

Vialis



Behaviour-based Simulation
2020_TAAI-V2BS-19_2_V
Vialis groep II / B
22/01/2021

Floris Videler	1758374
Niels Bijl	1754339
Jasper van Loon	1756617
Kai ter Horst	1762818

Inhoudsopgaven

Stakeholders	4
Vialis	4
Verkeersdeelnemers	4
Probleemstelling	5
Vraagstelling	6
Hoofdvraag	6
Deelvragen	6
Uiteenzetting van de gekozen scenario's	7
Visualisatie	7
Simulatie I huidige verkeerslichtlogica	7
Simulatie II	8
Simulatie III	8
Oplossing voor de probleemstelling	9
Visualisatie	10
Simulatie	11
Agents	11
Auto's	11
Lussen en verkeerslichten	11
Initialiseren	11
Data	11
Simulatie	12
Tools	13
XML-tool	13
Read-JSON-tool	13
Resultaten	14
Visualisatie	14
Simulatie	15
Variabelen uitleg	15
Het verschil in tijden waarin wij onze simulatie gedraaid hebben	15
Waar heeft het verlengen van het groene licht op het drukste verkeerslicht het meest effect op?	16
Het effect van het verlengen van het groene licht van het verkeerslicht op de gemiddelde aantal steps om voor een automobilist het kruispunt over te rijden.	17
Het effect van het verlengen van het groene licht van het verkeerslicht op de gemiddelde aantal steps dat een automobilist voor een rood verkeerslicht staat te wachten	17

Validatie van de simulatie	17
KPI's	18
Hoeveel secondes doet een auto er gemiddeld over om door de simulatie te rijden.....	18
Hoeveel secondes wacht een auto gemiddeld voor een verkeerslicht	18
Hoeveel procent lus activatie kan er verklaard worden.....	18
Resultaten KPI's voor klantprobleem	19
Hoeveel secondes wacht een auto gemiddeld voor een verkeerslicht	19
Hoeveel secondes doet een auto er gemiddeld over om door de simulatie te rijden.....	19
Validatie	20
Beantwoording van de vraagstelling	21
Hoofdvraag.....	21
Deelvraag I	21
Deelvraag II	21
Advies	22
Bijlage	23
Bijlage A.....	23
Bijlage B.....	24
Bijlage C	25
Bronnen	26

Stakeholders

Vialis

Vialis is in dit project de opdrachtgever. Vialis werkt in Nederland binnen het domein van infrastructuur en mobiliteit als solution provider voor haar klanten. Klanten zijn bijvoorbeeld andere verkeersdeskundigen of gemeentes. Vialis ontwikkelt onder meer oplossingen voor het managen van verkeer en realiseert ze intelligente oplossingen met complexe systemen.

Vialis is onze hoofdstakeholder, omdat het de opdrachtgever is. Daarnaast leveren ze de data van de twee kruispunten die nodig zijn voor het project. Zowel de sensorinformatie als de informatie van de verkeerslichten. Ook geven ze inzicht hoe het eindproduct eruit moet komen te zien.

Hun doelstelling voor dit project is om erachter te komen of de verkeerslichten zoals ze huidig gebruikt worden optimaal zijn voor de doorstroom. Ook willen ze een visualisatie kunnen zien van het kruispunt waarin real-life data wordt gevisualiseerd.

Verkeersdeelnemers

Het verkeer moet uiteindelijk van ons systeem winst maken. Verkeersdeelnemers staan vaak te lang te wachten voor een rood licht of staan in de file. Daarom zouden deze verkeersdeelnemers een beter verkeerslichtsysteem willen, of een manier hebben om verkeer te omzeilen. Technisch gezien is het product voor iedereen in Nederland die in het verkeer rijdt, zodat ze zoveel mogelijk gebruik kunnen maken van de innovatie.

Probleemstelling

Vialis heeft nog geen volledig zicht op wat er gebeurt op haar kruispunten. Daarom wil Vialis graag een visualisatie met real-life data van de twee kruispunten.

De doorstroom op de twee gegeven kruispunten zijn nog niet perfect. Dit levert irritaties op bij het verkeer dat gebruik maakt van de kruispunten. Vialis is benieuwd of er verbetering in de doorstroom zit doormiddel van aanpassingen in de verkeerslichten.

Vraagstelling

Hoofdvraag:

1. Is het mogelijk om de doorstroom op de twee gegeven kruispunten te verbeteren, doormiddel van het aanpassen van de logica van het verkeerslicht?

Deelvragen:

1. Kunnen wij de verkregen data van Vialis plotten (XML hoe het kruispunt eruitziet) en de data van de lussen/ verkeerslichten visualiseren wanneer ze aan en uit gaan (CSV wanneer welke lus/verkeerslicht geactiveerd wordt)?
2. Kunnen wij een simulatie maken waarin een gegeven kruispunt wordt gesimuleerd, met de verkeerslichten en de sensoren, waar willekeurige auto's kunnen rijden?

Uiteenzetting van de gekozen scenario's

Visualisatie

De visualisatie bestaat uit drie delen. Deze drie delen worden onafhankelijk van elkaar op een canvas gevisualiseerd. De data bevatten:

- De posities van alle attributen op de kruispunten (denk hieraan de verkeerslichten, stopstrepen, hoe de wegen lopen en de lussen in de weg)
- De status van elk verkeerslicht en alle lussen.
- Gps-data die wij zelf gegenereerd hebben door zelf op de kruispunten te rijden met een auto.

Alle attributen worden geplot op een canvas. Elk attribuut heeft zijn eigen icoon waarin duidelijk wordt om welk attribuut het gaat.

Wanneer je de visualisatie start, worden de real-life data met daarin de status van de verkeerslichten en lussen weergegeven. Je merkt dat de lussen in de weg oplichten als de status verandert. De verkeerslicht en veranderen van kleur, bijvoorbeeld van rood naar groen. Als een verkeerslicht op rood staat, is te zien dat op het kruispunt de weg niet te zien is. Pas als het verkeerslicht op groen komt te staan, wordt de weg weergegeven.

