

Casus: *Rotmaatregel*

Lucas van Wijk, Roeland Oostdam, Freek Gerrits Jans

Abstract

De overheid heeft besloten dat het snelheidslimiet op alle snelwegen wordt vastgezet op 100 km/u. Dit betekent een verschil tot 30 km/u. Hierbij beweert de overheid dat het niet alleen beter voor het milieu is, maar dat het voor de doorstroom van het verkeer niet zoveel zou moeten uitmaken. Sterker nog, de doorstroom zou in sommige gevallen zelfs verbeteren. Wij proberen door middel van het gebruik van een simulatie te achterhalen of dit klopt. Hiermee laten wij zien wat het effect op doorstroom van het verkeer is bij verschillende snelheidslimieten i.c.m. wegbezettingen. Uit deze simulatie blijkt dat een hogere snelheid tot een bepaalde wegbezetting optimaal is, en dat bij een te drukke weg een hoger snelheidslimiet averechts werkt.

Introductie	2
Onderzoeksvraag	3
Het plan	3
Suitability:	4
Feasibility:	6
Compatibility:	6
Experiment Design	7
Resultaten	8
Conclusie	12
Discussie	13
Bronnen	15

Introductie

De theorie van het shockwave effect vertelt ons dat wanneer er een auto remt op bijvoorbeeld een drukke snelweg, er een shockwave effect ontstaat. Dat wil zeggen dat alle auto's achter de remmende auto ook moeten remmen en een opstopping ontstaat. Snelheid speelt bij een shockwave effect een rol. Door middel van een simulatie en een visualisatie wordt het shockwave effect in kaart gebracht. Met deze informatie bepalen we of er een duidelijk verschil zit in het effect van een shockwave en de snelheid. We laten zien dat bij bepaalde weg druktes het handiger is om een lager snelheidslimiet toe te passen dan 130 km/h. Hiervoor zullen wij gebruik maken van het Nagel-Schreckenberg model [\[1\]](#).

Onderzoeksvraag

Onder welke omstandigheden heeft het verlagen van de maximum snelheid een positieve invloed op de doorstroom van het verkeer.

Het plan

Het is de bedoeling om de simulatie in een GUI te maken waarin de omgevingsvariabelen makkelijk aan te passen zijn.

De omgevingsvariabelen zijn als volgt:

- Verkeersdrukke (Auto's per minuut)
- Snelheidslimiet
- Acceleratie
- Deceleratie
- Zichtbereik

Deze zullen we bij het experiment niet veranderen. Alleen het snelheidslimiet is variabel.

Een tijdstap in deze simulatie is de tijd dat een auto erover doet om 10 autolengtes te verplaatsen. (Volgens het Nagel-Schreckenberg model)

In de GUI is het mogelijk om per stap te simuleren en te pauzeren. Tussendoor wordt door middel van grafieken de any state gevisualiseerd. Ook bekijken wij tussentijds de eigenschappen van een specifieke agent voor debuggen.

Opbouw

Ons project heeft een autobaan. Op deze weg zijn een aantal auto's gemodelleerd die daar rijden. Deze auto's hebben een bepaalde vision range en een maximum snelheid. Sommige auto's zullen random afremmen. De auto's zullen proberen te voorkomen om te botsen. Bij elke stap zal de auto vier acties uitvoeren:

1. Perceive
Bij perceive kijkt de auto of er andere auto voor hem is en hoe ver.
2. Update
Hierin wordt bepaald welke van de hieronder in "verschillende state" beschreven states de auto zal krijgen.
3. Act
Hier zal de auto zijn snelheid aanpassen naar de snelheid van de huidige state.
4. Move
De auto verplaatst zich naar voren met de huidige snelheid.

Verschillende states

Er zijn drie verschillende states die een auto kan hebben. Alle states hebben invloed op de snelheid van een auto. Wanneer de state wordt uitgelezen en uitgevoerd zal er het volgende gebeuren per state.

- acceleration : Bij deze state zal de snelheid per step een vooraf gegeven eenheid in snelheid stijgen.
- braking : Bij deze state zal de snelheid per step een vooraf gegeven eenheid in snelheid dalen.
- randomBraking : Bij deze state zal een auto random per step zoveel in snelheid afremmen dat er altijd minimaal 1 eenheid in steps tussen de auto's blijft.
- cruising : Bij deze step zal de auto de snelheid aanhouden van de vorige step. anders gezegd de snelheid veranderd niet.

