Beating the Nash Equilibrium in Traffic

Groep 10:
Ward Vandenbroucke
Warre Veys
Vincent Frippiat

Begeleiders en promotoren:
Kenneth Stoop
prof. dr. ir. Dirk Stroobandt
prof. dr. ir. Mario Pickavet
prof. dr. Pieter Audenaert

Project in het kader van het Vakoverschrijdend Project in de bachelor Computerwetenschappen of Elektrotechniek

2022-2023

Contents

1 Inleiding

Opstoppingen in het verkeersnetwerk zijn in het algemeen een groot probleem in België omdat het negatieve economische-, milieu- en gezondheidseffecten heeft. Aangezien in de meeste situaties een uitbreiding van dit netwerk niet mogelijk is wegens een gebrek aan beschikbare ruimte en hoge kosten moeten alternatieve oplossingen worden gezocht voor dit probleem. Routeplanning is de meest evidente keuze.

Veel automobilisten maken deze dagen gebruik van routeplanning-applicaties zoals Waze of Google Maps om de snelste route te berekenen van een startpunt naar een bestemming. De recentste versies van deze programmas kunnen bovendien actuele verkeersomstandigheden en verkeersgegevens gebruiken om de geplande route nog verder te optimaliseren. Deze applicaties hebben echter één groot nadeel: de enige kost die ze in rekening brengen is de individuele reistijd en daardoor houden ze weinig tot geen rekening met het mogelijke effect van de aanwijzingen die aan de individuele bestuurder worden gegeven op het volledig systeem van alle bestuurders. Dit heet een selfish wardrop of user equilibrium. Bij een zuiver user equilibrium gaat elke weggebruiker van het verkeersnetwerk de op dat moment snelste route kiezen zonder rekening te houden met de impact hiervan. Elke weggebruiker draagt zo bij tot de opstopping van de snelste route(s). Dit probleem wordt tegenwoordig opgelost door met real-time verkeersinformatie aan weggebruikers te signaleren dat er beter een alternatieve route wordt genomen. Maar kan dit niet worden voorkomen in plaats van het -vaak tevergeefs- te proberen oplossen? Verder introduceert de hoge belasting van een weg extra reistijden voor alle weggebruikers. Ook al is deze extra reistijd klein, maken vele druppels een emmer te vol, en is de impact op de totale reistijd enorm. Elk moment dat een auto verbruikt telt.

Hiervoor komen we aan een alternatieve methode van routeplanning waar gebruik wordt gemaakt van coördinatie van routebegeleiding tussen gebruikers van een systeem en het naleven van voorgestelde routes: het system equilibrium. Dit is een routeplanning waarbij de planning van elke gebruiker wordt bepaald op een manier waarop de totale reistijd van alle gebruikers zo klein mogelijk gemaakt wordt.

In dit project gaan wij ons richten op deze twee soorten routeplanning in Belgie: het user equilibrium en het system equilibrium. Tijdens dit project gaan we bestuderen hoe groot de impact is van het opleggen van routes die eventueel niet individueel optimaal zijn, maar die op hoger niveau het verkeersnetwerk vloeiender maken, en welke kost dit met zich meebrengt.

Het project maakt gebruik van statische gegevens over het Belgische verkeer. Het is geïmplementeerd met behulp van Python, en maakt gebruik van verschillende packages zoals onder andere "Osmnx", "Networkx", "Gurobi", "Plotly" en "Graphviz".

2 Organisatie en Samenwerking

Onze groep heeft de afgelopen weken hard gewerkt aan de uitvoering van dit project. De planning hiervan is gerepresenteerd in de figuur 2.1. In de eerste weken hebben we ons gericht op het vinden van de meest gepaste datastructuur voor ons project, en op het volledig begrijpen van het wiskundige model achter dit project. Vervolgens hebben we ons gericht op het implementeren van een efficiënte code die in staat is om de k-kortste paden te berekenen en de graaf en data aan te passen aan onze specifieke situatie. Daarna hebben we ons gericht op de implementatie van een programma dat de system equilibrium en user equilibrium berekent voor gegeven origin-destination paren in het netwerk en hun bijhorende flows. Zodra deze stap was afgerond, zijn we begonnen met het testen van verschillende parameters om optimale resultaten te behalen. Tegelijkertijd hebben we ook verschillende tests uitgevoerd om te bepalen wat de participatiegraad van weggebruikers in een eventueel systeem van system equilibrium routing zou zijn, in welke situaties dit praktisch toepasbaar is en op welke manier we de participatie zouden kunnen verhogen. Tijdens het volledige project hielden we ons ook bezig met de visualisatie van onze resultaten.

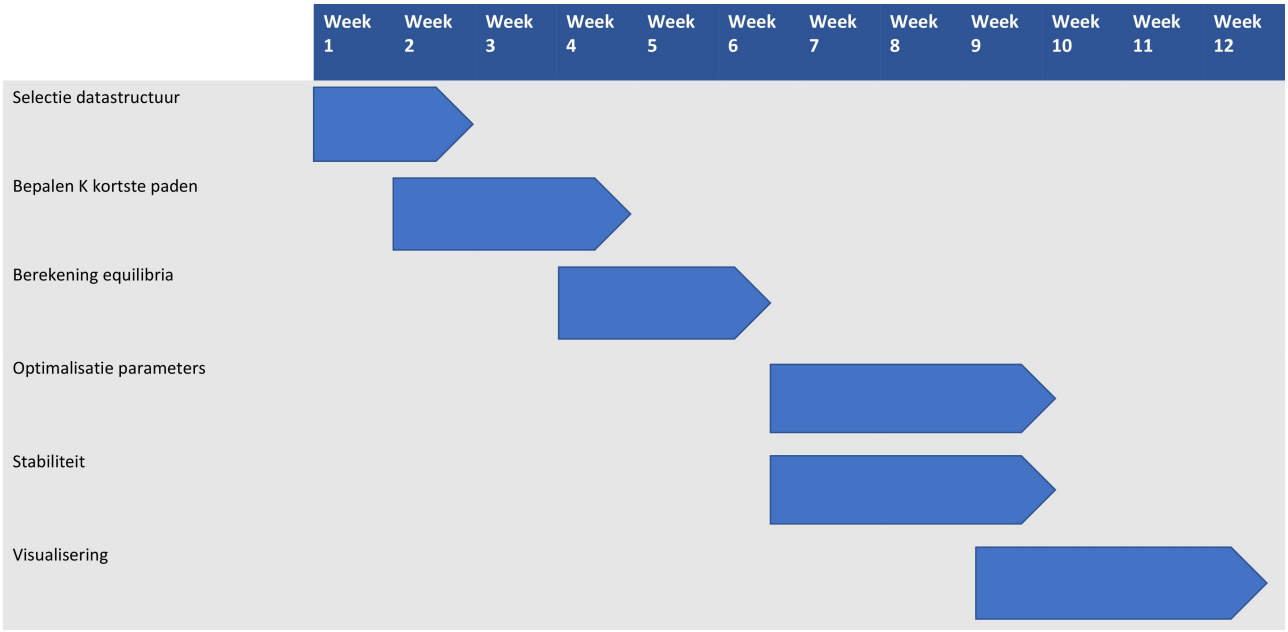


Figure 2.1: Finale planning van het project

In het algemeen hebben we voldaan aan de verwachtingen die we hadden over ons project. Onze initiële planning is redelijk similair aan onze huidige planning, omdat dit project in het algemeen verliep volgens onze initiële planning. Echter, zoals de figuur 2.2 het aangeeft, onze grootste teleurstelling is dat we niet genoeg tijd hebben gehad om ons project te herschalen naar een grotere (geografische) kaart en een bredere variëteit aan verkeersinformatie. Dit komt doordat de initiële stappen iets meer tijd vroegen dan we hadden voorspeld. Zeker het debuggen was moeilijk in het geval van ons project, omdat het niet direct zichtbaar is in welk deel van de code een fout wordt geïntroduceerd. Na het zien van een onregelmatigheid in de numerieke uitkomsten moest de volledige code doorzocht worden naar de boosdoener.
Ons huidig project is uitgevoerd in een regio rond Sint-Niklaas, terwijl we graag de scope hadden uitgebreid naar een kaart van België. Dit hebben we in de gegeven tijd niet kunnen doen Dit is mede mogelijk gemaakt door het feit dat de berekentijd van het wiskundig model met de huidige beschouwde regio al zeer groot was. Vervolgens hebben we gedurende het project gebruik gemaakt van een klein aantal datasets gebaseerd op "echte" statische verkeersinformatie, en ook eigen gecreëerde, ook hier hadden we graag meer tests uitgevoerd op een groter aantal situaties met uiteenlopende verkeersomstandigheden. Men moet zich bij het lezen dus focussen op concepten, conclusies en grootte-ordes, maar niet te veel op de exacte numerieke waarden.

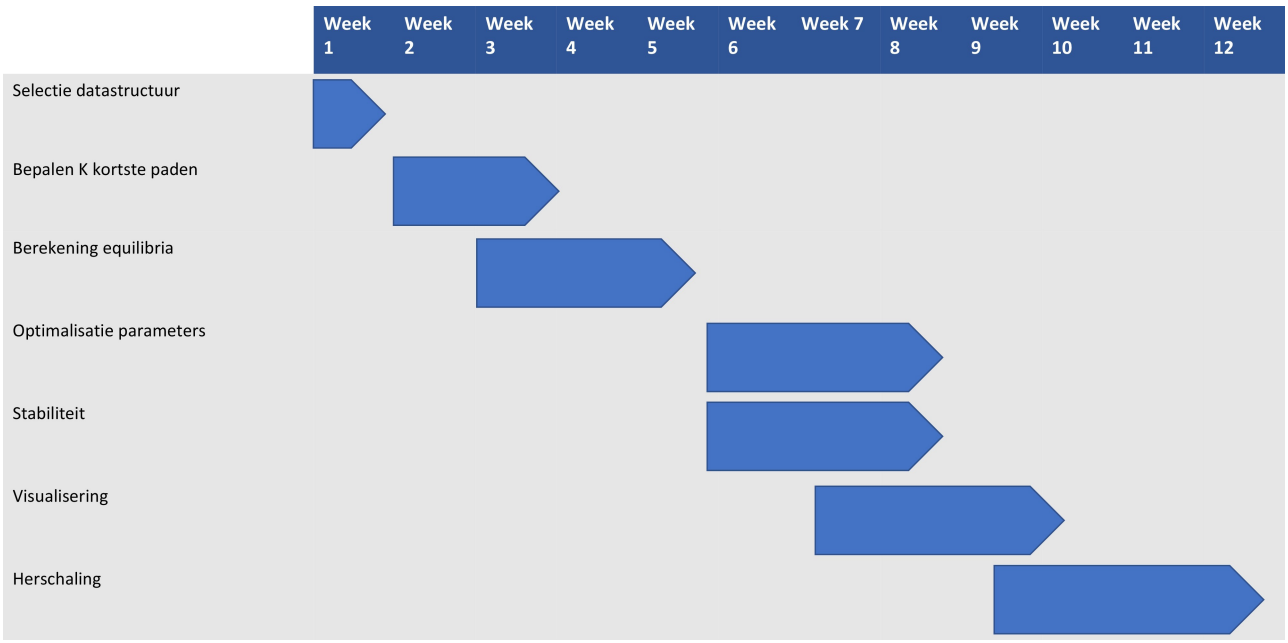


Figure 2.2: Initiële planning van het project

Onze samenwerking verliep over het algemeen zeer goed vanwege onze goede communicatie en efficiënte taakverdeling. Ondanks het feit dat het project, met uitzondering van de testfase aan het einde, moeilijk op te splitsen was in parallelle en onafhankelijke taken, was er een zeer goede samenwerking en coördinatie tussen de drie groepsleden. Er werd ook rekening genomen met elkaars sterke punten en zwakke punten, waardoor we op een effectieve manier het beste uit onze individuele vaardigheden konden halen.

3 Technische Beschrijving

3.1 Datacollectie

De eerste stap omvat het verzamelen van onderzoeksdata en het opslaan van deze data in een gemakkelijk te bewerken datastructuur. Met deze data onderzoeken we het user equilibrium en het system equilibrium. Onze datacollectie bevat twee hoofdsoorten gegevens.

