



# Rush Hour

**BigBrainz**

Jordy Schifferstein

Leon Besseling

Yassin El-Baz

# Case: Rush Hour

- Doel van spel
- Constraints
- Wat maakt de case moeilijk?
- Variaties:
  - 6x6
  - 9x9
  - 12x12

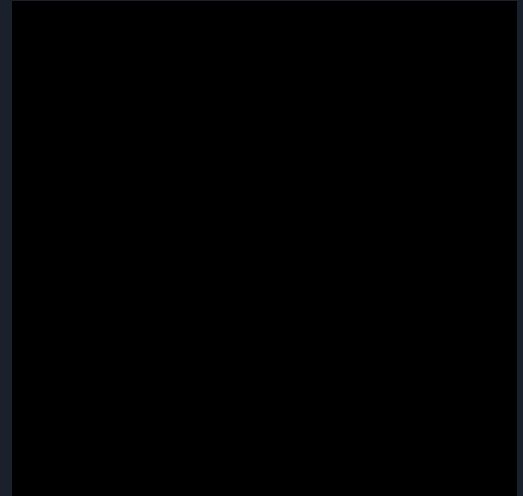




# Onderzoek

“Wat is de beste manier om een computer Rush Hour op te laten lossen?”

- Wat definieert een goede oplossing?
- Hypothese: breadth first search





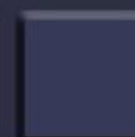
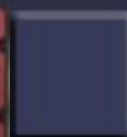
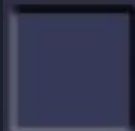
# State space

Aantal moves:

- Lower bound: aantal blokkerende auto's + 1
- Upper bound: willekeurig groot

Bordconfiguraties:

- Upper bound:  $(b - 1)^{\# \text{ auto's }} \cdot (b - 2)^{\# \text{ vrachtauto's }}$ 
  - $b = \text{bordgrootte}$
- 6x6\_3:  $(6 - 1)^6 \cdot (6 - 2)^3 = 1\,000\,000$
- 12x12:  $(12 - 1)^{28} \cdot (12 - 2)^{16} = 1.4 \cdot 10^{45}$





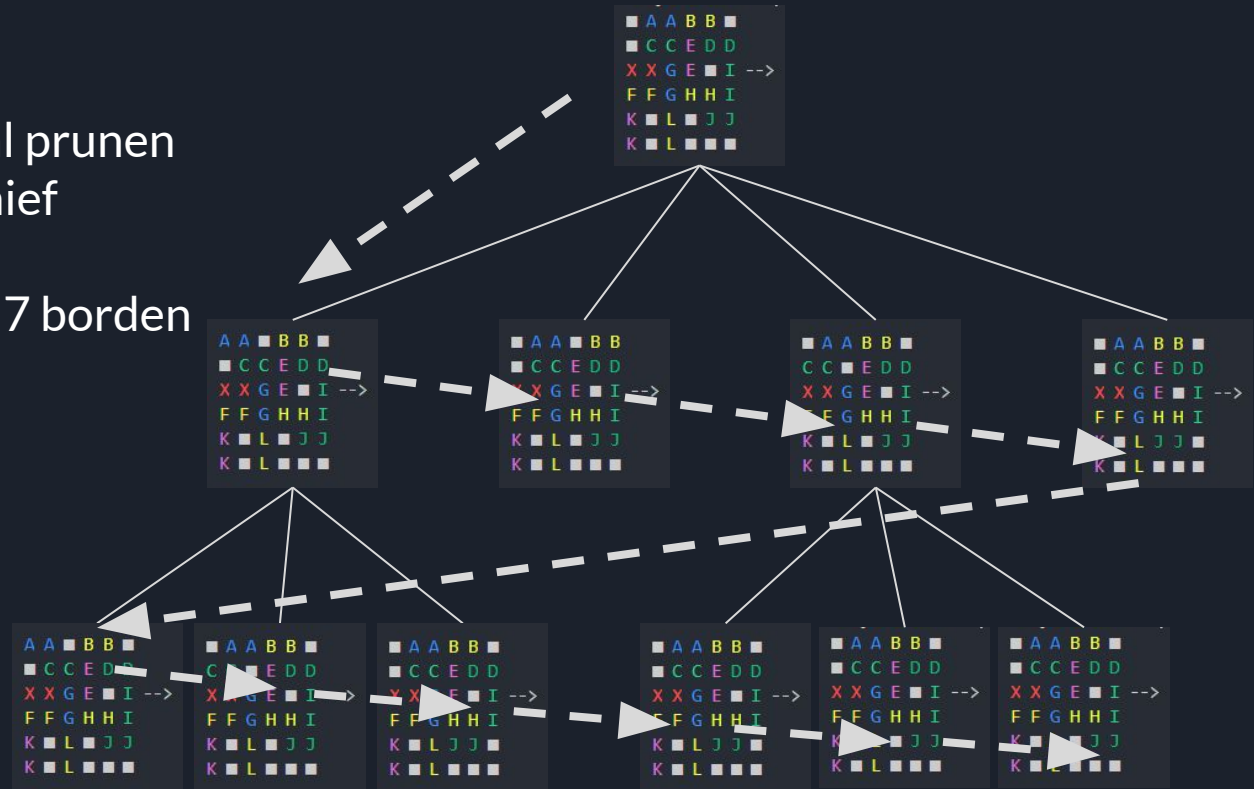
# Random

- Lost alle borden snel op
  - Veel stappen
- Heuristiek:
  - Beweeg rode auto naar uitgang indien vrij



