



RailNL

Team Dapper

Paco van der Vliet
David Verboom
Anouk Raanhuis

Inhoud

- I. Case uitleg
- II. Statespace
- III. Baseline
- IV. Algoritmes
- V. Resultaten
- VI. Vergelijking
- VII. Conclusie
- VIII. Future work



I. Case uitleg

- Dienstregeling voor treinen
- Noord-Holland & Nederland
- Connecties
 - ['Alkmaar', 'Hoorn'] -> 24 min
- Trajecten creëren binnen een tijdsframe
 - ['Den Helder', 'Alkmaar', 'Hoorn'] -> 60 min
- Kwaliteitsscore (K)
 - $K = p \cdot 10.000 - (T \cdot 100 + \text{Min})$



II. Statespace

- N = het aantal trajecten in je lijnvoering
- X = het aantal maximale connecties per station
- Y = het maximum aantal bereden verbindingsen in 1 traject

$$statespace = \sum_{i=0}^N (N - i) * X^Y$$

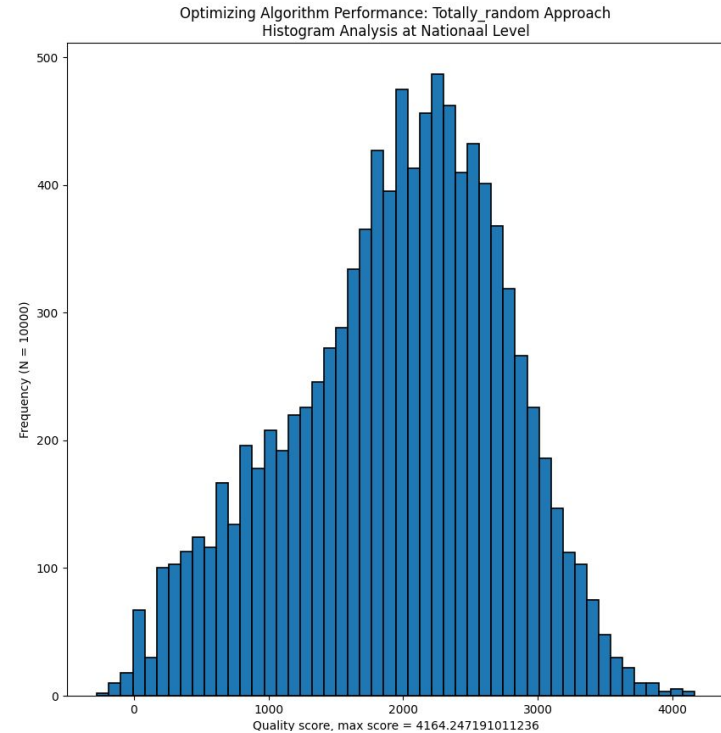
$$Y = \frac{tijdsframe}{minimum\ duur\ traject} - 1$$

- Voorbeeld:
 - Utrecht Centraal: 9 connecties
 - Tijdsframe = 180 minuten
 - Maximum trajecten = 20 trajecten
 - Minimum duur = 5 minuten
 - $20 * (9^{36}) + 19 * (9^{36}) + 18 * ... = 5,45 \times 10^{35}$



III. Baseline

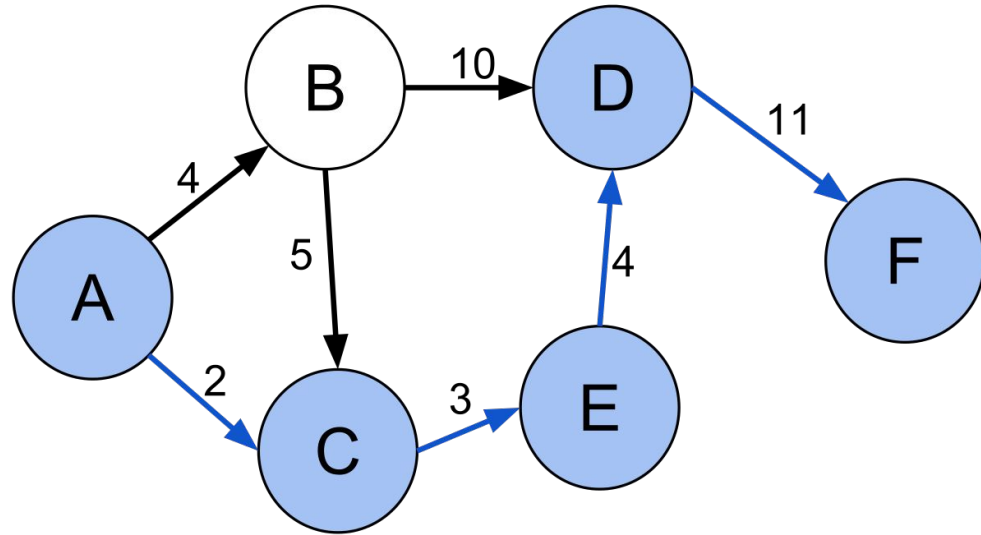
- Total Random Algorithm
 - Random hoeveelheid trajecten
 - Random startstation
 - Random keuze uit tracks



