Rush Hour

Liz Mooij,
Hiele Wilkes &
Coen Prins

Wat is Rush Hour ?

- *Een 1 persoons puzzel spel uit 1996*
- *Heeft een simpele opzet met schuifbare auto's*
- *Het doel is om de rode auto bij de uitgang te krijgen*

Rush Hour oplossen

- *Het vinden van de oplossing kan erg complex worden*
- *Er is geen uniforme bord configuratie voor een opgeloste puzzel*
- *er is geen duidelijke definitie van wat een bord “moeilijk” maakt*
- *Geen heuristieken die consistent beter presteren dan random*

Onze case

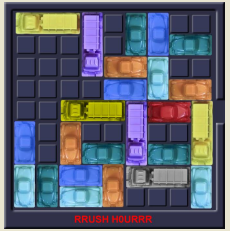
Het oplossen van zeven verstrekte bord configuraties

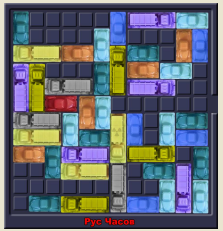
Gebruikte algoritmes

Random + heuristics

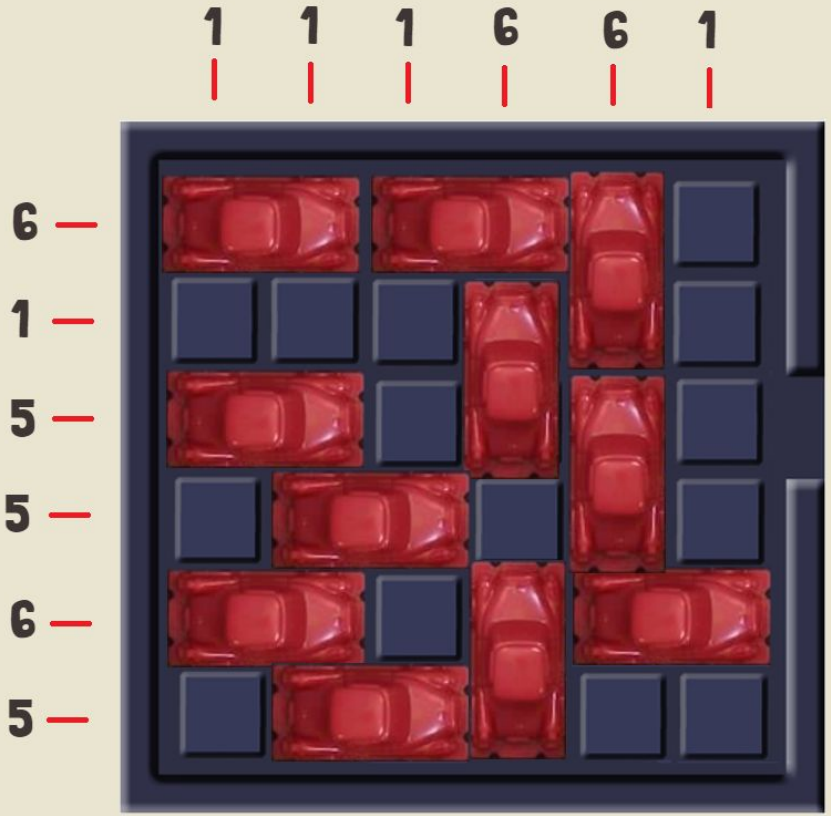
Breadth First search

Backtrack algorithm





State space = $6^4 \times 1^5 \times 5^3 = 162\,000$



State space per game

Game 1 – 6x6: 4 050 000
Game 2 – 6x6: 13 500 000
Game 3 – 6x6: 1 000 000
Game 4 – 9x9: 6 780 000 000 000
Game 5 – 9x9: 126 000 000 000 000 000
Game 6 – 9x9: 516 000 000 000 000 000
Game 7 – 12x12: 239 000 000 000 000 000 000 000 000 000

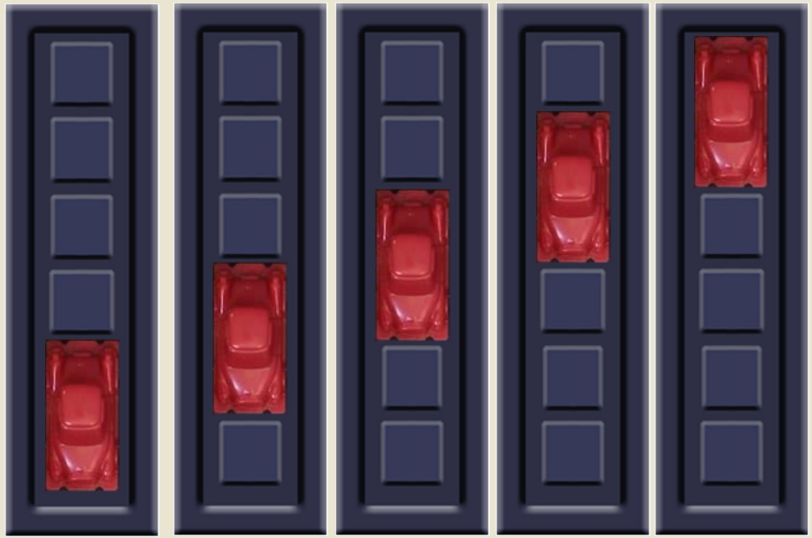
Random solver

- Pick a random car
- Move it a single step
- Repeat until red car is at exit coordinates

