

# Groeps opdracht verkeerssimulatie:

## Verkeerssimulatiecasus

2020, Corona tijd, december



Door Jelle, Kevin en Mick

# Inhoudsopgave

<b>Introductie</b>	<b>2</b>
<b>Onderzoeksvraag</b>	<b>2</b>
<b>Vooronderzoek</b>	<b>3</b>
Opdrachtbeschrijving	3
<b>Modules</b>	<b>4</b>
<b>Toolkeuze:</b>	<b>5</b>
Conclusie toolkeuze	6
<b>Aanpak</b>	<b>7</b>
<b>Experiment uitleg</b>	<b>7</b>
<b>Resultaten</b>	<b>8</b>
<b>Conclusie</b>	<b>8</b>
<b>Discussie</b>	<b>11</b>
<b>Bronnen:</b>	<b>12</b>

## Abstract

In deze paper wordt ons proces beschreven over hoe we tot een oplossing van onze casus zijn gekomen. Welke beslissingen we hebben genomen en waarom en welke keuzes bewust waren. We hebben eerst een plan van aanpak gemaakt, en daarin beschreven hoe we dit project hebben aangepakt. Vervolgens hebben we onze applicatie gemaakt en daar data uit kunnen krijgen. Ten slotte zijn we die data gaan analyseren en tot een conclusie en antwoord gekomen op onze vraag.

# Introductie

In Maart 2020 is de snelheidslimiet op snelwegen verlaagd van 130 naar 100 km/u. Zo hoopte de Tweede Kamer om het stikstofprobleem te verminderen. Dit vonden veel automobilisten minder fijn, maar misschien is dit wel ten goede gegaan van de doorstroom op de snelweg. Dit leidde tot de vraag wat de optimale snelheidslimiet is op de snelweg? In ons project gaan we proberen om een antwoord op die vraag te krijgen door middel van een agent-based simulation die we zelf maken. Wij hebben één rijstrook en een invoegstrook in onze simulatie waar onze agents op kunnen rijden. We kunnen in onze simulatie de maximumsnelheid en het aantal auto's op de weg aanpassen naar wat we willen. Qua modules zijn deze hetzelfde gebleven als in opdracht 4, behalve dat we nu een invoegstrook hebben in plaats van een extra rijbaan.

## Onderzoeksvraag

### **Vorige week hadden we deze casus als hoofdvraag:**

De maximumsnelheid overdag is verlaagd van 130 naar 100. Er wordt veel geklaagd over deze nieuwe maximumsnelheid, maar misschien komt het de doorstroming wel ten goede. We gaan door middel van het maken van een `agent based simulation` kijken wat het effect is van de maximumsnelheid van een weg op de doorstroming van die weg.

### **Maar nu gaan we deze simulatie uitbreiden:**

Wij willen hier graag een invoegstrook aan toevoegen om te kijken of dit voor veel extra files zou zorgen, en dat dit voor minder files zou zorgen als de maximum snelheid lager is. We verwachten dat dit een nog realistischere simulatie zou geven van de werkelijkheid, en verwachten dat hierdoor met een lagere snelheid op de snelweg er minder files ontstaan dan met een snellere snelheid.

### **Bij deze onze onderzoeksvraag:**

**Is een lagere snelheidslimiet beter voor de doorstroming op snelwegen? En wat is de invloed van een invoegstrook hierop?**

## Vooronderzoek

In een artikel van nu.nl (Gurp, 2020) legt Dick de Waard het volgende uit: "De ideale snelheid is 95 kilometer per uur. Dan is de weg benutting optimaal. Als je langzamer gaat rijden, gebruik je de wegcapaciteit niet optimaal. Als je sneller rijdt, zorgt het voor meer opstoppen, omdat je automatisch meer afstand neemt als je harder gaat en er dus minder auto's op de weg passen."

Ook zegt Dick de Waard in dit artikel: "Auto's die moeten inhalen, uitvoegen of invoegen, verhogen of verlagen de snelheid en wisselen van rijstrook. Dat zorgt voor extra files. Als je de snelheid verlaagt, is het verschil tussen automobilisten onderling een stuk kleiner"

Deze uitspraken zouden wij graag willen toetsen in onze casus. Niet specifiek met kilometer per uur, want dit is vrij lastig in netlogo. Maar wel met een snelle, middel en langzame test.

