv vähemuse poole. Andmete analüüs jätkub peatükis 3.5 Tulemuste valideerimine.

### 3.4 Optimeerimine lokaalse otsingu põhimõttel

Ahne algoritmi optimeerimiseks vähendatakse lokaalse otsingu põhimõttel kasutuses olevate busside arvu. Selleks käiakse läbi ahne algoritmiga koostatud bussigraafik 3 bussireisi kaupa, kus üritatakse tõsta võimalikult palju reisijaid esimesele bussireisile ning

ülejäänud 2 asendada 1 uue ahne bussireisiga, mille kogutulu oleks parem, kui eelnevalt olemas olnud kahel bussisõidul kokku. Taolist reisijate liigutamist esimesele bussile ning vabastamist ülejäänud bussidelt kirjeldab ka joonis 9.

```
def move_passengers(previous_bus, bus_travel, next_bus, free_passengers)
    cleared_passenger_list = bus_travel.get_passengers().copy()
    for passenger in bus_travel.get_passengers():
        if previous_bus.can_add_passenger(passenger):
            previous_bus.add_passenger(passenger)
            cleared_passenger_list.remove(passenger)
    free_passengers.add_all(cleared_passenger_list)
    cleared_passenger_list = next_bus.get_passengers().copy()
    for passenger in next_bus.get_passengers():
        if previous_bus.can_add_passenger(passenger):
            previous_bus.add_passenger(passenger)
            cleared_passenger_list.remove(passenger)
    free_passengers.add_all(cleared_passenger_list)
```

Joonis 9. Reisijate liigutamine bussireiside vahel

Uus bussireis proovitakse mahutada ajavahemikku, kus uue sõidu algusaeg on võrdne või hilisem kui varem olnud 2. bussisõidu algusaeg ning mitte hilisem kui 3. bussisõidu lõppaeg. Kõik vabad reisijad hoitakse listis, mida uuendatakse jooksvalt iga tsükli kord, kui proovitakse uut bussi lisada, et leida maksimaalse kasuga ajahetk, mis oleks tulusam kui esialgne graafik. Sellist minimeerimissüsteemi iseloomustab joonis 10.

```
optimized_schedule = greedy_schedule.copy()
for bus_travel in greedy_schedule:
    if bus_travel not in optimized_schedule:
        continue
    previous_bus = optimized_schedule[optimized_schedule.index(bus_travel)
- 1]
    next_bus = optimized_schedule[optimized_schedule.index(bus_travel) +
1]
    move_passengers(previous_bus, bus_travel, next_bus, free_passengers)
    for time in range(bus_travel.get_start_time(),
next_bus.get_end_time()):
        bus = create_bus_for_time(time)
        value = bus.get_value()
        if value > max_value:
            best_bus = bus
            max_value = value
```

Joonis 10. Ahne bussi genereerimine lokaalse otsingu tulemusele

Kuna mitme erineva bussiliiniga kõikide võimaluste läbiarvutamine võtab märkimisväärselt palju aega, kasutatakse antud töös kitsenduseks vaid ühte bussiliini, milleks on liin numbriga 135. Autor valis just selle bussiliini, sest tegu on pikema liiniga,

mistõttu hõlmab see endas rohkem peatuseid ning seetõttu sobib suuremale hulgale reisijatest.

### **3.5 Tulemuse valideerimine**

Ühe reisi tulu arvutatakse universaalse valemi järgi kasutades standardpileti hinda, mis antud töö raames on 4 eurot reisi kohta. See hõlmab endas piletihinda ja kompensatsiooni teenusepakkuja palkajalt, et tagada piirkonnas transpordi võimalus.

Töös kasutatakse standardhinda, sest reaalses elus on piletihinnad erinevad varieerudes maksimaalsest 1.9 eurost ühe sõidu kohta Harjumaa 1. ja 2. tsooni siseselt kuni kuupileti omamiseni, mis teeb keskmise sõidu hinna arvutamise kliendi jaoks keeruliseks nagu on välja toodud Põhja-Eesti Ühistranspordikeskuse lehel<sup>1</sup>.