Suitability:

Tool supports coding the modules described in part 2. For each of your modules, rate how well your tool is suited.

GUI

- Unity: 7. Unity heeft veel variabelen al in een format staan. Denkend aan zwaartekracht, lengte van een auto of de versnelling van een auto. Deze variabelen zijn handig om de beïnvloedende factoren te beschrijven bij een dergelijk simulatie. De simulatie wordt in 3D weergegeven wat erg mooi oogt, echter hebben we genoeg aan een 2D simulatie voor deze opdracht. Om de tool te snappen hebben we erg veel tijd nodig, iets wat we niet hebben voor dit project.
- Mesa: 8. de GUI van Mesa is voldoende voor deze opdracht. De tool geeft de mogelijkheid in 2D een simulatie te bouwen. Erg handig dat Mesa met Python werkt, want iedereen uit de groep heeft ervaring met Python.
- Netlogo: 6. De GUI van deze tool is ook voldoende. Het laat een 2D beeld zien van de simulatie.

Time

- Unity: 6. Heeft verschillende manieren om objecten te laten bewegen op tijd.
- Mesa: 8. Door de module Simultaneous Activation te gebruiken kunnen alle auto's eerst te horen krijgen wat een deze agent moet doen dat tijdstip en vervolgens als alle auto's een opdracht hebben gekregen doen zij allen tegelijkertijd wat hen is opgedragen.
- Netlogo: 3. Bij Netlogo kan men ask turtles (in ons geval auto's) gebruiken. Hiermee krijgen alle auto's een commando. De auto's kunnen per auto worden aangesproken of allen tegelijkertijd met allen hetzelfde command.

Collect data

- Unity: 8. Bij Unity kan makkelijk states worden opgeslagen en uitgelezen. Met de inspector methode is het mogelijk om te kijken wat een bepaalde auto zijn internal state is.
- Mesa: 7. Doordat Mesa Python gebruikt als taal kan er in classes objecten worden opgeslagen waarin dus data makkelijk kan worden opgeslagen en uitgelezen.
- Netlogo: 7. Heeft de mogelijkheid om agents te inspecteren.

interactive GUI

- Unity: 9. Unity is een uitgebreid programma gemaakt voor visualisatie en interactie. Dit maakt interactie met de simulatie erg makkelijk. Dit is echter erg uitgebreid en zal voor ons project een overbodige luxe zijn voor datgene wat we laten zien met onze simulatie.
- Mesa: 7. Mesa kan door middel van knoppen en een slider makkelijk variabelen veranderen. Dit is precies goed voor onze simulatie.
- Netlogo: 7. Netlogo kan ook makkelijk variabelen aanpassen door middel van knoppen en sliders. Ook deze mate van interactie is goed voor onze simulatie.

Agent

- Unity: 8. Bij de module agents van Unity kun je erg veel details geven aan een specifieke agent. De basis bevat al erg veel details ten opzichte van wat nodig is voor deze opdracht.
- Mesa: 9. De module geeft ons een mogelijkheid om object oriented te werk te gaan. We kunnen zo per agent erg fijn aangeven wat een state kan zijn van elke specifieke agent.
- Netlogo: 3. Kan een specifieke agent of groep agents een commando geven. Tot nu toe hebben wij niet kunnen vinden hoe een state aan een agent toewijst met code.

Performance efficiency: How long does running your simulation take? Is the tool fast enough?

We schatten dat we rond de 400 stappen nodig te hebben voor de simulatie.

- Unity: Unity kan zware programma's draaien en zal zeker snel genoeg zijn om onze simulatie te kunnen visualiseren. Naar ons idee is Unity zelf te krachtig, er zijn meerdere mogelijkheden om de simulatie in andere programma's te draaien die minder zwaar zijn.
- Mesa: Mesa is naar ons idee krachtig genoeg om onze simulatie te draaien. De simulatie kan met dit programma goed gevisualiseerd worden.
- Netlogo: Naar ons idee is Netlogo te vergelijken met Mesa qua hoe snel deze is.

