

Codecup 2014: Poly-Y

Lesley Wevers & Steven te Brinke

Wie zijn wij?

Lesley

- Saxion Informatica
- MSc Dependable and Secure Computing
- Promotie FMT

- Norvig Award

Steven

- BSc Informatica
- MSc Software Engineering
- Promotie SE/FMT

- Informatica Olympiade
- Twents Kampioenschap Programmeren

Studie informatica

- Programmeren/Algoritmiek
- Wiskunde/Abstractie
- Software ontwerp
- Databases/Security/HMI/Netwerken/...
- Onderzoek

Onderzoek



“If I have seen further, it is by standing on the shoulders of giants.”

Isaac Newton

1676

110101001001010110110011101011010101101

CODECUP

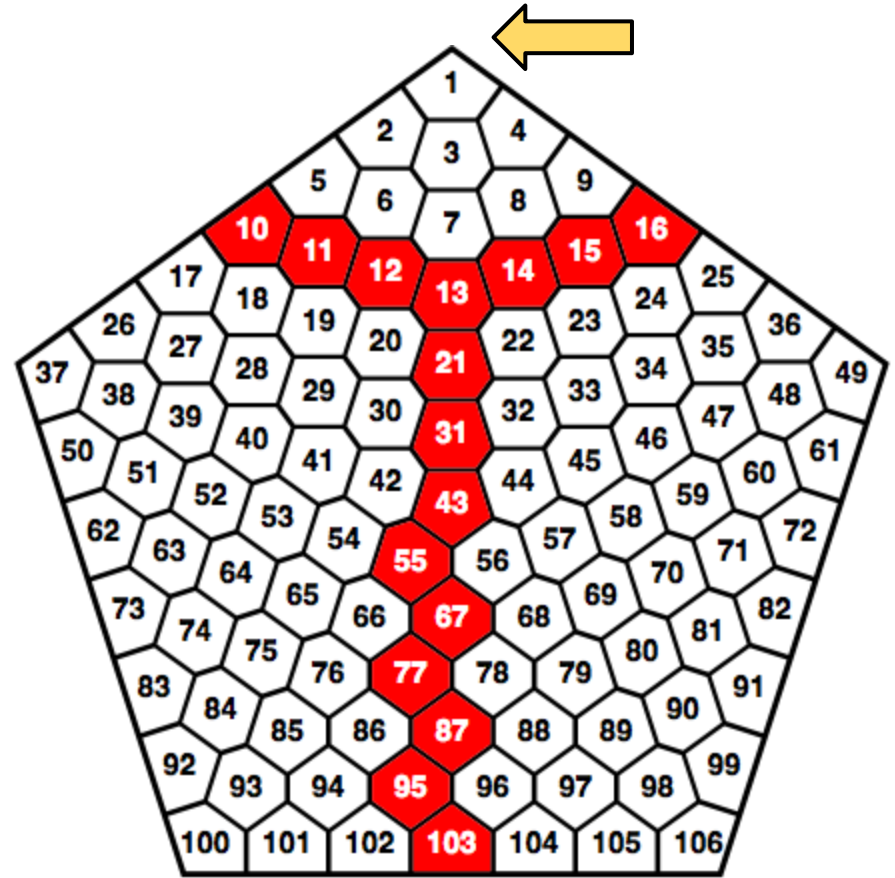
1010110011101 **2014** 01001101101011

Rang	Naam	Land	Taal	Punten
1	Lesley Wevers & Steven te Brinke	Nederland	Java	92
2	Abdessamad Elkasimi	Marokko	C++	89
	Sergey Povalikhin	Rusland	C++	89

Poly-Y

Generalisatie van Y

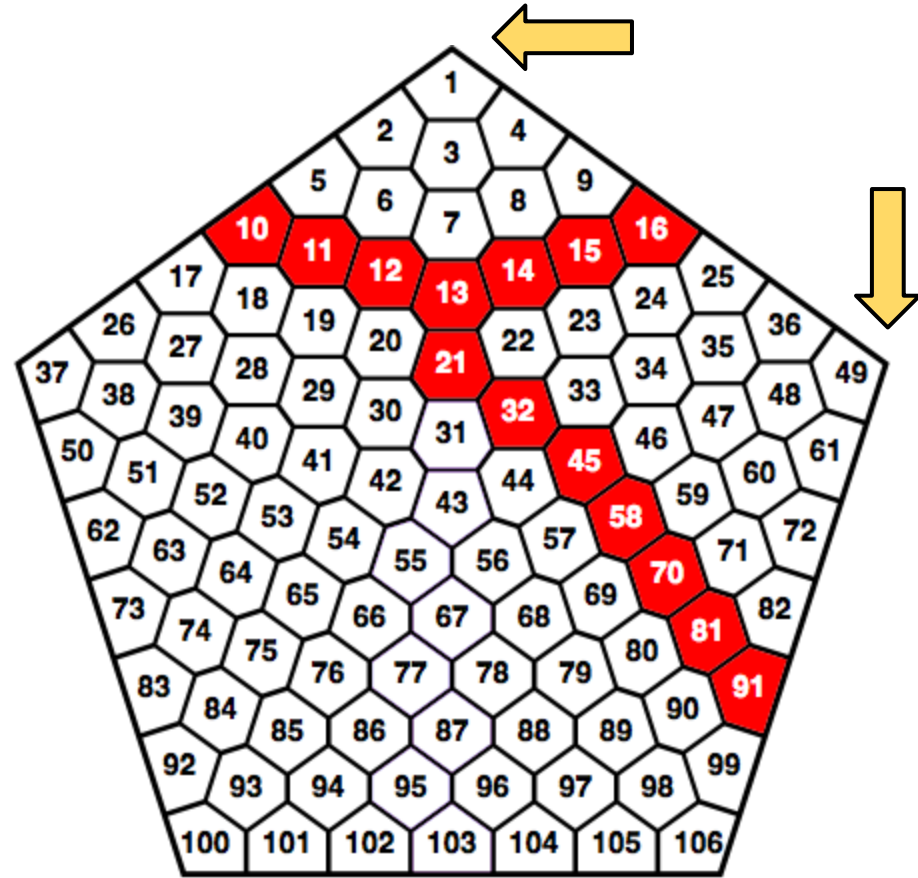
- Spelers zetten om de beurt
- Een Y verovert een hoek
- Drie hoeken is winst
- Swap regel



