

## Liite 1: Projektin toteutus

### 1. Kirjallisuuskatsaus

#### 1.1. Yleistä

Kirjallisuuskatsauksen tavoitteena on hankkia tietoa jo toiminnassa olevista ja suunnitelluista kimpakyytijärjestelmistä. Suurimman mielenkiinnon kohteena ovat kimpakyytimallien ratkaisuisissa käytetyt algoritmit, sekä mahdolliset mallinnusprosessissa esiin nousseet erityiskysymykset ja rajoitteet. Tavoitteena on myös löytää ideoita kimpakyytijärjestelmän houkuttelevuuden maksimointiin. Ajatuksena on, että aiheeseen tutustumalla nopeutetaan omaa suunnitteluprosessia. Eri artikkeleista ja kuvauksista on tarkoitus poimia käyttökelpoisia ideoita ja toteutustapoja, ja soveltaa niitä mahdollisuuksien mukaan oman järjestelmän toteutuksessa. Toiveena on myös löytää informaatiota epäonnistuneista hankkeista tai lähestymistavoista, sillä jo kerran tehtyjä virheitä ei haluta toistaa.

#### 1.2. Kirjallisuuskatsauksen sisältö

Kirjallisuuskatsaukseen on tarkoitus sisällyttää ainakin seuraavat tutkimukset/artikkelit:

1. *The Long Term Car Pooling Problem: on the soundness of the problem formulation and proof of NP-completeness* (Varrentrapp et al. 2002). Artikkelissa käsitellään muun muassa kimpakyytiongelman formulointia, ja on siten hyvä lähtökohta tämän projektin ongelman formuloinnille. Artikkeliksi keskittyy pitkän aikavälin (vastakohtana päiväkohtaisen) järjestelmän kuvaukseen.
2. *A distributed geographic information system for the Daily Car Pooling Problem* (Calvo et al. 2003). Artikkeliksi keskittyy päiväkohtaisen kimpakyytijärjestelmän kuvaukseen. Tämänhetkisen näkemyksemme mukaan kehittämämme kimpakyytijärjestelmä on luonteeltaan päiväkohtainen, jonka johdosta tämä artikkeli toimii hyvänä pohjana ongelman formuloinnille sekä ratkaisumenetelmille. Ongelman ratkaisussa on käytetty *Dijkstra*-algoritmia<sup>1</sup>, jota tulemme mahdollisesti käyttämään omassa ratkaisussamme. Artikkeliksi sisältää myös case –kuvauksen Milanon lähellä toteutetusta testiajosta. Artikkelin viiteluettelossa on myös mielenkiintoisia viitteitä, joihin mahdollisesti tutustumme.
3. *Procedures for planning multi-leg journeys with fixed-route and demand-responsive*

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Dijkstra%27s\\_algorithm](http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm)

*passenger transport services* (Horn 2004). Artikkel, jossa kuvataa YTV<sup>2</sup>:n *Reittioppaan*<sup>3</sup> kaltaista järjestelmää, jossa matkustajalle haetaan paras mahdollinen matkustustapa paikasta A paikkaan B. Matkustajan preferenssit otetaan huomioon, ja matka voi koostua useasta eri osasta. Artikkel on hyödyllinen, koska toteuttamassamme yhdistelyalgoritmissa tulee kyydin pyytäjille hakea näiden preferenssien perusteella heille sopivia tarjottuja kyytejä.

4. *Dynaamisen kimppakyytijärjestelmän toimivuus simuloitussa kuormitustilanteessa* (Kekoni 2005). Diplomityö, jossa kuvataan Amsterdamin Schipholin lentokentän tarpeisiin kehitetty kimppakyytijärjestelmä. Työssä tarkastellaan järjestelmän toimivuutta simuloitussa kuormitustilanteessa. Käytämme artikkelia erityisesti simuloinnin, joka on osa projektiamme, rakentamisen pohjana.