Daarnaast is onze eigen gegenereerde gps-data te zien. Er is namelijk een bewegende auto te zien op het canvas. Die auto is de visualisatie van de gps-data die wij zelf gegenereerd hebben door op de kruispunten te rijden. Als wij een lus naderen, is te zien dat die oplicht. Ook als wij voor een rood verkeerslicht staan is te zien dat wij op het canvas stilstaan en dat wij weer verder rijden als het verkeerslicht groen wordt.

Simulatie I huidige verkeerslichtlogica

In dit scenario hebben wij ons best gedaan om de realiteit te simuleren. Dit doen wij om een baseline/nulmeting te kunnen leggen.

Wij maken hier gebruik van de data van Vialis. Wij gebruiken de status van de verkeerslichten en lussen zoals we die gekregen hebben. De status van de verkeerslichten en lussen staan vooraf dus al vast gelegd wanneer en wat ze doen.

Bij het activeren van de eerste lus op de weg wordt een auto gegenereerd. Deze auto weet waar hij naar toe gaat en dat hij niet door rood mag rijden. De auto's proberen zo snel mogelijk hun targetpositie te behalen.

Deze en alle andere simulaties zijn gedraaid is drie verschillende tijdsintervallen:

1. Ochtendspits (07:00 - 09:00)
2. Middag (13:00 – 15:00)
3. Avondspits (16:00 – 18:00)

Simulatie II

In dit scenario gaan wij kijken wat het effect is als wij het verkeerslicht op de drukste weg 10% langer op groen laten staan.

Ook hier gebruiken wij weer de data van Vialis. Dit doen wij zodat onze simulatie zo dicht mogelijk bij de realiteit blijft. Ook omdat zo een betere vergelijking gemaakt kan worden met de baseline.

De auto's worden op dezelfde manier gegenereerd zoals we dat doen in simulatie I en behouden dezelfde logica.

Simulatie III

In dit scenario gaan wij kijken wat het effect is als het verkeerslicht op de drukste weg 20% langer op groen staat.

Ook hier gebruiken wij weer de data van Vialis op dezelfde manier als in Simulatie II. Dit geldt ook voor de auto's.

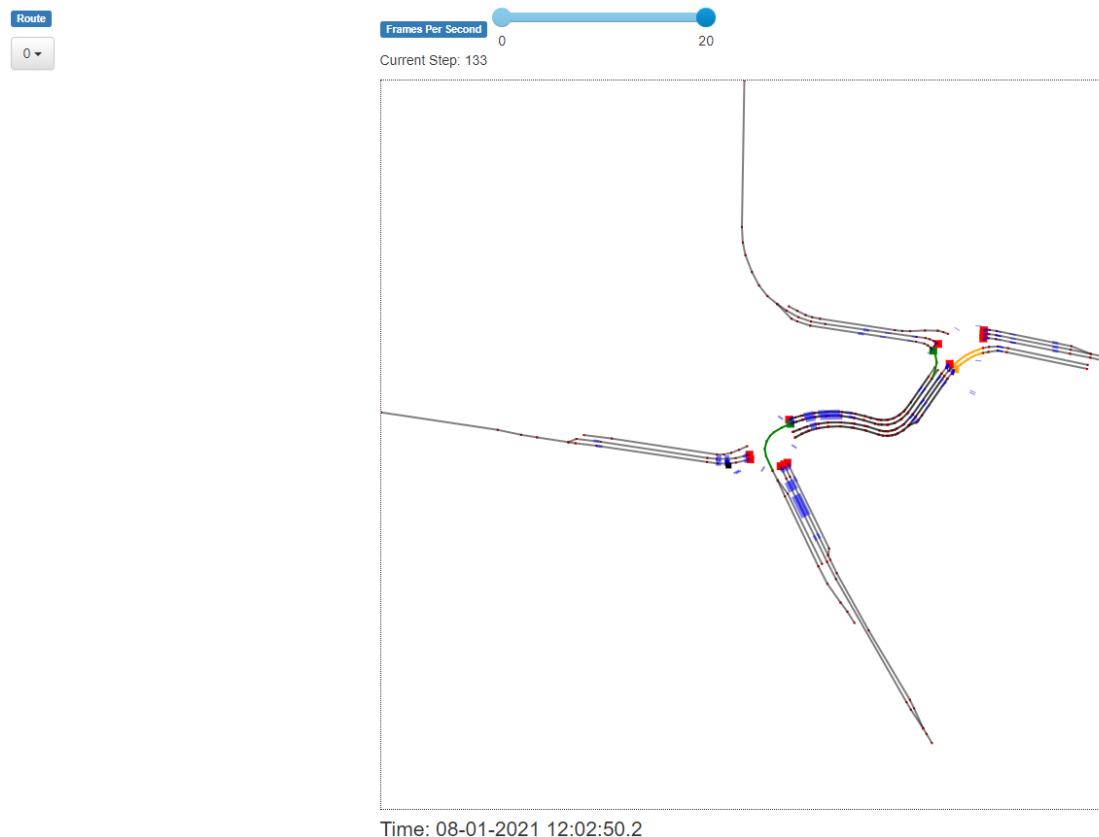
Oplossing voor de probleemstelling

Voor de simulatie en visualisatie hebben wij de python-library Mesa gebruikt. Hiermee hebben wij een agent-based simulatie gemaakt en een schematische visualisatie. In de simulatie is het mogelijk om de verschillende scenario's te simuleren. In de visualisatie is het mogelijk om verschillende aanvliegroutes te visualiseren.

Visualisatie

De visualisatie is te runnen door het python-bestand "run_visualization.py" te runnen. Hier zijn verschillende python-eisen aan verbonden, deze zijn in te zien in de bijbehorende GitHub.

Onze zelf gegenereerde gps-data bestaan uit verschillende routes, omdat wij er verschillende keren overheen zijn gaan rijden. In de GUI van Mesa is linksboven te zien dat je een keuze kunt maken welke route je wilt zien. Als je de route gekozen hebt, is het noodzakelijk om op reset te klikken, zodat de nieuwe route kan worden ingeladen. Het inladen kan even duren.