Gemiddeld aantal moves per oplossing:
600 (5k samples)

Single step VS max steps

4 moves



1 move

Gemiddeld aantal moves per oplossing:

vorige: 600 nieuwe: 160 (5k samples)

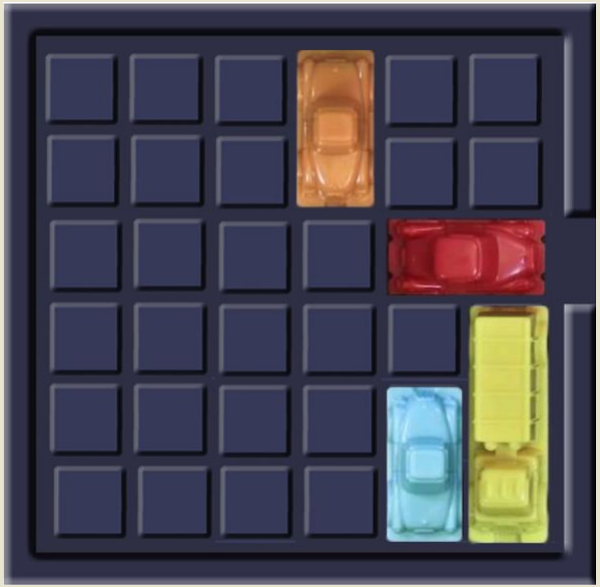
Non recurring algorithm

Gemiddeld aantal moves per oplossing:

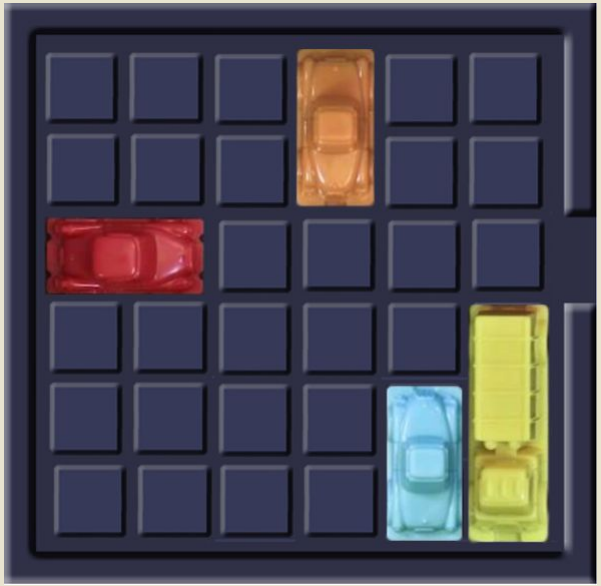
vorige: 160 nieuwe: 140 (5k samples)

Win conditie

win



check path free

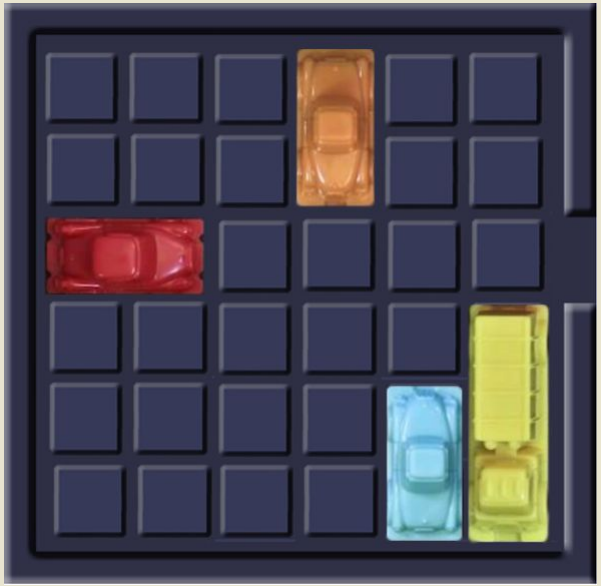


Gemiddeld aantal moves per oplossing:

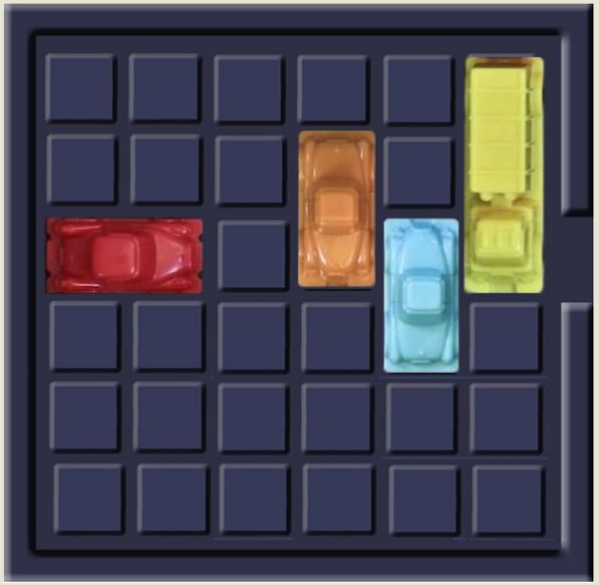
vorige: 140 nieuwe: 65 (5k samples)