3.2 Weginformatie

Ten eerste is er de data van OpenStreetMap (OSM), die alle noodzakelijke kaartinformatie over straten en wegen in België bevat. Dit omvat het aantal rijstroken, de snelheidslimiet, de lengte, enzovoort. Elk wegsegment en kruispunt tussen wegsegmenten heeft een unieke ID, en is dus onderscheidbaar.

Uiteindelijk wensen we deze informatie bij te houden in Python in een efficiënte en makkelijk manipuleerbare datastructuur. In onze situatie is een graaf uiteraard de best mogelijke representatie.We gebruiken de adjacency list als onderliggende datastructuur voor deze graaf, waarvan een voorbeeld wordt getoond in figuur 3.1 Dit is een lijst waarbij elk element A een knooppunt op de kaart vertegenwoordigt. Elk element bevat een gelinkte lijst van alle knooppunten waarmee het knooppunt A verbonden is en waarbij de gerichte tak vertrekt vanuit knooppunt A. Elk element in deze gelinkte lijst bevat informatie over deze verbinding, die een straat in de echte kaart voorstelt. Deze graaf wordt gemaakt met de klasse DiGraph uit de Python-package "Networkx", een package gemaakt voor de efficiënte creatie en manipulatie van zeer grote grafen. Bij de DiGraph wordt de adjacency list zelf voorgesteld door geneste woordenboeken, wat het zeer eenvoudig maakt om data op te vragen en aan te passen.

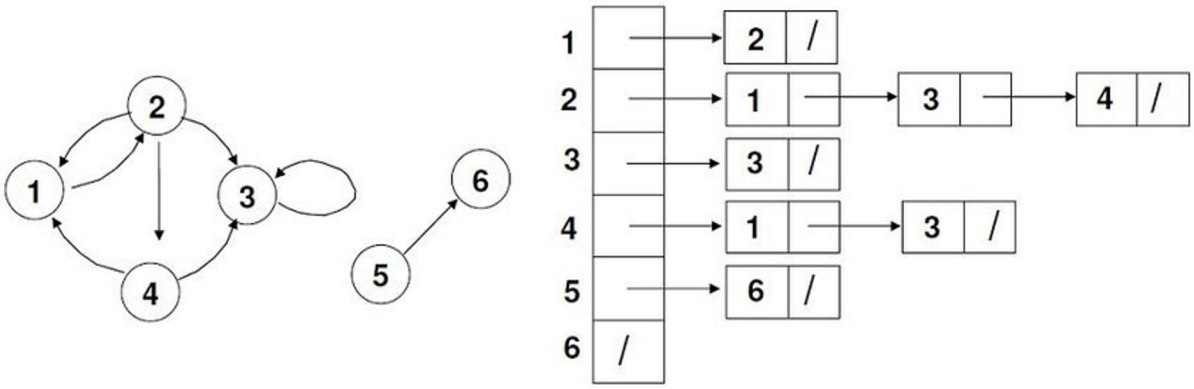
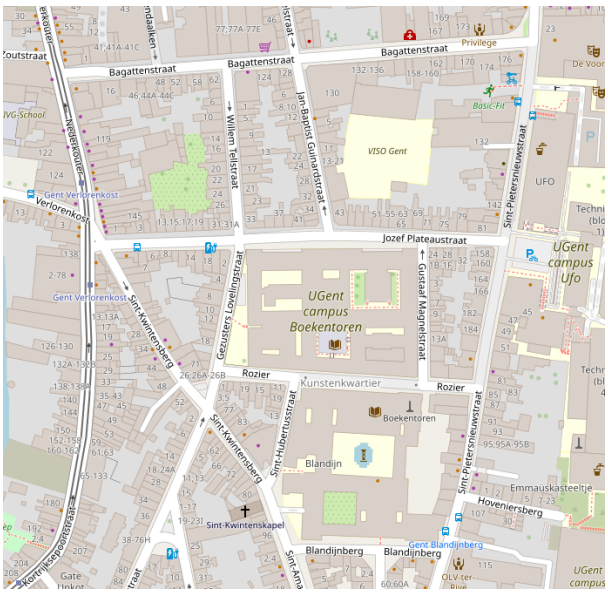


Figure 3.1: Voorbeeld van een graaf ondersteund door een adjacency list

Om uiteindelijk zo een graaf te bekomen met de juiste weginformatie gebruiken we de Python package "Osmnx" om de data in Python te importeren. De figuur 3.2 is een voorbeeld ervan: de linkerfiguur 3.2a is een kaart van Gent rond de Plateaustraat, en de figuur 3.2b rechts daarvan is een graaf die deze kaart representeert en informatie erover bevat.



(a) Kaart op OpenStreetMap

(b) Equivalent van kaart in een graaf

Figure 3.2: Voorbeeld van datacollectie van kaartinformatie

3.2.1 Vervolledigen van de weginformatie

Helaas is deze data niet compleet. Niet alle takken bevatten de attributen die nodig zijn voor de berekeningen en testen. Om de data te vervolledigen dienen de nodige benaderingen te worden gemaakt.

Zo worden alle ontbrekende maximale snelheden benaderd door het gemiddelde van alle maxspeeds (maximale snelheid van een route) van het bepaalde highway type, en het gemiddelde van alle maxspeeds wanneer er geen enkele tak met hetzelfde type is.

Andere benaderingen werden gemaakt met behulp van grondige inspectie van de echte wegsituatie van een aantal probleemgevallen via een streetview. Wanneer er bijvoorbeeld meerdere waarden voor het aantal rijvakken worden opgegeven komt dit doordat er op bepaalde plaatsen een extra voorsorteervak, een extra op-/afrit of een busstrook aanwezig is: hier wordt het aantal rijvakken dus het minimum van deze waarden. Een gelijkaardige situatie komt voor wanneer meerdere maximale snelheden zijn opgeslagen. Dit kan gebruikt worden voor talloze speciale gevallen zoals een lagere snelheid bij mistig weer of voor zwaardere voertuigen, enz. In deze situaties werd telkens het maximum gekozen. Na nog een aantal van deze simpele benaderingen bevat nu elke tak de -voor de berekeningen- nodige informatie over het bijhorende wegsegment.

3.2.2 OD-data

Ten tweede is de Origin-Destination (OD) flow data nodig. Deze gegevens zijn de verzameling van informatie over de bewegingen van voertuigen tussen specifieke oorsprongen en bestemmingen in België. De OD flow data toont aan hoeveel voertuigen per tijdseenheid reizen tussen twee knooppunten, dewelke dus aan routes worden toegekend volgens een bepaald model.

Echter kregen wij geen toegang tot de echte OD-data, maar verkregen we wel een aantal "artificiële" datasets gebaseerd op de echte OD-data van een regio rond Sint-Niklaas. Vandaar werd er logischerwijze ook besloten om deze regio te beschouwen als input graaf.

Het gebruikte model stelt een statische verkeerssituatie voor. Elke tijdseenheid wensen dezelfde hoeveelheid auto's dezelfde verplaatsing af te leggen. Een realistischere situatie zou zijn om een dynamische verkeerssituatie te beschouwen in plaats van de statische momentopname, en te zien hoe beide equilibria omgaan met dynamische gebeurtenissen.

In dit project worden enkel statische "momentopnames" van het verkeer beschouwd. De dynamische variant hiervan kan worden gezien als een uitbereiding. Aangezien de spitsuren vrij steady state zijn en het vooral in deze spitsuren is dat er nood is aan een alternatieve routering blijft de situatie geschetst door de statische

data zeer relevant. Bovendien is het hiermee eenvoudiger om eigen OD-data te genereren om een bredere variëteit aan verkeerssituaties te beschouwen. Dit project is dus eerder een onderzoek over de invloed van het system equilibrium in plaats van een exacte representatie van de system equilibrium routering in de huidige verkeerscontext.

3.3 User en system equilibrium

3.3.1 Wiskundige beschrijving

We beschouwen een graaf $G = (V, A)$, een gerichte graaf-voorstelling van het beschouwde verkeersnet, waar V de verzameling van alle knooppunten is en A de verzameling van alle takken.

Bovendien is $W \subseteq V \times V$ de verzameling van alle OD paren, de verzameling van de koppels $w = (a, b)$ waartussen de verkeersflows plaatsvinden. Verder definiëren we $\forall\ w\ \in W$:

- d_w als de flow (hoeveelheid weggebruikers) tussen een punt a en b in één tijdseenheid. Dit is dus de flow die we volgens een bepaald model wordt toegekend aan de verschillende beschouwde routes tussen a en b.
- R_w als de verzameling van de beschouwde routes tussen de oorsprong a en de bestemming b. Elke route is een lijst van opeenvolgende takken die worden gevolgd. Indien deze verzameling niet klein genoeg wordt gedefinieerd, kan het dat het programma te rekenintensief wordt voor een uitvoering op een gewone computer. Hierdoor worden enkel de k kortste paden tussen a en b gekozen, volgens een procedure die in het deel 3.5 van dit verslag wordt uitgelegd.
- f_r: de flow over een specifieke route $r \in R_w$

We beschouwen ook $\forall\ u\ \in A$:

- f_a: de flow over een tak a

De essentie van dit project komt dus neer op de correcte verdeling van de flow van elk beschouwd koppel $w = (a, b)$ over de $k-$kortste paden tussen de oorsprong a en bestemming b. Bij het system equilibrium worden deze zodanig toegekend zodat ze zorgen voor een coördinatie van de weggebruikers, om uiteindelijk de totale reistijd te minimaliseren. De aanpak bij het user equilibrium is eerder geconcentreerd op elke individuele weggebruiker waardoor er minder samenwerking is op een hoger niveau.

Om een verkeerssituatie correct voor te stellen moet voldaan zijn aan de volgende voorwaarden:

De flow tussen een OD-paar wordt volledig verdeeld over de beschouwde routes:

$$\sum_{r \in R_w} f_r = dw \tag{3.3.1}$$

Een flow mag niet negatief zijn:

$$\forall f_r : f_r \geq 0 \tag{3.3.2}$$

Het aantal weggebruikers die per tijdseenheid door een tak worden gestuurd is gelijk aan de som van alle flows toegekend aan routes die deze tak gebruiken:

$$\forall a \in A : f_a = \sum_{r \in R} f_r \delta_{ra},\ \text{met } \delta_{ra} = 1 \text{ indien r gebruik maakt van a, anders 0} \tag{3.3.3}$$

Om flows volgens een model aan routes toe te kennen, is een benadering van de individuele reistijd over een tak t_a wanneer f_a weggebruikers per tijdseenheid over deze tak reizen essentieel. Dit wordt verkregen via volgende vergelijking (wereldwijd gebruikt en uitgebracht door het U.S. Bureau of Public Roads in 1964) 3.3.4, die de reistijd t_a voorstelt in functie van de flow f_a:

$$t_a(f_a) = t_a^\circ \Big(1 + 0.15\big(\frac{f_a}{c_a}\big)^4\Big) \tag{3.3.4}$$

De reistijd hangt, naast de flow, ook af van de de capaciteit c_a en de reistijd zonder flow t_a°. Deze laatste kan eenvoudig worden benaderd met de verhouding van de lengte tot de maximale snelheid. De capaciteit c_a is het maximale aantal weggebruikers die per tijdseenheid aan deze tak kan worden toegekend zonder dat er congestie optreedt. Dit is ook duidelijk te zien in 3.3.4, waarbij de additionele reistijd door de aanwezigheid van een bepaalde flow bikwadratisch afhangt van $factor$ met $f_a = factor * c_a$. Deze zal dus aanzienlijk stijgen wanneer de capaciteit wordt overschreden. De capaciteit zelf wordt benaderd door $c_a = 1000 * n$, met n

het aantal rijvakken van het wegsegment. Deze benadering werd gekozen omdat er wordt gewerkt met een floweenheid van auto's/uur, aangezien dit neerkomt op een flow van 1 auto per rijvak per 3.6s, of dus een afstand tussen weggebruikers van x meter bij een maximale snelheid van x km/u, wat ook conform met de rem- en reactieafstanden de aangeraden afstand tussen voertuigen is.

