

TALLINNA TEHNIKAÜLIKOOL

Majandusteaduskond

Ärikorralduse instituut

Priidik Vilumaa

**BRAUSERIPÕHISTE MARSRUUDIPLANEERIMISE
RAKENDUSTE VÕIMEKUSE TEST**

Bakalaureusetöö

Juhendaja: dotsent Tarvo Niine

Tallinn 2017

SISUKORD

ABSTRAKT	3
SISSEJUHATUS	4
KASUTATUD LÜHENDID	7
1. MARSRUUDIPLANEERIMINE	8
1.1. Marsruudiplaneerimise faktorid	8
1.2. Optimeerimismudelid ja heuristikad	9
1.3. Rakendused	14
1.4. Kokkuvõte.....	16
2. METOODIKA	17
2.1. Rakenduste valim	17
2.1.1. GebWeb Optimap	18
2.1.2. GSMtasks	19
2.1.3. Onfleet	20
2.1.4. OptimoRoute	20
2.1.5. RouteXL.....	21
2.1.6. Routific	21
2.1.7. Võrdlustabel	22
2.2. Ettevõtete ja lähteülesannete kirjeldus.....	22
2.2.1. DPD Eesti AS	23
2.2.2. Eesti Keskkonnateenused AS	24
2.3. Kokkuvõte.....	25
3. MARSRUUDIPLANEERIMISE RAKENDUSTE TEST	26
3.1. Rakendustega lähteülesannete lahendamine ja kasutajakogemus	26
3.1.1. Lähteülesanne nr 1 - Halliste korjering	26
3.1.2. Lähteülesanne nr 2 - Lasnamäe jaotusring	30
3.1.3. Lähteülesanne nr 3 - Peetrimõisa korjering.....	31
3.2. Tarkvara testimise tulemused.....	34
3.2.1. Lähteülesannete tulemused	34
3.2.2 Ettevõtete tagasiside tulemustele.....	36
3.3. Järeldused ja arutelu	38
KOKKUVÕTE	40
SUMMARY	42
KASUTATUD ALLIKAD	44
LISAD	46
Lisa 1. Visual Basic makro.....	46

ABSTRAKT

Käesoleva bakalaureusetöö pealkiri on „Brauseripõhiste marsruudiplaneerimise rakenduste võimekuse test“.

Uurimisprobleemiks on puudulik ülevaade tänapäevaste marsruudiplaneerimise rakenduste võimekuse ning nende piirangute kohta Eesti jaotusettevõtete vastavasisuliste tüüpsituatsioonide kontekstis. Töö kirjutamise ajendiks oli soov orienteeruda marsruudiplaneerimise rakenduste kogumis ning õppida neid kasutama. Töös testis autor kahelt ettevõttelt saadud kolme marsruudikavandamise lähteülesanne lahendamist kuue erineva rakendusega. Töö eesmärk oli testida tänapäevaste nii vabavaraliste kui tasuliste marsruudiplaneerimise tarkvarade võimekust ning rakendatavust Eesti jaotusettevõtete praktilistes marsruudiplaneerimise olukordades.

Tulemusena selgus, et mitte ükski valimisse kuulunud marsruudiplaneerimise rakendustest ei suuda anda tiheasustuse korjeülesannete jaoks praktiliselt kasutatavaid lahendeid. Veel leidis autor, et rakenduste poolt genereeritud lahendi kvaliteet ei pruugi olla tarkvara valimisel esmane faktor, vaid loeb ka andmetega töötamise kiirus ning lihtsus. Lahendeid omavahel võrreldes ilmnes ka, et valimisse kuulunud rakendused andsid samadele ülesannetele erinevaid lahendeid, mis annab aimdust, et enamik lahendeid ei ole optimaalsed. Autor soovib tarkvara arendajatel panna rohkem rõhku andmetöötlusprotsesside lihtsustamisele ja algoritmide täiustamisele, et ka tiheasustuse korjeülesanne oleks lahend oleks praktiliselt kasutatav. Kasutajad peaksid kindlasti enne rakenduste hankimist veenduma, et just nende lähteülesanded lahendatakse tarkvara poolt edukalt, kiirelt ja mugavalt.

Võtmesõnad: marsruudiplaneerimise rakendused, transpordi optimeerimine, rakenduste testimine, rändkaupmehe ülesanne, veoringluse ülesanne, bakalaureusetööd.

SISSEJUHATUS

Internetipõhiste logistika tugiteenuste pakkumine on tänu interneti laialdasele levikule lihtsam ja levinum kui kunagi varem. Marsruudi optimeerimise tarkvaralisi tööriistu on maailmas palju ning mitmed on ettevõtetele ka suhteliselt soodsalt kättesaadavad. Pole aga selge, kui käepärased sellised rakendused on ning kas nende võimekus sobib Eesti-siseses kontekstis keeruka iseloomuga jaotusprobleemide optimeerimiseks. Ühtlasi puudub ülevaade, millised on ettevõtete nõuded marsruudiplaneerimise rakendustele ning kas pakkumine ja nõudlus ühilduvad. Marsruudi optimeerimise rakenduste võimekuse uurimine võimaldab tõhustada ettevõtete tööd olukorras, kus tootmine kallineb ja püütakse leida lahendusi kulude vähendamiseks.

Töö eesmärgiks on testida tänapäevaste nii vabavaraliste kui tasuliste marsruudiplaneerimise tarkvarade võimekust ning rakendatavust Eesti jaotusettevõtete praktilistes marsruudiplaneerimise olukordades.

- Uurimisküsimus 1: Millised on Eesti jaotusettevõtete marsruutide kavandamise tüüpsituatsioonid?
- Uurimisküsimus 2: Milline on brauseripõhiste marsruudiplaneerimise tarkvarade funktsionaalsus?
- Uurimisküsimus 3: Kuivõrd sobivad valimisse kuuluvad marsruudiplaneerimise rakendused eelnevalt püstitatud praktiliste transpordikavandamise ülesannete lahendamiseks?

Töö püüab anda vastuse tänapäevaste marsruudiplaneerimise rakenduste võimekuse ning nende piirangute kohta Eesti jaotusettevõtete vastavasisuliste tüüpsituatsioonide kontekstis.

Tartu Ülikoolis 2005. aastal kaitses Jaanus Jaeger oma magistritöö „Aja-akendega transpordiülesande lahendamine“, kus ta kasutas kahte marsruudioptimeerimise programmi, et lahendada fiktiivselt püstitatud ülesanne. (Jaeger, 2005) Kuna sellest on möödas 12 aastat, siis võib eeldada, et marsruudioptimeerimise rakendustes on vahepeal väga palju muutunud. J.

Jaeger kasutas eestikeelset mõistet transpordiülesanne (inglise keeles *Vehicle Routing Problem*, lühendatult *VRP*). Käesoleva töö autorit ajas see asjaolu esialgu segadusse, kuna inglisekeelne *Transportation Problem (TPP)*, mida samuti võiks tõlkida transpordiülesanne, on seotud hoopis kauba ladudest poodidesse toimetamise jaotuskulude minimeerimisega. Seega nimetatakse siinses töös *VRP*-d veoringluse ülesandeks, sest see tõlge on selgemalt *TPP*-st eristatav.

ORMS Today avaldab iga kahe aasta järel tehtud marsruudiplaneerimise rakenduste uuringu, mis on läbi viidud küsitluse vormis. Tarkvarafirmadelt küsitakse muuhulgas nende toote parameetreid, näidisülesande lahendamise kiirust ning arvamust valdkonna tuleviku kohta. Kõik sealsed esitatud andmed on aga küsitleja poolt valideerimata ning uuringu autorid ise vastavaid rakendusi kasutanud ei ole. Seetõttu ei sisalda need ülevaatlikku hinnangut ei rakenduse piiridest ega kasutusmugavusest. (ORMS Today, 2017)

Uurimisülesannete lahendamiseks viib autor käesolevas töös läbi mitmikjuhtumi analüüsi. Autori meetodika koosneb kahest faasist, millest esimene loob tausta juhtumianalüüsile ning teine tegeleb vahetu rakenduste funktsionaalsuse testimisega.

Esimeses faasis intervjuuerib autor kaht ettevõtet, kaardistades sellega taustsüsteemi, milles on vaja lahendada teatud marsruudikavandamise ülesanded, ning genereerides intervjuude baasil n -ö tüüpsituatsioonid (mida antud töös on kolm), mis peegeldavad hästi firmade optimeerimisprobleemide tegelikku keerukust ja toimivad edasises uuringus testi aluseks olevate lähteülesannetena. Valimi määratlemisel keskenduti jaotussüsteemidele, kus on kohalik mõõde, aga mitmeid kliente. Valimisse kuulus kaks ettevõtet, millest üks oli suurem rahvusvaheline, teine väiksem ning ainult Eestis tegutsev. Autor intervjueris mõlemas ettevõttes logistika optimeerimisega seotud inimesi. Nõnda saadi teada ettevõtetes ilmnevad näidisprobleemid ja kolm ülesannet, mida tarkvara abil lahendada.

Teises faasis kombineerib autor valimi sarnase kättesaadavusega ja reklaami järgi võrreldava võimekusega marsruudioptimeerimise rakendustest (mida antud töös on kuus) ning rakendab neid ükshaaval kõigi püstitatud lähteülesannete lahendamiseks.

Töös kasutatud tarkvara jaguneb kahte suuremasse rühma - veokipargi korraldamise tarkvara marsruudioptimeerimise funktsionaalsega ja puhas marsruudioptimeerimise tarkvara. Enamus töös kasutatud tarkvaradest on tasulised ning töö autor testis neid ajutise demoversioonina. Tarkvara testide tulemused esitati ülesanded püstitanud ettevõtetele

tagasisidestamiseks. Seega lõplik hinnang brauseripõhiste marsruudiplaneerimise tarkvarade võimekusele põhineb nii autori kui ettevõtete kombineeritud arvamusel.

Esimeses peatükis käsitletakse marsruudi planeerimise teooriat. Esmalt kirjeldatakse selle faktoreid – näiteks peatuste arv või lisakitsendused marsruudil. Seejärel antakse ülevaade matemaatilistest optimeerimismudelitest, kasutatavatest heuristilistest algoritmidest ja marsruudi planeerimise tarkvarade teoreetilisest käsitlusest. Teises peatükis kirjeldatakse lähemalt töös kasutatud metoodikat. Antakse ülevaade mõlemast ettevõttest, ilmnenuid tüüpsituatsioonidest ja konkreetsetest lähteülesannetest. Esitatakse kuue rakenduse üldine kirjeldus. Kolmandas peatükis tuuakse välja kõikide ülesannete lahenduskäik kuue tarkvara abil. Seejärel esitatakse tehtud testide tulemused, ettevõtete tagasiside ja järeldused.

KASUTATUD LÜHENDID

CPU – *central processing unit*, keskprotsessor

CVRP – *capacitated vehicle routing problem*, mahupiirangutega veoringluse ülesanne

mTSP – *multiple Traveling Salesman Problem*, mitme rändkaupmehe ülesanne

SaaS – *Software as a Service*, tarkvara kui teenus

TSP – *Traveling Salesman Problem*, rändkaupmehe ülesanne

VRP – *Vehicle Routing Problem*, veoringluse ülesanne

VRPTW – *Vehicle Routing Problem with Time Windows*, ajaakendega veoringluse ülesanne

1. MARSRUUDIPLANEERIMINE

Töö esimene peatükk on pühendatud uuritava teemaga seotud teooriale. Esimeses alapeatükis kirjeldatakse marsruudiplaneerimist ning sellega seonduvaid faktoreid nagu piirangud ja klasterdamine. Teises alapeatükis tuuakse välja nii TSP kui ka VRP matemaatilised optimeerimismudelid, vajalikud sisendid ja ülevaade heuristikatest. Kolmas alapeatükk kirjeldab välja toodud ülesannete lahendamise seotud rakendusi ja nende karakteristikaid.

1.1. Marsruudiplaneerimise faktorid

Marsruudiplaneerimine on kõige kuluefektiivsema mitme punkti või peatusega teekonna arvutamine, minimeerides läbitud teepikkust ja/või ajakulu (Business dictionary, 2017). Optimaalse marsruudi arvutamine transpordivõrgus kindla lähte- ning sihtkoha vahel on üks algoritmide päris elu rakendus. Seda funktsionaalsust kasutatakse autosõidu või ühistranspordiga liikumise planeerimisel väga tihti. (Sanders, Schultes, 2007, 2) Marsruudiplaneerimine võib sisaldada ka peatuste järjestuse valimist. Kuna suurem osa liikumistest on vahepeatusteta ehk punktist A punkti B, siis koosnebki marsruudiplaneerimine ainult parima tee valimisest. Kui autojuhil on vajalik teha ka vahepeatusi - näiteks teeninduskohad, postkastid või prügikonteinerid -, siis muutub marsruudiplaneerimine tunduvalt keerulisemaks. Enam ei loe mitte ainult järgmisesse punkti sõitmise pikkus, vaid ka läbitavate punktide järjestus. Sellest järeldub, et valdkonnades, kus transportimisel on peatusi rohkem kui alg- ja lõpp punkt, on marsruudiplaneerimisele vaja panna tunduvalt rohkem rõhku. Sellised valdkonnad on näiteks kullerteenused, prügi kogumine ja kirjavposti jaotus.

Kuigi lihtsustatud näidetes on vaja lihtsalt külastada kõiki peatusi, siis praktikas on tegelikult piiranguid rohkem. Näiteks kui peatus on seotud mingi ettevõtte või isikuga, siis võivad kehtida ajapiirangud, millal ettevõttel on tööaeg või inimene kodus. Ühtlasi on peatus tihti seotud füüsiliste kaupade korje või jaotusega ning seetõttu kehtivad ka mahupiirangud.

Nende kitsenduste lisamine probleemile muudab juba väikese peatuste arvu juures ülesandele teostatava lahendi leidmise võrdlemisi keerukaks.

Peatuste väikese arvu korral (kuni 20), saab osav dispetšer marsruutide planeerimisega lihtsalt peatustega tähistatud kaarti vaadates hakkama. Kui aga mahu kitsendus on piirav, siis vajab planeerimine palju suuremat süvenemist. (Desaulniers *et al*, 2014, 120) Seega on keerulisemate olukordade puhuks vaja tööriista, mis suudab genereerida eelkõige võimalikke ja lisaks kõrge kvaliteediga (heuristilisi) või optimaalseid lahendeid. Lahendi kvaliteedi defineerib mingi funktsioon, mille üheks piiriks on optimum ning teiseks piiriks kõige halvem lahend. Optimaalsed lahendid leitakse matemaatilise programmeerimise mudeli abiga ja muud „head“ lahendid erinevate heuristikatega.

