

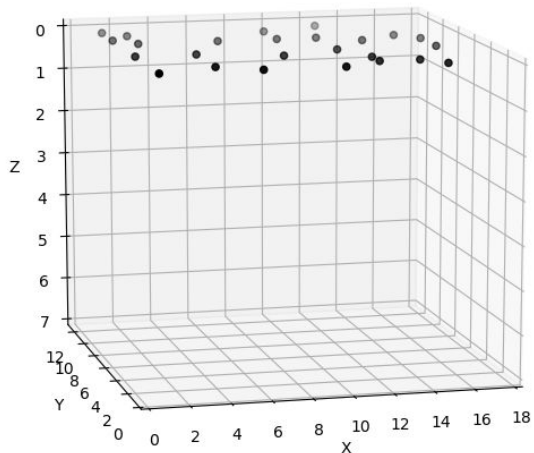
Chips and Circuits



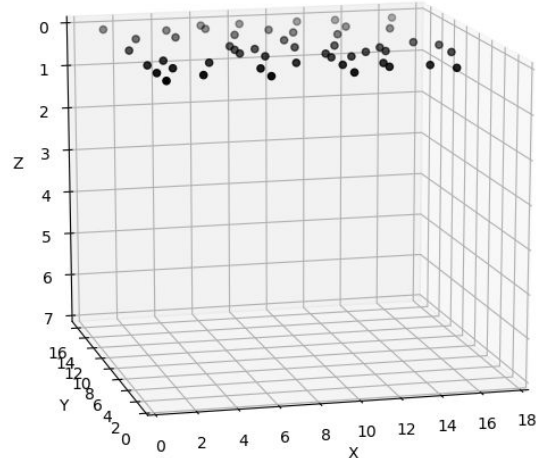
Inleiding

- >_ **Case**
- >_ **Complexiteit**
- >_ **Methoden**
- >_ **Resultaten**
- >_ **Conclusie**
- >_ **Discussie**

Printplaat #1



Printplaat #2



#1

23 > 4
5 > 7
1 > 0
... > ...
...
...

#2

12 > 20
23 > 20
6 > 9
... > ...
...
...

#3

0 > 13
0 > 14
0 > 22
... > ...
...
...

#4

42 > 3
3 > 48
14 > 6
... > ...
...
...

#5

34 > 21
48 > 47
38 > 16
... > ...
...
...

#6

16 > 10
25 > 17
1 > 11
... > ...
...
...

Vereisten

1. Paden mogen **niet kruisen**
2. Maak **alle verbindingen**
3. **Maximaal 8** lagen
4. Elke stap op het raster **kost 1 wire**
5. Vind de **goedkoopste oplossing**

= **constrained
optimization
problem**

>
CASE

Scorefunctie

$$\text{score}(S) = \text{wires} \times 1$$

> PROBLEM COMPLEXITY

Upper bound

Formula: $\text{ub}(S) = (\text{width} \times \text{height} \times \text{depth}) - \text{gates}$

Print 1: 2398

Print 2: 1847

> PROBLEM COMPLEXITY

Lower bound

Formula: $\mathbf{lb}(\mathbf{S}) = \sum_{i=1}^{\text{con}} (\sqrt{x_i^2 + y_i^2 + z_i^2})$

Print 1: netlist 1 **232.06**

netlist 2 **272.45**

netlist 3 **374.92**

Print 2: netlist 4 **461.55**

netlist 5 **447.59**

netlist 6 **585.00**

>
PROBLEM
COMPLEXITY

Lower bound (improved)