## Opdrachtbeschrijving

De maximumsnelheid overdag is verlaagd van 130 naar 100. Er wordt veel geklaagd over deze nieuwe maximumsnelheid, maar misschien komt het de doorstroming wel ten goede. We gaan door middel van het maken van een `agent based simulation` kijken wat het effect is van de maximumsnelheid van een weg op de doorstroming van die weg.

Maar aangezien dit de basis case is die van ons verwacht wordt willen wij hier graag een extra rijstrook aan toevoegen om te kijken of dit veel zou veranderen in de simulatie, want als je iemand ziet remmen zou je er mogelijkserwijs ook gewoon omheen kunnen zodat er geen files ontstaan.

## Modules

**Herstartbaar:** Het zou het maken van een simulatie een stuk makkelijker maken voor ons als deze ook gemakkelijk te herstarten is. Ook is het fijn dat we zonder veel gedoe instellingen kunnen veranderen (bijvoorbeeld de hoeveelheid auto's in de simulatie).

**Tijd:** We willen een simulatie die werkt per tick. Elke tick wordt de positie van elke auto geupdate en kan er worden berekend hoe de doorstroming ervoor staat. Het zou fijn zijn om het aantal ticks per seconde/minuut aan te kunnen passen zodat we de simulatie op een door ons gekozen tempo kunnen draaien.

**Staat:** Het maakt voor onze simulatie niet uit of we de individuele staat van een agent kunnen zien. Wel zou het mooi zijn om 1 auto te volgen en te kijken wat de snelheid van deze auto is en wat dit zegt over de doorstroming. Wel is het vereist dat de staat van bepaalde groepen of alle agents te meten is. Hieruit kunnen we data verzamelen over de doorstroming van de weg in verhouding tot de max snelheid.

### Gui:

We willen invoer en uitvoer laten zien in een GUI met invoer-boxes. Ook gaan we resultaten live laten zien met grafieken en displays waar specifieke waarden van agents in staan zoals bijvoorbeeld de (gemiddelde) snelheid, of the snelheid in loop van de tijd.

### Resultaten:

Om de data uiteindelijk te analyseren, moeten we de data opslaan. Bij voorkeur in een .csv bestand waarmee we later in bijvoorbeeld Python mee kunnen werken.

### Agents:

Onze agents representeren auto's die op een snelweg rijden. Ze spawnen in, maar later willen we een invoegstrook toevoegen waar auto's op kunnen spawnen en 'normaal' in kunnen voegen. Het is fijn als we in de tool makkelijk nieuwe agents kunnen toevoegen en aansturen.