Kulu arvutatakse kasutades bussiliini teekonnapiikkuse korrutist uue gaasibussi keskmise kuluga, milleks on 0.77 eurot kilomeetri kohta. Liinipikkus antud töö raames on mõõdetud käsitsi kasutades kaardirakendus Google Maps. Bussijuhi kulu arvutatakse korrutades bussijuhi keskmine palk, mis antud töö raames on arvestatud kui 12.5 eurot tunni kohta.

Lisaks saab lisada tuluarvutusele lisakuluna liisingumaksud, mis mõjutavad märgatavalt kulu rohkemate busside arvudega graafikute juures.

Olemasolev sõiduplaan võeti Tallinna transpordiameti Harju maakonna busside sõiduplaanist. Olemasolev sõiduplaan sisaldas liine 120, 120A, 125, 125A, 132, 132A, 135, 135A ja 135C, sest neid liine kasutab ka antud töö.

---

<sup>1</sup> <https://www.ytkpohja.ee/piletite-hinnad>

LINE120: 07:45-08:19    LINE120: 08:35-08:19  
 LINE120: 12:00-12:34    LINE120: 13:10-12:34  
 LINE120: 15:30-16:04    LINE120: 16:25-16:04  
 LINE120: 16:00-16:34    LINE120: 17:00-16:34  
 LINE120: 16:30-17:04    LINE120: 17:30-17:04  
 LINE120: 18:40-19:14    LINE120A: 19:35-19:14  
 LINE120: 19:15-19:49    LINE120: 20:25-19:49  
 LINE120: 20:40-21:14    LINE120A: 21:15-21:14  
 LINE120: 21:35-22:09    LINE120: 22:20-22:09  
 LINE120A: 05:20-05:56    LINE120: 06:00-05:56  
 LINE120A: 06:40-07:16    LINE120: 07:25-07:16  
 LINE120A: 07:25-08:01    LINE120: 08:10-08:01  
 LINE120A: 08:30-09:06    LINE120A: 09:15-09:06  
 LINE120A: 09:30-10:06    LINE120: 10:15-10:06  
 LINE120A: 11:05-11:41    LINE120A: 12:20-11:41  
 LINE125: 17:40-18:24    LINE125: 18:35-18:24  
 LINE125A: 06:10-06:56    LINE125: 06:55-06:56  
 LINE125A: 14:10-14:56    LINE125A: 15:10-14:56  
 LINE132: 06:15-06:59    LINE132: 07:05-06:59  
 LINE132: 07:05-07:49    LINE132: 07:55-07:49  
 LINE132: 08:50-09:34    LINE132: 09:45-09:34  
 LINE132: 10:30-11:14    LINE132: 11:25-11:14  
 LINE132: 13:20-14:04    LINE132: 14:15-14:04  
 LINE132: 15:35-16:19    LINE132: 16:30-16:19  
 LINE132: 16:50-17:34    LINE132: 17:45-17:34  
 LINE132: 19:35-20:19    LINE132: 20:40-20:19  
 LINE132: 21:05-21:49    LINE132: 21:55-21:49  
 LINE132: 22:25-23:09    LINE132: 23:15-23:09  
 LINE132A: 14:50-15:47    LINE132A: 15:55-15:47  
 LINE132A: 18:10-19:07    LINE132A: 19:15-19:07  
 LINE135: 06:20-07:23    LINE135: 07:15-07:23  
 LINE135: 05:15-06:18    LINE135: 06:10-06:18  
 LINE135: 12:50-13:53    LINE135: 14:15-13:53  
 LINE135: 17:05-18:08    LINE135: 18:25-18:08  
 LINE135: 18:15-19:18    LINE135: 19:25-19:18  
 LINE135: 20:10-21:13    LINE135: 21:20-21:13  
 LINE135A: 07:10-08:04    LINE135A: 08:30-08:04  
 LINE135A: 14:00-14:54    LINE135A: 15:00-14:54  
 LINE135A: 15:40-16:34    LINE135A: 16:45-16:34  
 LINE135C: 09:10-10:20    LINE135C: 10:30-10:20