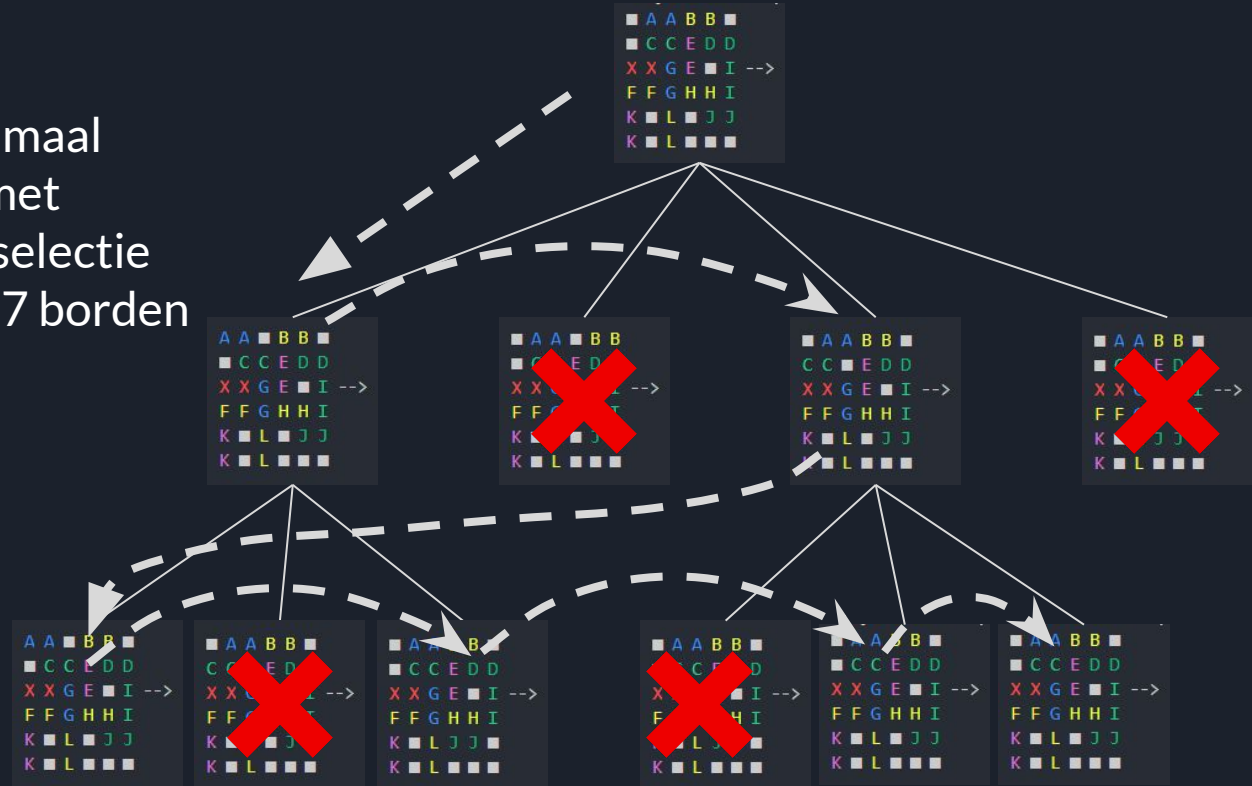
# Breadth first

- Optimaal prunen met archief
- Nadelen
- 5 van de 7 borden opgelost



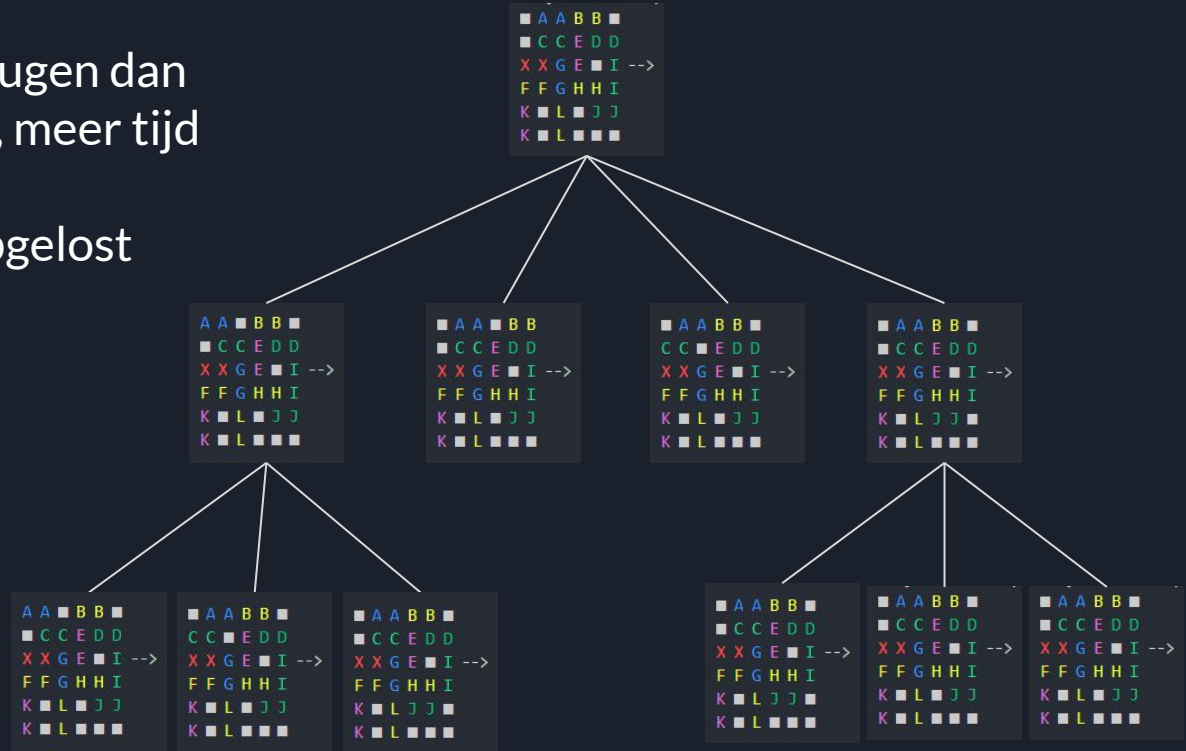
# Breadth first with beam search

- Niet