# Toolkeuze:

## Tool keuze model

SF(A)-Model	Netlogo	Unity	Mesa
<b>Herstartbaar</b>	<b>Score: 10</b> <hr/> Geschiktheid: 5 Haalbaarheid: 5 <hr/> In Netlogo is een simulatie herstarten heel makkelijk, 1 druk op een knop die je helemaal zelf kan instellen. Ook kan je zelf sliders en buttons toevoegen waardoor je niet in de editor moet gaan zitten klooiën.	<b>Score: 7</b> <hr/> Geschiktheid: 4 Haalbaarheid: 3 <hr/> In Unity is de simulatie makkelijk te herstarten met verschillende instellingen, dit kost soms wel wat tijd en dit moet aangepast worden in de editor. Met weinig moeite kan dit ook tijdens de simulatie worden gedaan.	<b>Score: 9</b> <hr/> Geschiktheid: 4 Haalbaarheid: 5 <hr/> In Mesa kan je makkelijk de simulatie herstarten, hier kan je sliders / textboxes toevoegen en aanpassen met andere argumenten. Daarna zul je de webpagina wel moeten herladen (F5).
<b>Tijd</b>	<b>Score: 10</b> <hr/> Geschiktheid: 5 Haalbaarheid: 5 <hr/> Je kan de simulatie per tick laten lopen of gewoon zo snel als je computer kan. Je kan zelf kiezen hoeveel ticks een bepaalde functie in neemt. Dit is allemaal ingebouwd en daarom heel simpel om te veranderen.	<b>Score: 8</b> <hr/> Geschiktheid: 4 Haalbaarheid: 3 <hr/> Het makkelijkste is om met ticks te werken, andere opties kosten veel tijd. Dit hoeft niet per se nadelig te zijn voor onze simulatie omdat elke tick gekeken moet worden naar de verkeerssituatie.	<b>Score: 8</b> <hr/> Geschiktheid: 3 Haalbaarheid: 5 <hr/> In Mesa heb je standaard een slider voor het aantal ticks per seconde (tussen de 1 en 20) dit is prima maar het is toch fijner om zelf wat meer vrijheid te hebben hierin.
<b>Staat</b>	<b>Score: 9</b> <hr/> Geschiktheid: 5 Haalbaarheid: 4 <hr/> De staat van een individuele agent bekijken is ook weer erg makkelijk in netlogo. Je kan rechtermuisknop op een agent klikken om zijn staat te bekijken, maar je kan ook live grafieken baseren op de staat van alle, een specifiek groepje of een individuele agent. Wel is hier wat diepere kennis van NetLogo voor nodig (over Reporters ed).	<b>Score: 8</b> <hr/> Geschiktheid: 3 Haalbaarheid: 5 <hr/> Binnen de editor kan je een preview van de simulatie bekijken, hierin kan je alle agents aanklikken en hun staat zien. Handig voor bugsolving maar misschien net iets minder geschikt voor het laten zien van live data.	<b>Score: 4</b> <hr/> Geschiktheid: 2 Haalbaarheid: 5 <hr/> De staat van een individuele agent is moeilijk te krijgen, hierover is ook niet veel informatie over te vinden. Wel kan je een live grafiek laten zien op de webpagina maar hier is wel kennis van JavaScript voor nodig. Voor onze casus is dit niet goed geschikt.



<b>Gui</b>	<b>Score: 9</b> <hr/> Geschiktheid: 4 Haalbaarheid: 5 <hr/> <p>In een NetLogo simulatie kunnen we alles laten zien wat we nodig hebben + de nodige slider en buttons. Helaas zijn de looks niet super en blijft alles er soms iets te simpel uit zien.</p>	<b>Score: 7</b> <hr/> Geschiktheid: 5 Haalbaarheid: 2 <hr/> <p>In unity zou je een hele mooie simulatie kunnen maken, dit kost helaas heel veel moeite in vergelijking met de andere tools. De geschiktheid is dus hoog maar de haalbaarheid niet. Als we unity voor dit project zouden gebruiken zouden we de editor gebruiken als GUI.</p>	<b>Score: 10</b> <hr/> Geschiktheid: 5 Haalbaarheid: 5 <hr/> <p>Mesa geeft de mogelijkheid om de simulatie in een webpagina te visualiseren. In deze webpagina is het mogelijk om sliders, inputboxes, boolean-sliders, etc. toe te voegen om argumenten mee te geven en te veranderen. Dit zou een mooie en simpele oplossing zijn voor onze casus.</p>
<b>Resultaten</b>	<b>Score: 7</b> <hr/> Geschiktheid: 3 Haalbaarheid: 4 <hr/> <p>In NetLogo kan je experimenten aanmaken en de uitkomsten daarvan opslaan in een CSV, Dit is niet heel duidelijk en het CSV bestand is ook niet mega duidelijk. Wel is het gewoon bruikbaar om te analyseren.</p>	<b>Score: 4</b> <hr/> Geschiktheid: 2 Haalbaarheid: 2 <hr/> <p>Het is best lastig om de resultaten van een simulatie op te slaan, unity is in de basis natuurlijk een game-maker. Simulaties draaien kan heel mooi worden gedaan maar de uitkomsten analyseren dus een stuk minder.</p>	<b>Score: 9</b> <hr/> Geschiktheid: 5 Haalbaarheid: 4 <hr/> <p>In Mesa heb je een klasse DataCollector. Deze verzamelt data uit je simulatie met behulp van pandas. Goed op te slaan en daarna analyseren (eventueel meteen in python)</p>
<b>Totaal</b>	<b>45</b>	<b>34</b>	<b>40</b>

## Conclusie toolkeuze

Zoals de scoren doen blijken hebben we gekozen voor **NetLogo**. Dit is voor het hele groepje makkelijk te leren en heeft precies alle mogelijkheden die we nodig hebben. Vooral de flexibiliteit, Gui & het delen en resultaten zijn belangrijk, hierin is NetLogo allemaal sterk. In Unity zouden we ook een mooie verkeerssimulatie kunnen maken maar dit zou al snel te moeilijk worden voor de tijd die we hebben en het zou het analyseren van de resultaten een stuk minder makkelijk maken.