Voor het berekenen van de evenwichtspunten wordt gebruik gemaakt van uitdrukkingen in "Studies in the Economics of Transportation" van Beckman Mcguire en Winsten, die beide neerkomen op een minimalisatieprobleem over de flows op de takken (f_a), maar elk met een eigen kostenfunctie, en rekening houdende met de bovenvernoemde constraints.

De berekening van het system equilibrium draait om het vinden van een evenwichtige verdeling van de verkeersbelasting voor een efficiënt gebruik van het wegennet, waarbij de totale reistijd van alle weggebruikers wordt geoptimaliseerd. De totale reistijd per weggebruiker is gelijk aan $\sum_{a \in r} t_a(f_a)$ met r de genomen route. Hierdoor kan de - voor het system equilibrium gewenste- minimalisatie van de totale reistijd per tijdseenheid -na herformulering- wiskundig als volgt worden geformuleerd:

$$min_{\vec{f}} \sum_{a \in A} f_a t_a(f_a) \tag{3.3.5}$$

Bij het user equilibrium is de kostenfunctie iets minder intuïtief. Hierbij worden flows toegekend volgens de oplossing van het volgende minimalisatieprobleem:

$$min_{\vec{f}} \sum_{a \in A} \int_0^{f_a} t_a(x)dx \tag{3.3.6}$$

Het is moeilijker om hier over flows toekennen te spreken, aangezien "in het echte user equilibrium" elke weggebruiker de route selecteert die op dat moment -gegeven de weggebruikers die op dat moment op de routes zitten- qua reistijd optimaal is, zonder hierbij rekening te houden met andere weggebruikers. Het bovengenoemde minimalisatieprobleem simuleert echter wel dit gedrag bij statische data, en na de toekenning van de flows via deze formule is het zo dat men van geen enkele "weggebruiker" de reistijd verbetert door -gegeven de huidige toewijzing van flows- zijn route te veranderen.

3.3.2 Eigen implementatie

De minimalisatieproblemen aangegeven in uitdrukkingen 3.3.5 en 3.4.1 zijn geïmplementeerd in Python met als input de graafrepresentatie van het wegennet $G = (V, A)$ en de verzameling W van alle OD paren (met hun bijhorende flow d_w), samen met de routes R_w die bepaald zijn via de procedure beschreven in 3.5.

De kostenfuncties zijn echter complexe hogeregraadsfuncties en er zijn een zeer groot aantal onbekenden. Het oplossen van een dergelijk optimalisatieprobleem kan te zware computationele vereisten met zich meebrengen. Om deze complexiteit te verminderen, hebben we ervoor gekozen om het initiele probleem te transformeren naar een lineaire approximatie, aangezien hier computationeel efficiëntere solve algoritmes voor bestaan. Hierbij wordt gebruik gemaakt van de Gurobi Optimizer voor de implementatie van efficiënte lineaire optimalisatiemethoden.
De Gurobi Optimizer is een krachtige tool die specifiek is ontwikkeld voor het oplossen van complexe wiskundige optimalisatievraagstukken. De Gurobi Optimizer staat bekend om zijn efficiënte en snelle oplossingsmethode, waardoor het in staat is om complexe (in ons geval: lineaire) optimalisatieproblemen snel en nauwkeurig op te lossen. Dit is met name van belang bij routeringsproblemen, waarbij het vinden van optimale routes en verkeersstromen een uitdagende taak is vanwege de omvang en complexiteit van het netwerk. De Gurobi Optimizer biedt de mogelijkheid om verschillende voorwaarden op te nemen in het optimalisatieprobleem. Hierdoor kunnen we met behulp van de voorwaarden (vergelijkingen 3.3.1, 3.3.2, 3.3.3, 3.3.4) en de optimalisatievergelijkingen 3.3.5 en 3.4.1 de optimale flowdistributie vinden voor beide equilibriumsoorten.

3.4 Linearisatie van de functies voor het user en system equilibrium

Elk van de kostenfuncties in de sommatie uit 3.3.5 en 3.4.1 is specifiek per tak en er is dus telkens slechts 1 veranderlijke, namelijk de flow in de tak. Deze wordt benaderd door een stuksgewijze lineaire functie zoals in 3.3 wordt gedaan met behulp van de volgende formule:

$$y = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) \quad voor\ x \in (x_i, x_{i+1}] \tag{3.4.1}$$

Waarbij y de waarde is van de functie voor x tussen x_i en x_{i+1}. De figuur 3.3 toont ook duidelijk dat de linearisatie-intervallen in het begin kleiner zijn en groter worden naarmate de x-waarde vordert. Deze methode

wordt gebruikt om een maximum x-waarde te bekomen waarbij de functie stopt. Dit eindpunt bepalen is echter een complexe taak aangezien er een afweging is tussen zo veel mogelijk waarden binnen het bereik van de functie houden en zo weinig mogelijk geheugen te gebruiken voor de functie zelf (door het eindpunt onredelijk groot te maken). Er moet ook worden opgemerkt dat de grootte van een interval zal stijgen en dus de precisie van de benadering zal dalen indien het eindpunt wordt verhoogd (bij een constant aantal linearisatie-intervallen). Om dit probleem enigzins op te lossen wordt de som van alle flows tussen de paren, die als input data gegeven worden, genomen. Dit getal wordt gebruikt als maximum voor de flows (deze flows zijn in ons geval de x-waarde van onze functie). We zijn op die manier zeker dat de flow op een tak ten hoogste dit getal kan zijn. Dit zal zo zijn indien alle routes door deze tak gaan. Aangezien dat dit zeer ruim genomen is en de werkelijke flow dus vaak ver van dit maximum zal liggen, hebben wij er voor gekozen om het aantal linearisatie-intervallen te concentreren naar het begin door de grootte van de intervallen kleiner te houden voor kleinere waarden van de flow.

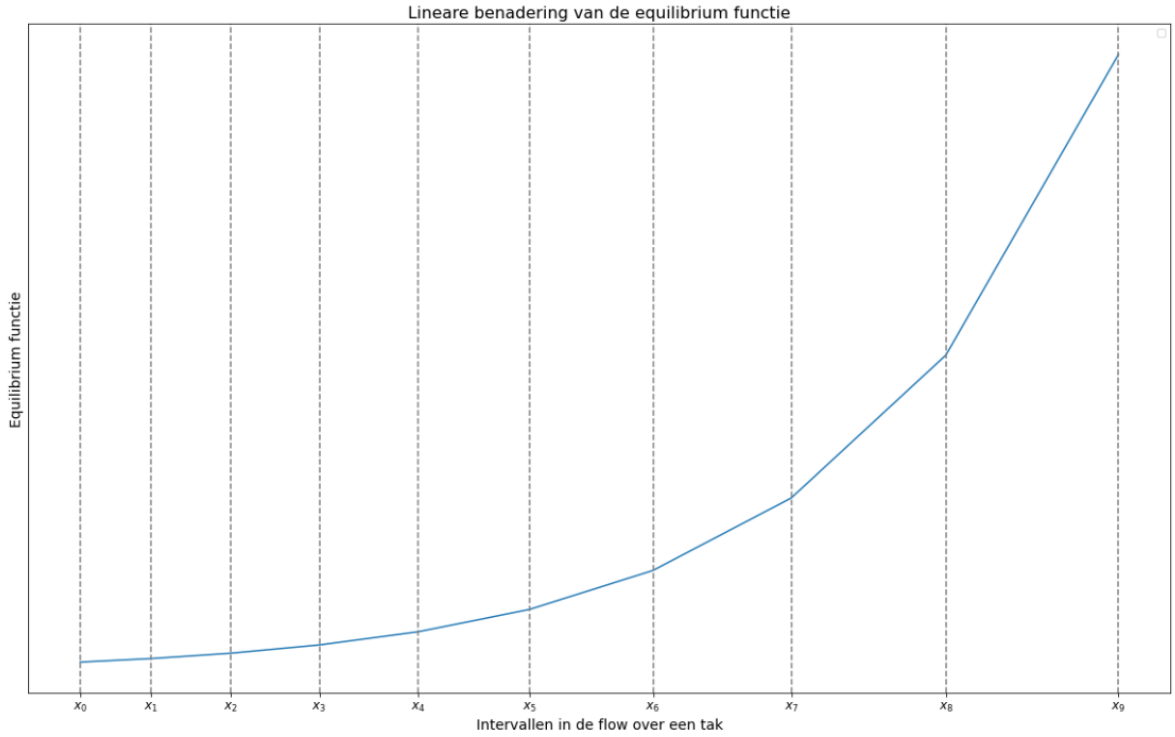


Figure 3.3: Voorbeeld van de linearisatie van een meerderegraadsfunctie

Er werd geprobeerd om op andere manieren te lineariseren. Ten eerste probeerden we te kijken naar alle OD-paren die een route hebben die deze tak gebruikt en gebruikten we de som van enkel deze flows als een strakkere bovengrens. Echter schaalde dit niet, de ene tak had een hoog aantal linearisatie-intervallen nodig voor een degelijke precisie, voor een andere werd dit aantal veel te groot door een lage bovengrens. Soms kan het bij lineariseren via de x-as voorkomen dat vrij stabiele stukken door veel intervallen worden voorgesteld, en sterk variërende stukken door te weinig, dit kan men oplossen door uniform te lineariseren volgens de functiewaarden. Bij ons heeft dit echter ook weinig zin, aangezien dit door het bikwadratisch gedrag (zie formule 3.3.4) zal zorgen voor te veel intervallen na de capaciteit van de tak, en het gebied voor (en net na) de capaciteit juist het belangrijkst is.

3.5 Vereenvoudiging met behulp van k kortste paden

Bij het berekenen van het system en user equilibrium is het niet mogelijk om alle mogelijke paden tussen elke origin en destination te beschouwen, aangezien dit een excessief aantal parameters zou representeren die in rekening moeten worden gebracht bij de optimalisatie: het programma zou te veel rekenkracht nodig hebben. Het is bovendien niet nodig om al deze paden in rekening te nemen omdat het praktisch gezien toch niet mogelijk is om een weggebruiker langs te grote omwegen te sturen, en er aan deze grote omwegen wegens het minimaliseren ook gewoon geen gebruikers worden toegekend bij beide modellen (zie 4.2). Hierdoor is het essentieel om voor elk Origin-Destination paar de k kortste paden van de Origin naar de Destination te bepalen. Deze paden zullen de enige paden zijn die in rekening zullen worden gebracht voor de distributie van de flows

van een Origin naar een Destination. De k kortste paden tussen een Origin en Destination worden berekend door de volgende stappen af te lopen:

1 . Bepaal het kortste pad tussen de Origin en Destination met behulp van het Algorithme van Dijkstra

2 . Vermenigvuldig de kost van elke tak op dit kortste pad met een factor x

3 . Indien er nog geen k kortste paden zijn gevonden, ga terug naar stap 1
Anders: stop

Het algoritme van Dijsktra werd gekozen als kortste-pad algoritme vanwege zijn relatief lage tijdscomplexiteit: O(E+VlogV). Aangezien er in dit project met grote grafen wordt gewerkt is de tijdscomplexiteit een fundamenteel criterium. Vervolgens werd er gekozen om de k kortste paden te berekenen met de methode die zonet werd beschreven omdat dit algoritme vermijdt dat meerdere kortste paden quasi-identische routes nemen (met één tak verschillend bijvoorbeeld). De keus van de factor x waarmee de kost van een tak wordt vermenigvuldigd is gebaseerd op een de volgende afweging:

- als de factor hoog wordt, zullen nieuwe kortste paden worden gekozen die een groot verschil in reistijd hebben met het effectieve kortste pad.
- als de factor laag wordt, zullen de kortste paden meer en meer gemeenschappelijke wegen (takken) hebben. Het uiterste geval, $x <= 1$, geeft k keer het allerkortste pad terug, wat natuurlijk niet de bedoeling is.

Aangezien de factor waarmee de kortste paden werden berekend van belang is, wordt in het vervolg van dit verslag gebruik gemaakt van de notatie k_{factor}. Ook zal er vaak route i gezegd worden, waarmee de i^{de} snelste route (snelste is route 0) wordt bedoeld.

