
Aufgabe (Evolutionäre Algorithmen für TSP)

Diese Aufgabe hat die Lösung des TSP mit populationsbasierenden Metaheuristiken zum Ziel. Bitte erstelle einen Bericht zu Deiner Lösung und sende den Bericht, die Antworten zu den untenstehenden Fragen und Deinen Sourcecodes an den Betreuer.

1. Entwerfe und Implementiere zwei Mutationsoperatoren für TSP. Erläutere Deine Entwurfsentscheidungen.
2. Entwerfe und Implementiere zwei Rekombinationsoperatoren für TSP. Erläutere Deine Entwurfsentscheidungen.
3. Verwende Deine Mutations- und Rekombinationsoperatoren, um drei verschieden konfigurierte Evolutionäre Algorithmen für TSP zu bauen. Lass die Algorithmen 30 Mal laufen, wobei die maximale Anzahl der Fitnessevaluationen auf 100.000 beschränkt sein soll. Welche der drei EA Konfigurationen hat die kürzesten Rundreisen finden können?
4. Kann Dein Algorithmus die kürzeste Rundreise finden? Erstelle bitte die Datei `bestresults.txt` mit 30 Rundreiselängen aus 30 Testläufen des besten Algorithmus (`python main.py -t`) und füge die Datei der Einsendung an. Deine Lösung wird mit den Lösungen anderer Gruppen verglichen und die Gruppe mit den kürzesten Rundreisen wird belohnt.

Um schnell starten zu können, ist die Hill Climber Implementierung im Code enthalten. Der HC kann gestartet werden mit

```
python main.py -c ./tspfiles
```

Der letzte Parameter verweist auf ein Verzeichnis mit TSP Probleminstanzen. Die TSP Probleminstanz für diese Übung ist `berlin52.tsp`. Mit

```
python main.py -h
```

kann die Hilfe-Seite angezeigt werden. Ein Funktionsrumpf eines genetischen Algorithmus ist in der Datei `algorithms_HC_EA.py` zu finden. Der EA kann gestartet werden mit

```
python main.py -g.
```