Yhdysvalloissa on toteutettu kimppakyytijärjestelmä, jossa pyritään yhdistelemään mannerta pitkin ilmaitse matkustavien liikemiesten ym. reittejä siten, että lentokoneiden käyttökapasiteetti olisi mahdollisimman suuri. Tämän järjestelmän kuvaukseen on tarkoitus tutustua mahdollisuuksien mukaan.

Lisäksi haemme tieteellisten artikkelien kokoelmista muita tutkimuksia ja muiden toteutettujen järjestelmien kuvauksia. Jo tiedossa olevien käyttökelpoisten artikkeleiden ja julkaisujen lähdeluetteloista löytynee myös tutustumisen arvoisia viitteitä.

## **2. Kimppakyytijärjestelmän algoritmit**

### **2.1. Reitinhakualgoritmi kyydin tarjoajille**

Reitinhakualgoritmi hakee (kyydin tarjoajille) parhaimman reitin kahden pisteen välille. Reitinhaussa voidaan minimoida joko

- matkaa tai
- matka-aikaa.

Mikäli reitinhakualgoritmi tuottaa reitin välille A-E, välipisteinä B, C ja D, syntyviä tarjottavia reittejä ovat A-B, ... A-E, B-C, ..., B-E, C-D, C-E. Esimerkiksi reitin A-B täytyminen ei vähennä reitillä B-C olevia vapaita paikkoja, mutta reitin A-E ollessa täynnä, ei myöskään reitille B-E ole mahdollista ilmoittautua.

---

<sup>2</sup> <http://www.ytv.fi/FIN/>

<sup>3</sup> <http://aikataulut.ytv.fi/reittiopas/fi/>

## 2.2. Yhdistelyalgoritmi

Yhdistelyalgoritmi etsii kyydin pyytäjälle tämän antamia kriteereitä vastaavan parhaan (tai joukon parhaita – riippuen lopullisesta toteutustavasta) reitin. Pyydetylle kyydille annetaan seuraavat *reunaehdot*:

- enimmäiskävelymatka pisteelle, josta kyytiin noustaan (Koska tieverkon ei sisällä kaikkia teitä, joudutaan kävelymatkan pituutta arvioimaan lähtöpisteen ja pisteen, josta kyytiin noustaan, välistä etäisyyttä pisteiden välisellä suoralla etäisyydellä)
- aikaikkuna, jolloin kyytiin nousun tulee tapahtua (esimerkiksi 8:00-8:15)

Algoritmi etsii reunaehdot täyttävät reitit sekä asettaa nämä paremmuusjärjestykseen (tai toteutustavasta riippuen etsii parhaimman) siten, että jokin seuraavista *kyytikriteereistä* täyttyy:

- kokonaismatka-aika on lyhin mahdollinen
- ajomatka on lyhin mahdollinen.

## 2.3. Yhdistelyprosessi

Tarjottujen ja pyydettyjen kyytien yhdistely voi tapahtua kahdella tavalla:

1. Pyytjä löytää käytettävissä olevista tarjotuista kyydeistä itselleen sopivan ja ilmoittautuu tähän.
2. Pyytjä ilmoittautuu kyydin pyytäjäksi (kyytijonoon) asettamalle kyydille tietyt reunaehdot ja kriteerit.

Ensimmäisessä tapauksessa pyytäjälle haetaan yhdistelyalgoritmillä ensin reunaehdot täyttävät kyydit, jonka jälkeen sopivat kyydit järjestetään paremmuusjärjestykseen (tai toteutustavasta riippuen etsitään paras kyyti) kriteerien perusteella.