Bovenin zijn de Frames Per Second (FPS) aan te passen. Bij 1 FPS wordt elke seconde één stap gedaan. Eén stap is 0.1 seconden in de visualisatie, dus als je de visualisatie in real-time wilt bekijken moet je de FPS op 10 zetten.

Het is ook mogelijk om met de hand stap voor stap elke stap uit te voeren. Dit kan door op Step te klikken, dit doet dus één stap.

Door op Start te klikken begint de GUI met het afspelen van de visualisatie met de gekozen instellingen.

Onderaan is de datum en tijdstip van de afgespeelde data te zien.

Simulatie

Agents

De kern van een agent-based simulatie zijn de agents. Dit kun je zien als de objecten die deelnemen aan de simulatie. Ze kunnen zelf denken en beslissingen nemen. Omdat onze simulatie is gemaakt met Mesa is bijna alles een agent. Alles wat wil bewegen, van kleur wil veranderen of iets in de simulatie wil doen is voor Mesa een agent.

Auto's

De auto's in de simulatie zijn agents die zelf denken. De auto's worden gegenereerd als de eerste lus van de weg wordt geactiveerd. Voor de functionaliteit van de auto's hebben wij een variant van het Nagel–Schreckenberg^[1] model gebruikt (Luna-Benoso et al., 2013b). Hoe de auto's precies per step denken is te zien in de flow-chart (zie bijlage A). De maximale snelheid van de auto's is 50km/h en ze trekken op met een snelheid van 12.5km/h per seconde.

Lussen en verkeerslichten

De lussen en verkeerslichten krijgen continu hun staat binnen door middel van de data van Vialis. Afhankelijk van hun staat veranderen ze van kleur. De verkeerslichten kunnen veranderen in rood, groen en oranje. De lussen worden donkerblauw als ze geactiveerd worden, normaal zijn ze licht blauw.

Initialiseren

Als de simulatie start, wordt als eerste de data, die geformatteerd zijn door onze XML en JSON tools, ingeladen. De data worden gebruikt om het kruispunt of de kruispunten op te bouwen. Eerst zullen alle wegen en verkeerslichten worden gemaakt en daarna worden alle sensoren geplaatst. Naast de data worden ook de instellingen voor het model ingeladen, deze instellingen worden gebruikt om verschillende simulaties te kunnen maken.

De posities van de wegen worden uitgelezen uit de data die zijn verwerkt door onze tools. Eerst worden de zogeheten “nodes” neer gezet, tussen deze nodes zal daarna een weg worden gemaakt. De auto's kunnen dan met behulp van de nodes hun route vinden naar hun eind bestemming.

Voor meer informatie over het Initialiseren van het model zie bijlage B

Data

Voor het verzamelen van de data wordt een zogeheten batch runner gebruikt. Deze is speciaal gemaakt voor het veel uitvoeren van verschillende simulaties. Mesa beschikt zelf over een batch runner, echter paste deze niet helemaal goed op ons onderzoek en hebben wij er zelf eentje geschreven. De batch runner krijgt mee welk verkeerslicht langer op groen moet blijven en vanaf hoe laat de simulatie moet draaien. Deze data wordt daarna geëxporteerd naar een CSV bestand en daarna geanalyseerd te worden.

Simulatie

De verschillende simulaties zijn uitgevoerd met behulp van de eerdergenoemde batch runner. Deze simulaties gebruiken de instellingen zoals beschreven in de uitleg over de scenario's. Deze simulaties zijn allemaal gedaan over hetzelfde kruispunt in Den Bosch: "BOS210". Dit kruispunt is ook het kruispunt dat standaard wordt ingeladen in de simulatie en visualisatie.

Tools

XML-tool

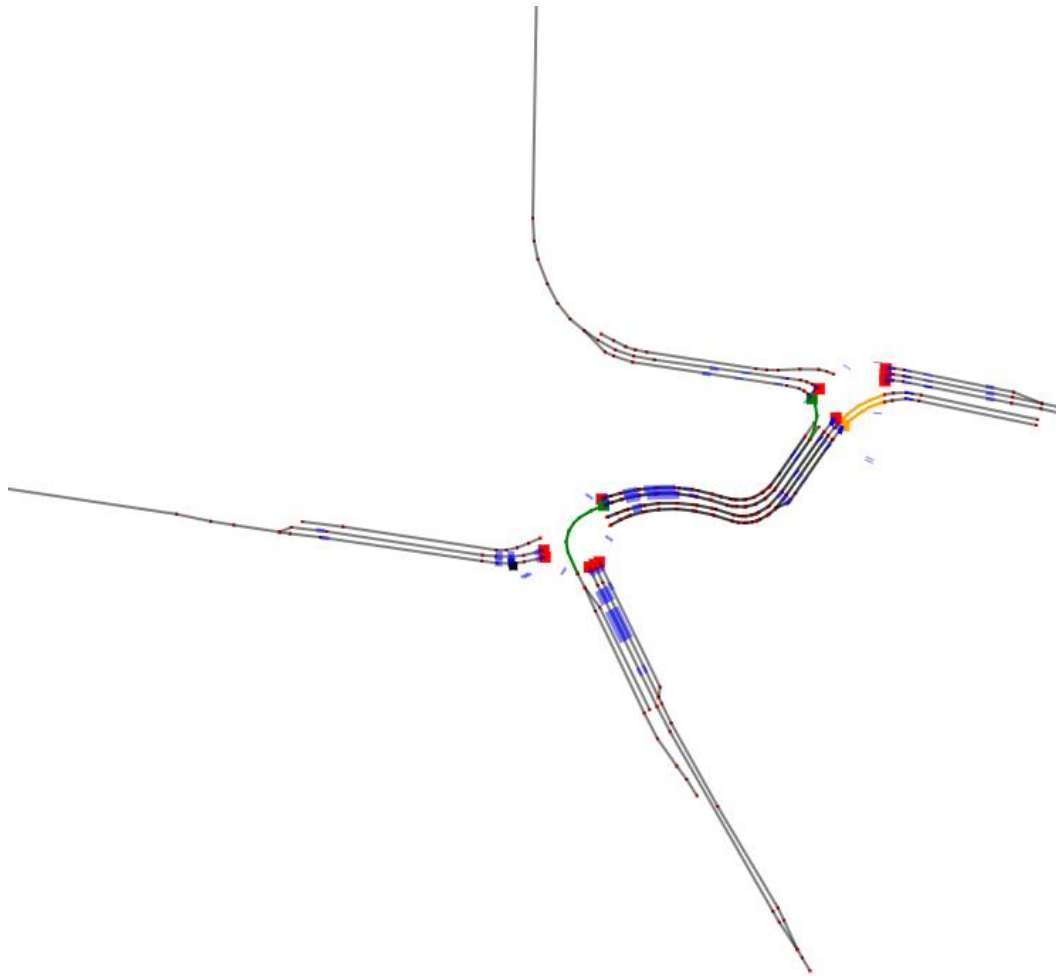
Van Vialis kregen wij verschillende XML-files aangeleverd. Hierin staat data over de kruispunten zoals de locaties van sensoren, verkeerslichten en relaties tussen wegen. Er is een XML-tool geschreven om al deze data eruit te halen zodat het gebruikt kan worden bij de visualisatie en simulatie. Het script loopt door de XML-file heen en zoekt de informatie die wij nodig hebben. De informatie wordt vervolgens geëxporteerd naar de JSON-file laneset of sensorlist. Sommige data konden niet direct gebruikt worden dus die werd eerst geformatteerd. Een voorbeeld hiervan is de latitude en de longitude. Beide waardes waren een reeks getallen dus die is eerst omgevormd door een punt op de juiste plek te zetten.