Feasibility:

Are the skills of the developers sufficient to use the tool (user-friendliness)?

- Unity: Unity maakt gebruik van de taal C#. Een taal die een beetje bekend is onder ons. Unity is in eerste instantie niet heel gebruiksvriendelijk. Er zou vele tutorials gekeken moeten worden, voordat er een goede simulatie kan worden gebouwd.
- Mesa: Mesa maakt gebruik van Python. Een taal die erg makkelijk is in vergelijking tot andere talen naar onze mening. De modules zijn goed beschreven en kunnen makkelijk worden gebruikt.
- Netlogo: Netlogo maakt gebruik van een variant van java. Deze taal is ons onbekend maar nog wel te begrijpen. De interface is erg simpel en gebruiksvriendelijk.

Is it technically feasible to create a MVP in two weeks with the current tool?

- Unity: De tijd die we nodig zullen hebben om in Unity een MVP te maken zal langer zijn dan twee weken. Dit programma is dus voor ons uitgesloten.
- Mesa: We denken dat ons het gaat lukken om een Minimum viable product te bouwen binnen twee weken in Mesa met wat extra's.
- Netlogo: Voor Netlogo denken we ongeveer aan een MVP te voldoen in twee weken.

Compatibility:

Are you using external data, and does your tool support this use?

Voor de basis gebruiken we nog geen externe data, wellicht gaan we dit gebruiken wanneer de basis af is. Mesa ondersteund het gebruik van een externe dataset evenals Unity. Naar ons weten doet Netlogo dat niet. Wij konden geen informatie vinden over het gebruik van externe data binnen Netlogo.

Experiment Design

Wat er gemeten wordt bij dit experiment.

Wij hebben doorstroom gedefinieerd als volgt: Hoe meer auto's van a naar b kunnen in een bepaalde tijd hoe hoger de doorstroom en de gemiddelde snelheid. Wij nemen deze aanname, omdat bij een hoge gemiddelde snelheid auto's minder lang doen om een afstand af te leggen dan lagere gemiddelde snelheden. Want $s = v * t$ dus als s gelijk blijft en v toeneemt moet t afnemen. Wij zullen primair testen op wat het effect is op de gemiddelde snelheid als de maximumsnelheid wordt aangepast.

Doeltreffend experiment

Door verschillende snelheden te nemen kunnen we vergelijken waarbij de gemiddelde snelheid het meest constant is. De uitkomst zal bepalen wat de beste snelheid is om de doorstroming van auto's zo vloeiend mogelijk te laten lopen. Ook testen wij deze snelheden op verschillende druktes op de weg. Wij houden de rest van de waarden constant de variabelen zijn als volgt.

Vaste waarden

Acceleration = 1
Randomization = 0.05 (Randomization is de kans dat een auto abrupt afremt.)
Width = 100 (Width houdt de grote van de baan in. In andere woorden hoeveel "tiles" de baan groot is. De baan is een grid van $1 * width$)
Vision range = 50 (is hoeveel "tiles" de auto voor zich kan zien.)

Variabele

car count = [1,2,3,4,5,6,7,8,9,10,15,20,25,50]
speed limit = [5,8,10,13]

Op elke combinatie van deze variabele wordt er getest. Bij elke combinatie worden er tienduizend steps uitvoert, voordat het programma naar de volgende combinatie gaat.

Resultaten

tabellen

Gemiddelde vertraging, standaarddeviatie en modus in km/h.

wegbezetting 1%	gemiddelde	standaarddeviatie	modus
snellheidslimiet 50	0.27	0.84	0.00
snellheidslimiet 80	0.70	1.40	0.00
snellheidslimiet 100	0.92	1.84	0.00
snellheidslimiet 130	2.65	3.19	0.00

wegbezetting 5%	gemiddelde	standaarddeviatie	modus
snellheidslimiet 50	0.39	0.38	0.00
snellheidslimiet 80	1.63	0.91	1.20
snellheidslimiet 100	2.04	1.31	1.60
snellheidslimiet 130	4.94	2.27	6.40

wegbezetting 10%	gemiddelde	standaarddeviatie	modus
snellheidslimiet 50	0.45	0.22	0.40
snellheidslimiet 80	2.65	0.51	3.00
snellheidslimiet 100	4.92	0.42	4.80
snellheidslimiet 130	7.91	0.55	7.80

wegbezetting 15%	gemiddelde	standaarddeviatie	modus
snellheidslimiet 50	1.28	0.40	1.6
snellheidslimiet 80	4.31	0.48	4.53
snellheidslimiet 100	5.98	0.62	6.46
snellheidslimiet 130	9.17	0.68	9.17

