Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

Optimaliseren van de service van een taxisysteem met zelfsturende voertuigen

(Engelse titel: Optimizing the service of a taxi system with autonomous vehicles)

Verslag ten behoeve van het
Delft Institute of Applied Mathematics
als onderdeel ter verkrijging

van de graad van

BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE

door

Irene Vooijs

Delft, Nederland
Juli 2017

BSc verslag TECHNISCHE WISKUNDE

"Optimaliseren van de service van een taxisysteem met zelfsturende voertuigen"

(Engelse titel: "Optimizing the service of a taxi system with autonomous vehicles")

Irene Vooijs
Studienummer: 4360656

Technische Universiteit Delft

Begeleiders

Dr. J.W. van der Woude
Prof.Dr.Ir. H.X. Lin

Overige commissieleden

Drs. E.M. van Elderen   Dr.Ir. L.J.J. van Iersel

Juli, 2017   Delft

Abstract

Dit project behandelt de programmering en toepassing van een taxiservice, waarbij de passagiers zich vanaf of naar het treinstation willen verplaatsen. Met behulp van geheeltallig lineair programmeren wordt bepaald welke ritten door de taxi's moeten worden gereden om de winst te maximaliseren. Dit model is gebaseerd op het model dat beschreven is in het verslag van Xiao Liang et al. uit 2016.
Er worden twee modellen vergeleken: in Model 1 is het systeem vrij om verzoeken te accepteren of te weigeren, terwijl in Model 2 per zone beslist wordt of alle ritten al dan niet geaccepteerd worden. De taxiservice wordt eerst toegepast op kleine schaal, waarna enkele aanpassingen gedaan worden om het model ook op grote schaal te kunnen toepassen. Bij de toepassing op kleine schaal wordt altijd verlies gemaakt, omdat de ratio taxi's per zone erg hoog is. Voor de toepassing op grote schaal blijkt dat het model voor veel taxi's sterk overeenkomt met het model van Liang, maar voor kleinere hoeveelheden taxi's minder, omdat de ritten in Liang's model minder homogeen gegenereerd worden. De resultaten voor 20, 40, 60 en 80 taxi's worden met elkaar vergeleken. Het optimale aantal taxi's om te gebruiken is altijd 20 voor Model 1, en 40 voor Model 2.

Voorwoord

Voor u ligt het verslag van het bacheloreindproject dat ten behoeve van het behalen van het bachelordiploma Technische Wiskunde te TU Delft is geschreven.
Dit project is uitgevoerd op de wiskundesectie Mathematische Fysica onder leiding van J.W. van der Woude en H.X. Lin, maar heeft voornamelijk raakvlakken met programmeren en optimalisatie.

Het afgelopen kwartaal heb ik mij bezig gehouden met het analyseren van het optimalisatieprobleem van een taxisysteem, en voornamelijk met het programmeren van dit model. Vervolgens heb ik de resultaten onderzocht en de nauwkeurigheid ervan gecontroleerd.

Het artikel van Xiao Liang uit 2016 is een belangrijke bron geweest voor het oorspronkelijke optimalisatieprobleem. De formulering van het probleem heb ik aangepast en vervolgens heb ik er een eigen programma voor geschreven [Liang et al., 2016]. Bij interesse in de code van de programmering kunt u contact met mij opnemen.

Ik wil graag mijn begeleiders hartelijk bedanken voor hun ondersteuning bij het project, en L.J.J. van Iersel en E.M. van Elderen voor het plaatsnemen in de beoordelingscommissie.

Irene Vooijs
Juli 2017

Inhoudsopgave

1 Inleiding **11**

2 Model **13**
2.1 Modelkeuze 13
2.2 Aannames 14

3 Probleemstelling **15**
3.1 Verzamelingen 15
3.2 Variabelen 15
3.3 Parameters 16
3.4 Doelfunctie 16
3.5 Voorwaarden 17
3.6 Overige berekeningen 19
3.7 Programmeren in Matlab 19

4 Toepassing op kleine schaal **21**
4.1 Resultaten en discussie 22
4.2 Conclusie 24

5 Toepassing op de case study Delft Zuid **25**
5.1 Genereren van ritverzoeken 26
5.2 Aanpassingen voor toepassing op grote schaal 27
5.3 Resultaten en discussie 29
5.4 Conclusie 32

Bibliografie **32**

Hoofdstuk 1

Inleiding

In de laatste tien jaar is de ontwikkeling van de technologie voor het automatiseren van auto's zeer toegenomen. De ontwikkeling van zelfsturende auto's is al zo ver, dat verschillende voertuigproducenten verwachten in 2019 een zelfsturende auto op de markt te kunnen brengen [BI intelligence, 2016]. Nederland speelt een belangrijke rol in de ontwikkelng van de zelfsturende auto, omdat het het eerste land ter wereld is waar een zelfsturende auto zonder stuur op de openbare weg mag rijden. In 2015 zijn de regels aangepast zodat grootschalige testen op de openbare weg met zelfsturende auto's en vrachtwagens mogelijk werden, en in februari dit jaar is een wet aangenomen zodat zelfsturende voertuigen zonder bemanning ook op de openbare weg kunnen testen [Rijksoverheid, 2015], [Engineers Online, 2017] .

Een zelfsturende auto is in staat om over verschillende soorten wegen te navigeren, met zo min mogelijk menselijke input. In welke mate de auto geautomatiseerd is, is in 6 niveaus te onderscheiden, van 0 (niet automatisch) tot 5 (volledig automatisch). Het gebruik van volledig automatische voertuigen kan ervoor gaan zorgen dat er minder ongelukken gebeuren. Volgens een onderzoek van KPMG zullen zelfsturende auto's voor 2.500 minder dodelijke ongevallen in het verkeer zorgen tussen 2014 en 2030 [BI intelligence, 2016]. Ook kunnen zelfsturende voertuigen efficiënter en dus zuiniger rijden dan conventionele voertuigen.

Er is al veel onderzoek gedaan naar de toepassing van volledig geautomatiseerde auto's, waarbij de voertuigen in de meeste gevallen onderdeel zijn van een taxiservice. Zo heeft Zhang et al. in 2014 een artikel over gedeelde geautomatiseerde voertuigen (shared autonomous vehicles, of SAV's in het Engels) geschreven, waarin een model wordt gepresenteerd voor een carpooltaxisysteem [Zhang et al., 2015]. Een jaar eerder verscheen een artikel over SAV's van Fagnant en Kockelman [Fagnant and Kockelman, 2014]. Om het gebied te modelleren gebruikten ze, net als Zhang, een rooster. Een van de resultaten van dit verslag is dat 1 geautomatiseerd voertuig 11 conventionele voertuigen kan vervangen.
In januari 2017 heeft een onderzoek van het Massachussets Insitute of Technology, onder leiding van Daniela Rus, berekend dat alle taxi's in New York City vervangen kunnen worden door 3.000 zelfsturende auto's [Alonso-Mora et al., 2017]. Momenteel rijden er 14.000 taxi's in de stad om aan de vraag te kunnen voldoen. In dit onderzoek wordt gebruik gemaakt van ridesharing, wat betekent dat de taxi's gebruik maken van een carpoolsysteem waar de passagiers hun ritten moeten delen.

Xiao Liang heeft in 2016 een artikel gepubliceerd over het gebruik van zelfsturende elektrische auto's als taxiservice van of naar een treinstation [Liang et al., 2016]. Met behulp van geheeltallig lineair programmeren wordt bepaald welke ritten door de taxi's worden gereden.

Het gebruik van de zelfsturende auto's is dus beperkt tot kleine afstanden: om passagiers van het station naar hun eindbestemming te brengen, en vice versa. Hierdoor wordt het gebruik van het openbaar vervoer, in het bijzonder de trein, gestimuleerd, aangezien het laatste stuk reizen naar de eindbestemming of het eerste stuk van huis naar het station sneller en gemakkelijker wordt.

Liang heeft zijn model in Xpress gerund, maar het is ook mogelijk om de taxiservice in Matlab te modelleren. Dit verslag behandelt de programmering in Matlab, en beschrijft het model van Liang met kritische kanttekeningen. Het model van Liang wordt eerst toegepast op een kleine schaal, waarbij de interpretatie van het model uitgebreid beschreven wordt en toevoegingen aan de resultaten van Liang worden gemaakt. Daarna worden er enkele aanpassingen aan het model gemaakt om op grote schaal te kunnen worden toegepast. Ook deze resultaten zullen behandeld worden, waarna de conclusie en discussie volgen.

