

Resultaat individueel onderdeel

Spoorschavuiten

Sipke de Boer

12480797 - Algoritmen en Heuristieken

UNIVERSITEIT VAN AMSTERDAM

June 21, 2024

1 Functionaliteit

Als individueel onderdeel wil ik een Hill Climber-algoritme implementeren. Dit moet een dienstregeling van bijvoorbeeld `randomise.py` of de greedy variant `randomise2.py` als input nemen. Vervolgens worden er iteratief kleine aanpassingen gedaan en wordt er gekeken of dit de oplossing beter maakt. Concreet kan dit gebeuren door één van de treinen opnieuw in te plannen en de waarden van de doelfunctie te vergelijken.

2 Pseudocode

We hebben een class `HillClimber` nodig, die in een bestand `hillclimber.py` staat en die wordt aangeroepen in `main_script.py`. We beginnen met de class:

```
class HillClimber:
    def __init__(Schedule):
        # Maak een kopie van Schedule-object
        # Bereken de doelfunctie

    def mutate_train():
        # Plan een nieuwe trein in indien er nog sporen ongebruikt zijn

    def mutate_schedule(nr_of_trains=1):
        # Verwijder n treinen
        # Doe n keer mutate_train()

    def check_solution():
        # Vergelijk waarden doelfuncties
        # Sla nieuw Schedule op indien het beter is

    def run(iterations, nr_of_trains=1):
        # Sla iterations op

        # Loop over de iterations
        # Doe mutate_schedule(nr_of_trains)
        # Doe check_solution()
```

Nu doen we main:

```
# Creëer een random Schedule
# Creëer een instance van HillClimber
# Run deze
# Print de beste score
```

3 Aansluiting op overige code

De aansluiting op de overige code is tweeledig. Als input neemt `HillClimber` een reeds gegenereerde oplossing `Schedule` (en een integer). Deze oplossing moet door een ander algoritme gegenereerd worden, bijvoorbeeld een random algoritme. Als output geeft `HillClimber` volgens de bovenstaande pseudocode strikt genomen niets. Er kan echter wel opgeslagen data genomen worden middels `HillClimber.<data>`, waar `<data>` refereert naar de beste kopie van de dienstregeling of de bijbehorende beste waarde van de doelfunctie.

4 Anders uitgepakte gedeeltes

Ik had niet voorzien dat ik een methode `update_used_connections` zou moeten toevoegen. Dat bleek noodzakelijk, omdat het niet zo is dat elke trein die uit de dienstregeling wordt verwijderd, resulteert in het verwijderen van elke verbinding die die trein bereed uit de verzameling bereden verbindingen. Het kan namelijk zo zijn dat één of meerdere van die verbindingen ook bereden worden door één of meerdere andere treinen. Deze methode controleert voor elke verbinding opnieuw of hij bereden wordt en zet de boolean juist.

Tevens heb ik een methode `generate_output` toegevoegd. Deze spreekt eigenlijk voor zich, maar ik was vergeten dat ik hem nog moest implementeren.

Als aanvulling op het Hill Climber-algoritme heb ik een variant toegevoegd die stopt als er na een vast aantal iteraties geen verbetering is gevonden. Ook heb ik de heuristiek "laat in elk kopstation een trein starten" toegevoegd aan zowel het Greedy- als het Hill Climber-algoritme. Opvallend is dat deze heuristiek consistent lagere scores oplevert, maar dat de hoogste scores die we zonder de heuristiek hebben gevonden wel overeenkomen met het statement.

5 Beschrijving van commits

In deze sectie geef ik van elke voor het persoonlijk deel relevante commit een korte beschrijving. Specifiek doe ik dat voor het Hill Climber algoritme, de overige commits laat ik buiten beschouwing. Ik ga ervan uit dat de persoonlijke input daarvoor wordt erkend als *group effort*.

<https://github.com/Sverremans/spoorschavuiten/commit/b2d26dacecc145f4afecce82b062c6c91e985ec7>

In deze commit heb ik het grootste deel van het Hill Climber-algoritme gecodeerd. Het betreft de algemene implementatie ervan, hierna volgde specificatie op de case en bugfixing.

<https://github.com/Sverremans/spoorschavuiten/commit/99f40bfe1eaf46c691c3ea8a0597e583445a3270>

In deze commit is een heleboel bugfixing doorgevoerd, met als resultaat dat de check50 slaagde.

<https://github.com/Sverremans/spoorschavuiten/commit/a55aec3c78834931f109125d85f15868faafe4c6>

In deze commit heb ik de code voor de Hill Climber compatibel gemaakt met de code van Sverre zodat er een visualisatie kan worden gemaakt. Daarnaast heb ik de code iets overzichtelijker en gebruiksvriendelijker gemaakt.

<https://github.com/Sverremans/spoorschavuiten/commit/3a4bc1075358bf32fb0cc265d99aaaf8ee3c5a9a>

In deze commit is meer ondersteunende code toegevoegd ten behoeve van de visualisatie. In de hierop volgende commits wordt deze code verbeterd en opgeschoond.

<https://github.com/Sverremans/spoorschavuiten/commit/3869638c5f72e7651d81caa519ee737e94ce5033>

In deze commit heb ik een versie van het Hill Climber-algoritme toegevoegd die stopt als er na een vast aantal iteraties geen verbetering is gevonden.

<https://github.com/Sverremans/spoorschavuiten/commit/76f24b7b186cd6d867262dfbbcd26bc7a1fba0b3>

In deze commit heb ik bugs gefixt, maar op een onhandige manier. Later kom ik hierop terug, want ik heb hierdoor wel weer iets geleerd.

<https://github.com/Sverremans/spoorschavuiten/commit/a8e4f0056c4c15d5829494f74085c7b36f16daf4>

In deze code is de stijl van de variabelen meer 'Pythonic' gemaakt.

<https://github.com/Sverremans/spoorschavuiten/commit/9ad9b6e1665f8e87f389331e20814a77a18dcb6f>

In deze commit is een stuk code efficiënter opgeschreven.

<https://github.com/Sverremans/spoorschavuiten/commit/08ee36d34077c226a3e236f8934f2b0ea7bb2449>

In deze commit heb ik de bovengenoemde heuristiek toegevoegd aan het Greedy-algoritme.

<https://github.com/Sverremans/spoorschavuiten/commit/626225984dcb2936b541a1b254a469ced070bc9b>

In deze commit heb ik de bovengenoemde heuristiek toegevoegd aan het Hill Climber-algoritme.

<https://github.com/Sverremans/spoorschavuiten/commit/bcf6195d89a081a9167a8828a941dbd692251345>

In deze commit heb ik Nederlandstalige variabelen vertaald, zodat de algehele stijl van de code consistent is.

<https://github.com/Sverremans/spoorschavuiten/commit/33bbdecf4a1d3c4f6432e7b2bbe6fb96030fb04b>

En deze en de hierop volgende commits heb ik de code geoptimaliseerd op runtime. De eerder genoemde onhandige bugfix heb ik teruggedraaid en verbeterd. Voor deze optimalisatie heb ik SnakeViz en Jelle's kennis hierover gebruikt. De code is nu significant sneller.