# Aanpak

We begonnen dit project met een PVA (Plan van aanpak). Terwijl we dit maakten zijn we al uitgebreid aan het nadenken geweest over hoe we onze simulatie willen maken en kunnen uitbreiden. In deze eerste week hebben we ook al een basis opzet gemaakt van de code.

In de volgende weken zijn we aan de gang gegaan met het uitbreiden van de simulatie om zo de onderzoeksvraag nog beter te kunnen beantwoorden. We hebben de resultaten van de simulatie verzameld zodat we deze kunnen analyseren.

Als uitbreiding hadden we een extra (invoegstrook bedacht. We kunnen daarmee kijken hoe deze extra verkeersstroom het overige verkeer zal beïnvloeden. Mensen die op de invoegstrook rijden zijn namelijk vaak langzamer dan mensen op de snelweg, dit kan verkeer of tegenwerken, of misschien juist helpen.

## Experiment uitleg

Wij gaan een snelweg met een invoegstrook simuleren in Netlogo. Tijdens de simulatie slaan we de data over de auto's en en rijstroken op. Deze data bevat de gemiddelde snelheid van alle auto's, en de losse gemiddelde snelheden van de auto's op de hoofdrijbaan en de invoegstrook

Elke auto (agent) kijkt elke tick of er een andere auto voor hem rijdt, wanneer deze te dichtbij komt zal de auto afremmen of zelfs tot stilstand komen. Als de voorligger weer gaat rijden zal de auto zelf ook weer versnellen. Ditzelfde principe gebeurt ook met de auto's op de invoegstrook. Wanneer een auto aan het einde van de strook komt kijkt hij of er een ander auto in de buurt komt op de hoofdrijbaan, als dit niet zo is kan de auto doorrijden.

Binnen onze simulatie kan de gebruiker het aantal auto's en de speed limit aanpassen. Ook kan de gebruiker beslissen om de invoegstrook uit te schakelen en kan het percentage auto's dat op de invoegstrook spawnen aanpassen.

### Design

We willen weten wat de invloed is van de maximumsnelheid op de doorstroom van de weg. Ook hebben we als uitbreiding een invoegstrook toegevoegd. Hiervoor hebben we bepaalde instellingen nodig die we aan kunnen passen. Dit proberen we zo simpel mogelijk te houden.

We hebben gekozen voor een 1 baans 'snelweg' met een invoegstrook. 2 rijbanen was erg ingewikkeld omdat auto's dan moeten wisselen van rijbaan, nu rijden ze allemaal 1 kant op en kunnen ze alleen maar van de invoegstrook naar de rijbaan. Zo krijgen we een simpele simulatie van een snelweg.

De gebruiker heeft de optie om deze invoegstrook aan en uit te zetten, zo kan je bijvoorbeeld alleen maar testen wat de maximale snelheid voor invloed heeft op eventuele filevorming op de snelweg. Ook kan de gebruiker het percentage auto's dat op de invoegstrook spawnen aanpassen.

Verder kan de gebruiker het aantal auto's en de maximale snelheid instellen. Dit wordt automatisch gedaan wanneer we de simulatie runnen om er resultaten uit te krijgen. Dit is ingebouwd in Netlogo. Deze tool runt de simulatie een aantal keer (160 x bij ons) met veel verschillende combinaties aan instellingen. Dit wordt vervolgens omgevormd naar een csv dat we kunnen analyseren in python

# Resultaten

Het is voor ons niet mogelijk om bijvoorbeeld standaardafwijkingen te berekenen, omdat onze dataset al bestaat uit gemiddelden. We hebben in totaal 160 verschillende combinaties aan parameters getest in onze simulatie, elk van deze combinaties hebben wij één keer uitgevoerd. Dit kan beter qua betrouwbaarheid, maar dit is niet moeilijk uit te breiden. Om dit te realiseren hoeven we alleen maar nog een keer het gemiddelde te nemen.

