

Voronoi Diagramme

Elmar Langetepe
University of Bonn

All nearest neighbors! Untereinander!

All nearest neighbors! Untereinander!

Nächster Nachbar liegt in Nachbarzelle!

All nearest neighbors! Untereinander!

Nächster Nachbar liegt in Nachbarzelle!

Lemma 5.7 Sei $S = P \cup Q$ eine Zerlegung der endlichen Punktmenge S in zwei disjunkte, nicht-leere Teilmengen P und Q . Seien $p_0 \in P$ und $q_0 \in Q$ so gewählt, dass

$$|p_0q_0| = \min_{p \in P, q \in Q} |pq|$$

gilt. Dann haben die Regionen von p_0 und q_0 im Voronoi-Diagramm $V(S)$ eine gemeinsame Kante.

All nearest neighbors! Untereinander!

Nächster Nachbar liegt in Nachbarzelle!

Lemma 5.7 Sei $S = P \cup Q$ eine Zerlegung der endlichen Punktmenge S in zwei disjunkte, nicht-leere Teilmengen P und Q . Seien $p_0 \in P$ und $q_0 \in Q$ so gewählt, dass

$$|p_0q_0| = \min_{p \in P, q \in Q} |pq|$$

gilt. Dann haben die Regionen von p_0 und q_0 im Voronoi-Diagramm $V(S)$ eine gemeinsame Kante.

Beweis: Lokal, Dreiecksungleichung

Weitere Anwendungen: All nearest neighbors!

Weitere Anwendungen: All nearest neighbors!

Korollar 5.8 Jeder nächste Nachbar von p in S sitzt im Voronoi-Diagramm in einer Nachbarzelle, d.h. in einer Voronoi-Region, die mit $VR(p, S)$ eine gemeinsame Kante besitzt.

Weitere Anwendungen: All nearest neighbors!

Korollar 5.8 Jeder nächste Nachbar von p in S sitzt im Voronoi-Diagramm in einer Nachbarzelle, d.h. in einer Voronoi-Region, die mit $VR(p, S)$ eine gemeinsame Kante besitzt.

Beweis: $\{p\}$ und $S \setminus \{p\}$ als Mengen

Weitere Anwendungen: All nearest neighbors!

Korollar 5.8 Jeder nächste Nachbar von p in S sitzt im Voronoi-Diagramm in einer Nachbarzelle, d.h. in einer Voronoi-Region, die mit $VR(p, S)$ eine gemeinsame Kante besitzt.

Beweis: $\{p\}$ und $S \setminus \{p\}$ als Mengen

Theorem 5.9 Ist das Voronoi-Diagramm $V(S)$ vorhanden, kann in Zeit $O(n)$ für alle $p \in S$ der nächste Nachbar bestimmt werden.

Weitere Anwendungen: All nearest neighbors!

Korollar 5.8 Jeder nächste Nachbar von p in S sitzt im Voronoi-Diagramm in einer Nachbarzelle, d.h. in einer Voronoi-Region, die mit $VR(p, S)$ eine gemeinsame Kante besitzt.

Beweis: $\{p\}$ und $S \setminus \{p\}$ als Mengen

Theorem 5.9 Ist das Voronoi-Diagramm $V(S)$ vorhanden, kann in Zeit $O(n)$ für alle $p \in S$ der nächste Nachbar bestimmt werden.

Beweis:

Weitere Anwendungen: All nearest neighbors!

Korollar 5.8 Jeder nächste Nachbar von p in S sitzt im Voronoi-Diagramm in einer Nachbarzelle, d.h. in einer Voronoi-Region, die mit $VR(p, S)$ eine gemeinsame Kante besitzt.

Beweis: $\{p\}$ und $S \setminus \{p\}$ als Mengen

Theorem 5.9 Ist das Voronoi-Diagramm $V(S)$ vorhanden, kann in Zeit $O(n)$ für alle $p \in S$ der nächste Nachbar bestimmt werden.

Beweis: Durchlaufen des Diagramms!

Weitere Anwendungen: All nearest neighbors!

Korollar 5.8 Jeder nächste Nachbar von p in S sitzt im Voronoi-Diagramm in einer Nachbarzelle, d.h. in einer Voronoi-Region, die mit $VR(p, S)$ eine gemeinsame Kante besitzt.

Beweis: $\{p\}$ und $S \setminus \{p\}$ als Mengen

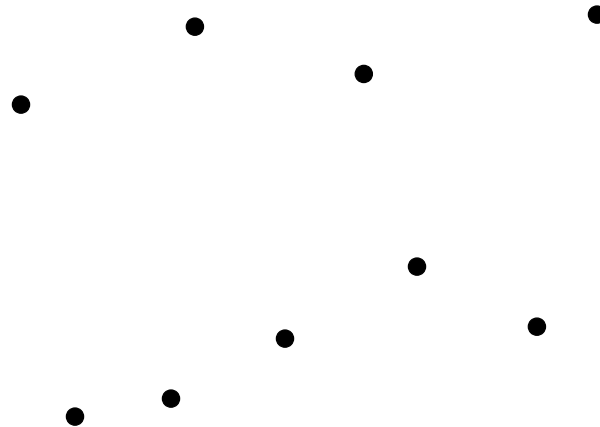
Theorem 5.9 Ist das Voronoi-Diagramm $V(S)$ vorhanden, kann in Zeit $O(n)$ für alle $p \in S$ der nächste Nachbar bestimmt werden.

