Voronoi- / Delaunay-Diagramm in 3D

Allgemeine Lage = keine 5 Pkte auf Kugel

V.-Region = konvexes (evtl. unbeschränktes) Polyeder

Bisektor = Ebene (senkrecht zu $\overline{pq}$)

$D(S)$ = Menge von Tetraedern

4 Pkte aus S bilden Delaunay-Tetraeder $\Leftrightarrow$ die Kugel durch diese 4 Pkte enthält keinen weiteren Pkt aus S.

Jeder V.-Knoten / -Kante ist inzident zu genau 4/3 V.-Regionen.

Achtung: Die max-min-Winkel-Eigenschaft der Delaunay-Triangulierung gilt nicht mehr in 3D! [Eberschul Diss]

Komplexität: $V(S)$ in 3D hat $O(n^2)$ viele Komponenten

Es gibt Mengen S, wo diese Schranke angenommen wird! [unser Buch]

Anwendungen

Bezeichnung:

Sei S = Menge von Pkten, x ein Pkt

$nns(x) = \arg\min_{q \in S} d(x,q)$ ("nächster Nachbar")

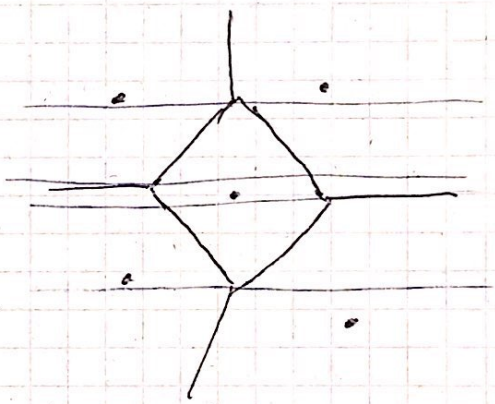
Das Post-Office-Problem:

[Klein S. 249
unser Buch
S. 123

Eg. Menge S von Sites (post offices) und Query-Pkt q (eigener Standort).
gesucht $nns(q)$ (= nächstes post office).

Lsg mittels $V(S)$:

Precompute $V(S)$, zerlege Ebene in "Slabs" durch eine hor. Gerade durch je einen V.-Knoten.
Sortiere Slabs bzgl. y, sortiere Trapeze innerhalb eines Slabs bzgl. x.



Laufzeit: $O(\log n)$, Platz: $O(n^2)$

Diese Lsg. ist ein Bsp. für den allgemeinen "Locus Approach": zerlege den Raum in Zellen gleicher Antwort

Bestimmung der nächsten Nachbarn:

Lemma (closest pair = Delaunay-Kante)

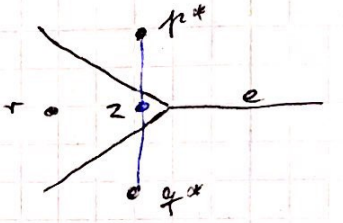
Sei S = Ptmenge, $P \dot\cup Q = S$ Partitionierung derselben.

Seien $p^* \in P$, $q^* \in Q$ so daß $d(p^*, q^*) = \min_{p \in P,\, q \in Q} d(p,q)$.

Dann gilt:

$R(p^*)$ und $R(q^*)$ in $\mathcal{V}(S)$ haben eine gemeinsame Kante e, und diese wird von $\overline{p^* q^*}$ geschnitten.

Bew.:



kon.: $R(p^*)$ und $R(q^*)$ sind nicht benachbart, oder $\overline{p^* q^*}$ schneidet nicht e.

$\Rightarrow \exists\, z \in \overline{p^* q^*},\ r \in S: z \in R(r)$;

o.BdA $r \in Q$

$$\Rightarrow d(p^*, r) \overset{\text{Dreiecksungl.}}{\leq} d(p^*, z) + d(z, r) \leq d(p^*, z) + d(z, q^*) = d(p^*, q^*)$$

$$\leq d(p^*, r)$$

(Min.-Eig. von p^*, q^*)

($d(z,r) < d(z,q^*)$ da $z \in R(r)$)

$\Rightarrow \lightning$!