We hebben veel verschillende en bruikbare grafieken en DataFrames kunnen maken waar we heel veel informatie uit hebben kunnen halen. Zoals een correlatiematrix die ons aangaf dat bij een hogere maximumsnelheid een hogere potentiële gemiddelde snelheid zat, maar dat na een bepaalde snelheidslimiet de laagste gemiddelde snelheid niet meer omhoog ging.



# Conclusie

De gemiddelde snelheid op snelwegen **met** invoegstrook, lag hoger dan de gemiddelde snelheid op snelwegen **zonder** invoegstrook. Dit is te blijken uit onze data die uit onze simulatie hebben gekregen.

## 5 beste gemiddelde snelheden **met** invoegstrook

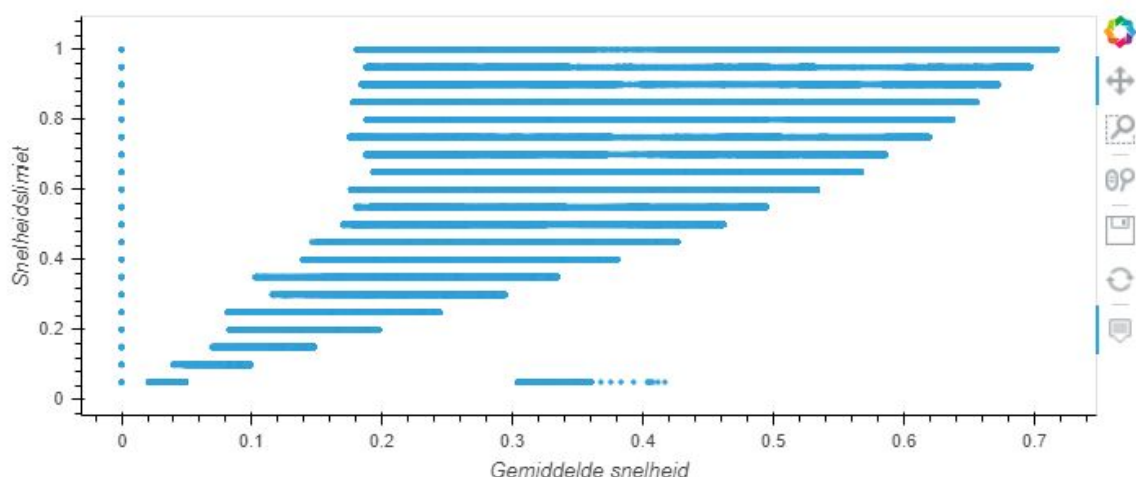
	[run number]	speed_limit	aantal_autos	invoeg	[step]	avr_speed	avr_speed_1	avr_speed_2
246874	121.0	0.80	15.0	True	2000.0	0.635558	0.658420	0.375012
265139	129.0	0.85	15.0	True	2000.0	0.636150	0.669419	0.321276
278411	137.0	0.90	15.0	True	2000.0	0.666697	0.714233	0.326027
297581	145.0	0.95	15.0	True	2000.0	0.696924	0.741156	0.347436
315863	153.0	1.00	15.0	True	2000.0	0.711465	0.758053	0.353197

## 5 best gemiddelde snelheden **zonder** invoegstrook

	[run number]	speed_limit	aantal_autos	invoeg	[step]	avr_speed	avr_speed_1	avr_speed_2
266926	130.0	0.85	15.0	False	2000.0	0.623779	0.623779	0.0
261735	122.0	0.80	15.0	False	2000.0	0.630919	0.630919	0.0
288651	138.0	0.90	15.0	False	2000.0	0.655707	0.655707	0.0
306472	146.0	0.95	15.0	False	2000.0	0.685602	0.685602	0.0
311911	154.0	1.00	15.0	False	2000.0	0.696059	0.696059	0.0