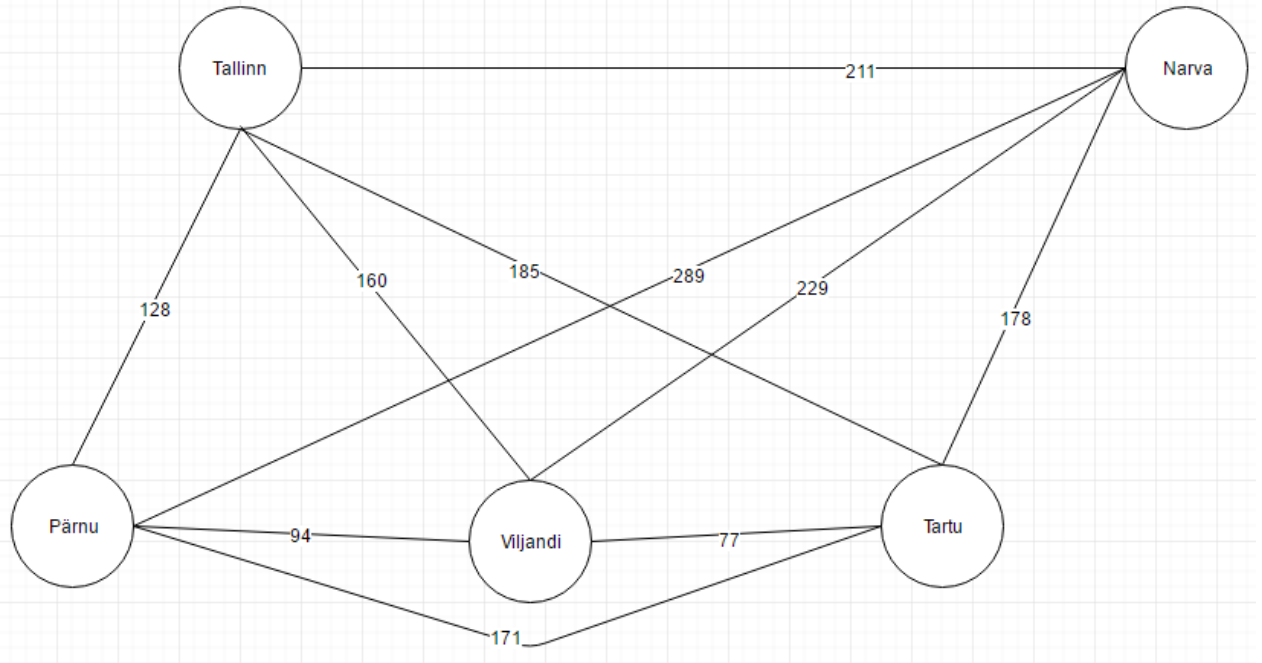
Võrreldes manuaalse marsruudiplaneerimisega on optimeerimisel põhinev tehnika kulusäästlik ning masinaparki saab efektiivsemalt kasutada. Lisaks võimaldab selliste tööriistade kasutamine ettevõtte automatiseerimist, standardiseerimist ja integreerimist teistesse planeerimisprotsessidesse. (Irnich *et al*, 2014, 2)

Reaalses maailmas on kliendid enamasti jaotatud geograafiliselt ebaühtlaselt. Suurem hulk kliente on koondatud linnadesse, seevastu maapiirkondades leidub neid suhteliselt hõredalt. Läbitavaid marsruute arvestades on üsna mõistlik, et ühte piirkonda kuuluvad kliendid teenindatakse ära ühekorraga ega toimu liikumist erinevate linnade vahel edasi-tagasi. Nimelt, kui lühemate marsruutide korral teenindatakse lähestikku asetsevad kliendid ära järjest, võib kõiki vastava piirkonna kliente esindada üheainsa teenindamist vajava metakliendi ehk klastrina, kaotamata seejuures palju lahendi headuses. Peale klastreid sisaldava ülesande lahendamist tuleb veel vaid välja pakkuda konkreetsed läbimisjärjekorrad klatri klientide jaoks. Sellise lähenemise korral on korraga lahendatav ülesanne algandmete hulga mõttes tunduvalt väiksem ja seega lihtsam lahendada. (Jaeger, 2005, 4)

1.2. Optimeerimismudelid ja heuristikad

Marsruudiplaneerimises kasutatakse erinevaid matemaatilise programmeerimise mudeleid, mis visualiseerivad lähteülesandeid. Enim kasutatust leidnud taoline mudel on rändkaupmehe ülesanne (*Travelling Salesman Problem*). Selle eesmärk on minimeerida läbitud teekonna pikkust nii, et rändkaupmees läbib kõiki linnu ning naaseb lähtekohta. See on kombinatoorne optimeerimisprobleem, mida saab väljendada järgnevalt: „Teades n -i

liikmega tippude hulka ning kaugust iga paari tippude vahel leida lühim teekond nii, et kõiki tippe läbitakse ühe korra ning lõpuks naastakse algpunkti. (Anbuudayasankar *et al*, 2014, 2) Rändkaupmehe ülesannet kujutatakse täisgraafil (*complete graph*), mis koosneb tippudest ja servadest. Täisgraaf mõnede Eesti linnadega on kujutatud joonisel 1.



Joonis 1. Kaalutud servadega täisgraaf.

Allikas: koostatud autori poolt.

Matemaatiliselt formuleeritakse rändkaupmehe ülesannet järgnevalt (Minner, 2016):

$$\min C = \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij} \quad (1)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j = 1, 2, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1, 2, \dots, n \quad (3)$$

$$z_i - z_j + n \cdot x_{ij} \leq n - 1 \quad \forall i, j = 2 \dots n; i \neq j \quad (4)$$

$$x_{ij} \in \{0,1\}, i = 1 \dots n, j = 1 \dots n$$

- $i, j \in N$ ning hulk $N = \{1, 2, \dots, n\}$
- Otsustusmuutuja $x_{ij} \begin{cases} = 1, \text{ kui tee } i \text{ ja } j \text{ vahel on validud} \\ = 0, \text{ muudel juhtudel} \end{cases}$
- Parameeter c_{ij} – kaugus tipu i ja j vahel
- Sihifunktsiooni (1) eesmärgiks on minimeerida otsustusmuutuja x_{ij} ja parameetri c_{ij} korrutiste summat.
- Kitsendus (2) - igasse tipu saabub ainult üks tee
- Kitsendus (3) - igast tipust lahkub ainult üks tee
- Kitsendus (4) – Miller – Tucker – Zemlin lühiringide eemaldamise kitsendus, mis tagab, et depoost depooni on pidev tee, mis läbib kõiki tippe
- Kitsendus (5) – binaarkitsendus ning indeksite vahemik

Rändkaupmehe ülesande vajalikuks sisendiks on distantsimaatriks, ehk kõikide tippude vahelisi kaugusi. Kuna lõppkasutaja selliseid kauguseid ei tea, siis genereeritakse need tarkvara poolt tippude geograafiliste koordinaatide või eelkõige aadresside järgi. Kaugus võib olla kas linnulennult (eukleidiline), sõidukaugus või isegi sõidu aeg.

Kui kõik andmed on enne töö alustamist teada, siis saab luua staatilised plaanid (Goel, 2008, 5). Vastandina staatilisele planeerimisele esineb ka dünaamiline. See võimaldab näiteks peatusi tööpäeva jooksul juurde lisada ning selle rakenduseks on vaja planeerijate ja autojuhtide vahel andmeühendust. *Online* marsruudiplaneerimine tegeleb seega pideva teekonna kavandamisega, kus saabuvad klienditellimused paigutatakse efektiivselt marsruudiplaani. (Bieding *et al*, 2009, 29)

Teiseks võib välja tuua mitme rändkaupmehe ülesande (*multiple Traveling Salesman Problem, mTSP*) on tavalise TSP laiendus, kus m kaupmeest peavad läbima n tippu nii, et kõiki tippe läbib üks kaupmees (Anbuodayasankar *et al*, 2014, 2). Veoringluse ülesanne (*Vehicle Routing Problem, VRP*) on omakorda mTSP laiendus, kuhu lisatakse sõidukite kandekoormused ning nõudlus igas tipus. (Anbuodayasankar *et al*, 2014, 3) Aja akendega VRP-s (*VRPTW*) tuleb iga asukohta külastada kindlal ajavahemikul ning sõiduk peab teenindusaja vältel paikseks jääma. Aja aknaid on võimalik modelleerida kas „kõvaks“ ehk paindumatuks või „pehmeks“, kus trahvi eest on võimalik aja piiranguid rikkuda. (Cordeau *et al*, 2000, 1)

VRP mudeleid saab kasutada erinevatel aladel, näiteks kaupade liikumisel tarneahelas, kullerteenustel, ühistranspordis, prügi kogumisel jne. (Anbuodayasankar *et al*, 2014, 3)

VRPTW on üsna lähedane praktilistele probleemidele. Matemaatiline mudel, mis arvestab enamuste praktiliste aspektidega vähendab oluliselt manuaalset kontrolli, mis tavaliselt järgneb automaatselt plaani genereerimisele ning selle kohandamisele (Goel, 2008, 96).

Üks võimalik VRPTW matemaatiline kuju (Minner, 2016):

$$\min Z = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K c_{ij} \cdot x_{ijk} \quad (6)$$

$$\sum_{i=0}^n x_{ij} = y_{jk} \quad \forall j = 1, 2, \dots, n; k = 1, 2, \dots, K \quad (7)$$

$$\sum_{j=0}^n x_{ij} = y_{ik} \quad \forall i = 1, 2, \dots, n; k = 1, 2, \dots, K \quad (8)$$

$$z_i - z_j + n \cdot \sum_{k=1}^K x_{ijk} \leq n - 1 \quad \forall i, j = 2 \dots n; i \neq j \quad (9)$$

$$\sum_{k=1}^K y_{ik} = 1 \quad \forall i = 1, 2, \dots, n \quad (10)$$

$$\sum_{k=1}^K y_{0k} = K \quad (11)$$

$$\sum_{i=1}^n y_{ik} \leq Q \quad \forall k = 1, 2, \dots, K \quad (12)$$

$$\sum_{i=0}^n \sum_{j=0}^n c_{ij} \cdot x_{ijk} \leq T \quad \forall k = 1, 2, \dots, K \quad (13)$$

$$s_{ik} + t_{ij} - L_{ij}(1 - x_{ijk}) \leq s_{jk} \quad \forall i = 1, 2, \dots, n; k = 1, 2, \dots, K \quad (14)$$

$$a_i \leq s_{ik} \leq b_i \quad \forall i = 1, 2, \dots, n; k = 1, 2, \dots, K \quad (15)$$

$$x_{ij} \in \{0, 1\}, y_{ik} \in \{0, 1\}, i = 1 \dots n, j = 1 \dots n, k = 1 \dots n \quad (16)$$

- $i, j \in N, k \in Kh$ ning hulk $N = \{1, 2, \dots, n\}$ ja $Kh = \{1, 2, \dots, K\}$

- Otsustusmuutuja $x_{ijk} \begin{cases} = 1, \text{ kui marsruut } k \text{ jaoks on tee } i \text{ ja } j \text{ vahel on valitud} \\ = 0, \text{ muudel juhtudel} \end{cases}$
- Otsustusmuutuja $y_{ik} \begin{cases} = 1, \text{ kui asukoht } i \text{ kuulub marsruuti } k \\ = 0, \text{ muudel juhtudel} \end{cases}$
- Parameetrid c_{ij} – kaugus asukoha i ja j vahel; z_i – järjenumber, mitmes asukoht i marsruudil on; Q maksimaalne stoppide arv; T maksimaalne läbisõidu ühikute arv
- Sihifunktsiooni (6) eesmärgiks on minimeerida otsustusmuutuja x_{ijk} ja parameetri c_{ij} korrutiste summat.
- Kitsendus (7) - igasse sihtkohta saabub $\begin{cases} 1 \text{ tee, kui asukoht } i \text{ kuulub marsruuti } k \\ 0, \text{ muudel juhtudel} \end{cases}$, ei kehti depoo kohta
- Kitsendus (8) - igast asukohast lahkub $\begin{cases} 1 \text{ tee, kui asukoht } i \text{ kuulub marsruuti } k \\ 0, \text{ muudel juhtudel} \end{cases}$, ei kehti depoo kohta
- Kitsendus (9) – lühiringide piirang, takistab lühiringide tekkimise
- Kitsendus (10) – iga asukoht i kuulub ainult ühte marsruuti k
- Kitsendus (11) – depoo peab lahkuma k teed
- Kitsendus (12) – mahupiirang, marsruuti võib kuuluda maksimaalselt Q asukohta
- Kitsendus (13) – mahupiirang, marsruudi summaarne pikkus võib olla maksimaalselt T
- Kitsendus (14) sunnib kõigepealt teenindama tippu i ja seejärel j ning kitsendus (15) sunnib tippu teenindama peale avamist ning enne sulgemist.
- Kitsendus (16) – binaarkitsendused ning indeksite piirid

VRPTW vajalikuks sisendiks on lisaks distantsimaatriksile aja akende piirid, teeninduse kestvus, sõiduaeg i -st j -ni ning vajadusel mahupiirangute parameetrid näiteks maksimaalne peatuste (tippude) arv ja maksimaalne tööpäeva kestvus või läbisõit.

A. Goel väidab (Goel, 2008), et klassikalised, nagu eelmainitud mudelid lihtsustavad päris elu liialt ning vastavas kirjanduses pannakse rohkem rõhku klassikalistele mudelitele heade lahenduste leidmisele selle asemel, et arendada mudeleid, mis suudavad päris elus ilmneva kitsendusi arvesse võtta. Irnich, Toth ja Vigo (Irnich *et al*, 2014, 1) väidavad aga, et optimeerimine on väga edukas ning selle üheks põhjuseks on heade mudelite arendus, mis võimaldab mudelitel peaaegu kõigi faktoritega arvestada. Need vastanduvad väited tekitasid

töö autoril küsimuse - milline on tänapäevaste brauseripõhiste marsruudiplaneerimise rakenduste võimekus.