~~Lemma~~ Corollar:

Sei $p = nn_S(q)$, $p, q \in S$.
Dann ist $\overline{pq}$ eine Kante in $D(S)$.

Beweis:
Vorigen Satz auf $S \setminus \{q\}$ und $\{q\}$ anwenden.

Corollar:
Sei $NNG(S)$ der "nearest neighbor graph" von S.
Dann gilt

$$NNG(S) \subseteq D(S).$$

Let $NNG(S)$ be the "nearest neighbor graph" over S,
i.e., for each $p \in S$ there is an edge
$(p, nn_S(p)) \in E_S$.

→ Suggest simple algo to construct NNG, i.e., store NN's for all $p \in S$.

Der minimum spanning tree:

Def.: MST(S) := Graph, der alle $p \in S$ verbindet, und dabei minimale Gesamtlänge hat.

Beispiel: Anzahl Orte durch Glasfasernetz verbinden.
(ohne zusätzliche Knotenpkte)

Vorgehensweise: Algo von Kruskal

Input ~~$G = (S, E)$~~, $S = \{p_1, \dots, p_n\}$

Output ~~MST(S) $\subseteq G$~~

$\mathcal{T} := \{T_1, \dots, T_m\}$ = Wald von Bäumen

initial $T_i := \{p_i\}$

Set $E := V \times V$ (all possible edges), sort E by length

~~$Q :=$ Menge aller Kanten E, sortiert nach Länge~~ / Sort E

~~Schleife über alle $e \in E$:~~ loop over all $e \in E$

if e verläuft zwischen zwei $T_i, T_j \in \mathcal{T}$:

verbinde $T_i, T_j \to T'$ (merge), replace T_i, T_j by T'

else:

verwerfe e (skip)

Laufzeit: $O(|E| \log |E|)$

$\Rightarrow$ MST über S benötigt damit maximal $O(n^2 \log n)$

Lemma:

$$MST(S) \subseteq D(S)$$

Bew.:

Wenn Kruskal T_i, T_j verschmilzt durch Kante e, dann ist e die kürzeste Kante zwischen $p \in T_i$ $q \in T_j$, d.h. $p = nn(q)$ aber $q = nn(p)$

$\Rightarrow$ e muß Delaunay-Kante sein.

↑ Lemma "davor gezeigt"

Corollar:

$MST(S)$ läßt sich in Zeit $O(n \log n)$ berechnen.

Bew.:

Starte Kruskal mit $D(S)$; $D(S)$ läßt sich in Zeit $O(n \log n)$ berechnen. , $|E| \in O(n)$

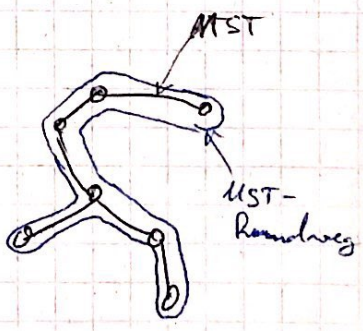
Anwendung auf das TSP:

Bezeichne $TST(S)$ = traveling salesman tour (= cycle)

Theorem:

Der Rundweg um den $MST(S) \leq 2 \cdot |TST(S)|$.

Heißt auch 2-approximation of the TST.



Beweis:

Sei e eine Kante des TST; sei $W = TST \setminus e$.

W ist ein spanning tree $\Rightarrow |MST| \leq |W| \leq |TST|$

$\Rightarrow \underbrace{2 \cdot |MST|}_{\text{Rundweg aus dem Lemma}} \leq 2 \cdot |TST|$

Anwendung: Scheduling Record Access

Bsp.: Datenbank auf Festplatte

Vereinfachung: 1 Record pro Sektor

Bsp.: n random Requests, Request = Record^T = (Spurnr. (track no.), Sektornr. (sector no.))