Poly-Y

Generalisatie van Y

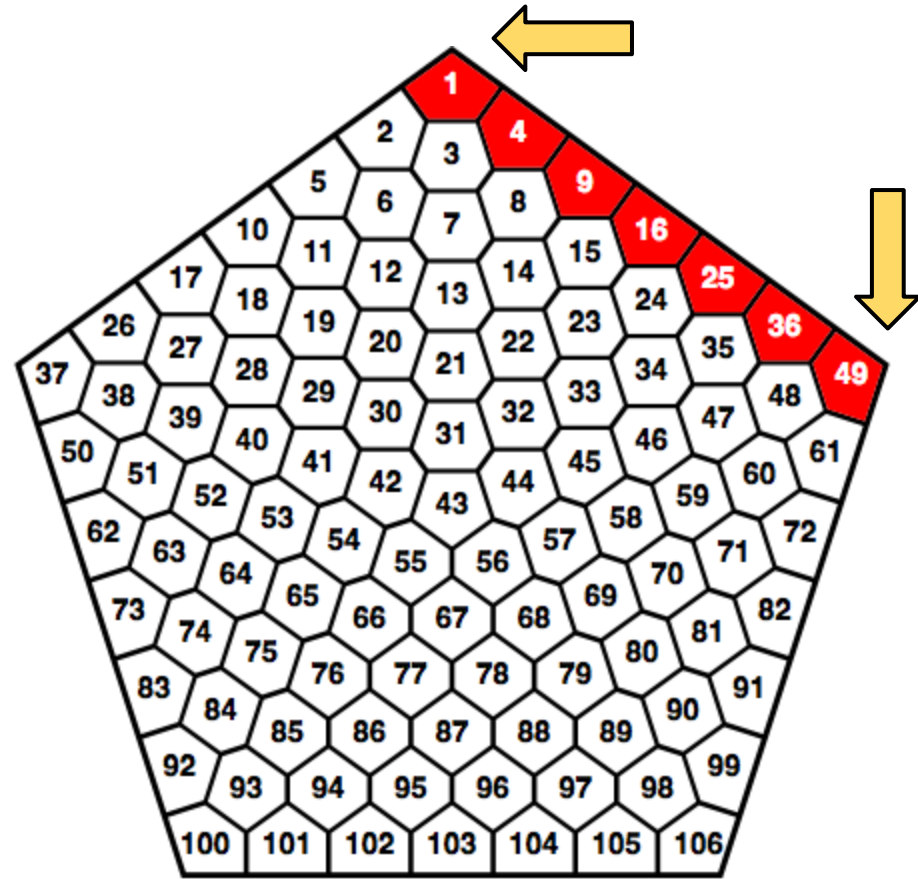
- Spelers zetten om de beurt
- Een Y verovert een hoek
- Drie hoeken is winst
- Swap regel