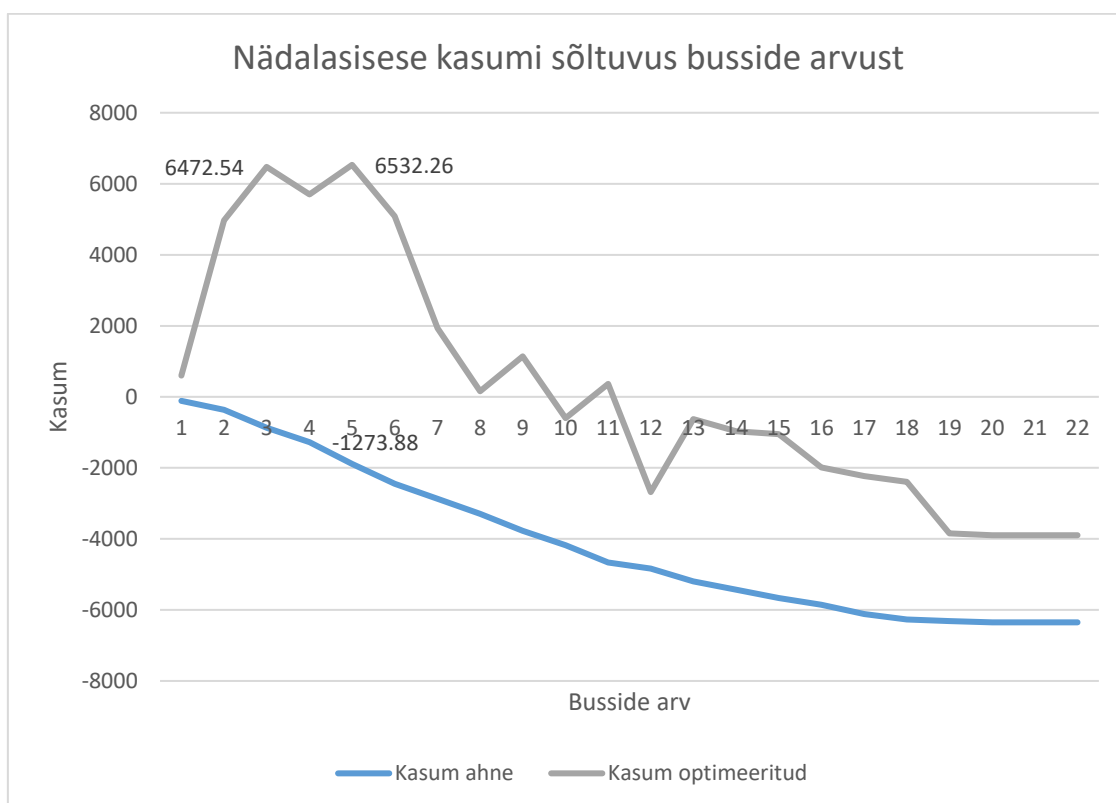
Joonis 11. Kasutuses olev sõidugraafik

Kasutades saadud lähteandmeid selgus, et olemasolev sõidugraafik, mida on kujutatud joonisel 11, ei ole sõiduteenuse pakkuja jaoks kasumlik, vaid teenib kahjumit. Nädalasisene kahjum terve kuu koondtulemusena jäi 1300.3 eurot ning nädalavahetusesti 512.03 eurot. See tähendab ligi 1800 eurost kahjumit kuus antud lähteandmetel.



Kasutuses olev sõidugraafik kasutab nädalasiseselt 4 erinevat füüsilist bussi ning nädalavahetustel 2, mis vahetavad jooksvalt liine, et teenuse kasutajaskond jõuaks oma soovitud sihtpunkti.

Joonisel 12 on näha nädalasiseste sõitude kasumi sõltuvust busside arvust, kus on võrdlusesse toodud ahne ja optimeeritud algoritmi tulemuste erisus. Numbriliselt on näha optimeeritud algoritmiga koostatud graafiku kasumihinnang 3 ja 5 füüsilise bussi kasutusega graafikutel ning ahne algoritmiga koostatud 2 füüsilist bussi kasutava graafiku kasumihinnang.