Hoofdstuk 2

Model

De taxiservice zoals omschreven in het artikel van Liang bestaat uit N zones (inclusief station) en T tijdstappen waarin de service aangeboden wordt. De service is het best te vergelijken met taxiservices als Uber, met als belangrijk verschil dat de zelfsturende auto's geen chauffeurs hebben, en dus geen personeelskosten per taxi. De ritverzoeken worden een dag van tevoren bekend verondersteld. Passagiers kunnen zich op elk tijdstip aanbieden op één van de zones, en ze reizen altijd van of naar het treinstation.
Het doel van het programma is om de opbrengst van de taxiservice te maximaliseren. Passagiers betalen voor het gebruik van de taxiservice, en aan de taxiservice zijn kosten verbonden, waarvan de voornaamste onderhoudsgerelateerde en brandstofgerelateerde kosten zijn. De taxi's hebben de keuze om een rit wel of niet te accepteren, afhankelijk van de opbrengsten die de rit zal genereren. Het weigeren van een rit heeft geen negatieve gevolgen voor de taxiservice.
Als de taxi's elektrisch zijn, dan hebben ze na een bepaald aantal kilometer ook de tijd nodig om op te laden. Dit wordt later in hoofdstuk 5 toegepast.

2.1 Modelkeuze

Er zijn twee modelkeuzes mogelijk om te bepalen welke ritten de taxi's accepteren.
In Model 1 is het systeem vrij om verzoeken te accepteren of te weigeren, met als doel de winst van het systeem te optimaliseren. Zodra er een aanbod is, wordt deze direct of wel geholpen, of geweigerd. Dit model is onwenselijk wat betreft klantvriendelijkheid. Het kan namelijk zijn dat de rit van een potentiële passagier wordt geweigerd, waarna vijf minuten later een andere rit vanaf dezelfde zone wel wordt geaccepteerd. De eerste passagier kan zich hierdoor benadeeld voelen, maar dit nadelige effect wordt niet meegenomen in het model.
In Model 2 wordt het wel of niet accepteren van ritten afhankelijk van de zones. Ofwel worden alle ritten in een zone wel geaccepteerd, ofwel alle ritten in die zone worden niet geaccepteerd. Hierdoor wordt het weigeren van een rit vanuit de passagier gezien eerlijker, want als de rit van de eerste passagier op een zekere zone wordt geweigerd, geldt dit ook voor de volgende potentiële passagier op dezelfde zone. De opbrengst van Model 2 is mogelijk wel minder dan van Model 1.
De modellen zijn dus als volgt kort samen te vatten:

- Model 1: Per ritverzoek beslissen of het wel of niet aangenomen wordt.
- Model 2: Per zone beslissen of alle verzoeken wel of niet aangenomen worden.

Aangezien het vanuit Model 1 eenvoudig is om uit te breiden tot Model 2 door het toevoegen van twee extra voorwaarden, zullen we beide situaties behandelen.

2.2 Aannames

Voor het model van Liang worden een aantal aannames gemaakt. Ten eerste worden er alleen ritverzoeken van of naar het station aangeboden, aangezien de service ervoor bedoeld is om passagiers van of naar het station te vervoeren.
Ook worden alle ritverzoeken van tevoren bekend verondersteld. Het systeem bepaalt voorafgaand aan de dag welke ritten er gereden worden, en kan niet tijdens de dag nieuwe ritverzoeken verwerken. Het is echter aannemelijk dat de ritverzoeken van tevoren bekend zijn, omdat reizen met de trein vaak gepland is.
Daarnaast wordt van alle ritten uitgaande vanuit dezelfde zone verondersteld dat ze vanuit het middelpunt van de zone vertrekken, om het model te vereenvoudigen.

Hoofdstuk 3

Probleemstelling

In dit hoofdstuk wordt het optimalisatieprobleem voor de geheeltallige lineaire programmering geformuleerd. Eerst worden de verzamelingen, variabelen en parameters geformuleerd, en vervolgens worden de doelfunctie en de randvoorwaarden gegeven.

3.1 Verzamelingen

De verzameling van alle zones en treinstation is $\mathcal{N} = \{0, 1, \ldots, i, \ldots, N\}$. Het treinstation wordt weergegeven door $i = 0$. De verzameling zones zonder het treinstation is dus $\mathcal{N}' = \{1, \ldots, i, \ldots, N\}$.
$\mathcal{T} = \{0, 1, \ldots, t, \ldots, T\}$ is de verzameling van tijden waarop de ritverzoeken zich aanbieden. Buiten deze tijden kunnen de taxi's zich wel vast verplaatsen richting een ritverzoek of terugrijden naar het station. Deze verplaatsingstijd bestaat uit $\{-\delta_{max} \ldots 0\}$ en $\{T \ldots T + \delta_{max}\}$, waar δ_{max} de maximale reistijd tussen het treinstation en elk ander station is. De totale tijd dat de taxi's onderweg kunnen zijn is dan $\mathcal{T}' = \{-\delta_{max}, \ldots, -1, 0, 1, \ldots, t, \ldots, T, T+1, \ldots, T + \delta_{max}\}$.

3.2 Variabelen

Het model bestaat uit vijf verschillende soorten variabelen.

- x_i: binaire variabelen die aangeven of de taxi's in zone i passagiers accepteren ($x_i = 1$) of weigeren ($x_i = 0$), $\forall i \in \mathcal{N}'$.

- $D_{ij}^{t,t+\delta_{ij}}$: de ritverzoeken die geaccepteerd worden vanaf zone i naar zone j op tijdstip t tot tijdstip $t + \delta_{ij}, \forall i, j \in \mathcal{N}, i \neq j, \forall t \in \mathcal{T}, t + \delta_{ij} \leq T$. Er geldt $i = 0 \vee j = 0$, aangezien alle aangeboden ritten van of naar het station gaan.

- $S_i^{t,t+1}$: de variabelen die het aantal taxi's dat stil staat op zone i van tijdstip t tot $t+1$ aangeven, $\forall i \in \mathcal{N}, \forall t \in \mathcal{T}', t + 1 \leq T + \delta_{max}$.

- $U_{ij}^{t,t+\delta_{ij}}$: het aantal verplaatsingen van de taxi's van zone i naar j op tijdsinterval $[t, t + \delta_{ij}], \forall i, j \in \mathcal{N}, \forall t \in \mathcal{T}', t + \delta_{ij} \leq T + \delta_{max}$.

- V_i^t: het aantal taxi's op zone i op tijdstip $t, \forall i \in \mathcal{N}, \forall t \in \mathcal{T}'$.

Om in het vervolg de notatie van de variabelen overzichtelijk te houden, nemen we aan dat voor $t+1$ en $t+\delta_{ij}$ altijd geldt dat ze binnen de verzameling $\mathcal{T}$ vallen als $t \in \mathcal{T}$ en binnen de verzameling $\mathcal{T}'$ als $t \in \mathcal{T}'$. Dit betekent dus:

$$\begin{cases} t+1, t+\delta_{ij} \leq T + \delta_{max} & \text{als } t \in \mathcal{T}' \\ t+1, t+\delta_{ij} \leq T & \text{als } t \in \mathcal{T} \end{cases}$$

Ook nemen we aan dat $i \neq j$ in alle situaties, aangezien $i = j$ de situatie voorstelt waar het vertrekpunt van de taxi hetzelfde is als de bestemming. De reistijd en afstand hiervan zijn beiden 0 en dus is deze situatie triviaal.

3.3 Parameters

De parameters voor het programma zijn als volgt:
F: Het totale aantal taxi's.
δ_{ij}: De reistijd van zone i naar $j, \forall i, j \in \mathcal{N}$
δ_{max}: De maximale reistijd vanaf het station naar elke andere zone.
d_{ij}: De afstand tussen zone i en $j, \forall i, j \in \mathcal{N}$.
$Q_{ij}^{t,t+\delta_{ij}}$: Het aantal ritverzoeken vanaf zone i naar zone j vanaf tijdstip t tot $t+\delta_{ij}$, $\forall i, j \in \mathcal{N},\ i = 0 \vee j = 0, \forall t \in \mathcal{T}$.
P: Prijs per ritafstand.
C_{m1}: Reparatiekosten voor de voertuigen per ritafstand.
C_{m2}: Reparatiekosten voor de parkeerplekken op het station per plek per dag.
C_d: Afschijvingskosten per auto per dag.
C_p: Parkeerkosten bij de zones per plek per tijdstap.

Het model kan uitgebreid worden met een voorwaarde op de batterijduur van de elektrische taxi's. Hiervoor zijn twee aanvullende parameters nodig:
R: Batterijduur uitgedrukt in kilometers te rijden met een volle accu.
E: Afstand die afgelegd kan worden als de batterij in één tijdstap opgeladen kan worden.

