

Touring Machines Traffic Simulation

Sinem Ertem, Martijn Knecht & Gijsbert Nutma



Inhoudsopgave

Inhoudsopgave	2
1. Inleiding	3
1.2 Onderzoeksvraag	3
1.3 Plan van aanpak	4
1.3.2 Agent	4
1.3.3 Environment	4
1.3.5 Toolkeuze	4
1.3.5.2 Suitability:	4
1.3.5.3 Feasibility:	4
2. Uitleg design	6
3. Resultaten experiment	7
4. Conclusie	8
5. Discussie	9

1. Inleiding

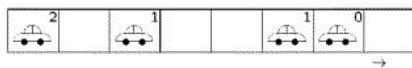
Voor opdracht 5 van Simulation Tooling zullen we een ABM (agent-based model) gaan bouwen voor het verkeer. Wij zullen ons vooral gaan focussen op verkeerscongestie. Dit is een tijdelijke verstopping in het verkeersnetwerk wat in ons geval veroorzaakt wordt door deceleratie. Voor het model wat wij gaan maken bestaat er al een theoretisch model voor filevorming genaamd: 'The Nagel-Schreckenberg model'. Dit model is rond 1990 ontworpen door twee Duitse natuurkundigen K. Nagel en M. Schreckenberg. Het is een 2-dimensionaal model bestaande uit één rijbaan. De rijbaan bestaat als het ware uit patches en per patch kan er slechts 1 auto rijden, anders heb je een overlapping. Verder rijden de auto's dezelfde richting op met een snelheid tussen 0 kilometer per uur en de maximale toegestane snelheid die wij hebben gesteld op 1.

De volgende stappen worden door een iteratief proces genomen per auto op de rijbaan:

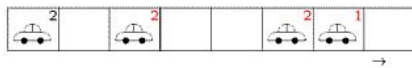
1. **Acceleratie:** de snelheid van alle auto's worden met 0.1 kilometer verhoogd als de snelheid groter is dan 0 en kleiner is dan 1
2. **Deceleratie:** als de afstand tussen de auto's klein is neemt de snelheid van de auto erachter af
3. **Randomization:** de snelheid van alle auto's met een minimale snelheid van 1 worden verlaagd met een kans voor p . Als $p=0.5$ (dus 50%) en de huidige snelheid is 4, wordt de snelheid in 50% van de gevallen gereduceerd naar 3.
4. **Beweging:** alle auto's bewegen x aantal patches vooruit (naar rechts) gelijk aan de snelheid. Als de snelheid 2 is zijn dat 2 patches op de rijbaan vooruit.

figuur 1: Nagel-Schreckenberg model per iteratie

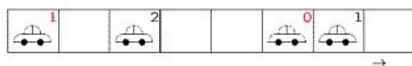
Configuration at time t :



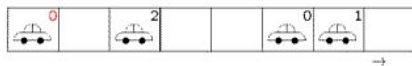
a) Acceleration ($v_{max} = 2$):



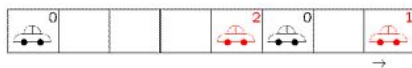
b) Braking:



c) Randomization ($p = 1/3$):



d) Driving (= configuration at time $t + 1$):



1.2 Onderzoeksvraag

Met de kennis die we hebben opgedaan door grotendeels het Nagel-Schreckenberg model en voorbeeld simulaties die NetLogo heeft vrijgegeven hebben we een onderzoeksvraag geformuleerd die ons helpt bij het verder onderzoeken van filevorming:

“Bij hoeveel auto's ontstaat er bij verschillende constante snelheden file op een weg wanneer er door een auto hard geremd wordt, een file wordt gedefinieerd door auto's die niet meer hun originele snelheid hebben?”

1.3 Plan van aanpak¹

1.3.2 Agent

De agenten uit ons model zullen alleen bestaan uit auto's. Deze hebben weer elk hun attribuut 'speed' (=snelheid). Zo beginnen ze met een gegeven start-speed en zal de snelheid tijdens de simulatie verder toenemen tot een maximum snelheid van 1 (waar mogelijk). De regels die de agenten verder meekrijgen zijn:

- Remmen/stoppen als de agenten te dicht bij elkaar in de buurt komen. Hiervoor hebben we een 'target-brake' die dit beïnvloedt. Zodra de gebruiker van de simulatie klikt op deze button zal de target-auto (rood van kleur) op de rem gaan, dit betekent dat de snelheid van die auto daalt naar 0 en vervolgens automatisch zijn snelheid weer zal oppakken.
- Op een baan blijven. Aangezien onze simulatie is gebaseerd op een rijbaan met de y-coördinaten op kleiner dan 2 en groter dan -2.
- Auto's kunnen niet door elkaar heen. Het is de bedoeling dat er op 1 patch van de rijbaan maar 1 auto op kan rijden omdat de auto's elkaar anders gaan overlappen.