Joonis 12. Erinevate algoritmide nädalasisese kasumi sõltuvus busside arvust

Ahne algoritm saavutas olemasolevast graafikust pisut parema tulemuse, kuid sisaldab endiselt mitmeid sõite, mida kasutavad väiksem arv reisijaid. Nädalasiselt teenib ahne algoritmiga koostatud graafik kahjumit 1273.88 eurot, kui kasutuses on 4 erinevat füüsilist bussi.

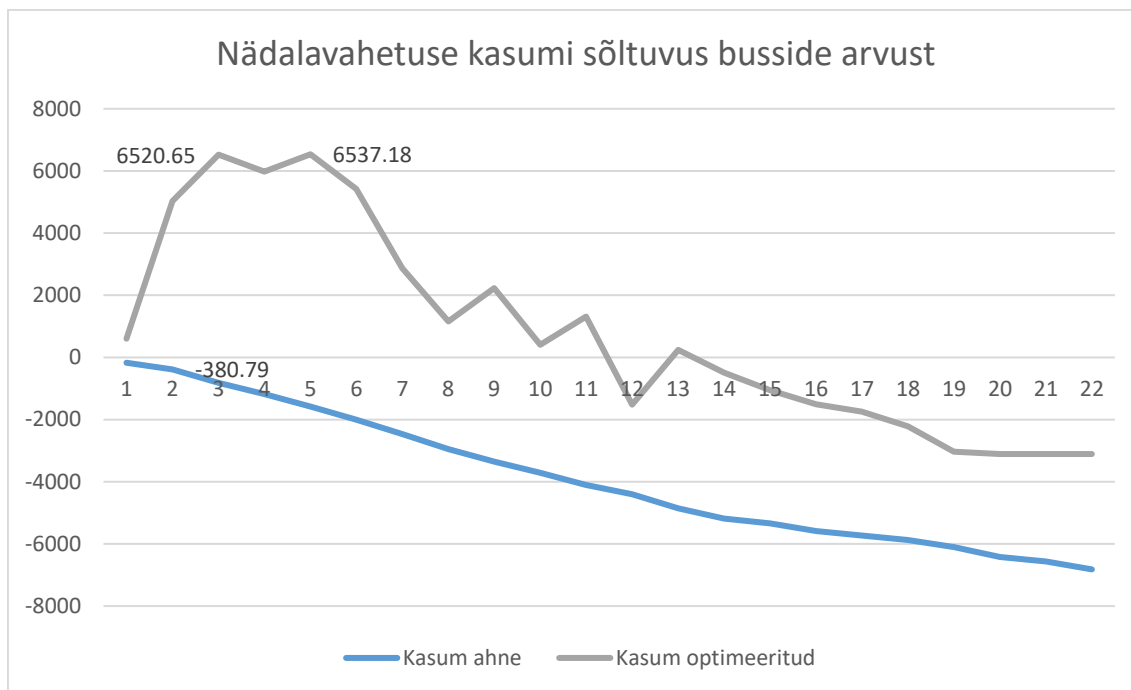
Optimeeriva algoritmi tulemused olid märksa paremad, kuna mitmed tühjad sõidud asendati väiksemate arvude sõitudega, mistõttu oli kahjum märksa väiksem. Graafikult on näha, et optimaalsem on kasutada 3 või 5 füüsilist bussi, mis teeniksid kuus nädalasiseselt kasumit vastavalt 6472.54 ja 6532.26 eurot.

Nädalasisest 5 erinevat füüsilist bussi kasutatavat graafikut kujutab joonis 13. Selle graafikuga on kuus teenitav kasum suurim.