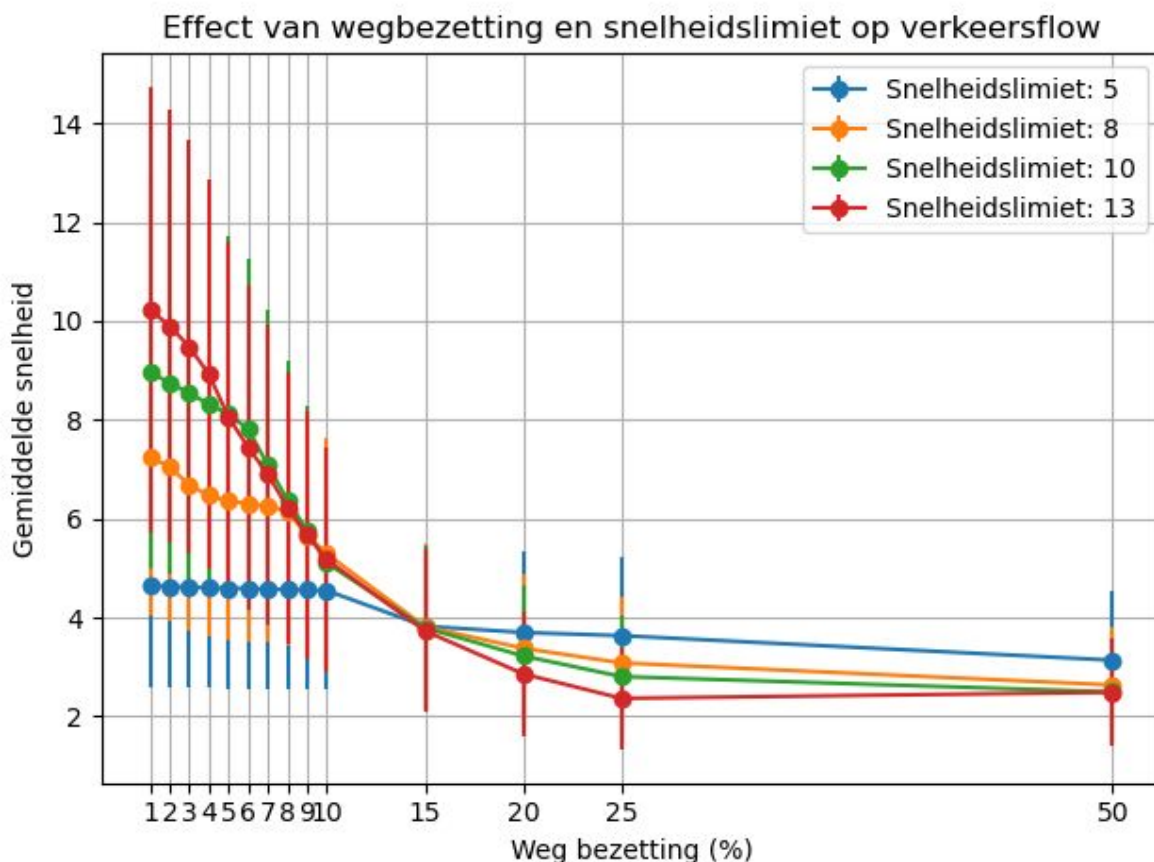
wegbezetting 20%	gemiddelde	standaarddeviatie	modus
snelheidslimiet 50	1.27	0.51	1.00
snelheidslimiet 80	4.52	0.71	5.35
snelheidslimiet 100	6.79	0.72	7.30
snelheidslimiet 130	10.23	0.55	10.50

wegbezetting 25%	gemiddelde	standaarddeviatie	modus
snelheidslimiet 50	1.55	0.58	2.00
snelheidslimiet 80	5.58	0.32	5.80
snelheidslimiet 100	7.67	0.47	7.84
snelheidslimiet 130	10.40	0.68	10.80