Formula: $lb(s) = \sum_{i=1}^{con} (\Delta x_i + \Delta y_i + \Delta z_i)$

Print 1: netlist 1 **290**

netlist 2 **341**

netlist 3 **472**

Print 2: netlist 4 **595**

netlist 5 **563**

netlist 6 **754**

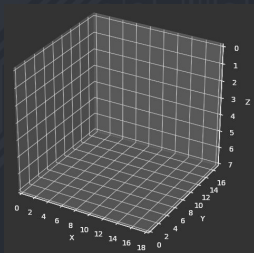
>_

PROBLEM COMPLEXITY

State space

$$\dot{x}(s) = \text{?????}$$

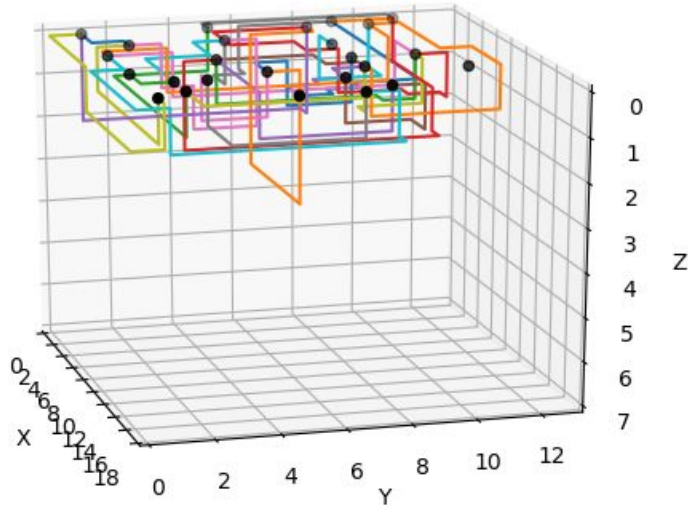
Numpy



Tekenen van een pad



Tekenen van een pad Dijkstra



1. Kortste route mogelijk
2. Ontwijken van gate als deze nog connecties moet maken

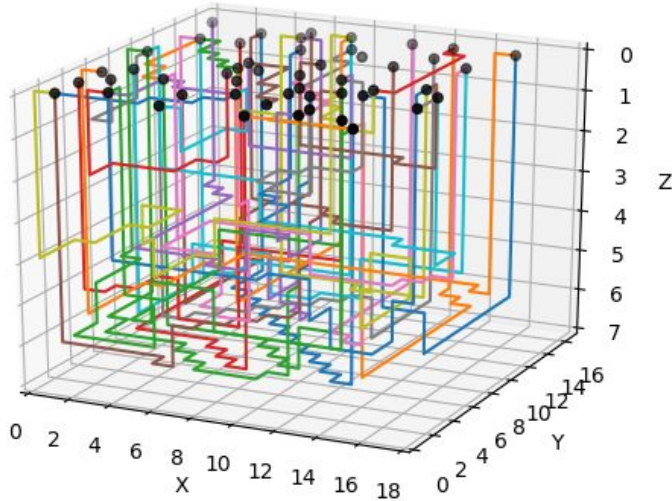
A				?			B
				Z			

3. 0 / 6 borden opgelost

>_ METHODEN

Tekenen van een pad

A*



1. “Goedkoopste” route
2. Lopen langs een onvoltooide gate is “duur”

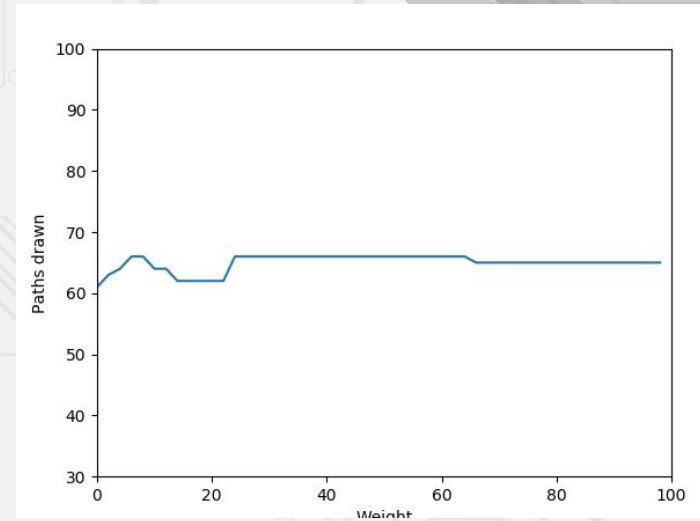
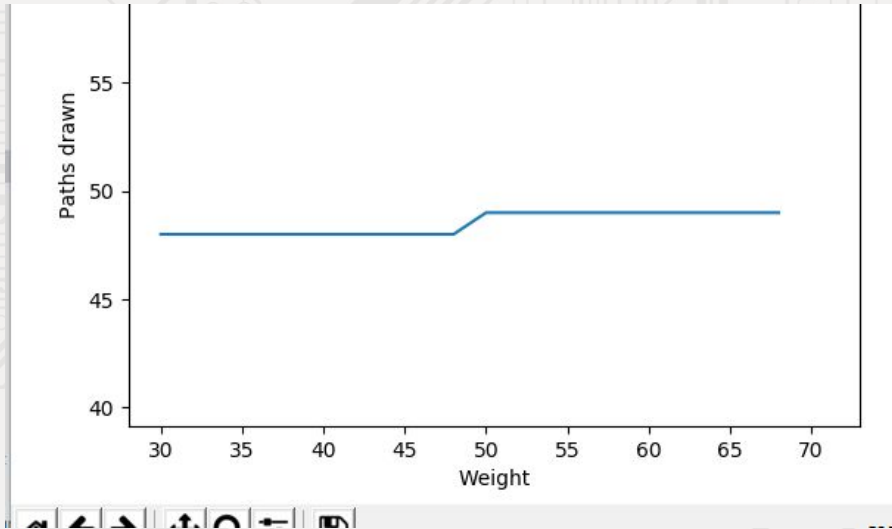
A				?			B
				Z			

3. Diepere lagen zijn goedkoper
4. 3 / 6 borden opgelost

>_ RESULTATEN

Heuristiek

“Lopen langs een onvoltooide gate is duur”



>_

CONCLUSIE

Wat hebben we geleerd?

1. 'De diepte in' = sleutel tot oplossen
2. **Soft heuristics** en **hillclimbing** voor bepalen 'cost'
3. Verschillende netlists vragen **verschillende "instellingen"**

Verder te onderzoeken

1. Een efficiëntere volgorde van uitvoeren van netlist?

2. Heuristiek “hemelsbreed tot doel” wordt ondergesneeuwd. Hoe vind je een goede balans?

3. Bekend met pathfinding door IDA* (Iterative deepening A*) of Trace of ... ?

4. Sommige paden later opnieuw tekenen? Welke? Waarom?

1. Aanpassen van de volgorde van uitvoeren van netlist
2. Netlist met de meeste benodigde verbindingen vooraan zetten

Voorbeeld netlist:

`[(0,1),(1,2),(2,0),(0,1),(1,2),(2,0)]`