LINE120: 07:45-08:19    LINE120: 08:35-08:19  
LINE120: 12:00-12:34    LINE120: 13:10-12:34  
LINE120: 15:30-16:04    LINE120: 16:25-16:04  
LINE120: 16:00-16:34    LINE120: 17:00-16:34  
LINE120: 16:30-17:04    LINE120: 17:30-17:04  
LINE120: 18:40-19:14    LINE120A: 19:35-19:14  
LINE120: 19:15-19:49    LINE120: 20:25-19:49  
LINE120: 20:40-21:14    LINE120A: 21:15-21:14  
LINE120: 21:35-22:09    LINE120: 22:20-22:09  
LINE120A: 05:20-05:56    LINE120: 06:00-05:56  
LINE120A: 06:40-07:16    LINE120: 07:25-07:16  
LINE120A: 07:25-08:01    LINE120: 08:10-08:01  
LINE120A: 08:30-09:06    LINE120A: 09:15-09:06  
LINE120A: 09:30-10:06    LINE120: 10:15-10:06  
LINE120A: 11:05-11:41    LINE120A: 12:20-11:41  
LINE125: 17:40-18:24    LINE125: 18:35-18:24  
LINE125A: 06:10-06:56    LINE125: 06:55-06:56  
LINE125A: 14:10-14:56    LINE125A: 15:10-14:56  
LINE132: 06:15-06:59    LINE132: 07:05-06:59  
LINE132: 07:05-07:49    LINE132: 07:55-07:49  
LINE132: 08:50-09:34    LINE132: 09:45-09:34  
LINE132: 10:30-11:14    LINE132: 11:25-11:14  
LINE132: 13:20-14:04    LINE132: 14:15-14:04  
LINE132: 15:35-16:19    LINE132: 16:30-16:19  
LINE132: 16:50-17:34    LINE132: 17:45-17:34  
LINE132: 19:35-20:19    LINE132: 20:40-20:19  
LINE132: 21:05-21:49    LINE132: 21:55-21:49  
LINE132: 22:25-23:09    LINE132: 23:15-23:09  
LINE132A: 14:50-15:47    LINE132A: 15:55-15:47  
LINE132A: 18:10-19:07    LINE132A: 19:15-19:07  
LINE135: 06:20-07:23    LINE135: 07:15-07:23  
LINE135: 05:15-06:18    LINE135: 06:10-06:18  
LINE135: 12:50-13:53    LINE135: 14:15-13:53  
LINE135: 17:05-18:08    LINE135: 18:25-18:08  
LINE135: 18:15-19:18    LINE135: 19:25-19:18  
LINE135: 20:10-21:13    LINE135: 21:20-21:13  
LINE135A: 07:10-08:04    LINE135A: 08:30-08:04  
LINE135A: 14:00-14:54    LINE135A: 15:00-14:54  
LINE135A: 15:40-16:34    LINE135A: 16:45-16:34  
LINE135C: 09:10-10:20    LINE135C: 10:30-10:20

Joonis 13. Optimeeritud nädalasisene graafik kasutades 5 füüsilist bussi

Joonisel 14 on välja toodud samane võrdlus nädalavahetuse sõidugraafikul. Andmetena on välja toodud optimeeritud algoritmi tulemuse hinnatud väärtused 3 ja 5 bussi kasutusega graafikutel ning ahne algoritmiga 2 füüsilist bussi kasutava graafiku väärtus.



Joonis 14. Erinevate algoritmide nädalasisese kasumi sõltuvus busside arvust

Nädalavahetuste sõidugraafikute koostamisel saavutas ahne algoritm samuti kasutuses olevast graafikust pisut parema tulemuse, teenides kahjumit 380.79 eurot kasutades selleks 2 erinevat füüsilist bussi.

Optimeeriva algoritmi tulemused olid märksa paremad, kuna mitmed tühjad sõidud asendati väiksemate arvude sõitudega, mistõttu oli kahjum märksa väiksem. Graafikult on näha, et optimaalsem on kasutada 3 või 5 füüsilist bussi, mis teeniksid kuus nädalasiseselt kasumit vastavalt 6520.65 ja 6537.18 eurot.