Lisaks eelmainitud täisarvulise lineaarse programmeerimise mudelitele kasutatakse rändkaupmehe ning sõidukite juhtimise ülesande lahendamiseks ka heuristikaid. Heuristikad sisaldavad meetodit, kuidas õppida, avastada või lahendada ülesandeid ekperimentaalselt ja eriti katse-eksitus-meetodil (Merriam-Webster, 2017). Need on lihtsad protsessid või algoritmid lahendi arvutamiseks, kuid ei taga optimaalset lahendit ega anna aimdust lahendi kvaliteedist. Rändkaupmehe ülesande jaoks kasutatavad heuristikad saab jagada printsiibi järgi mitmesse rühma: näiteks teostatava lahendi leidmise või lahendi parendamise heuristikad. Esimeste hulka kuuluvad lähima naabri heuristika säästmise meetod ning paljud muud. Mõned lahendi parendamiseks loodud heuristikad on 2 – opt meetod, simuleeritud lõõmutamine ja geneetiline algoritm. Protsessis võetakse saadud lahend, muudetakse seda süstemaatiliselt ja korduvalt mingi reegli järgi ja loodetakse saada parem tulemus. Näiteks geneetilise algoritmi kasutamisel luuakse mõnest parimast lahendist kombinatsioon, mis võib olla veelgi parem. Kuna praktikas tekkivad olukorrad võivad olla mahukad ning vajavad kiiret lahendit, siis võib ka praktikas efektiivseid heuristilisi algoritme tarvis minna (Laporte *et al*, 2014, 87).

1.3. Rakendused

TSP ja VRP formuleerimises ning lahendamises peetakse mudeli keerukust väga oluliseks faktoriks. Juba mõõduka suurusega probleemi lahendamine nõuab arvutilt väga palju ressursi ning vahel on optimaalse lahendi otsimine isegi tänapäevaste arvutitega mitteotstarbekas. (Anbuudayasankar *et al*, 2014, 9)

Kuigi oleks soovitav, et VRP lahendataks polünoomselt piiratud kestvusega algoritmi poolt, siis üldiselt kuulub VRP ülesannete klassi NP ehk ülesanded, mida ei pruugi saada polünoomse tööajaga lahendada. On tõestatud, et TSP on „NP-complete“ ehk ülesanne, mille lahendi korrektsust saab kontrollida polünoomse tööajaga. VRP, mis on TSP laiendus, on „NP-hard“ ehk ülesanne, mille lahendi õigsust ei pruugi saada kontrollida polünoomse tööajaga. Üldjuhtudel on TSP ja VRP lahendamiseks vaja heuristilisi algoritme, et leida optimumi lähedane lahend mõistliku ajaga. (Anbuudayasankar *et al*, 2014, 10)

Praktikas on tavaliselt külastatavate asukohtade arv väike, sest planeerimine toimub päeva kaupa ning tööpäeva piiratud pikkus ei võimalda väga palju tippe läbida. Umbes 100 tipuga ülesanded saab optimaalsuseni lahendada, kuid arvutuse kestvus võib suuresti kõikuda (Laporte *et al*, 2014, 87). Optimeerimistehnikate edu ning kasutus ei ole tekkinud ainult tänu tänapäevaste arvutisüsteemide võimsusele ja täielikult ettevõtte protsessidesse integreeritud infosüsteemidele. Selle põhjuseks võib lugeda ka rangete matemaatiliste mudelite arenduse, mis on võimalised arvesse võtma peaaegu kõiki VRP faktoreid, mis reaalses elus võivad tekkida. Vastavad algoritmid ning nende tarkvaralised lahendused mängivad kõrge kvaliteediga sobivate lahendite mõistliku ajaga leidmisel väga suur rolli. (Irnich *et al*, 2014, 1)

Viimaste aastate jooksul on arendatud telemaatikateenuseid (elektroonilist andmevahetust sõidukite ning planeerijate vahel) sisaldavat tarkvara. See võimaldab dünaamilise transpordisüsteemi kavandajatel kiiremat reageerimist võimalikele sõiduki riknemise või raskete liiklusolude tõttu tekkinud vigadele. (Irnich *et al*, 2014, 2) Transpordis on üks tähtsamaid rakendusi punktist A punkti B väikseimate kaasnevate kuludega teekonna leidmine. Alternatiivselt võib vaja olla ka lühima kilometraažiga, kiireima läbisõiduajaga või muu marsruudi leidmine. (Goel, 2008, 23) Laiendusena eelmainitule, võib ka välja tuua, et kui peatusi on rohkem kui alguses ja lõpus, siis ilmneb ka vajadus arvutada parim peatuste järjestus, et punktist A punkti B sõit oleks odavam. Arvutiga teostatud planeerimine võimaldab töötajatel mitmeid erinevaid potentsiaalseid situatsioone võrrelda ning seeläbi valida kulusid ning teenindusnäitajaid arvestades parim variant (Irnich *et al*, 2014, 2). Marsruudi optimeerimise tarkvara eesmärk on leida kas ühele või mitmele sõidukile optimeeritud teekond. Marsruudi optimeerimise funktsionaalsuse võib leida kahte sorti rakendustest: puhtast marsruudiplaneerimise tarkvarast (*route optimization software*) ja veokipargi korraldamise tarkvarast (*fleet management software*). Sellised programmid on olemas alla laadimiseks ning nii kohalikus arvutis installeerimiseks kui ka veebibrauseris kasutamiseks. Brauseris kasutatavate programmide suureks eeliseks on pilvandmetöötlus. Kuna optimeerimisülesanded nõuavad arvutilt väga palju CPU ning vahemälu ressursi, siis on pilvandmetöötlus efektiivne lahendus (An, Lee 2015, 34).

Pilvandmetöötlus on mudel, mis võimaldab asukohast sõltumatult ja mugavalt võrgu kaudu tark- või riistvarale (näiteks võrgud, serverid, andmebaasid, rakendused ja teenused) ligipääsu. Ligipääsu loomine toimub kiirelt ja minimaalse haldamise ja teenusepakkujapoolse

tegevusega. Selline pilvemudel koosneb viiest fundamentaalsest karakteristikust, kolmest teenusemudelitest ja neljast paigaldusmudelitest. (Mell, Grance, 2011, 2)

Marsruudiplaneerimise tarkvara pakutakse enamasti teenusena. (*Software as a Service, SaaS*). National Institute of Standards and Technology defineerib SaaS-i ja ütleb, et kliendile pakutakse võimalust kasutada teenusepakkuja rakendusi, mis jooksevad pilvandmetöötlusega seotud infrastruktuuris. Rakendused on ligipääsetavad kliendi erinevatest seadmetest läbi õhukese kliendiliidese nagu veebibrauser. Tarbija ei halda ega kontrolli seotud infrastruktuuri, mille hulka kuuluvad võrk, serverid, operatsioonisüsteemid, andmebaasid või isegi rakenduste võimalused. Erandiks võivad olla minimaalselt konfigureeritavad kasutaja-põhised seaded. (Ibid.) Marsruudiplaneerimisega seotud rakendused on enamasti seotud paigaldusmudeliga „*Private Cloud*“. Infrastruktuur pakutakse mitmele tarbijale (näiteks äriühingud) ühe organisatsiooni poolt. Infrastruktuur võib kuuluda või olla hallatud selle organisatsiooni, kolmanda osapoole või mingisuguse kombinatsiooni poolt. (Ibid.)

1.4. Kokkuvõte

Marsruudiplaneerimise ja sellega seonduvate faktorite nagu piirangud ja klasterdamine uurimisega koondas käesolev peatükk ülevaadet teoreetilistest alustest järgnevatele lähteülesannete formuleerimisele ja nende lahendamiseks rakenduste paremini mõtestatud testimisele. Marsruudiplaneerimise teooria toel valitakse tööstusvaldkonnad ja seeläbi valimisse ettevõtted, mida hiljem töös uuritakse. Teises alapeatükis kirjeldati TSP ja VRP matemaatilisi optimeerimismudeleid, vajalikke sisendeid ja heuristikaid, mis annavad aluse marsruudiplaneerimise rakenduste kasutamisele. Viimane on sisuliselt võrdne eelmainitud mudelite lahendamisega. Kuna perfektne mudeli lahendamine on päris elus ebatõenäoline, siis toodi ka ülevaade heuristikatest, mis aitavad ligikaudseid lahendeid leida. Kolmandas alapeatükis anti ülevaade nii rändkaupmehe kui ka veotsüklite ülesande lahendamisega seotud rakendustest ja nendega seotud karakteristikutest.

2. METOODIKA

Töö eesmärk on testida tänapäevaste brauseripõhiste marsruudiplaneerimise tarkvarade võimekust ning rakendatavust Eesti jaotusettevõtete praktilistes olukordades. Tänapäevased marsruudiplaneerimise tarkvarad ei pruugi vastata ettevõtete praktilistele vajadustele. See tõstab esile kolm põhilist uurimisküsimust: millised on Eesti jaotusettevõtete marsruutide kavandamise tüüpsituatsioonid; milline on brauseripõhiste marsruudiplaneerimise tarkvarade funktsionaalsus ja kuivõrd sobib marsruudiplaneerimise tarkvara praktiliste ülesannete lahendamiseks. Küsimused tekkisid kokkupuutel mõne marsruudiplaneerimise rakendusega ning seoses nende sagedase mittevastavusega ettevõtte nõudmistele. Autorit motiveerisid lisaks ka tema jaoks huvitavad teemad – operatsioonianalüüsi meetodid logistikas ning vastavad tarkvaralised lahendused.

Uurimisülesannete lahendamiseks viib autor käesolevas töös läbi mitmikjuhtumi analüüsi. Autori uurimiskava koosneb kahest faasist, millest esimene loob tausta juhtumianalüüsile ning teine tegeleb vahetu rakenduste funktsionaalsuse testimisega. Esimene alapeatükk kirjeldab autori poolt moodustatud valimit rakendustest ja võrdleb nende funktsionaalsust. Teine alapeatükk kommenteerib intervjuusid, millega kaasnevalt formuleerisid ettevõtted oma tüüpprobleemide taustsüsteemi, eesmärgi ja täpsustavad parameetrid, ning esitab need edasisele testimisele kuuluvate lähteülesannetena. Täpsemad tähelepanekud testimise läbiviimisest ja selle käigus kaasnenud asjaoludest leiavad käsitlust uurimuse kolmandas peatükis.

2.1. Rakenduste valim

Marsruudiplaneerimise rakenduste valimi koostamisel lähtutakse töös suurema valiku printsiibist pidades samaaegselt silmas töömahtu. Valimisse otsiti rakendusi internetist märksõnade *route optimization software* abil ja valituks osutusid kuus rakendust. Kokku leiti 14 erinevat tarkvara, mis ei ole tõenäoliselt ainsad olemasolevad. Autor üritas saada ligipääsu

kõigile rakendustele ning selleks tuli enamasti täitsa ankeet enda isiku ning organisatsiooni andmete kohta. Täpsustavas lahtris mainis autor, et otsib võimalust antud rakendust lühiajaliselt uurimistöö eesmärgil kasutada. Erandiks 14 tarkvara hulgas olid tasuta brauseris kasutatavad GebWeb Optimap ja RouteXL ning tasulised rakendused GSMtasks ning Onfleet, millele autoril oli juba oma varasemate töökogemuste tõttu ligipääs olemas. Valimisse veel kuulunud OptimoRoute ning Routific nõudsid registreerumiseks lühikest vormi ning seejärel sai rakendusi nädalase kuni kuuajalise perioodi jooksul tasuta proovida. Ülejäänud kaheksa ei vastanud autori kirjale või selgitasid, et nende tarkvara jaoks ei ole tasuta demoversiooni olemas. Seetõttu jäigi valimisse kuus brauseripõhist rakendust. Järgneb iga valimisse kaasatud rakenduse lühikirjeldus.

2.1.1. GebWeb Optimap

Optimap on tasuta internetipõhine tarkvara, mis võimaldab lahendada kuni 100 tipuga rändkaupmehe ülesande, et leida kiireim marsruut või arvutada lihtsalt etteantud marsruudi pikkus ja/või ajakulu (Gebweb Optimap, 2017). Vajalikeks sisenditeks on tippude asukohad kas aadresside või koordinaatide näol. Soovi korral saab täpsustada liikumise tüüpi (jalgsi või jalgratas) ning lisapiirangud nagu kiirteede või tasuliste teede kasutamise keeld. Kui valida liikumise tüübiks jalgsi või jalgrattaga, siis võivad võimalikud marsruudid olla teistsugused ning seetõttu ka lahend erinev. Aadresse saab sisestada kolmel viisil: ühe kaupa tekstina, hulgi tekstina või ühe kaupa kaardil klikkides. Asukohtade sisestamisel aadressid geokodeeritakse ning asukohad visualiseeritakse kaardil. Selles faasis on võimalik täpsustada alg- ja lõpp-punkt, lisada tippudele nimed ja vajadusel kustutada. Kaardil visualiseerimine on väga informatiivne ning annab võimaluse teha otsuseid juba enne optimeerimist.

Optimeerimisel on kolm võimalust: arvutada kiireim ringreis, kiireim algusest lõpuni reis või arvutada distantis ja aeg aadresside või koordinaatide importimise järjekorras. Algusest lõpuni reis tähendab ülesannet, kus on täpsustatud alg- ning lõpp-punkt ning need ei lange kokku. Viimane on fikseeritud otsustusmuutujatega optimeerimine ehk ülesanne, kus on ainult üks võimalik lahend. Kehtib reegel, et suuremate kui 15 tipuga ülesannete lahend ei pruugi olla optimum, vaid on tõenäoliselt parimale lahendile väga lähedal.

Pärast optimeerimist kujutatakse kaardil optimaalne teekond, kuvatakse marsruudi kestvus ja pikkus ning samm-sammult sõidujuhend. Vajadusel saab marsruuti korrigeerida vahetades näiteks mõne peatuse järjestus. Lisaks on võimalik väljund eksportida kas Google Maps'i, tekstina, TomTom või Garmin GPSi. Ühtlasi kuvatakse eksporditav ajakulumatriks, mida optimeerimisel kasutati. Ülesande lahendi internetiaadress on võimalik salvestada, et vaadata hiljem sama lahendit ning vältida sama ülesande mitu korda optimeerimist.