Win conditie

"check path free"



"make path free"



Gemiddeld aantal moves per oplossing:

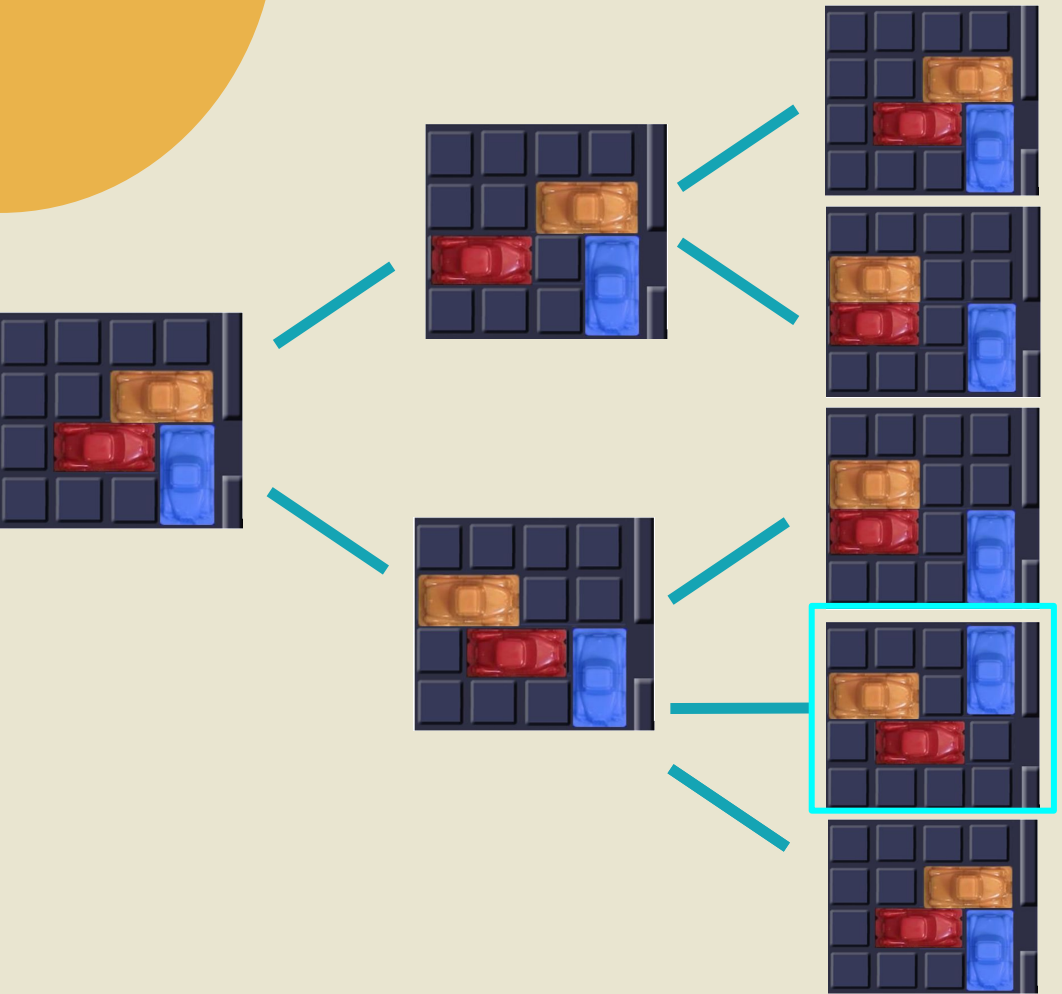
vorige: 65 nieuwe: 17 (5k samples)

Resultaten

step size	non recurring	win condition	game 1	game 2	game 3	game 4	game 5	game 6	game 7
single	no	win	597	2856	80200	80085	166694	49145	x
single	no	check path free	103	1921	68061	72973	162786	42700	x
single	no	make path free	29.5	1178	61333	70418	x	36119	x
max	no	win	163	1014	5185	7725	6535	9222	x
max	no	check path free	77	810	4490	6871	6037	8068	x
max	no	make path free	21.5	598	3537	6117	5761	6978	x
max	yes	win	138	783	3774	5820	5735	7982	x
max	yes	check path free	65	617	3353	5075	5417	6631	20834
max	yes	make path free	17	476	2704	4478	5225	5815	13064