optimaal prunen met random selectie
- 5 van de 7 borden opgelost



# Iterative deepening

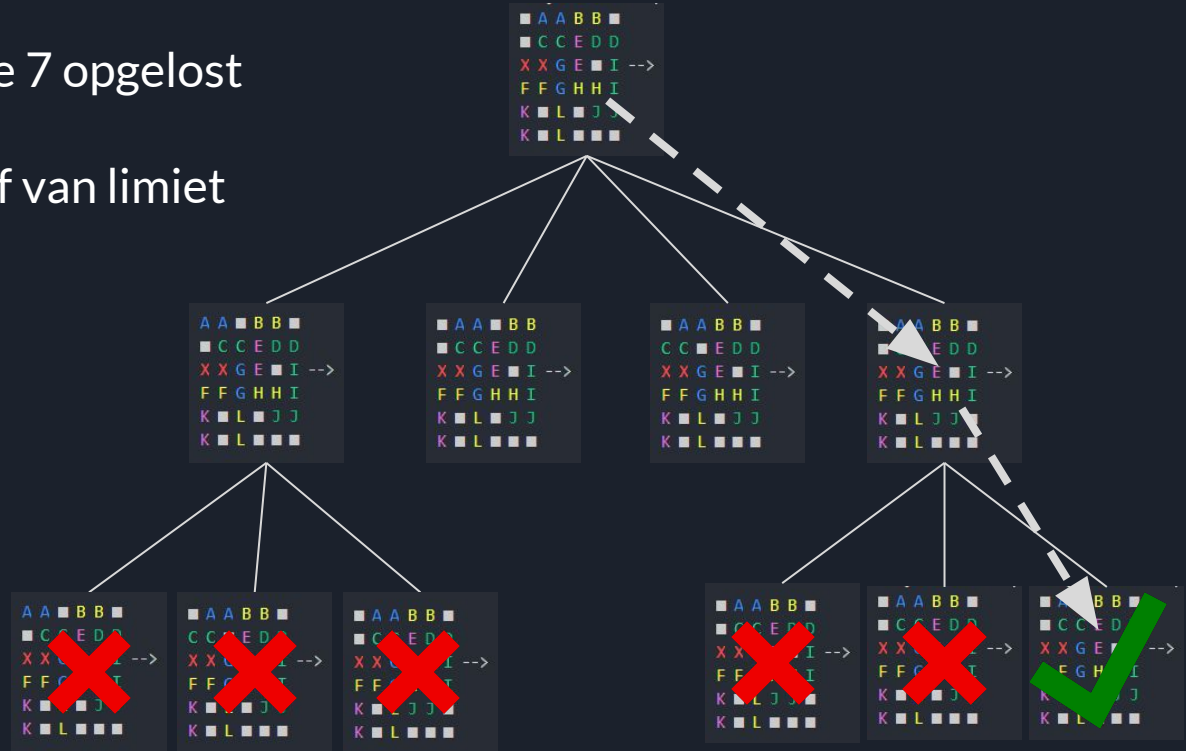
- Minder geheugen dan breadth first, meer tijd
- 6 van de 7 opgelost





