Euklidisches TSP

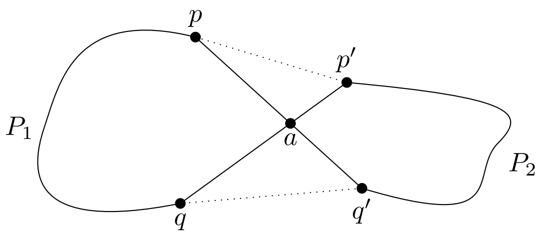
Definition. *Ein symmetrisches TSP auf Knoten* $1, \ldots, n$ *mit Kantengewichten* c_{ij}, $1 \leq i, j \leq n$ *heißt* Euklidisch, *falls jedem Knoten* i *ein Punkt* $p_i = (x_i, y_i) \in \mathbb{R}^2$ *zugeordnet werden kann, so dass die Kantengewichte den Euklidischen Distanzen der den Endknoten zugeordneten Punkten entsprechen, d.h.*

$$c_{i,j} = d(p_i, p_j) := \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad \triangle$$

Weil die Euklidischen Distanzen in der Ebene eine Metrik sind, ist jedes Euklidische TSP auch ein metrisches TSP. Die Gütegarantien für die früher besprochenen Heuristiken gelten also auch hier. Bei der Auswahl einer Starttour für Insertion-Heuristiken kann man sich jetzt die zusätzliche geometrische Information zunutze machen.

Lemma. *Sei ein Euklidisches TSP durch Punkte* $p_i = (x_i, y_i) \in \mathbb{R}^2$, $1 \leq i, j \leq n$ *gegeben. Dann gibt es in einer optimalen Tour keine Überschneidung.* $\triangle$

Beweis. Sei eine Tour $T = (P_1, p, q', P_2, p', q)$ (mit geeigneten Pfaden P_1, P_2) gegeben, in der sich die Kanten pq' und $p'q$ im Punkt a überschneiden. Der Einfachheit halber nehmen wir an, dass a mit keinem der anderen Punkte p, p', q, q' zusammenfällt; in einem solchen Fall verläuft der Beweis aber analog.

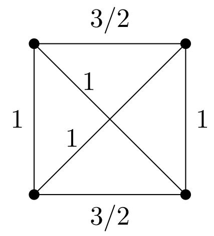


Dann ist die Tour $T' = (P_1, p, p', P_2^-, q', q)$ (wobei P^- der Pfad P, in umgekehrter Kantenreihenfolge durchlaufen, sei) eine Tour und es gilt

$$\begin{aligned}
c(T') - c(T) &= c_{pp'} - c_{pq'} + c_{q'q} - c_{p'q} \\
&= d(p,p') - d(p,a) - d(a,q') + d(q',q) - d(p',a) - d(a,q) \\
&= d(p,p') - \underbrace{\big(d(p,a) + d(a,p')\big)}_{>d(p,p')} + d(q',q) - \underbrace{\big(d(q',a) + d(a,q)\big)}_{>d(q',q)} \\
&< 0
\end{aligned}$$

d. h. T kann nicht optimal gewesen sein. $\square$

Bemerkung. Für metrische, aber nicht-Euklidische TSP (mit in die Ebene eingebetteten Knoten) gilt das Lemma nicht, wie folgendes Beispiel zeigt:



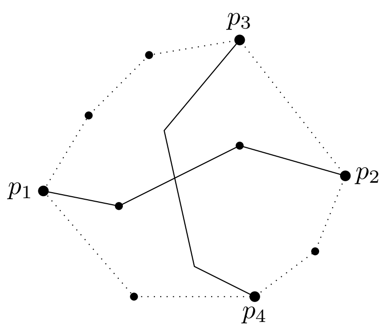
$\triangle$

Satz. *Sei ein Euklidisches TSP durch Punkte $p_i = (x_i, y_i) \in \mathbb{R}^2$, $1 \leq i, j \leq n$ gegeben. Dann werden in einer optimalen Tour die Punkte in der konvexen Hülle in der gleichen Reihenfolge durchlaufen wie in der konvexen Hülle; genauer: bezeichnet $(p_1, \ldots, p_k)$ den Pfad entlang der konvexen Hülle (in einer der beiden Richtungen), dann ist jede optimale Tour gegeben durch*

$$(P_1, p_1, P_2, p_2, \ldots, P_{k-1}, p_k)$$

mit geeigneten Pfaden $P_1, \ldots, P_{k-1}$ durch die übrigen Punkte. $\triangle$

Beweis. (Skizze) Angenommen, in einer optimalen Tour T würden die Punkte der konvexen Hülle nicht in der gleichen Reihenfolge durchlaufen wie dort. Dann gibt es vier Punkte $p_1, \ldots, p_4$, so dass sich die Teilpfade von T von p_1 nach p_2 und von p_3 nach p_4 überkreuzen (s. Skizze), im Widerspruch zum Lemma.