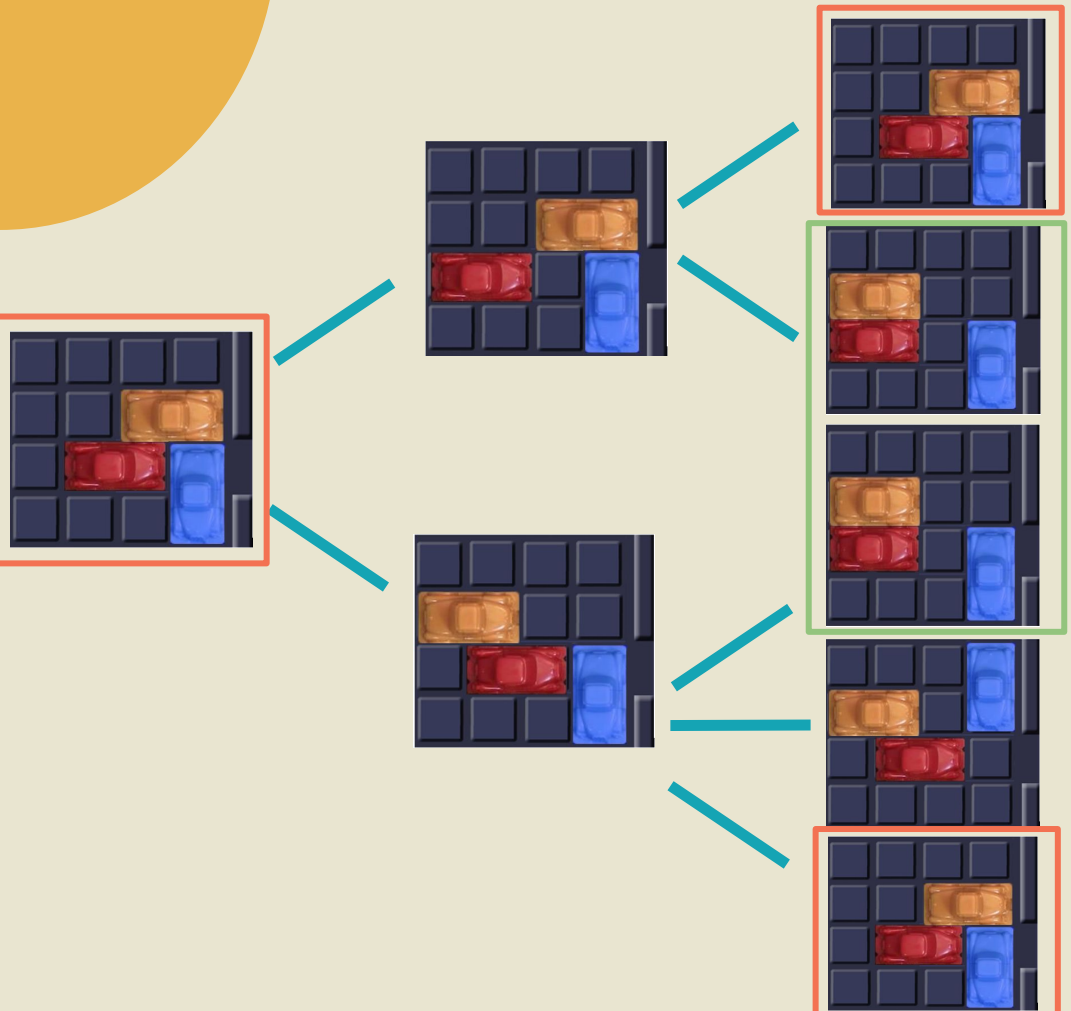
Tabel 1: Resultaten random algoritmes met verschillende heuristieken (per 5000 samples)

Breadth First search



- Creëert een systematische stamboom van alle mogelijke moves
- Vindt altijd de kortste oplossing
- Erg intensief in zowel betreffende geheugen als rekentijd

Breadth First search



- Het is essentieel om via prunen duplicates te verwijderen
- Pre Pruning checkt voordat een node aangemaakt wordt
- Post Pruning checkt pas voordat de afstammelingen worden aangemaakt

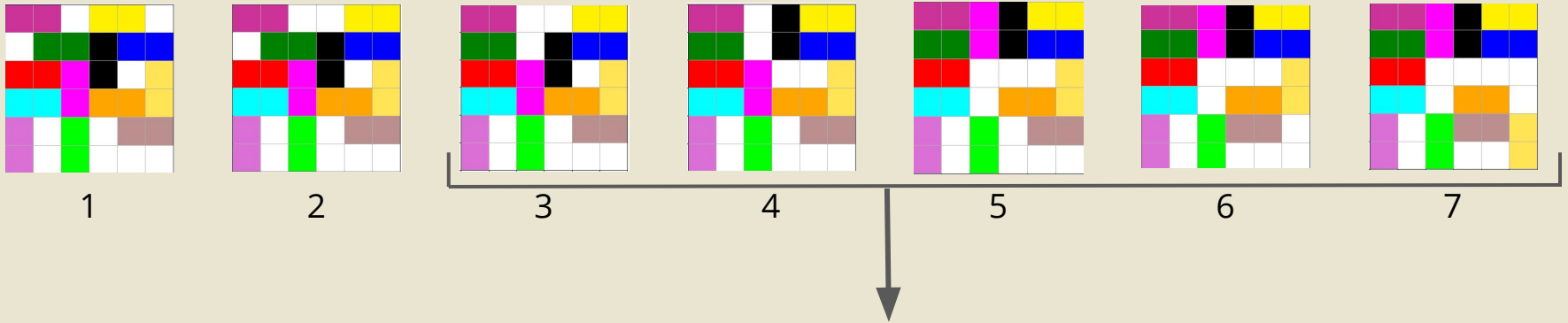
Results Breadth first

	Single Step	Max Step	Time Max-step
Game 1	**9**	**8**	0.05 s
Game 2	**27**	**15**	0.23 s
Game 3	**80**	**33**	7.69 s
Game 4	**45**	**27**	1h 10m
Game 5	*Geen oplossing*	*Geen oplossing*	*x*
Game 6	*Geen oplossing*	*Geen oplossing*	*x*
Game 7	*Geen oplossing*	*Geen oplossing*	*x*