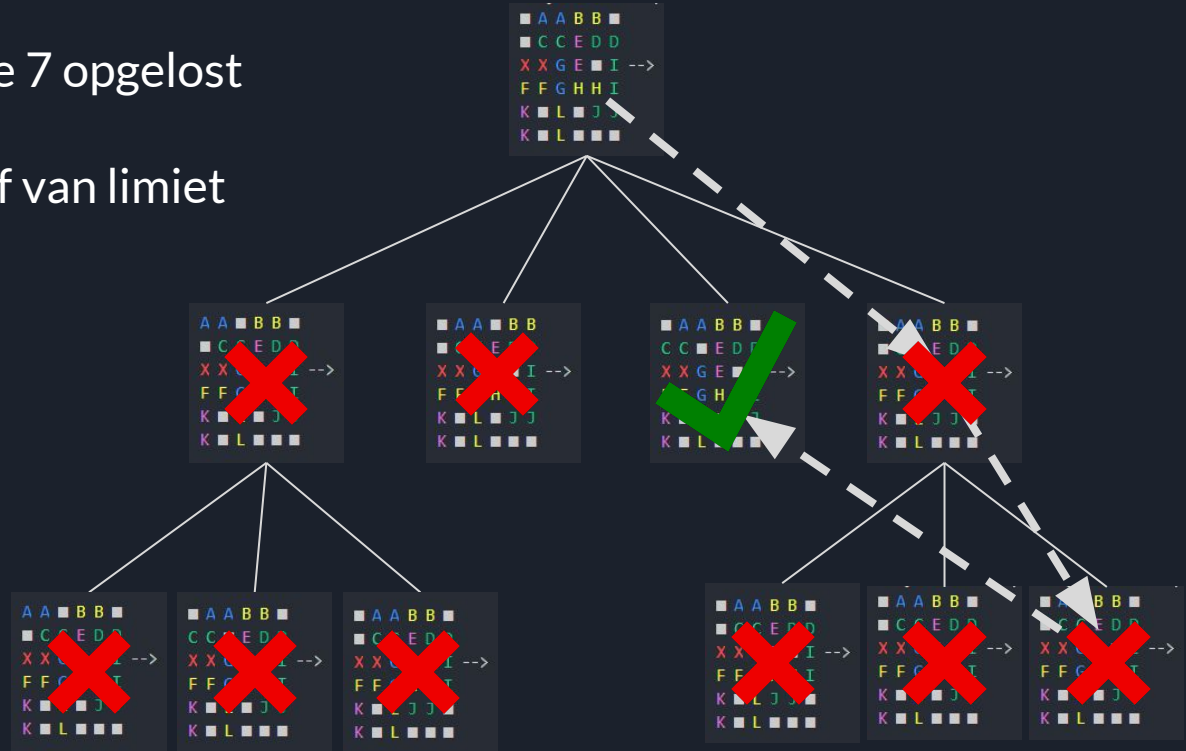
# Depth first: Branch and bound

- BnB: 4 van de 7 opgelost
- Hangt zeer af van limiet op diepte



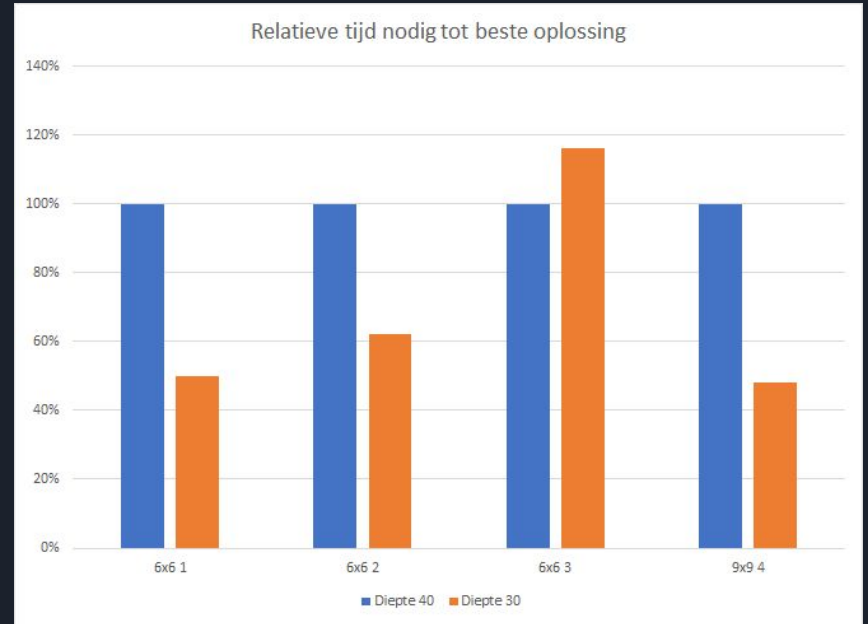
# Depth first: Branch and bound

- BnB: 4 van de 7 opgelost
- Hangt zeer af van limiet op diepte



# Onderzoek: limiet Branch & Bound

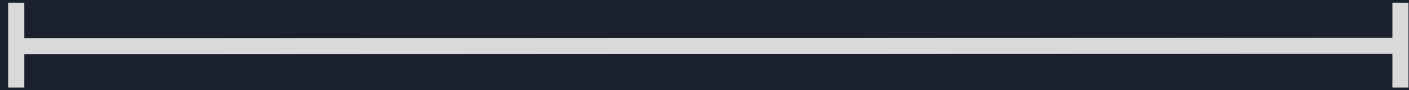
- Alle borden onder 40 moves, maar slechts 2 boven 30.
- Dynamisch door fail-save:
  - Iteratief dieper
- Dus: diepte 30 als instelling