Beweis: Durchlaufen des Diagramms! Jede Kante zweimal besuchen!

Weitere Anwendungen: Minimum Spanning Tree

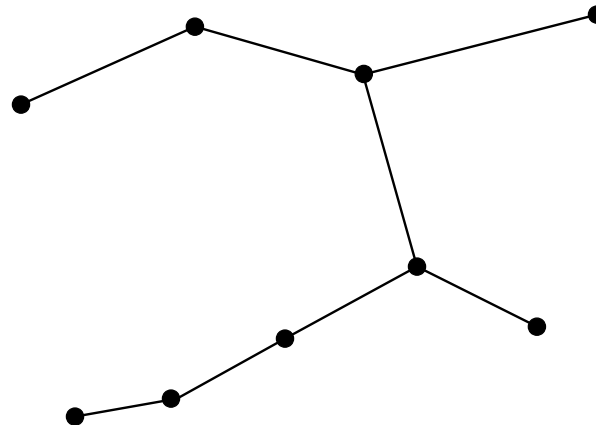
Weitere Anwendungen: Minimum Spanning Tree

- Motiv.: Punktmenge geg., kleinster Graph, der alle Punkte enthält



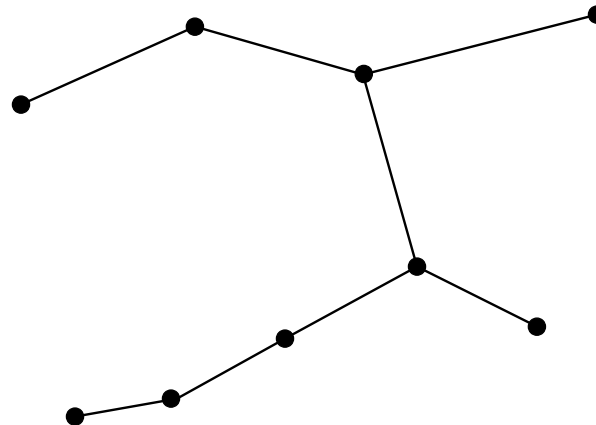
Weitere Anwendungen: Minimum Spanning Tree

- Motiv.: Punktmenge geg., kleinster Graph, der alle Punkte enthält
- Baum (klar): Minimum Spanning Tree



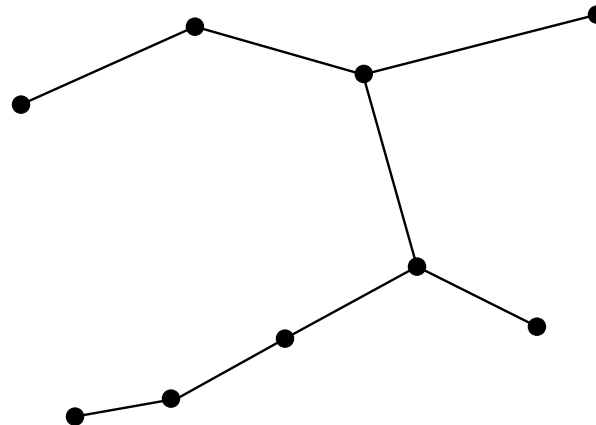
Weitere Anwendungen: Minimum Spanning Tree

- Motiv.: Punktmenge geg., kleinster Graph, der alle Punkte enthält
- Baum (klar): Minimum Spanning Tree
- $G = (V, E)$, eine Zs.-hangskomp., nur Kanten aus E verwenden



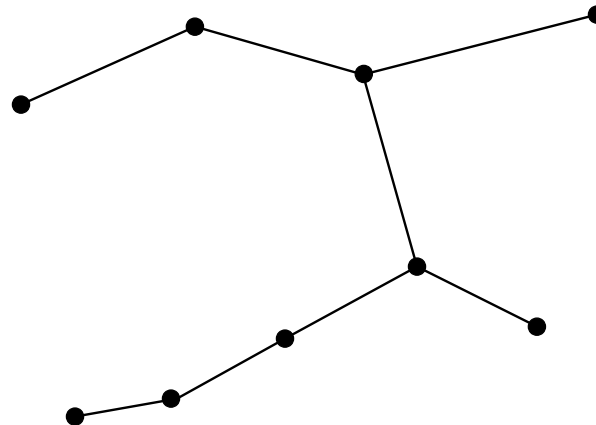
Weitere Anwendungen: Minimum Spanning Tree

- Motiv.: Punktmenge geg., kleinster Graph, der alle Punkte enthält
- Baum (klar): Minimum Spanning Tree
- $G = (V, E)$, eine Zs.-hangskomp., nur Kanten aus E verwenden
- Kruskal: $O(|E| \log |E|)$ Laufzeit



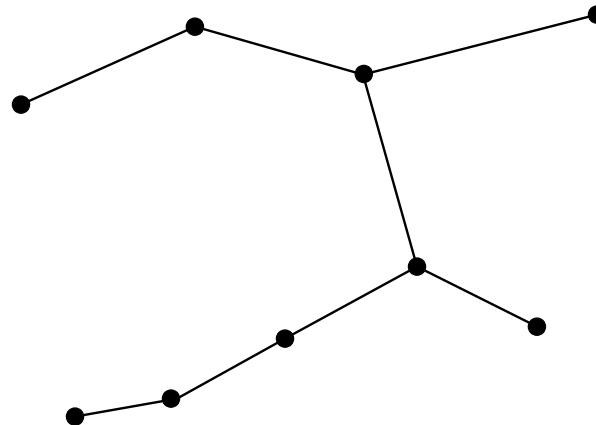
Weitere Anwendungen: Minimum Spanning Tree