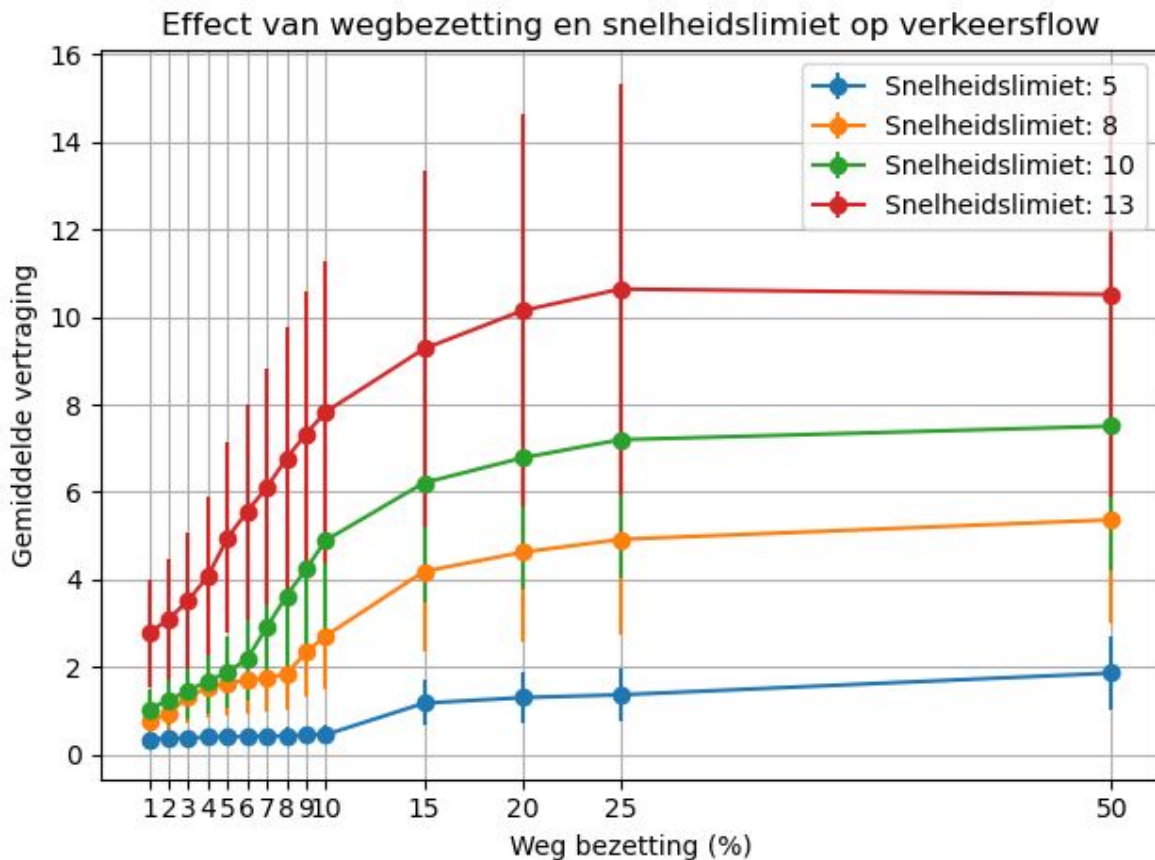
wegbezetting 50%	gemiddelde	standaarddeviatie	modus
snelheidslimiet 50	1.83	0.59	2.00
snelheidslimiet 80	5.19	0.98	5.84
snelheidslimiet 100	7.48	0.87	7.90
snelheidslimiet 130	10.49	0.85	10.86

Grafieken

De uitwijking bij elk punt geeft aan wat de afwijking kan zijn binnen een 90% betrouwbaarheid. Dat deze zo gigantisch groot is komt door het shockwave effect.