# Hillclimbing: selective elimination

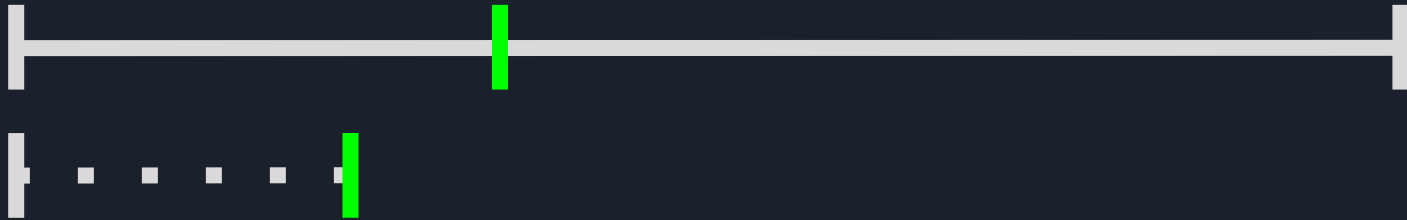
- Random oplossing





# Hillclimbing: selective elimination

- Random oplossing
- Zet random stappen vanaf het begin





# Hillclimbing: selective elimination

- Random oplossing
- Zet random stappen vanaf het begin





# Hillclimbing: selective elimination

- Random oplossing
- Zet random stappen vanaf het begin



# Onderzoek: iteraties Hillclimber



Iteraties: 100





# Hillclimbing: selective elimination

- Random oplossing