Jälkimmäisessä vaihtoehdossa kyydin pyytäjät laitetaan kyytijonoon ilmoittautumisjärjestyksessä. Kyytijonon käsittely etenee ajastetusti seuraavan prosessin mukaisesti:

1. Ennen kyytijonon käsittelyä tarjottuja kyytejä ilmestyy ja häviää. Tänä aikana myös kyytijonoon tulee tai sieltä häviää pyytäjiä.
2. Kyytijonoa aletaan käsitellä järjestyksessä (*First In First Out*). Mikäli kyytijonossa olevan pyytäjän *reunaehdot* ja *kyytikriteereitä* vastaava kyyti löytyy varataan kyyti kyseiselle pyytäjälle.
3. Ne pyytäjät, joille ei löytynyt reunaehdot vastaavia kyytejä siirretään takaisin käsittelyjonon viimeisiksi, sikäli, kun heidän määrittämänsä aika, johon asti he ovat valmiita vastaanottamaan tietoja kyydeistä, ei ole umpeutunut.

4. Ne pyytäjät, joille on varattu kyyti, joko vahvistavat tai peruvat kyytiin liittyvän varauksen. Mikäli pyytjä hylkää varauksen, hänet siirretään kyytijonon viimeiseksi.

## 2.4. Muut toteutustavat

Käytettävissä olevan ajan puitteissa tullaan kokeilemaan myös muunlaisia yhdistelyprosesseja. Näiden tehokkuutta voidaan mitata simuloinnilla ja siten verrata ensisijaiseen toteutustapaan (kuvattu yllä). Mahdollisia muita toteutustapoja kyytien yhdistelemiseksi ovat ainakin:

- erilaiset *jononkäsittelysäännöt*, esimerkiksi erilaisin priorisointisäännöin; tai
- *maksimoidaan systeemin kokonaisyötyä*, esimerkiksi kyydinhaun onnistumisprosenttia.

Tässä on kiinnitettävä huomiota siihen seikkaan, että kyytejä tehokkaimmin järjestelmä ei välttämättä ole käyttäjän kannalta houkuttelevin. Sikäli, kun aika ei riitä useampien toteutustapojen testaamiseen, tullaan muita menetelmiä tutkimaan ainakin kirjallisuuskatsauksen puitteissa.

## 3. Kimppakyytijärjestelmän tietotekninen toteutus

Projektsuunnitelmassa kuvattu esimerkkitoteutus tehdään Java-ohjelmana. Java-kielen valitsimme siksi, että sitä kaikki osaavat ainakin jonkin verran. Matrex Oy:ltä on saatu Uudenmaan tiepiirin kuvaus. Kuvaus koostuu solmuista (risteykset) ja niitä yhdistävistä linkeistä (tienpätkät). Linkkeihin on sisällytetty matka-aika eli kustannus. Lisäksi solmuille on koordinaattitiedot. Solmuja on noin 2000 kappaletta ja linkkejä noin 5000 kappaletta, joten kuvauksesta muodostettua graafia on mahdollista käsitellä tavallisella kotikoneella.

Ohjelman sisältämät alustavat komponentit ovat *tieverkon esitys*, *reittien hallinnointi*, *algoritmit* ja *simulaattori*. Tieverkon esitys tarkoittaa tiepiirin kuvauksen tuomista ohjelmaan ja siitä muodostettavaa linkitettyä graafia. Reittien hallinnointi -komponentti pitää kirjaa pyydetyistä ja tarjotuista reiteistä ja tarjoaa rajapinnan niiden lisäämiseen ja poistamiseen. Algoritmit -komponentti on tarkoitus toteuttaa siten, että niiden muuntelu ja vaihtaminen on yksinkertaista. Simulaattori käyttää reittien hallinnointi -komponenttia ja luo satunnaista kysyntää ja tarjontaa, sekä laskee *tunnuslukuja*.

Tarkoituksena on tuottaa ohjelmisto, jonka hallinta ja muokkaus on yksinkertaista ja eri komponenttien korvaaminen paremmilla on mahdollista, jos jokin toteutus ei täytä vaatimuksia. Tämän mahdollistamiseksi komponenttien välille määritellään rajapinnat, jotka käyttävät yhteisiä tietotyyppejä.

