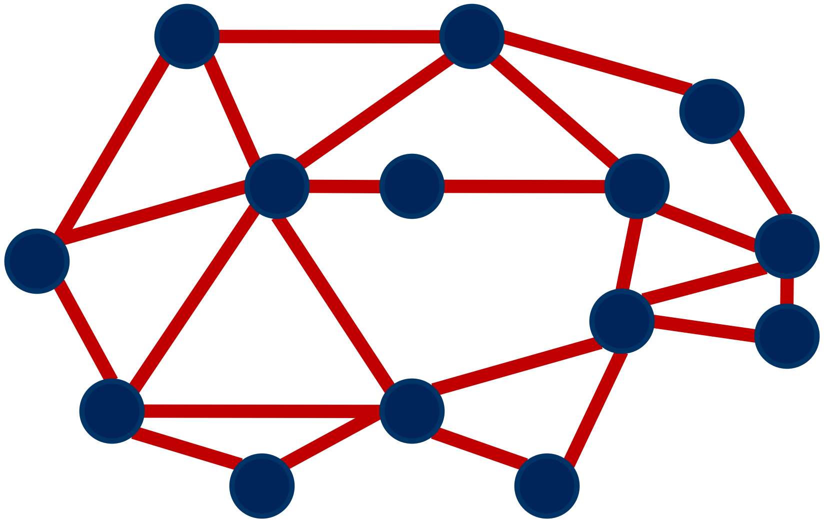
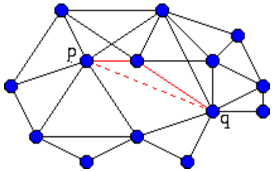


Spanner-Netzwerke

Kevin Busse

Seminar über Algorithmen
Freie Universität Berlin
SS 2009



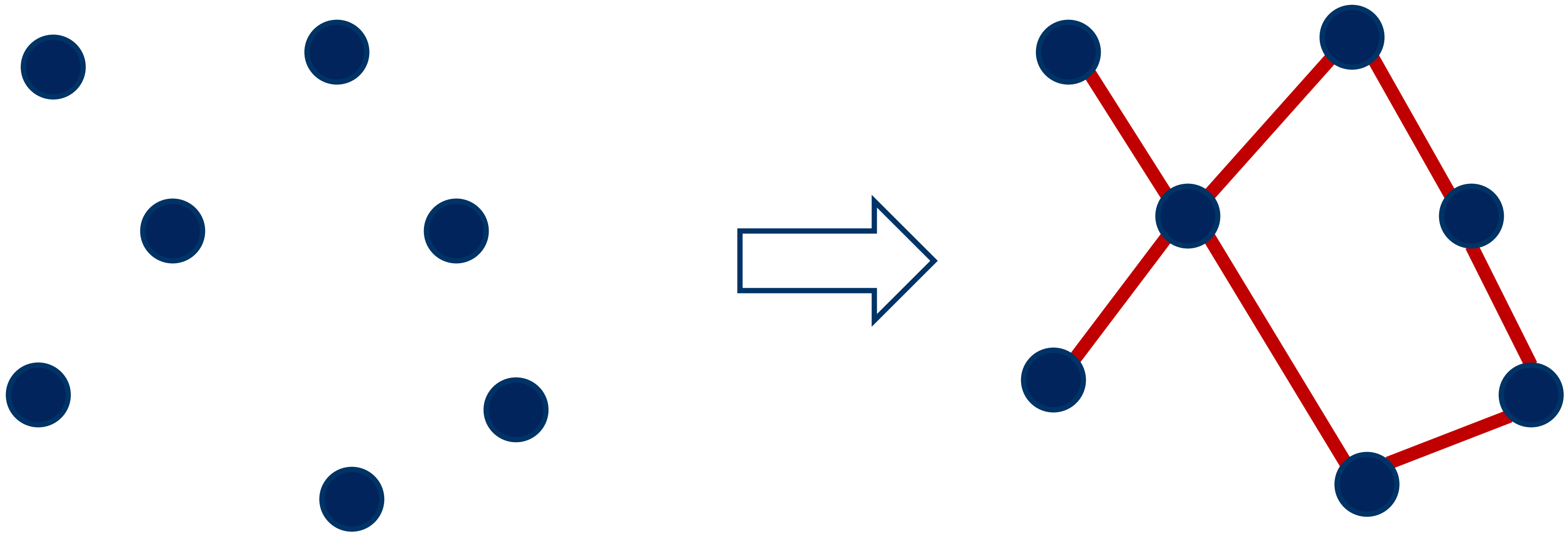
Aufspannende Netze

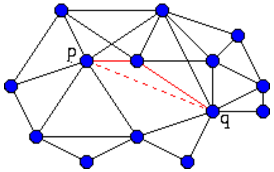
Allgemeine Problemstellung:

Sei S eine Menge von n Punkten aus dem R^d.

Wie konstruiert man *gutes* Netzwerk,

um diese Punkte zu verknüpfen?

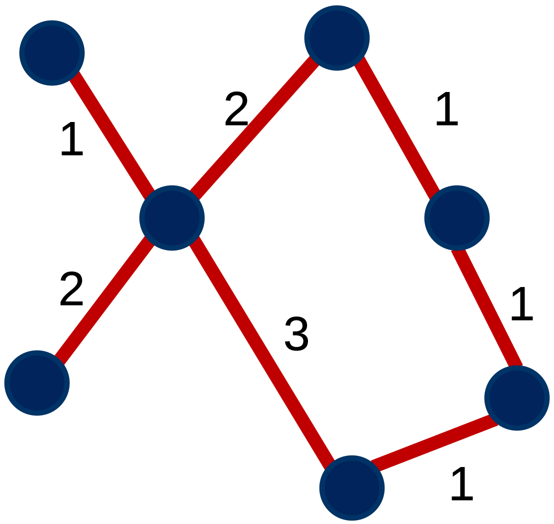




„Qualitätskriterien“

- Größe
  - gemessen in Anzahl der Kanten
  - bevorzugt klein
  - linear zur Anzahl der Ecken