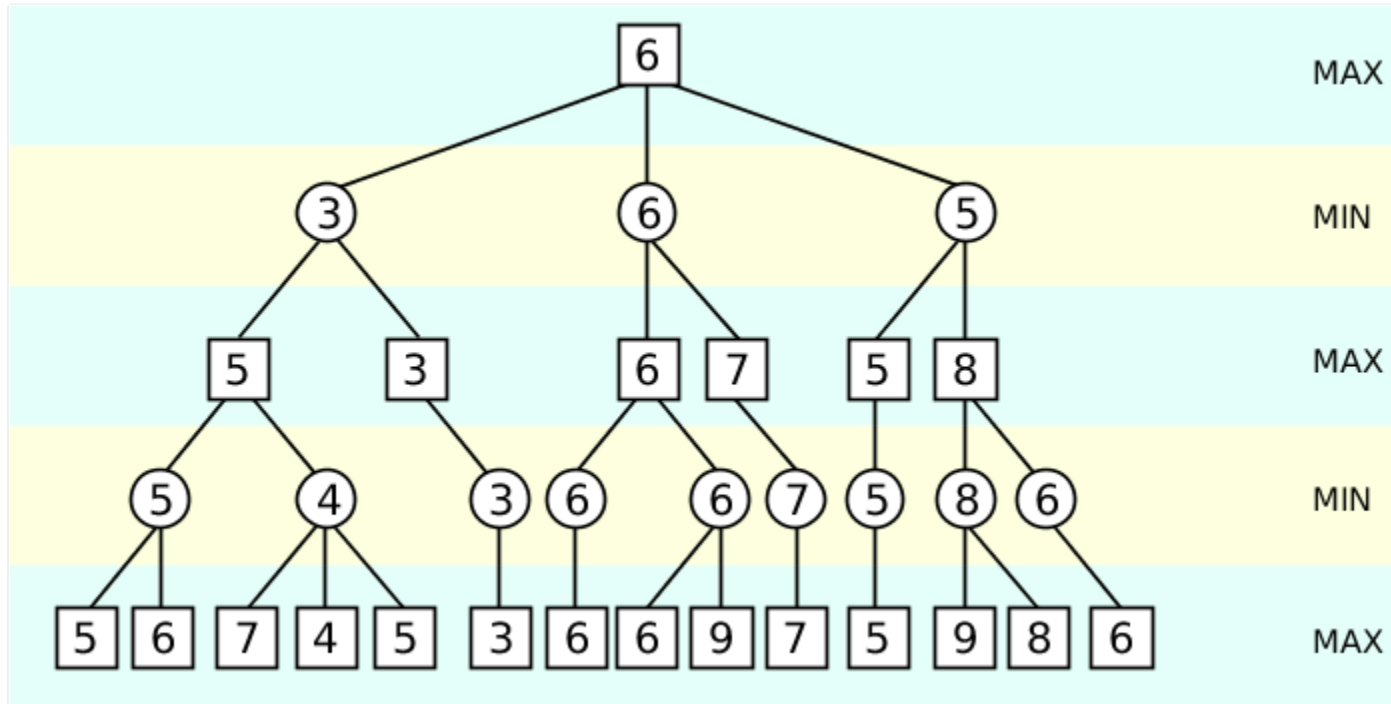
Poly-Y

Generalisatie van Y

- Spelers zetten om de beurt
- Een Y verovert een hoek
- Drie hoeken is winst
- Swap regel



Minimax



Poly-Y

Uitdagingen

- Hoge branching factor:
Minimax kan maar ~4
zetten diep kijken.
- Sterkte van een toestand
is moeilijk te bepalen.

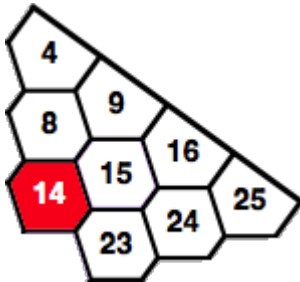
Zet	Toestanden
1	1
2	106
3	11.130
4	1.157.520
5	119.224.560
6	12.160.905.120

Poly-Y - Virtuele Connecties

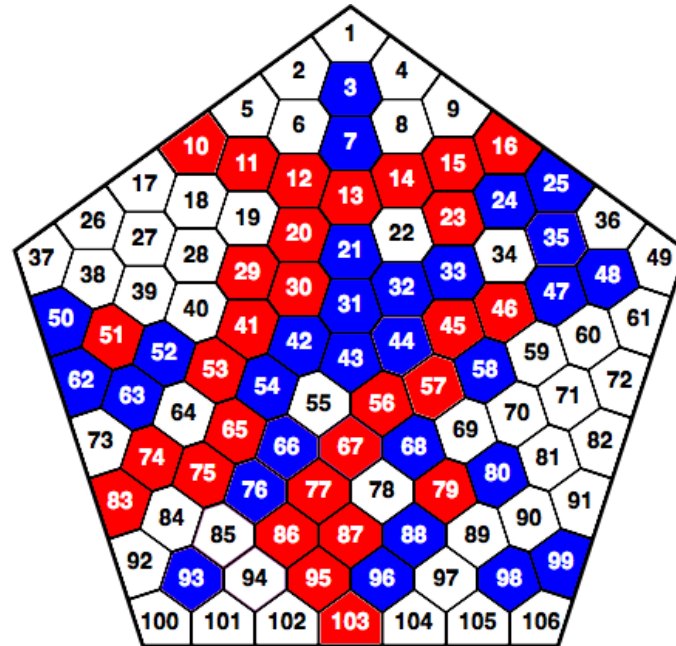
Brug:



Zijde:



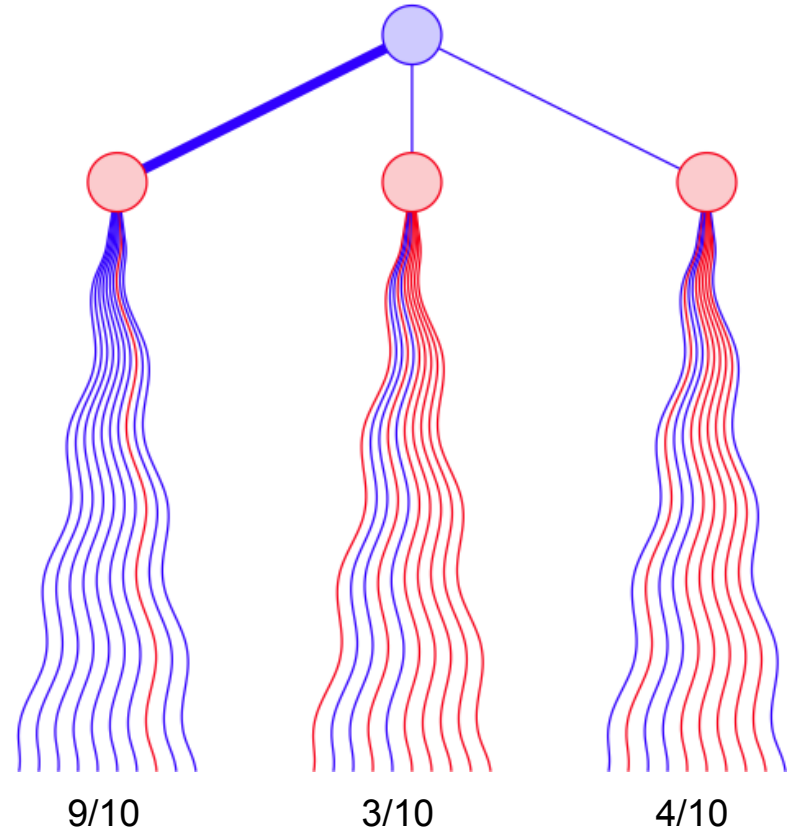
Complex voorbeeld:



Pure Monte Carlo Search

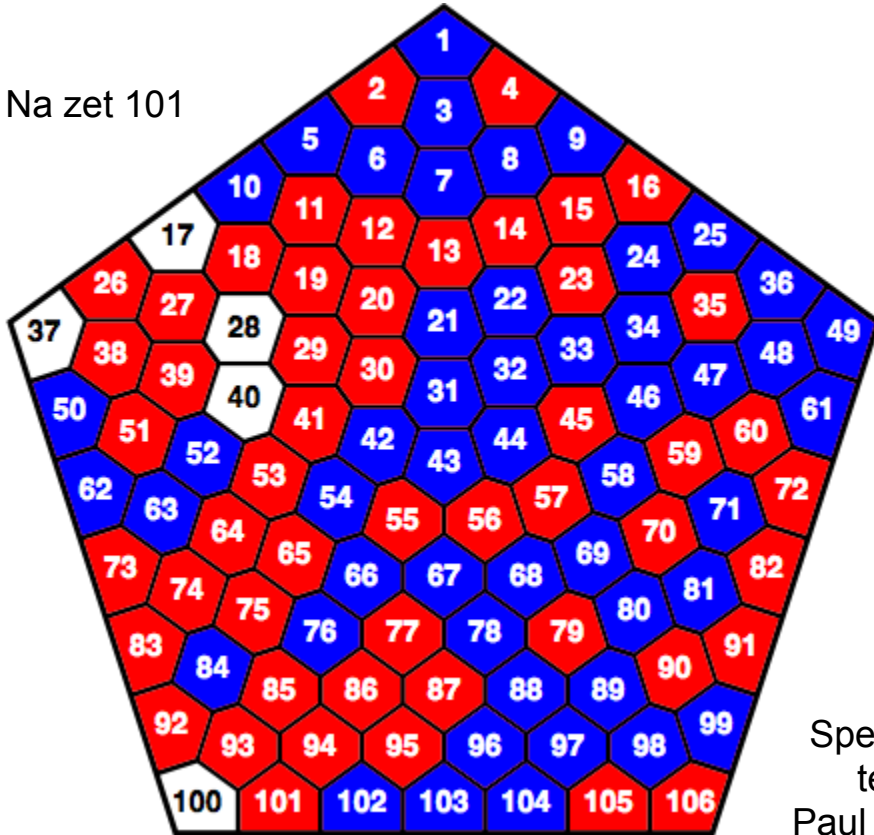
Willekeurige potjes in self-play
Kies zet met de hoogste winkans

Speelt niet optimaal, maar best sterk.

