ARTIFICIAL INTELIGINCE
MASTER'S IN DATA SCIENCE AND ENGINEERING, FEUP
2022-2023

Optimal Inspection Routes Problem - ASAE Use Case

Farzam Salimi - up201007922
Luís Henriques - up202204386

Introduction

The presented problem includes finding the optimal inspection routes for ASAE, which is responsible for ensuring compliance with regulations among economic operators in Portugal. The problem is a simplified version of the actual ASAE inspection routes problem and is classified as an instance of the Vehicle Routing Problem (VRP). ASAE uses inspection brigades and each brigade is assigned to a specific route.

We focused on minimizing the total travel time, including distance times, waiting times and inspection times. We then created an implementation for a way to also minimize the number of routes or vehicles.

Table of Contents

ROPTIMAL INSPECTION ROUTES PROBLEM
- ASAE USE CASE

Specifications of the work performed

Establishments dataset

Information on 1000 establishments, including their unique identifier (ID), name, geographical location, latitude, longitude, estimated inspection duration, inspection utility value, and opening hours.

Distances dataset

Travel time matrix, in seconds, between each of the 1000 establishments and between the departure/arrival establishment (depot), shown by ID zero. Therefore, the size of the matrix is 1001x1001

Problem Formulation

1

Solution representation

How are the solutions given to the algorithms and to the evaluation function?

2

Neighborhood/Mutation and crossover functions

How are the solutions changed so that new solutions are found, and hopefully avoid local optima?

3

Evaluation functions and constraints

How to evaluate the solutions found by the algorithms and to what are they constrainted?

Random solution for the very small set (20 establishments, 2 vehicles/routes...

[1, 2, 1, 1, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2]

...where each value is the route and the index (+1) is the establishment

Converting to sets with establishments per route and...

Route 1: {1, 3, 4, 7, 8, 10, 11, 15, 18, 19}
Route 2: {2, 5, 6, 9, 12, 13, 14, 16, 17, 20}

- We can easily convert the solution representation into routes after getting the order and adding the depot.
- Doesn't cause the creation of unfeasible solutions using genetic or neighborhood operators.

Solution representation

Neighborhood/Mutation and crossover functions

Neighborhood operators

One generates a new solution by randomly selecting an establishment and assigning it to different vehicles. The second system generates a new solution by exchanging the vehicles between two establishments

Crossover functions

Midpoint crossover, Randompoint crossover and UX crossover (each gene has a 50% chance of being chosen for the child's chromosome)

Mutation functions

One takes a solution and swaps two random positions within it, and the other randomly changes one position in a solution to a value chosen uniformly at random between 1 and the number of vehicles available

Evaluation functions

How to set the order of the routes?... Informed search methods

Greedy-search

We first set the routes according to how close the following establishment is to the depot.

A* Algorithm

More complex approach (and better)... For the available establishments we calculate the total travel time to get there plus the time to reach the depot.

Heuristic: Geocentric time matrix

Uses latitude and longitude of every establishment to calculate the shortest distance between then on Earth, and from that extract time in seconds. It's sort of euclidean distance but in a curved surface.

Approach and algorithms applied

Meta-heuristics

GENETIC ALGORITHM

Based on the evolution theory of Darwin

SIMULATED ANNEALING

Based on the process of making glass bottles, for example.

TABU SEARCH

Based on the meaning of the word "tabu"