    → *sparse*

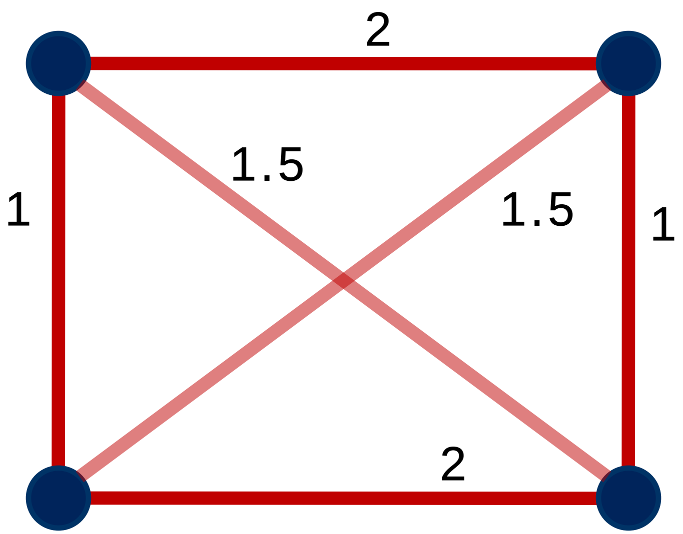


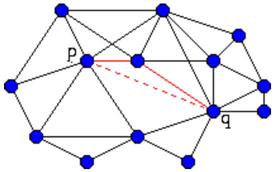
- Gewicht
  - Summe der Kantengewichte
  - bevorzugt klein
  - untere Schranke durch minimal aufspannende Bäume

„Qualitätskriterien“

- Ausdehnung
  - Zwei Punkte:
    - Verhältnis zwischen der Länge des kürzesten Weges und der Länge der direkten Verbindung
  - Spanner Netzwerk:
    - Maximum aller disjunkten Punktpaare
  - möglichst klein

    → *t*-SPANNERS

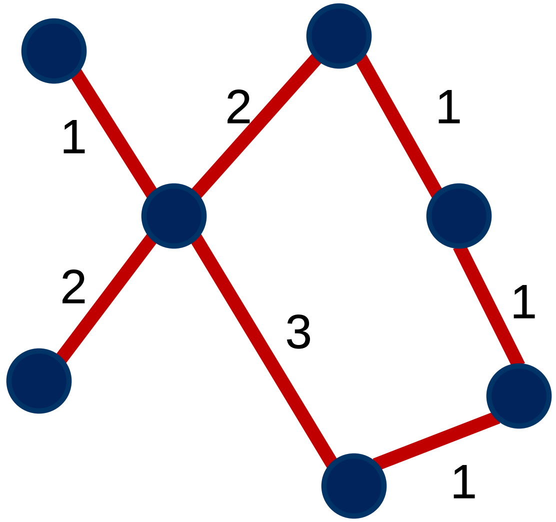




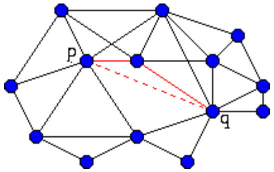
„Qualitätskriterien“

- Grad
  - maximale Anzahl an Kanten zu jedem Punkt

- ungewichteter Durchmesser
  - maximale Anzahl an Kanten auf allen kürzesten Wegen

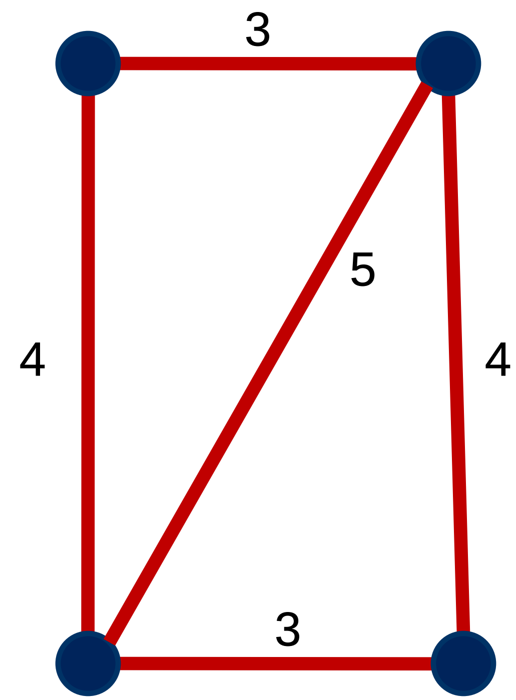


- gewichteter Durchmesser
  - maximale Summe der Kantengewichte auf allen kürzesten Wegen

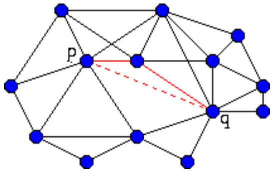


„Qualitätskriterien“

- Zusammenhang
  - Anzahl an Punkten oder Kanten, die entfernt werden müssen, damit das Netzwerk nicht mehr zusammenhängend ist



- Fehlertoleranz
  - Anzahl an Punkten oder Kanten, die entfernt werden müssen, damit andere Eigenschaften auf das Netzwerk nicht mehr zutreffen

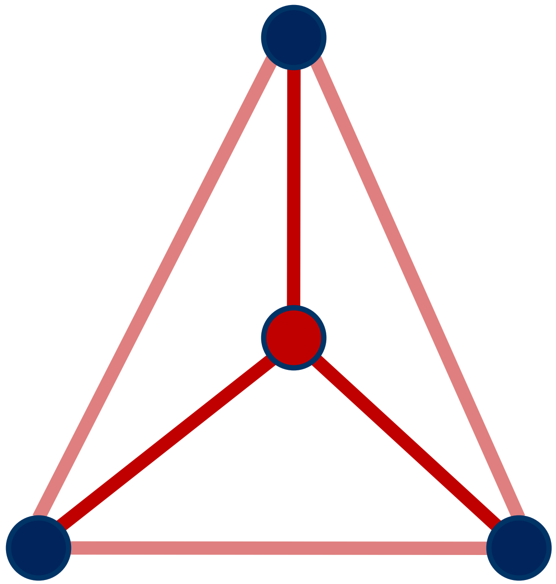


„Qualitätskriterien“

- Genus
  - Kriterium, das angibt, wie planar ein Graph ist
  - Alternativen
    - Größe des größten planeren Teilgraphen
    - Anzahl der Überkreuzungen von Kanten
  - bevorzugt planare Graphen