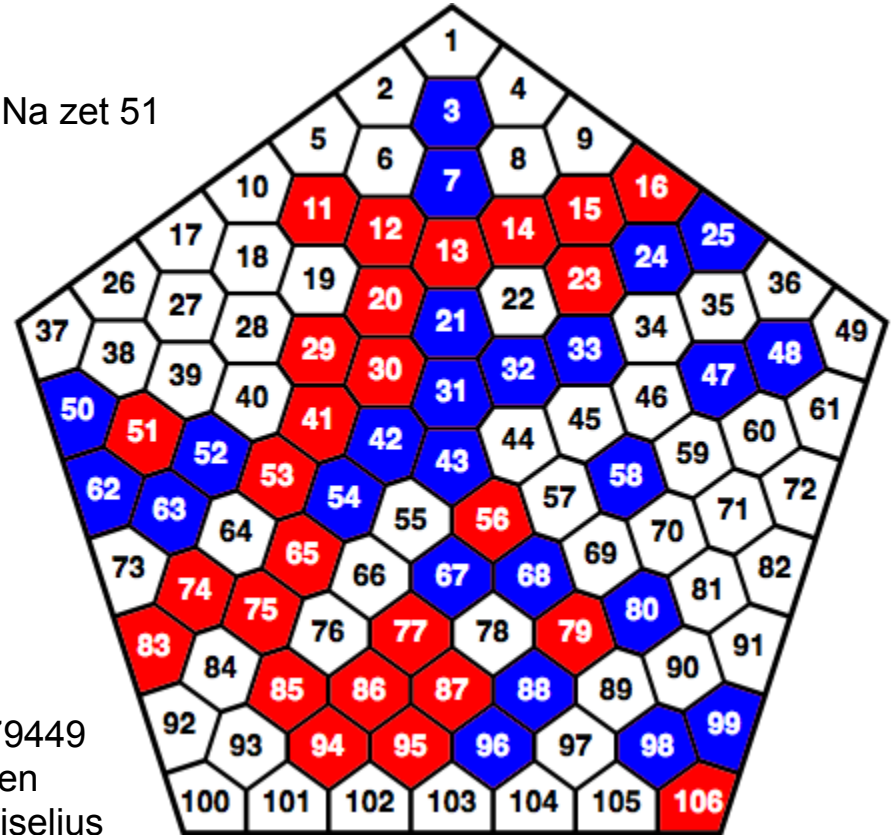


Monte Carlo Methode

Na zet 101



Na zet 51

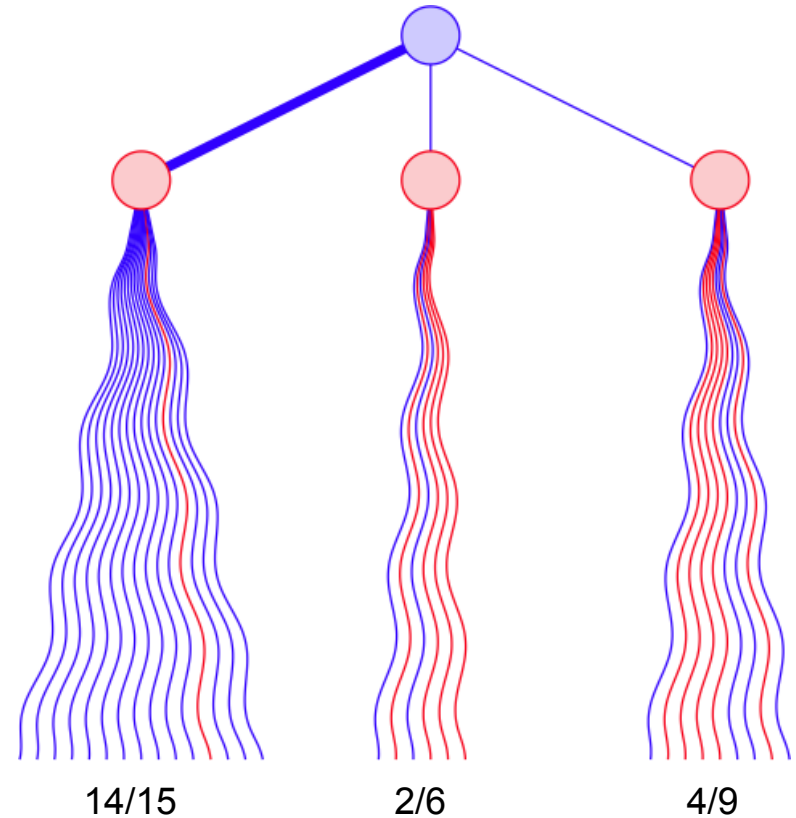
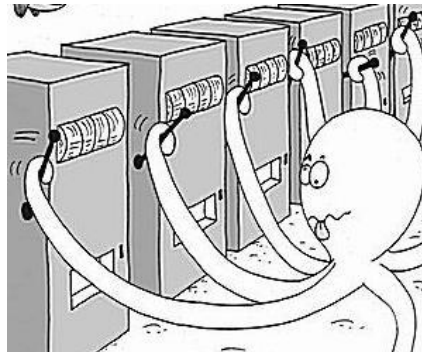


Spel 79449
tegen
Paul Wiselius

Monte Carlo - Optimalisatie

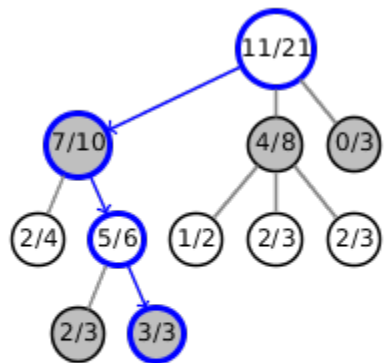
Idee: Besteed niet te veel tijd aan slechte zetten.