$\square$

Das lässt erwarten, dass eine Insertion-Heuristik mit der konvexen Hülle als Startkreis im Allgemeinen recht gute Touren liefert.

Algorithmus (Convex Hull Insertion).
Eingabe: Symm. TSP als Punkte $p_i = (x_i, y_i) \in \mathbb{R}^2$, $1 \leq i, j \leq n$
Ausgabe: Tour durch $p_1, \ldots, p_n$

1. Berechne die konvexe Hülle der Punkte $p_1, \ldots, p_n$ und setze als Startkreis C die Knoten i mit p_i in der konvexen Hülle in der entsprechenden Reihenfolge.

2. Ist $V \setminus V(C) = \emptyset$, STOP. C ist eine Tour.

3. Für jedes $k \in V \setminus V(C)$ finde $ij \in C$, so dass $c_{ik} + c_{kj} - c_{ij}$ minimal ist und setze $P_k := (i, k, j)$. Anschließend finde (i^*, k^*, j^*), so dass

$$\frac{c_{i^*k^*} + c_{k^*j^*}}{c_{i^*j^*}} = \min\Big\{\frac{c_{ik} + c_{kj}}{c_{ij}} \mid k \in V \setminus V(C), P_k = (i, k, j)\Big\}.$$

4. Ersetze i^*j^* in C durch P_k und gehe zu Schritt 2. $\triangle$

Statt "cheapest insertion" wie in Schritt 3 kann man auch andere Kriterien zum Einfügen neuer Punkte benutzen, z. B. "maximum angle" (wähle i^*, j^*, k^* so dass $i^*j^* \in C$, $k \in V \setminus V(C)$ und der (kleinere) Winkel zwischen den Strecken i^*k^* und k^*j^* maximal ist) oder "ratio times difference" (wähle i^*, j^*, k^* so dass $i^*j^* \in C$, $k \in V \setminus V(C)$ und das Produkt

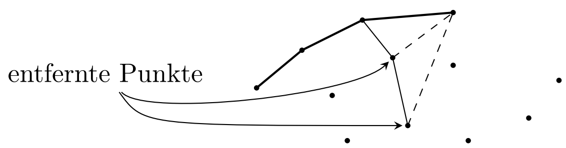
$$(c_{i^*k^*} + c_{k^*j^*} - c_{i^*j^*}) \cdot \frac{c_{i^*k^*} + c_{k^*j^*}}{c_{i^*j^*}}$$

minimal ist).

Zur Berechnung der konvexen Hülle von n Punkten gibt es verschiedene Möglichkeiten. Die triviale Variante (für jedes Paar p,q von Punkten überprüfe ob alle weiteren Punkte auf der gleichen Seite der durch p und q definierten Gerade liegen) hat Laufzeit $O(n^3)$. Man kann aber relativ einfach bessere Verfahren angeben, so wie „Graham's scan“ (Ron Graham, 1972):

1. Sortiere die Punkte nach x-Koordinate aufsteigend.

2. Berechne die *obere konvexe Hülle*: Gehe von links nach rechts durch alle Punkte, baue dabei eine Liste der Punkte auf der oberen konvexen Hülle auf und entferne dabei jeweils Punkte, wenn in einem Schritt kein „Rechtsknick“ gemacht wird (s. Skizze).



3. Berechne die *untere konvexe Hülle* auf analoge Weise.

Man kann zeigen, dass man damit insgesamt nur $O(n \log n)$ Schritte braucht, um beide Teile der konvexen Hülle zu berechnen und das ist bestmöglich. Es gibt auch output-sensitive Algorithmen, die in $O(k \cdot n)$ laufen, wobei k die Anzahl der Punkte auf der konvexen Hülle ist, und sehr effiziente Divide-and-Conquer-Algorithmen. Für Details siehe z. B. de Berg et al. (2000).

Literaturverzeichnis

M. de Berg, M. van Krefeld, M. Overmars, and O. Schwarzkopf. *Computational Geometry – Algorithms and Applications*. Springer Verlag, 2nd edition, 2000.