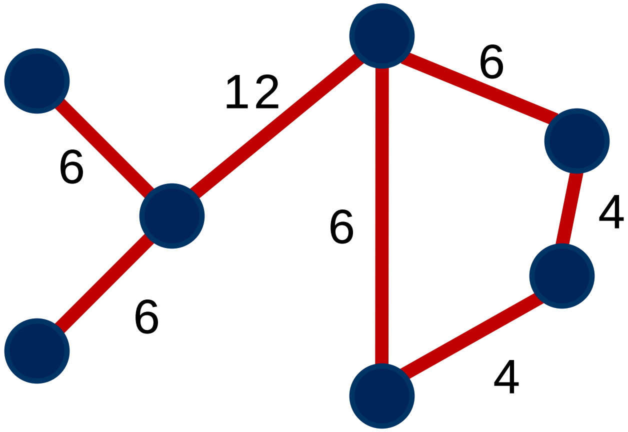
- Anzahl an Steinerpunkten
  - zusätzliche „Hilfs“-Punkte
  - kürzere Wege

    → *steiner minimal tree*
  - möglichst gering

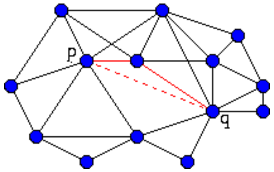


„Qualitätskriterien“

- Auslastung einer Kante
  - Anzahl an kürzesten Wegen aller Punktpaare durch diese Ecke
  - andere Definitionen möglich



- Auslastung des Netzwerks
  - maximale Auslastung aller Kanten
  - maximale Kante
    → *bottleneck*
  - möglichst klein



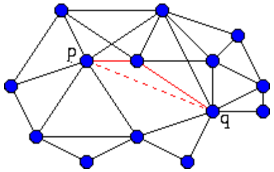
Design Anwendungen

- geg.: veraltetes Straßennetz zwischen Städten, mit beschränktem Budget Straßen auszubauen oder zu erneuern

→ Auswahl der zu restaurierenden Straßen, so dass

- alle Städte erreichbar
  - über restaurierte oder ausgebaute Straßen
  - ohne zu hohe Einbußen

- minimales Spannernetz
  - Anzahl an Straßen
  - Gesamtlänge der Straßen



Analyse Anwendungen

Freie Universität Berlin

- geg.: vorhandenes Verkehrsnetz

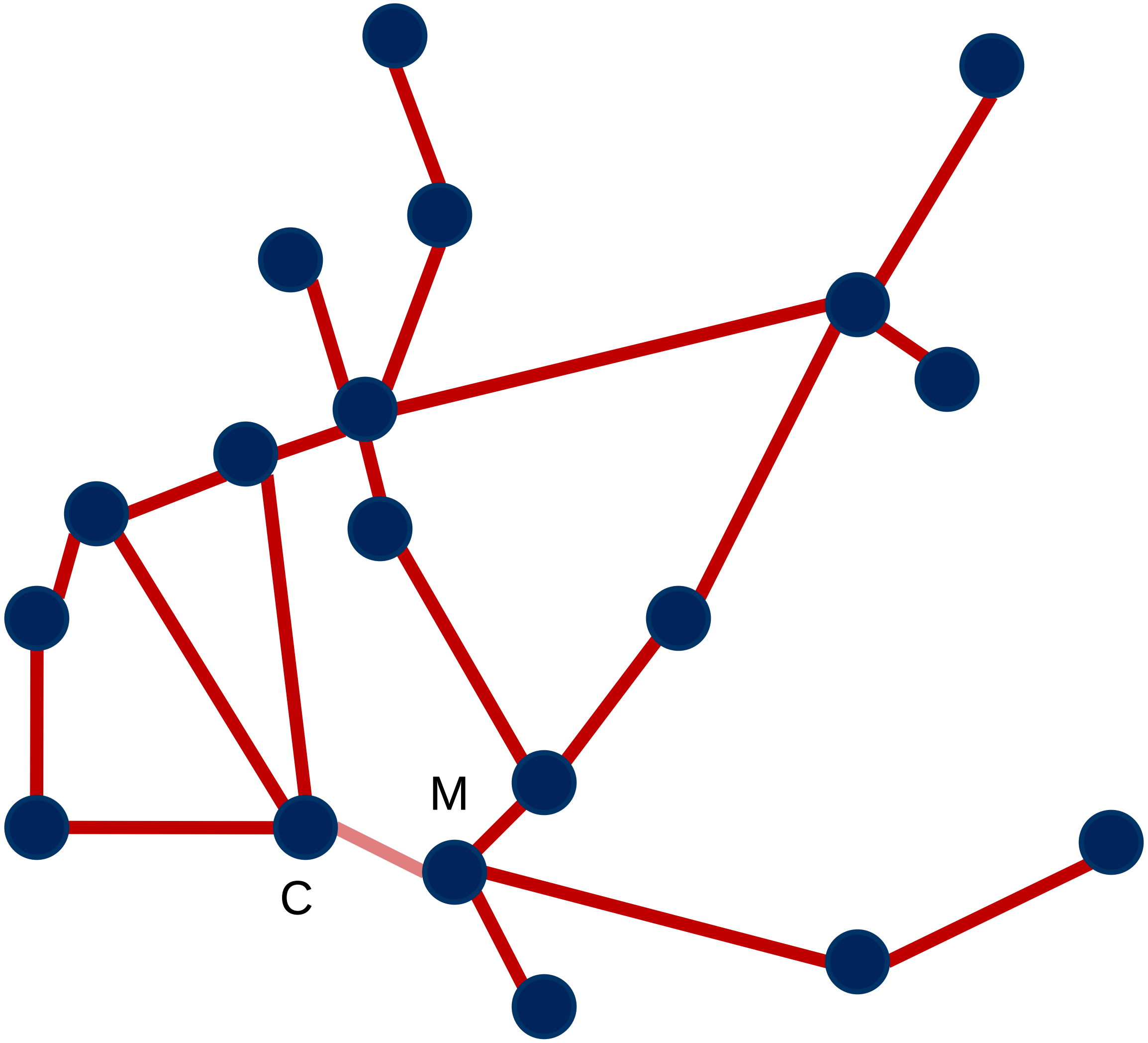
- Grundsätzliche Fragen
  - Was ist die Größe, das Gewicht, Ausdehnung, Durchmesser?
  - Welche sind die am weitesten entfernten Orte?

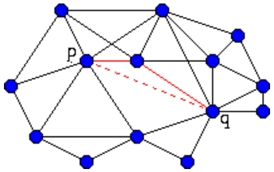
- Notfälle
  - Wie verändert sich die Ausdehnung des Netzes, wenn Streckenabschnitte ausfallen?

- Ausbau
  - Welche Strecken können eingefügt werden, um Ausdehnung oder Durchmesser zu verringern?

