

Baseline Chips and Circuits (De gigahertjes)

Het is ons tot nu toe nog niet gelukt om van alle chips en alle bijbehorende netlists een volledige oplossing te vinden, oftewel dat alle nets in een chip met elkaar verbonden zijn zonder collisions. Wel kunnen we de resultaten van het tot nu gemaakte algoritme uitdrukken in het percentage nets dat verbonden is van het totaal aantal te verbinden nets. De resultaten hiervan zijn in de afbeelding hieronder weergegeven:

```
the python man.py
Gevonden resultaat voor chip 0 en netlist 1 is:
100.0
Gevonden resultaat voor chip 0 en netlist 2 is:
100.0
Gevonden resultaat voor chip 0 en netlist 3 is:
100.0
Gevonden resultaat voor chip 1 en netlist 4 is:
87.16666666666666
Gevonden resultaat voor chip 1 en netlist 5 is:
74.25000000000001
Gevonden resultaat voor chip 1 en netlist 6 is:
46.160000000000004
Gevonden resultaat voor chip 2 en netlist 7 is:
71.179999999999994
Gevonden resultaat voor chip 2 en netlist 8 is:
67.99999999999997
Gevonden resultaat voor chip 2 en netlist 9 is:
58.94285714285715
```

Deze resultaten zijn bereikt door voor elke chip met een netlist honderd keer het algoritme uit te voeren en vervolgens het gemiddelde aantal verbonden wires te berekenen in procenten. Hierbij is er gebruik gemaakt van een algoritme die net voor net een wire probeert te maken. De keuze die per stapje van elke wire gemaakt wordt, is gebaseerd op de Manhattan-distance en het aantal bewegingsmogelijkheden van de volgende state. In het geval dat twee of meer opties voor het volgende stapje even goed zijn, wordt er een willekeurige keuze gemaakt tussen deze opties. Omdat de totale kosten van de niet oplossingen niet zoveel zeggen, hebben we moeten kijken naar het percentage van de keren dat we wel een oplossing hebben.

```

Gevonden resultaat voor chip 0 en netlist 1 is:
1000
Gevonden resultaat voor chip 0 en netlist 2 is:
989
Gevonden resultaat voor chip 0 en netlist 3 is:
993
Gevonden resultaat voor chip 1 en netlist 4 is:
354
Gevonden resultaat voor chip 1 en netlist 5 is:
25
Gevonden resultaat voor chip 1 en netlist 6 is:
0
Gevonden resultaat voor chip 2 en netlist 7 is:
28
Gevonden resultaat voor chip 2 en netlist 8 is:
0
Gevonden resultaat voor chip 2 en netlist 9 is:
0

```

Deze cijfers laten zien hoe vaak van de duizend pogingen er een goede oplossing gevonden is.

Om meer in detail te treden over de kostenfuncties van de oplossingen, beperken we ons tot de chips en netlists die ons oplossingen geven.

De oplossingen zijn niet volledig willekeurig gegenereerd omdat de kans dat er een oplossing komt dan te klein is en dat we dan onvoldoende resultaten zouden hebben om nu over te hebben. Het algoritme die deze oplossingen genereert is dezelfde als hierboven beschreven. De bias zit hem er dus in dat de nets op een relatief directe manier met elkaar verbonden zijn, vanwege het gebruik van de Manhattan-distance. De gemiddelde kosten en de varianties van de oplossingen zijn hieronder te vinden.

```

(base) MacBook-Pro-van-Tijmen:chipsandcircuits tijmen$ python main.py
Resultaten voor chip 0 en netlist 1 over 1000 oplossingen:
Gemiddelde kosten: 22.864
[Variantie in kosten: 1.8455039999999994
Resultaten voor chip 0 en netlist 2 over 996 oplossingen:
Gemiddelde kosten: 58.29016064257028
Variantie in kosten: 202.6798630264028
Resultaten voor chip 0 en netlist 3 over 987 oplossingen:
[Gemiddelde kosten: 84.21175278622087
Variantie in kosten: 94.92476562074958
Resultaten voor chip 1 en netlist 4 over 355 oplossingen:
Gemiddelde kosten: 499.4478873239437
Variantie in kosten: 1810.4839039873043
Resultaten voor chip 1 en netlist 5 is over 27 oplossingen:
Gemiddelde kosten: 683.8888888888889

```