3.4 Doelfunctie

De doelfunctie bepaalt de maximum totale opbrengst, door de kosten en opbrengsten per geaccepteerde rit in acht te nemen. Dit brengt de volgende formule tot stand:

$$\begin{aligned} Max\ \Pi = P \cdot \left(\sum_{i \in \mathcal{N}',\ t \in \mathcal{T}} D_{0i}^{t,t+\delta_{0i}} \cdot d_{0i} + \sum_{i \in \mathcal{N}',\ t \in \mathcal{T}} D_{i0}^{t,t+\delta_{i0}} \cdot d_{i0} \right) - \\ C_{m1} \cdot \sum_{i,j \in \mathcal{N},\ t \in \mathcal{T}'} U_{ij}^{t,t+\delta_{ij}} \cdot d_{ij} - C_p \cdot \sum_{i \in \mathcal{N}',\ t \in \mathcal{T}'} S_i^{t,t+1} \end{aligned} \tag{3.1}$$

De eerste regel van de vergelijking bepaalt de winst, afhankelijk van P en het aantal geaccepteerde ritten en de ritafstand. Op de tweede regel worden de variabele kosten berekend. De eerste factor bepaalt de reparatiekosten, afhankelijk van het aantal ritten en de ritafstand. Daarnaast zijn er parkeerkosten bij de zones, afhankelijk van de tijd dat de taxi's daar stilstaan.

Naast de variabele parkeerkosten zijn er ook vaste kosten: de afschrijvingskosten en de reparatiekosten voor de parkeerplekken op het station. Hierbij wordt er vanuit gegaan dat het maximum aantal parkeerplekken dat op het station nodig is, gelijk is aan het totale aantal taxi's. Deze constanten zijn overbodig voor het maximaliseren, maar voor de berekening van de uiteindelijke winst zijn ze wel belangrijk. De winst is als volgt te berekenen:

$$Winst = \Pi - F \cdot (C_d + C_{m2}) \tag{3.2}$$

3.5 Voorwaarden

De eerste voorwaarden die op de doelfunctie worden toegepast, bepalen de positie van de taxi's op het allereerste en allerlaatste tijdstip. De taxi's beginnen op het station ($i = 0$) en zijn nergens anders:

$$V_0^{-\delta_{max}} = F \tag{3.3}$$

$$V_i^{-\delta_{max}} = 0 \quad \forall i \in \mathcal{N}' \tag{3.4}$$

De taxi's eindigen ook allemaal op het station:

$$V_0^{T+\delta_{max}} = F \tag{3.5}$$

$$V_i^{T+\delta_{max}} = 0 \quad \forall i \in \mathcal{N}' \tag{3.6}$$

De volgende voorwaarde zorgt ervoor dat de hoeveelheid taxi's die stilstaan op zone i op tijdsinterval $[t, t+1]$ gelijk is aan het aantal taxi's dat er stond op interval $[t-1, t]$, plus het aantal taxi's dat binnenrijdt op zone i op tijdstip t, min het aantal taxi's dat wegrijdt van zone i op tijdstip t:

$$S_i^{t,t+1} = S_i^{t-1,t} + \sum_{j \in \mathcal{N}} U_{ji}^{t-\delta_{ji},t} - \sum_{j \in \mathcal{N}} U_{ij}^{t,t+\delta_{ij}} \quad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T}' \tag{3.7}$$

Een soortgelijke voorwaarde geldt voor het aantal beschikbare taxi's op zone i op tijdstip t+1, namelijk dat dit gelijk is aan de hoeveelheid beschikbare auto's op tijdstip t, min het aantal taxi's dat wegrijdt van zone i op tijdstip t, plus het aantal taxi's dat aankomt op zone i op tijdstip t+1:

$$V_i^{t+1} = V_i^t + \sum_{j \in \mathcal{N}} U_{ij}^{t,t+\delta_{ij}} - \sum_{j \in \mathcal{N}} U_{ji}^{t+1-\delta_{ji},t+1} \quad \forall i \in \mathcal{N}, \forall t \in \mathcal{T}' \tag{3.8}$$

Er is een voorwaarde nodig die ervoor zorgt dat het aantal geaccepteerde ritten vanaf het treinstation naar zone i vanaf tijdstip t tot tijdstip $t + \delta_{0i}$, kleiner is dan het aantal ritverzoeken vanaf dezelfde zone en hetzelfde tijdstip. Eenzelfde soort voorwaarde geldt voor het aantal geaccepteerde ritten vanaf zone i naar het treinstation van tijdstip t tot tijdstip $t + \delta_{i0}$:

$$D_{0i}^{t,t+\delta_{0i}} \leq Q_{0i}^{t,t+\delta_{0i}} \cdot x_i \quad \forall i \in \mathcal{N}',\ \forall t \in \mathcal{T} \tag{3.9}$$

$$D_{i0}^{t,t+\delta_{i0}} \leq Q_{i0}^{t,t+\delta_{i0}} \cdot x_i \quad \forall i \in \mathcal{N}',\ \forall t \in \mathcal{T} \tag{3.10}$$

Vervolgens moet gelden dat wanneer een zone niet bediend kan worden, er geen taxi's van of naar deze zone rijden:

$$\sum_{j \in \mathcal{N},\ t \in \mathcal{T}'} U_{ij}^{t,t+\delta_{ij}} + \sum_{j \in \mathcal{N}, t \in \mathcal{T}'} U_{ji}^{t,t+\delta_{ji}} \leq M \cdot x_i \quad \forall i \in \mathcal{N}' \tag{3.11}$$

waarbij M een groot getal is. De voorwaarde waarmee bepaald wordt of een zone bediend wordt, is als volgt:

$$x_i \leq \sum_{t \in \mathcal{T}} D_{0i}^{t,t+\delta_{0i}} + \sum_{t \in \mathcal{T}} D_{i0}^{t,t+\delta i0} \quad \forall i \in \mathcal{N}' \tag{3.12}$$

Als er geen geaccepteerde ritten zijn, moet dan gelden dat $x_i = 0$. Voorwaarde (3.13) zorgt ervoor dat het aantal geaccepteerde ritten op tijdstip t naar zone i kleiner of gelijk is aan het aantal taxi's wat op hetzelfde tijdstip en dezelfde zone rijden.

$$D_{0i}^{t,t+\delta_{0i}} \leq U_{0i}^{t,t+\delta_{0i}} \ \forall i \in \mathcal{N}' \ \forall t \in \mathcal{T} \tag{3.13}$$

Eenzelfde soort voorwaarde geldt voor geaccepteerde ritten vanaf zone i naar het station vanaf tijdstip t.

$$D_{i0}^{t,t+\delta_{i0}} \leq U_{i0}^{t,t+\delta_{i0}} \ \forall i \in \mathcal{N}' \ \forall t \in \mathcal{T} \tag{3.14}$$

Voorwaarde (3.15) is nodig om ervoor te zorgen dat het aantal taxi's dat zone i op tijdstip t verlaat, kleiner of gelijk is aan het aantal beschikbare taxi's op die zone op dat tijdstip:

$$\sum_{j \in \mathcal{N}} U_{ij}^{t,t+\delta_{ij}} \leq V_i^t, \ \forall i \in \mathcal{N} \ \ \forall t \in \mathcal{T}' \tag{3.15}$$

Met de voorgaande voorwaarden kunnen we het probleem voor Model 1 (zonder voorwaarden op de batterijduur) oplossen.
Voor Model 2 hebben we twee extra voorwaarden nodig, zodat als een zone wordt bediend, dat alle ritten dan worden geaccepteerd:

$$D_{0i}^{t,t+\delta_{0i}} \geq Q_{0i}^{t,t+\delta_{0i}} + M \cdot (x_i - 1) \ \ \forall i \in \mathcal{N}', \ \forall t \in \mathcal{T} \tag{3.16}$$

$$D_{i0}^{t,t+\delta_{i0}} \geq Q_{i0}^{t,t+\delta_{i0}} + M \cdot (x_i - 1) \ \ \forall i \in \mathcal{N}', \ \forall t \in \mathcal{T} \tag{3.17}$$

Om rekening te houden met de batterijduur van een elektrische auto, wordt de volgende voorwaarde gebruikt:

$$L_t - R \leq \sum_{t_1 \in \mathcal{T}'} S_0^{t_1,t_1+1} \cdot E/F \ \ \forall t \in \mathcal{T}' \tag{3.18}$$

waarbij

$$L_t = \sum_{i,j \in \mathcal{N}, t_1 \in \mathcal{T}'} U_{ij}^{t_1,t_1+\delta_{ij}} \cdot d_{ij}/F \ \ \forall t \in \mathcal{T}' \tag{3.19}$$

Hierdoor wordt de gemiddelde afstand die taxi's hebben afgelegd berekend. Als dit groter wordt dan R, moeten de taxi's op het treinstation blijven om op te laden.