Beispiel

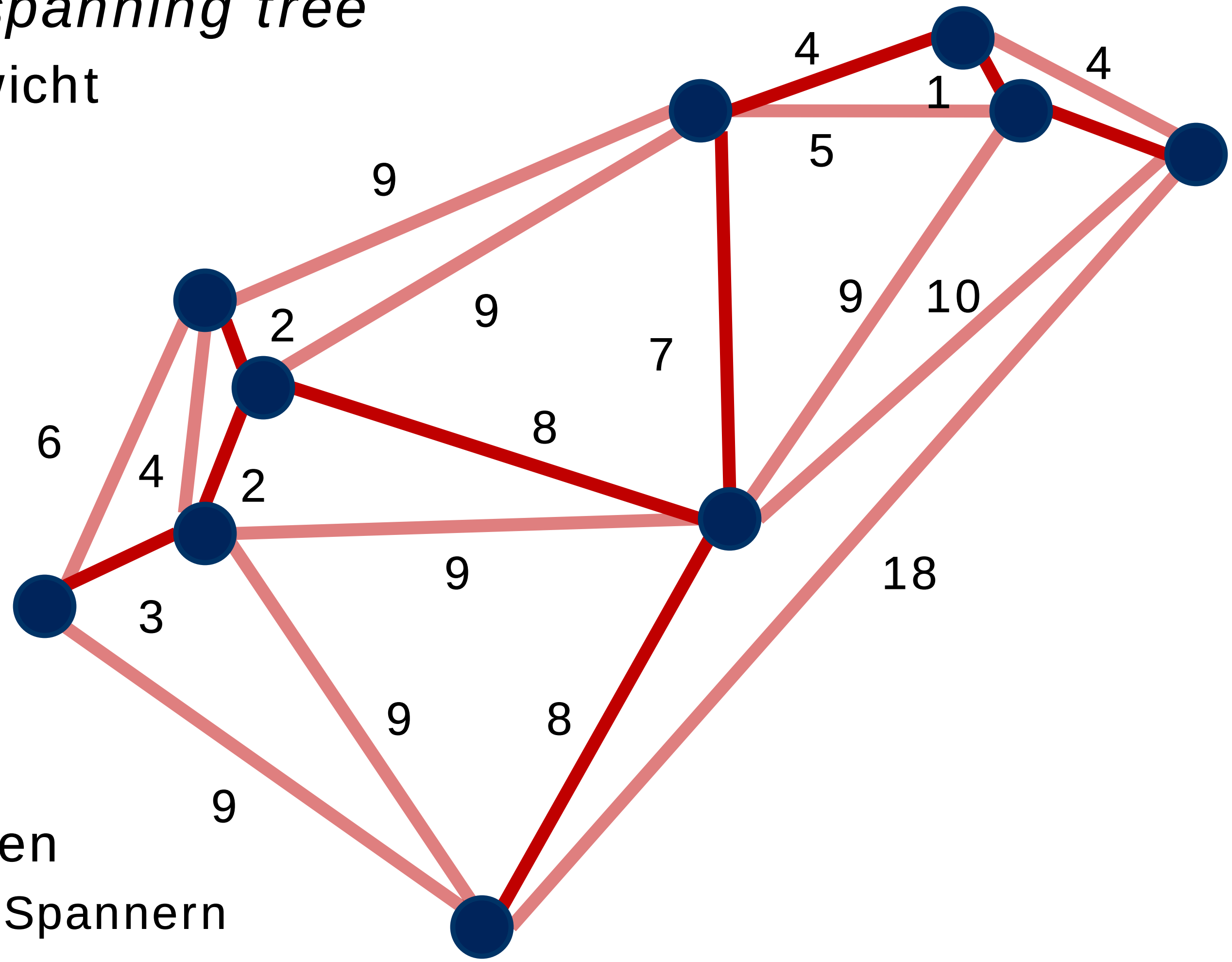
- Skandinavisches Zugnetz vor 2000
  - C – Kopenhagen
  - M – Malmö





Aufspannende Bäume

- MST – *minimal spanning tree*
  - minimales Gewicht
  - kleiner Grad
  - n-1 Kanten
  - minimal aufspannendes Netz
  - weitere Kanten nötig für weitere Qualitätskriterien
    → Vorlage von Spannern

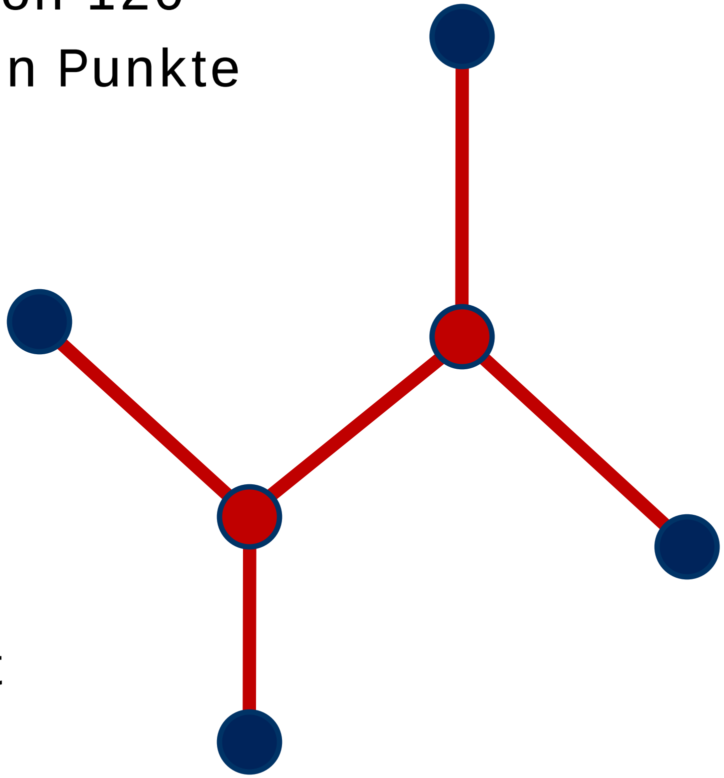


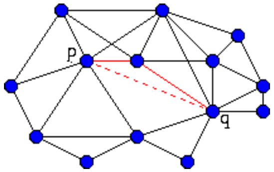
Aufspannende Bäume

- SMT – *steiner minimum tree*
- zusätzliche Punkte: Steinerpunkte
  - in Dreiecken auch bekannt als Fermat Punkt
  - verbindet 3 Punkte im Winkel von 120°
  - maximal n-2 Steinerpunkte für n Punkte