2.1.2. GSMtasks

GSMtasks võimaldab ettevõttel oma autojuhte ning tööülesandeid jälgida. GSMtasks töötab paralleelselt kahel platvormil: administreerimiskeskus GSMtasks veebisaidilt ning välitöö nutiseadmest. Selline süsteem võimaldab info kiiret liigutamist võrdlemisi odavalt, sest puudub vajadus osta spetsiaalseid seadmeid välitöölistele – nutitelefon ning andmesideühendus on tänapäeval laialdaselt levinud. GSMtasks on tasuline ning maksab kuus 45 eurot pluss iga kasutaja kohta 7 eurot. GSMtasks on testimiseks lühiajaliselt tasuta saadaval. (GSMtasks, 2017)

GSMtasks koosneb valdavalt kahest struktuurist: tööülesanne (*task*) ning kasutaja (*user*). Mõlemas struktuuris on palju seotud andmeid - näiteks tööülesande juurde kuuluvad aadress ning ajaaknad ja kasutaja juurde nimi ning telefoni number. Tööülesanded (näiteks kohaletoimetus või peatus) saab luua süsteemi siseselt või need importida ning seejärel visualiseeritakse need kaardil. Ülesande saab ühele kasutajale (autojuhile) määrata. Kasutajad luuakse süsteemi siseselt ning saavad süsteemile ligipääsu läbi GSMtasks rakenduse oma nutiseadmest. Nutiseadmest on näha kasutajale määratud ülesanded ning sellega seotud detailid. Väga mugav on kasutada nutiseadet kui GPSi ning navigeerida end ülesandeni. Ülesande lõpetamisel on võimalik koguda allkiri, teha pilte või lisada ülesandele märkmeid. Kõik erinevates seisundites ülesanded (määratud kasutajale, lõpetatud jne) on andmetena administreerimiskeskusest kättesaadavad.

Kuna ühele kasutajale võib koguneda palju ülesandeid, siis on olemas marsruudi-optimeerimise võimalus, mis arvestab olemasolul ka ajaakendega. Marsruudi-optimeerimiseks vajalikud sisendid on kasutaja, alg- ja lõpp-punkt ning alustusaeg. Algpunktina saab kasutada ka autojuhi asukohta eeldusel, et see on kättesaadav. Optimeerimise väljundiks on järjestatud

ülesannete nimekiri, mida saab koheselt täitma hakata. Kõik optimeerimised salvestatakse ning hiljem on võimalik võrrelda optimeeritud marsruuti tegelikuga.

2.1.3. Onfleet

Onfleet on masinapargi juhtimise tarkvara, mis ühendab ülesanded ning kasutajad ühtsesse süsteemi (Onfleet, 2017). Onfleet ning GSMtasks on sisuliselt väga sarnased. Ka Onfleet paikneb nii veebibrauseris ning telefonirakenduses ning on lõppkasutajale võrdlemisi odav. Kuumakse on sõltuv paketist ehk ülesannete arvust kuus ning algab 125 dollarist. Keskmiselt maksab üks ülesanne 0,15 USD. Optimeerimise eest tuleb maksta fikseeritud lisatasu 0,09 USD ülesande kohta. Onfleeti müügimeeskonnaga ühendust võttes on võimalik tarkvara 30 päeva tasuta testida. (*Ibid.*)

Sarnaselt GSMtasks rakendusele toimub ülesannete ning kasutajate loomine kas veebikeskkonnas või andmeimpordil. Ülesanded jaotatakse kasutajatele, kes kuuluvad kindlasse tiimi ning tegutsevad kindla keskuse ümber. Ülesannetele saab lisada mahuomadused ning siduda omavahel, täpsustades järjestuse, kuidas need tuleb lahendada. See võimaldab praktilise olukorra täpset kujutamist juhul, kus ese on vaja esmalt peale ning hiljem maha laadida. Marsruudioptimeerimine lahendatakse Onfleet rakenduses kasutajatele jaotamata ülesannete hulgas. Optimeerimisel tuleb valida tööde kuupäev, kasutajad, kellele ülesanded jaotatakse, maksimaalne ülesannete arv kasutaja kohta, ülesande kestus (*service time*) ning optimeerimise liik, kas ajaaknaid jälgiv või mittejälgiv. Ühele kasutajale ei saa jaotada üle 200 ülesande, seetõttu on ühele kasutajale optimeerimine efektiivselt piiratud. Ettevõtte kodulehel on kirjas, et praegu ei ole marsruudi optimeerimise algoritm veel võimeline seotud ülesannete kitsendusi jälgima. Optimeerimise lõppedes jaotatakse ülesanded kasutajatele laiali ning kõik kasutajad saavad õiges järjestuses ülesannete nimekirja.

2.1.4. OptimoRoute

OptimoRoute on spetsiaalselt marsruudi optimeerimisteks loodud tarkvara, mis paikneb täielikult veebibrauseris. Hind sõltub paketist ning jääb vahemikku 17-34 USD autojuhi kohta kuus. Tarkvara on võimalik 30 päeva tasuta testida. Sõltuvalt paketist on korraga optimeeritavate asukohtade arv maksimaalselt 250, 500 või 750. (OptimoRoute, 2017) OptimoRoute võimaldab optimeerida ülesandeid projekti kaupa ja kaasata sobiv arv

autojuhte. Asukohad on võimalik importida .csv, .xlsx või .txt failist või tekstina kopeerida. Järgneb geokodeerimise protsess, ning ülevaate leht, kus on võimalik vigased või mitteleitud aadressid parandada. OptimoRoute saab arvestada aja akendega, massi- ning mahupiirangutega, kasutaja oskuste või auto iseärasustega. Saab valida marsruudiga seotud kasutajad ning optimeerida marsruuti. Marsruudi saab eksportida kas .xlsx (Exceli) , .gpx (Garmini), .itn (TomTom) või .freshroute (Navigon) failina. Ühtlasi saab marsruudid saata autojuhtidele e-postiga või SMSina telefoni.

2.1.5. RouteXL

RouteXL on mitme peatusega marsruudiplaneerija, mis arvutab parima teekonna läbi kõikide peatuste ning kuvab sihtkohad parimas järjestuses. RouteXL on tasuta kui korraga optimeerida kuni 20 asukohta ning 35 – 70 euro eest koos on võimalik osta rohkemate asukohtadega pakett. Olukordadeks, kus marsruute ei optimeerita iga päev, on olemas ka päevapilet. (RouteXL, 2017) Asukohtade lisamine toimub kas läbi kaardirakenduse ühe kaupa või tekstina importides. Asukohad lisatakse kaardile ning võimalik on täpsustada marsruudi alg- või lõpp-punkt. Pärast importimist on võimalik leida optimeeritud marsruut ning lahendi veebileht hilisemaks kasutamiseks salvestada. Lahend on võimalik eksportida erinevates formaatides nagu .csv või vaadata marsruuti erinevates kaardirakendustes nagu Google Maps või Graphhopper.

2.1.6. Routific

Routific on marsruudioptimeerimise tarkvara, mis võimaldab korraga optimeerida kuni 2000 asukohta mitme autojuhi vahel. Routific töötab nii veebibrauseris ja autojuhile mõeldud nutiseadme rakenduses. Lisaks optimeerimisele on seetõttu võimalik ka kasutajaid jälgida, klientidelt allkirju koguda ning autojuhtidel lihtsalt marsruutidele ligi pääseda. Kuutasu algab 39 USD ning kõige kallim pakett maksab 79 USD. (Routific, 2017) Optimeeritavad aadressid või koordinaadid saab lisaparaameetritega (peatuse aeg, ajaaknad jne) importida .csv või .xls failist. Autojuhte saab luua brauseris ning võimalik on täpsustada erinevad paraameetreid, näiteks auto mahutavus. Pärast importimist geokodeeritakse kõik aadressid ning vigade mitteilmnemisel on võimalik leida optimeeritud marsruut. Lahend on

võimalik .csv failina alla laadida või saata otse autojuhi telefoni, mida on võimalik Routific nutirakendusest täitma asuda.

2.1.7. Võrdlustabel

Kokkuvõtvalt esitatakse valimi kuue rakenduse parameetrid tabelis 1. Ülesannete piirang kehtib autori poolt testitud rakenduste versioonides.

Tabel 1. Kuue rakenduse võrdlus

	GebWeb Optimap	GSMtasks	Onfleet	OptimoRoute	RouteXL	Routific
Tüüp	Mars.opt*	Fleet mgmt**	Fleet mgmt**	Mars.opt*	Mars.opt*	Fleet mgmt**
Minimaalne hind	Tasuta	52 EUR/kuu	125 USD/kuu	17 USD/kuu	Tasuta	39 USD/kuu
Ülesannete piirang optimeerimisel	100	100	200	250	20	100
Platvormid	Veebibrauser	Veebibrauser + Nutiseade	Veebibrauser + Nutiseade	Veebibrauser	Veebibrauser	Veebibrauser + Nutiseade

*Marsruudi optimeerimise tarkvara

**Veokipargi korraldamise tarkvara

Allikas: koostatud autori poolt.

Tabeli järgi jagunevad rakendused tüübi ja platvormi alusel kahte gruppi: puhas marsruudioptimeerimise tarkvara, mida kasutatakse ainult veebibrauseris ning masinapargi haldamise tarkvara, mida kasutatakse lisaks veebibrauserile ka nutiseadmetes. Ainult marsruudioptimeerimisega tegelevad rakendused on ühtlasi odavamad. Piirangu poolest on tasulised rakendused suurema lubatud peatuste arvuga kui tasuta tarkvarad.

2.2. Ettevõtete ja lähteülesannete kirjeldus

Ettevõtted valiti valimisse arvestades kirjandusest pärinenud väiteid, et VRP kasutatakse näiteks kullerteenustes, prügi korjes või posti jaotusel. Seetõttu otsustati autori poolt, et üks valimisse kuulunud ettevõtte tegeleks kullerteenuste ning teine prügikorjega. Kui kullerteenuste ettevõttega õnnestus hõlpsasti kontakti saada, siis prügifirmadest paljud autori

e-kirjadele ei vastanud ning sobivat ettevõtet tuli kauem otsida. Järgnevad mõlema valimisse kuulunud ettevõtte ning neilt saadud lähteülesannete kirjeldused.

2.2.1. DPD Eesti AS

DPD on rahvusvaheline kullerfirma, mis tegutseb ka Eestis. Nende kontor ning peaterminal asuvad Tallinna külje all aadressil Taevavärava tee 1, Lehmja küla, Rae vald. Ettevõtte põhitegevuse hulka kuuluvad linnasisene jaotus, korje, linnadevaheline ja rahvusvaheline transport. DPD on Eesti juhtiv standard- ja ekspress-pakiveoteenuse pakkuja. (DPD, 2017) Käesoleva töö jaoks valiti ettevõtte tegevustest välja linnasisene jaotus ning ettevõttelt saadi ühe piirkonna päevase jaotusringi aadressid. Piirkonda kuuluvad Tallinna linnaosad Lasnamäe, Pirita ning Viimsi. Marsruudid on dünaamilised ehk muutuvad iga päev.

DPD operatsioonide osakonna projektide spetsialistiga viidi 3. mail 2017 läbi intervjuu eesmärgiga uurida ettevõtte tüüpsituatsioone ja saada marsruudiplaneerimise rakenduste võimekuse uurimiseks lähteülesanded. Lisaks läbiviidud kohtumisele küsiti korduvalt e-posti teel täiendavaid küsimusi. Selgus, et DPD kasutab oma igapäevatöös marsruudiplaneerimise rakendusi. Teenus on sisseostetud ning ettevõtte ise rakendust ei halda. Ettevõttel on võimalik muuta parameetreid, mille alusel marsruudiplaanid genereeritakse. Teenus on üles ehitatud selliselt, et kulleri käsiterminalist saadetakse asukohaandmed teenusepakkujale, kes siis automaatselt marsruudi tagastab. Hommikuti optimeerivad kullerid peale ülesannete jaotamist ning kauba laadimist oma teekonna ja täidavad selle tööpäeva jooksul ilma uusi peatusi lisamata. Marsruudiplaneerimine toimub üldjoontes seega korra päevas.

Intervjuust selgus, et osadel, kuid mitte kõikidel peatustel on aja aknad ning üldjuhul on need kahe tunni laiused. Aja aknad on tellimustel, mis on seotud DPD erinevate teenustega (DPD10, DPD12) või mis on tehtud läbi suurte partnerite e-poodide. Viimasel juhul pakutakse kliendile võimalust määrata endale sobiv saadetise kätte toimetamise aeg. Teenused DPD10 ning DPD12 tähendavad, et saadeti jõuab kliendini vastavalt järgmise päeva kella 10:00 või 12:00. Kuna ettevõtte poolt jagatud andmed aja aknaid ei sisaldanud, siis loodi need Microsoft Excelis tõenäosuslikult funktsiooniga $RANDBETWEEN(0,1)$. Eelmainitud funktsioon jälgib ühtlast jaotuskõverat. Kui väljastatud väärtus oli alla 0,025 (tõenäosus 2,5%), siis on tegemist teenusega DPD10, vahemik $[0,025; 0,05)$ tähendab DPD12 ning vahemik $[0,05;0,15)$ tähendab vabalt valitud aja akent. Vabalt valitud aken genereeriti samuti

ühtlase tõenäosusega vahemikus 07:00 kuni 22:00. Kokkuvõtteks saadi näidisülesanne, mis koosnes 66 asukohast, millest üheksa olid ajaakende kitsendusega ning kõigi teenindusajaks on 10 minutit. Depooks valiti DPD Eesti AS terminal aadressil Taevavärava tee 1, Lehmja, 75306 Harju maakond. Kuigi intervjuust selgus, et pärast jaotusringi hakkavad kullerid tegelema korjega, siis võrreldavuse ning lihtsuse mõttes modelleeriti ka Lasnamäe jaotusringi lõpp-punktiks depoo Lehmja külas.