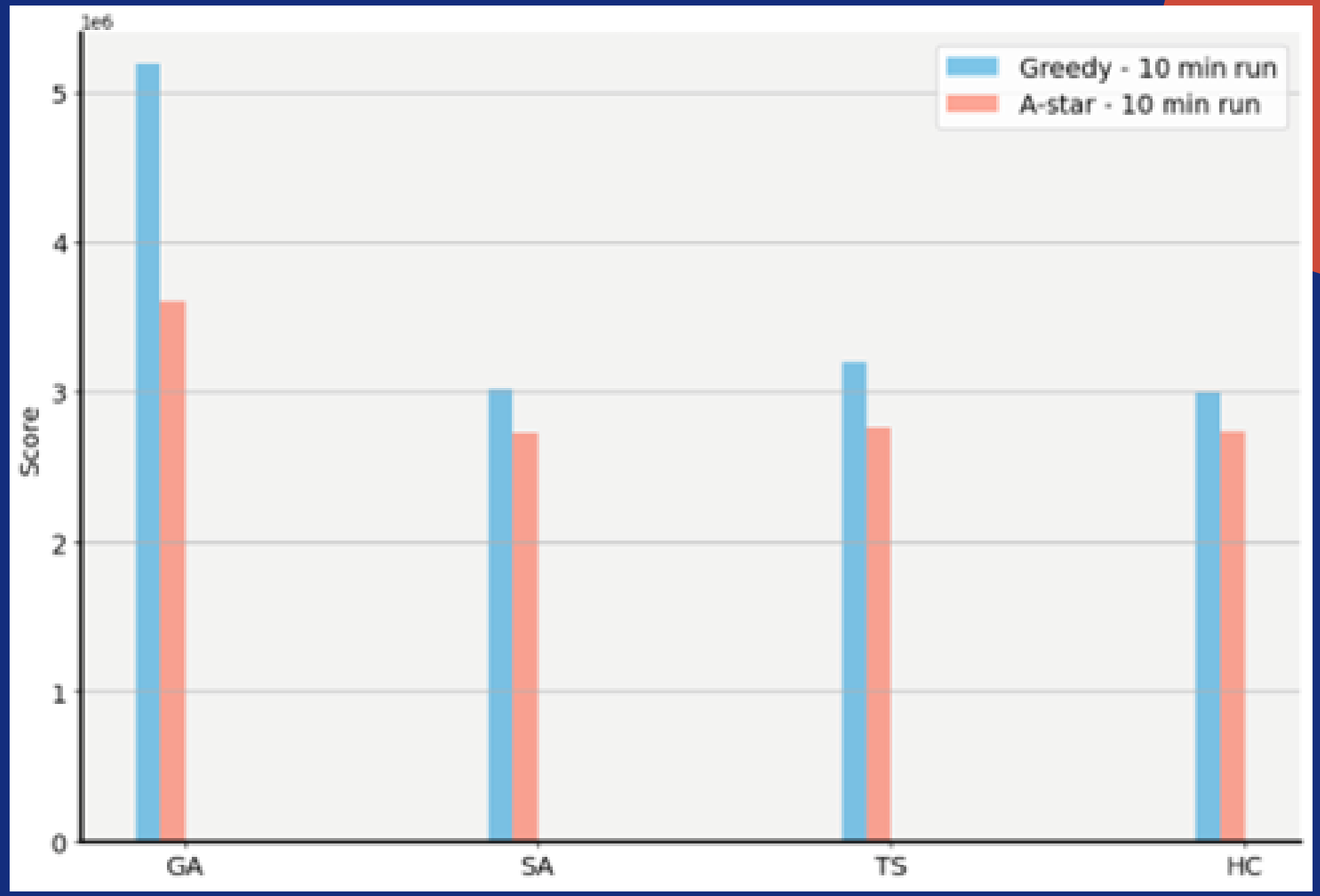
HILL-CLIMBING

Standard meta-heuristic used

Results

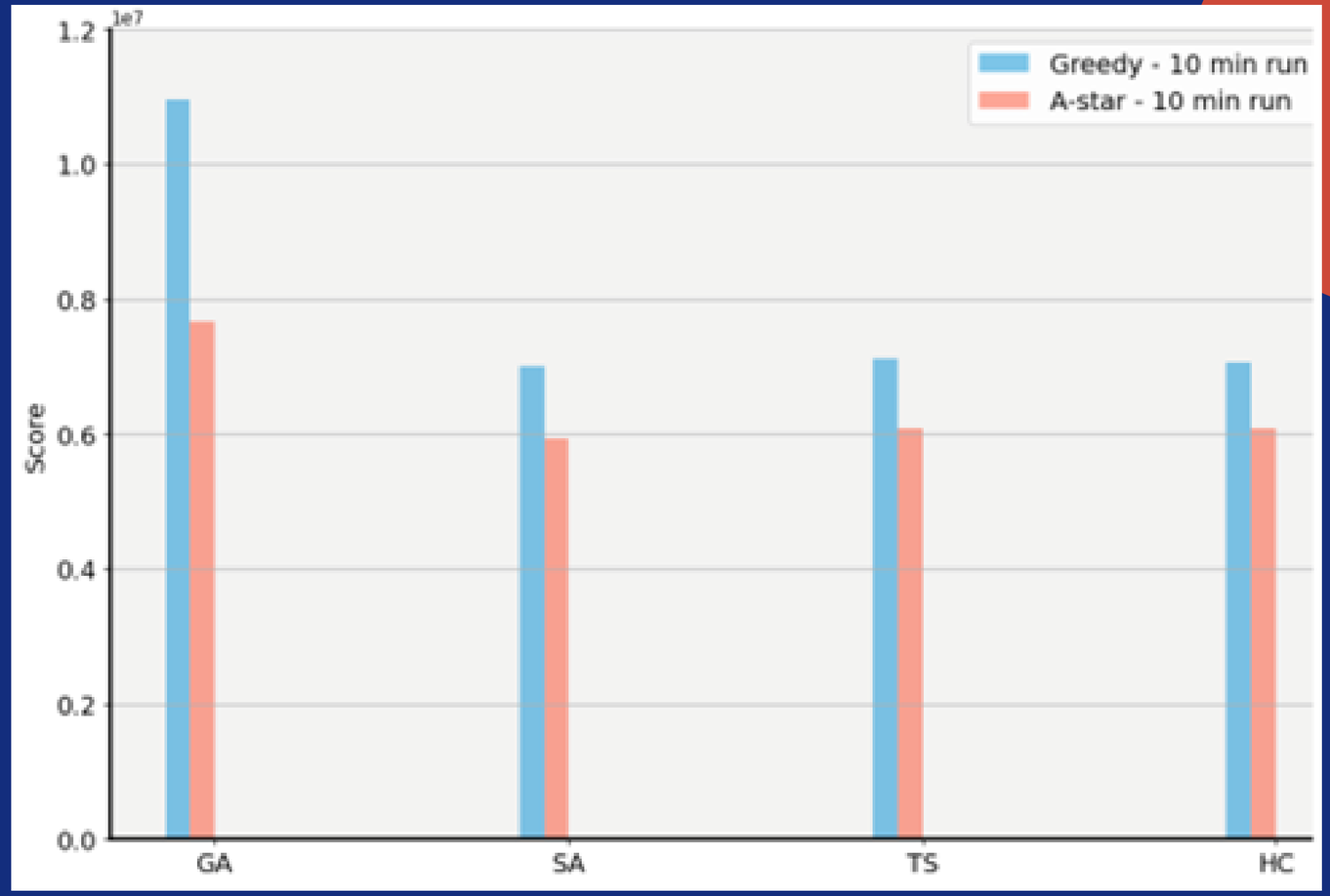
Comparing the meta-heuristics and objective functions