- Motiv.: Punktmenge geg., kleinster Graph, der alle Punkte enthält
- Baum (klar): Minimum Spanning Tree
- $G = (V, E)$, eine Zs.-hangskomp., nur Kanten aus E verwenden
- Kruskal: $O(|E| \log |E|)$ Laufzeit
- Punktmenge geg.: Vollst. Graph mit n^2 Kanten: $O(n^2 \log n)$



Weitere Anwendungen: Minimum Spanning Tree

- Motiv.: Punktmenge geg., kleinster Graph, der alle Punkte enthält
- Baum (klar): Minimum Spanning Tree
- $G = (V, E)$, eine Zs.-hangskomp., nur Kanten aus E verwenden
- Kruskal: $O(|E| \log |E|)$ Laufzeit
- Punktmenge geg.: Vollst. Graph mit n^2 Kanten: $O(n^2 \log n)$
- Anzahl Kanten auf $O(n)$ beschränken



Algorithmus von Kruskal

Algorithmus von Kruskal

- Verwalte Wald von Bäumen

Algorithmus von Kruskal

- Verwalte Wald von Bäumen
- Sukzessive Kante mit kürzester Länge auswählen

Algorithmus von Kruskal

- Verwalte Wald von Bäumen
- Sukzessive Kante mit kürzester Länge auswählen
- Falls Teilbäume zusammenwachsen, einfügen

Algorithmus von Kruskal

- Verwalte Wald von Bäumen
- Sukzessive Kante mit kürzester Länge auswählen
- Falls Teilbäume zusammenwachsen, einfügen
- Kanten nach Länge sortieren: $O(n^2 \log n)$

Algorithmus von Kruskal

- Verwalte Wald von Bäumen
- Sukzessive Kante mit kürzester Länge auswählen
- Falls Teilbäume zusammenwachsen, einfügen
- Kanten nach Länge sortieren: $O(n^2 \log n)$

Flashanimation

Anwendung MST: Traveling Salesman Problem

Anwendung MST: Traveling Salesman Problem

- n Orte, kürzeste Rundtour, die alle Orte besucht: $TSP - Opt$

Anwendung MST: Traveling Salesman Problem

- n Orte, kürzeste Rundtour, die alle Orte besucht: $TSP - Opt$
- MST, kleinster Baum, der Orte verbindet

Anwendung MST: Traveling Salesman Problem

- n Orte, kürzeste Rundtour, die alle Orte besucht: $TSP - Opt$
- MST, kleinster Baum, der Orte verbindet
- Beweis: $|MST| \leq |TSP-Opt|$

Anwendung MST: Traveling Salesman Problem

- n Orte, kürzeste Rundtour, die alle Orte besucht: $TSP - Opt$
- MST, kleinster Baum, der Orte verbindet
- Beweis: $|MST| \leq |TSP-Opt|$
- Umrunde den TSP:

Anwendung MST: Traveling Salesman Problem

- n Orte, kürzeste Rundtour, die alle Orte besucht: $TSP - Opt$
- MST, kleinster Baum, der Orte verbindet
- Beweis: $|MST| \leq |TSP-Opt|$
- Umrunde den TSP: $2|MST| \leq 2|TSP-Opt|$

Anwendung MST: Traveling Salesman Problem

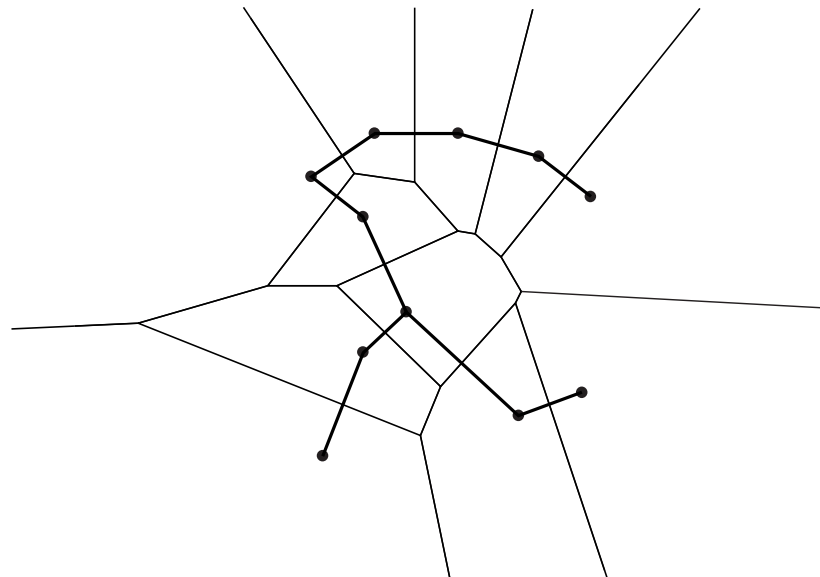
- n Orte, kürzeste Rundtour, die alle Orte besucht: $TSP - Opt$
- MST, kleinster Baum, der Orte verbindet
- Beweis: $|MST| \leq |TSP-Opt|$
- Umrunde den TSP: $2|MST| \leq 2|TSP-Opt|$
- Einfache 2-Approximation