Y-as staat voor tientallen kilometers, lees 3 is 30 km/h. In de grafiek wordt omschreven per snelheidslimiet hoeveel de gemiddelde snelheid is van een auto bij een bepaalde wegbezetting op de weg. Opvallend is dat bij een wegbezetting van 5% de rode en de groene lijn elkaar kruisen. Dat deze lijnen elkaar kruisen zegt ons dat bij een wegbezetting hoger dan 5% en lager dan 8% er meer doorstroming is bij een snelheidslimiet 100 km/h dan een snelheidslimiet van 130 km/h. Uit de significantietest blijkt dat er geen significant verschil zit tussen deze twee snelheden. Wat ons het volgende verteld: We kunnen met minder dan 95 procent zekerheid zeggen dat het snelheidslimiet van 100 voor een betere doorstroming zorgt dan die van 130 bij een wegbezetting tussen de 6 en 8%. We kunnen geen goede meting met de significantietest doen. Dit ligt aan onze manier van het project. Onze test bestaat uit heel veel stappen en wij kijken op die stappen wat de snelheid is. Door de aanwezigheid van het shockwave effect lopen die snelheden erg uiteen. Dit zorgt er voor dat de significantie test de kans veel hoger schat dat een punt ergens kan liggen dan het daadwerkelijk licht. (Zo is bijvoorbeeld op het eerste punt op de lijn te zien dat er een 95% kans is dat het punt tussen 14 en 5,8 licht maar het kan nooit 14 zijn die kans is 0,0 omdat de maximumsnelheid 13 is. Het zelfde is te zien bij de andere lijnen.)



Y-as staat voor tientallen kilometers, lees 4 is 40 km/h. In bovenstaande grafiek wordt per snelheidslimiet weergegeven hoeveel vertraging er gemiddeld per percentage wegbezetting ontstaat. We zien in de grafieken dat bij een lagere snelheidslimiet de gemiddelde vertraging minder snel oploopt dan de hogere snelheidslimieten. Dit zegt ons dat er meer geremd wordt bij een hogere dan een lage snelheidslimiet. Ook valt het op dat de gemiddelde vertraging bij wegbezetting boven de 20% minder snel oploopt dan alle wegbezetting percentages onder de 20%. Hieruit is op te maken dat bij rustige wegen die drukker worden er meer vertraging ontstaat dan bij een wegen die gemiddeld bezet zijn en drukker worden.

Conclusie

De onderzoeksvraag was: Onder welke omstandigheden heeft het verlagen van de maximum snelheid een positieve invloed op de doorstroom van het verkeer.

Uit de simulatie blijkt dat wanneer de wegbezetting hoger wordt en de er beter een lagere maximumsnelheid gehanteerd kan worden. Bij de vertragingen die ontstaan bij minder grote snelheden is de overgang van de max snelheid naar de snelheid van een vertragingen en weer terug naar max snelheid een stuk kleiner. Uit de grafieken komt zelfs naar voren dat bij een snelheidslimiet van 100 er gemiddeld genomen tussen een wegbezetting van 6% en 8% harder kan worden gereden dan bij een snelheidslimiet van 130. Uit de significantietest blijkt dat er geen significant verschil zit tussen deze twee snelheden. Wat ons het volgende verteld: We kunnen met minder dan 95 procent zekerheid zeggen dat het snelheidslimiet van 100 voor een betere doorstroming zorgt dan die van 130 bij een wegbezetting tussen de 6 en 8%.

Bij het bekijken van een visualisatie van de simulatie komt duidelijk naar voren dat wanneer er hogere snelheidslimieten worden gehanteerd de shockwave intenser wordt. Dat wil zeggen auto's meer moet afremmen en de vorm van de shockwave ook duidelijker wordt: Er lijken meer compactere golven te ontstaan en de afstand tussen auto's die nog niet hebben geremd groter. Het stuk waar de golf compact wordt, lijkt minder ruimte tussen de auto's te ontstaan dan bij lagere snelheden.

Welke snelheidslimiet gehanteerd moet worden ligt aan de drukte op de weg. Bij een wegbezetting van <5% zal een hoog snelheidslimiet van 130 beter werken voor een goede doorstroming. Tussen 5% en 8% is 100 ideaal. Tussen 8% en 15% is 80 ideaal. En boven de 15% bezetting is een lage snelheid van 50 optimaal. Omdat wij geen accurate significantie testen hebben kunnen uitvoeren is er helaas geen significant verschil gevonden wat ons gegronde informatie geeft over de snelheidslimieten, maar aan de grafiek te zien zal de realiteit niet veel afwijken van bovenstaande stellingen.