Read-JSON-tool

Vanuit de JSON-files gaan wij door met het verwerken van de informatie. Als eerste worden alle regels van de JSON-bestanden weer gelezen. Zo krijgen wij alle coördinaten in een lijst. Eerst worden alle x- en y- coördinaten in een globale lijst geplaatst. Dan worden de coördinaten van ons onderzoek in den Bosch erin gezet, om ze vervolgens te normaliseren. Zo zullen alle coördinaten passen op een één bij één vierkant, zodat de coördinaten netter en overzichtelijker geplot worden. Omdat alle data tussen nul en één zijn genormaliseerd is het voor ons een makkelijk om deze getallen op een canvas te plotten zo groot als wij zelf willen. En tot slotte zetten wij alles weer op zijn plek, en zetten wij de data in nieuwe JSON-bestanden, om ze vervolgens te gebruiken in ons model.

Resultaten

Visualisatie



In de afbeelding hierboven is een schematische weergave van de twee kruispunten te zien. Deze twee kruispunten zijn gemaakt met de data van Vialis, beide kruispunten liggen in Den Bosch. De namen van de kruispunten zijn: “BOS210” en “BOS211”.

Deze kruispunten hebben wij geplot met gebruik van Mesa. Zo zijn alle coördinaten geplot zodat ze te zien zijn. De wegen worden weergegeven als grijze lijnen. De rode punten die daarop liggen zijn de nodes om de weg te plotten. Ook zie je blauwe vakken op de weg, dat zijn de lussen die in de weg verwerkt zitten. Deze lussen verschillen in lengte en dat zie je in de afbeelding terugkomen. Bij de conflictgebieden, het gedeelte van het kruispunt waar de wegen kruisen, zie je dat er geen wegen zijn. Die worden alleen weergegeven als je er overheen mag rijden. Deze wegen worden dan in de kleur die het verkeerslicht op dat moment heeft weergegeven. Verkeerslichten die op rood staan worden weergegeven als rode vierkanten.

Simulatie

Variabelen uitleg:

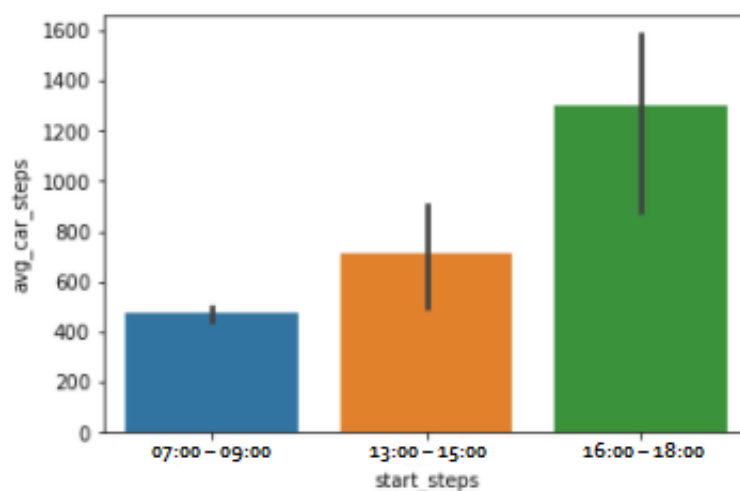
light: Het verkeerslicht 0/10/20 % langer op groen laten

start_steps: (07:00 – 09:00), (13:00 – 15:00) en (16:00 – 18:00)

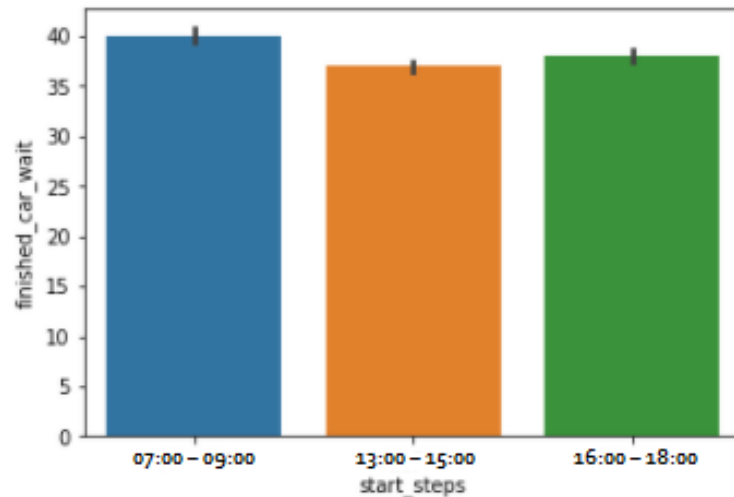
avg_car_steps: Hoeveel steps het gemiddeld kost om over het kruispunt te rijden

finished_car_wait: Hoelang een auto gemiddeld voor een rood verkeerslicht wacht.