500 Establishments, 50 routes/vehicles, running time: 10 minutes



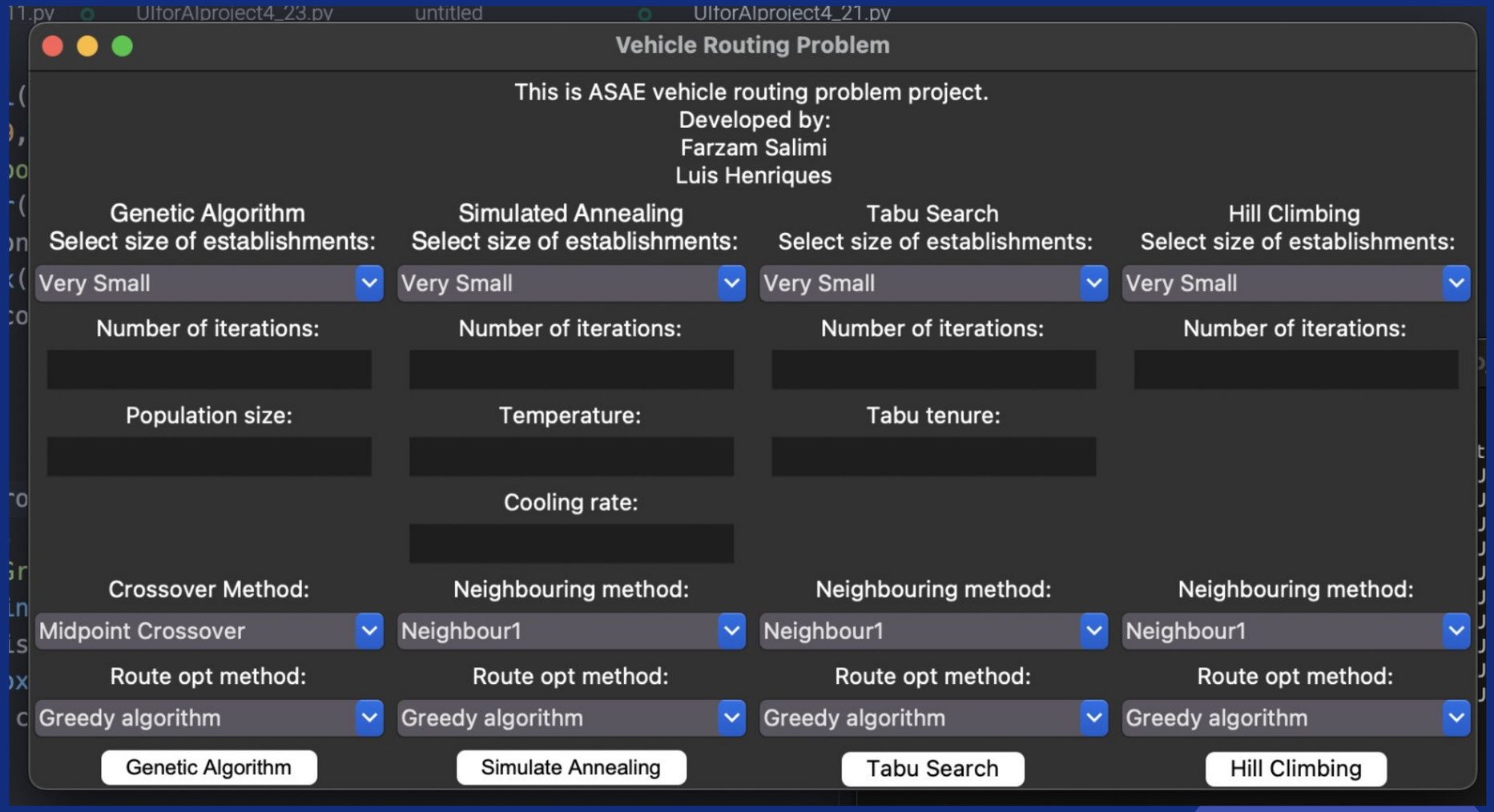
Comparing the meta-heuristics and objective functions

1000 Establishments, 100 routes/vehicles, running time: 10 minutes



Main results

User Interface



UI for map generation