- Zet random stappen vanaf het begin
- Selectieve eliminatie





# Hillclimbing: selective elimination

- Random oplossing
- Zet random stappen vanaf het begin
- Selectieve eliminatie





# Hillclimbing: selective elimination

- Random oplossing
- Zet random stappen
- Selectieve eliminatie

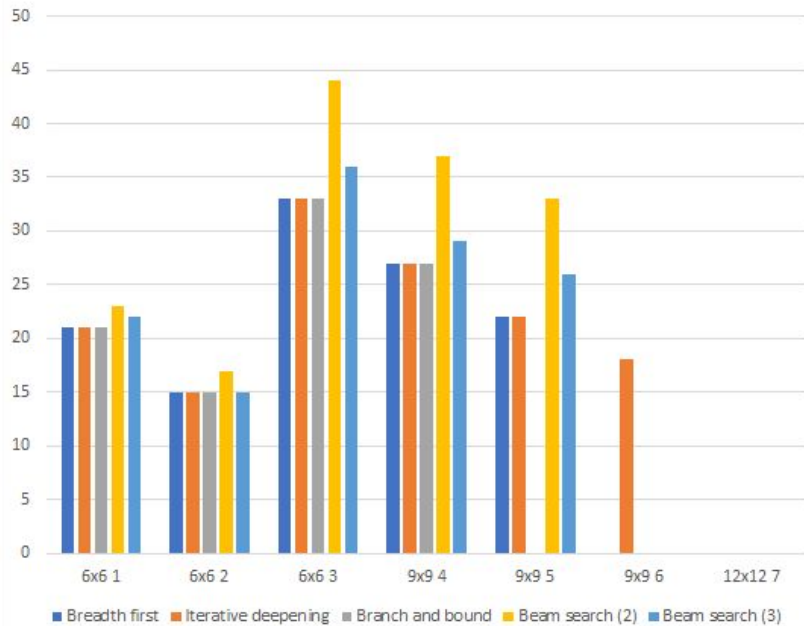




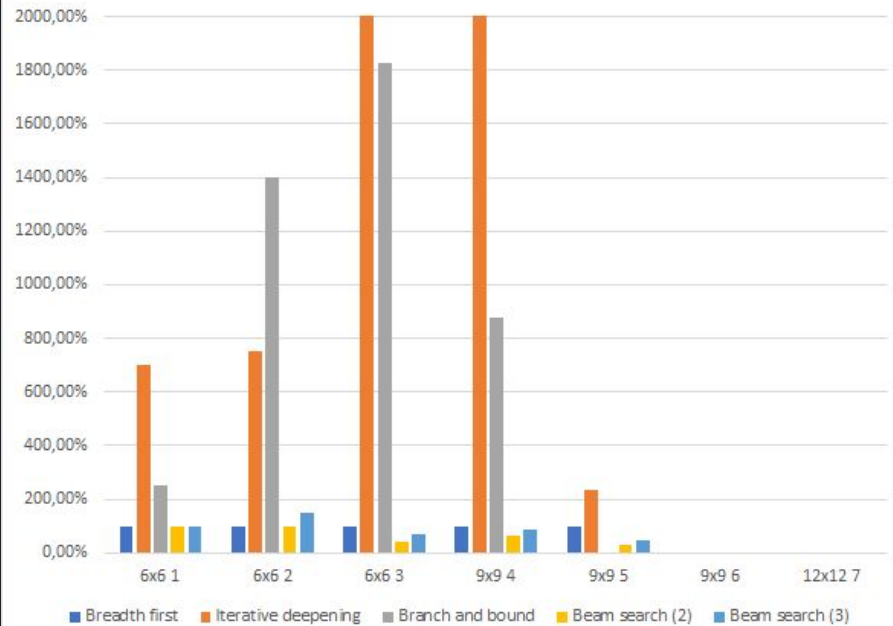
# Resultaten

# Vergelijking algoritmen

Moves nodig tot oplossing

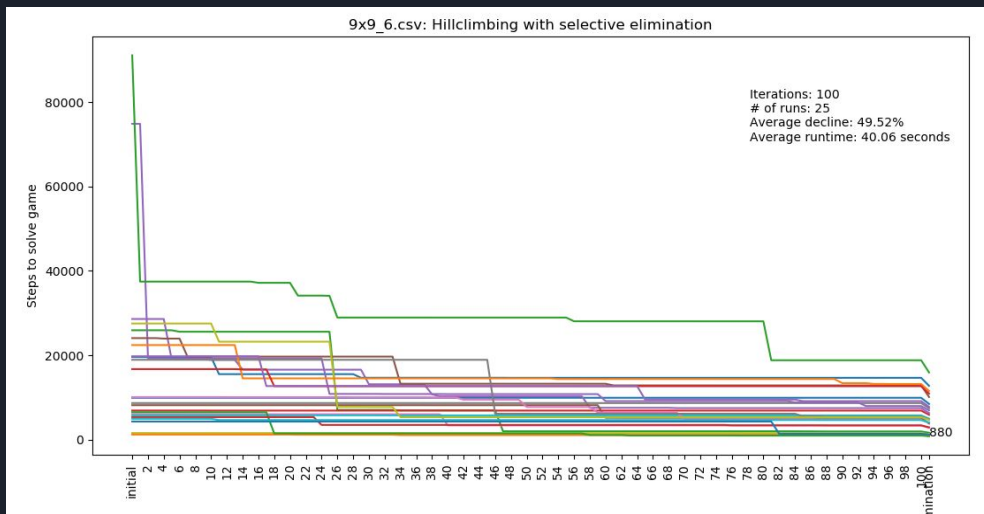
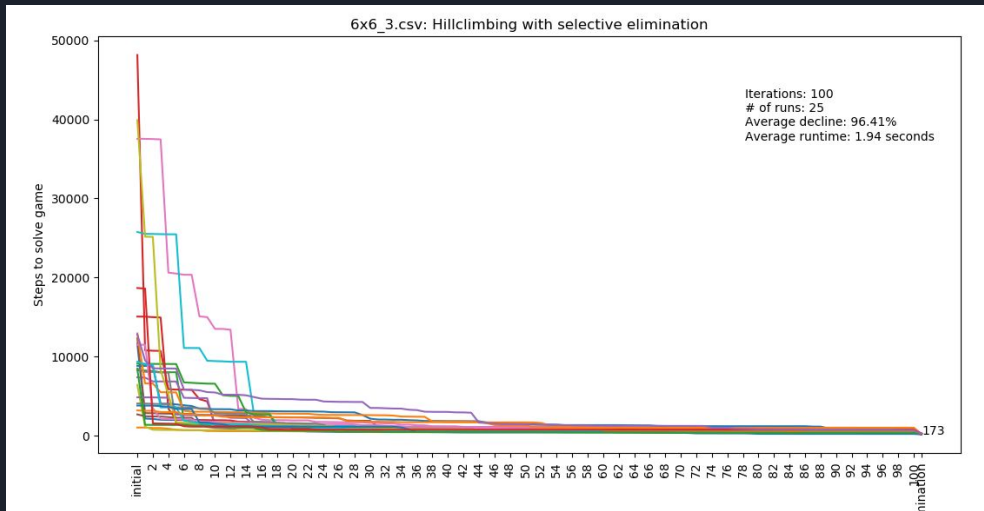