Het verschil in tijden waarin wij onze simulatie gedraaid hebben



Er is hier te zien dat het tijdens de ochtendspits niet heel lang duurde voordat een auto over het kruispunt is gekomen. Het duurde iets langer in de middag. Maar het kruispunt oversteken duurde duidelijk het langst tussen 16:00 en 18:00.



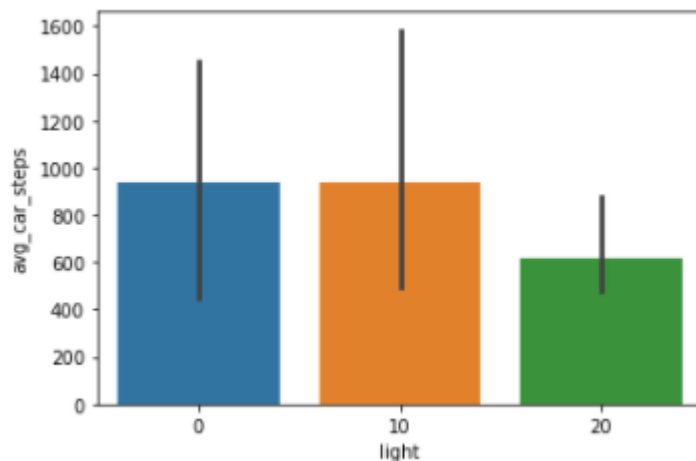
Hier is te zien dat de tijd die de agents stonden te wachten voor een rood verkeerslicht tussen alle drie de tijdstippen ongeveer even lang is.

Waar heeft het verlengen van het groene licht op het drukste verkeerslicht het meest effect op?

light	
light	1.000000
start_steps	0.000000
avg_car_steps	0.319575
finished_car_wait	0.022501

In deze correlatiematrix is te zien dat het verlengen van de tijd dat het groene licht brandde van het verkeerslicht 32% correlatie heeft met hoelang het duurt om het kruispunt over te steken en 2% correlatie met hoelang je voor een rood verkeerslicht staat te wachten.

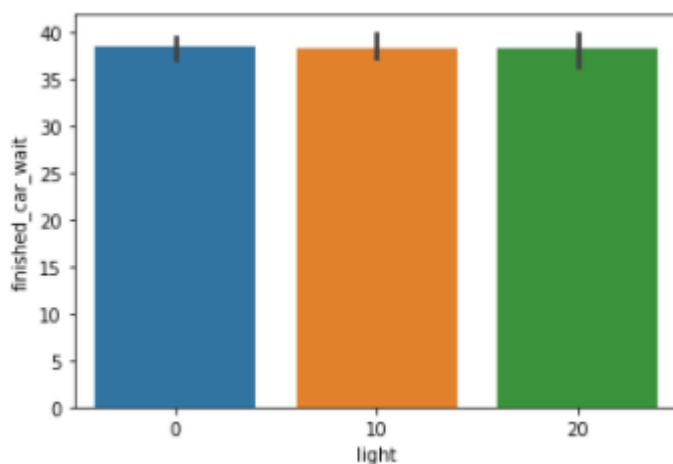
Het effect van het verlengen van het groene licht van het verkeerslicht op de gemiddelde aantal steps om voor een automobilist het kruispunt over te rijden.



Hier is te zien dat het verlengen van 10% en de baseline dezelfde resultaten geven over hoelang het gemiddeld duurt om het kruispunt over te steken. Maar het verlengen van het groene licht met 20% wel een effect heeft op hoelang het gemiddeld duurt om het kruispunt over te steken.

Bij de baseline en de 10% is de avg_car_steps 935 en bij de 20% is die 620.

Het effect van het verlengen van het groene licht van het verkeerslicht op de gemiddelde aantal steps dat een automobilist voor een rood verkeerslicht staat te wachten.



Het verlengen van het groene licht geeft geen verschil in resultaten. Bij alle simulaties blijft de wachttijd voor het groene verkeerslicht hetzelfde.

Validatie van de simulatie:

Het verklaren van de lusactivaties van 7:00 tot 09:00 in de baseline simulatie is 46.96696716158677%.

KPI's

KPI's zijn de key performance indicators. Dit zijn de variabelen waarmee wij onze resultaten meten. Voor welke KPI's hebben we gekozen en waarom?

Hoeveel secondes doet een auto er gemiddeld over om door de simulatie te rijden.

Het meten hoeveel secondes het duurt voor een auto door de simulatie te rijden wordt berekend door te tellen hoeveel stappen het duurt vanaf dat de auto in de simulatie komt totdat hij zijn eindpunt heeft gehaald. Elke stap is 0.1 seconden. Dit geeft een duidelijk beeld van de doorstroom weer.

Hoeveel secondes wacht een auto gemiddeld voor een verkeerslicht.

Het meten hoeveel secondes het duurt hoelang een auto voor een verkeerslicht wacht. Dit wordt berekend door te tellen vanaf dat een auto stilstaat voor een rood verkeerslicht totdat het verkeerslicht groen is. Dit geeft een duidelijk beeld hoelang een automobilist voor een rood verkeerslicht moet wachten en zicht iets over de gebruiksvriendelijkheid.

Hoeveel procent lus activatie kan er verklaard worden.