- erste Verwendung durch Gauß
  - Ziel: 4 Städte durch Zugschienen zu verbinden

→ ebenfalls als Grundlage zum Aufbau von Spannern geeignet





MST vs. SMT

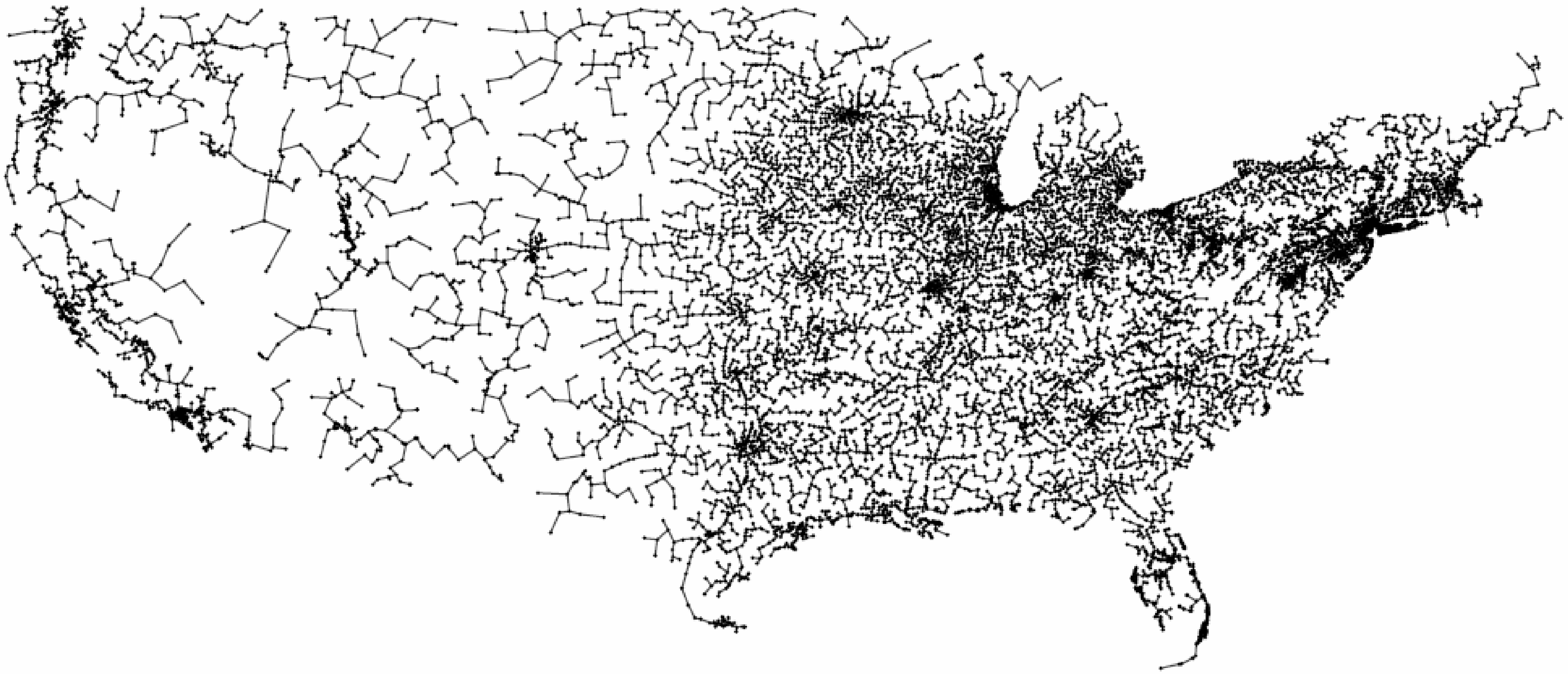
- Gewichtsvergleich der Aufspannenden Bäume

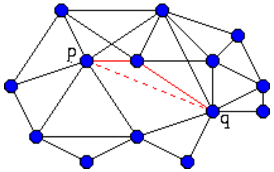
$$wt(SMT(S)) \leq \quad wt(MST(S))$$

$$wt(MST(S)) \leq 2\, wt(SMT(S))$$

- Konstruktion
  - MST Laufzeit: $O(n \log n)$
  - SMT NP-schwer

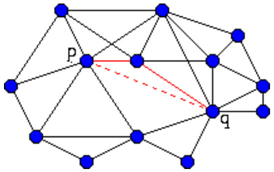
Beispiel





Triangulierung

- Triangulierung der Konvexen Hülle
  - planar
  - maximal $3n$-$6=O(n)$ Kanten
    → *sparse network*
- Anzahl an Triangulierungen: $O(59^n)$
- Bestimmung der minimalen Triangulierung: NP-schwer
- Greedy Triangulierung
  - Sortiere alle Kanten
  - Füge nacheinander kleinste Kanten hinzu, wenn diese keine Kanten aus dem Graphen schneiden