2.2.2. Eesti Keskkonnateenused AS

Eesti Keskkonnateenused AS on riigi üks suuremaid jäätmeveoga tegelevaid ettevõtteid, mis tegeleb olme- ja ehitusjäätmete kogumise, konteinerite ja mahutite müügiga. Käesoleva töö jaoks valiti tegevustest olmeprügi korje, sest see hõlmab väga paljude asukohtade teenindamist ning seeläbi marsruutide kavandamise vajadust. Ettevõtte Viljandi piirkonna logistikuga viidi 25. aprillil läbi intervjuu, mille kaudu saadi andmeid kahe erineva olukorra kohta: olmeprügi korje haja- ja tiheasustuses. Saadud andmed olid prügiauto ühe täiskoorma kogumiseks vajalikud aadressid. Hajaasustuse piirkond oli Halliste vald Viljandimaal ning tiheasustuse ala Viljandi linnaosa Peetrimõisa. Aja aknaid ega teisi kitsendusi sisse ei arvestatud, sest need ei ole piiravad. Aja aknad on realsuses küll olemas – näiteks kehtiv öörahu kell 6.00 - 22.00. Depooks valiti mõlemal juhul Musta tee 9 Viljandis, kus asuvad Keskkonnateenuste kontor ning masinapark. Lisaks traditsioonilisele rändkaupmehe ülesandele oli kindlaks kitsenduseks jäätmejaama külastamine vahetult enne depoose naasmist. Kõikide teeninduspiirkondade marsruudid on staatilised ning korduvad iga ühe, kahe, nelja või kaheteistkümne nädala järel.

Halliste vald asub Viljandi maakonna lõunaservas. Valla territoorium on 266,4 km². (Halliste, 2017) Halliste ring sisaldas 664 aadressi, millele lisati depoo ning jäätmejaam. Vald koosneb 24 külast ning ühest alevikust. Üheks tööpäevaks on kõigi aadresside teenindamine liiga ajamahukas, seetõttu valiti küladest välja mõned lähestikused, mis sisaldasid umbkaudu 160 aadressi. Arv 160 tulenes asjaolust, et haja-asustuses suudetakse koguda keskmiselt 20 konteinerit tunnis ning tööpäev on umbes kaheksa tunni pikkune. Kuna tegemist on maapiirkonnaga, olid algandmetes asukohad valdavalt formaadis talu nimi, vald, küla. Mõned näited on Napsihansu, Halliste vald Kulla küla ning Vedru Märt, Halliste vald, Mõõnaste küla. Halliste alevikus ning ka Õisu külas esines ka nimedega tänavaid ning nummerdatud

maju. Paljud sellised aadressid puuduvad enim kasutatavatel kaartidel ning seetõttu oleks nende geokodeerimine väga keeruline. Seetõttu küsis autor ka ligipääsu nende aadresside geograafilistele koordinaatidele.

Peetrimõisa on Viljandi kirdepoolseim linnaosa. Peetrimõisa olmeprügi korjering sisaldas algandmetes 221 aadressi, millele lisati autori poolt veel kümmeimikond imiteerimaks vastava piirkonna täielikku teenindamist. Põhjus selles peitus intervjuus mainitud faktis, et kogu Peetrimõisa teenindatakse korraga, kuid andmetes puudusid mõne tänava aadressid. Õnneks oli tegemist valdavalt tupiktänavatega ning terve tänava teenindamiseks oli vaja lisada ainult tänava kinnises otsas asuv aadress.

Eesti Keskkonnateenused AS kasutab oma tegevuste juhtimisel spetsiaalselt nende jaoks arendatud programmi Toits, mis on loodud kasutamiseks nii kontorist operatsioonide jälgimiseks ja prügiautost pideva informatsiooni talletamiseks. Programm võimaldab liikuvaid autosid kaardilt jälgida, planeerida marsruute, jälgida autode peale paigutatud kaameraid (näiteks reklamatsioonide lahendamiseks ja tõestamiseks), täita sihtpunkti saabumisel staatus (korras, konteiner puudu jne), vajadusel autojuhi nutiseadmega fotosid teha jpm. Programmi kohta palju informatsiooni hankida võimalik ei olnud, sest intervjuueeritava teate põhjal on tegemist tundliku infoga, mida ettevõtte väliselt ei jagata.

2.3. Kokkuvõte

Teises peatükis esitas autor tööga seotud metoodika. Sissejuhatavalt toodi välja uurimisprobleem, uurimisküsimused ja autori motivatsioon vastava teema valikul. Esimeses alapeatükis kirjeldati töös uuritud marsruudiplaneerimise rakenduste valimi moodustamist ja esitati lühike ülevaade kõigist kuuest valimisse kuulunud tarkvarast. Rakendused jagunesid kahte rühma: puhtad marsruudioptimeerimise tarkvarad ja veokipargi korraldamise tarkvarad. Teises alapeatükis leiduvad tööga seotud kahe ettevõtte kirjeldused, intervjuudest tulenev informatsioon ning rakenduste testimise jaoks küsitud lähteülesannete kirjeldused. Peatükis väljatoodu andis lähteülesanded marsruudiplaneerimise rakenduste testimiseks ja rakendused, mida testimisel kasutatakse.

3. MARSRUUDIPLANEERIMISE RAKENDUSTE TEST

Käesolevas peatükis tuuakse välja marsruudiplaneerimise rakenduste testimise käik ning tulemused. Esimeses alapeatükis kirjeldatakse üksikasjalikult ettevõtete kolme lähteülesande lahendamist kuue tarkvaraga. Esitatakse ülesande lahendamisega seotud detailid, ilmnunud vead ning täpsustatakse lahendusprotsessi mugavust. Teises alapeatükis paiknevad lähteülesannete testimise tulemused ning hinnatud parameetrite kirjeldus. Lisaks tulemustele on selles alapeatükis kirjas autori nägemuses parimad rakendused iga ülesande kohta, mis esitati ülesande andnud ettevõtetele, ja ettevõtete tagasiside rakenduste poolt genereeritud lahendite kohta.

3.1. Rakendustega lähteülesannete lahendamine ja kasutajakogemus

Töös toimunud rakenduste testimine toimus järgmiselt. Esmalt valiti ülesande ning rakenduse spetsifikatsioone arvestades välja lahendusstrateegia. Seejärel kohandati andmeid, et need vastaksid valitud lahendusstrateegiale. Viimane osa sisaldas testi läbiviimist nii, et mõõdeti kogu tööks ja rakendustel lahendi arvutamiseks (optimeerimiseks) kulunud aega. Pärast testi läbiviimist hinnati kasutusmugavust, märgiti üles väljundite põhiparameetrid ringi pikkus ning ringi aeg ja hinnati, kas väljund sobib päris elus täitmiseks. Kõikide lähteülesannete lahendamise tulemused rakenduste kaupa on esitatud koondtabelites alapeatükis 3.2.

3.1.1. Lähteülesanne nr 1 - Halliste korjering

GebWeb Optimap - Optimapi kasutamine on lihtne ja mugav, kuid ajamahukas. Kuna andmed imporditakse tekstina, siis saab vajalikud koordinaadid lihtsalt sisendlahtrisse kopeerida. Koordinaadid peavad olema formaadis (laiuskraad, pikkuskraad). Tähele tuleb

panna, et kahe koordinaadi eraldajaks on koma ning seetõttu ei tohi see ühes reas ilmnedagi rohkem kui korra. See tähendab, et kümnendkohad koordinaatides tuleb eraldada täiskraadidest punktiga nagu tehakse seda Ameerika Ühendriikides. Kui aadressid on sisestatud, tuleb vajutada nupule *Add list of locations* ning need visualiseeritakse kaardil. Koordinaatide paigutamine kaardile toimub hetkeliselt ning pärast seda avaneb võimalus täpsustada alg- ja lõpp-punkt.

Halliste korjering optimeeriti kahetasemeliselt. Kõrgemal tasemel valiti peatusteks 11 vallas asuvat küla, eelviimaseks peatuseks pidi olema Viljandi külje all asuv jäätmekäitlusjaam ning lõpp- ja algpunktiks Eesti Keskkonnateenuste depoo Viljandi linnas. Pärast esmast optimeerimist oli jäätmekäitlusjaam teine peatus, kuid kasutades Optimapi pakutud võimalus *Reverse Order* saadi kõigile kitsendustele vastav marsruut. Kõrgema taseme optimeerimisest tulenes külade teenindamise järjekord.

Madalamal tasemel võeti mitme küla kaupa kõik sinna kuulunud asukohad, paigutati need koordinaatidega kaardile ning optimeeriti teekond, üritades sõidusuunas esimene maja sättida alg- ning viimane lõpp-punktiks. Kuna Optimap on optimeerimisega võrdlemisi aeglane, siis hoiti korraga optimeeritavate asukohtade arv 30 lähedal. See teadmine tuleneb autori kogemusest varasemal optimeerimisel, kus 66 asukoha korraga optimeerimine kestis üle kolme tunni. 30 peatust andis optimeerimisajaks umbes 2 minutit. Summaarseks teekonna pikkuseks on madalama tasemete optimeerimiste teepikkuste summa.

GebWeb Optimap oli oma lihtsusega valimist kõige kiiremini hoomatav ning õpitav tarkvara. Väga selge ning ühel küljel paiknev menüü andis kohe märku, kuidas käib asukohtade sisestamine. Optimeerimise kiirus oli teiste rakendustega võrreldes väga aeglane. Kõige enam meeldis autorile registreerumise ja sisselogimise puudumine ning seetõttu võiks Optimap olla harval kasutamisel esimene valik.

GSMtasks - Halliste korjeringi optimeerimine GSMtasks'is ei õnnestunud. Probleem seisnes asukohtade importimisel koordinaatidena, mis tõlgendati portaali poolt korduval proovimisel siiski valesti. Pikkus- ja laiuskraadid vahetati pidevalt ümber nii, et autoril ei õnnestunud kõiki 100 aadressi korraga Eesti kaardile paigutada. Kuna ka GSMtasks'i veebilehel selle kohta midagi mainitud ei olnud, siis loeti lahendamise võimatuks. Väga ajamahukaks osutus valesti imporditud aadresside kustutamine, sest seda sai teha ainult ühe kaupa.

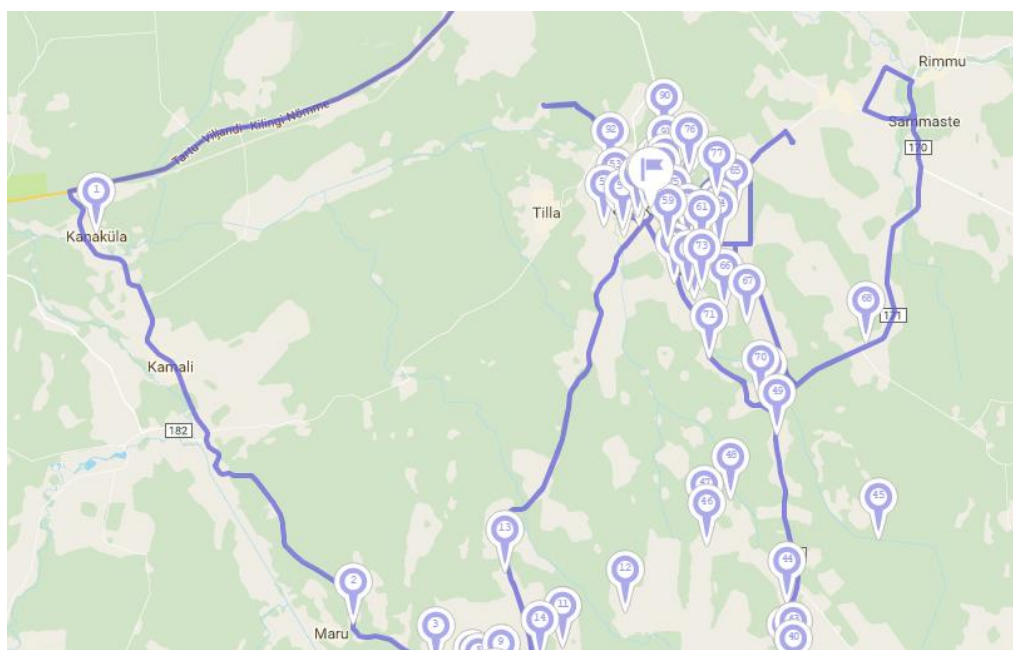
Onfleet - Halliste ring lahendati Onfleetis ühe optimeerimisega. Kõik asukohad imporditi koordinaatidena Onfleeti. Ülesanded optimeeriti ning eksporditi uues järjekorras tagasi Excelisse. Kui kõik aadressid olid järjestatud, siis mõõdeti tulemus 100 kaupa GebWeb Optimapi *Calculate in Order* funktsiooniga. Kasutajakogemuse vaatepunktist oli Onfleetis optimeerimine meeldiv, sest pärast optimeerimist saadud järjestuse sai mõne hiireklikiga ekspordida ning ülesanded nimistust kustutada. Optimeerimisel saadud marsruut oli disaini poolst ebaselge. Punktide vahelisi teekondi kuvati ainult nooltena, mis tekitas kasutajas veidi segadust. Pole kindel, kas optimeerimisel võeti aluseks teede võrk või kasutati eukleidilisi pikkusi. Autor eelistab näha ka optimeerimise tulemusel saadud marsruudi pikkust kui optimeerimise kokkuvõtet.

OptimoRoute - OptimoRoute's optimeeriti Halliste korjering ühe optimeerimisega. See osutus võimalikuks kuna peatuste ülempiir oli testversioonis 250. Andmete importimisel määrati teenindusajaks 1 minut, mis hiljem koguajast maha lahutati. Asukohad sisestati koordinaatidena ning probleeme ei tekkinud, andmeimport toimus võrdlemisi kiirelt. Seejärel määrati autojuhile lõpp- ning algpunkt, mille määramine oleks võinud autori arvates toimuda vahetult enne optimeerimist. OptimoRoutes tuli avada autojuhtide loend, kus sai autojuhi sätteid (k.a. alustus- ning lõpetuskoht) muuta. Optimeerimine toimus väga kiirelt ning optimeeritud ringi pikkus ning aeg olid nähtavad vastava autojuhi juures. Tulemuseks saadud teepikkus oli teiste rakendustega võrreldav, kuid ajakulu oli oluliselt suurem. Autor seletust sellele ei leidnud.

RouteXL - Esmane plaan Halliste korjeringi lahendamiseks oli kahe-tasemeline optimeerimine. Kõrgem tase koosnes algselt 25st külast, jäätmejaamast ning depoost. Kuna RouteXL peatuste piirang on tasuta versioonile 20, siis moodustati mõningatest külge külje kõrval asuvatest küldes miniklastrid ning lahendati siis 20 peatusega rändkaupmehe ülesanne. Seejärel sisestati algselt väljaarvatud külad autori jaoks loogilisse kohta nimekirjas. Teise taseme optimeerimine kahjuks ei õnnestunud, sest koordinaatide sisestamisel kuvati ligi pooltel juhtudel veateade, et koordinaat ei ole täpne aadress. Ühtlasi oli kogu geokodeerimise protsess isegi koordinaatidega võrdlemisi aeglane. Aadresside kasutamisel jäid samuti paljud asukohad geokodeerimata. Seetõttu loeti Halliste ring RouteXL rakendusega lahendamatuks.

