[]: Een snel algoritme voor het spel Rushhour

Oscar Keur\*

Universiteit van Amsterdam

emailadres@email.uva.nl

Kyra Kieskamp\*

Universiteit van Amsterdam

Kyra.kieskamp@student.auc.nl

# 1. Inleiding

*Hier schrijf je een inleiding die in elk geval kort, bondig en compleet de hele vraagstelling bevat. Ook moet er een inschatting komen van de toestandsruimtegrootte (belangrijk, daar deden we het voor), en eventuele restricties op transities in beschreven worden, maar nog niet de methodes die je gebruikt om de toestandsruimte te doorzoeken.*

Game

Rushhour is a sliding block puzzle created in the seventies by Nob Yoshigahara. The initial form of the game is a 6x6 playing field in which various cars and trucks are situated. The cars have a lenght of 2 boardplaces and the trucks have a length of three board places. The cars and trucks can be placed either horizontally or vertically on the board. The goal of the game is to get the red car to the exit of the board, which is the opening at the right hand side, see figure 1. This can be achieved by step-wise moving the cars and trucks on the playing field to a new position untill the red car can exit the board. The movement restrictions are that cars and trucks can only be moved in their length direction, only sliding movements are allowed (so no picking up), and no movement through another car/truck is allowed.

*Rushhour is een schuifpuzzel die in de jaren zeventig is bedacht door* [Nob Yoshigahara](https://nl.wikipedia.org/w/index.php?title=Nob_Yoshigahara&action=edit&redlink=1) (Wikipedia). Het spel draait om het rode autotje, die als het ware in de file staat in een 6 x6 speelvak en door andere auto’s en vrachtwagens wordt geblokkeerd. Het doel is om alle auto’s en vrachtwagens zo te verplaatsen (in hun lengte richting), dat het rode autotje uiteindelijk bij de uitgang terecht komt. Er zijn verschillende beginposities van alle auto’s en vrachtwagens mogelijk op het 6 bij 6 speelbord, die de moeilijkheidsgraad van het spel bepalen. Ook zijn er alternatieve spellen qua grootte van het speelbord. De autotjes hebben een lengte van 2 vakjes en de vrachtwagens een lengte van 3 vakjes.

*Het doel van onze algoritmen, is om het spel op te lossen in een zo min mogelijk aantal stappen.*

1. *Inschatting van de toestandruimtegrootte (afhankelijk van aantal mogelijke zetten, dus hoeveel lege plekken er zijn)*
2. *Restricties*
3. *Alle auto’s zijn qua grootte gelijk aan elkaar, zo ook alle vrachtwagens. Behalve de kleur rood (de kleur van de auto die naar de uitgang moet), hebben de kleuren in het spel alleen een esthetisch doeleinde.*



Figuur 1 Small-scale 6x6 version of Rushhour, with both cars as trucks. Both cars and trucks are only allowed to move length-wise is a , cannot be picked up, and cannot move through anohter car/truck. If an adjacent position in front or behind the car is free, the user can move the car to that spot. The game is solved if all the cars and trucks are moved in such a way that the red car can exit the board at the opening on the right side.

Eternity II is een edge-matching puzzle uitgegeven door Tomy in november 2012. Het doel is bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla $1.000.000 te winnen.

*Figuren moeten altijd genummerd zijn, en \_eigenlijk\_ moet er ook altijd naar verwezen worden in de tekst. Een goede figuur maken is een kunst. Het kan enorm bijdragen aan de toegankelijkheid van je artikel, maar een slechte figuur doet afbreuk. Iedere figuur heeft een bijschrift (of ‘caption”). Bijschriften zijn zo kort mogelijk, maar niet korter dan dat (moeilijke nuance). Als richtlijn kun je voor een eenvoudige figuur als deze met twee tot drie korte duidelijke zinnen klaar zijn.*

Ieder stukje kan op 4 manieren geplaatst worden bla bla bla bla bla bla bla bla bla bla bla. De bruto toestandsruimte van Eternity bevat daarom 81226 \* 1027 elementen. Stukjes 13 en 71 zijn identiek, net als bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla. De netto toestandsruimte bevat daarom 2341 \* 1023 elementen, dit aantal is te groot om met een exhaustive method binnen redelijke tijd tot een goed einde te komen.