Naast excessieve rekenkracht vermijden is ook het innemen van zo weinig mogelijk opslagruimte essentieel. Door zo min mogelijk informatie op te slaan in de graaf wordt er minder opslagruimte ingenomen, en gaat minder rekenkracht verloren aan het beheer van onnodige data. De gebruikte graaf wordt in de eerste plek vereenvoudigd door, na de berekening van de k kortste paden voor elk Origin-Destination paar, de takken van wegen die zich in geen enkel kortste pad bevinden te verwijderen van de graaf. Hier kunnen door de in 3.3.1 beschreven constraints toch geen flows aan worden toegekend en deze zorgen dus gewoon voor extra variabelen. Ook worden bij elk element in de graaf enkel de essentiële attributen bijgehouden: bij het importeren van data uit OpenStreetMap heeft elk element in de graaf vaak attributen die niet relevant zijn voor de rest van het project.

3.6 Visualisering

In het kader van ons onderzoek hebben we een kaart gecreëerd om een gedetailleerd overzicht van de stromen van het verkeer en routekeuzes te tonen bij een system equilibrium en bij een user equilibrium. Dit hielp enorm bij het visualiseren van en het redeneren over onze resultaten.

Middels deze visuele presentatie kunnen we op een toegankelijke en overtuigende wijze de resultaten van ons onderzoek communiceren naar geïnteresseerden die de dynamiek van verkeersrouting binnen verschillende evenwichtsmodellen willen begrijpen.

Het was een uitdaging om een goede visualisatie te vinden, de eerste problemen (bij het werken met de package "Graphviz") kwamen door de grootte van de graaf, en hierdoor duurde het ellenlang om de afbeelding te renderen. Wanneer dit op een andere manier wel lukte, was het visueel meestal een grote chaos. Uiteindelijk lukte het ons om door middel van de package "Plotly" een HTML-bestand te genereren dat een boeiende interactieve (zoomen, verschuiven, hoveren voor info) visualisatie van onze graaf en de toegewezen flows weergeeft.

Een voorbeeld hiervan is te vinden in de appendix, op figuur 8.1. Dit is een voorbeeld waarbij 1 OD-paar wordt beschouwd, $k_{10} = 5$ (hoge factor voor visueel duidelijk verschillende paden) en een -overdreven- flow van 5000, vandaar het gigantische verschil in totale reistijd.

Echter is het wel een zeer mooi voorbeeld van beide concepten. Zo ziet men na het hoveren over een tak van de vierde snelste route dat deze niet meer wordt gebruikt (flow 0), en dus dat het user equilibrium de totale flow verdeelt over de drie snelste routes (dit zijn in deze figuur de geel gekleurde, routes waarbij de flow toegekend door het UE groter is dan deze door het SE worden geel weergegeven, anders groen, en indien de flows hetzelfde zijn grijs). Elke weggebruiker neemt voor deze situatie de snelste route, maar hierdoor worden deze wegen overbelast, dit leidt tot een vertraging -die niet groot genoeg is om een alternatieve route aantrekkelijk te maken- voor elke weggebruiker, maar door de grote totale flow zorgen al deze extra vertragingen voor een

zeer hoge totale reistijd.

Bij het system equilibrium daarentegen worden de verkeersstromen evenwichtiger over de wegen verdeeld. Een deel van de flows krijgt een langere route toegekend (wat theoretisch makkelijk is, echter mee moet opgelet worden, zie 5.2), zodat de overige routes voor een deel ontlast worden en al deze overblijvende weggebruikers een lagere reistijd krijgen, en de totale reistijd hierdoor aanzienelijk daalt.

Om het wegennetwerk mooi voor te stellen wordt gebruik gemaakt van het attribuut "Geography" die in bepaalde edges aanwezig was na inladen uit Openstreetmap. Dit attribuut was een LineString van de cooördinaten van de longitude en latitude om zo de vorm van het wegsegment voor te stellen. Hieruit konden we dus ook de longitude- en latitude-coördinaten van de knopen halen (begin- en eindpunt). Opnieuw was dit attribuut allesbehalve altijd aanwezig, en werd gezocht naar de coördinaten van de knopen in andere naburige takken, om zo de tak waarbij het attribuut ontbreekt voor te stellen als een rechte lijn. Wanneer er in geen enkele tak coordinaten van een bepaalde knoop aanwezig zijn, dan wordt deze benaderd door het gemiddelde van de (eventueel ook benaderde) coordinaten van zijn buren.

4 Testen

4.1 Optimalisatie parameter: optimaal aantal linearisatie-intervallen

Bij het berekenen van zowel het gebruikers- als het systeemevenwicht is het aantal linearisatie-intervallen van groot belang. Deze keuze heeft namelijk impact op twee essentiële elementen: het resultaat, namelijk de totale reistijd bij zowel het gebruikers- als het systeemevenwicht, en de rekentijd van het programma dat deze evenwichten berekent. Er moet dus een afweging worden gemaakt.

De eerste meespelende factor is de rekentijd. Bij observatie van figuur 4.1 wordt opgemerkt dat de rekentijd een lineare trend volgt met toenemende aantal van linearisatie-intervallen. Zoals te verwachten moeten we dus in het achterhoofd houden dat hoe hoger we de precisie kiezen, hoe langer er gerekend wordt tot een optimale verdeling van flows.

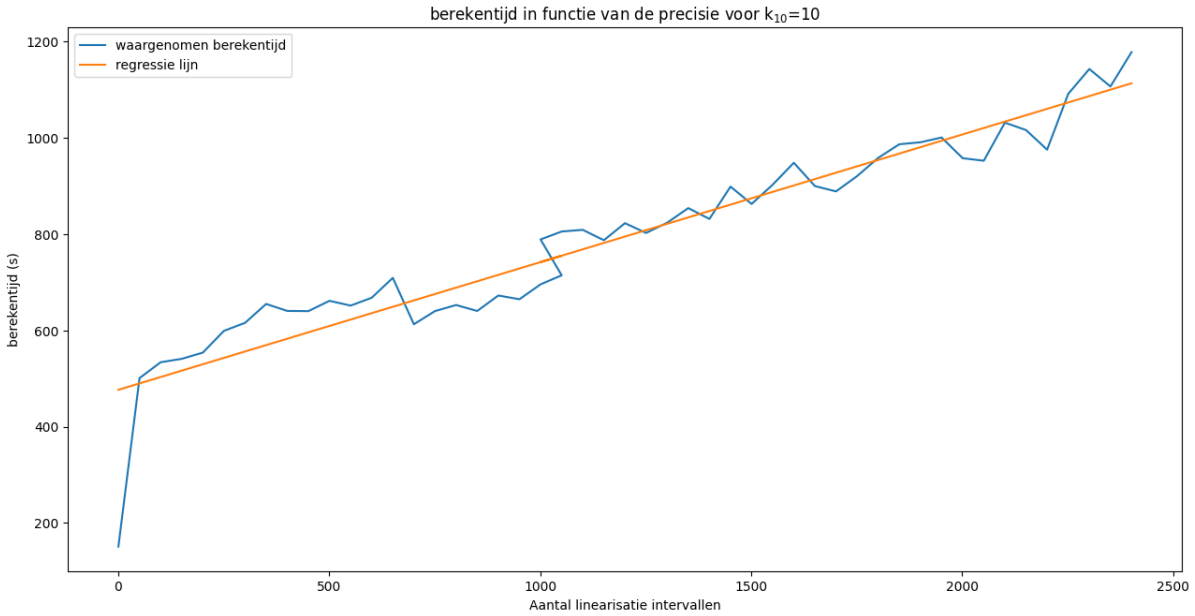
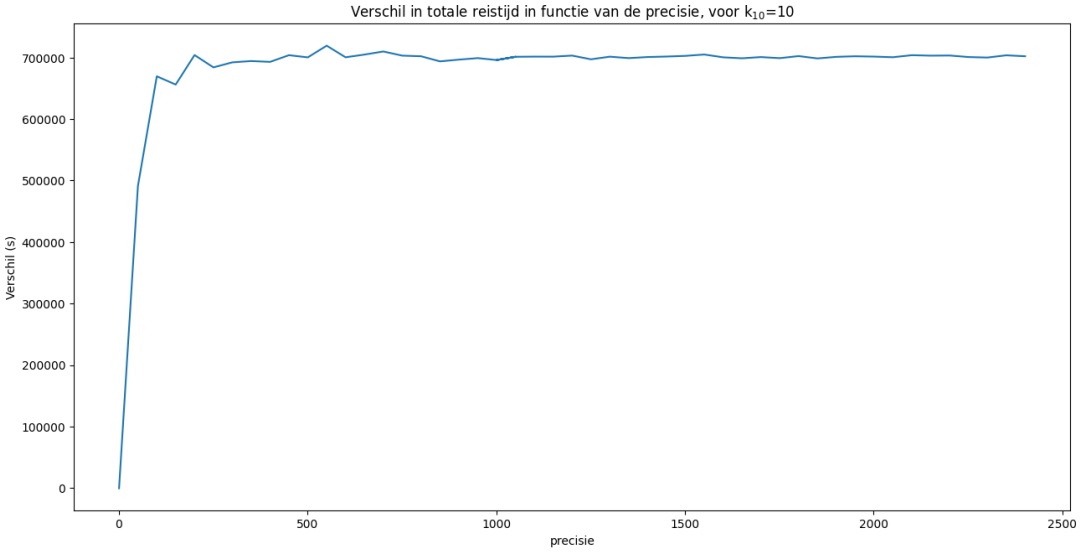


Figure 4.1: Invloed van het aantal linearisatie-intervallen op de rekentijd

Ten tweede is er de precisie. Deze wordt logischerwijze, zoals ook te zien in figuur 4.2, hoger bij meer linearisatie-intervallen. In 4.2a is te zien hoe het verschil in totale reistijd tussen het user equilibrium en het system equilibrium zich begint te stabiliseren bij voldoende groot aantal linearisatie-intervallen. Rond de 1000

linearisatie-intervallen zijn de grote pieken voorbij, er zit duidelijk nog altijd een variatie op -zeker gezien de eenheid op de as-, echter moeten we gezien onze eerdere conclusie over de rekentijd een bepaalde fout aanvaarden. Ook op de onderste figuur (4.2b) wordt geen warm water uitgevonden, hoe meer linearisatie-intervallen er worden gebruikt, hoe kleiner de fout geïntroduceerd door het lineariseren. De fout die voor deze figuur werd gebruikt is het verschil tussen de uiteindelijke uitkomst van de objective functie (3.3.5, deze is dus berekend door de Gurobi Optimizer met de lineaire benaderingen van de kostenfuncties per tak) en de totale reistijd achteraf -gegevens de flows toegekend aan elke tak- "exact" berekend via 3.3.3.



(a) Verschil in totale reistijd

(b) Linearisatiefout in functie van het aantal linearisatie-intervallen

Figure 4.2: Hoe hoger het aantal intervallen, hoe preciezer de berekeningen

Op basis van deze bevindingen werd besloten om rond de 2000 linearisatie-intervallen te gebruiken, aangezien de benadering nauwkeurig genoeg is en de rekentijd binnen redelijke grenzen blijft. Deze keuze varieert echter wel per laptop aangezien de rekentijd afhangt van specificaties zoals de grootte van het geheugen.

4.2 Optimalisatie parameter: aantal k_{factor} kortste paden

Naast de keuze van het aantal linearisatie-intervallen is ook de keuze van het aantal kortste paden, k, zeer belangrijk bij de berekening van het user equilibrium en system equilibrium. Deze keuze heeft namelijk impact op zowel de totale reistijd als de rekentijd.

De figuur 4.3 representeert de situatie met $factor = 10$: een nieuwe weg heeft slechts in uitzonderlijke situaties takken gemeenschappelijk heeft met een vorige snelste route. Verder zijn de grafieken in deze sectie ook gemaakt met een statische verkeerssituatie die een rush hour voorstelt (dit is een zeer hoge totale flow verdeeld over vele OD-paren over de volledige graaf, in dit geval van 33406).