Discussie

Is lage doorstroom optimaal voor de automobilist

Wij hebben nu onderzocht wat het effect op de doorstroom is. Daar komt dan wel de vraag bij of het minimaliseren van de doorstroom een juiste keuze is. In andere woorden of het minimaliseren van de doorstroom de maatschappij en de automobilisten ten goede komt. Door onze definitie van doorstroom is het een mogelijkheid dat een auto erg lang in de file staat maar daarna weer erg hard rijdt wat zorgt voor een hoge gemiddelde snelheid. Terwijl een auto die geen file heeft maar een lagere maximumsnelheid een lagere gemiddelde snelheid heeft. De doorstroom zal dus in de eerste situatie hoger zijn maar is dit wel positief voor de automobilist want die geeft misschien wel de voorkeur voor situatie 2 waarin de automobilist geen vertraging heeft.

Mogelijke uitbreidingen vervolg experiment

Ons experiment is beperkt gebleven tot een baan. Zolang het gaat over het effect op het shockwave effect is dit geen probleem omdat dat hetzelfde of erger is met meerdere banen (een shockwave effect wordt erger als auto's geforceerd vaart moeten minder om bijvoorbeeld een afslag te maken). Maar omdat wij ons wel op maar een baan hebben gefocust is onze conclusie niet doortrekken tot concreet beleid of tot een conclusie over het ontstaan van files.

Verwachtingen en resultaat

Het resultaat komt overeen met onze eerdere verwachtingen. Wij hadden verwacht dat als de maximum snelheid afneemt het shockwave effect minder wordt, omdat er met een lagere snelheid minder afgeremd en versneld hoeft te worden. Wat ervoor zorgt dat het shockwave effect afneemt, omdat het afremmen en optrekken zo minder tijd kost.

Opvallende resultaten

We verwachten dat de vertraging geleidelijk op zou lopen naarmate de wegbezetting stijgt tot er een file ontstaat. Echter klopt dit niet helemaal volgens de grafieken. De file ontstaat namelijk niet en het laagst gemeten gemiddelde snelheid is 22 km/h. Mogelijk is dat de auto's hier bij een hoge bezetting allemaal een stap tegelijkertijd vooruit kunnen rijden. In het echt zou een auto niet zo snel kunnen anticiperen op de voorganger.

Extra uitbreidingen

Wij hebben onmiddellijk van tijd een aantal dingen niet meegenomen een aantal uitbreidingen die wij kunnen toevoegen.

Meerdere banen

Door gebruik te maken van meerdere banen en eventueel ook een invoeg en uitvoegstrook kan het effect van auto's die geforceerd moeten afremmen om van baan te wisselen of afslaan meegenomen worden. Wij verwachten dat het shockwave effect dan groter wordt. Of het verlagen van de maximum snelheid dan nog effect heeft is onbekend. Zoals hierboven als is beschreven kan door het ontbreken hiervan onze conclusie niet worden doorgetrokken tot concreet beleid of tot een conclusie over het ontstaan van files.

Auto's met verschillende snelheden

Al onze auto's willen nu de maximum snelheid proberen te rijden. Dit is niet realistisch omdat sommige automobilisten er voor kiezen om onder de maximum snelheid te rijden. Dit om veiligheid en zuinigheids redenen. Met name vrachtwagens kiezen hier vaak voor. Dit zorgt ervoor dat andere auto's de betreffende auto moeten inhalen of ook moeten afremmen. Wij verwachten dat het shockwave effect dan groter wordt. Of het verlagen van de maximum snelheid dan nog effect heeft is onbekend.

Wegbezetting van Nederland

Er kunnen op verschillende plekken in de wereld een opstopping ontstaan. Om de antwoorden van de simulatie te specificeren kan gekeken worden wat de gemiddelde wegbezetting is in Nederland op de snelwegen. Met de Nederlandse wegbezetting data kan in combinatie met de info vanuit de grafieken uitgelezen worden wat ons model als beste snelheid voorschrijft voor de Nederlandse wegbezettingen.

Bronnen

1. Schadschneider, A. (1999). The nagel-schreckenberg model revisited. *The European Physical Journal B-Condensed Matter and Complex Systems*, 10(3), 573-582.