De domeinen van de variabelen zijn als volgt:

$$V_i^t \geq 0 \qquad \forall i \in \mathcal{N}, \ \forall t \in \mathcal{T}' \tag{3.20}$$

$$S_i^{t,t+1} \geq 0 \qquad \forall i \in \mathcal{N}, \forall t \in \mathcal{T}' \tag{3.21}$$

$$D_{ij}^{t,t+\delta_{ij}} \geq 0 \qquad \forall i,j \in \mathcal{N}, \ i = 0 \vee j = 0, \ \forall t \in \mathcal{T} \tag{3.22}$$

$$U_{ij}^{t,t+\delta_{ij}} \geq 0 \qquad \forall i,j \in \mathcal{N}, \forall t \in \mathcal{T}' \tag{3.23}$$

$$x_i \in \{0,1\} \qquad \forall i \in \mathcal{N}' \tag{3.24}$$

Er geldt voor elk van de variabelen dat ze geheeltallig zijn. Het aantal taxi's, het aantal zones en het aantal ritten bestaan immers uit gehele, positieve getallen.

In het artikel van Liang et al. wordt nog een voorwaarde genoemd, namelijk dat $V_0^t \leq Z_0 \ \ \forall t \in T'$, waarbij Z_0 een constante is die het aantal parkeerplekken op het station weergeeft. Deze voorwaarde is echter overbodig als $Z_0 = F$ wordt aangenomen. Dit komt omdat $V_0^{-\delta_{max}} = F$ volgens voorwaarde (3.3), terwijl volgens Liang's voorwaarde ook geldt $V_0^{-\delta_{max}} \leq Z_0$. Er moet dus wel gelden $Z_0 \geq F$, en om de winst zo hoog mogelijk te houden, moet Z_0 zo klein mogelijk gekozen worden.

3.6 Overige berekeningen

Om de resultaten van het model te kunnen interpreteren, kunnen extra berekeningen gemaakt worden. Dit zijn geen voorwaarden, maar kunnen juist na het runnen van het model berekend worden. Ten eerste kan het percentage geaccepteerde ritten θ berekend worden:

$$\theta = \left(\sum_{i \in N', t \in T} D_{0i}^{t,t+\delta_{0i}} + D_{i0}^{t,t+\delta_{i0}} \right) / \left(\sum_{i \in N', t \in T} Q_{0i}^{t,t+\delta_{0i}} + Q_{i0}^{t,t+\delta_{i0}} \right)$$

Daarnaast kan ook de totale parkeertijd per zone σ_i berekend worden:

$$\sigma_i = \sum_{t \in T'} S_i^{t,t+1} \quad \forall i \in N'$$

Ook de winst en de gemiddelde afstand die taxi's hebben afgelegd kunnen achteraf berekend worden, deze zijn in (3.2) en (3.19) weergegeven.

3.7 Programmeren in Matlab

Om het model in Matlab te kunnen programmeren, wordt de functie intlinprog($f, intcon, A, b, A_{eq}, b_{eq}, lb, ub$) gebruikt. Deze functie maakt gebruik van de notatie die vaak bij lineair programmeren gezien wordt:

$$\text{minimaliseer } x \text{ in } \{f^T x\}$$

onder de voorwaarden:

$$Ax \leq b \qquad \text{(ongelijkheidsvoorwaarden)} \tag{3.25}$$

$$A_{eq}x = b_{eq} \qquad \text{(gelijkheidsvoorwaarden)} \tag{3.26}$$

$$lb \leq x \leq ub \qquad \text{(randvoorwaarden)} \tag{3.27}$$

Aangezien het hier om geheeltallig lineair programmeren gaat, geldt ook de volgende voorwaarde voor ten minste 1 element van x:

$$x_i \in \mathbb{Z}$$

De invoer voor f is gelijk aan de coëfficiënten van de variabelen in de doelfunctie. Aangezien intlinprog() altijd met minimalisatieproblemen werkt, en de doelfunctie van de probleemstelling het maximaliseren van de opbrengst is, is het nodig om $-f$ in te voeren.

Bij *intcon* is de input de index van de elementen van x welke geheeltallig zijn. Voor de probleemstelling zoals hierboven beschreven geldt dus dat intcon alle variabelen bevat, aangezien deze allemaal geheeltallig zijn.

Er geldt dat elke rij van A de coëfficiënten van de variabelen van een gelijkheidsvoorwaarde bevat. Neem bijvoorbeeld voorwaarde (3.3), dan geeft de bijbehorende rij in matrix A een 1 voor $V_0^{-\delta_{max}}$, en een 0 voor alle overige variabelen. De invoer voor elk element van b is gelijk aan de constanten van een gelijkheidsvoorwaarde. Voor dezelfde voorwaarde (3.3), betekent dit $b_{3.3} = F$. Het kan ook zijn dat een element van b gelijk is aan 0, zoals bij voorwaarde (3.4). Voor A_{eq} en b_{eq} geldt ongeveer hetzelfde: elke rij van de matrix A_{eq} bevat de coëfficiënten van de variabelen van een ongelijkheidsvoorwaarde, en b_{eq} de bijbehorende constante.

lb is de ondergrens en ub de bovengrens van x. Voor deze probleemstelling geldt dat de ondergrens voor elke variabele gelijk is aan 0, en de bovengrens gelijk is aan oneindig voor alle variabelen, behalve voor x_i, $\forall i \in N$. In die gevallen is de bovengrens gelijk aan 1. Doordat we aannemen dat $x_i \in \mathbb{Z}$, moet gelden dat x_i binaire variabelen zijn.

Door gebruik te maken van de simplex methode kan het programma de optimale uitkomst bepalen. De output bestaat uit de oplossingswaarde, en daarnaast ook de oplossingsvector x, die aangeeft welke waarden de variabelen aannemen om de optimale waarde te bereiken.

Hoofdstuk 4

Toepassing op kleine schaal

Ter illustratie wordt het model eerst toegepast op kleine schaal. Er zijn naast het station nog 4 andere zones, en de ritverzoeken vinden plaats in 9 tijdstappen, van 0 tot 8. De afstanden en reistijden zijn weergegeven in respectievelijk Tabel 4.1 en Tabel 4.2. De verzoeken zijn vast gekozen, en zijn te zien in Tabel 4.3. Uit Tabel 4.2 is te zien dat de maximale reistijd vanaf het station tot elke andere zone gelijk is aan 3 tijdstappen, dus is de totale tijd dat de taxi's actief kunnen zijn gelijk aan 15 tijdstappen. Er geldt dus $T' = \{-3, \ldots, 0, \ldots, 11\}$.
De parameters worden als volgt gekozen:

$$
\begin{aligned}
P &= 1,00 && \text{€/km} \\
C_{m1} &= 0,05 && \text{€/km} \\
C_d &= 17,00 && \text{€/dag} \\
C_{m2} &= 5,00 && \text{€/plek} \times \text{dag} \\
C_p &= 0,25 && \text{€/plek} \times \text{tijdstap} \\
F &= 5
\end{aligned}
$$

De oplossingen die voor deze situatie gegenereerd worden, zijn beschreven in de volgende paragraaf.

Tabel 4.1: Reisafstanden

Zones	Reisafstand (km)				
	0	1	2	3	4
0	-	2	3	4	6
1	2	-	4	2	6
2	3	4	-	2	3
3	4	2	2	-	4
4	6	6	3	4	-

Tabel 4.2: Reistijd

Zones	Reistijd (tijdstap)				
	0	1	2	3	4
0	-	1	1	2	3
1	1	-	2	1	3
2	1	2	-	1	2
3	2	1	1	-	2
4	3	3	2	2	-

Tabel 4.3: Ritverzoeken voor toepassing op kleine schaal

Verzoeken vanaf station			Verzoeken naar station		
Begintijd	Bestemming	Aantal verzoeken	Begintijd	Oorsprong	Aantal verzoeken
0	4	1	0	1	1
0	2	1	2	3	1
2	3	1	2	4	1
3	2	1	3	2	1
4	1	1	4	4	1
6	3	1	6	1	1
6	2	1			

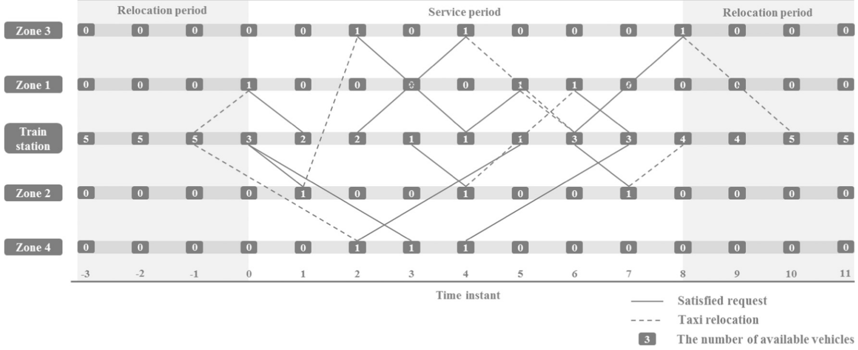
4.1 Resultaten en discussie

In het artikel van Liang wordt voor deze situatie de oplossing gegeven in Figuur 4.1 voor Model 1 en in Figuur 4.3 voor Model 2. De doorgetrokken lijnen zijn de ritverzoeken die geaccepteerd zijn, en de stippellijnen zijn de andere verplaatsingen van de taxi's. De getallen op de zones zijn het aantal taxi's dat op dat moment beschikbaar is, oftewel V_i^t. De oplossing zoals weergegeven door Liang et al. is echter geen unieke oplossing, omdat er voor veel afstanden geldt dat de directe rit even lang duurt als een route die ten minste via 1 ander punt rijdt. Bijvoorbeeld, van station naar zone 4 rijden is net zo snel als via zone 2 van station naar zone 4 rijden. Het resultaat van het Matlab programma voor Model 1 is te zien in Figuur 4.2. Te zien is dat er inderdaad een andere oplossing is met dezelfde waarden. Twee keer verplaatst de taxi zich via een tussenstop: van het treinstation naar zone 4 vanaf tijdstip -1, en van zone 2 naar zone 1 vanaf tijdstip 4. Voor Model 2 geeft Matlab wel precies hetzelfde resultaat als Liang.