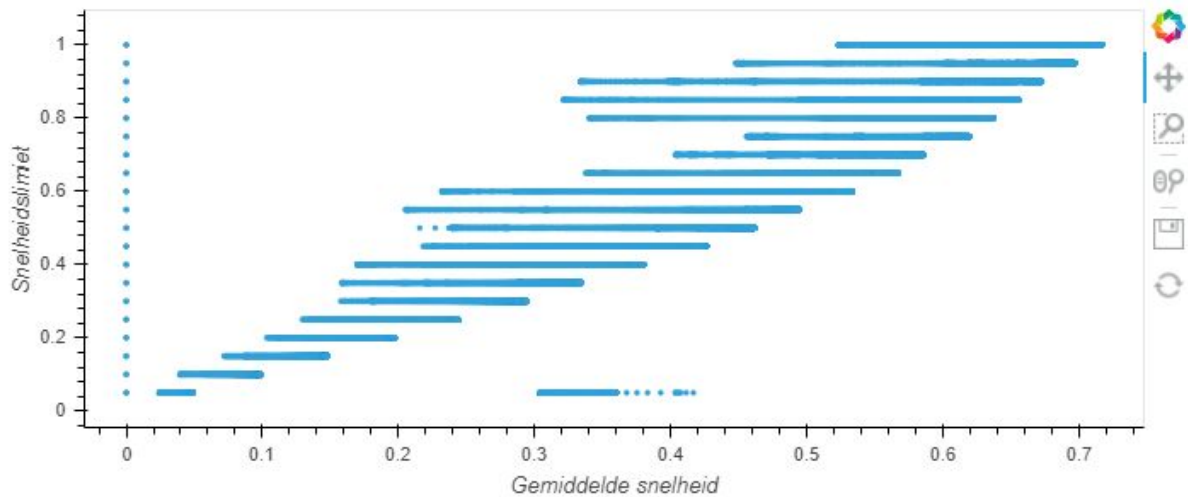
Zoals je kunt zien, zijn alle waarden in kolom 'avr\_speed\_1' in de bovenste tabel hoger dan in de tabel eronder. Wij denken dat dit komt, doordat er auto's op de invoegstrook rijden, dat het aantal auto's op de snelweg afneemt, en dat ze dus in de meeste gevallen sneller kunnen rijden.

Tijdens deze analyse zijn we er ook achter gekomen dat op een normale of drukke snelweg is een iets lagere maximumsnelheid iets beter voor de doorstroming. Tijdens een file zou een lagere maximumsnelheid ook niet veel uitmaken, maar op rustige snelwegen hindert een lagere maximumsnelheid de doorstroming. Er moet dus een balans zijn of er moet per snelweg gekeken worden hoe druk deze is, en aan de hand daarvan een keuze worden gemaakt wat de maximumsnelheid wordt op die weg.

