Resultaten van experimenten

**Alle gegevens zijn gegenereerd met 1.000.000 iteraties.**

# Tabel Nederland: random beginpunten vs. uithoeken beginpunten

|  |  |  |
| --- | --- | --- |
|  | **Random beginpunten score** | **Uithoeken beginpunten score** |
| Aantal stations | 61 |  |
| Aantal verbindingen: | 89 |  |
| Aantal minuten: | 1755 |  |
| **Score:** | **9779,8245** |  |

# Tabel Nederland variatie van het aantal trajecten per lijnvoering:

# Tabel Nederland: scorefunctie aanpassen

*We willen onderzoeken welke scorefunctie een redelijke score geeft met de hillclimber met willekeurige beginstations, er worden steeds 11 trajecten gemaakt. Met 100.000 iteraties.*

|  |  |  |
| --- | --- | --- |
| **Score** | **Scorefunctie** | **Toelichting** |
| 9667.6236 | p\*10000 - (t\*20 + min/**100000**) | Sporen: 88  Stations: 61  Minuten: 1689 |
| 9667.4711 | p\*10000 - (t\*20 + min/**10000**) | Sporen:88  Stations: 61  Minuten: 1693 |
| 9665.8804 | p\*10000 - (t\*20 + min/**1000**) | Sporen:88  Stations:61  Minuten:1760 |
| 9650.2604 | p\*10000 - (t\*20 + min/**100**) | Sporen: 88  Stations:61  Minuten: 1738 |
| 9496.7404 | p\*10000 - (t\*20 + min/**10**) | Sporen: 88  Stations:61  Minuten:1709 |

Naar aanleiding van het experiment waarin we met 100.000 iteraties de hillclimber runnen. Hebben we gekozen voor de scorefunctie p\*10000 - (t\*20 + min/**10000**). Er is vergeleken met de originele scorefunctie een 0 van “min/100000” afgehaald. Zo was de scorefunctie ook bedoeld. Echter, zien we dat er weinig veranderd wanneer deze 0 eraf wordt gehaald. En ook wanneer er meer nullen van de functie worden afgehaald veranderd de score die uit de score functie komt nauwelijks. Vandaar dat wij ervoor hebben gekozen de score functie aan te nemen zoals deze oorspronkelijk was bedoeld: p\*10000 - (t\*20 + min/**10000**).

We hebben dit experiment gedaan om te onderzoeken wat het zou betekenen voor onze scores berekend door de scorefunctie, als de minuten zwaarder zouden wegen. Dit blijkt echter niet van groot belang.

# Tabel Nederland: verschillende neighbour algoritmes

*We willen onderzoeken welk algoritme het beste resultaat oplevert* ***in de hillclimber.***

*De volgende algoritmes hebben we gehillclimbed:*

**Nearest neighbour algoritme:** Het nearest neighbour algoritme.

**Farest neighbour algoritme:** Het nearest neighbour algoritme, maar in plaats van het nearest spoor, kiest het het farest spoor.

**Random neighbour algoritme**: Het nearest neighbour algoritme, maar in plaats van het nearest spoor, kiest het een random spoor.

**Random algoritme:**  Willekeurige beginsporen, willekeurige keuzes, willekeurige trajectvorming. Er is maar één constraint: de trajectlengte mag niet langer dan 180 minuten duren.

Uit de hillclimbing komen de volgende scores:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *1000.000 iteraties* | Nearest Neighbour algoritme | Farest neighbour algoritme | Random neighbour algoritme | Random algoritme |
| Aantal stations | 61 | 61 | 61 |  |
| Aantal sporen | 89 | 87 | 71 |  |
| Aantal minuten | 1755 | 1792 | 1564 |  |
| Score | **9779,8245** | 9555.1017 | 7757,3717 | *Geen verbetering t.o.v. nearest Neighbour.* |

Tabel: algoritmes tegen elkaar uitgezet.

**Conclusie:** Naar aanleiding van dit experiment, hebben we ervoor gekozen door te gaan met het … algoritme.