

OR-Competition

Die Aufgabe, die beste Tour zu finden, teilt sich in zwei Probleme, und zwar

1. Die geeigneten Knoten auswählen und
2. eine schnelle Tour zwischen diesen Knoten zu finden.

Die geeigneten Knoten auswählen:

Für diese Herangehensweise gibt es mehrere Ansätze, von denen keiner, ohne eine Route zu suchen, ein bestes Ergebnis liefern kann. Folgende Ansätze probiert der Algorithmus aus:

- Die Route wird aus den ersten Punkten berechnet, die in der Eingabe vorhanden waren. Dies dient (vor allem in der Entwicklungszeit) als Kontrollgröße.
- Ein Algorithmus startet beim wichtigsten* Knoten und betrachtet alle im Umkreis liegenden.
- Die wichtigsten* Punkte werden (unabhängig von ihrer Entfernung zueinander) betrachtet.
- Die leichtesten Punkte werden (unabhängig von ihrer Entfernung zueinander) betrachtet.
- Die profitabelsten Punkte werden (unabhängig von ihrer Entfernung zueinander) betrachtet.
- Zunächst werden die wichtigsten* zehn Prozent der Punkte (mindestens zehn, maximal 100) herausgesucht. Von jedem dieser wird die Umgebung betrachtet, bis die Kapazität gefüllt ist. Anschließend wird für jede dieser Sammlung der „Profit pro Radius“ berechnet. Der Bereich mit der höchsten dieser Kennziffern wird zurückgegeben.

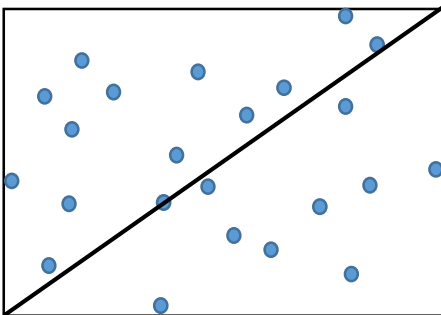
*Für einige Ansätze wird die „Wichtigkeit“ in Betracht gezogen. Dies ist eine Größe aus Profit und Gewicht.

In allen Fällen werden so lange Knoten zur Tour hinzugefügt, bis ein weiterer über die Kapazitätsgrenze gehen würde. Das „Rucksackproblem“ wird hier nicht beachtet, da es nur für sehr kleine Lösungen in Betracht kommt.

Eine Entscheidung, welche der Herangehensweisen letztendlich zur Tour führt, wird in Problem zwei entschieden.

Eine schnelle Tour zwischen den Knoten finden:

Da die Tour verhältnismäßig schnell berechnet wird, ist es vertretbar, sie für alle in Problem 1 gefundenen Herangehensweisen zu testen und somit die geeigneten Knoten erst nach Berechnung auszuwählen. Für jede dieser Selektionen wird das Folgende ausgeführt:



Ein Rechteck wird um alle Punkte gespannt und diagonal halbiert. Dies trennt die Punkte in einen Hin- und einen Rückweg.

Der Hinweg wird anfangs gebildet aus den Punkten, die sich unterhalb der Diagonale befinden, sortiert nach der Entfernung zur Ecke unten links. Gleiches passiert mit den oberen Punkten und der Ecke oben rechts. Anschließend werden diese beiden Touren miteinander verbunden.

Dann unterteilt ein weiterer Algorithmus mehrfach die Strecke in Teile von vier Punkten und testet, ob sie kürzer wäre, wenn man die mittleren beiden vertauschen würde. Ist dies der Fall, werden sie auch für die gesamte Strecke getauscht.

Für alle Strecken wird diejenige zurückgegeben, die den höchsten Quotienten aus Profit und gefahrener Strecke ergibt.