1.3.3 Environment

Zoals hierboven al toegelicht zullen de auto's alleen op één rijbaan gaan rijden. De rijbaan bestaat uit een grid met cellen/patches. De 'wereld' heeft een grootte van 25 bij 4 en de rijbaan is zoals in het *Nagel-Schreckenberg model* oneindig. Het einde van de rijbaan ($x_{cor} = 25$) is verbonden met het begin ($x_{cor} = -25$). De agent en environment hebben een wisselwerking per tijdstap voor $t = 0, 1, 2, 3, \dots$. Voor elk tijdstap ontvangt de agent de huidige staat van de environment en zal op basis van die informatie actie ondernemen. In ons model zullen de agents alleen op elkaar reageren.

1.3.5 Toolkeuze

Aan de hand van het SFA model zullen we de verschillende tools: Mesa, Unity en NetLogo tegen elkaar afwegen.

1.3.5.2 Suitability:

- *Mesa*: de user story kan zeker gemaakt worden in mesa het zou echter alleen niet gemakkelijk zijn om agent tot agent action te krijgen waardoor ze de afstand van elkaar altijd kunnen zijn. Verder wordt het in python gecodeerd en python is niet de snelste meest efficiënte programmeertaal waardoor het kan zijn dat stappen langzamer uitgevoerd worden dan gewild hierdoor zal de animatie haperend verlopen.
- *NetLogo*: De User Story kan gemaakt worden in NetLogo, de tool is geschikt voor simulaties, ook hebben we een voorbeeldsimulatie in NetLogo gezien die lijkt op de simulatie die wij willen maken.
- *Unity*: Unity is per definitie geschikt voor de User Story, een baan met een object (auto) erop is te modelleren in het programma. Hier moet wel rekening gehouden worden met de soms lastige omgeving van Unity, aangezien deze meer geschikt is voor games dan simulaties.

1.3.5.3 Feasibility:

- *Mesa*: Mesa heeft geen inspector, soms kan het lastig zijn om te volgen hoe de simulatie loopt.

¹ Zie *model.png* voor volledige visualisatie van de planontwikkeling

- *NetLogo*: Het gebruiksvriendelijke aan NetLogo is dat het al een ‘Models Library’ bevat waardoor gebruikers al gemaakte simulaties kunnen inladen en uitvoeren. Verder zijn er drie verschillende kolommen; interface waarin de simulatie zichtbaar is, waarin verdere informatie staat met uitleg en code waarin de gebruiker kan zien hoe de simulatie is opgezet en zelf nog aanpassingen kan doorvoeren. Door al deze elementen is het mogelijk om binnen de deadline een eigen simulatie te maken.
- *Unity*: Doordat Unity erg uitgebreid is, kan het lastig worden om deze tool binnen twee weken te leren en te gebruiken om een goede simulatie mee te maken.

Door ons te focussen op de geschiktheid en haalbaarheid zijn we tot de conclusie gekomen om voor NetLogo te gaan.

2. Uitleg design

Met het model wat we hebben gebouwd willen we onderzoeken bij hoeveel auto's met een aangehouden snelheid er een file ontstaat, wanneer de target-auto hard op de rem gaat. Het experiment begint met een x aantal auto's die met behulp van een slider vastgesteld kan worden. Zodra de maximumsnelheid ook is vastgesteld wordt er op setup geklikt en het model zal bij elke set-up een willekeurige auto als target auto nemen. De target auto die in het rood is aangegeven zal na x -ticks, die wij hebben gesteld op 20, zijn snelheid gaan verlagen naar 0.1. Omdat de target auto als het ware op de rem gaat heeft dit effect op de rest van de auto's en ontstaat er een file omdat de auto's voor een bepaalde tijd niet meer hun maximale snelheid behouden.