Een te laag aantal kortste paden k heeft weinig nut omdat er niet genoeg mogelijkheden zijn om flows te distribueren tussen een origin en destination. Dit is makkelijk op te merken in figuur 4.3, die de totale reistijd toont in functie van het aantal kortste paden k_{10}. Conform de conclusies van het vorige deel nemen we een precisie van 2000 linearisatie-intervallen. Bij k gelijk aan 1 zijn de totale reistijden van het UE en SE exact hetzelfde, omdat er tussen elk OD-paar maar één pad is. Hierdoor gebruiken het UE en SE in deze situatie natuurlijk exact dezelfde routes.

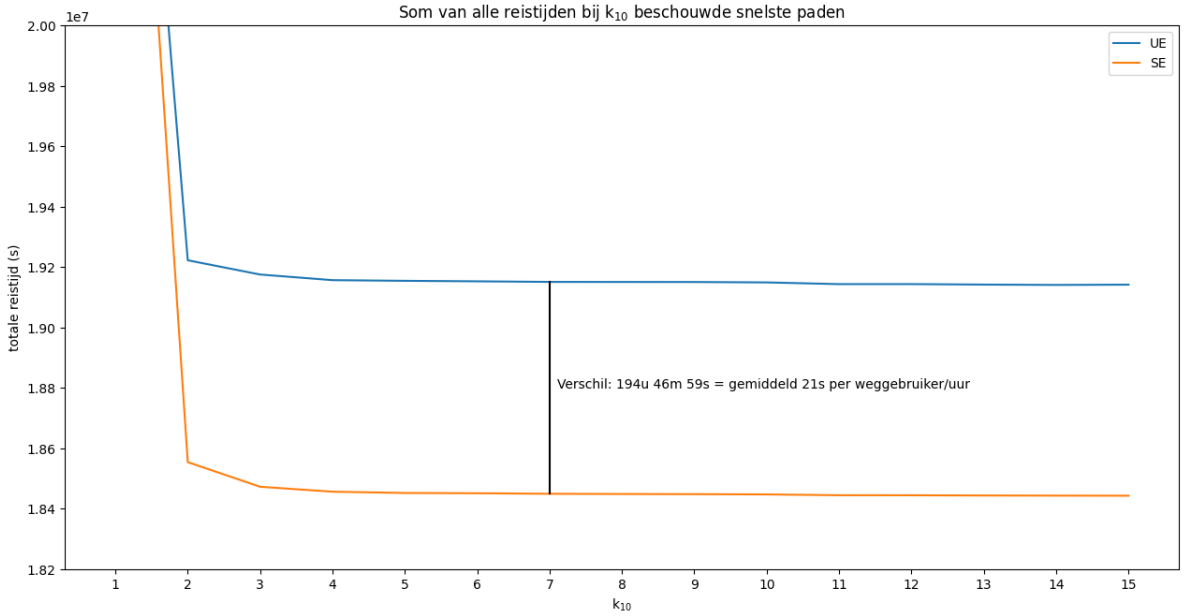
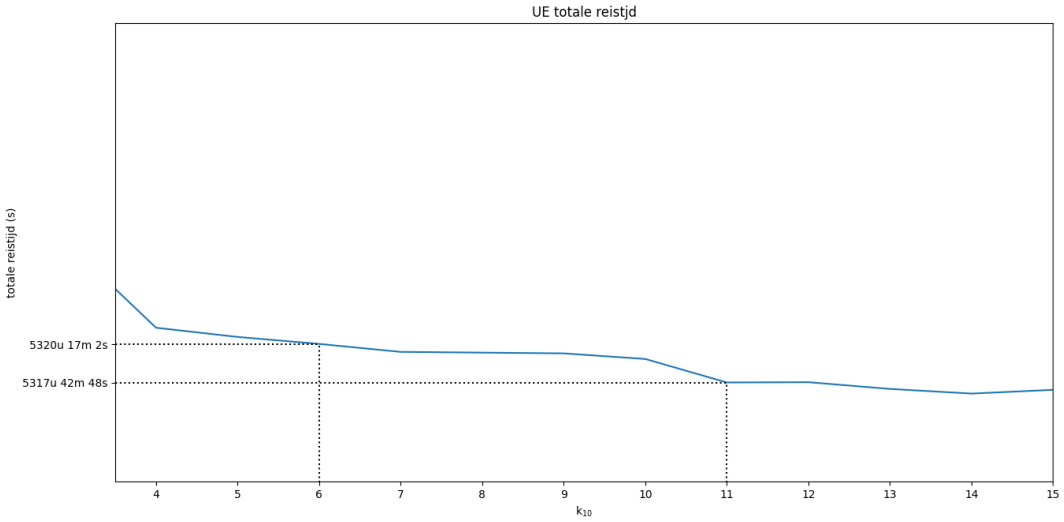
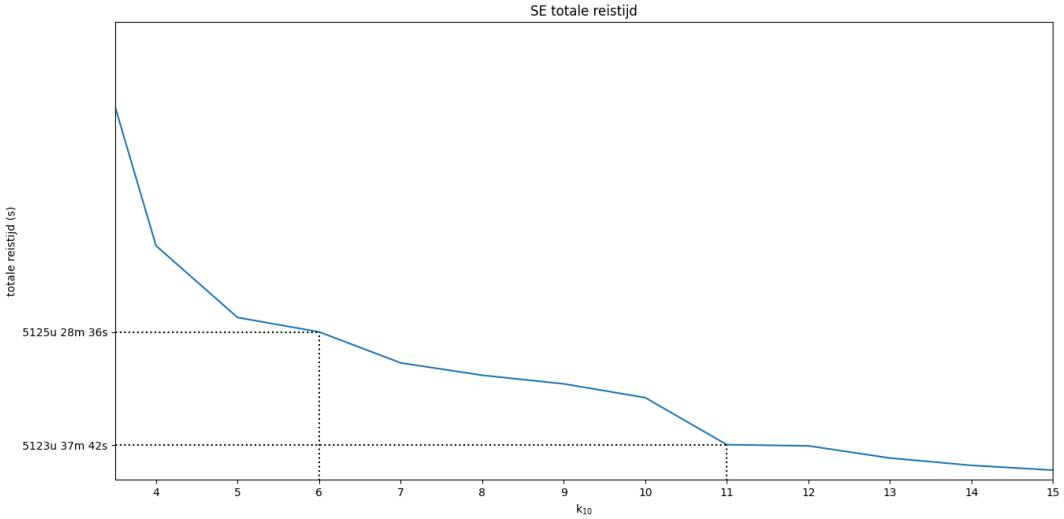


Figure 4.3: Invloed van het aantal kortste paden k_{10} op de totale reistijd

Daarna lijkt het alsof de totale reistijd bij beide niet meer daalt, maar dit komt door de misleidende eenheid van de y-as, zoals de figuren 4.4a en 4.4b het aangeven, waarbij duidelijk is te zien hoe bij toenemende waarde van het beschouwde aantal paden er nog steeds een noemenswaardige daling is in de totale reistijd. Zeker bij het system equilibrium lijkt de daling wel te verminderen, maar zet de trend zich wel door. Bij het user equilibrium komt een eigenaardige stijging van de totale reistijd voor bij de overgang van $k_{10} = 14$ naar $k_{10} = 15$. Bij toenemende k_{10}-waarde zal er meer interactie zijn tussen de verschillende routes. Hierdoor zullen sommige weggebruikers hun reistijd verlagen door de nieuwe route te nemen. Echter hierbij kan het gebeuren dat door de interactie met paden van andere OD-paren bepaalde anderen ook moeten uitwijken naar langere routes om het user equilibrium in stand te houden. Dit kan eventueel de totale reistijd negatief beinvloeden. De belangrijkste conclusie is dat het system equilibrium doet wat we verwachten: handig gebruik maken van extra routes om belastingen te verdelen en de totale reistijd te verlagen, en hier mag er natuurlijk nooit een stijgende trend zijn, aangezien alle routes van $k_{10} = i$ ook in $k_{10} = i + 1$ zitten.



(a) User equilibrium



(b) System equilibrium

Figure 4.4: Invloed voor het UE en SE apart bekeken

Hierbij zou men kunnen concluderen dat meer paden altijd beter is. Dit is echter niet het geval, verklaringen hiervoor komen hieronder aan bod.

4.2.1 Verklaring 1: rekentijd

Figuur 4.5 beschrijft de totale rekentijd van het UE en SE in functie van het aantal kortste paden. De curve heeft over het algemeen een stijgende lineaire trend, waarbij de rekentijd hoger wordt bij een stijgend aantal kortste paden. Aangezien de totale reistijd vanaf k=4 kortste paden nog zeer weinig daalt, is het aan te raden om een aantal kortste paden te kiezen rond k=4 omdat de rekentijd snel stijgt voor stijgende k-waarden.

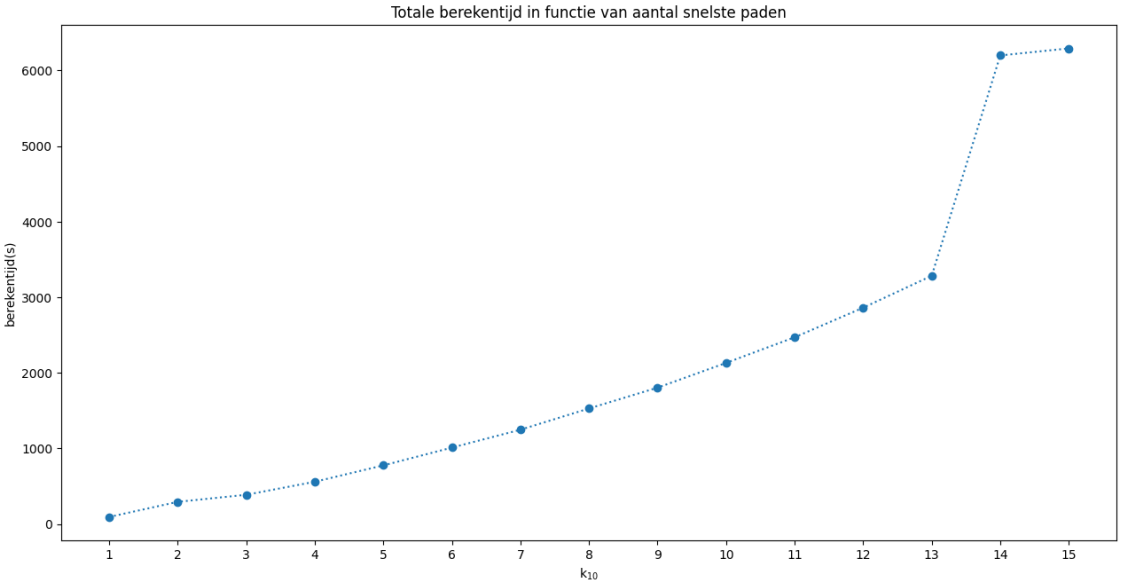


Figure 4.5: Invloed van het aantal kortste paden k op de rekentijd

Het is echter wel belangrijk om in rekening te nemen dat de rekentijd enorm afhangt van de machine waarop en de context waarin het programma wordt uitgevoerd. De berekeningen gebruikt om de figuur 4.5 te maken werden bijvoorbeeld uitgevoerd doorheen een nacht, waarbij alle andere programma's gesloten waren en waarbij er dus een minieme extra belasting was. Vervolgens hangt de rekentijd ook af van het programma zelf en de grootte van het probleem dat wordt opgelost. Aan de manier waarop het model wordt geoptimaliseerd kan er niet veel meer worden gedaan aangezien de Gurobi Optimizer één van de snelste technieken in zijn domein in de wereld levert. Er kan echter wel getracht worden een verandering te brengen in de rekentijd van het programma door de manier waarop het model wordt opgebouwd en welke startwaarden (en hoe deze) worden doorgegeven aan de variabelen.