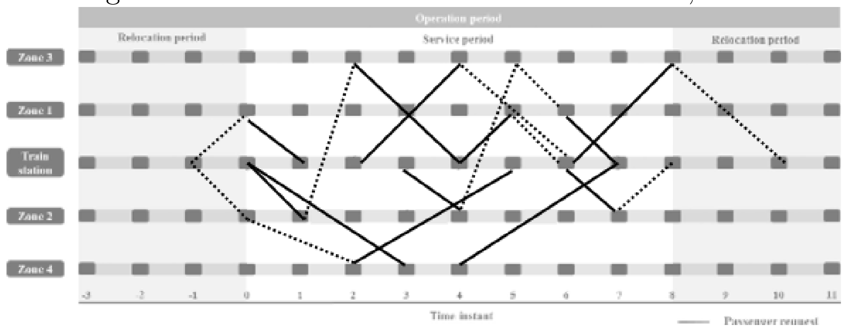
Aan de figuren is te zien dat er in Model 1 twee ritten meer geaccepteerd worden. Dit is te verwachten, aangezien er een minder restricties op het accepteren van ritten zijn dan in Model 2. Bij Model 2 is te zien dat zone 3 niet geaccepteerd wordt, en alle andere zones wel. Het gemiddelde aantal gereden kilometers is voor Model 1 gelijk aan 14,4 km, en voor Model 2 is het 11,8 km.

Liang behandelt van dit kleine voorbeeld alleen de resultaten zoals gegeven in Figuur 4.1 en Figuur 4.3, maar beschrijft niet wat de opbrengst of het percentage geaccepteerde ritten is. Toch is het voor de interpretatie van deze situatie ook belangrijk om ook naar deze factoren te

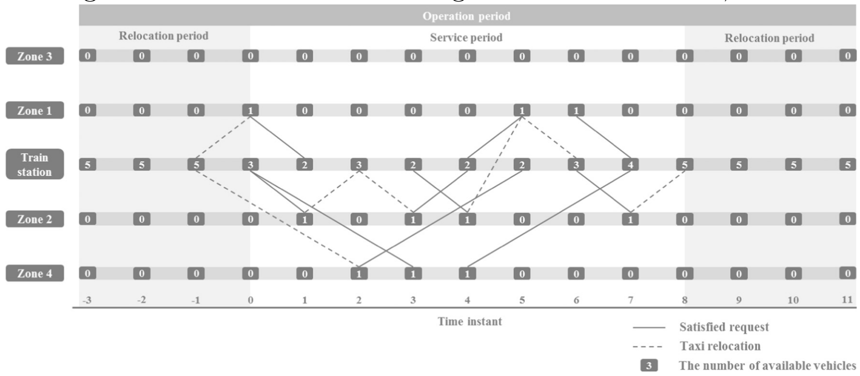
Figuur 4.1: Resultaat van Liang et al. voor Model 1, $F = 5$



Figuur 4.2: Resultaat van Matlab voor Model 1, $F = 5$



Figuur 4.3: Resultaat van Liang et al. voor Model 2, $F = 5$



kijken. In dit voorbeeld is het aantal taxi's ten opzichte van het aantal ritverzoeken erg hoog. Dit zorgt voor veel vaste kosten per taxi, waardoor er in elke situatie verlies wordt gemaakt. Deze resultaten zijn te zien in Tabel 4.4. De oplossing voor Model 1 heeft een verlies van €68,85, en voor Model 2 een verlies van €77,20. Het percentage ritten dat wordt geaccepteerd is respectievelijk 92,3% en 76,9%. Om een positief resultaat te behalen, moet de ritprijs worden verhoogd naar ongeveer €2,60 per kilometer voor Model 1, en naar €3,20 voor Model 2. Omdat Model 2 wel een eerlijker systeem heeft voor het weigeren of accepteren van ritten, is het in deze situatie logisch om toch voor Model 2 te kiezen.

Omdat het aantal taxi's ten opzichte van het aantal ritverzoeken hoog is, is het interessant om naar een situatie te kijken waarin het aantal taxi's verminderd is. Als het aantal taxi's 4 is, geldt voor Model 1 dat het nog maar €49,60 verlies maakt, met een ritpercentage van 84,6% en 16,75 gemiddeld gereden kilometers.
Voor Model 2 wordt dit een verlies van €55,20 en een ritpercentage van 76,9% en 14,75 gemiddeld gereden kilometers. Opmerkelijk is dat het percentage voor Model 2 gelijk is gebleven, en de

Tabel 4.4: Resultaten voor toepassing op kleine schaal

	Model 1			Model 2		
F	Winst	θ	$L_{T+\delta_{max}}$	Winst	θ	$L_{T+\delta_{max}}$
5	€-68,85	92,3%	14,4 km	€-77,20	76,9%	11,8 km
4	€-49,60	84,6%	16,75 km	€-55,20	76,9%	14,75 km
3	€-34,80	62,0%	17,00 km	€-44,05	46,0%	12,00 km

ritten van de taxi's zijn ook precies gelijk aan die in Figuur 4.3. Dit betekent dat de vijfde taxi voor Model 2 overbodig is. Om in deze situatie winst te maken, moet de prijs per kilometer ook verhoogd worden naar €2,20 voor Model 1 en naar €2,60 Model 2. Het is nu dus nog beter om voor Model 2 te kiezen, omdat het verschil in geaccepteerde ritten tussen de twee modellen kleiner is.
Aangezien er met 4 taxi's nog steeds verlies gemaakt wordt, is in Tabel 4.4 ook het resultaat van de situatie met 3 taxi's te zien. Nu is het percentage geaccepteerde ritten voor Model 2 wel verminderd, omdat er nu twee zones zijn die niet meer geaccepteerd worden. Het gemiddeld aantal gereden kilometers is daardoor ook minder dan wanneer er 4 taxi's gebruikt worden. Aangezien het aantal geaccepteerde ritten minder dan de helft is en er bijna €10,- meer aan verlies gemaakt wordt dan bij Model 1, is de keuze voor Model 1 in deze situatie beter. Voor Model 1 is het verlies minder, maar ook hier is het ritpercentage met meer dan 20% gedaald.

4.2 Conclusie

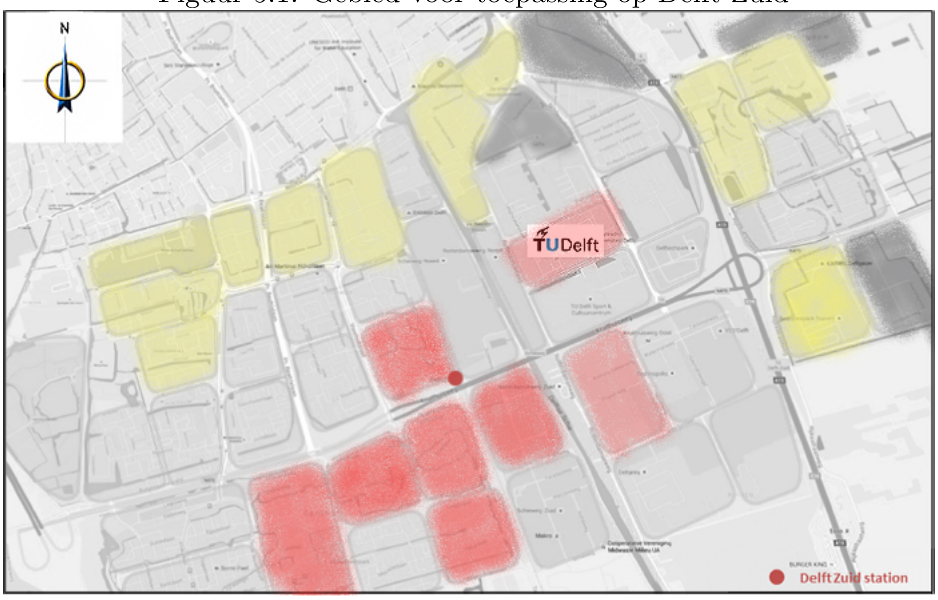
Door de toepassing op kleine schaal is de hoeveelheid taxi's per zone erg hoog, wat zorgt voor een verlies in alle situaties die in de vorige paragraaf zijn behandeld. Voor Model 2 is de hoeveelheid ritten die geaccepteerd wordt bij 5 taxi's zelfs gelijk aan de hoeveelheid geaccepteerde ritten bij 4 taxi's. Het aantal geaccepteerde ritten is bij de situatie met maar 3 taxi's voor beide modellen sterk gedaald, waardoor de keuze voor 3 taxi's niet klantvriendelijk meer is. De beste optie is om 4 taxi's te gebruiken en de ritprijs te verhogen zodat er wel winst gemaakt wordt.