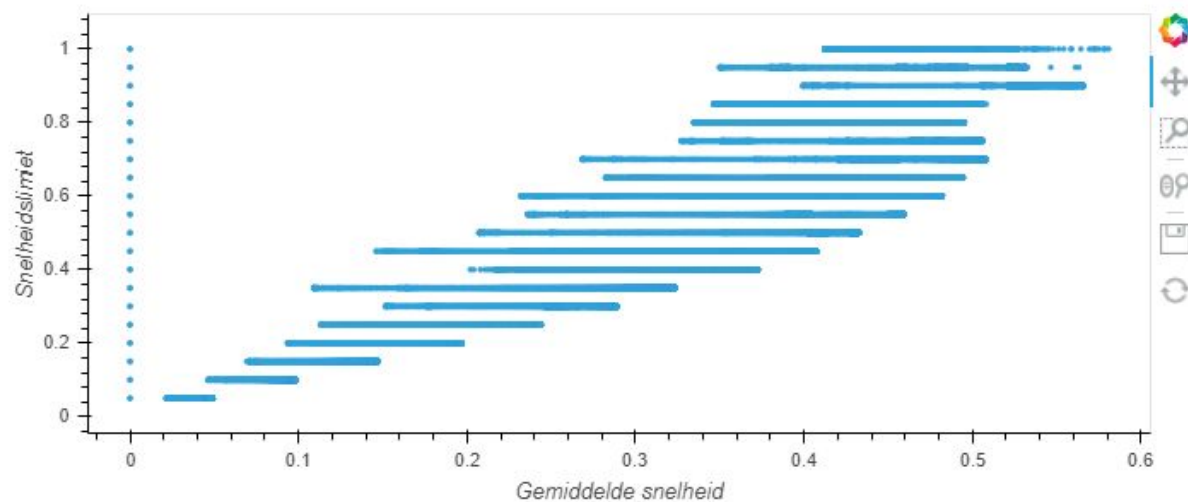


Wij hebben een correlatiematrix gemaakt, en aan de hand daarvan een scatterplot gemaakt waarin te zien is dat de potentiële gemiddelde snelheid hoger ligt, naarmate de snelheidslimiet toeneemt. Maar ook zien we dat de laagste gemiddelde snelheid niet meer omhoog gaat bij een snelheidslimiet hoger dan 0.575.

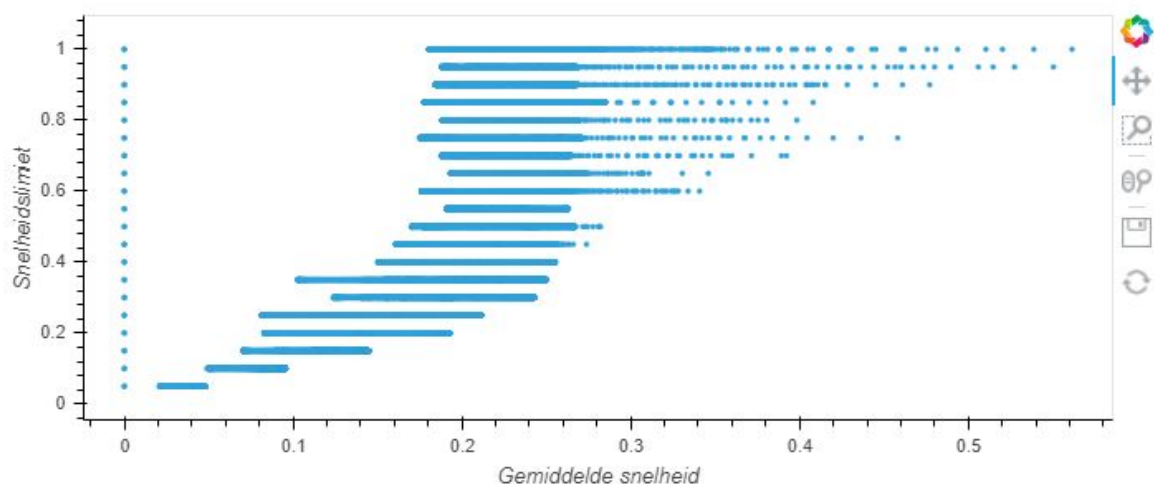
*Op een rustige snelweg is een optimale snelheid 1.00. ( zo hard rijden als het kan. )*



*Op een normale/drukke snelweg is een optimale snelheid 0.90 - 0.95*



*Op snelwegen met regelmatige files kan de snelheid tussen 0.65 - 1.00 liggen zonder dat dit veel effect heeft.*



# Discussie

## **Algemeen:**

Meerdere dingen in deze paper zijn erg versimpeld om de simulatie vorm te kunnen geven binnen een vrij korte periode in dit onderzoek. Hierom willen wij graag aangeven dat er nog meer onderzoek nodig is om deze simulatie nog beter vorm te kunnen geven, net zoals deze in de echte wereld ook zou gebeuren.

## **Basis:**

Deze simulatie is natuurlijk een stuk versimpeld in opzichte van de echte wereld, alle acties die een agent doet is gebaseerd op voorgeprogrammeerde mogelijkheden en ze denken niet zelf. Op een echte weg gebeuren veel onverwachte dingen, en daar houden we nu geen rekening mee. We denken ook niet dat Netlogo daar het geschikte programma voor zou zijn, maar voor onze simpele casus zeker zeer nuttig.