Routific - Halliste korjering optimeeriti kahe-tasemeliselt. Kõigepealt määrati kindlaks külade ning seejärel kuni 100 kaupa majade järjestus. Koordinaatide importimisega probleeme ei tekkinud ning ajakulu oli seetõttu väike. Lisaks koordinaatidele oli aga tarvis

importida identifitseerimisnumber ning väljundi loetavuse parendamiseks ka aadressid. Kokku toimus 3 optimeerimist, millest üks oli külade järjestuse määramiseks. Üldine liikumissuund leiti esimese, kõrgema taseme optimeerimisega. Kui tükikaupa optimeerides ei olnud liikumissuund üldoptimeerimisega sama, kasutati fiktiivseid asukohti, et marsruuti suunata.



Joonis 2. Fiktiivne peatus soosib lähenemist loodesuunast.

Allikas: Routific.

Esimese tükikaupa optimeerimise algpunkt sätiti depoosse ja lõpp-peatus viimasesse blokki kuulunud külasse. Teine optimeerimine algas esimese lõpust ning lõppes jällegi depoos. Väljastatud marsruudid ei sisaldanud võimalusel tagasipöördeid, kuid haja-asustuse tõttu tekkisid väga pikad ümbersõidud. Need tagasipöörded, mis ilmnesisid paigaldati ristmike juurde, mis võisid olla üsna kaugel ning põhjustasid jälle tohutult palju lisakilomeetreid. See on ka põhjuseks, miks Routific kilometraaz on teiste tarkvaradega võrreldes väga suur. Kasutajakogemus oli Routific kasutades positiivne ja meeldejääv. Kõik marsruudi optimeerimiseks vajalikud sammud kuvati suurelt ekraanile ja nende täitmine oli lihtne. Esmalt luuakse vabalt valitud kuupäevale uus projekt, seejärel imporditakse ülesanded ning kui kõik on korras, tuleb vajutada ekraanil ainsat oranži ning suurt nuppu *Optimize Route!*

3.1.2. Lähteülesanne nr 2 - Lasnamäe jaotusring

Lasnamäe jaotusringi tulemused rakenduste kaupa on toodud tabelis 3 alapeatükis 3.2.

GebWeb Optimap - Lasnamäe jaotusringi optimeerimine osutus väga keeruliseks ning ajamahukaks, sest Optimap ei ole võimeline aja akendega arvestama. Seetõttu tuli aja aknaid jälgida manuaalselt ning ühtlasi pidada kellaaja arvestust. Protseduuri esimene osa oli kõige varem aeguva ülesande täitmine ning seejärel „vaba ajaga“ täita läheduses nii palju ülesandeid kui võimalik, enne järgmise aja aknani sõitmist. Optimeerimist kasutati vaba aja faasis ning üldjuhul mahtus sellesse tühja aega alla kümne peatuse. Seetõttu olid optimeerimiste tulemused optimaalsed, kuid tõenäoliselt optimeerimiste järjekord ning klastrid mitte. Summaarne teepikkus arvutati klastrite optimeerimiste teepikkuste ja klastrite vaheliste liikumiste liitmisel. Ringi läbimise aja leidmiseks liideti kahele eelmainitule komponendile veel 66 asukoha teenindusajad. Nii Lasnamäe kui ka Peetrimõisa ülesannetes toimus aadresside geokodeerimine ühekaupa, ligikaudse kiirusega 1 aadress sekundis. Seetõttu võttis suuremate ülesannete lahendamine juba importimisel võrdlemisi kaua aega. Geokodeerimisel tuvastati aadressid ilma rangete formaadi reegliteta, mis on kasutajamugavuse poole pealt väga suureks plussiks.

GSMtasks - Lasnamäe jaotusringi optimeerimine oli GSMtasks'iga võrdlemisi lihtne. Kuna algandmed asusid Microsoft Excelis, siis oli neid lihtne GSMtasks'i importida. Excelis oli vajalik portaalil töö vähendamiseks määrata kasutaja nimi ning täpsustada teenindusaja kestvus. Viimase välja ning aja akende korrektne importimine autoril ei õnnestunud ning seetõttu kasutas ta võimalust muuta neid välju portaalil sees. Kuna peatusi oli 66 ning maksimaalne, mida GSMtasks lubab, on 100, siis õnnestus autoril ülesanne ühe optimeerimisega lahendada. Eelnevalt tuli täpsustada lõpp- ning algpunkt, mis mõlemad sätiti ühtseks depooga Rae vallas, ning töö alustamise ajaks pandi hommikune aeg kell 7.00. Optimeerimine kestis vähe aega ning tulemused ilmusid koheselt *Route Optimizations* lehele.

Onfleet - Lasnamäe jaotusring koosnes 66 aadressist, millest üheksa omasid kahe tunni laiust aja akent. Aja aknad lisati aadressidele formaadis dd/mm/yyyy HH:mm. Sellised formaadis kellaegade lisamine oli väga tülikas. Aadressidele lisati ka riigi nimetus. Andmed imporditi ning loodi töökeskus (*hub*) Tallinn aadressiga Taevavärava tee 1, Lehmja küla, Rae vald. Optimeerimisel kasutati kasutajat *Priidik*, teenindusajaks sätiti 10 minutit ning valiti aja aknaid järgiv optimeerimistüüp. Optimeerimine ei õnnestunud, kuna portaal kuvas veateate, et

ükski autojuht ei ole ülesannete jaoks saadaval. Klienditoega ühendust võttes teavitati autorit, et aja aknaid on importimisel vigade vältimiseks parem sisestada formaadis UNIX (millisekundeid alates 1. jaanuar 1970 UTC). Pärast muudatuste tegemist ei õnnestunud autoril endiselt marsruudi optimeerimist edukalt läbi viia ning teistkordsel suhtlusel klienditoega tunnistasid klienditeenindajad, et suutsid probleemi ka enda arvutites taaslavastada. Siiski suutsid Onfleeti töötajad marsruudi edukalt optimeerida ning väljund oli määratud kasutajale *Priidik*. Kuna Onfleetis ei kuvata summaarset teepikkust ega aega, siis eksporditi kasutajale *Priidik* määratud ülesanded optimaalses järjestuses Excelisse ning teepikkuse ning aja mõõtmiseks kasutati GebWeb Optimapi *Calculate in Order* funktsiooni.

OptimoRoute - OptimoRoute's oli ülesanne võimalik lahendada ühe ajaakendega optimeerimisega. Sisendid imporditi Microsoft Excelist ning sisaldasid täisaadressi (formaadis tänav maja number korteri number, linn, postiindeks), teenindusaega (10 minutit) ning aja-aknaid. Aadressid geokodeeriti kiirelt ning lahendamise marsruut algas ning lõppes DPD terminali aadressil Lehmja külas. Väljund eksporditi Excelisse.

RouteXL - Lasnamäe jaotusringi lahendamine RouteXL oli identne lahendamis- protseduuriga, mida kasutati GebWeb Optimapi puhul, sest ka RouteXL ei arvesta aja akendega. Kui võimalus oli, siis täideti üks ajaaknaga ülesanne, seejärel täideti läheduses nii palju ülesandeid kuni järgmise aja akna aegumiseni aeg on ning korrati protsessi. Optimeeriti vabal ajal täidetavaid ülesandeid. Tulemused olid optimeerimiste ning optimeerimisvaheliste liikumiste summa.

Routific - Optimeerimine oli Routific väga lihtne, sest ülesande sai lahendada ühe aja akendega optimeerimisega. Aadresside importimisel lisati aja aknad ning teenindusaja kestvus. Optimeerimisel kasutati autojuhti *Priidik* ning marsruut pidi algama ning lõppema DPD depoes. Tulemus ilmus kaardile ning oli autori arvates päris elus sõidetav.

3.1.3. Lähteülesanne nr 3 - Peetrimõisa korjering

Peetrimõisa ringi tulemused rakenduste kaupa on toodud tabelis 4 alapeatükis 3.2.

GebWeb Optimap - Peetrimõisa korjering koosnes ligikaudu 230st aadressist ning kuna Optimap üle 100 asukoha korraga optimeerida ei lase, siis tuli teha olulisi kärpimisi. Tiheasustuse korjering on olemuselt *arc routing problem* ehk läbida ei tule mitte niivõrd iga

aadressi, vaid tänavat. Seetõttu ei kaotata aadresside eemaldamisega tähtsat informatsiooni. Protseduur näeb ette, et esmalt lisatakse mudelisse depoo, jäätmejaam, kõikide tupiktänavate otsas asuvad aadressid ning kõikide ristmikvaheliste lõikude keskpunkti. See eeldab andmete kaardilt nägemist. Autor kasutas tupiktänavate aadresside ning tänavate keskpunktide leidmiseks kaardirakendust Google Maps. Samuti oli autoril eelnev tutvus Peetrimõisa linnaosaga, mis oluliselt lihtsustas planeerimist. Pärast esmast aadresside leidmist oli kogutud 32 aadressi ning teekond optimeeriti Optimapis. Kuna lahend sisaldas palju tagasipöördeid, mis on päris elus kitsaste tänavate tõttu raskendatud või võimatud, siis lisati mudelisse läbimata tänavalõikude juures asuvaid aadresse, et katta kõik läbimist vajavad tänavad. Mudel lahendati uuesti ning vajadusel lisati täiendavaid aadresse. Protsessi korrati, kuni lahend rahuldab autorit. Autor leppis mõne lõigu mitteläbimisega, kui see oleks päris elus prügiauto tegevusraadiusesse jäänud. Lõplik ring koosnes 53 aadressist.

GSMtasks - Marsruudi genereerimine oli GSMtasks'is lihtne, kuid võttis aega. Kuna aadresse oli üle 200 ja piirang on 100, siis tuli valida umbes pooled. Kui aadressid on järjestatud tänava kaupa ning tõusva maja numbriga, siis eemaldades iga teise rea jäävad alles kõikide tänavate ühe tänavapoolse majade aadressid. Excelis on võimalik seda teha väga efektiivselt Visual Basic makrodega (vt Lisa 1). Eeldusel, et prügikorje toimub samaaegselt mõlemal tänavapoolel ning kõikidel tänavatel on majad pidevalt mõlemal poolel tänavat, ei kaota poolte aadresside eemaldamine väärtuslikku informatsiooni. Esimene eeldus kehtib Peetrimõisas täielikult, kuid teine alati mitte. Seetõttu tuleb jälgida, et andmeridade kustutamisega ei kaotataks väärtuslikku informatsiooni. Et kärpida peatuste arvu sajani, eemaldati tänavalõikude keskelt veel üksikuid aadresse. Valmis aadressid imporditi koos kasutaja nimega GSMtasks'i. Järgnes optimeerimine, mille algus ning lõpp olid depoos. Ülevaade tulemusest ilmus *Route Optimizations* lehele.

Onfleet - Onfleet lubab ühele kasutajale optimeerimisel määrata maksimaalselt 200 aadressi, seetõttu tuli algset 230 kärpida. Eemaldati tänavalõikude keskelt üksikud aadressid ning seejärel valmistati Onfleeti importimiseks ette. Onfleet vajab sisendite importimisel lisaks tänava nimele ning maja numbrile kindlasti ka linna ning riigi nime. Seetõttu tuli need Microsoft Excelis olemasolevatele aadressidele lisada. Kuna optimeerida saab ülesandeid, mis ei ole ühelegi kasutajale veel määratud, siis jäeti kasutaja nime väli Excelist importimata ning kõik ülesanded liikusid automaatselt *Määramata* ülesannete kausta. Onfleeti portaalil tuli luua uus töökeskus (*hub*), sest kõik optimeerimised algavad ja lõppevad töökeskuses. Keskus

paigutati Eesti Keskkonnateenuste Viljandi depoosse aadressil Musta tee 9. Optimeerimisel kasutati ühte kasutajat *Priidik*, maksimaalne ülesannete arv kasutaja kohta sätiti 200, teenindusajaks minimaalne 1 minut ning optimeerimise tüübiks aja aknaid mitte jälgiv, sest aja aknaid mudelisse ei kaasatud. Pärast optimeerimist määrati ülesanded kasutajale *Priidik* ning teekond oli kaardilt jälgitav. Kuna Onfleetis ei kuvata summaarset teepikkust ega aega, siis eksporditi kasutajale *Priidik* määratud ülesanded optimaalses järjestuses Excelisse ning teepikkuse ning aja mõõtmiseks kasutati GebWeb Optimapi *Calculate in Order* funktsiooni.

OptimoRoute - Optimeerimine oli OptimoRoutes väga lihtne, sest ülesanne tuli lahendada ainult korra. Aadresside importimine oli kiire tänu võimalusele teha seda tekstina. Ülesande algandmed sai otse sisendlahtrisse kopeerida ning geokodeerimisega probleeme ei tekkinud. Kasutaja-spetsiifilised lõpp- ning algpunkti seadistamine olid intuiiivsed ning optimeerimine toimus mõne sekundiga. Saadud marsruut kontrolliti töö autori poolt üle ning vastas kõigile tingimustele. OptimoRoute poolt väljastatud marsruudi pikkus ning ajakulu olid võrreldes teiste tarkvaradega autori jaoks ootamatult suured. Kilometraaz oli umbes 50% ning ajakulu ligi 2 korda suurem. Põhjus peitub tõenäoliselt OptimoRoute algoritmides, mis tunduvad lisavat iga peatuse kohta mingi suurusega pikkuse ning aja trahvi. Kinnitus sellele aga puudub ning selle välja uurimine ei mahu selle bakalaureusetöö skoopi.

RouteXL - Väikese peatuste piirangu tõttu osutus Peetrimõisa ülesande lahendamine ebamõistlikuks. Klasterdamine oleks suutnud efektiivselt lahendada mõne tänava jagu ülesannet, kuid nii väikeste klastrite puhul on mõistlikum kasutada kaarti ja pliiatsit või mõnda kaardirakendust ning joonistada marsruut inimõistust kasutades. See viis oleks ühtlasi kiirem ning väljund vastaks paremini päris elu kitsendustele (näiteks tagasipöörde keeld, poolik tänavate teenindamine jne).