Anwendung MST: Traveling Salesman Problem

- n Orte, kürzeste Rundtour, die alle Orte besucht: $TSP - Opt$
- MST, kleinster Baum, der Orte verbindet
- Beweis: $|MST| \leq |TSP-Opt|$
- Umrunde den TSP: $2|MST| \leq 2|TSP-Opt|$
- Einfache 2-Approximation

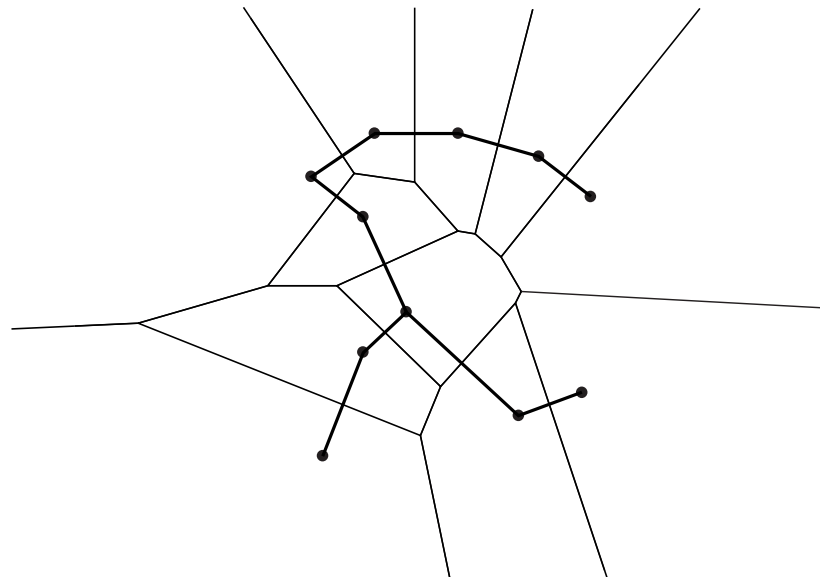
Theorem 5.12 Der Rundweg um einen MST ist weniger als doppelt so lang wie eine optimale TSP Tour.

MST berechnen: Vorteil Voronoi-Diagramm



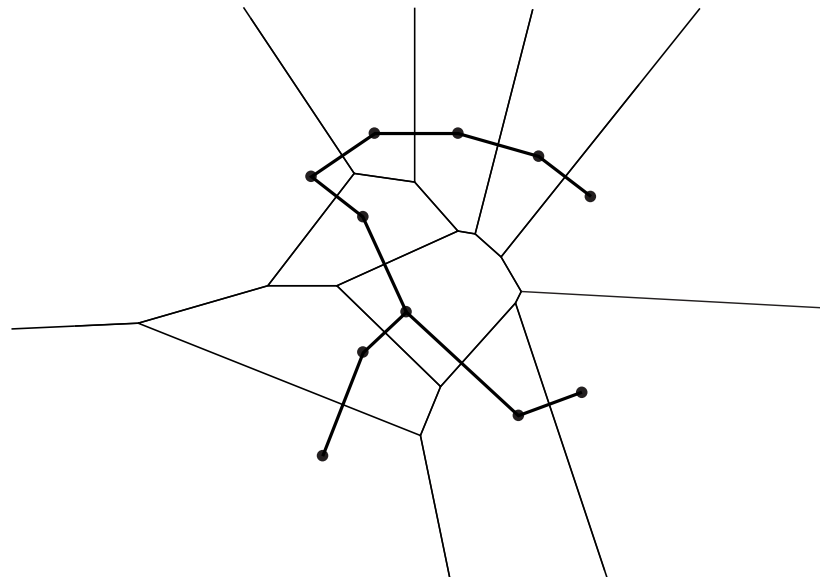
MST berechnen: Vorteil Voronoi-Diagramm

- Annahme: Voronoi Diagramm gegeben



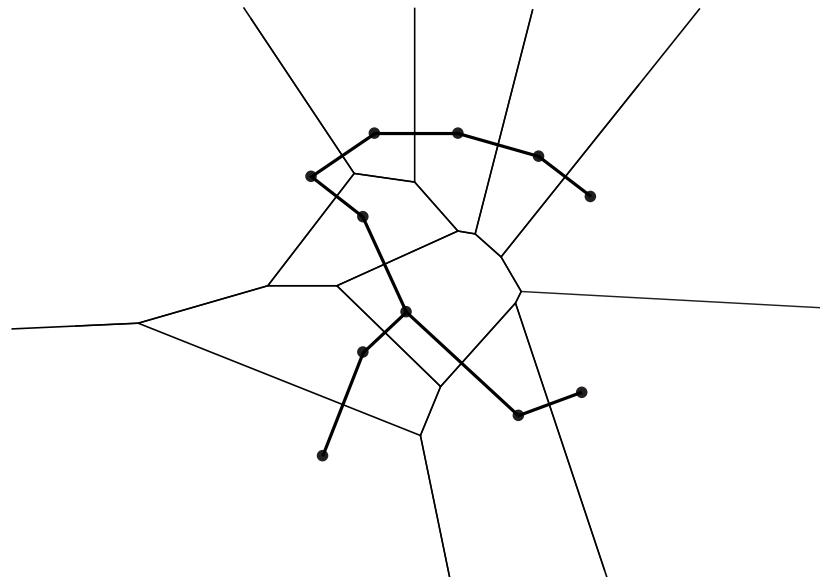
MST berechnen: Vorteil Voronoi-Diagramm

- Annahme: Voronoi Diagramm gegeben
- Lemma 5.7 Nächster Nachbar liegt in Nachbarzelle!