In ons model maken gebruik van de volgende variabelen en buttons:

- *Setup*: om een simulatie goed te laten verlopen wordt setup gebruikt om alle ticks te verwijderen en de turtles opnieuw te plaatsen. Voor elke setup wordt er ook een nieuwe target-car gekozen.
- *Go*: deze button zorgt ervoor dat de turtles gaan bewegen en de simulatie runt.
- *Target-brake*: hierdoor zal de target-car zijn huidige snelheid vervangen door de snelheid die hierin wordt gegeven, in ons geval is dat 0.1.
- *Simulation*: om de simulatie te starten kan er op deze knop gedrukt worden. De simulatie zorgt er voor dat de target auto bij een x aantal ticks automatisch zal gaan remmen.

Verder maken we ook gebruik van sliders voor:

- *Max-speed*: De maximumsnelheid die de auto's kunnen rijden. Dit is ook de snelheid waarop de auto's zullen beginnen met rijden.
- *Amount-of-cars*: Het aantal auto's die rijden op de baan. Deze blijven constant doordat auto's die aan de rechterkant verdwijnen aan de linkerkant weer op de baan komen.
- *Max-ticks*: De maximale duur van 1 simulatie.

Om het experiment te visualiseren gebruiken we een grafiek die de snelheid meet van de target auto en de minimale snelheid van de rest van de auto's. We willen hiermee zien hoe lang het duurt voordat de auto's weer hun maximale snelheid aanhouden. De factoren die we in het onderzoek mee willen nemen zijn:

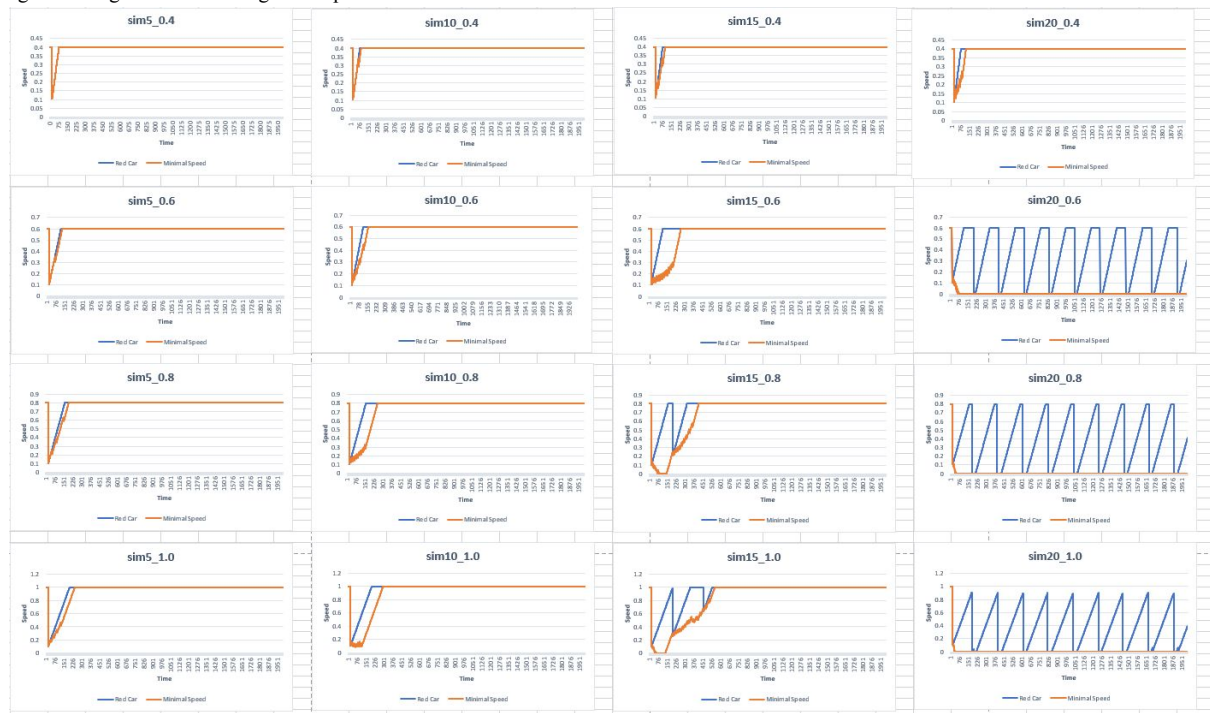
- Aantal auto's: zal een file inderdaad langer duren wanneer er meer auto's op de rijbaan rijden.
- Maximum snelheid: maakt het uit of je het model begint met een maximumsnelheid van 0.0, 0.2, 0.5 etc.
- Target Brake: dit houdt bij naar welke snelheid de target-auto remt. Als de target-auto zijn maximale snelheid heeft behaald en vervolgens op de rem gaat en zijn snelheid daalt. In hoeverre heeft de daling van zijn snelheid effect op de rest van de auto's en daarmee de langdurigheid van de file.