Het kunnen verklaren van de lusactivatie wordt gedaan door elke keer te kijken wanneer een lus geactiveerd wordt of er wel een auto op staat. Door dit te delen door de totale hoeveelheid lusactivatie krijg je het percentage verklaarbare activatie. Dit zal geen 100% zijn omdat in de simulatie een auto wordt gegenereerd wanneer de eerste lus geactiveerd wordt, omdat wij niet precies weten hoeveel auto's er dan op die lus zitten is de activatie van de lus niet altijd te verklaren. Het percentage verklaarbare lusactivatie geeft een indicatie hoe dicht onze simulatie op de werkelijkheid zit.

Resultaten KPI's voor klantprobleem

Hoeveel secondes wacht een auto gemiddeld voor een verkeerslicht

Dit is gemiddeld 3.8 seconden. Hier zit geen significant verschil bij het verlengen van het groene licht van het verkeerslicht.

De reden dat dit verschil niet groot is, heeft te maken met de rest van het kruispunt. Het idee achter het verlengen van het groene licht voor het drukste verkeerslicht was dat de kans groot was dat, omdat het zo druk was, er een auto nog langs kon die nog niet was gezien door een sensor.

Ook als dat niet het geval is, blijft het verkeerslicht langer op groen en zullen de andere verkeerslichten moeten wachten. Als wij echt een significant verschil willen zien moet de toestroom van auto's voor het gekozen verkeerslicht zo hoog zijn dat de gemiddelde wachttijd lager wordt, ondanks dat de rest moet wachten. De kans dat dit ooit zal gebeuren is echter heel klein, dit ook omdat de verkeerslichtlogica al heel erg goed in elkaar steekt.

Hoeveel secondes doet een auto er gemiddeld over om door de simulatie te rijden.

Bij de baseline van onze simulatie duurt het gemiddeld 94 seconden om het kruispunt over te rijden. Bij het verlengen van het groene licht met 10% is dit ook gemiddeld 94 seconden.

Maar bij het verlengen van het groene licht met 20% is het gemiddeld 62 seconden. Dit geldt dan wel alleen voor 8 januari 2021 met de gebruikte tijden. Maar dit betekent dat het verlengen van het groene licht bij het drukste verkeerslicht met 20% een verbetering in de doorstroom geeft van 34%.

De verbetering van 34% is te verklaren omdat de simulatie maar 47% overeenkomt met de werkelijkheid en omdat deze simulatie maar is gedraaid op data van één specifieke dag. Deze dag viel ook nog eens binnen de corona lockdown. Het is dus vrij lastig om te zeggen wat deze 34% betekent.

Validatie

Om onze simulatie te valideren met de werkelijkheid hebben wij de volgende KPI gebruikt: *“Hoeveel procent lusactivatie kan er verklaard worden?”*. Wij kijken hier alleen naar de baseline/nulmeting, (de simulatie waar geen aanpassingen worden gedaan) omdat na de aanpassingen in de simulatie de lusactivatie totaal niet meer klopt en ook niet hoort te kloppen. Doordat wij de aanpassingen doen op de baseline simulatie, zal dit ook wat zeggen over alle andere simulaties.

Onze baseline heeft een verklaarbare lusactiviteit van 47%

Dit komt omdat er nogal veel factoren in de simulatie komen kijken die heel moeilijk te simuleren zijn. Elke auto trekt weer sneller op dan de andere, de ene auto is langer dan de ander en zo zijn er nog meer factoren die het moeilijk maken. Ook is de lusactivatie heel precies. De auto moet in de simulatie echt precies op de lus staan wil die verklaarbaar zijn. In de realiteit kan een lus ook al geactiveerd worden als een auto er iets voor staat.

Beantwoording van de vraagstelling

Hoofdvraag

De hoofdvraag is: *Is het mogelijk om de doorstroom op de gegeven twee kruispunten te verbeteren, door middel van het aanpassen van de logica van het verkeerslicht?*

Uit de resultaten blijkt dat het verlengen van de tijd van het groene licht bij het drukste verkeerslicht een verbetering in de doorstroom geeft van 34% op het kruispunt BOS210. Dit is wel een specifiek geval op 8 januari 2021 tijdens een corona lockdown. Dus wat het effect is over een langere periode en een periode waar geen spraken is van een lockdown kunnen wij niet zeggen.

Kortom wij kunnen dus niet vaststellen of het mogelijk is om de doorstroom te verbeteren.

Deelvraag I

De eerste deelvraag is: *Kunnen wij de verkregen data van Vialis plotten (XML hoe het kruispunt er uit ziet) en de data van de lussen/verkeerslichten visualiseren wanneer ze aan en uit gaan (CSV wanneer welke lus/verkeerslicht geactiveerd wordt)?*

Dit is ons zeker gelukt. Wij hebben een schematische visualisatie van beide kruispunten gemaakt. De visualisatie is heel flexibel opgezet en zou zonder al te veel moeite een ander kruispunt kunnen weer geven.

In de huidige visualisatie zijn verschillende routes die wij zelf hebben gereden. Deze data kunnen goed worden gebruikt als verificatie voor de sensoren. Deze verificatie kan dan ook worden gebruikt om bijvoorbeeld te kijken of een sensor kapot is of verkeerd is afgesteld.

Deelvraag II

De tweede deelvraag is: *Kunnen wij een simulatie maken waarin een gegeven kruispunt wordt gesimuleerd, met de verkeerslichten en de sensoren, waar willekeurige auto's kunnen rijden?*

Ook dit is ons gelukt. Wij hebben een mooie simulatie gemaakt, door de data van de sensoren en lichten te combineren met willekeurig verschijnende auto's hebben wij een werkend product gemaakt. Alle auto's volgen een willekeurig pad en zo is een verkeerssituaties in vogelvlucht te zien. Daarnaast kan uit de data van deze simulatie nieuwe informatie gehaald worden, zoals de benodigde tijd voor groen licht en dat het rode licht te lang aan staat.