## **Invoegstrook:**

Verder kunnen we ook aangeven dat onze aanpak voor een invoegstrook een vrij goede poging was, maar deze in de werkelijkheid natuurlijk anders zou kunnen functioneren. Dit zorgt er voor dat de data uit de simulatie niet één op één hetzelfde zou zijn zoals in de werkelijkheid verwacht zou worden. Een voorbeeld hiervan is dat in de simulatie een variable hoeveelheid autos of terugkomen op de invoegstrook, of op de normale weg. Alleen zorgt dit er voor dat de auto's meer "weg"/"ruimte" hebben dan in de situatie waar de invoegstrook niet bestaat. Hierdoor zou dit wellicht een vertekend beeld kunnen ontstaan, en is de mogelijk de simulatie niet honderd procent betrouwbaar.

## **Snelheden & Acceleratie:**

Omdat we in Netlogo niet echt met Kilometers/uur kunnen rekenen hebben we de snelheid op een schaal van 0 tot 1 gehouden. Dit lijkt erg op 0 tot 100 km (dus 0.65 is dan 65 km/uur). Zo komen we uiteindelijk uit op de beste snelheid van 0.95, dit zou in verhouding staan tot de 95 km/uur waarvan gezegd werd dat het goed zou werken. Maar omdat we dit dus uiteindelijk niet wiskundig hebben kunnen bewijzen is dit een "magic-number".

We hebben een poging gedaan door een aanname te doen met de gemiddelde lengte van een auto en hiermee te berekenen hoe snel deze rijden. Maar dit is natuurlijk ook niet accuraat en hebben we daarom weggelaten.

De lengte van een tick ten opzichte van echte tijd (seconde) is ook niet echt te meten, dit ligt natuurlijk aan de framerate. We meten dus eigenlijk in snelheid/tick.

Hierbij komt ook nog dat we de acceleratie en deceleratie zelf hebben beschreven. Hierbij komt steeds een vaste snelheid waarmee de auto afremt of optrekt en daarbij een random snelheid (voor elke auto apart berekend) omdat in het echt ook iedereen anders rijdt. Ook dit is niet gebaseerd op echte meters of reactiesnelheid, daarvoor is Netlogo net te simpel.

Een extra onderzoek naar de daadwerkelijke afstand/snelheid/tijd verhouding in netlogo kan helpen deze simulatie realistischer te maken. Ook zou je kunnen onderzoeken wat de gemiddelde remweg en reactietijd bij die snelheid is. Helaas had dit voor ons teveel tijd gekost, we zijn ons hier wel bewust van!

## Resultaten

De meegeleverde data kon accurater. Wij hebben voor elke combinatie van parameters (aantal auto's, snelheidslimiet, invoegstrook wel/niet) één simulatie gedraaid. Dit kwam uiteindelijk uit op 160 verschillende simulaties. Dit kon accurater én betrouwbaarder door elke combinatie meerdere keren te draaien, en vervolgens het gemiddelde te nemen.

De data kon ook beter meegeleverd worden met meer kolommen. Nu wordt de gemiddelde snelheid per tick opgeslagen in een variabele, en dit wordt opgeteld met alle gemeten gemiddelde waarden van vorige ticks. Vervolgens wordt deze waarde gedeelt door de huidige tick. Dit kan beter door de gemiddelde snelheid per tick op te slaan en mee te geven, in plaats van dat je het gemiddelde van het totaal opslaat. Zo kan je altijd in je analyse nog het gemiddelde van het totaal berekenen dat niet zo moeilijk is. Je kan zo ook grafieken maken met wat de huidige gemiddelde snelheid was per tick, en beter zien of er files ontstonden.

Doordat de data niet enorm accuraat is, is onze conclusie dus niet erg betrouwbaar, en kan dus beter onderbouwd worden met betrouwbaardere data.

## Bronnen:

Gurp, T. v. (2020, 12 7). *verkeerskundigen-sneller-thuis-bij-verlaging-maximumsnelheid*. verkeerskundigen-sneller-thuis-bij-verlaging-maximumsnelheid.

<https://www.nu.nl/binnenland/5998958/verkeerskundigen-sneller-thuis-bij-verlaging-maximumsnelheid.html>

### Netlogo:

While loop: <http://ccl.northwestern.edu/netlogo/docs/dict/while.html>

Let/Nobody: <http://ccl.northwestern.edu/netlogo/docs/dict/let.html>

Turtles-on/patch-ahead: <http://ccl.northwestern.edu/netlogo/docs/dict/turtles-on.html>