Tabel 2: Resultaten Breadth First met single step en max step

Back Track algorithm

Stap 1: *Grids toevoegen aan dictionary*

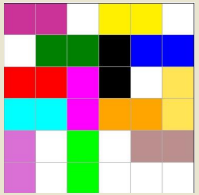


Voorbeeld: Game met 7 stappen
→ Laatste 5 stappen worden opgeslagen

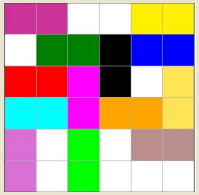
Key: Grid
Value(s): Stap(pen) tot de eindgrid

Back Track algorithm

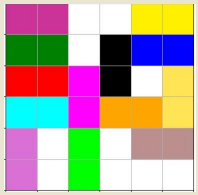
Stap 2: *Random met dictionary check*



1



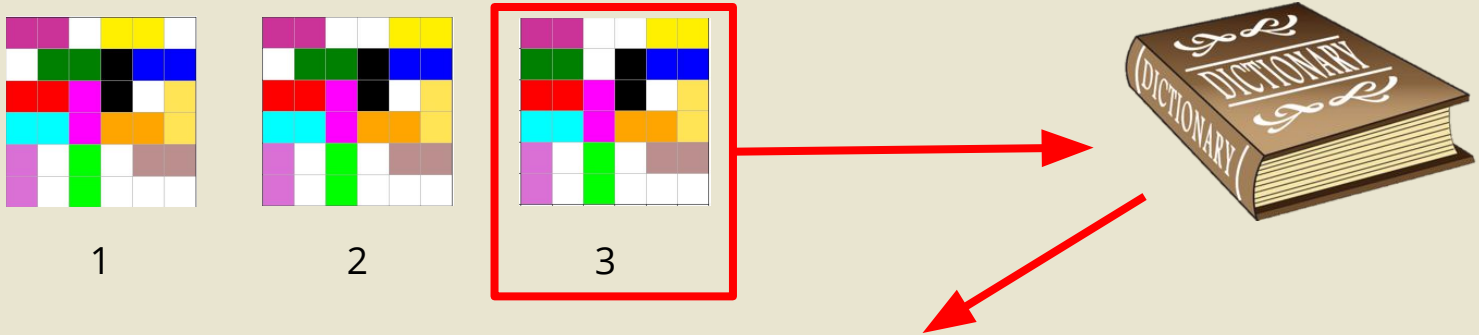
2



3

Back Track algorithm

Stap 2: *Random met dictionary check*



Resultaten Back Track

	Back Track	Beste random resultaat
Game 1	**8**	17
Game 2	**33**	476
Game 3	**411**	2704
Game 4	**3652**	4478
Game 5	**4522**	5225
Game 6	5920	**5815**
Game 7	*Geen oplossing*	13064

Tabel 3: Resultaten gemiddelde Back Track vs random per 5000 metingen

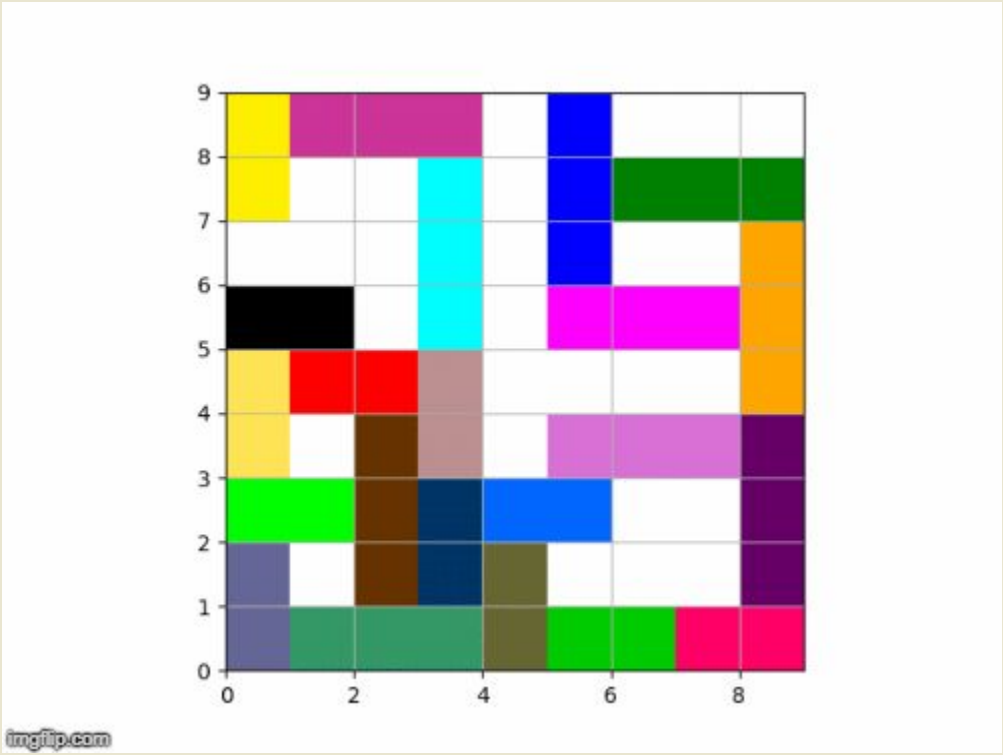
Conclusie

- Er is niet een algoritme goed voor alle game boards
- Breadth First en Back Track in hun huidige vorm zijn ongeschikt voor complexere borden