Om dit te testen zijn talloze random OD-paren met random flows gegenereerd (waarbij een hoge totale flow is). Hiermee hoopten we een licht patroon te vinden bij de procentuele toekenning van de flows aan de verschillende routes.

route i	mean (%)	std dev (%)
0	0.836756	0.362253
1	0.112986	0.309587
2	0.030333	0.165901
3	0.007028	0.079287
4	0.004189	0.061589
5	0.001730	0.039356
6	0.001959	0.042409
7	0.001118	0.032173
8	0.001390	0.036519
9	0.001415	0.036869
10	0.001001	0.030951
11	0.000787	0.027367
12	0.001474	0.037857
13	0.001111	0.032815
14	0.001187	0.034026

Figure 4.6: Gemiddelde en standaardafwijking na vele random testen voor het system equilibrium

route i	mean (%)	std dev (%)
0	0.935067	0.240650
1	0.049078	0.210361
2	0.006683	0.077340
3	0.001606	0.037297
4	0.001615	0.038943
5	0.000903	0.029572
6	0.001244	0.034952
7	0.000829	0.028495
8	0.001111	0.033151
9	0.000888	0.029629
10	0.000762	0.027378
11	0.000566	0.023377
12	0.001196	0.034456
13	0.000866	0.029146
14	0.000877	0.029391

Figure 4.7: Gemiddelde en standaardafwijking na vele random testen voor het system equilibrium

De standaardafwijkingen in figuur 4.7 voorspellen niet veel goeds, maar toch testten we even door aan Gurobi

Optimizer een startwaarde aan de variabelen die een route i voorstellen mee te geven, namelijk het gemiddelde percentage van de totale te verdelen flow van het OD paar waarbij de route behoort. Ook al is $k_{10} < 14$, zullen we gewoon bovenstaande percentages gebruiken, het gaat eigenlijk zuiver over het doorgeven aan de Gurobi Optimizer van het grote gebruik van de eerste routes.

Er werd weinig tot geen verschil opgemerkt, en dus raden we opvolgers aan net zoals ons tijd te steken in het schrijven van vele csv-functies.

4.2.2 Verklaring 2: extra individuele reistijd

Kijken we naar onderstaande tabel dan zien we al snel een tweede en veel belangrijker probleem.

route i	mean (%)	std dev (%)
0	0.0	0.0
1	0.525	0.7233
2	1.1648	3.131
3	1.9034	10.2226
4	2.3704	14.3061
5	3.0169	23.9499
6	3.3217	21.2698
7	3.7104	26.7637
8	4.1035	24.8261
9	4.1958	27.3538
10	5.13	42.63

Figure 4.8: Gemiddelde en standaardafwijking van de procentuele toename in reistijd van route i in vergelijking met de snelste route voor factor 10

Bij toenemende k_{10} waarde wordt de reistijd gewoonweg veel te lang in vergelijking met de snelste route, en moeten er dus al een grote hoeveelheid weggebruikers aanwezig zijn voordat dit de snelste route lijkt. Van nature heeft een doorsnee weggebruiker altijd de neiging om de op dat moment snelste route te kiezen en zo het gebruikersevenwicht te behouden. Verder (sectie 5.2) zullen we kijken wat de invloed is wanneer bepaalde weggebruikers niet overtuigd geraken van de nieuwe routering en hun eigen ding doen, en uiteindelijk zoeken we in 5.3 ook naar een manier om hardnekkig de mensen richting het system equilibrium te sturen . Echter is het natuurlijk wel de bedoeling om het system equilibrium iets natuurlijker te maken, en is een hoog verschil in reistijd met de snelste route zeer demotiverend. Willen we toch totaal verschillende routes, dan lijkt $k_{10} = 4$ een goede oplossing, de grootste daling (figuur 4.3) lijkt voorbij en het verschil blijft nog aanvaardbaar (zie tabel 4.8).

Een eerste opmerking die kan worden gemaakt is de gebruikte factor bij het berekenen van de k paden, namelijk 10. Deze werd hoog gekozen voor totaal verschillende paden, maar wat gebeurt er wanneer we deze lager laten worden en hiermee beginnen toestaan dat er eventueel goedkope paden hergebruikt worden?

Nemen we bijvoorbeeld $k_{factor} = 6$, en laten we de factor eens variëren in dezelfde voorgaand gebruikte statische rush hour situatie. Ter vergelijking was het totale reistijd verschil bij $k_{10} = 6$ ongeveer 194u 48m 26s.

Dan zien we op figuur 4.9 dat we de totale reistijd van beide modellen nog ferm kunnen verlagen door de factor te laten verlagen. Op te merken is dat bv. bij $factor = 2$ het verschil in reistijd een stuk lager ligt (131u 56m 9s). Ook is op te merken dat de totale reistijd van het user equilibrium bij $k_2 = 6$ lager ligt dan het system equilibrium bij $k_{10} = 6$, echter mag men geen verschillende factoren met elkaar vergelijken, dit is een theoretische studie waarvan er wordt uitgegaan dat er voorafgedefinieerde routes zijn, en dat elke weggebruiker gegeven de huidige wegsituatie dan zuiver volgens het user equilibrium of system equilibrium concept een van deze routes kiest.

Kijken we even wat er -in deze testsituatie- gebeurt met de paden wanneer we $factor$ laten afnemen. In figuur 4.10 plotten we het percentage van de takken die door n van de k_{factor} paden wordt gebruikt, en dit uitgemiddeld over alle OD-paren. Hierin is te zien dat er geen gigantisch verschil zit bij bijvoorbeeld $k_2 = 6$, er wordt gewoon handig hergebruik gemaakt van goedkope takken die bij een te hoge factor blijkbaar onderbelast zijn. Verder naar factor 1 toe neemt het hergebruik natuurlijk ferm toe.

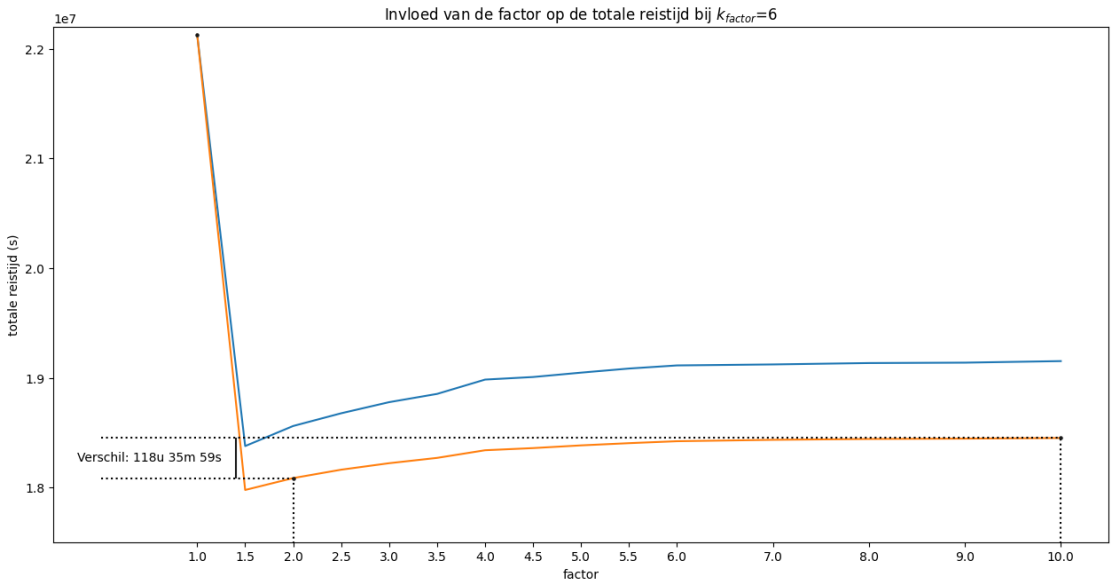


Figure 4.9: Invloed van de factor op de totale reistijden bij $k_{factor} = 6$

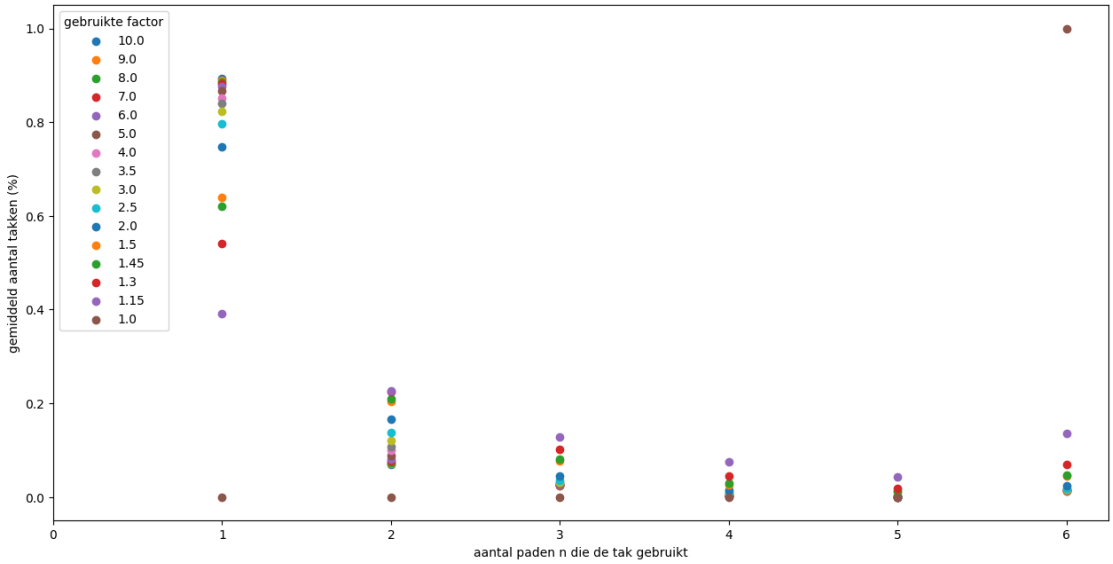


Figure 4.10: Invloed van de factor op het hergebruik van takken bij $k_{factor} = 6$

We krijgen een lagere totale reistijd voor $k_2 = 6$, dus zal het moeten zijn dat door een lagere factor goedkope takken worden hergebruikt, die ervoor zorgen dat de reistijd van extra routes minder afwijkt van de snelste, en die blijkbaar bij een hoge factor door andere takken uit de route (met een lagere capaciteit bv.) nog niet overbelast werden. Tabel 4.11 bevestigt dit.

route i	mean (%)	std dev (%)
0	0.0	0.0
1	0.1888	0.1597
2	0.3606	0.3389
3	0.536	0.5621
4	0.7535	0.9787
5	0.9193	1.5959
6	1.1237	2.2766
7	1.3387	3.6432
8	1.531	4.9997
9	1.7864	9.4385
10	1.916	8.9223

Figure 4.11: Gemiddelde en standaardafwijking van de procentuele toename in reistijd van route i in vergelijking met de snelste route voor factor 2

Bovenstaande afwijkingen zijn al een stuk meer aanvaardbaar, echter zullen er nog altijd routes aanwezig zijn die een veel hogere reistijd hebben dan de snelste, en dus demotiverend werken voor het system equilibrium. We kunnen kijken of de totale reistijd nog altijd benoemenswaardig verschilt wanneer routes die meer dan x% langer zijn dan de snelste worden gefilterd. In dat geval kan er zelfs eventueel een hogere factor worden gekozen, wil men bijvoorbeeld voor het minder fysiek belasten van wegen dat routes zo verschillend mogelijk zijn. We voegden deze optie toe aan de functie voor het berekenen van de k_{factor} snelste paden.

k	aantal routes	gem. routes per OD	user equilibrium	system equilibrium	verschil
$k_2 = 6$	158772	6	5155u 53m 26s	5023u 57m 17s	131u 56m 9s
$k_{2,filtered} = 6$	36997	1.398118	5323u 18m 16s	5266u 12m 48s	57u 5m 28s
$k_{10} = 6$	158772	6	5320u 17m 2s	5125u 28m 36s	194u 48m 26s
$k_{10,filtered} = 6$	30978	1.170660	5674u 36m 12s	5644u 21m 15s	30u 14m 57s