3. Resultaten experiment²

Om een aantal tests uit te voeren hebben we verschillende waarden genomen om onze simulatie te testen. We hebben gekozen om het aantal auto's variabel te maken door stappen van 5 auto's te nemen met een maximum van 20. Op deze manier krijgen we aantallen van 5, 10, 15 en 20 auto's. Daarnaast hebben we de maximumsnelheid van de auto's variabel gemaakt, de snelheid loopt van 0.4 tot 1.0 met stappen van 0.2. Op deze manier krijgen we maximumsnelheden van 0.4, 0.6, 0.8 en 1.0.

We kunnen ook de lengte van de simulatie aanpassen, wij hebben bij ons experiment gekozen voor een lengte van 2000 ticks. Deze waarde hebben we gekozen omdat in het geval van een kleine hoeveelheid auto's de maximumsnelheid weer bereikt kan worden. De rest van de simulatie blijft dan gelijk waardoor we heel veel constante data overhouden.

figuur 2: Nagel-Schreckenberg model per iteratie



Bij de resultaten van de simulatie kunnen we zien dat de meeste gesimuleerde vertragingen zichzelf oplossen. Bij 20 auto's is te zien bij hogere snelheden het niet mogelijk is om de vertraging definitief op te lossen, hier blijft de opstopping voor de rest van de simulatie.

² Volgende resultaten staan ook in het bestand "Simulation results.xlsx"

4. Conclusie

Onze onderzoeksvraag zoals in het begin van het verslag staat vermeld luidt:

“Bij hoeveel auto's ontstaat er bij verschillende constante snelheden file op een weg wanneer er door een auto hard geremd wordt, een file wordt gedefinieerd door auto's die niet meer hun originele snelheid hebben?”

In de resultaten van ons experiment kunnen we een verhouding terugzien tussen het aantal auto's en de maximumsnelheid die de auto's kunnen hebben. Hierbij kunnen we zien dat hoe meer auto's de weg bevat, hoe groter het effect is op de vertraging. Hetzelfde geldt voor de snelheid van de auto's: Hoe groter de snelheid, hoe groter het effect op de vertraging.

Om antwoord te geven op onze onderzoeksvraag: Bij een constante snelheid van 1.0 en 18 auto's ontstaat er een niet-oplosbare file op de weg. Na het uitvoeren van het experiment hebben de we exacte bottleneck berekend, deze komt uit op 18 auto's. We hebben hier gekozen voor de maximale snelheid die we hebben getest, deze snelheid is tevens ook ingesteld als maximaal in de simulatie.

5. Discussie

Voordat we daadwerkelijk onderzoek hadden gedaan, hadden we wel verwacht dat er uiteindelijk een file ontstaat wanneer de auto's niet meer hun eigen snelheid behouden en door een lage snelheid de rest van de weggebruikers belemmert. Het was nog de vraag bij *hoeveel* auto's dit ontstaat en welke factoren daadwerkelijk effect hebben. In de simulatie met 20 auto's kun je bijvoorbeeld zien dat er bij een snelheid van 0.4 de auto's uiteindelijk weer terug vallen naar de maximumsnelheid. Maar passen we de snelheid aan naar 0.6, dan het fileprobleem eigenlijk niet opgelost worden.

Aanvullend aan het onderzoek wat wij hebben uitgevoerd, zou er wat aan het model aangepast kunnen worden. Zo hebben wij zoals het *Nagel-Schreckenberg model* een oneindige rijbaan waarin de auto's aan de rechterkant eruit rijden en via de linkerkant weer binnenrijden. Een aanpassing zou kunnen zijn om het niet oneindig te laten lopen en te onderzoeken wat het effect hiervan is en veel of niet afwijkt van het resultaat wat wij nu hebben.

De simulatie is niet volledig realistisch omdat de agent's in het model allemaal dezelfde snelheid aanhouden en de afstand tussen de agent's hetzelfde is. In het echte leven zijn de bestuurders verantwoordelijk voor hun gedrag in de auto en dit heeft op elk moment van de dag, rekening houdend met bijvoorbeeld het weer en status van zowel de auto als de wegen.