Game 1	Breadth First Search (max step) / Back Track Algorithm
Game 2	Breadth First Search (max step)
Game 3	Breadth First Search (max step)
Game 4	Breadth First Search (max step)
Game 5	Back Track Algorithm
Game 6	Random - max step - non recurring - make path free
Game 7	Random - max step - non recurring - make path free

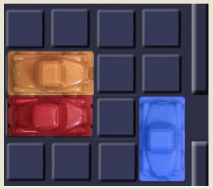
Tabel 4 : Beste algoritme per game

Vragen?



Breadth First search

- Twee verschillende win condities
  - path free
  - Single blocker

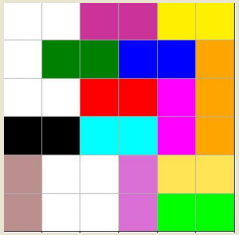


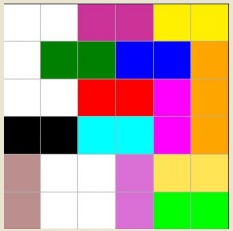
- Compromis tussen rekentijd per node en totale hoeveelheid nodes

Breadth First search

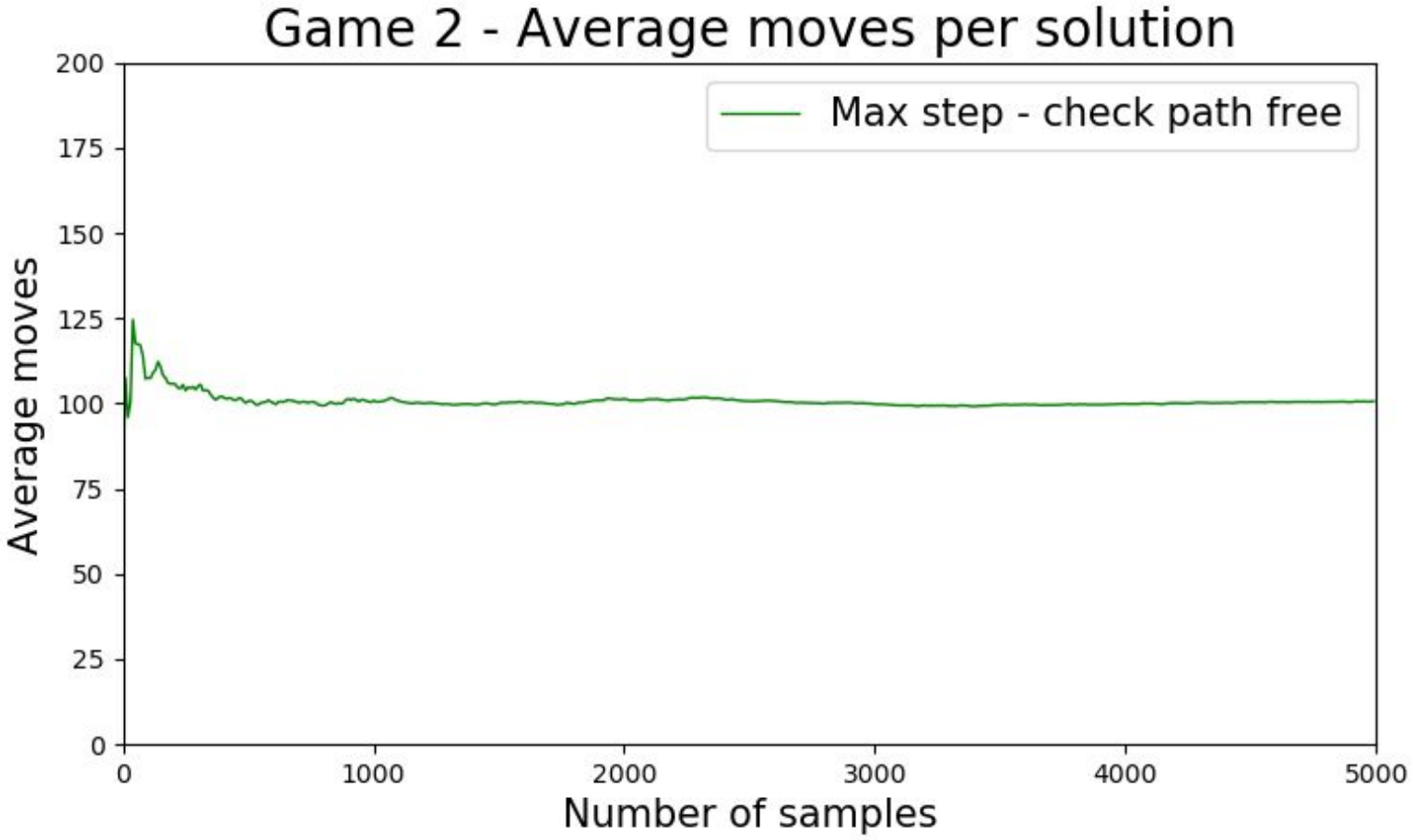
- Twee verschillende move definities

Single Step & Max Step





- Voor max step zijn minder moves nodig, maar het test niet alle bord-configuraties