histogram 1: Totally Random

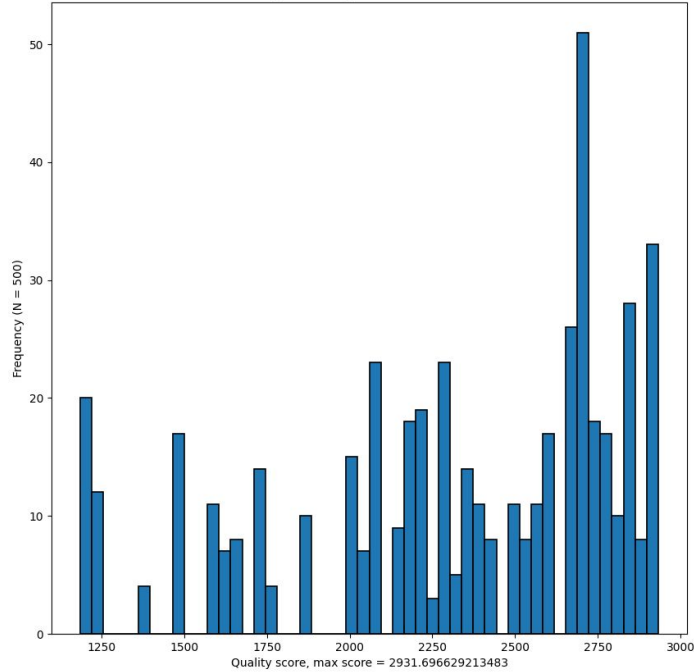
IV. Algoritmen

- Totally random
- Semi random
- Greedy
- Breadth First
- Depth First
- Hillclimber
- Local Search

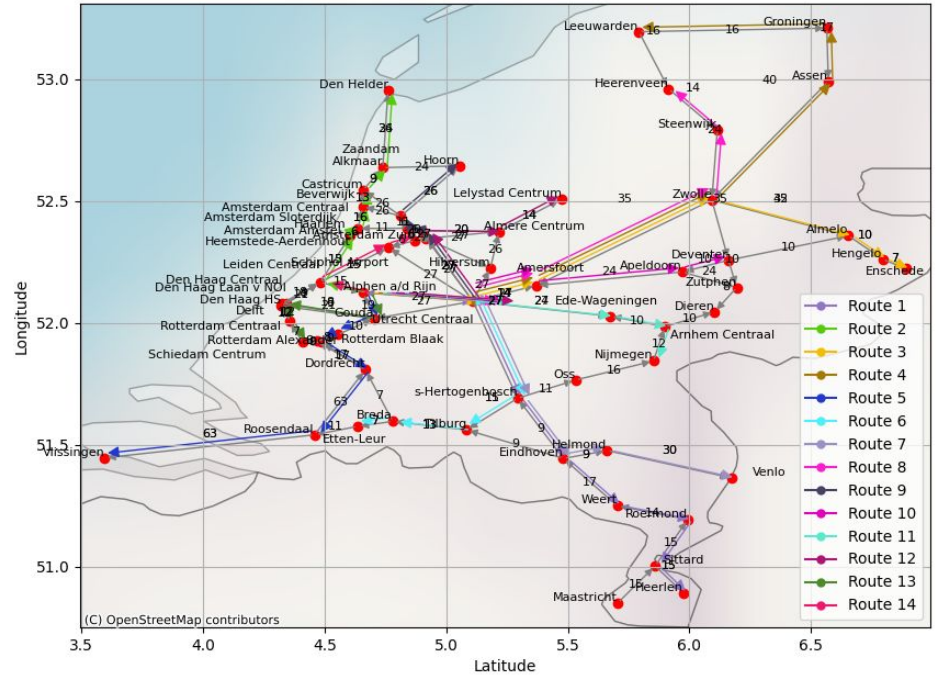


V. Resultaten Breadth-First

Optimizing Algorithm Performance: Breadth first Approach
Histogram Analysis at Nationaal Level