Multi-armed bandit probleem:
Zoek balans tussen
exploratie en
exploitatie

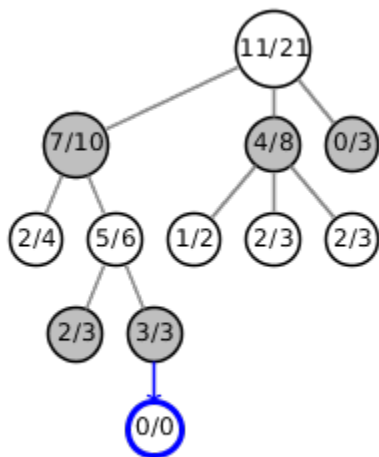


Monte Carlo Tree Search

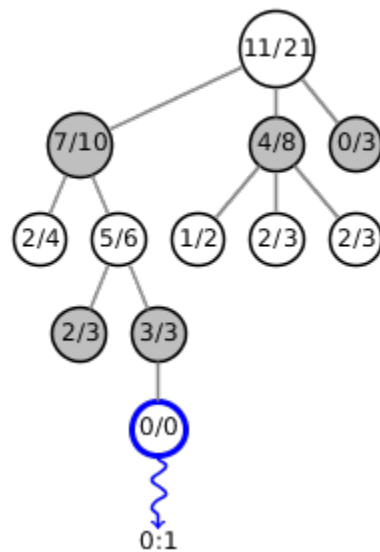
Selectie



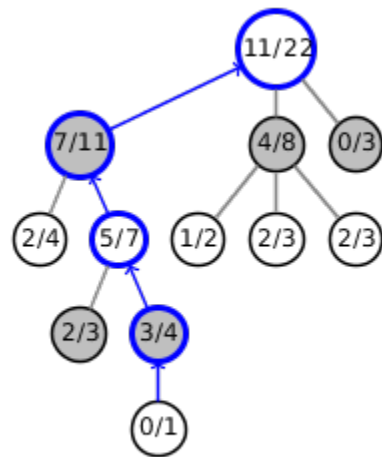
Expansie



Simulatie



Propagatie



Monte Carlo Tree Search

- Exploratie in MTCS is duur: We moeten elke mogelijke zet in een node onderzoeken voordat we dieper kunnen kijken.

100 zetten x 100 simulaties -> 10.000 simulaties per node

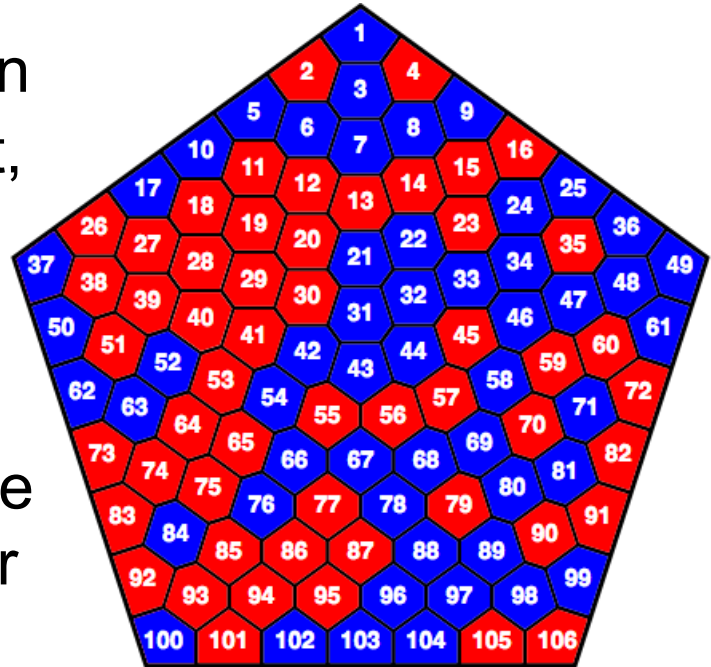
We doen ~100.000 simulaties per seconde

In de praktijk kunnen we maar ~10 nodes per seconde bekijken.

All Moves As First

AMAF: de volgorde van zetten in een willekeurig spel maakt niet uit, elke zet zou als eerste gedaan kunnen zijn.

We kunnen alle zetten in een node tegelijk onderzoeken: ~50x sneller



Heuristieken

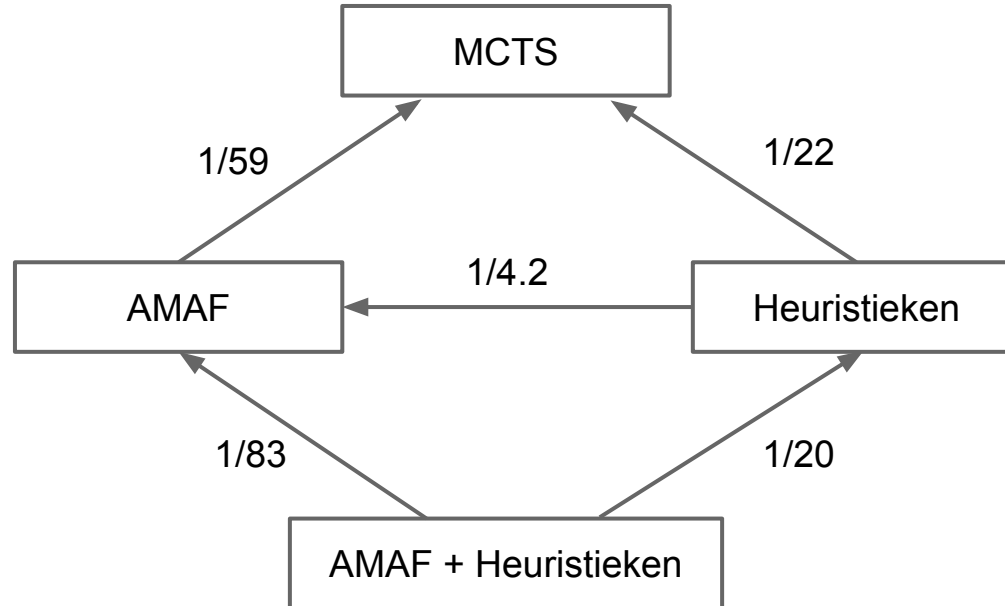
Geheel willekeurige monte carlo simulaties zijn geen goede representatie van een echt spel.