Game 2 - Average moves per solution
Max step - check path free
Average moves
Number of samples
200
175
150
125
100
75
50
25
0
0
1000
2000
3000
4000
5000

All possible combinations of algorithms

step size	non recurring	win condition	game 1	game 2	game 3	game 4	game 5	game 6	game 7
single	no	win	597	2856	80200	80085	166694	49145	x
single	no	check path free	103	1921	68061	72973	162786	42700	x
single	no	make path free	29.5	1178	61333	70418	x	36119	x
max	no	win	163	1014	5185	7725	6535	9222	x
max	no	check path free	77	810	4490	6871	6037	8068	x
max	no	make path free	21.5	598	3537	6117	5761	6978	x
max	yes	win	138	783	3774	5820	5735	7982	x
max	yes	check path free	65	617	3353	5075	5417	6631	20834
max	yes	make path free	17	476	2704	4478	5225	5815	13064

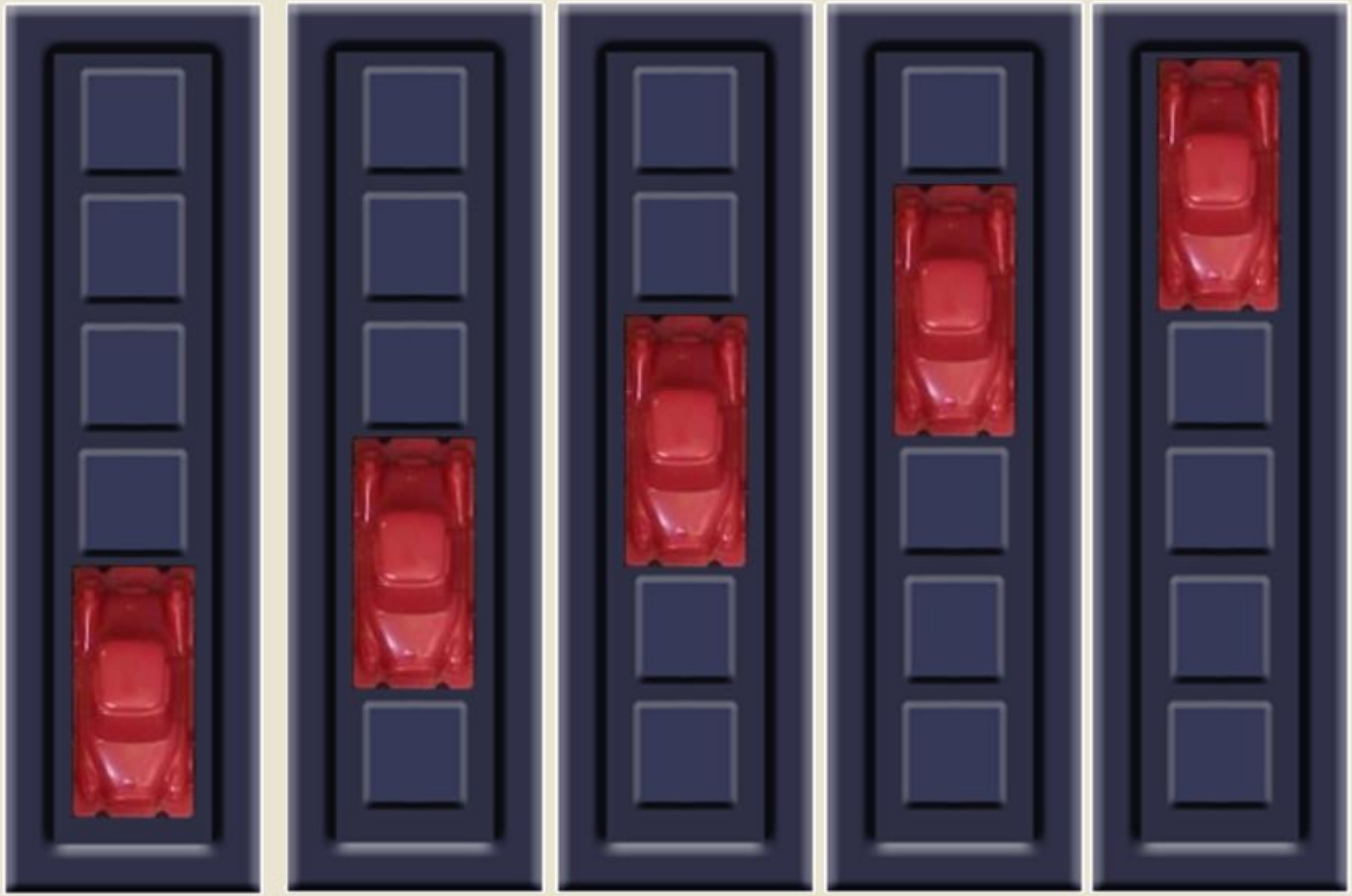
All possible comparisons of algorithms

step size	non recurring	win condition	game 1	game 2	game 3	game 4	game 5	game 6
single	no	win	597	2856	80200	80085	166694	49145
single	no	check path free	103	1921	68061	72973	162786	42700
single	no	make path free	29.5	1178	61333	70418	X	36119
max	no	win	163	1014	5185	7725	6535	9222
max	no	check path free	77	810	4490	6871	6037	8068