Advies

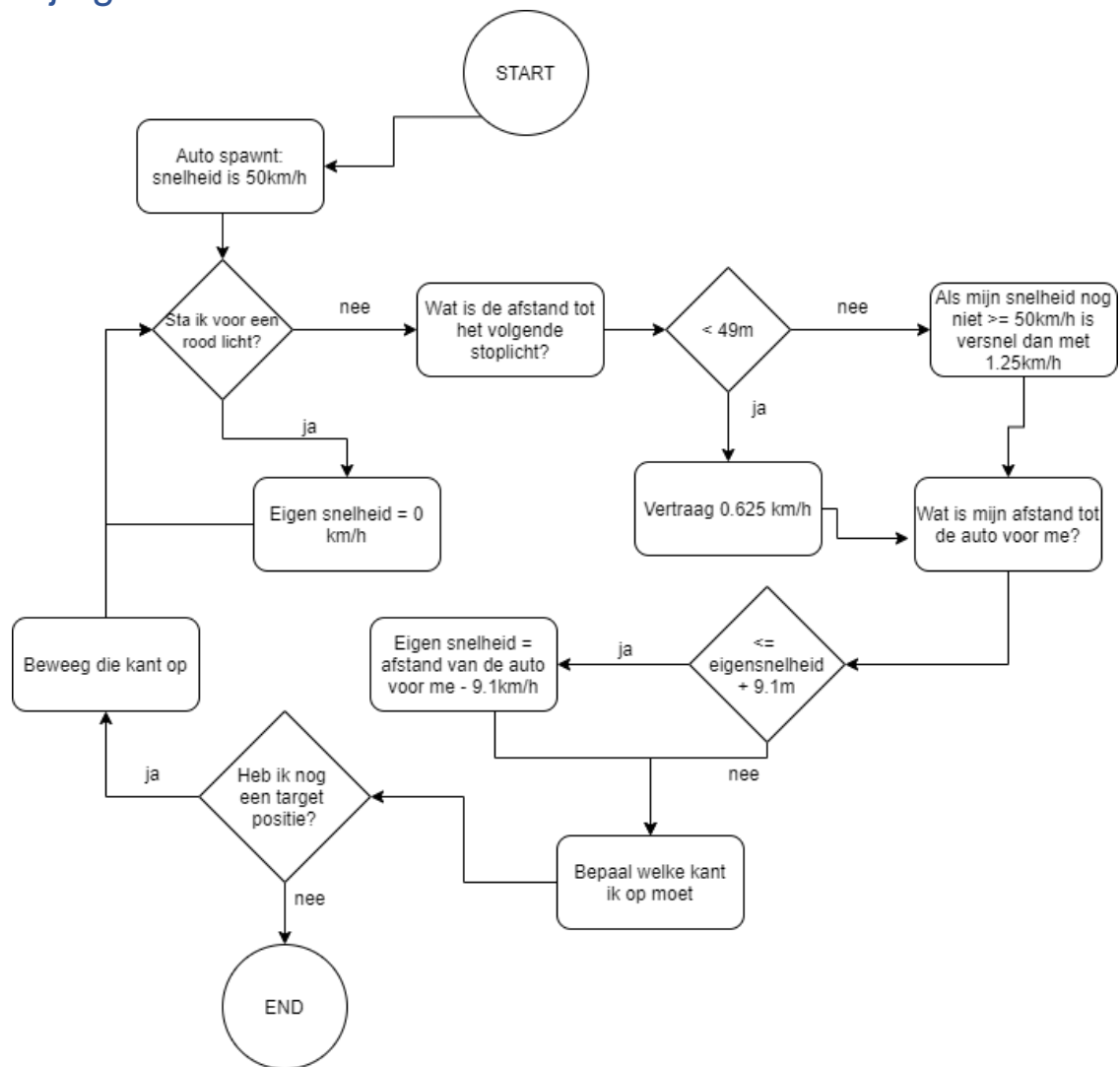
Na drie weken aan dit project te hebben gewerkt hebben we een redelijke basis neergezet voor het simuleren en visualiseren van een kruispunt. We kunnen kruispunt data uitlezen en auto's simuleren die over het kruispunt rijden. We hebben voor dit onderzoek drie verschillende simulatie instellingen gebruikt en deze onderzocht. Uit dit onderzoek bleek dat in sommige situaties we wel een betere doorstroom leken te hebben, maar is dit resultaat nuttig?

Of het onderzoek in de huidige vorm veel waarde heeft is de vraag. Er staat zeker een prima basis met behulp van extra data zou dit onderzoek misschien wel uitkomen op een significant verschil. Als we bijvoorbeeld naar elke vrijdag van de afgelopen vijf jaar zouden kijken.