Tijd nodig tot oplossing, relatief aan BFS



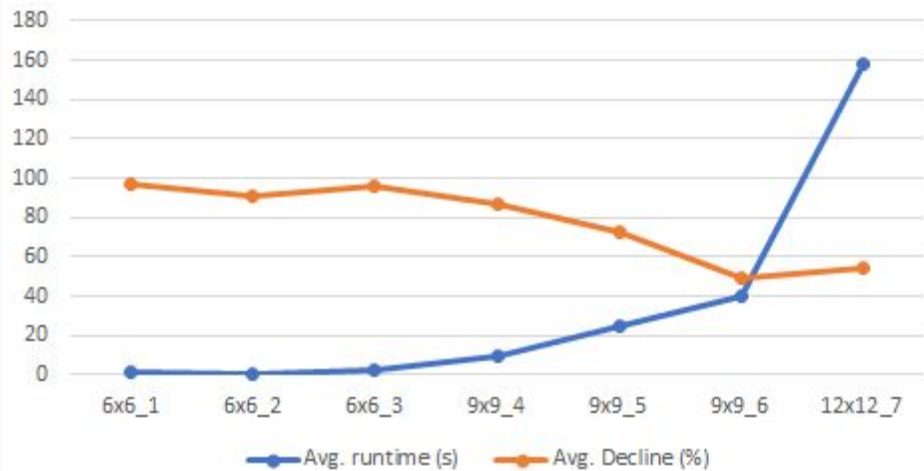
# Hillclimb resultaten

- Grotere borden leiden tot slechtere oplossingen



# Hillclimb resultaten

Runtime and decline



## Board

## difference from best solution

6x6\_1

13

6x6\_2

12

6x6\_3

140

9x9\_4

257

9x9\_5

611

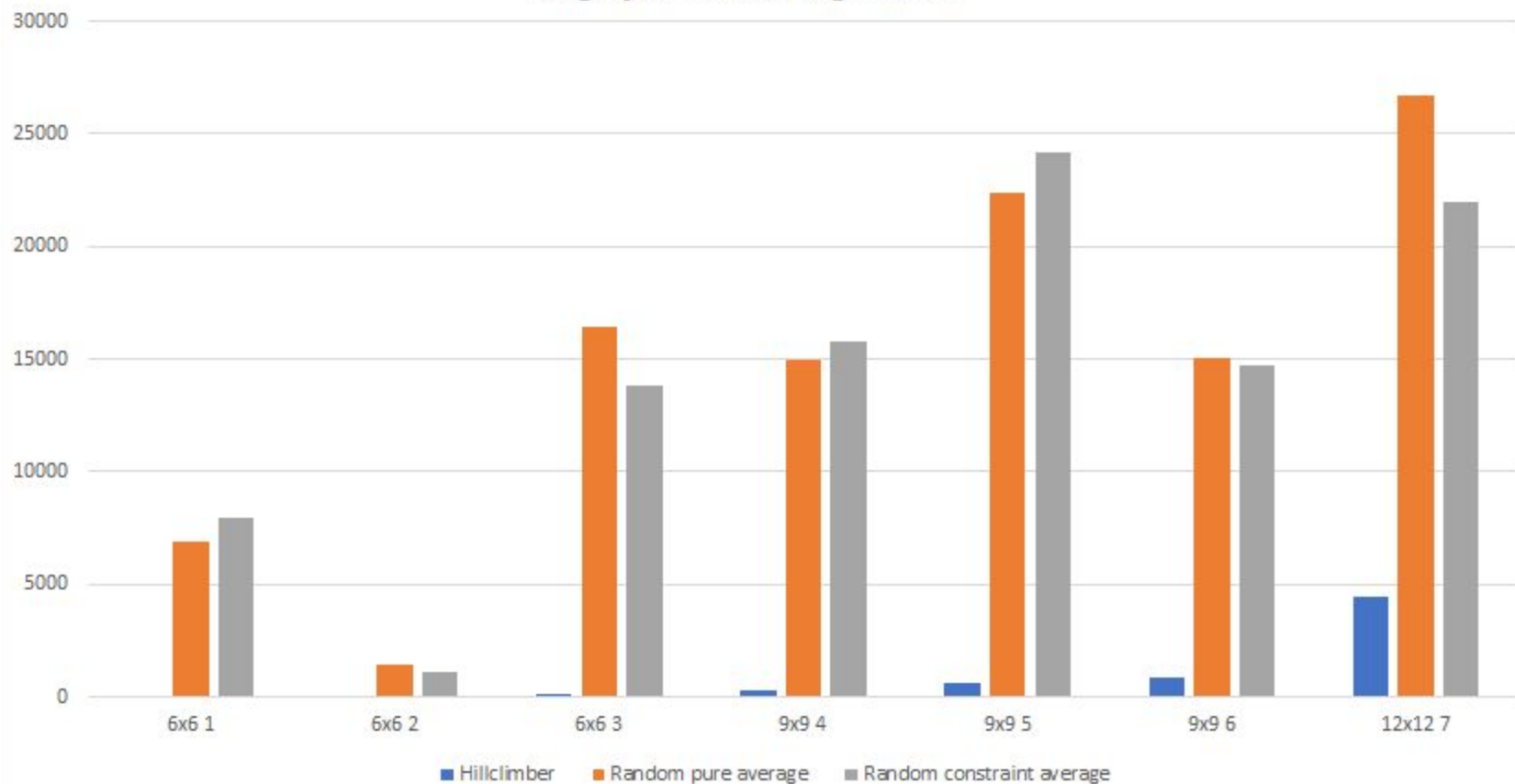
9x9\_6

862

12x12\_7

~4447

## Vergelijken random algoritmen







# Advanced

EASY



HARD



- Obstacle chain



# Computer VS. Human

- Makkelijk bord is moeilijk voor computer
  - Vrije ruimte
  - Computer kijkt naar children (BFS / DFS)
- Mens kijkt naar blokkerende auto's



## Future work

- A\* algoritme
- Geheugenprobleem:
  - Auto's geheugen efficiënt opslaan
- 12x12 borden oplossen met alle algoritmen



# Conclusie

“Wat is de beste manier om een computer Rush Hour op te laten lossen?”

- Hypothese: breadth first search
- Genoeg RAM: BFS
- Genoeg tijd: Iterative deepening/Branch and bound



Vragen?



# Simulated annealing algoritme

- Slechtere resultaten
- Focus op andere algoritmen



- [illegible]

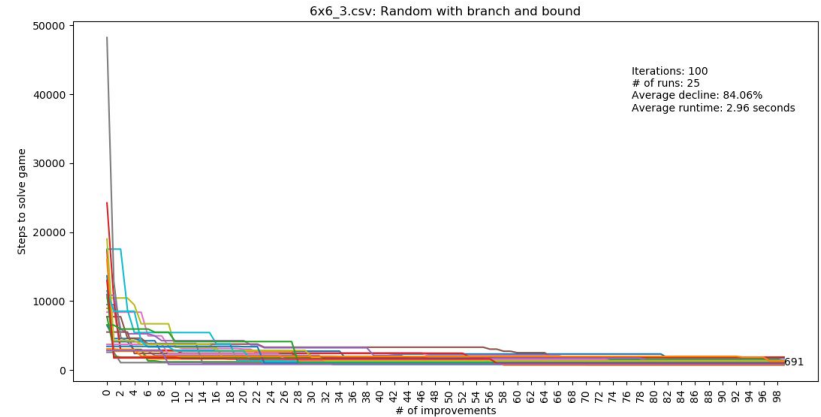
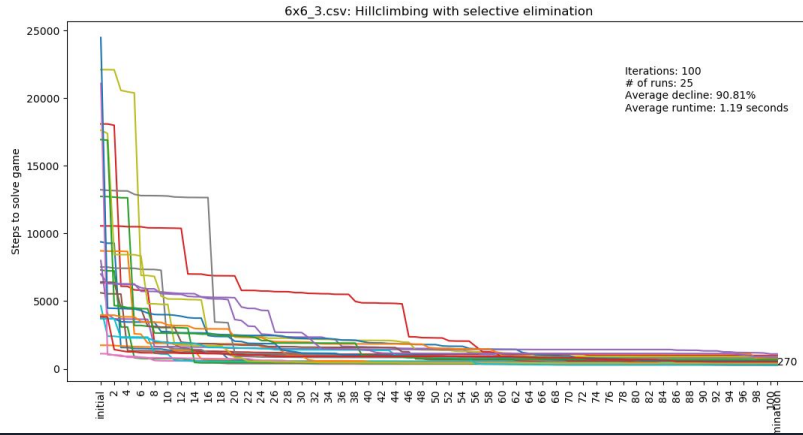


# A\* algoritme

- Lastig om waardes aan moves toe te kennen
- Onduidelijk wat een verbetering is op vorige situatie



# Hillclimb zonder selective elimination



# Hillclimb vs random branch & bound