Joonisel 15 on kujutatud optimeeritult genereeritud 5 erinevat füüsilist bussi kasutatavat sõidugraafikut, mille kuukasum on kõrgeim.

LINE125A: 05:33-06:19    LINE125A: 06:34-07:20  
LINE135: 05:59-07:02    LINE135: 07:17-08:20  
LINE125A: 06:24-07:10    LINE125A: 06:25-07:11  
LINE132: 06:51-07:35    LINE132: 06:55-07:39  
LINE132: 07:01-07:45    LINE132: 07:05-07:49  
LINE132: 07:11-07:55    LINE132: 07:15-07:59  
LINE132A: 07:38-08:35    LINE132A: 07:40-08:37  
LINE132: 07:46-08:30    LINE132: 07:50-08:34  
LINE132: 07:57-08:41    LINE132: 08:56-09:40  
LINE132: 08:26-09:10    LINE132: 08:30-09:14  
LINE132: 08:26-09:10    LINE132: 08:30-09:14  
LINE135: 08:39-09:42    LINE135: 09:57-11:00  
LINE135: 09:28-10:31    LINE135: 10:46-11:49  
LINE135: 10:01-11:04    LINE135: 11:19-12:22  
LINE132: 10:52-11:36    LINE132: 11:51-12:35  
LINE125A: 10:59-11:45    LINE125A: 11:00-11:46  
LINE135: 11:19-12:22    LINE135: 12:37-13:40  
LINE132: 11:41-12:25    LINE132: 11:45-12:29  
LINE135: 12:14-13:17    LINE135: 13:32-14:35  
LINE132: 12:16-13:00    LINE132: 12:20-13:04  
LINE135: 13:07-14:10    LINE135: 14:25-15:28  
LINE135: 14:09-15:12    LINE135: 15:27-16:30  
LINE135: 14:27-15:30    LINE135: 15:45-16:48  
LINE135: 15:54-16:57    LINE135: 17:12-18:15  
LINE135: 16:26-17:29    LINE135: 17:44-18:47  
LINE135: 17:26-18:29    LINE135: 18:44-19:47  
LINE135: 18:21-19:24    LINE135: 19:39-20:42  
LINE135: 18:52-19:55    LINE135: 20:10-21:13  
LINE135: 19:53-20:56    LINE135: 21:11-22:14  
LINE135: 20:27-21:30    LINE135: 21:45-22:48  
LINE135: 20:57-22:00    LINE135: 22:15-23:18  
LINE132: 21:27-22:11    LINE132: 22:26-23:10  
LINE132: 21:57-22:41    LINE132: 22:56-23:40  
LINE132: 22:02-22:46    LINE132: 23:01-23:45  
LINE120: 22:22-22:56    LINE120: 23:11-23:45

Joonis 15. Optimeeritud nädalavahetuse graafik kasutades 5 füüsilist bussi

Seega on ahne algoritmi tulemusena koostatud graafikul kogukahjum ümardatult 1650 eurot, mis on 150 eurot väiksem kui hetkel kasutuses oleval graafikul.

Antud arvutusmeetodid tõestavad, et lokaalse otsingu põhimõtetel on võimalik koostada kasumlikum sõiduplaan nii teenusepakkuja kui -kasutajate jaoks.

Tulemus üle reaalseks üle kontrollimiseks oleks vaja rohkem reaalelulisi andmeid, kuna kogutud algandmed ei pruugi kajastada kogu kasutajaskonda. Lisaks tasub uurida kindlaks teha, kui palju muutuks lõpptulemus optimeerimisel kasutades teisi bussiliine.

## 4 Kokkuvõte

Maakonnaliinid ei allu ühtlasele reisijate paiknemisjaotusele bussipeatustes, mis on üheks oluliseks eelduseks juba olemasolevatel programmidel. Antud töös keskendutakse Harju maakonnaliinide Tallinn-Jüri suunale ning lahendati sõidugraafiku genereerimise probleemi lokaalse otsingu põhimõttel.

Tööle püstitati 2 hüpoteesi, mis mõlemad leidsid kinnitust.