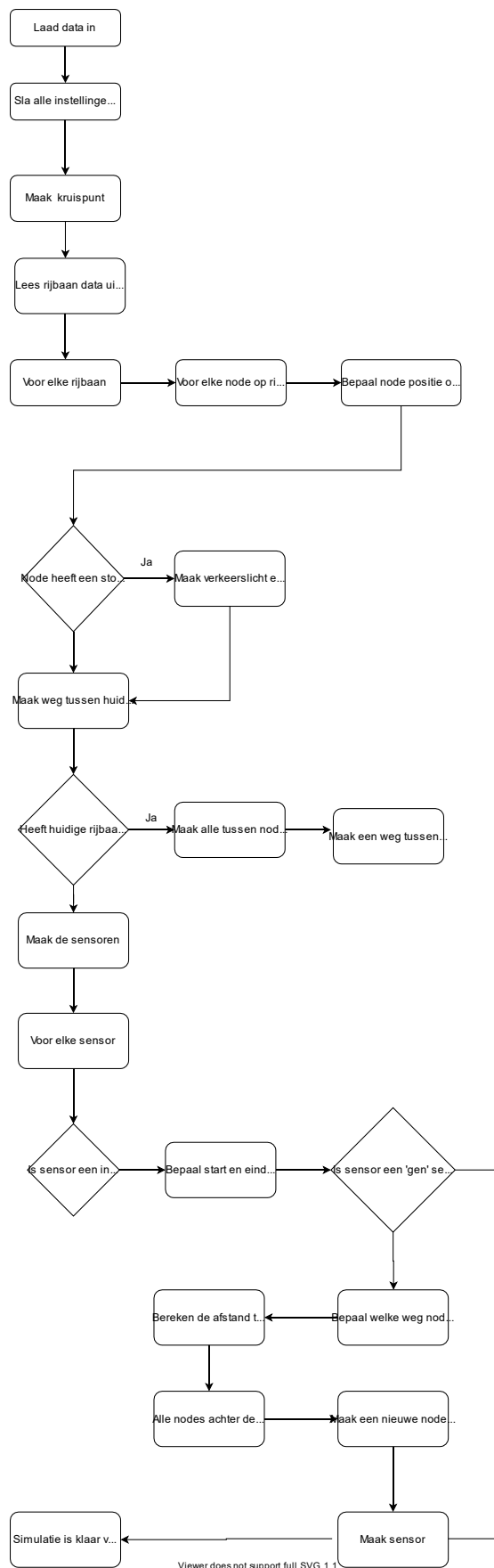
Ondanks dat onze simulatie geen slechte basis is voor een vervolgonderzoek zijn er dingen die wij zelf graag hadden toegevoegd als we meer tijd hadden gehad. Een paar van deze dingen zijn: zelf gemaakt verkeerslicht algoritmes, meerdere en geavanceerdere modellen voor de auto's en een iets duidelijkere visualisatie. Zelf hebben we wel wat tijd gestopt in het ontwikkelen van een eigen verkeerslichtalgoritme, dit is bleek toch lastiger dan gedacht en hebben we dan ook maar niet geïmplementeerd. De flowchart voor dit algoritme is te vinden in bijlage (zie bijlage C).

Kortom is ons advies dat de simulatie nog wat realistischer kan, maar al zeker een goede basis is voor een soort gelijk vervolgonderzoek, vooral omdat de simulatie zo dynamisch is opgebouwd. Dit zorgt voor snel onderzoek en biedt de mogelijkheid meerdere kruispunten tegelijk te onderzoeken.

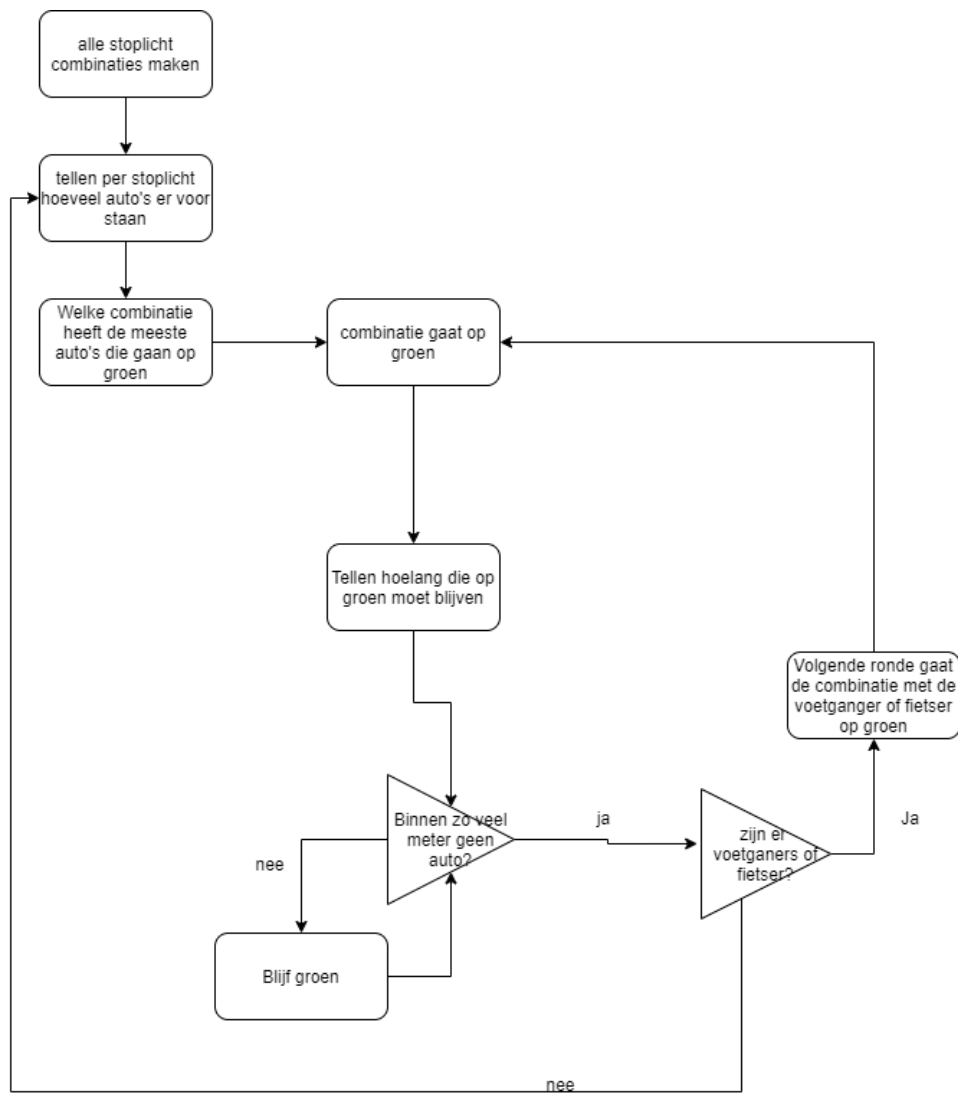
Bijlage



Bijlage A:
Auto agent flow-chart



Bijlage B:
Flow-chart van de simulatie



Bijlage C:

Concept zelf bedacht verkeerslichte logica (niet geïmplementeerd)

Bronnen

1. Luna-Benoso, B., Mart'inez-Perales, J. C., & Flores-Carapia, R. (2013b). Modeling Traffic Flow for Two and Three Lanes through Cellular Automata. International Mathematical Forum, 8(22), 2.
<https://pdfs.semanticscholar.org/2d1a/7ee8e0304bae4ff39e58d958150852974621.pdf>