We voegen heuristieken toe aan de simulatie:

- Verdedig bruggen altijd.
- Probeer langs tegenstander naar de rand te lopen.
- De rand van het bord als laatste vullen.

Het is niet makkelijk om goede heuristieken te vinden.

Verbeteringen



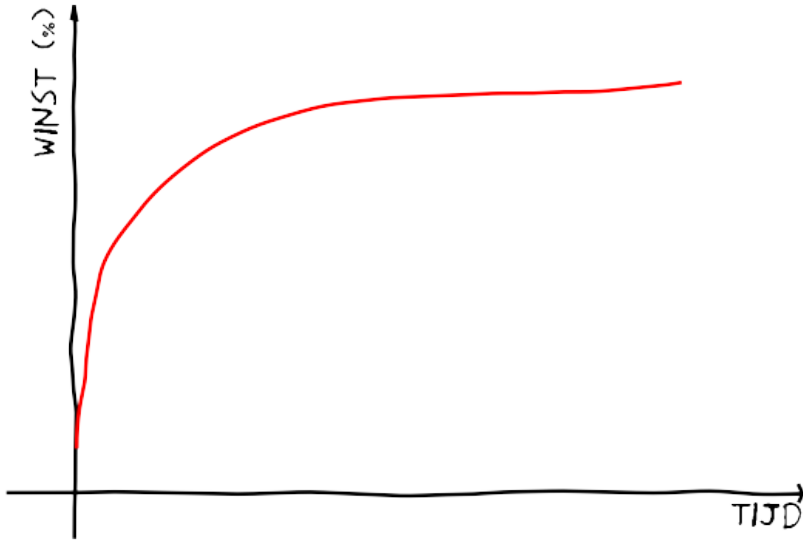
Aantal keer verlies, 5 seconden per speler, 512 samples

Self-Play

Samples	8 core	Win rate - Run 1	Win rate - Run 2	Win rate - Run 3
16	20 s	~ 37%	~ 63%	~ 43%
32	40 s	~ 34%	~ 41%	~ 46%
64	80 s	~ 31%	~ 38%	~ 44%
128	160 s	~ 39%	~ 45%	~ 50%
256	320 s	~ 42%	~ 47%	~ 52%
512	640 s	~ 46%	~ 48%	~ 52%

Win rate in self-play voor 5 seconden per speler.

Performance



Rekentijd	vs. 20s	
1,25s	9.4%	Onze bot in een geïntepreterde taal
2,5s	16.8%	Abdessamad's bot
5s	24.5%	
10s	40.2%	
20s	50%	Onze bot
40s	60%	C++ versie van onze bot
80s	72%	
160s	80%	
320s	83.6%	

Openingsboek

Opening	MAX	MIN	MAX	MIN
9	40.8%	24.6%	-	-
1	39.4%	27.3%	-	-
4	38.8%	30.0%	-	-
16	40.2%	30.0%	-	-
8	45.4%	33.6%	-	-
3	46.2%	33.6%	-	-
7	59.4%	46.5%	60.5%	-
14	58.0%	53.1%	64.8%	-
43	66.5%	49.6%	66.8%	-
15	49.6%	37.9%	67.6%	39.0%
23	63.7%	61.3%	67.6%	-
13	65.0%	48.8%	69.1%	-
22	64.9%	59.7%	78.5%	-
21	67.2%	64.8%	80.0%	-
31	79.4%	76.6%	81.3%	-
32	76.7%	61.3%	81.6%	-

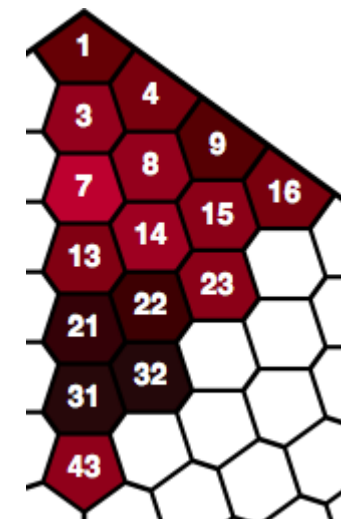
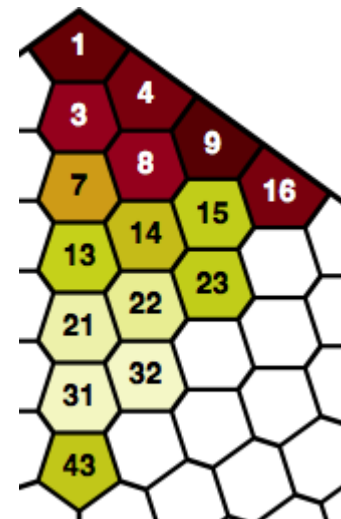
Opmerkingen