Koostatud optimeeriv algoritm oli võimeline koostama optimaalsema algoritmi, kui töö kirjutamise ajal kasutuses olnud sõiduplaan. Kasutuses olev sõiduplaan antud töös kogutud lähteandmete puhul tõestas, et ei ole kasumlik, samas kui optimeeritud algoritmiga on mitmeid erinevaid valikuid, kui suurte busside arvudega on võimalik kasumit teenida.

Ka teine hüpotees osutus tõeseks, sest maksimaalse soovitud reisidega tõuseb kulu ja tulu suhe. Põhjuseks on busside vähene täituvus ja liisingumaksud.

Edaspidi tasuks uurida, kas on võimalik ligi pääseda reaalsematele andmetele, kuna antud töö käigus koguti andmed küsitlusega, mille tulemused võivad varieeruda reaalsest elust. Antud töö käigus tuli teha ka mitmeid kitsendusi, mistõttu oleks tulevikus mõistlik laiendada neid piiranguid, et leida veelgi täpsemaid tulemusi. Mõningad laiendatavad kitsendused on näiteks ajalised piirangud. Antud töö käigus prooviti 2 soovitud reisi asemele 1 mahukamat koostada, siis võiks ka vaadata, kas mõnel täiesti teisel ajal oleks reisi omamine kasumlikum ning seetõttu jätta ära eelneval ajal üldse sõidud. Lisaks võiks ka optimeeritud graafikut üle jooksutada korduvalt, et võrrelda saadud tulemust omakorda optimeerides saadava tulemusega.

## Kasutatud kirjandus

[1] – T. Pott. „Maakonnaliinide bussid kipuvad tasuta sõitjate rohkusel kitsaks jääma”. 2018. Loetud aadressil: <https://www.err.ee/861835/maakonnaliinide-bussid-kipuvad-tasuta-soitjate-rohkusel-kitsaks-jaama> 17.03.2021

[2] – Maanteeamet. „Maakondliku bussiliiniga sõitnute rahulolu uuring”. 2020. Loetud aadressil: <https://www.transpordiamet.ee/media/7719/download> 02.04.2023

[3] – Mõnus Minek OÜ. „Ühistranspordis kasutatavate diisel, (bio)gaasi-, gaasihübriid-, diiselhübriid- ja elektribusside tasuvusuuring“. 2014 Loetud aadressil: [https://www.tartu.ee/sites/default/files/5462\\_MBBB\\_4\\_bussi\\_vordlev\\_analyys\\_2014.pdf](https://www.tartu.ee/sites/default/files/5462_MBBB_4_bussi_vordlev_analyys_2014.pdf) 10.04.2023

[4] – NEC Group. Schedule Optimization System. 2021. Loetud aadressil: <https://www.nec.com/en/global/solutions/transportation/task/sos.html#:~:text=What%20is%20Schedule%20Optimization%20System,buses%20arrived%20at%20regular%20intervals> 17.03.2021

[5] – Qun Chen. „Global Optimization for Bus Line Timetable Setting Problem“. 2014. Loetud aadressil: <https://www.hindawi.com/journals/ddns/2014/636937/> 17.03.2021

[6] – X. Yang, X. Li, Jun. Liang, C. Peng. „Bus Scheduling Optimization Based on Improved Strength Pareto“. 2015. Loetud aadressil: [https://link.springer.com/chapter/10.2991%2F978-94-6239-102-4\\_64](https://link.springer.com/chapter/10.2991%2F978-94-6239-102-4_64) 18.03.2021

[7] – F.D. Wihartiko, A. Buono, B. P. Silalahi „Integer programming model for optimizing bus timetable using genetic algorithm“. 2017. Loetud aadressil: <https://doi.org/10.1088/1757-899X/166/1/012016> 18.03.2021

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Aaron Avaste

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Maakonnaliini bussiliikluse optimeerimine Tallinn-Jüri suuna näitel lokaalse otsingu põhimõttel“, mille juhendaja on Marko Kääramees
  - 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.

22.05.2023

---

<sup>1</sup> 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.