

Programmierprojekt: TSP Algorithmen

Preiß Philipp, Perner Manuel, Dickbauer Yanick

PS Entscheidungsunterstützung in Produktion und Logistik, WS 2016/17

January 31, 2017

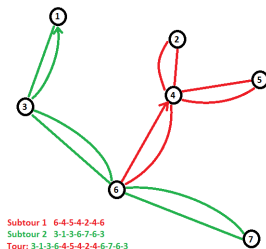
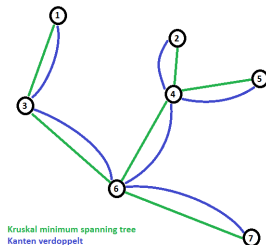
Aufgabenstellung

- Grafischer Vergleich verschiedener TSP Heuristiken:
 - MST Heuristik
 - Multi Fragment Heuristik
 - Nearest Neighbour Heuristik
 - Nearest/Cheapest Insertion Heuristik
- Optionale Anforderung:
 - Berechnung einer optimalen Tour durch Verwendung eines ILP Solvers

- MST Heuristik:
 - Länge durch MST Heuristik zwischen optimale Lösung und 2*optimale Lösung
- Ablauf MST Heuristiken:
 - 1 minimum spanning tree mit Algorithmus von Kruskal erstellen
 - 2 Kanten des minimum spanning tree verdoppeln
 - 3 Hierholzer Algorithmus
 - einzelne Subtouren erstellen
 - Eulertour bilden \Rightarrow Subtouren zusammenfügen
 - 4 Doppelte Punkte in Reihenfolge löschen

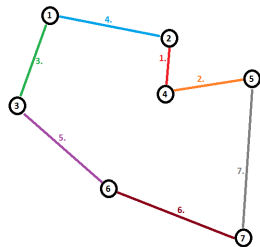
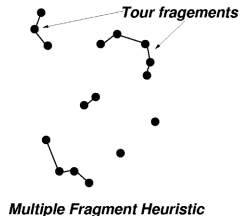
MST Heuristik

- Kruskalbaum und verdoppelte Kanten
- Hierholzer Algorithmus:
 - 1 einzelne Subtouren erstellen
 - 2 Eulertour bilden \Rightarrow Subtouren zusammenfügen
- Doppelte Punkte in Reihenfolge löschen
- Endtour: 3-1-6-4-5-2-7-3



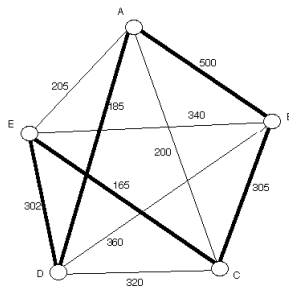
Multi Fragment Heuristik

- Ablauf Multi Fragment Heuristik
 - kleinste Kanten zwischen zwei Punkten verbinden, ohne ein Verzweigung zu erzeugen bis alle Punkten genau zwei Kanten besitzt
 - führt zu einem Pfad, bis er am ende verbunden wird
 - am anfang können mehrere kleine Pfade entstehen, die irgendwann miteinander verbunden werden
 - Endtour: 2-4-5-7-6-3-1-2



Nearest Neighbour

- Nearest Neighbour wurde so implementiert, dass das Programm die Lösungen für alle Startpunkte miteinander vergleicht und die beste auswählt.
- Grundsätzliche Vorgangsweise:
 - 1 Startknoten zufällig wählen
 - 2 Knoten mit geringster Entfernung zu aktuellem Knoten wird in Tour aufgenommen.
 - 3 Wiederhole Schritt 2 bis nur noch die Verbindung zum Startknoten übrig bleibt.
 - 4 Verbinde den zuletzt hinzugefügten

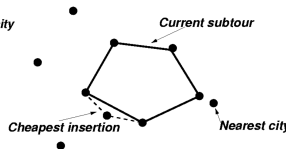
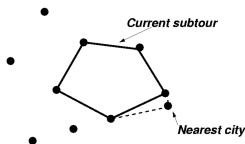


in node A

starting

Nearest/Cheapest Insertion

- Vorgehen bei beiden Heuristiken gleich.
- Unterschied liegt alleine im Insertionskriterium.
 - 1 Starte mit 2 Knoten.
 - 2 Einfügen des nächsten Knoten nach Auswahlkriterium:
 - Nearest \Rightarrow Knoten mit kürzester Entfernung
 - Cheapest \Rightarrow Knoten mit geringsten Einfügekosten
 - 3 Wiederhole Schritt 2 bis alle Knoten Teil der Tour sind.



Optimale Tour

- 4 Varianten implementiert:
 - ① Concorde - TSP Solver ¹
 - ② Gurobi - Kommerzieller ILP Solver ²
 - ③ COIN OR - Open Source ILP Solver ³
 - ④ Lpsolve ⁴
- Ermöglicht den Qualitätsvergleich der Heuristiken mit der Optimallösung

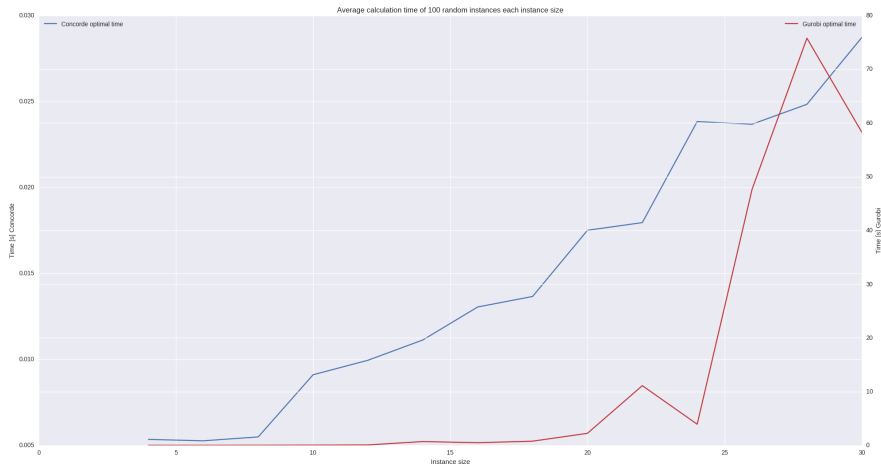
¹<http://www.math.uwaterloo.ca/tsp/concorde/>

²<http://www.gurobi.com/>

³<https://www.coin-or.org>

⁴<https://sourceforge.net/projects/lpsolve>

Geschwindigkeitsvergleich Concorde vs. Gurobi



Geschwindigkeitsvergleich Gurobi vs. COIN