Definitionen

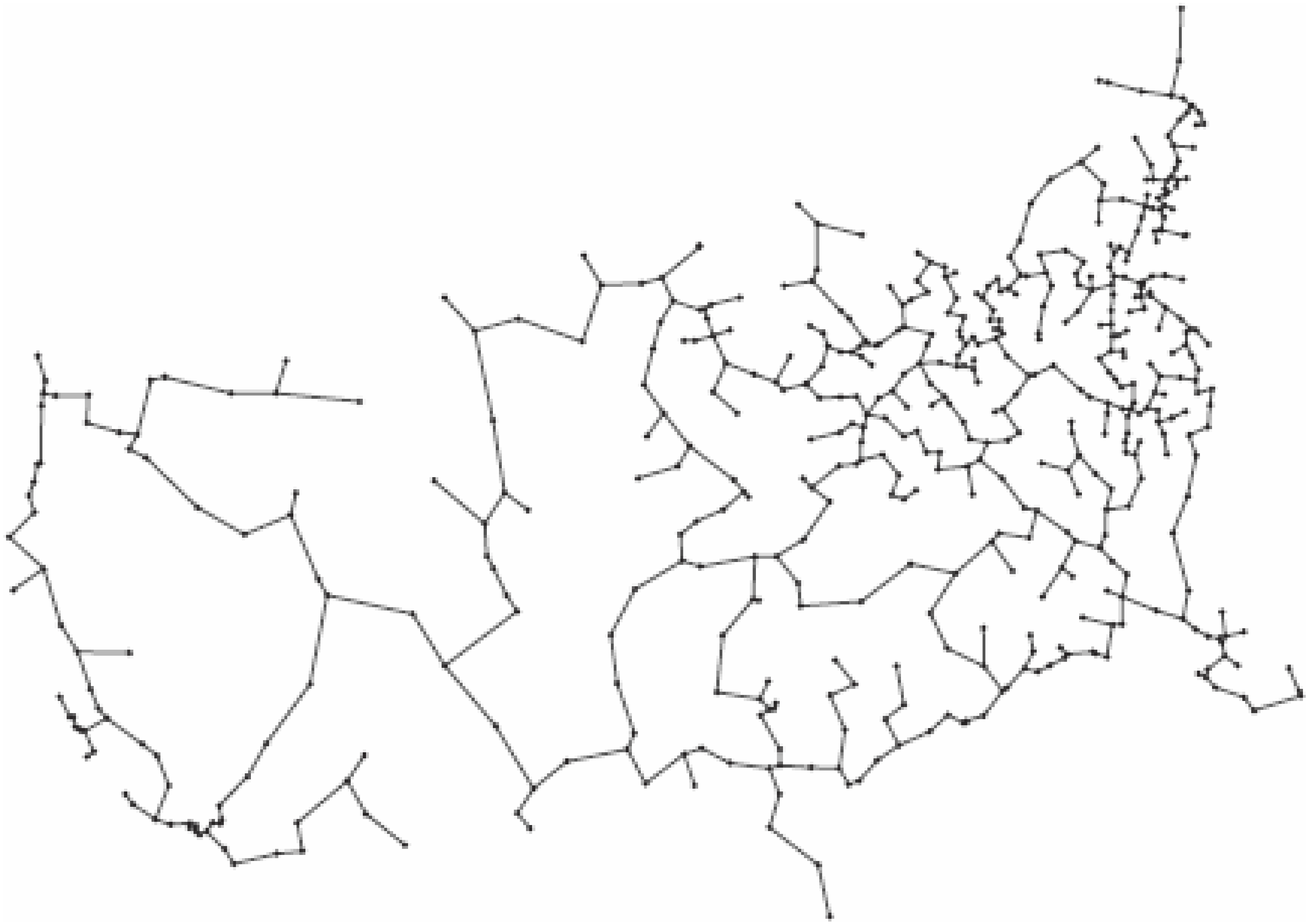
Sei S eine Menge von n Punkten aus dem R^d *und* $t \geq 1$

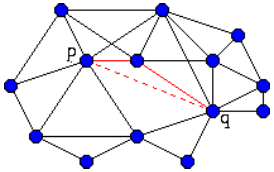
- *t-spanner path*: Weg zwischen Punktpaar p und q, sodass

$$wt(\text{Weg}(pq)) \leq t\,|pq|$$

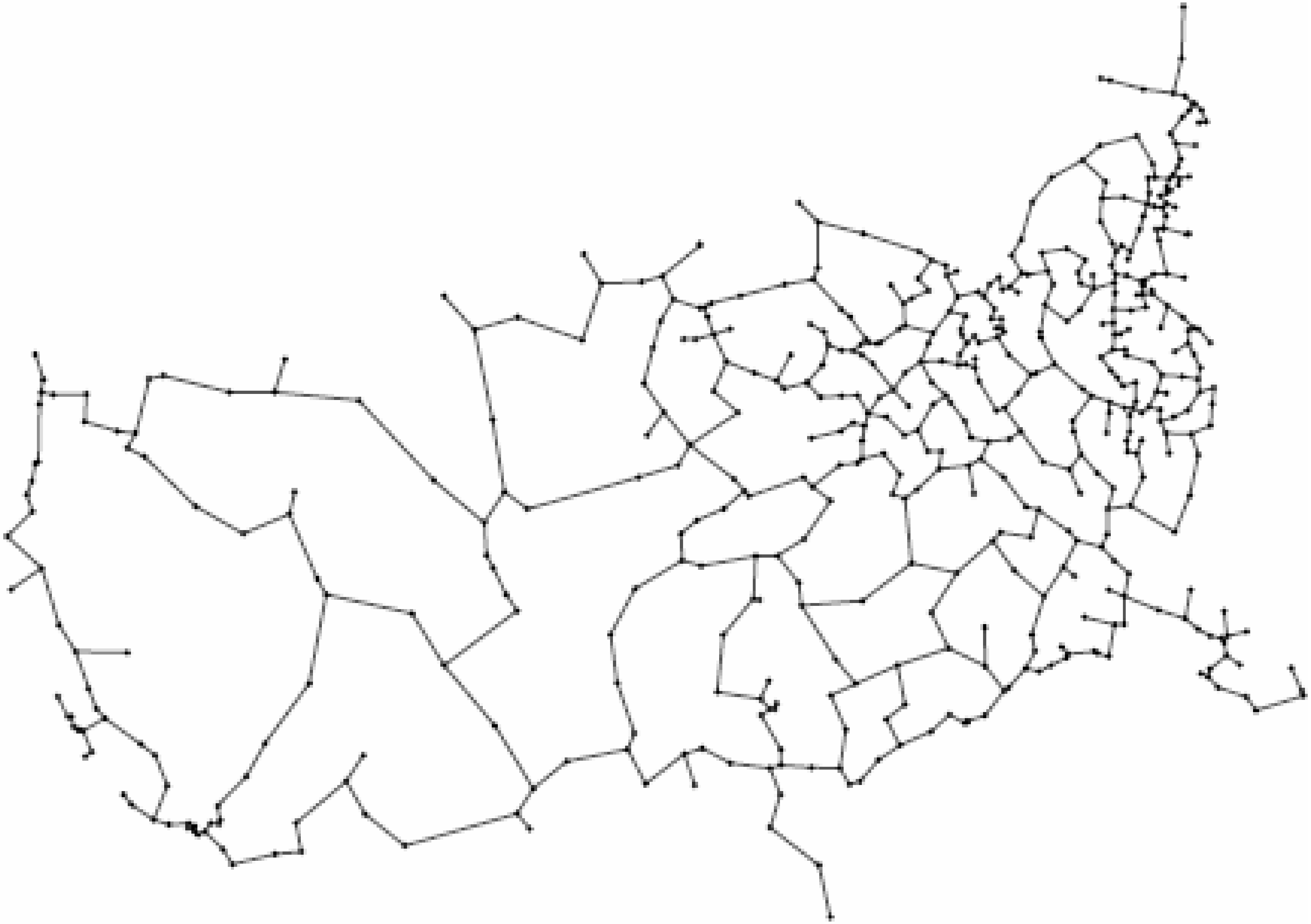
- *t-spanner*: ungerichteter Graph mit den Punkten aus S, so dass für jedes Punktpaar einen *t-spanner path* gibt
- *directed t-spanner path/directed t-spanner*: gerichteter Weg, bzw. gerichteter Graph
- *stretch factor*: kleinste reele Zahl t, sodass es *t-spanner* für S existiert