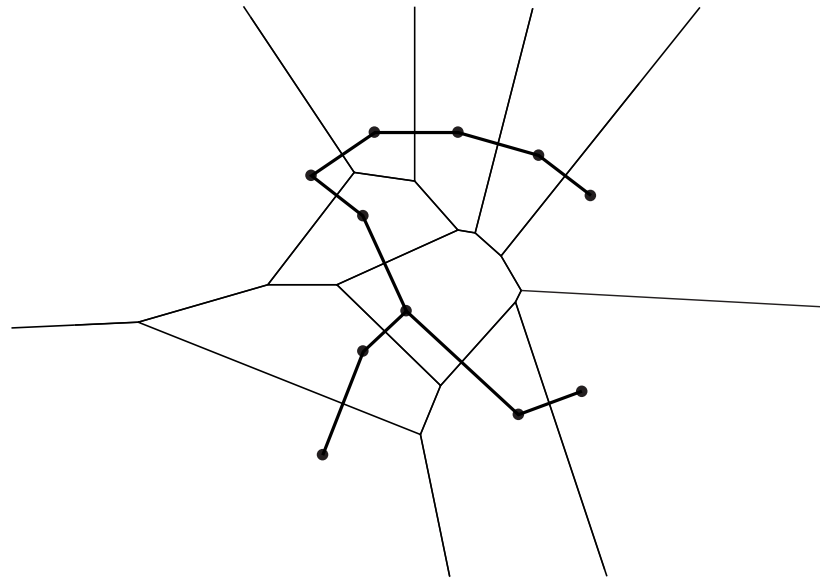
MST berechnen: Vorteil Voronoi-Diagramm

- Annahme: Voronoi Diagramm gegeben
- Lemma 5.7 Nächster Nachbar liegt in Nachbarzelle!
- Kantenmenge: Kanten zwischen Voronoi-Nachbarn



MST berechnen: Vorteil Voronoi-Diagramm

- Annahme: Voronoi Diagramm gegeben
- Lemma 5.7 Nächster Nachbar liegt in Nachbarzelle!
- Kantenmenge: Kanten zwischen Voronoi-Nachbarn
- Nur $O(n)$ viele Kanten: Sortieren nach Länge

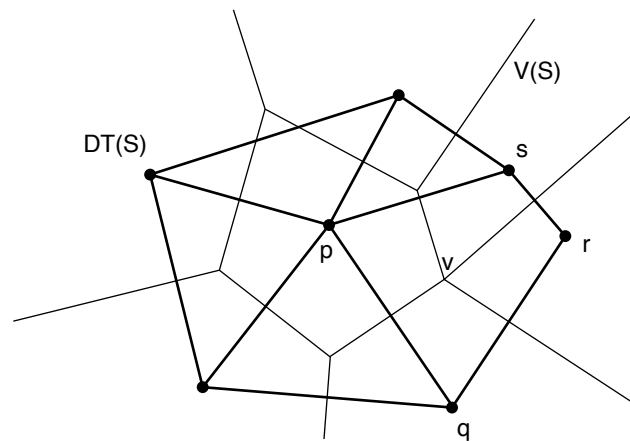


MST berechnen: Vorteil Voronoi-Diagramm

MST berechnen: Vorteil Voronoi-Diagramm

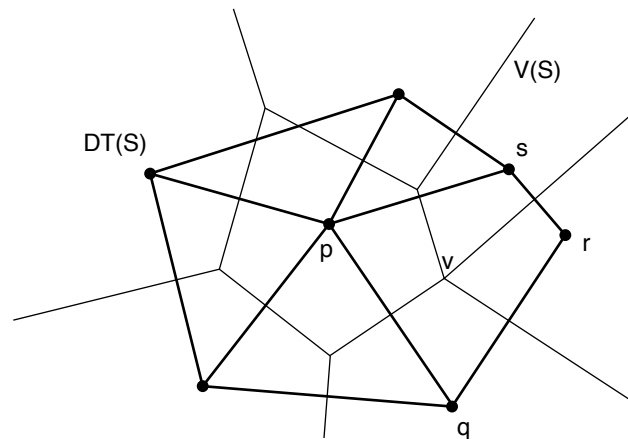
Theorem 5.11 Falls das Voronoi Diagramm $V(S)$ für n -elementige Punktmenge S gegeben ist, kann der MST in $O(n \log n)$ berechnet werden.

Dualer Graph von $V(S)$



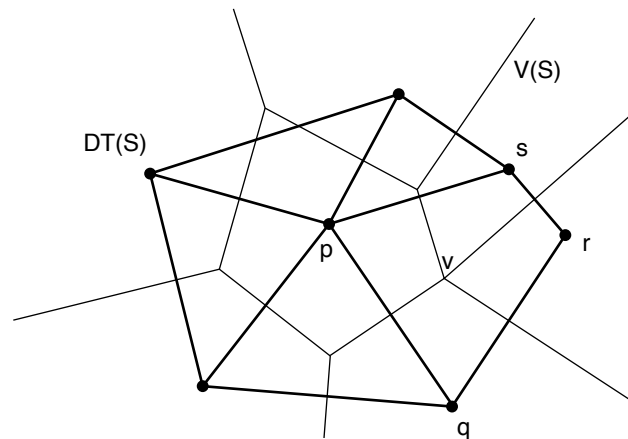
Dualer Graph von $V(S)$

- $V(S)$ gegeben: $VR(p, S)$ und $VR(q, S)$ gemeinsame Kante $\implies pq$ heißt *Delaunay* Kante