# 2. Methodes

*In deze paragraaf beschrijf je de methodes die je hebt gebruikt, liefst in de volgorde waarin je ze hebt toegepast (in dit geval drie). Die hoeft niet per sé te kloppen met de tijdsvolgorde, echte wetenschappers proberen ook vanalles door elkaar. Wat wél moet kloppen zijn de details van de algoritmes, en de bijbehorende resultaten in de volgende paragraaf. Je moet zoveel informatie geven dat je experiment in principe herhaalbaar is, en je resultaten reproduceerbaar, ook je algoritme een stochastisch element (random-functie) bevat. Echte stoere mensen maken al hun resultaten, data en sourcecode ook online beschikbaar. Dat is nu nog niet overal gangbaar, maar gaat het wel worden (mijn inschatting)*

Er zijn voor dit onderzoek drie verschillende algoritmes gebruikt, namelijk een aangepaste versie van Random, Breadth-first search en Depth-first search.

## 2.1 Random (met aanpassingen)

*Gebruik meerdere subparagrafen als je dat nodig vindt. Bij meerdere methodes is dat vaak een goed idee.*

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla, maar in onze variant geven de voorkeur aan bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla omdat bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

## 2.2 Breadth-first

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

## 2.3 Depth-first

In tegenstelling tot eerdergebruikte methodes, heeft het BIDIBENCH bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

# 3. Resultaten

*Hier bespreek je heel droog je resultaten. Als je statistieken hebt: toevoegen. Als je vergelijkingen hebt met randomposities: toevoegen. Alles is woord en getal, alle details en het liefst ook het één en ander in grafieken, plaatjes of anderzins.*

Fig. 2. Aantal misfits per algoritme, gemiddeld over 5 trials. Ondanks dat alle methodes de puzzel relatief snel oplosten, is het BIDIBENCH algoritme zowel qua eindresultaat als qua algehele trajectperfomance te prefereren over beide andere methodes. *Een goede grafiek is, net als een goed plaatje, in één oogopslag duidelijk en vereist weinig arbeid van de lezer. Voorzie de assen van labels, de grafiek van een titel en kies je kleuren zo dat ze maximaal contrasteren en dus gemakkelijk leesbaar zijn. De legenda rechts van deze grafiek had rechtsbovenin het plaatje gemogen, dat had ruimte bespaard, had de grafiek iets groter kunnen zijn en dus beter leesbaar.*

# 4. Conclusies

*Hier schrijf je je conclusies, eventuele overdenkingen (hoe zou het nog beter kunnen, is het algoritme ook in andere gebieden toepasbaar).*

# 5. Referenties

*Als je literatuur hebt gebruikt, hier toevoegen. Als je eraan refereert in de tekst, zet je op die plek alleen [1], zodat mensen achterin de details kunnen vinden. Als je geen literatuur gebruikt, weglaten.*

*Wat ook nog kan is een dankwoord, bijvoorbeeld voor mensen die wel geholpen hebben maar geen auteur zijn, mensen die je een inzicht hebben gegeven, of administrators die je even hun supercomputer hebben laten gebruiken. Altijd naam en bedrijf noemen en zorgen dat de bedankte persoon zich er goed over voelt.*

*Als je zowel een dankwoord als een referentiesectie hebt: de referentiesectie is \*altijd\* het laatste onderdeel van je verslag.*

[1] Artificial Intelligence, a modern approach, Russel & Norvig,3rd Edition, Addison-Wesly, pg 287-387.

[2] “A packing problem with applications to lettering of maps” Michael Formann and Frank Wagner (1991) SCG '91 Proceedings of the seventh annual symposium on Computational geometry