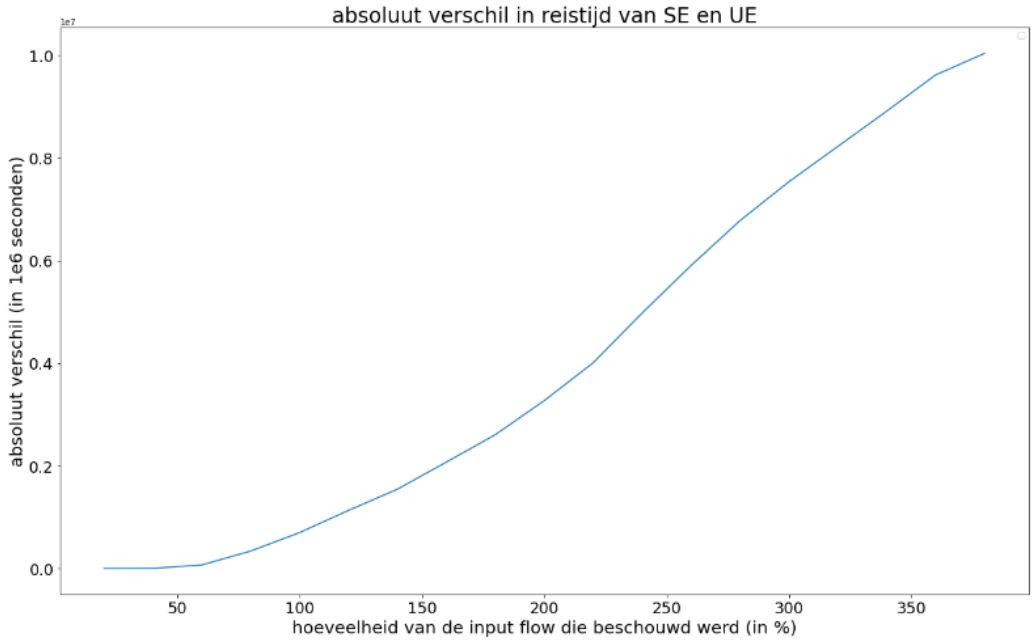
Figure 4.12: totale reistijd van beide equilibria en hun verschil bij verschillende manieren van berekenen snelste paden

In bovenstaande tabel 4.12 bekeken we het filteren van alle routes die meer dan 25% verschillen van de snelste in de gevallen van een hoge (10) en een eerder lage (2) factor. Er wordt gezien hoe het verschil in totale reistijd samen met het totale aantal routes drastisch daalt. Ook is aan de hoge toename bij het user equilibrium te zien dat deze filter te hoog ligt, vele routes zouden met dit beschouwde verkeer eigenlijk al in aanmerking genomen worden bij een user equilibrium. Het system equilibrium wordt door het overmatig filteren van de routes zelfs slechter dan wanneer we dit niet doen en een user equilibrium zijn gang laten gaan. Er kan dus wel een filter op het verschil in reistijd worden opgelegd, echter kan deze nooit zo drastisch zijn zodat weggebruikers in grote mate vrijwillig meedoen met een system equilibrium, er is dus nog altijd een hardnekkigere manier nodig, zoals in 5.3.

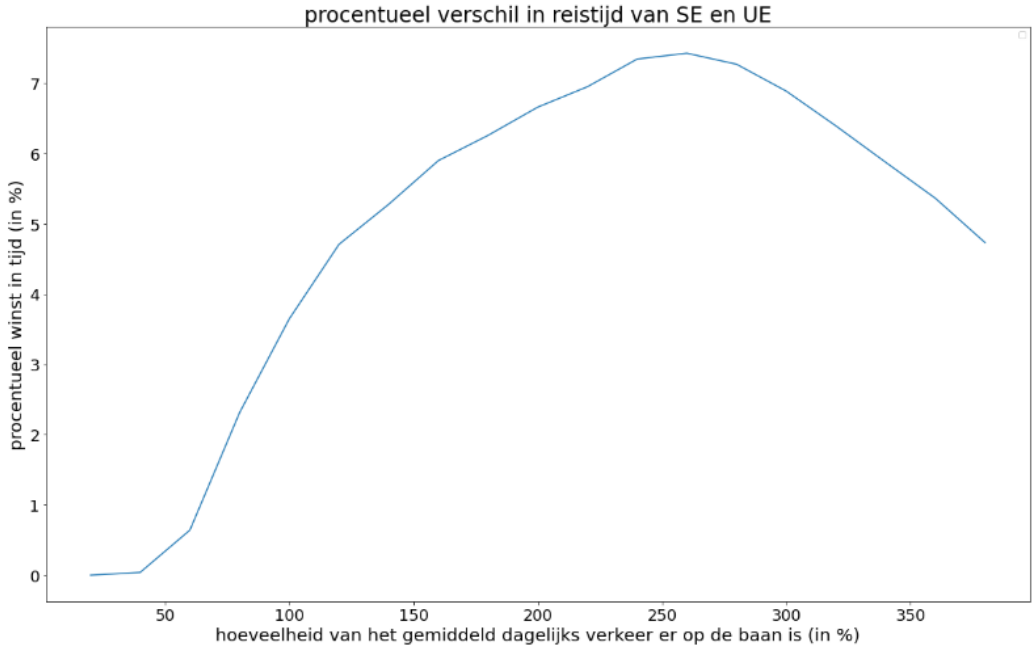
5 Resultaten en Realistische Implementatie

5.1 Resultaten

Dit deel bespreekt een centrale vraagstelling van het project: is het voordelig genoeg om het system equilibrium, vergeleken met de user equilibrium, wel degelijk door te voeren?



(a) Absoluut vershil in reistijd volges hoeveelheid van de statische flow



(b) Procentueel verschil in reistijd volgens hoeveelheid van de statische flow regio Sint-Niklaas

Figure 5.1: Invloed voor het UE en SE apart bekeken

De figuur 4.3 geeft aan dat de totale reistijd bij het system equilibrium duidelijk lager is dan de totale reistijd bij het user equilibrium. Het is echter wel belangrijk om te bepalen in welke context het system equilibrium werkelijk bruikbaar is: vanaf welke hoeveelheid procentuele tijdswinst is het lucratief genoeg? Figuren 5.1a en 5.1b geven het verschil in reistijd tussen het system equilibrium en het user equilibrium wanneer een toenemend procent van de input flow wordt beschouwd. Het is duidelijk zichtbaar dat er weinig verschil is tussen de reistijden van beide equilibria voor een quantiteit flows die veel lager is dan die van het initieel netwerk. Vanaf ongeveer 75% begint echter wel een groeiend verschil tussen de totale reistijden op te treden. De gebruikte data

is, zoals voorheen vermeld, statische data en de flows zijn dus ook een dagelijks gemiddelde. Hieruit volgt dat het system equilibrium enkel relevant blijkt te zijn in de spitsuren aangezien bij andere uren de moeite en kost nodig voor het implementeren van dit systeem niet opweegt tegen de minieme $< 2\%$ verbetering in reistijd. Daarentegen is er wel een duidelijke impact bij hogere percentages: de figuur 5.1b geeft aan dat de verbetering in reistijd kan oplopen tot 7.2% in de spitsuren (waarbij het aantal flows hoger is dan de gewoonlijke 100%). Voor te hoge flows begint de procentuele winst in totale reistijd echter weer te dalen. Dit is te wijten aan het feit dat er bij te veel verkeer congestie optreedt in het hele netwerk. In de praktijk zullen dergelijke hoge flows echter zelden tot nooit bereikt worden.

5.2 Stabiliteit

Tot nu toe zagen we vele wiskundige voordelen van het toepassen van het system equilibrium. Echter werd volgend groot probleem in sectie 4.2.2 al even kort aangehaald, is het wel mogelijk om zomaar iedereen routes op te leggen? Het user equilibrium houdt zichzelf in stand, aangezien er geen alternatieve paden zijn die beter lijken. Het system equilibrium daarentegen is individueel oneerlijk en tracht soms sommige weggebruikers op een veel langere route te sturen (om zo een minimale totale reistijd te verkrijgen). In sommige situaties, zoals wanneer er weinig verkeer (en dus weinig file) is, kan het dat een weggebruiker niet overtuigd is en dus niet de voorgestelde omweg neemt. Hierdoor is er geen garantie dat elke weggebruiker de routering, voorgesteld door het system equilibrium, volgt. Daarom is het van essentieel belang om te onderzoeken wat de impact is van de participatiegraad van weggebruikers in het system equilibrium op de totale reistijd van het netwerk. Bij een lagere participatie zal de totale reistijd hoger zijn dan die van het system equilibrium. De figuur 5.2 toont de vermindering in reistijd ten opzichte van het huidig selfish wardrop equilibrium. Er is een quasi-lineair verband is tussen de participatiegraad en het effect van het equilibrium. Indien slechts 20% van de weggebruikers weigert deel te nemen, resulteert dit nog steeds in een gemiddelde tijdwinst van 3% in ons geval.

De stabiliteit werd bekeken door gelijkmatig over de routes een procent toe te kennen volgens het user equilibrium. Daarna werden, gegeven deze flows al, de overige toegekend volgens het system equilibrium. In principe passen beide equilibria zich opnieuw aan en zal het itereren hiervan een beter resultaat geven.

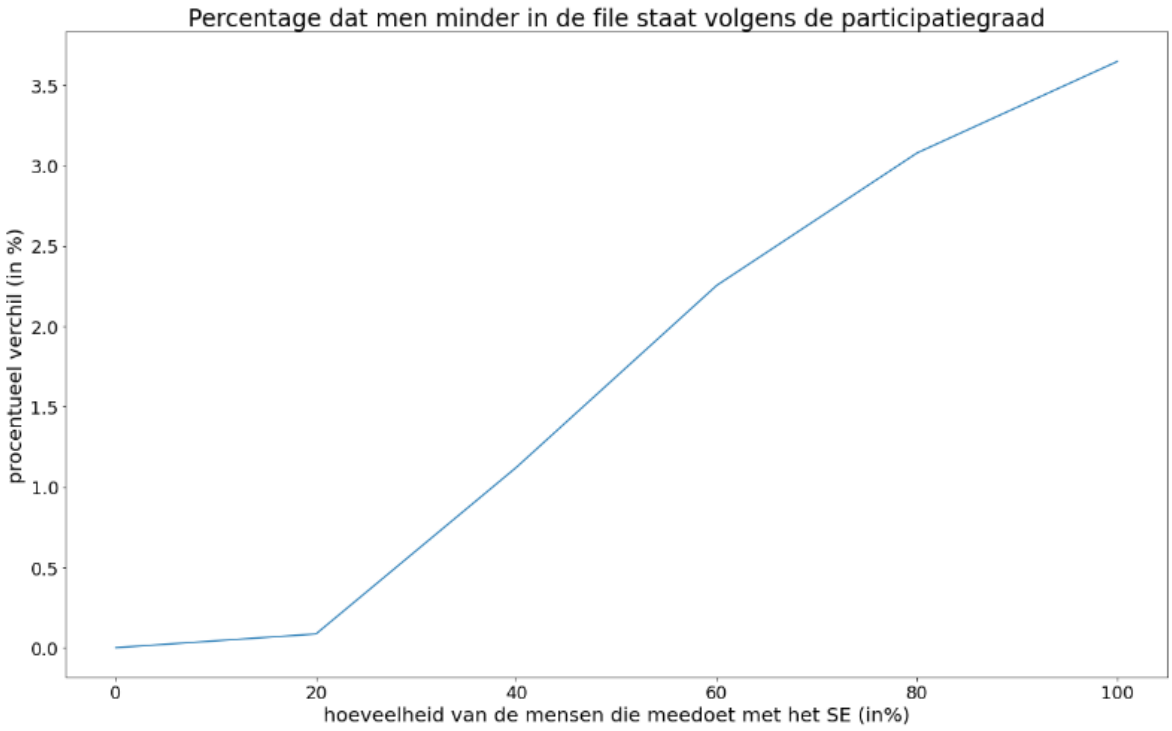


Figure 5.2: procentuele tijdswinst volgens de participatiegraad

5.3 Realistische Implementatie

Deze vaststelling is wel hoopgevend: degelijk procent weggebruikers het system equilibrium volgt, zal de totale reistijd aanzienlijk dalen, wat de -in de inleiding vermeldde- gewenste voordelen met zich meebrengt. Er is echter wel nog nood aan een manier waarop toch zoveel mogelijk mensen kunnen worden gestimuleerd tot het volgen van de alternatieve routering.