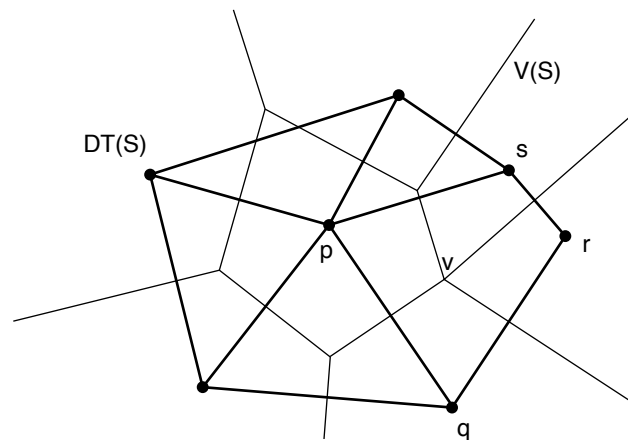
Dualer Graph von $V(S)$

- $V(S)$ gegeben: $VR(p, S)$ und $VR(q, S)$ gemeinsame Kante $\implies pq$ heißt *Delaunay* Kante
- Menge aller Delaunay Kanten: Delaunay Zerlegung $DT(S)$



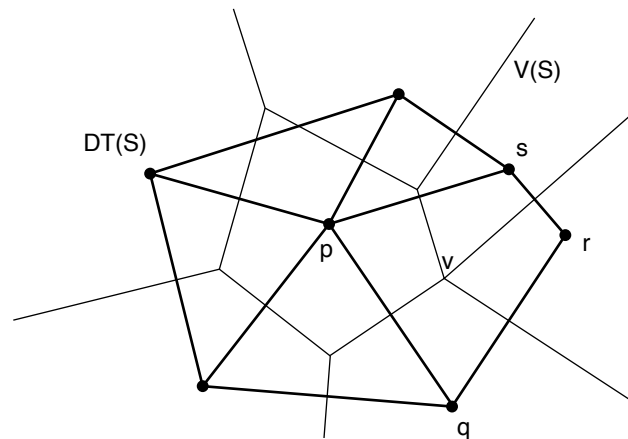
Dualer Graph von $V(S)$

- $V(S)$ gegeben: $VR(p, S)$ und $VR(q, S)$ gemeinsame Kante $\implies pq$ heißt *Delaunay* Kante
- Menge aller Delaunay Kanten: Delaunay Zerlegung $DT(S)$
- Geometrische Realisation des Dualen Graphen von $V(S)$

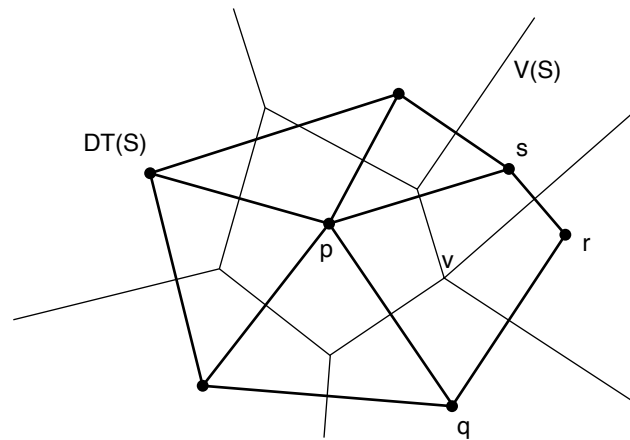


Dualer Graph von $V(S)$

- $V(S)$ gegeben: $VR(p, S)$ und $VR(q, S)$ gemeinsame Kante $\implies pq$ heißt *Delaunay* Kante
- Menge aller Delaunay Kanten: Delaunay Zerlegung $DT(S)$
- Geometrische Realisation des Dualen Graphen von $V(S)$
- $MST(S)$, $CH(S)$ ist Teilmenge von $DT(S)$

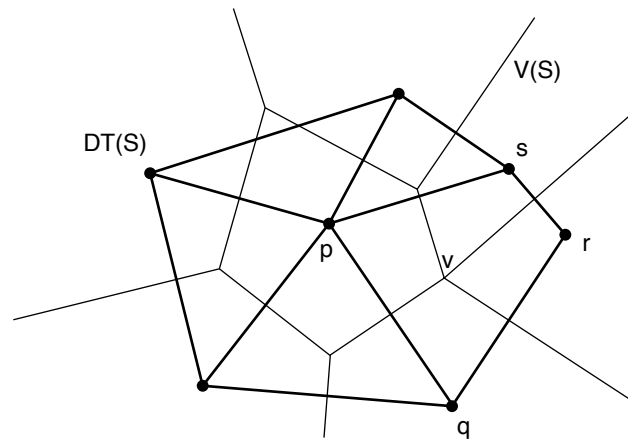


Delaunay Zerlegung $DT(S)$



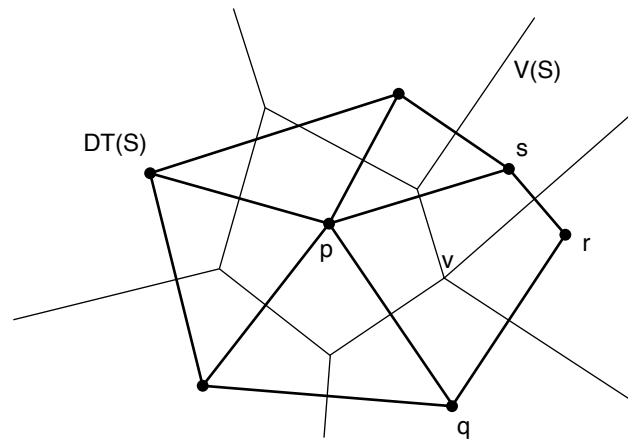
Delaunay Zerlegung $DT(S)$

- Allgemeine Lage: Keine 4 Punkte auf einem Kreis, keine drei Punkte auf einer Linie