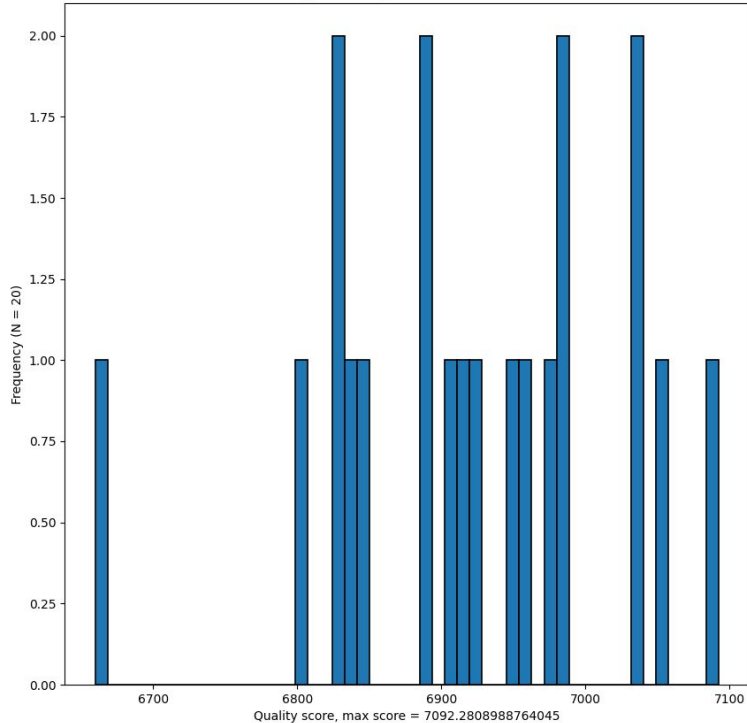


Optimizing Algorithm Performance: Breadth first Approach
Visualisation at Nationaal Level