Hoofdstuk 5

Toepassing op de case study Delft Zuid

Om het model naar een realistische situatie te schalen, wordt in het artikel van Liang het station Delft Zuid gebruikt als case study. Het gebied om het station heen wordt opgedeeld in 48 zones, elk ongeveer 500 x 500 meter groot. Uit het artikel blijkt dat in in ieder geval 4 zones nooit geaccepteerd worden, en dus kunnen we het aantal zones al verkleinen naar 44. Het station ligt vlakbij de campus van de Technische Universiteit Delft, maar er rijdt maar 1 bus richting het centrum. De invoer van de taxiservice heeft dus grote toegevoegde waarde, aangezien de studenten en professoren die buiten Delft wonen zo makkelijker naar de campus kunnen rijden. De service kan daarnaast gebruikt worden voor de omliggende wijken van Delft. Het gebied wordt weergegeven in Figuur 5.1, waar de donkergrijze zones de gebieden aangeven die we buiten beschouwing laten.

Figuur 5.1: Gebied voor toepassing op Delft Zuid



Voor het bepalen van de reistijden en -afstanden tussen alle 44 zones, is gebruik gemaakt van Google Maps. Vanuit het midden van elke zone is steeds de afstand in kilometer en reistijd in aantal tijdstappen naar elke andere zone genoteerd. Dit resulteert in twee symmetrische

44 x 44 matrices.
De grootte van de tijdstappen is 5 minuten, en de periode waarin de taxi's hun service verlenen duurt 193 tijdstappen. Dit betekent dus dat de service 16 uur beschikbaar is, van 6:00 tot 22:00. De grootste reistijd vanaf het station naar elk omliggende zone is gelijk aan 3, dus de totale tijd waarin de taxi's onderweg kunnen zijn is 199 tijdstappen.
De tijdstap van 5 minuten lijkt klein, maar voor het gebied waarin het model wordt toegepast zijn de meeste reistijden kleiner dan 10 minuten. Daarnaast is de grootste reistijd gelijk aan 16 minuten, en dus 3 tijdstappen. De nauwkeurigheid van het model wat betreft reistijden is dus niet heel groot. Maar aangezien er ook tijd nodig is voor het in- en uitstappen van de passagiers, is de aanname dat 5 minuten de kleinst mogelijke tijdstap is, realistisch. Daarnaast is het voor het vereenvoudigen van het model ook wel noodzakelijk om tijdstappen van 5 minuten te gebruiken. Zou een tijdstap van 2 minuten gebruikt worden, dan wordt het totale aantal tijdstappen 2,5 keer zo groot, en dus zou het aantal variabelen van de vorm $U_{i,j}^{t,t+\delta_{i,j}}$ ook ongeveer 2,5 keer zo veel worden. De hoeveelheid geheugen die in totaal gebruikt wordt, wordt dan meer dan verdubbeld.

Voor de parameters worden dezelfde waarden gekozen als in het vorige hoofdstuk. Omdat de taxi's nu meer kilometers moeten afleggen, wordt er ook rekening gehouden met de batterijduur en oplaadtijd van de batterij van de taxi:

$$R = 80,0 \qquad \text{km/volle accu} \tag{5.1}$$
$$E = 1,67 \qquad \text{km/tijdstap opladen} \tag{5.2}$$

In paragraaf 5.3 zullen de resultaten van elektrische en niet-elektrische taxi's met elkaar vergeleken worden.

5.1 Genereren van ritverzoeken

Voor het genereren van ritverzoeken kunnen we de data van het artikel niet gebruiken, aangezien deze uit een enquête zijn gekomen en deze informatie verder onbekend is. Ook wordt er in het artikel rekening gehouden met de aankomst- en vertrektijden van treinen. Omdat er 4 treinen per uur vanaf Delft Zuid vertrekken, kan de invloed van de aankomst- en vertrektreinen verwaarloosd worden voor de nieuwe toepassing. Om de ritten wel op een natuurlijke manier te genereren, gebruiken we een Poisson distributie met $\lambda \leq 1$ voor elke zone en elke tijdstap, zodat er vaak geen rit of 1 rit wordt gegenereerd. Met de functie poissrnd() kunnen deze waarden worden gevonden.

Door λ te variëren wordt de totale hoeveelheid ritten anders. Met een waarde $\lambda = 0,2$ worden er gemiddeld per zone 40 ritverzoeken gevonden, voor $\lambda = 0,3$ is het gemiddelde 60 per zone. Om het model realistischer te maken, worden aan gebieden waar meer ritverzoeken worden verwacht, hogere waarden voor λ gekozen. Uit het artikel van Liang blijkt dat de ritverzoeken variëren van minder dan 10 in de zones aan de randen van het gebied, tot meer dan 150 bij de wijken dicht bij het station en het terrein van de Technische Universiteit Delft. Ook wordt er rekening gehouden met de piektijden van de spits. De ochtendspits, die plaatsvindt van 7:00 tot 9:00, bevat tijdstappen $t \in (12, 36) \subset \mathcal{T}$. De avondspits vindt plaats van 16:00 tot 18:00 en bevat de tijdstappen $t \in (120, 144) \subset \mathcal{T}$.

We kiezen voor alle tijdstappen en zones $\lambda = 0,09$ als basis. Bij de zones met een grotere vraag wordt een factor bij λ opgeteld, en zo ook bij de tijdstappen waarop de spits plaatsvindt. Bij de zones met een lagere vraag wordt er een factor van λ afgetrokken. Dit wordt zodanig gekozen dat het totaal aantal ritten dat wordt aangeboden rond de 2061 ligt. Dit aantal is volgens de enquête van Liang een goede schatting voor de hoeveelheid gebruikers van de taxiservice per dag. De factor voor de spits is $\lambda_{spits} = 0,105$, de factor voor de zones waar veel vraag is, is $\lambda_{hoog} = 0,14$ en bij weinig vraag geldt $\lambda_{laag} = -0,08$. Deze zones zijn te zien in Figuur 5.1, waar geel de zones met lage vraag aanduidt, en rood de zones met hoge vraag. Voor een zone met hoge vraag in de spits wordt λ dan 0,335 en voor een zone met weinig vraag buiten de spits geldt $\lambda = 0,01$.

Het aantal verzoeken dat door Matlab gegenereerd wordt en het daadwerkelijke aantal verzoeken $Q_{i0}^{t,t+\delta_{i0}}$ en $Q_{0i}^{t,t+\delta_{0i}}$ verschilt, doordat het programma alle waarden $\forall i \in \mathcal{N}, \forall t \in \mathcal{T}, t+1 \leq T$ neemt, zonder de voorwaarden $t+\delta_{i0} \leq T$ of $t+\delta_{i0} \leq T$. Dit geldt alleen voor het tellen van het aantal verzoeken en heeft verder geen invloed op de rest van het programma, omdat er bij de voorwaarden altijd alleen naar de elementen gekeken wordt die wel voldoen aan $t+\delta_{0i} \leq T$ of $t+\delta_{i0} \leq T$. Bovendien is dit verschil verwaarloosbaar. Dit komt doordat de meeste ritten een reistijd van 1 tijdstap hebben, waardoor het aantal extra punten minder dan 1% van het totale aantal is. Bovendien wordt niet op elk punt een ritverzoek gegenereerd. Het aanpassen van dit verschil zou veel rekentijd in beslag nemen, omdat er dan een loop over alle ritverzoeken nodig is. Door dit verschil is het echter vrijwel onmogelijk om $\theta = 100\%$ als resultaat te krijgen. Dit wordt uitgelegd in paragraaf 5.3.

5.2 Aanpassingen voor toepassing op grote schaal

Voor de toepassing op grote schaal zijn de hoeveelheid geheugen die gebruikt moet worden en de tijd die nodig is om het programma te runnen, veel groter. Als er daarom manieren zijn om het gebruikte geheugen te verkleinen of de duur van het programma in te korten, moet hier zeker gebruik gemaakt van worden. Eerst wordt besproken hoe er minder geheugen gebruikt kan worden, en vervolgens hoe de snelheid van het programma verbeterd kan worden.