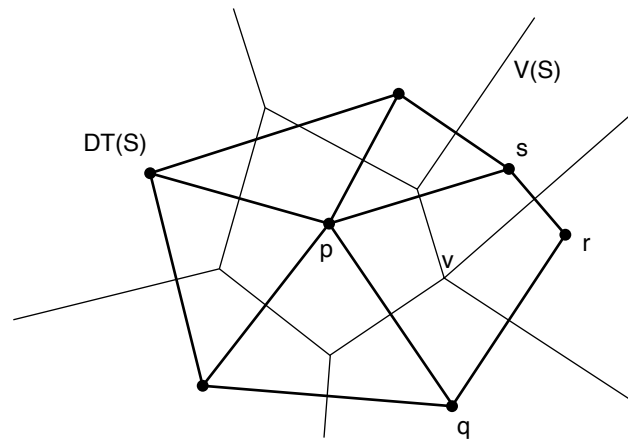
Delaunay Zerlegung $DT(S)$

- Allgemeine Lage: Keine 4 Punkte auf einem Kreis, keine drei Punkte auf einer Linie
- Jeder Voronoi Knoten hat Grad exakt 3



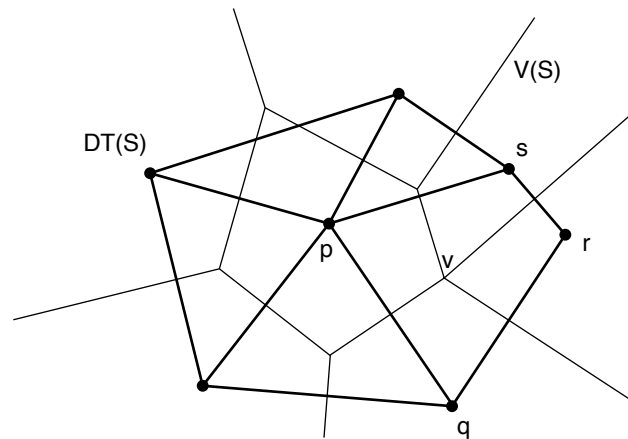
Delaunay Zerlegung $DT(S)$

- Allgemeine Lage: Keine 4 Punkte auf einem Kreis, keine drei Punkte auf einer Linie
- Jeder Voronoi Knoten hat Grad exakt 3
- $DT(S)$ ist Triangulation der Punktmenge



Delaunay Zerlegung $DT(S)$

- Allgemeine Lage: Keine 4 Punkte auf einem Kreis, keine drei Punkte auf einer Linie
- Jeder Voronoi Knoten hat Grad exakt 3
- $DT(S)$ ist Triangulation der Punktmenge
- Delaunay Triangulation



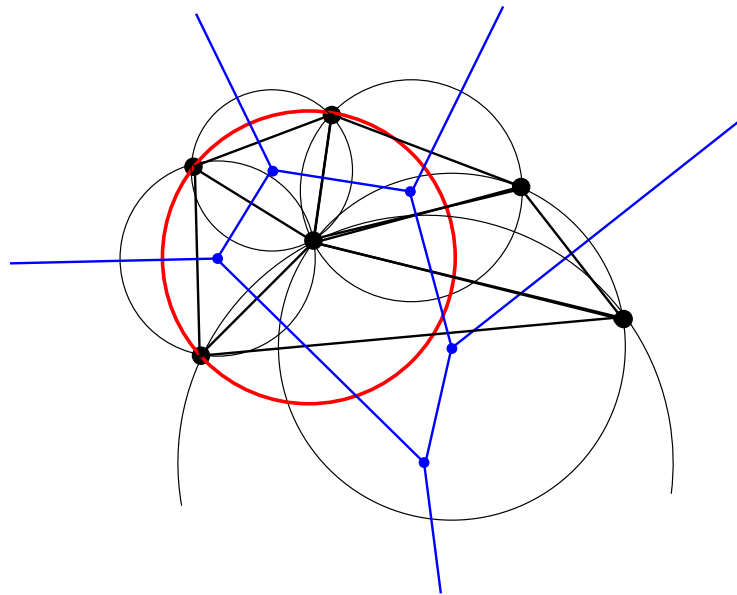
Delaunay Dreieck: Unkreisdefinition

Delaunay Dreieck: Unkreisdefinition

Lemma 5.18 Drei Punkte p, q, r aus S bilden genau dann ein Dreieck $tria(p, q, r)$ von $DT(S)$, wenn der eindeutig bestimmte Kreis $UK(p, q, r)$ durch p, q, r keinen anderen Punkt aus S im Inneren enthält.