Waarom mensen spontaan een user equilibrium in gang zetten is evident: de individuele reistijd wordt gezien als een kost en hierdoor tracht iedereen dit voor zichzelf te minimaliseren. We kunnen dus op zoek gaan naar een beloning die opweegt tegen de beloning van minimale individuele reistijd, of met andere woorden een kost die de mensen onbewust het gewenste system equilibrium laat creeren. Een financiële kost zou dit kunnen verwezenlijken.

Bij het user equilibrium kiest iedereen voor, gegeven de weggebruikers op elke route, de voor hem snelste route, maar hierbij veroorzaakt het ook een -kleine- extra reistijd bij de andere weggebruikers op deze route, wat wegens de hoge flows op dezelfde wegen leidt tot een veel hogere totale reistijd. Koos deze weggebruiker echter voor een alternatieve minder belaste route, zoals de weggebruiker naar zou gestuurd worden volgens het system equilibrium, dan is de totale extra kost voor de weggebruikers op de wegen op deze route geïntroduceerd door de keuze van de weggebruiker vele malen lager. Men kan dus een soort tax op de marginale kost invoeren: dit is hoeveel een automobilist de totale reistijd op de tak laat stijgen, en dan kan men de automobilist taxeren naargelang deze marginale kost, een soort additionele-reistijdverheffing in plaats van kilometerverheffing.

Bekijkt men dit even wiskundig. Per tak zijn er nu twee factoren die de weggebruiker in rekening brengt bij zijn keuze tot het nemen van deze tak, dit kunnen we dus modelleren als twee kosten die in rekening worden gebracht, de reistijd op deze tak gegeven de flow al aanwezig (zie formule 3.3.4), en de marginale kost:

$$MK = \frac{\partial t_a}{\partial f_a} \tag{5.3.1}$$

En deze marginale kost wordt voor elke weggebruiker op de tak geintroduceerd, waardoor de nieuwe beschouwde kostenfunctie per tak gedefinieerd wordt als:

$$c(f_a) = f_a * \frac{\partial t_a}{\partial f_a} \tag{5.3.2}$$

Gegeven deze nieuwe kostenfunctie per tak zal er zich nu opnieuw een user equilibrium vormen, aangezien de weggebruiker opnieuw theoretisch gezien telkens voor de goedkoopste route zal kiezen. Bekijken we daarom eens wat er gebeurt wanneer men kijkt naar de vergelijking die statische flows verdeelt om een user equilibrium te volgen:

$$\begin{aligned}
&min_{\vec{f}} \sum_{a \in A} \int_0^{f_a} c_a(x)dx \\
&= min_{\vec{f}} \sum_{a \in A} \int_0^{f_a} t_a(x) + x * \frac{\partial t_a}{\partial x} dx \\
&= min_{\vec{f}} \sum_{a \in A} (\int_0^{f_a} t_a(x)dx + \int_0^{f_a} x * \frac{\partial t_a}{\partial x} dx)
\end{aligned}$$

Met behulp van partiële integratie bekomen we:

$$\begin{aligned}
&min_{\vec{f}} \sum_{a \in A} (\int_0^{f_a} t_a(x)dx + \int_0^{f_a} x * \frac{\partial t_a}{\partial x} dx) \\
&= min_{\vec{f}} \sum_{a \in A} (f_a * t_a(f_a) - \int_0^{f_a} x * t_a(x)dx + \int_0^{f_a} x * t_a(x)dx) \\
&= min_{\vec{f}} \sum_{a \in A} f_a * t_a(f_a)
\end{aligned}$$

En dus zal bij een zuiver user equilibrium met onze additionele-reistijdverheffing een minimale totale reistijd ontstaan. Op deze manier kan men dus ervoor zorgen dat er een zelfonderhoudend system equilibrium bekomen wordt, net als er op dit moment een user equilibrium in stand wordt gehouden met individuele reistijden als katalisator. Echter kunnen hier nog een aantal bedenkingen bij worden gemaakt.

Eerst en vooral zijn er de extra kosten voor de weggebruikers. Op zich wordt het system equilibrium dus onrechtstreeks nog geforceerd via deze taksen, en kan men conform met de conclusies uit 5.1 besluiten de kosten te laten vallen bij een laag weggebruik.

Verder blijft het unfair, het kan zijn dat telkens bij dezelfde weggebruiker de verkeerssituatie zo is dat snelle routes duur zijn. Om dit op te lossen kan er eventueel een unfairness-factor worden bijgehouden per weggebruiker (bv. $\prod_{r \in R} \frac{reistijd\ r}{reistijd\ snelste\ alternatieve\ route\ voor\ r}$ met R de verzameling van afgelegde routes in een jaar). Er zou dan bijvoorbeeld een vergoeding kunnen worden uitgedeeld met het geld die wordt uitgespaard aan files voor de mensen die toevallig sterk benadeeld werden door het system equilibrium, en dus een hoge unfairness factor hebben.

Om af te sluiten misschien wel het grootste nadeel, dit is dat iedereen via gps dient te rijden om de opvolging mogelijk te maken, wat de gewoonlijke strijd tussen de digitale wereld en de privacy met zich meedraagt..

6 Conclusie

In het kader van dit project lag de focus op de studie van user equilibrium en system equilibrium in het domein van routering. We hebben vastgesteld dat in specifieke situaties, vooral wanneer congestie optreedt, het toepassen van system equilibrium-routering aanzienlijke voordelen kan opleveren.
Het positieve effect van system equilibrium-routing is echter afhankelijk van diverse variabelen. Op technisch niveau kunnen factoren zoals het aantal beschikbare kortste paden en de manier waarop deze paden gekozen worden invloed hebben op dit effect. Het is van groot belang om deze technische aspecten grondig te overwegen en af te stemmen op de specifieke omstandigheden en doelen van het routeringsprobleem.
Daarnaast spelen bredere sociale factoren eveneens een cruciale rol bij het realiseren van het gewenste positieve effect van system equilibrium-routing. De betrokkenheid van weggebruikers zelf vormt een essentiële factor voor de stabiliteit en het succes van het systeem. Het is van belang om methodes te ontwikkelen om de participatie van weggebruikers te bevorderen en hen te betrekken bij het besluitvormingsproces omtrent routering. Dit vereist een holistische benadering waarin sociale, economische en beleidsmatige aspecten geïntegreerd worden.
Het is van essentieel belang om de technische en sociale aspecten nauwkeurig op elkaar af te stemmen om de gewenste positieve effecten te bereiken. Dit vereist voortdurend onderzoek en een multidisciplinaire aanpak om een efficiënt en evenwichtig routeringssysteem te realiseren.

7 Referenties

References

[1] E. Angelelli and M. G. Speranza V. Morandi, M. Savelsbergh. System optimal routing of traffic flows with user constraints using linear programming. 2021.

[2] M. Beckmann, C. B. McGuire, and C. B. Winsten. Studies in the economics of transportation. 1956.

[3] I. Cidon, R. Rom, and Y. Shavitt. Analysis of multi-path routing. 1999.

[4] Jose R. Correa and Nicolas E. Stier-Moses. Wardrop equilibria. *Encyclopedia of Operations Research and Management Science*, 1996.

[5] David Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28, 1998.

[6] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. *Automata, Languages and Programming: 29th International Colloquium, ICALP 2002 Málaga, Spain, July 8–13, 2002 Proceedings*, 1:123–134, 2002.

[7] M. Frank and P Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, (3):95–110, 1956.

[8] P. Marcotte, S. Nguyen, and A. Schoeb. A strategic flow model of traffic assignment in static capacitated networks. *Operations Research*, 2(52):191–212, 2004.

[9] V Morandi. Bridging the user equilibrium and the system optimum in static traffic assignment: how the cooperation among drivers can solve the congestion problem in city networks. 2021.

[10] Rolf H. Möhring, Andreas S. Schulz, and Nicolas Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. 2004.

[11] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49:236–259, 2002.

[12] Tim Roughgarden. Selfish routing and the price of anarchy. 2006.

[13] H. Tajtehranifard, A. Bhaskar, N. Nassir, M. M. Haque, and E. Chung. A path marginal cost approximation algorithm for system optimal quasi-dynamic traffic assignment. *Transportation Research part C: Emerging technologies*, 88:91–106, 2018.

[14] Yi Wang and Szeto W. Y. Multiobjective environmentally sustainable road network design using pareto optimization. *Computer-Aided Civil and Infrastructure Engineering*, pages 1–24, 2017.

8 Appendix

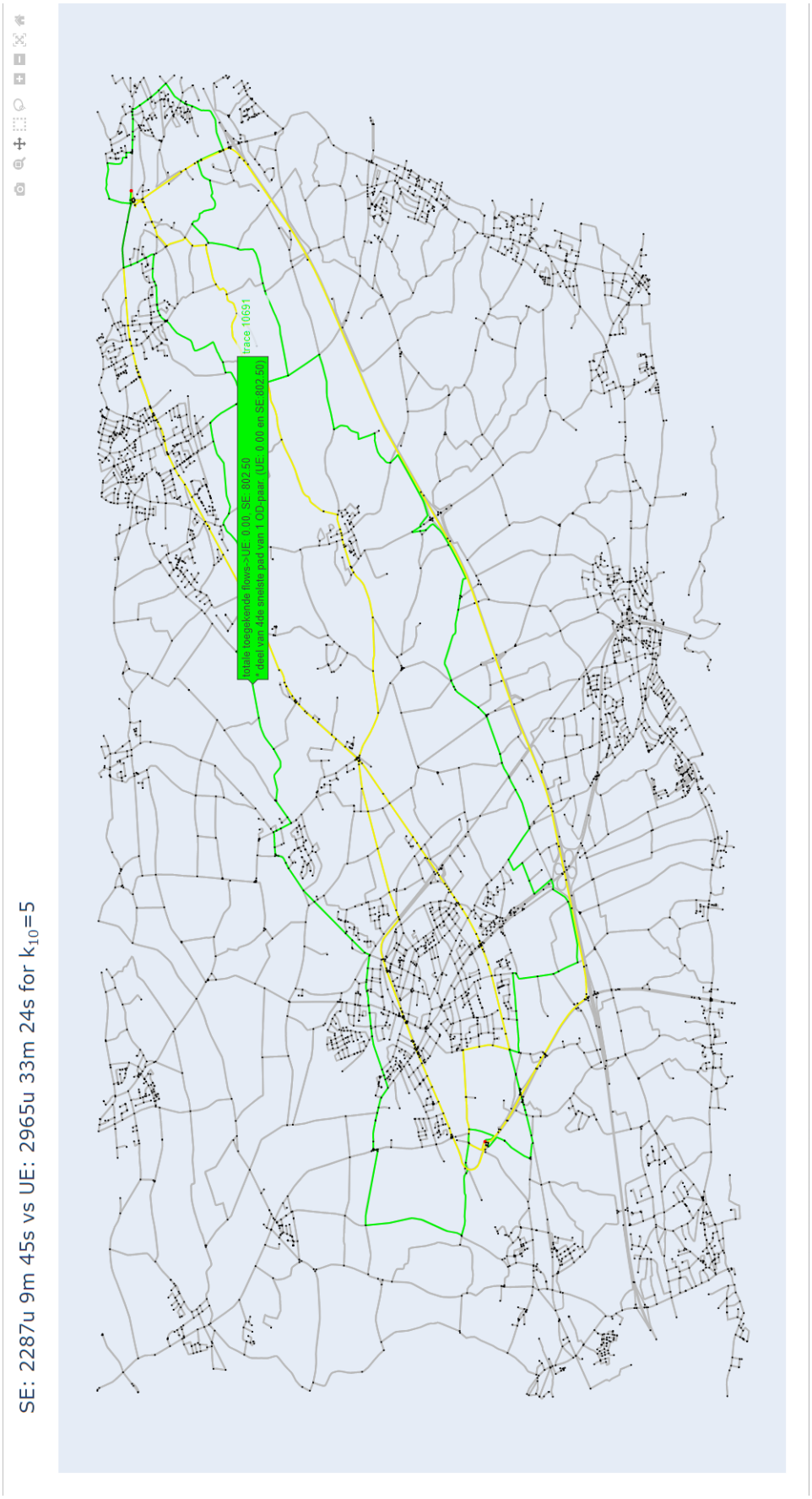


Figure 8.1: Afbeelding van de visualisatie