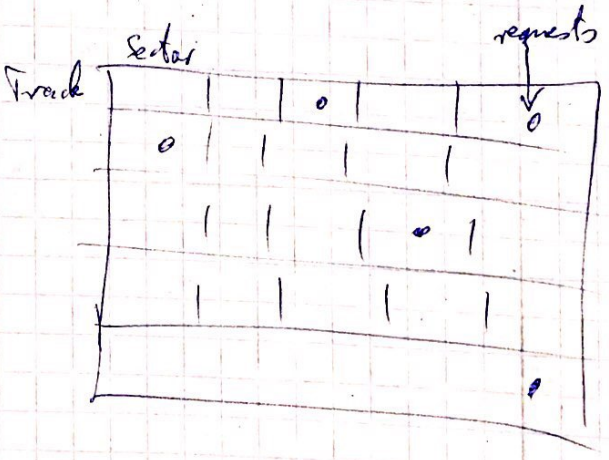
Aufgabe: alle n Requests in min. Zeit abarbeiten

Lsg.:

Abstand zwischen zwei Records = $\| r_1 - r_2 \|_1 = |t_1 - t_2| + |s_1 - s_2|$.

Gesucht ist TST über die n Records bzgl. $\|\cdot\|_1$

$\Rightarrow$ konstruiere das V-Diagramm (bzw. den D-Graph) bzgl. $\|\cdot\|_1$ über n Sites (= Records), und daraus den MST, und daraus 2-approximative Tour.



Größter leerer Kreis:

Bsp.: Gebiet als konvexes Polygon gegeben, darin n Störquellen (Flughafen, Kraftwerke, etc.)
Aufgabe: den Ort für einen neuen Wohnsitz finden, der am weitesten von allen Störquellen entfernt.

Spez.: konvexes Polygon $A \subseteq \mathbb{R}^2$, Sites $S = \{p_1, \dots, p_n\}$.
Gesucht: $x \in A$, so daß es maximalen leeren Kreis $C(x)$ gibt.

Lemma:
Sei $C(x)$ der größte Kreis, der kein p_i enthält, mit $x \in A$. Dann gilt
a) x ist ein Voronoi-Knoten von $V(S)$; oder,
b) x liegt auf dem Schnittpunkt einer V.-Kante mit A; od.
c) x liegt auf einer Ecke von A.

Satz:
Der größte leere Kreis läßt sich in $O(m+n)$ bestimmen, wenn $V(S)$ schon vorhanden ist; wobei $m = \#$ Ecken von A und $n = |S|$.

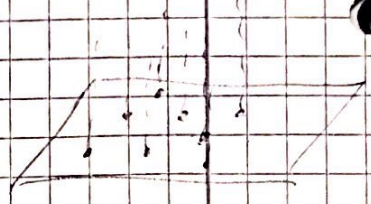
Bew.:
Inspiziere die $O(n)$ vielen V.-Knoten; die $O(n)$ vielen dann wandert man auf A entlang und hangelt sich so von V.-Region zu V.-Region, und kann die Ecken von A gleich mit erschlagen.

Terrain-Visualisierung:

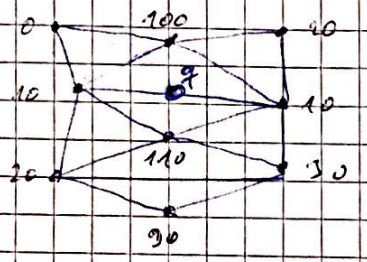
[Blau-gelbes Buch]

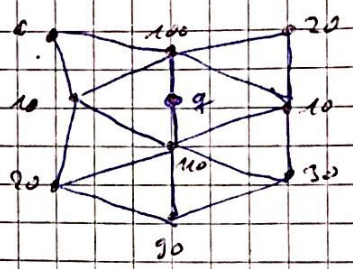
Geg.: Pkte in der Ebene und Höhe darüber

Frage: welche Pkte soll man zu Dreiecken verbinden?



Bsp.:





$h(q) = 10$!

$h(q) = 105$

Fazit: Dreiecke mit spitzen Winkeln sind schlecht

Also: wähle gleich die Triangulierung mit dem besten Winkelkriterium

Ist data-independent approach, da Höhe über den Datenpkten nicht eingeht. Inzwischen viele data-dependent approaches untersucht.