

# Baseline RailNL

Spoorschavuiten  
Sverre, Massimo & Sipke  
Algoritmen en Heuristieken  
UNIVERSITEIT VAN AMSTERDAM

June 13, 2024

In dit document wordt de baseline van ons project toegelicht. Het betreft een algoritme dat iteratief treinen toevoegt aan een dienstregeling.

## 1 Het algoritme `randomise.py`

Om een dienstregeling te genereren wordt per trein een traject aangemaakt. Daartoe wordt eerst een willekeurig station als startpunt aangewezen. Vervolgens wordt gecontroleerd welke verbindingen er vanuit dat station mogelijk zijn en wordt één daarvan willekeurig geselecteerd. Dit proces herhaalt zich totdat de lengte van het traject dat de trein aflegt de maximale duur in minuten bereikt heeft.

Na het toevoegen van een trein aan de dienstregeling, wordt gecontroleerd of elk stuk spoor op de kaart gebruikt is. Als dat zo is, eindigt het algoritme. Als dat nog niet zo is, wordt een nieuwe trein toegevoegd op de manier die hierboven beschreven is. Dit proces stopt wanneer elk stuk spoor gebruikt is, of wanneer het maximale aantal treinen is toegevoegd aan de dienstregeling.

## 2 Uitkomsten

Met `randomise.py` worden eerst willekeurig stations geselecteerd en daarna willekeurig trajecten vanuit dat beginpunt. Daarom is het waarschijnlijker dat een stuk spoor vanuit een station met relatief weinig verbindingen wordt opgenomen in een traject, dan een stuk spoor vanuit een station met relatief veel verbindingen. Het gevolg is stukken spoor aan de rand met grotere kans bereden worden dan stukken spoor in een meer verzadigd deel van de kaart. We verwachten daarom geen perfect normaal verdeelde uitkomsten. Het algoritme is 100.000 keer uitgevoerd en de resultaten van de doelfunctie

$$K = p \cdot 10000 - (T \cdot 100 + Min) \tag{1}$$

zijn weergegeven als histogram in Figuur (1). Te zien is dat uitkomsten rond de 6000 het vaakst voorkomen. Het gemiddelde van de uitkomsten is 5870. De uitkomsten zijn inderdaad niet geheel normaal verdeeld; het valt op dat de verdeling linksscheef is (dat wil zeggen dat de kans dat de uitkomst lager is dan het gemiddelde, groter is). Linksscheefheid heeft onder andere als eigenschap dat het gemiddelde links van de piek ligt.

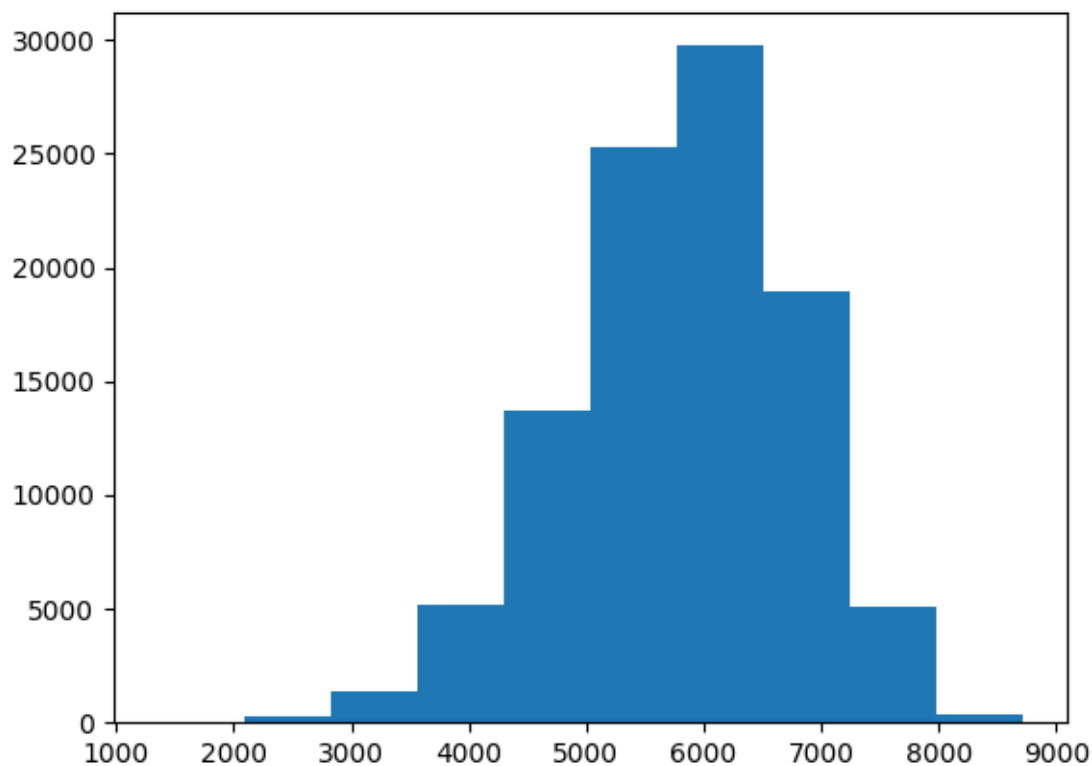


Figure 1: Een histogram van de uitkomsten van `randomise.py`. Op de horizontale as staan de uitkomsten van de doelfunctie (1), op de verticale as het aantal keer dat deze uitkomsten voorkwamen in de simulatie.

In het kader van RailNL is het interessant om op te merken dat er geen waarden boven de 9000 worden gegenereerd door `randomise.py`. Dat betekent dat de kans dat een dienstregeling met een score boven de 9000 wordt gegenereerd door dit algoritme – gebaseerd op deze ene simulatie – kleiner is dan één op honderdduizend. De implicatie is ook dat deze score niet verwacht kan worden bij een runtime van zeven minuten (zestien minuten op een andere laptop, dit verschilt). Een algoritme is daarom verdedigbaar beter dan deze baseline als de kans op een hogere score in eenzelfde simulatie groter is.