Routific - Lahendusstrateegia oli siin üks optimeerimine. Routific demoversiooni maksimaalne korraga optimeeritavate peatuste arv on 100, seetõttu tuli Peetrimõisa aadresse kärpida ligikaudu 130 võrra. Andmete ettevalmistamise protseduur oli identne GSMtasks'iga. Esmalt eemaldati Excelist Visual Basic makroga (vt Lisa 1) iga teine rida ning seejärel eemaldati tänavalõikude keskelt veel üksikuid aadresse. Allesjäänud 100 aadressi imporditi Routific'usse ning optimeeriti. Väljund sisaldas palju tagasipöördeid ning mõnda ebamõistlikku ümberpööramisringi.

3.2. Tarkvara testimise tulemused

3.2.1. Lähteülesannete tulemused

Järgnevalt on esitatud kõigi kolme ülesande lahendusi ning tulemusi kirjeldavad tabelid. Tabelites leidub järgnev informatsioon. Lahendusstrateegia kirjeldab üldist lähenemist optimeerimisele. Lahendatud ülesanne on matemaatiline mudel, mida rakenduse taustal lahendati. Optimeerimiseks kulunud aeg on stopperiga mõõdetud aeg, mis kulus rakenduse algoritmidel lahendi arvutamiseks. Kogu töö aeg on koos sisendite töötlemise, importimise ja väljundiga seotud tegevusteks kulunud aeg. Ringi pikkus on summaarne teepikkus ja ringi aeg on summaarne ringi läbimise ajakulu. Hinnang kasutamisele vastab küsimusele kui mugav oli antud tarkvaraga antud ülesannet lahendada ning hinnang tulemustele vastab küsimusele kas lahend sobib päris elus kasutamiseks.

Tabel 2. Halliste korjering

Halliste vald (11 küla)	GebWeb Optimap	GSM-tasks	Onfleet	OptimoRoute	Route XL	Routific
Lahendusstrateegia	2-tasemeline optimeerimine	X	Üks optim.	Üks optim.	X	2-tasemeline fiktiivsete asukohtadega optimeerimine
Lahendatud ülesanne	TSP	X	TSP	TSP	X	TSP
Optimeerimiseks kulunud aeg	8 min	X	50 s	5s	X	20 s
Kogu töö aeg	32 min	X	15 min	12 min	X	30 min
Ringi pikkus	188 km	X	227 km	190 km	X	311 km
Ringi aeg	5 h 24 min	X	6 h 9 min	11h 39min	X	6 h 39 min
Hinnang kasutamisele	Ajamahukas, lihtne	X	Lihtne	Väga lihtne	X	Keerukas, mugav
Hinnang tulemustele	Kasutuseks sobiv	X	Parema puudumisel rahuldav	Kasutuseks sobiv	X	Parema puudumisel rahuldav

Allikas: koostatud autori poolt.

Halliste valla marsruudi optimeerimine õnnestus nelja tarkvaraga. Parimaid lahendeid andsid GebWeb Optimap ja OptimoRoute. Mõlema rakenduse poolt soovitatud marsruudi kilometraaz oli konkurentidest oluliselt lühem ning marsruut tundus ka loogiline. OptimoRoute hinnang ajakulule on veider ning kilomeetreid vaadates ei ole ajakulu

hindamise süsteem näiteks Optimapiga sama. Parimate lahendite hindamisel ei võetud arvesse töö aega, sest mitme aasta lõikes staatiliste marsruutide plaanimisel ei ole see oluline. Kriitika Onfleeti tulemuste pihta on ebaselge lõppmarsruudi kuvamine.

Tabel 3. Lasnamäe jaotusring

Lasnamäe jaotusring	GebWeb Optimap	GSM-tasks	Onfleet	OptimoRoute	RouteXL	Routific
Lahendusstrateegia	Manuaalne klasterdus, tükikaupa optim	Üks ajaakendega optim	Üks ajaakendega optim	Üks ajaakendega optim	Manuaalne klasterdus, tükikaupa optim	Üks ajaakendega optim
Lahendatud ülesanne	TSP	TSPTW	VRPTW	VRPTW	TSP	VRPTW
Optimeerimiseks kulunud aeg	1 min	16 s	20 s	2 s	47 s	7 s
Kogu töö aeg	1h	8 min	10 min	3 min	1h	4 min
Ringi pikkus	127 km	106,64 km	155 km	122,57 km	127 km	126.29
Ringi aeg	15h 28min	14h 43min	15h 55min	15h 25min	15h 28min	15h 20min
Hinnang kasutamisele	Väga keeruline	Lihtne, kuid ebamugav	Lihtne*	Lihtne, mugav	Väga keeruline	Lihtne, mugav
Hinnang tulemustele	Parema puudumisel rahuldav	Kasutuseks sobiv	Kasutuseks sobiv	Kasutuseks sobiv	Parema puudumisel rahuldav	Kasutuseks sobiv

* Kasutamine oli lihtne, kuid autoril ei õnnestunud lahendit leida - seda tegi klientidugi.

Allikas: koostatud autori poolt.

Lasnamäe jaotusring sai lahendatud kõigi rakendustega. Onfleeti puhul lahendati ülesanne klientide poolt. Ajaaknaid mitte arvestavad Optimap ja RouteXL olid selle ülesande lahendamise väga aeganõudvad ning keerulised. Parimad lahendid saadi GSMtasksi, OptimoRoute ja Routificu poolt. Onfleeti pakutud marsruut osutus ligi poole pikemaks kui GSMtasksi marsruut. Pakutud ringi ajad on väga pikad ning sõltuvad suuresti teenindusaegadest, mis moodustavad tabelis 3 kujutatud aegadest umbes kaks kolmandikku. Kasutushinnang GSMtasksi kohta on „lihtne, kuid ebamugav“, sest eksimuste tegemisel on vanu ülesandeid väga tüütu ja ajamahukas ühekaupa kustutada. Peale selle oli kasutamine võrdlemisi lihtne ning kergelt mõistetav.

Tabel 4. Peetrimõisa korjering

Peetrimõisa linnaosa	GebWeb Optimap	GSM-tasks	Onfleet	OptimoRoute	RouteXL	Routific
Lahendusstrateegia	Korduv, täienev optim	Üks 100 aadressiga optim	Üks 200 aadressiga optim	Üks optim	X	Üks 100 aadressiga optim
Lahendatud ülesanne	TSP	TSP	TSP	TSP	X	TSP
Optimeerimiseks kulunud aeg	11min 10s	40s	1min	3s	X	10s
Kogu töö aeg	50min	20min	15 min	5 min	X	8 min
Ringi pikkus	19 km	19,74 km	21,4 km	27,04 km*	X	25,01 km
Ringi aeg	45min 30s	58 min	59min 25s	1h 56min*	X	50 min
Hinnang kasutamisele	Ajamahukas, keeruline	Ebamugav	Mugav	Mugav, lihtne	X	Mugav, lihtne
Hinnang tulemustele	Parema puudumisel rahuldav	Parema puudumisel rahuldav	Parema puudumisel rahuldav	Parema puudumisel rahuldav	X	Parema puudumisel rahuldav

* OptimoRoute arvutusalgoritmid töötavad konkurentidest väga erinevalt. Sama aadressijärjestuse mõõtmine GebWeb Optimapis andis tulemuseks 19,19 km ning 53 min 9 s.

Allikas: koostatud autori poolt.

Peetrimõisa korjeringi ülesandele ei andnud ükski rakendus häid lahendeid. Ajakulu erinevused põhjustas eelkõige lahendusstrateegia, mis tulenes rakenduses seatud maksimaalsest peatuste arvust. Lahendiks pakutud marsruudi pikkus oli valdavalt sarnane, erinesid ainult Routific oma veidra ümberpööramisega ja OptimoRoute, mis marsruudi jälgides tundus täiesti mõistlik ja teise rakendusega üle mõõtes andis palju parema tulemuse.

3.2.2 Ettevõtete tagasiside tulemustele

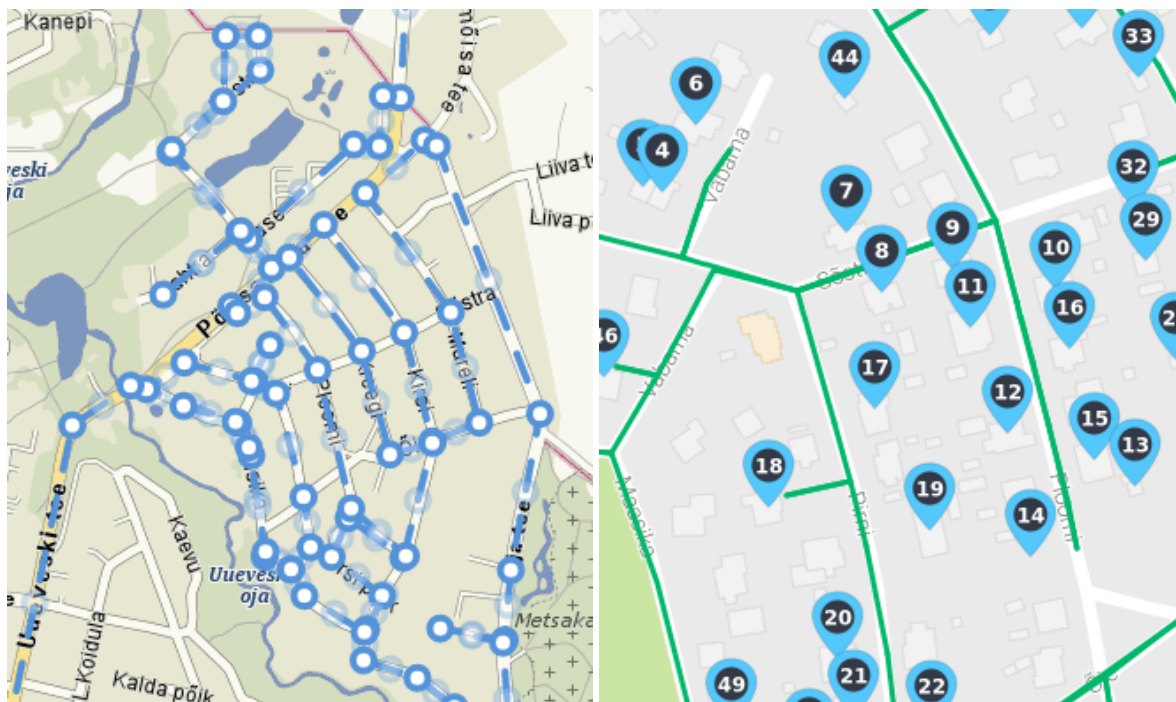
Saadud testide tulemustest lähtuvalt valiti autori poolt iga ülesande kohta välja mõned sobivaimad tarkvarad ning tulemused koos tarkvara kirjeldusega saadeti tagasi ettevõtetele. Rakenduste valimisel hinnati nii väljundi kvaliteeti kui ka kasutusmugavust.

Halliste korjeringile andsid parimaid lahendeid OptimoRoute ning üllatuslikult vabavara GebWeb Optimap. OptimoRoute andis suhteliselt hea lahendi, nõudis vähe tööaega ning võimaldas lahendada ülesanne ühe optimeerimisega. Optimapi saja peatuseline piirang muutis küll lahendamise veidi keerukamaks ja ajamahukamaks, kuid saadud lahend oli

hinnanguliselt kvaliteetne. Eesti Keskkonnateenused logistik hindas mõlemad lahendid kasutatavaks eeldusel, et marsruudid saab kasutamiseks navigatsioonisüsteemi importida.

Lasnamäe ülesande parimad lahendid olid kõiki parameetreid arvestades genereeritud GSMtasks'i, OptimoRoute ja Routificu poolt. Pärast lahenditega tutvumist andis intervjuueeritav teada, et vastavad rakendused pigem ei ole DPD Eesti vajadustele vastavad. DPD Eesti kindel nõudmine on, et nemad saaksid valida, kas optimeerimise tulemus oleks lühima kilometraaži või aja. Soovitatud rakendused sellist funktsionaalsust ei oma. Kuna DPD Eesti pakub palju erinevaid teenuseid näiteks C.O.D. ehk lunateenus, siis peavad kõik optimeerimisega seotud faktorid - nagu näiteks teenindusaeg - ka pakutavast teenusest sõltuma. Üks tähtsaim faktor rakenduse sobivuse koha pealt on andmete liigutamine. Kujul, mida kasutas autor testide tegemisel (andmete importimine Excelist veebibrauseris kasutatavasse rakendusse) ei ole samuti DPD Eesti jaoks sobiv. DPD Eesti kullerid optimeerivad endale jaotatud ülesanded ise, kuid kõik soovitatud rakendused töötavad dispetšer-autojuht stiilis, kus dispetšer tegeleb planeerimisega ja korraldamisega ning autojuht täidab marsruudiplaani. Autojuhtidele on kindlasti vaja platvormi, kus nemad saavad mugavalt enda marsruutide planeerimisega tegeleda.

Peetrimõisa teenindamiseks vajalik marsruut loodi aktsepteeritaval tasemel GSMtasks'i, Optimapi ja OptimoRoute poolt.



Joonis 3. ja 4. Peetrimõisa ülesande lahendid Delfi kaardil vasakul ja GSMtasksis paremal.

Allikas: Delfi Kaart, GSMtasks.

Kuna kõik kolm rakendust sisaldasid manöövreid, mis on päris elus raskendatud siis saatis autor võrreldavuse mõttes ka kaardirakenduse Delfi kaart abil käsitsi joonistatud kõikidele päris elu kitsendustele vastava marsruudi, mis oli optimeeritud marsruudist marginaalselt pikem. Lahendid saadeti intervjueeritavale tutvumiseks ning intervjueeritav tões, et ka nende tarkvara annab tihti kasutatuid lahendeid. Lõplik marsruudiplaneerimine toimub seega siiski autojuhi poolt. Kui tarkvara genereeritud marsruut ei ole loogiline, siis autojuht sekkub ning sõidab nii kuidas saab ja on mõistlik.

3.3. Järeldused ja arutelu

Selles peatükis kirjeldati ettevõtelt intervjuude käigus saadud lähteülesannete lahendamist ja testimist kolme jaotus- või korjeringi näitel Hallistes, Lasnamäel ja Peetrimõisas. Ühtlasi analüüsiti kuue rakenduse kasutamist ja lõpuks esitati tarkvara testimise tulemused võrdlustabeli kujul.