max	no	make path free	21.5	598	3537	6117	5761	6978
single	no	win	597	2856	80200	80085	166694	49145
max	no	win	163	1014	5185	7725	6535	9222
single	no	check path free	103	1921	68061	72973	162786	42700
max	no	check path free	77	810	4490	6871	6037	8068
single	no	make path free	29.5	1178	61333	70418	163435	36119
max	no	make path free	21.5	598	3537	6117	5761	6978
max	no	win	163	1014	5185	7725	6535	9222
max	yes	win	138	783	3774	5820	5735	7982
max	no	check path free	77	810	4490	6871	6037	8068
max	yes	check path free	65	617	3353	5075	5417	6631
max	no	make path free	21.5	598	3537	6117	5761	6978
max	yes	make path free	17	476	2704	4478	5225	5815

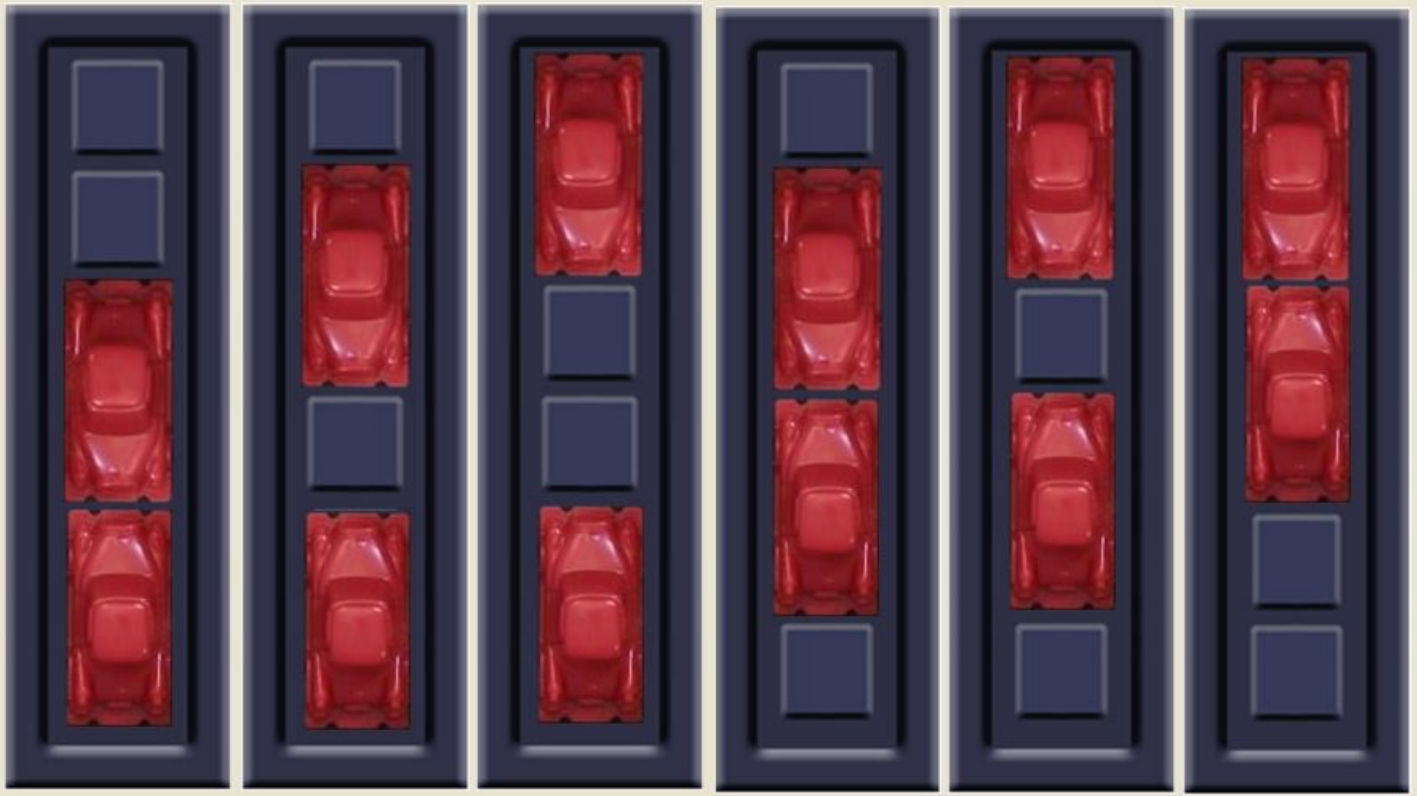
Back Track algorithm

Conditions	Average Back Track	Best average random
Game 1 - 5000x dict vullen met 8 stappen - 5000x random	8	17
Game 2 - 5000x dict vullen met 14 stappen - 5000x random	33	476
Game 3 - 5000x dict vullen met 33 stappen - 5000x random	411	2704
Game 4 - 5000x dict vullen met 27 stappen - 5000x random	3652	4478
Game 5 - 5000x dict vullen met 30 stappen - 5000x random	4522	5225
Game 6 - 5000x dict vullen met 30 stappen - 5000x random	5920	5815

State space = 5



State space = 6



State space new = 6 * 6 = 36