Abdessamad

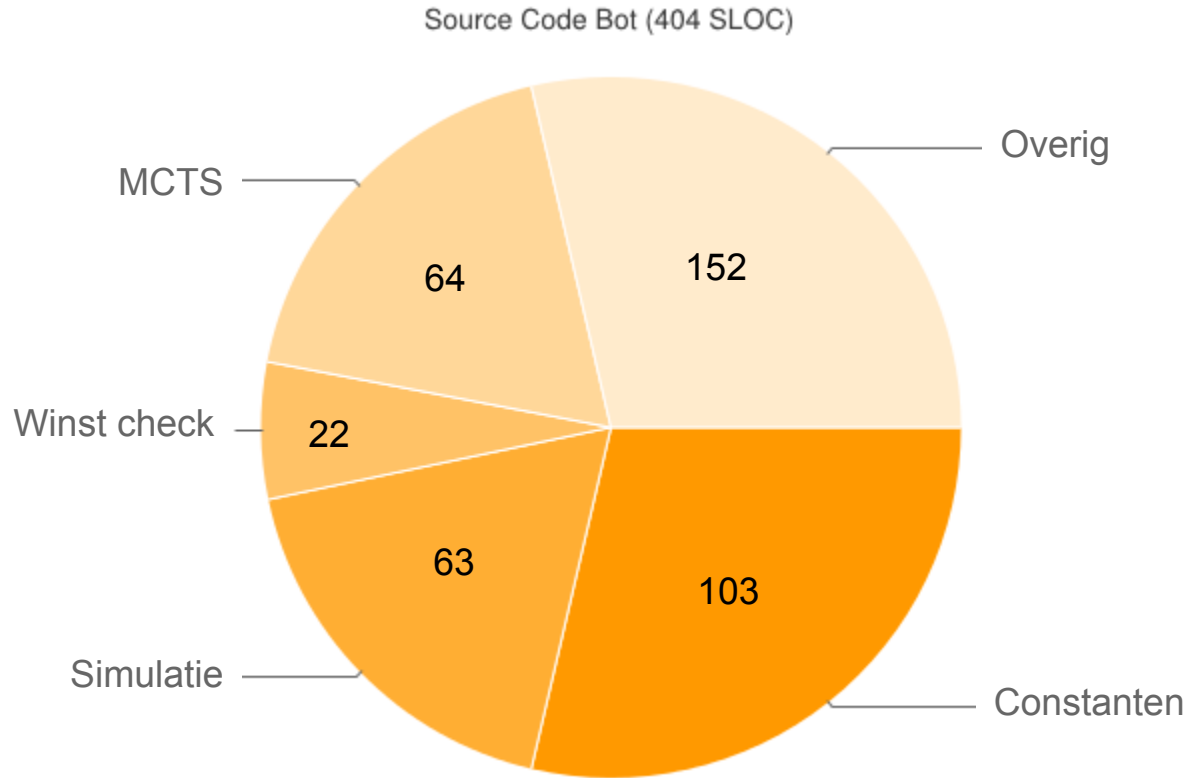
Sergey

Wij, final competitie

Wij, test competitie



Source Code



Poly-Y

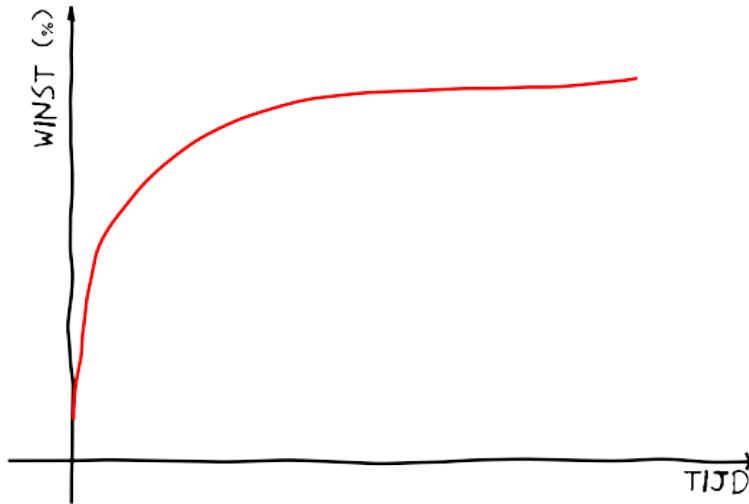
Onze bot:

<https://github.com/lwevers/lynx>

Speel zelf:

<http://elysium.student.utwente.nl/~maks/poly-y/>

Performance



Rekentijd	vs. 20s	vs. t/2	
1,25s	9.4%	-	Onze bot in een geïntepreterde taal
2,5s	16.8%	67.5%	Abdessamad's bot
5s	24.5%	65%	
10s	40.2%	63%	
20s	50%	61%	Onze bot
40s	60%	60%	C++ versie van onze bot
80s	72%	-	
160s	80%	-	
320s	83.6%	-	