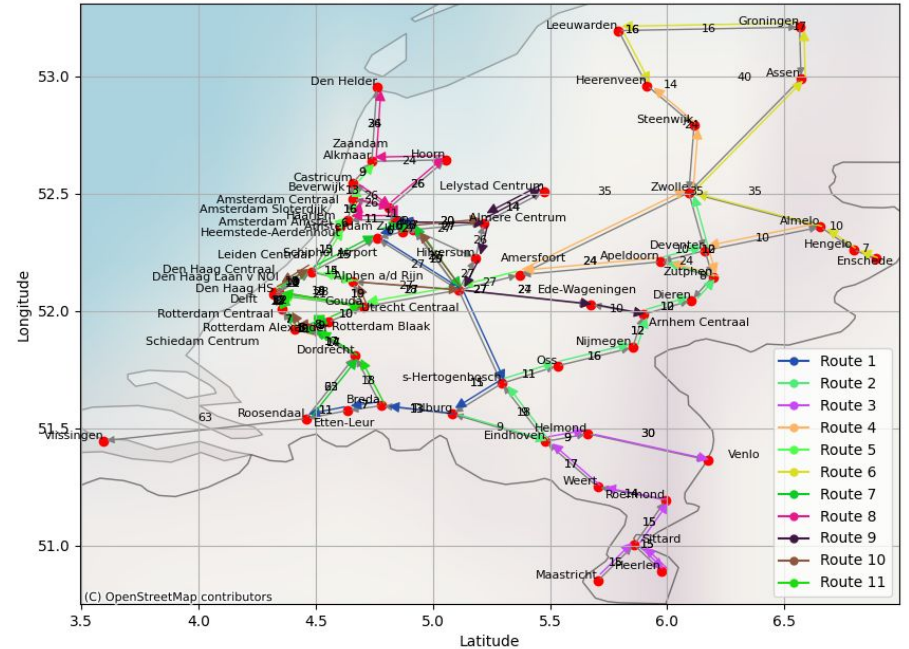


V. Resultaten Local Search

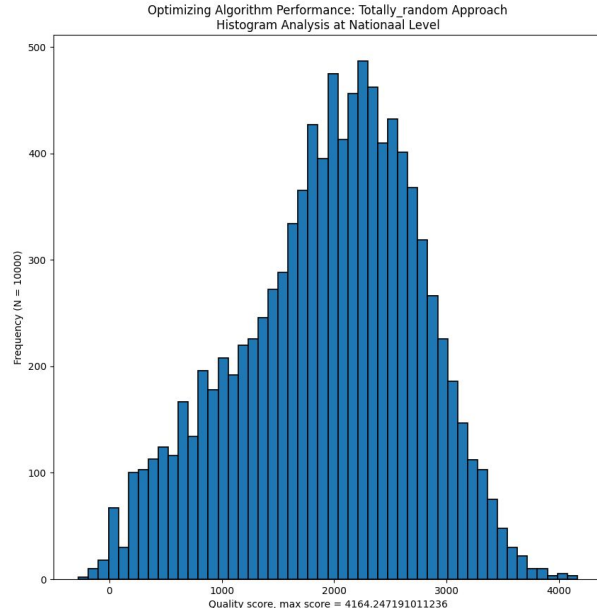
Optimizing Algorithm Performance: Local_search Approach
Histogram Analysis at Nationaal Level



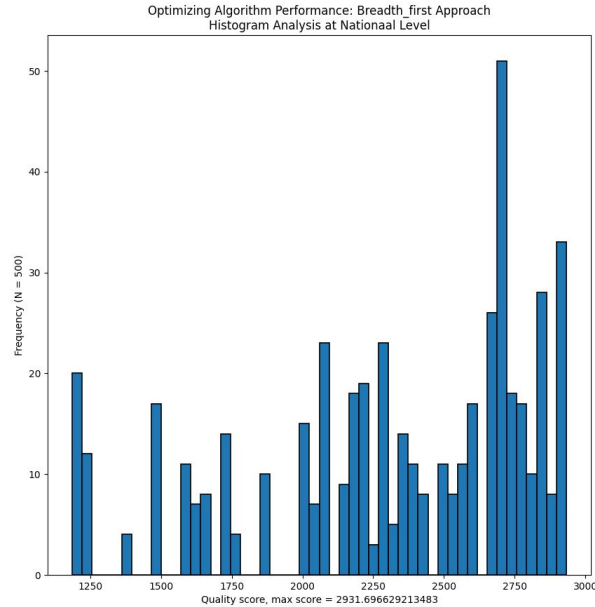
Optimizing Algorithm Performance: Local_search Approach
Visualisation at Nationaal Level



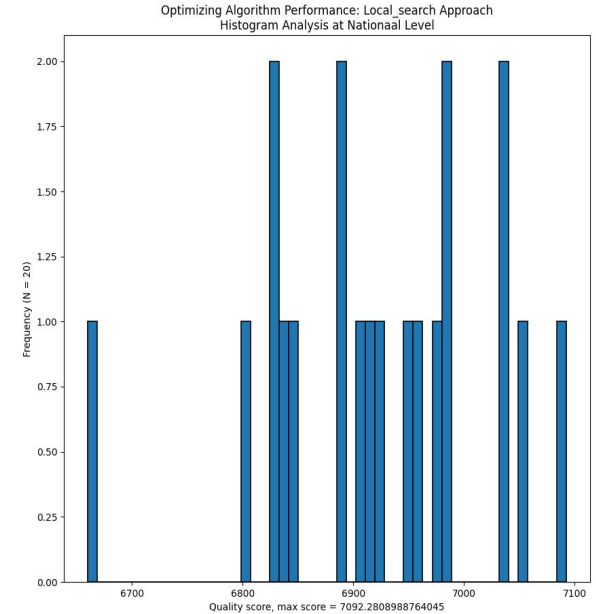
VI. Vergelijking



histogram 1: Totally Random



histogram 2: Breadth First



histogram 3: Local Search

VII. Conclusie

- Iteratieve algoritmes werken beter dan constructieve algoritmes
 - Alle niet dezelfde routes doorlopen zonder bijhouden score.
 - Iteratief gevonden oplossingen verbeteren is sneller. Depth- en Breadth First werken niet goed bij de state space van onze case¹.
- Beste score gemeten is 7092,2808988764045
 - 11 trajecten
 - P = 97,75%

1. Liang, P., [Stanford Online]. (2022, 31 mei). *Constraint Satisfaction Problems (CSPs) 1 - Overview* | Stanford

CS221: AI (Autumn 2021) [Video]. YouTube. <https://www.youtube.com/watch?v=-lO4fPO0rxk>

VIII. Future work

- Heuristieken
- Randomheid toevoegen
 - Breadth First: limiet is start vanaf 1 station, ook mogelijkheid om per traject van een nieuw station te beginnen. Bezochte stations, eindstation eerste traject
- Probability
 - Aannemen slechtere oplossingen, simulated annealing/aangepaste local search



Vragen?

- EINDSTATION -