De grootte van de doelfunctie is gelijk aan het aantal variabelen, welke afhangt van de waarden van T en N. Het aantal variabelen van de vorm $U_{i,j}^{t,t+\delta_{i,j}}$ is gelijk aan $N(N-1)(T'-1)$, en is gelijk aan 57% van de totale hoeveelheid variabelen. Voor $N = 45$ en $T' = 199$, geldt dus dat de doelfunctie bijna 400.000 elementen lang is. Matlab heeft echter een limiet op de lengte van vectoren van 10.000. Toch is het mogelijk om Matlab de berekeningen te laten doen, omdat de matrices veelal ijl (sparse) zijn. Door gebruik te maken van sparse() in plaats van zeros(), hoeft Matlab alleen de waarden en indices te onthouden van alle niet-nulelementen. Dit bespaart genoeg ruimte om het programma te kunnen laten runnen.

De duur van het runnen van het programma kan mogelijk ingekort worden door extra voorwaarden toe te voegen die het oplossingsgebied kleiner maken. Voor de toepassing op kleine schaal is dit onnodig, aangezien de runtime hier niet langer dan 2 seconden duurt. Bij deze tweede situatie kan het verkleinen van het oplossingsgebied echter wel effect hebben. De twee extra voorwaarden die Liang voorstelt te gebruiken zijn als volgt:

$$S_i^{t,t+1} \leq F \cdot x_i \quad \forall i \in \mathcal{N}', \ \forall t \in \mathcal{T}' \tag{5.3}$$

$$V_i^t \leq F \cdot x_i \quad \forall i \in \mathcal{N}', \ \forall t \in \mathcal{T}' \tag{5.4}$$

Aan de andere kant kost het toevoegen van voorwaarden tijd en geheugen. Om te testen of de voorwaarden het programma sneller maken, is de doorlooptijd van het gehele programma en van de functie intlinprog() gemeten, zie Tabel 5.1. De tijden zijn gemiddeld genomen uit de doorlooptijden van runs met verschillende aantallen taxi's. Te zien is dat zowel de totale tijd als de tijd van intlinprog() groter zijn wanneer de extra voorwaarden worden toegevoegd en dat bijna alle extra tijd uit intlinprog() komt. Het toevoegen van de voorwaarden heeft dus geen voordelig effect, maar zorgt ervoor dat de functie intlinprog() minder snel tot een optimum kan komen. Een reden hiervoor is dat deze functie zelf ook methodes gebruikt die het oplossingsgebied kleiner maken door middel van het toevoegen van extra voorwaarden. Er wordt bijvoorbeeld gebruik gemaakt van Gomory snedes. Hierdoor is het zelf toevoegen van extra voorwaarden overbodig. De extra voorwaarden worden in het vervolg dus ook niet meer toegepast.

Tabel 5.1: Metingen doorlooptijd

	Totale tijd	Tijd intlinprog()
zonder extra voorwaarden	769 s	83 s
met extra voorwaarden	828 s	133 s

De snelheid van intlinprog() kan wel verhoogd worden door het aanpassen van de opties van het programma. Twee opties die de nauwkeurigheid van het programma beïnvloeden zijn IntegerTolerance en TolGapAbs. De eerste bepaalt voor de variabelen die als geheeltallig zijn aangegeven door *intcon*, de nauwkeurigheid waarop de variabelen afronden tot een geheel getal. Deze waarde is standaard $1 \cdot 10^{-4}$. Om het programma sneller te laten verlopen kan deze waarde worden verhoogd, en om de nauwkeurigheid te verbeteren moet de waarde juist worden verlaagd. Aangezien het programma voor elke situatie 10 keer gerund wordt om de nauwkeurigheid te verbeteren, kan de waarde iets verhoogd worden.

De optie TolGapAbs bepaalt hoe groot het absolute verschil tussen de bovengrens en ondergrens van de oplossing mag zijn. Deze waarde is standaard 0. Aangezien het om een absoluut verschil gaat, moet voor het aanpassen van deze waarde rekening gehouden worden met de oplossingswaarde. Deze ligt meestal tussen de -3000 en -5000, dus als we TolGapAbs een waarde van 5 toekennen, is de relatieve fout hoogstens 0,17%. De nauwkeurigheid van de resultaten wordt verder behandeld in de volgende paragraaf.

5.3 Resultaten en discussie

Het programma is voor beide modellen, voor alle verschillende aantallen taxi's, 10 keer gerund. De resultaten zijn te zien in Tabel 5.2, Tabel 5.3, Tabel 5.4 en Tabel 5.6. Om de nauwkeurigheid van de resultaten aan te duiden is in Tabel 5.3 de variantiecoëfficiënt per aantal taxi's te zien. De opbrengst is gelijk aan $-\Pi$, en de winst is berekend zoals in vergelijking 3.2. De laatste kolom geeft de gemiddeld gereden kilometers per taxi weer.

Voor de resultaten van Model 2 blijkt dat, ondanks dat alle zones gebruikt worden, niet altijd alle verzoeken geaccepteerd worden. Dit komt doordat er meer ritten gegenereerd worden dan daadwerkelijk nodig is (zie paragraaf 5.1), en deze worden meegenomen in de berekening van het aantal verzoeken en het percentage geaccepteerde ritten. Bij 80 en 60 taxi's worden altijd alle ritten geaccepteerd, en is de echte θ dus gelijk aan 100%. Te zien is dat er gemiddeld 11,2 extra verzoeken gegenereerd worden. Dit verschil geldt ook voor de resultaten van Model 1. De waarde van θ van Model 1 in de gevallen van 60 en 80 taxi's is dus ook eigenlijk 100%. De winst, het aantal zones en het aantal kilometer per taxi worden hierdoor niet beïnvloed.

Tabel 5.2: Resultaten voor Model 1, niet elektrisch

F	Opbrengst	Winst	Verzoeken	Geaccepteerd	θ	Zones	km per taxi
10	€3228,52	€3008,52	2065,0	1177,8	57,02 %	43,4	469,86 km
20	€4356,33	€3916,30	2083,1	1808,2	86,81 %	44,0	320,66 km
40	€4660,29	€3780,29	2060,6	2051,7	99,57 %	44,0	172,46 km
60	€4693,65	€3373,65	2062,2	2052,8	99,54 %	44,0	115,82 km

In Tabel 5.2 is te zien dat voor Model 1 met niet elektrische auto's, het optimale aantal taxi's 20 is, omdat de winst in dit geval het hoogst is. De opbrengst is uiteraard groter bij hogere aantallen taxi's, maar de kosten worden veel hoger. In bijna alle gevallen wordt uit alle zones een rit geaccepteerd, wat te verwachten is voor Model 1. Daarom is er voor dit model ook gekeken naar de situatie met 10 taxi's in plaats van 80, omdat bij 40 en 60 taxi's ook al alle ritten gereden worden.
Opvallend is dat het gebruik van elektrische auto's geen significante invloed heeft op de gereden kilometers en op de winst. De resultaten zijn vrijwel gelijk aan de resultaten van niet-elektrische taxi's. De extra voorwaarde 3.18 beïnvloedt het resultaat dus niet of nauwelijks. Dit concludeert Liang ook in zijn verslag: in zijn programma is het verschil alleen bij Model 1 met 20 taxi's merkbaar. In de door Matlab geproduceerde resultaten is dit niet het geval, en is het optimale aantal taxi's voor Model 1 met elektrische taxi's ook gelijk aan 20. Dit betekent dus dat het gebruik van elektrische taxi's voor de service geen belemmering is voor het aantal ritten dat geaccepteerd kan worden, en dat de keuze voor elektrische auto's een mogelijkheid is.

Tabel 5.3: Resultaten voor Model 2, niet elektrisch

F	Opbrengst	Winst	Verzoeken	Geaccepteerd	θ	Zones	km per taxi
40	€4679,03	€3799,03	2067,8	2046	98,93 %	43,5	173,30 km
60	€4652,41	€3332,41	2057,6	2049,2	99,60 %	44	114,66 km
80	€4729,30	€2969,30	2082,0	2068,1	99,33 %	44	87,83 km

In Tabel 5.3 en Tabel 5.6 zijn de resultaten voor Model 2 in de situaties met 40, 60 en 80 taxi's te zien. De winst ligt het hoogst bij 40 taxi's, waarbij ook nog steeds bijna alle ritten gereden worden. Het verschil in aantal zones bij 40 taxi's komt waarschijnlijk door toeval, aangezien de hoeveelheid geaccepteerde ritten en de winst wel ongeveer gelijk is. Het ontbreken van de resultaten met 20 taxi's komt doordat Matlab in deze situatie niet snel genoeg tot een antwoord kan komen. De maximale tijd die de functie intlinprog() mag gebruiken is 2 uur, en binnen deze tijd wordt er geen oplossing gevonden. Dit komt doordat in deze situatie er waarschijnlijk veel minder dan 44 zones gebruikt kunnen worden, en het programma voor elke x_i moet bepalen of het 1 of 0 moet zijn. Er zijn hiervoor heel veel verschillende mogelijkheden en om deze allemaal langs te gaan kost te veel tijd. Voor een vervolgonderzoek zou nog verder gekeken kunnen worden naar het versnellen van de functie, of de maximale tijd van het programma verhogen en lang genoeg wachten om de resultaten alsnog te verkrijgen. Echter, omdat voor de nauwkeurigheid elke situatie 10 keer wordt gerund, zou dit minstens nog 40 uur in beslag nemen (10 keer niet-elektrisch, 10 keer elektrisch, minimaal 2 uur per run). Het zoeken naar een manier om sneller tot een resultaat te komen is waarschijnlijk een beter idee.