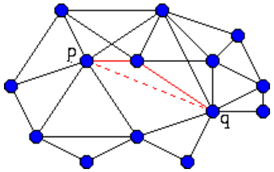
Beispiel: 10-spanner



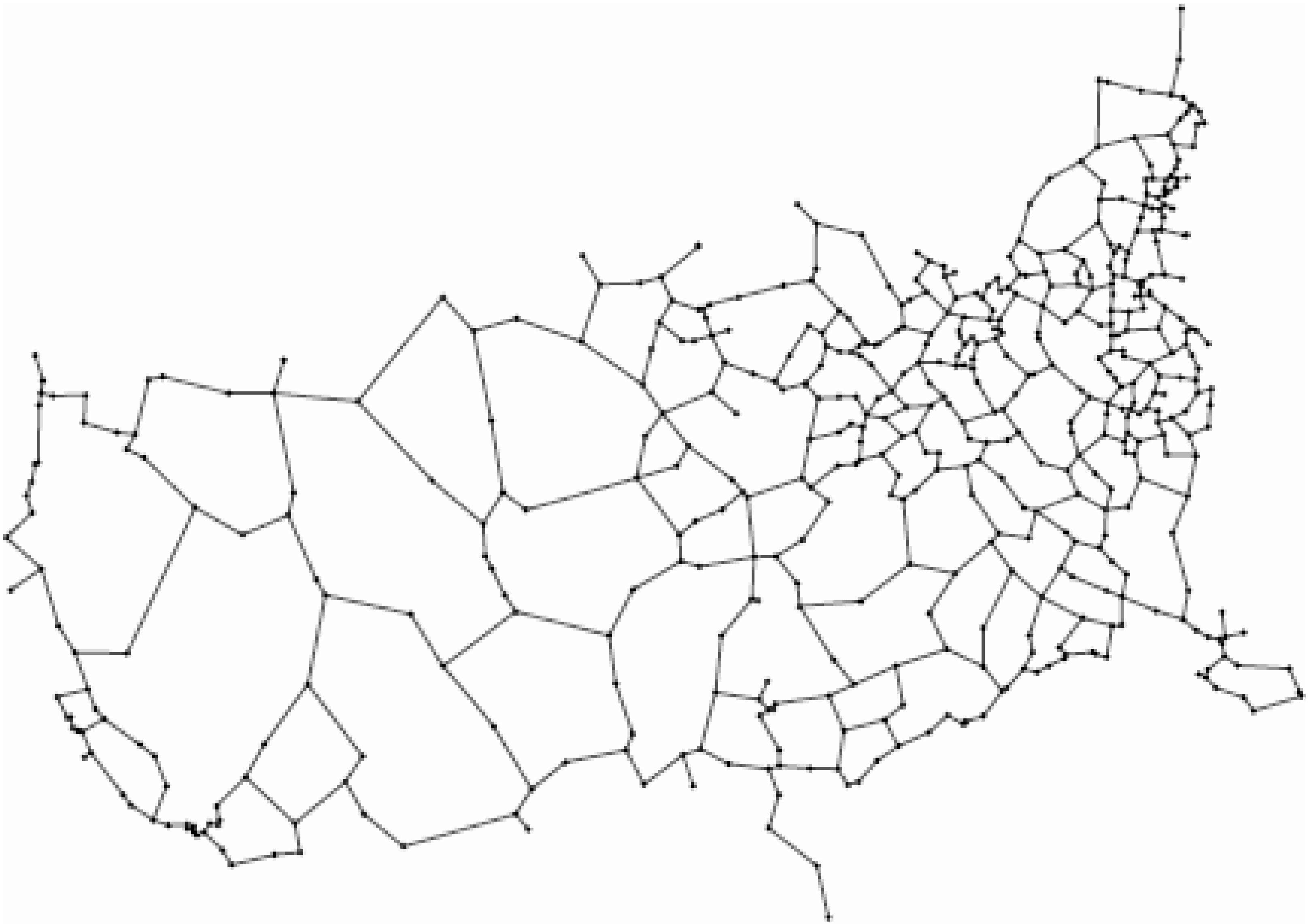


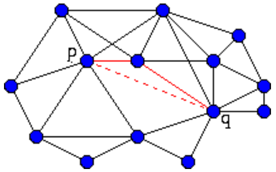
Beispiel: 5-spanner