Halliste valla lähteülesannet iseloomustasid hajali paiknevad peatused ning võrdlemisi suur peatuste arv - 160. Peatuste arv oli küll suurem kui mõne tarkvara ülempiir, kuid sellest hoolimata oli võimalik leida häid lahendeid mitme tasemelise ja/või tükikaupa optimeerimisega. Kui marsruudid on staatilised ehk neid kasutatakse pikas perspektiivis palju kordi, siis sobivad eelnimetatud lahendusstrateegiad paremini, sest need pikendavad töö aega. Sobivad lahendid saadi nii tasulise rakenduste OptimoRoute kui ka vabavaralise Optimapi kasutamisel. Rakendustega, mis võimaldavad korraga rohkem asukohti optimeerida, on sellise lähteülesande lahendamine oluliselt lihtsam, kuid see ei taga paremat kvaliteeti. Parimad saadud lahendid kategoriseeriti ettevõtte poolt kasutatavaks, juhul kui need oma infosüsteemidesse edukalt üle kanda saab.

Lasnamäe jaotusringi puhul andsid kõik valimisse kuulunud ajaakendega arvestavad rakendused mõistlikke lahendeid. Kuna jaotuslogistika on tänapäeval tänu e-kaubanduse levikule suur ning kasvav valdkond, siis on paljud marsruudiplaneerimise rakendused just selle segmendi endale teenindamiseks võtnud. Siis on tähtis, et oleksid olemas lisavõimalused, mis võimaldavad kulleril ise optimeerida. Ettevõtte poolt hinnati rakendused kasutuskõlbmatuks andmete aeglase importimise tõttu.

Peetrimõisa lähteülesande jaoks oli vastuvõetavaid lahendeid raske leida. Sellele läheneti rakenduste funktsionaalsuse tõttu kui rändkaupmehe ülesandele ning mõistlikke lahendeid ei andnud ükski töös kasutatud rakendus. Enimlevinud vead olid järsud tagasipöörded ning peatuste ülempiiri tõttu tekkinud mudelisse lisamata majade mitteteenindamine. Ettevõtte tagasisides seisis, et ka nende tarkvara teeb samu vigu ning lõplik otsus tehakse töö ajal autojuhi poolt. Samalaadsete ülesannete lahendamisel võibki olla optimeerimiselt saadud lahendi käsitsi parendamine kõige mõistlikum variant.

Rakenduste testi kokkuvõtteks võib öelda, et Halliste lähteülesandele suudeti leida nii autori kui ka ettevõtte arvates mõistlik lahend nii tasulise kui ka vabavaralise rakendusega; Lasnamäe lähteülesande lahendid olid autori hinnangul head, kuid ei sobinud ettevõttele aeglase andmete impordi tõttu; Peetrimõisa lähteülesandele ei andnud ükski töös kasutatud rakendus ega ka ettevõtte enda tarkvara sobivat kasutusvalmis marsruuti. Seetõttu võib öelda, et isegi vabavaralise rakendusega on mõnedel juhtudel (Halliste korjering) võimalik saada ettevõttes kasutatavaid lahendeid; sobiva lahendi genereerimine ei taga kasutatavust ettevõttes (Lasnamäe jaotusring) ning häid lahendeid tiheasustuse korjeülesannetele on väga keeruline marsruudiplaneerimise rakendustega leida. Tarkvara arendajad võiksid seega panna rohkem rõhku andme importimise lihtsustamisele ning anda ka võimaluse autojuhtidele enda marsruut ise optimeerida. Samuti võiksid arendajad lisada praegustele algoritmidele lisavõimalusi (nt tagasipööride keeld, kogu tänavalõigu järjest teenindamine), mis suudaksid genereerida mõistlikke lahendeid ka tiheasustuse korjele. Tarkvara kasutajad peaksid rakenduste valimisel kindlasti jälgima, kas nende ülesanded lahendatakse mõistlikult, mitte eeldama, et marsruudioptimeerimise funktsionaalsus lahendab kõik nende marsruudi planeerimisega seotud mured.

KOKKUVÕTE

Töö eesmärgiks oli seatud testida tänapäevaste nii vabavaraliste kui tasuliste marsruudiplaneerimise tarkvarade võimekust ning rakendatavust Eesti jaotusettevõtete praktilistes marsruudiplaneerimise olukordades. Töös kasutati kuut brauseripõhist marsruudiplaneerimise rakendust, et lahendada kahelt ettevõttelt saadud kolm lähteülesannet. Lahendamisel keskenduti lahendusprotsessi mugavusele, lihtsusele ja eelkõige lahendi kvaliteedile ning hinnangulisele praktilisele kasutatavusele. Sobivamad lahendid esitati ettevõtetele kommenteerimiseks. Töö olulisteks tulemusteks olid hinnangud, mis saadi kas autori või ettevõtete poolt erinevate rakenduste kasutatavuse kohta praktikas. Püstitatud eesmärk sai täidetud.

Töö autor leidis vastused kõigile püstitatud uurimisküsimustele. Neist esimene uuris, millised on Eesti jaotusettevõtete marsruutide kavandamise tüüpsituatsioonid. Töö käigus leiti kolm erinevat tüüpsituatsiooni. DPD Eesti AS kirjeldas linnasisest jaotusringi, mis koosnes ligikaudu seitsmekümnest peatusest. Mõnedel peatustel olid ka ajaaknad, mis tulenesid kas suurklientide erikohtlemisest või erinevatest teenustest. Eesti Keskkonnateenused AS tõi välja kaks tüüpsituatsiooni – olmeprügi korje tihe- ja haja-asustuses.

Teiseks uurimisküsimuseks oli, milline on brauseripõhiste marsruudiplaneerimise tarkvarade funktsionaalsus. Kokku uuriti kuut rakendust. Rakendused jagunesid kahte suuremasse rühma puhas marsruudiplaneerimise tarkvara ja veokipargi korraldamise tarkvarast. Viimased omasid lisaks veebrauseri versioonile ka autojuhtidele mõeldud nutiseadme rakendust. Kõigil rakendustel oli piiratud korraga optimeeritavate peatuste arv, mis jäi töö autori kasutatud versioonidel vahemikku 20 – 250. Kaks rakendust olid tasuta kasutamiseks (GebWeb Optimap ja RouteXL) ning ülejäänud neli kuupõhise kasutustasuga.

Kolmandaks uuriti, kuivõrd sobib marsruudiplaneerimise tarkvara praktiliste ülesannete lahendamiseks. Kõik ettevõtetelt saadud lähteülesanded üritati kõigi rakendustega läbi lahendada. Kolmel juhul 18 seast lahendamine ei õnnestunud. Halliste korjering lahendati edukalt nelja rakendusega, millest kaks esitati ettevõttele ning kiideti tingimuslikult heaks. Lasnamäe jaotusring lahendati kõigi, kuid edukalt nelja rakenduse poolt, mis suutsid

ajaakendega arvestada. Lahendid olid mõistlikud, kuid selgus, et andmetega töö võtab liialt kaua aega ning rakenduste muu funktsionaalsus ei vasta ettevõtte soovidele. Peetrimõisa korjeringile ei andnud mõistlikke lahendeid mitte üksi rakendus. Lahendid sisaldasid päris elus väga keerukaid manöövreid ning tagasipöördeid. Ettevõtte mainis, et ka nende enda tarkvara teeb selliseid vigu ning lõpliku marsruudi määrab autojuht.

Marsruudiplaneerimise rakenduste töös esitatud testimisest on eelkõige kasu uuringus osalenud rakenduste haldajatele, kes saavad ilmnenuid vead parandada, märkused arvesse võtta ning edasistes plaanides päris elu vajadustest lähtuda. Samuti võib kasu olla ettevõtetele, kes jaotuse, korje või mobiilse teenindamisega tegelevad ning marsruudiplaneerimise rakendusi enda tööks vajavad. Autoripoolsed ettepanekud tarkvarapakkujatele on parandada töös väljatoodud vead ning ebamugavad aspektid rakenduse kasutamisel. Kindlasti tasub pakkujatel ka õppust võtta ettevõtetele hindamiseks saadetud ja parimaid lahendeid andnud rakendustelt ning täiustada lahendeid genereerivaid algoritme või lihtsustada andmetöötlusprotsesse. Kasutajad peaksid kindlasti enne rakenduste hankimist veenduma, et just nende lähteülesanded lahendatakse tarkvara poolt edukalt, kiirelt ja mugavalt.

SUMMARY

THE CAPABILITIES OF BROWSER-BASED ROUTE PLANNING APPLICATIONS

Priidik Vilumaa

The aim of this thesis is to test the capabilities of modern freeware and SaaS route planning applications. Six route planning applications were used to solve three problems posed by two Estonian companies. When solving the problems the author focused on convenience, simplicity and most important of all – the quality and usability of output. The best solutions were to be commented on by the original companies. Important results of this thesis included the evaluations of usability by either the author or the aforementioned companies.

The author of this thesis found answers to all three proposed research questions. The first research question aimed to understand the basic planning situations of route planning in Estonian distribution related companies. The author found three different situations. DPD Eesti AS described an intercity distribution run which comprised of roughly 70 stops. Some of those had time windows which were caused by either special treatment of large customers or different services. Eesti Keskkonnateenus AS, a waste management company, suggested two situations – waste collection run in a city and in a rural area.

The second research question looked into the functionality of browser based route planning applications. Six route planning applications were investigated in total. The applications could be split into two distinct groups: pure route optimization software and fleet management software. The latter included in addition to web browser version also a mobile device version designed for drivers. All of the applications had upper bounds for the number of stops optimized simultaneously. The upper bound for the versions of applications used by the author were between 20 and 250. Two of the applications were free and the rest had periodic costs for using.

Third and final research question was to find out how well does route planning software fit the needs of Estonian distribution companies. All of the problems posed were solved by all six applications. However, not all of the tests could be successfully completed due to different reasons. The Halliste Parish waste collection problem was successfully solved with four applications of which two were sent back to the company and were conditionally approved. The Lasnamäe distribution problem was solved well by all applications capable of taking time window constraints into account. The solutions were reasonable nonetheless the company did not approve the applications because of long time needed to import the input data. Also these applications lacked other functionality required by the company. None of the applications used in this thesis gave reasonable solutions to the Peetrimõisa waste collection problem. All proposed results included sharp U – turns which are impossible in real world. The same shortcomings are generated by the software used by the company.

Beneficiaries of this premise include software developers of the applications used in this thesis. They are able to take note of the flaws discovered and base their future efforts on real world needs. This thesis might also be beneficial to companies who are operating in the fields of distribution and are about to use such applications. Author of this thesis suggests to fix all errors encountered in some of the tests. Moreover software developers should take note of the applications which produced the best results and improve their own calculation algorithms or simplify working processes in their applications.

KASUTATUD ALLIKAD

An, M., Lee, H. (2015). Design of the Integrated Logistics Information System Based on Cloud Computing. <http://www.inse.org/vol3/160-A016.pdf> (15.04.2017).

Anbuudayasankar, S.P., Ganesh, K., Mohapatra, S. (2014). Models for Practical Routing Problems in Logistics. Design and Practices. Springer.

Bieding, T., Görtz, S., Klose, A. (2009). On line Routing per Mobile Phone A Case on Subsequent Deliveries of Newspapers. https://link.springer.com/chapter/10.1007%2F978-3-540-92944-4_3 (21.05.2017)

Business dictionary. (2017). www.businessdictionary.com/definition/route-planning.html (21.05.2017)

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. (2000) The VRP with Time Windows. Gerad. <http://www.bernabe.dorrnsoro.es/vrp/data/articles/VRPTW.pdf> (21.05.2017)

Delfi Kaar. (2017). www.kaart.delfi.ee (29.05.2017)

Desaulniers, G., Madsen, O.B.G, Ropke, S. (2014). The Vehicle Routing Problem with Time Windows. Siam.

DPD. (2017) www.dpd.com/ee (20.05.2017)

GebWeb Optimap. (2017). www.gebweb.net/optimap/ (20.05.2017)

Goel, A. (2008). Fleet Telematics. Real-Time Management And Planning Of Commercial Vehicle Operations. Springer.

GSMtasks. (2017). www.gsmtasks.com/ (20.05.2017)

Halliste. (2017). www.halliste.ee (20.05.2017)

Irnich, S., Toth, P., Vigo, D. (2014) The Family of Vehicle Routing Problems. Siam.

Jaeger, J. (2005) Ajaakendega transpordiülesande lahendamise. Tartu Ülikool. www.dspace.ut.ee/bitstream/handle/10062/998/jaeger.pdf;jsessionid=6B462CB2EBEE6BF5C77F1DE3A1E8E54F?sequence=5

Laporte, G., Ropke, S., Vidal, T. (2014) Heuristics for the Vehicle Routing Problem. Siam.

Mell, P., Grance, T. (2011). The NIST Definition of Cloud Computing. National Institute of Standards and Technology. www.nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (21.05.2017)

Merriam-Webster. (2017). www.merriam-webster.com/dictionary/heuristic (21.05.2017)

Minner, S. (2016). Transportation Logistics ainekonspekt Müncheni Tehnikaülikool.

Onfleet. (2017). www.onfleet.com (20.05.2017)

OptimoRoute. (2017). www.optimoroute.com (20.05.2017)

ORMS Today. (2017) www.orms-today.org/surveys/Vehicle_Routing/vrssmain.html (23.05.2017)

RouteXL. (2017). www.routexl.com (20.05.2017)

Routific. (2017). www.routific.com (20.05.2017)

Sanders, P., Schultes, D. (2007). Engineering Fast Route Planning Algorithms. www.link.springer.com/chapter/10.1007/978-3-540-72845-0_2#page-1 (21.05.2017)

LISAD

Lisa 1. Visual Basic makro.

```
Sub DeleteEveryOtherRow()  
Dim rng As Range  
Dim InputRng As Range  
xTitleId = "Excel"  
Set InputRng = Application.Selection  
Set InputRng = Application.InputBox("Range :", xTitleId, InputRng.Address, Type:=8)  
Application.ScreenUpdating = False  
For i = InputRng.Rows.Count To 1 Step -2  
    Set rng = InputRng.Cells(i, 1)  
    rng.EntireRow.Delete  
Next  
Application.ScreenUpdating = True  
End Sub
```

Visual Basic makro iga teise rea eemaldamiseks

Allikas: www.extendoffice.com/documents/excel/808-excel-delete-every-other-row.html