Tabel 5.4: Resultaten voor Model 1, elektrische taxi's

F	Opbrengst	Winst	Verzoeken	Geaccepteerd	θ	Zones	km per taxi
20	€4311,63	€3877,03	2065,1	1795,3	86,94%	44	320,17 km
40	€4726,53	€3846,53	2084,1	2074,1	99,52 %	44	174,64 km
60	€4645,05	€3325,05	2048,9	2040,2	99,58 %	44	114,97 km
80	€4685,36	€2914,46	2070,5	2062,6	99,62 %	44	86,41 km

Tabel 5.5: Liang's resultaten voor Model 1, elektrische taxi's

F	Winst	Verzoeken	Geaccepteerd	θ	Zones	km per taxi
20	€2365,30	2061	1242	60,2 %	43	204,8 km
40	€3391,30	2061	1955	94,9 %	44	185,0 km
60	€3128,00	2061	2059	99,9 %	44	129,4 km
80	€2691,20	2061	2061	100 %	44	97,1 km

De resultaten met elektrische taxi van Tabel 5.4 kunnen vergeleken worden met resultaten van Liang die te vinden zijn in Tabel 5.5. De resultaten voor 80 en 60 taxi's zijn vergelijkbaar. Dat het resultaat op alle vlakken overeenkomt voor situaties met veel taxi's is noemenswaardig, omdat de datasets voor d_{ij} en δ_{ij} van Liang onbekend zijn. Dit betekent dat de geschatte waardes sterk overeenkomen met de waarden van Liang.
Echter, het aantal ritten dat Liang's model kan rijden met 40 en vooral 20 taxi's is een stuk minder. Daarnaast is te zien dat bij 20 taxi's, de afgelegde hoeveelheid kilometers per taxi veel kleiner is bij het model van Liang. Dit komt waarschijnlijk omdat bij het model van Liang, de ritten minder homogeen verspreid zijn. Als er veel ritverzoeken op eenzelfde moment zijn, dan is het namelijk in het geval van weinig taxi's niet mogelijk om aan alle ritten te voldoen en worden er veel ritten geweigerd. Hierdoor is ook de winst lager bij de resultaten van Liang voor weinig taxi's. Hetzelfde geldt voor het verschil in aantal gebruikte zones bij Tabel 5.6 en 5.7. Als de ritten minder verspreid zijn over de gehele periode, moet er voor elk piekmoment gekozen worden welke zones worden gereden, en dus ook welke zones helemaal niet meer geaccepteerd kunnen worden.

Tabel 5.6: Resultaten voor Model 2, elektrische taxi's

F	Opbrengst	Winst	Verzoeken	Geaccepteerd	θ	Zones	km per taxi
40	€4697,16	€3817,16	2070,0	2060,3	99,53 %	44	174,11 km
60	€4698,43	€3378,43	2057,8	2054,5	99,57 %	44	115,94 km
80	€4657,14	€2897,14	2054,9	2045,2	99,52 %	44	86,22 km

Tabel 5.7: Liang's resultaten voor Model 2, elektrische taxi's

F	Winst	Verzoeken	Geaccepteerd	θ	Zones	km per taxi
20	€1645,80	2061	913	44,3 %	21	188,2 km
40	€2716,30	2061	1648	80,0 %	35	185,0 km
60	€3078,20	2061	2034	98,7 %	43	128,0 km
80	€2691,20	2061	2061	100 %	44	97,1 km

In Tabel 5.3 is te zien dat de variantiecoëfficiënt op zijn hoogst 3,6% is. Voor de alle variabelen heeft het aantal taxi's weinig invloed op de nauwkeurigheid. Om de variantiecoëfficient voor minder taxi's beter te krijgen, kunnen er meer runs voor deze situaties gedaan worden. Aangezien 10 taxi's alleen bij Model 1 zonder elektrische auto wordt gebruikt, zijn deze coëfficiënten ook minder nauwkeurig.

Tabel 5.8: Variantiecoëfficiënt per aantal taxi's

	Opbrengst	Winst	Verzoeken	Geaccepteerd	θ	Zones	km per taxi
10	2,1 %	2,2 %	2,3 %	3,6 %	1,5 %	1,1 %	1,5 %
20	1,8 %	2,0 %	1,0 %	1,3 %	0,9 %	0,0 %	1,1 %
40	2,2 %	2,7 %	2,0 %	2,2 %	0,4 %	0,6 %	1,9 %
60	2,2 %	3,1 %	1,8 %	1,9 %	0,1 %	0,0 %	2,3 %
80	2,2 %	3,5 %	1,9 %	2,0 %	0,6 %	0,0 %	1,8 %

5.4 Conclusie

Het model geeft ook op grote schaal realistische uitkomsten en de resultaten zijn vergelijkbaar met de resultaten van Liang. Dit geldt voornamelijk voor de situaties waarin alle ritten worden gereden. Het verschil in de modellen ligt vooral bij de verspreiding van de ritverzoeken. Het optimale aantal taxi's is altijd 20 voor Model 1, en 40 voor Model 2. Voor het verbeteren van het model zou dus vooral gekeken moet worden naar het veranderen van de waarden van λ en de factoren voor spits, hoge vraag en lage vraag, of er zou naar een andere manier gekeken kunnen worden om de ritten te genereren.
Omdat tussen het gebruik van elektrische en niet-elektrische taxi's geen significant verschil is, ligt de keuze voor elektrische taxi's voor de hand. De elektrische taxi's zijn beter voor het milieu, hebben genoeg tijd om op te laden en hoeven dan niet naar een tankstation te rijden.
Om preciezer te kunnen bepalen hoeveel taxi's de meeste winst leveren, zou er in het vervolg ook nog gekeken kunnen worden naar situaties met bijvoorbeeld 30 of 50 taxi's.

De extra voorwaarden die Liang voorstelt op het oplossingsgebied kleiner te maken en zo de rekensnelheid van het programma te verkleinen, zijn in de programmering in Matlab overbodig en zorgen er zelfs voor dat het programma langzamer wordt. Aanpassingen die wel nuttig zijn, zijn de afronding op geheeltallige variabelen minder nauwkeurig maken en de oplossingswaarde minder nauwkeurig maken.
Zelfs met de aanpassingen om het programma sneller te maken, blijkt het produceren van resultaten waarvoor niet alle zones gebruikt worden, erg langzaam. Voor verder onderzoek naar dit project is het daarom aan te raden om de snelheid van het programma te verbeteren, of om de code in een andere taal te programmeren.

Bibliografie

[Alonso-Mora et al., 2017] Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *PNAS*, 114(3):462–467.

[BI intelligence, 2016] BI intelligence (2016). 10 million self-driving cars will be on the road by 2020. http://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2015-5-6?international=true&r=US&IR=T.

[Engineers Online, 2017] Engineers Online (2017). Minister schultz wil dat nederland in eu voorop loopt bij zelfrijdende auto's. http://www.engineersonline.nl/nieuws/id27999-minister-schultz-wil-dat-nederland-in-eu-voorop-loopt-bij-zelfrijdende-autos.html.

[Fagnant and Kockelman, 2014] Fagnant, D. and Kockelman, K. (2014). The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Elsevier, Transportation Research Part C: Emerging Technologies*, 40:1–13.

[Liang et al., 2016] Liang, X., de Almeida Correia, G. H., and van Arem, B. (2016). Optimizing the service area and trip selection of an electric automated taxi system used for the last mile of train trips. *Elsevier, Transportation Research Part E: Logistics and Transportation Review*, 93:115–129.

[Rijksoverheid, 2015] Rijksoverheid (2015). Nederland wordt testland voor zelfrijdende voertuigen.

[Zhang et al., 2015] Zhang, W., Guhathakurta, S., Fand, J., and Zang, G. (2015). The performance and benefits of a shared autonomous vehicles based dynamic ridesharing system: an agent-based approach.