## 4. Kimppakyytijärjestelmän testaus ja simulointi

### 4.1. Mallin testaus

Mallin testauksella pyritään selvittämään

1. Täyttääkö malli sille saavutetut tavoitteet
2. Miten malli toimii satunnaisilla syötteillä
3. Kuinka hyvin malli toimisi todellisessa tilanteessa

Ensimmäisen tyypin testausta suoritettaneen jo mallin kehitysvaiheessa. Antamalla yhdistelyalgoritmille yksinkertaisia syötteitä voidaan varmistaa, että algoritmi tuottaa ”järkeviä” reittejä. Yllätyksiä tässä testauksen vaiheessa voidaan vähentää algoritmin huolellisella suunnittelulla. Tässä vaiheessa voidaan vielä tutkia erilaisia muutoksia algoritmiin, kuten kyydin tarjoajien joustoja tai erilaisia tapoja käyttää yhdistelyalgoritmia

### 4.2. Mallin simulointi

Kun malli on saatu valmiiksi ja sen tiedetään toimivan, voidaan sen ominaisuuksia selvittää Monte Carlo -simulaation avulla. Monte Carlo -simulaatiossa mallin syötteet ovat satunnaisia. Simuloinnissa voidaan esimerkiksi generoida kyydin pyytäjien ja tarjoajien määrät sekä määränpäätt jostakin satunnaislukujakaumasta. Tällä menetelmällä saadaan kuva siitä miten malli toimii erilaisissa tilanteissa. Selvitettäviä asioita ovat ainakin

1. Mikä on alaraja kyydin tarjoajien ja pyytäjien määrälle, jotta algoritmilla saataisiin järkeviä reittejä (kyydin pyytäjille)? (On selvää, että jos tarjolla on vain kyyti Tampereelta Poriin ja ainoa kyytipyyntö on Kemistä Rovaniemelle, ei kyytejä löydy.)
2. Onko systeemillä jokin *saturaatoraja*, jonka jälkeen käyttäjien määrän lisääntyminen ei enää oleellisesti paranna reittejä?
3. Miten autojen täyttöaste kehittyy?

Viimeisenä testauksen vaiheena on testaaminen todellista tilannetta kuvaavalla datalla. Tässä vaiheessa jouduttaneen jollakin tavalla arvioimaan, kuinka suuri osa ihmisistä olisi valmis vaihtamaan oman auton tai julkisen liikenteen kimppakyytijärjestelmään. Yksinkertaisimmillaan tämä voisi tapahtua skaalaamalla nykyistä työmatkaliikennettä jollakin vakiolla väliltä (0..1). Pendeliliikenteen määrästä on olemassa tilastoja ja voidaankin olettaa, että sieltä, mistä ihmiset ajavat paljon töihin omilla autoillaan, olisi myös paljon halukkaita kimppakyytiläisiä. Nämä

## Työ 5: Matrex Oy

halukkaat kimppakyytiläiset voivat nykytilanteessa liikkua joko omilla autoillaan tai julkisilla kulkuvälineillä. Jotta päästäisiin lähelle todellisuutta on satunnaisuutta syytä lisätä simulaatioon tässäkin vaiheessa. Käyttäjien määrä voidaan olettaa olevan otos jostakin jakaumasta jonka odotusarvo on järjestelmän käyttäjien arvioitu määrä. Määränpäät on ehkä myös syytä satunnaistaa, siten että tarjottavat reitit ja pyydetyt reitit vastaavat suhteellisilta osuuksiltaan suunnilleen työmatkaliikenteen todellisia suuntia. Näin kyettäisiin ainakin osittain vastamaan seuraaviin kysymyksiin

1. Onko järjestelmä toteuttamiskelpoinen, toisin sanoen: ylittäisikö käyttäjien määrä suurella todennäköisyydellä edellisessä kohdassa selvitetyn alarajan käytettävyydelle?
2. Mikä olisi autojen täyttöaste? Kuinka paljon liikenne vähenisi jos järjestelmä saataisiin toimimaan?