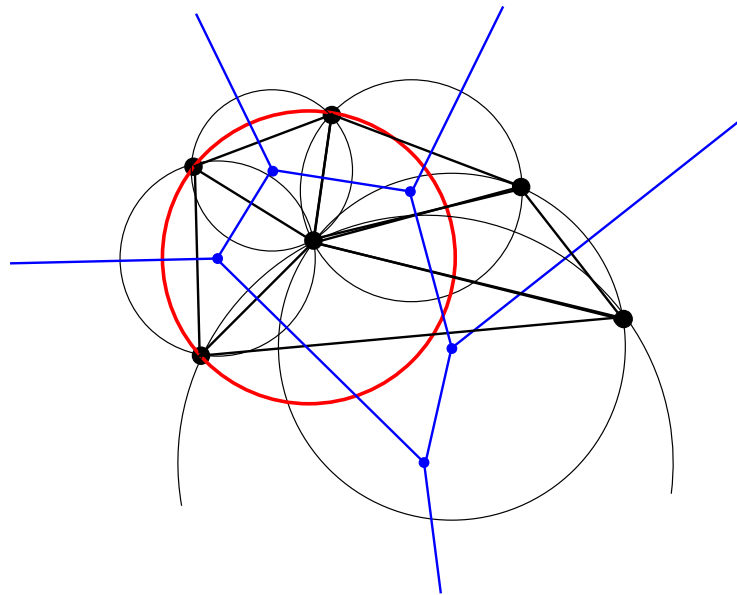
Delaunay Dreieck: Unkreisdefinition

Lemma 5.18 Drei Punkte p, q, r aus S bilden genau dann ein Dreieck $tria(p, q, r)$ von $DT(S)$, wenn der eindeutig bestimmte Kreis $UK(p, q, r)$ durch p, q, r keinen anderen Punkt aus S im Inneren enthält.



Delaunay Dreieck: Unkreisdefinition

Lemma 5.18 Drei Punkte p, q, r aus S bilden genau dann ein Dreieck $tria(p, q, r)$ von $DT(S)$, wenn der eindeutig bestimmte Kreis $UK(p, q, r)$ durch p, q, r keinen anderen Punkt aus S im Inneren enthält.



Beweis: Allgemeine Lage, Verwendung von Lemma 5.1

Besonderheit: Delaunay Triangulation

Besonderheit: Delaunay Triangulation

Theorem 5.17 Sei S eine Menge von Punkten in allgemeiner Lage. Die Delaunay Triangulation von S hat unter allen Triangulationen von S die größte Winkelfolge hat.

Besonderheit: Delaunay Triangulation

Theorem 5.17 Sei S eine Menge von Punkten in allgemeiner Lage. Die Delaunay Triangulation von S hat unter allen Triangulationen von S die größte Winkelfolge hat.

Winkelfolge (Innenwinkel der Dreiecke, sortiert)

$$w(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3d})$$

Besonderheit: Delaunay Triangulation

Theorem 5.17 Sei S eine Menge von Punkten in allgemeiner Lage. Die Delaunay Triangulation von S hat unter allen Triangulationen von S die größte Winkelfolge hat.

Winkelfolge (Innenwinkel der Dreiecke, sortiert)

$$w(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3d})$$

$w(DT(S)) > w(T(S))$ nach lexikographischer Ordnung

Besonderheit: Delaunay Triangulation

Theorem 5.17 Sei S eine Menge von Punkten in allgemeiner Lage. Die Delaunay Triangulation von S hat unter allen Triangulationen von S die größte Winkelfolge hat.

Winkelfolge (Innenwinkel der Dreiecke, sortiert)

$$w(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3d})$$

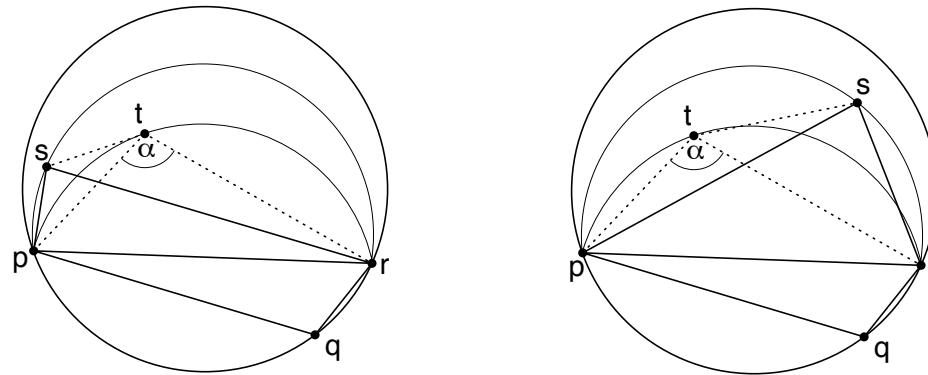
$w(DT(S)) > w(T(S))$ nach lexikographischer Ordnung

Beweis: Triangulation $T(S)$ ungleich $DT(S)$ verbessern

Beweis!

Theorem 5.17 Sei S eine Menge von Punkten in allgemeiner Lage. Die Delaunay Triangulation von S hat unter allen Triangulationen von S die größte Winkelfolge hat.

t mit max. Winkel liegt im Nachbardreieck



Beweis!

Theorem 5.17 Sei S eine Menge von Punkten in allgemeiner Lage. Die Delaunay Triangulation von S hat unter allen Triangulationen von S die größte Winkelfolge hat.

Edge Flip verbessert Winkelfolge!

