Advanced Software Software Engineering 2023

Green Vehicle Routing Problem

Beschreibung | http://www.vrp-rep.org/variants/item/g-vrp.html

The vertex set is made up of a depot (vertex 0), a set of n customers and a set of s refuelling stations. It is assumed that all the refuelling stations can handle an unlimited number of vehicles. An unlimited fleet of alternative fuel vehicles (AFVs) is based at the depot. The objective is to find m routes for m AFVs (m unrestricted) such that each customer is visited by exactly one route and the sum of the routes' distances is minimized. Each route can travel a maximum driving distance without refuelling of Q units and has a maximum driving time T. A route travelling from i to j consumes c(ij) units of distance and t(ij) units of time. The time t(ij) is assumed to be proportional to the distance c(ij) from i to j and computed as t(ij)=c(ij)/v where v is the vehicle speed. A route visiting a customer i consumes a service time s(i) and a route visiting a refuelling station k consumes a refuelling time r(k) (service times are assumed equal for all customers, refuelling times are assumed equal for all stations). It is also assumed that the refuelling time is incurred when leaving the depot for the first time. The time consumption of a route cannot exceed T and the distance consumption of a route cannot exceed Q. However, whenever a route visits a refuelling station its distance consumption is reset to 0.

Bearbeitung | **Team** | **2** (S01, S02), **3** (S01, S02, S03) oder **4** Studierende (S01, S02, S03, S04).

Implementierung einer technisch einwandfrei lauffähigen Applikation in **Java 17.0.7 (LTS)**.

Dataset | http://www.vrp-rep.org/datasets/item/2020-0013.html

- **S01 | Import und Zielfunktion**
  [01a] Import der Dateninstanz (small und large)
  20 Punkte (Implementierung: 15 Punkte | Test: 5 Punkte)
  [01b] Zielfunktion zu Dateninstanz (small und large)
  30 Punkte (Konzeptionierung: 15 Punkte | Implementierung: 10 | Test: 5 Punkte)

- **S02 | Algorithmus**
  [02a] Single-Threaded ACO (Implementierung: 15 Punkte | Test: 5 Punkte)
  [02b] Multi-Threaded ACO (Implementierung: 25 Punkte | Test: 5 Punkte)

- **S03 | Visualisierung mit JavaFX und Data Analytics**
  [03a] Konvergenz (10 Punkte), [03b] Touren (15 Punkte),
  [03c] Bereiche mit hoher/niedriger Konvergenz (15 Punkte),
  [03d] Anzahl Verbesserungen je Ameise bezüglich Gesamtanzahl Iterationen (10 Punkte).

- **S04 | Graphischer Editor in JavaFX zwecks Erstellung eigener Dateninstanz**
  (50 Punkte)

Abgabe | 7-Zip-Datei mit IntelliJ-Projekt sowie readme.txt mit Zuordnung Aufgabe/Matrikelnummer.

Prof. Dr. Carsten Müller | Software Engineering