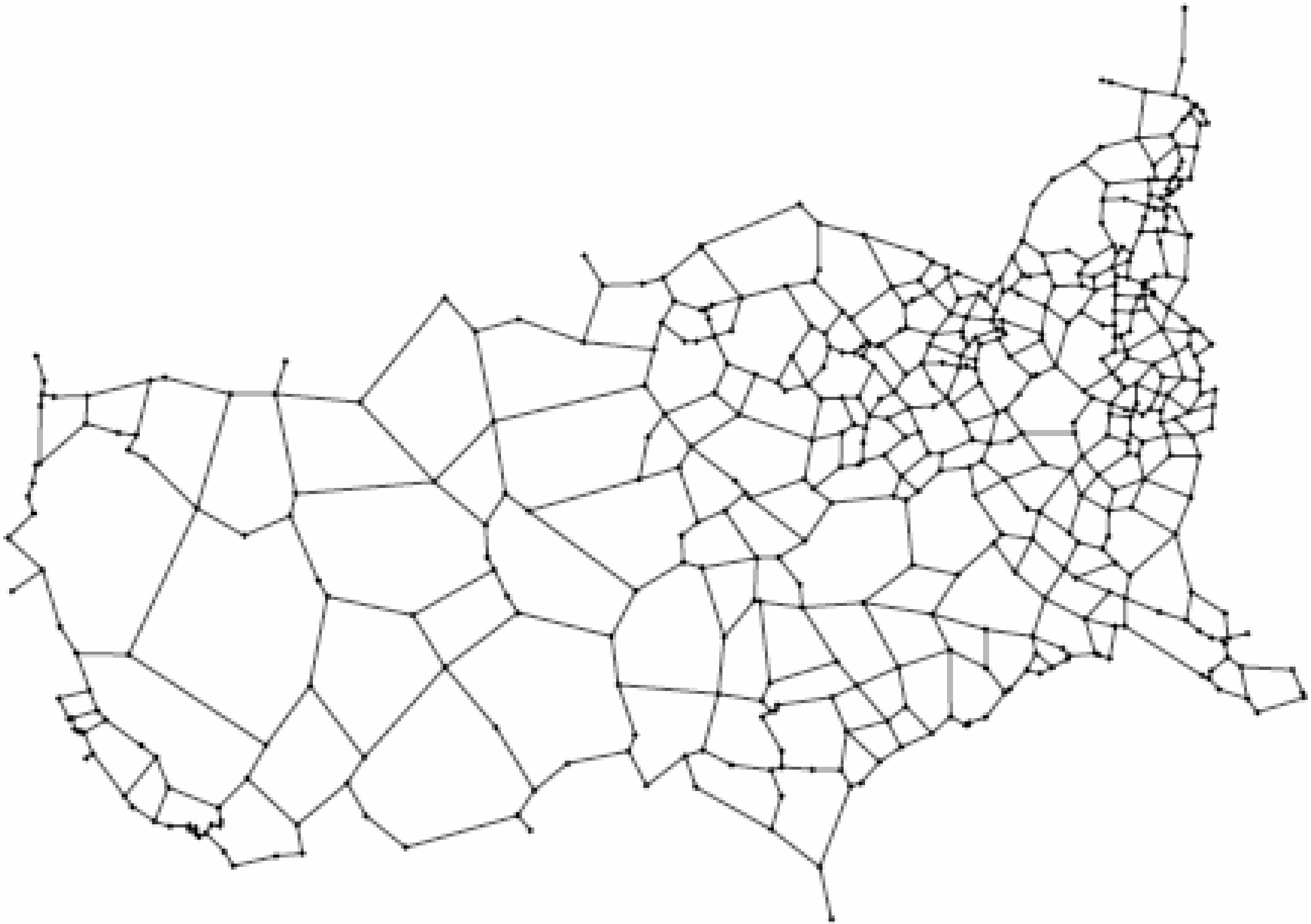


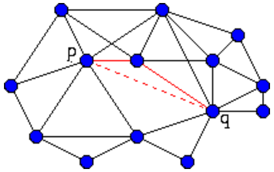
Beispiel: 3-spanner



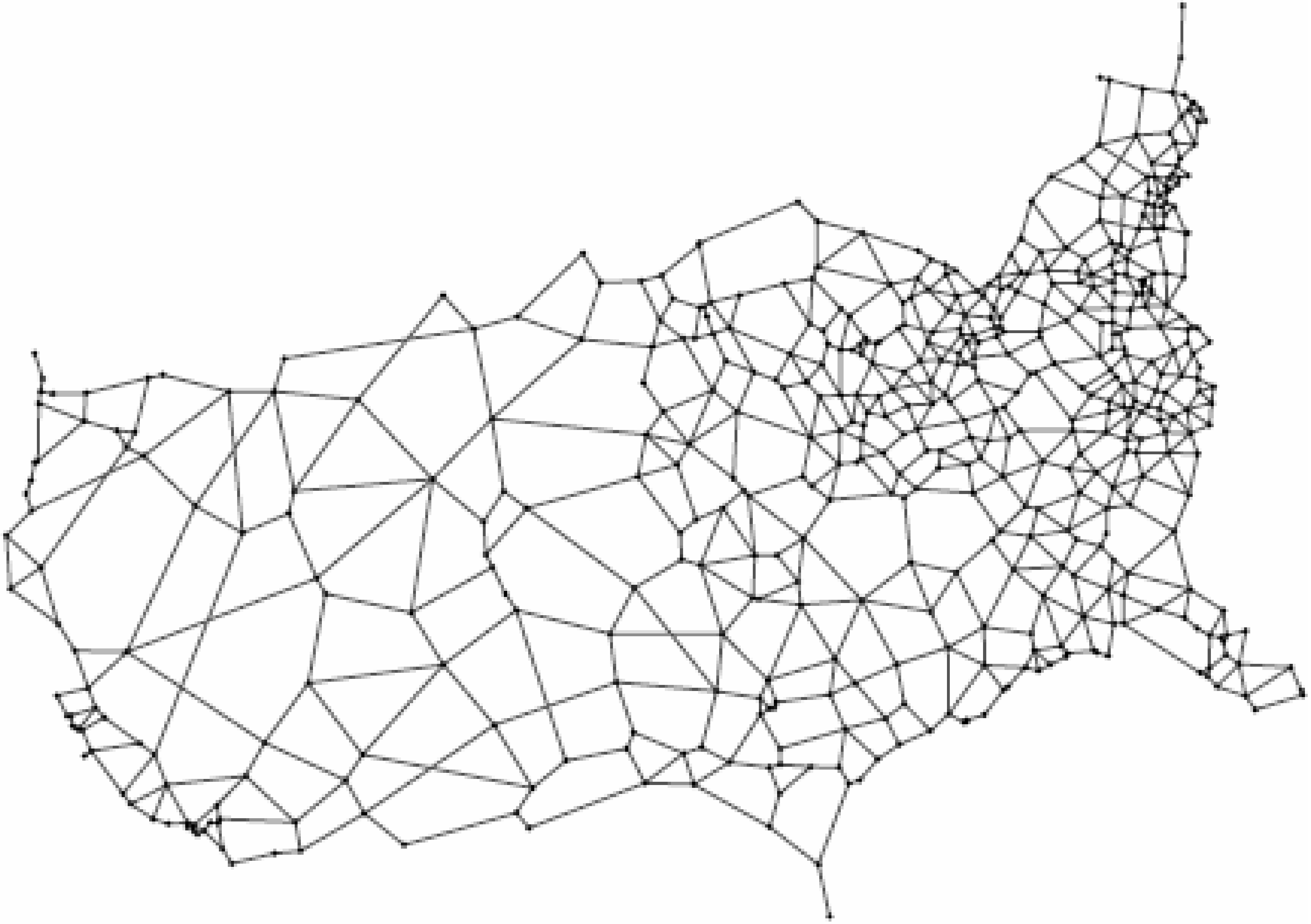


Beispiel: 2-spanner

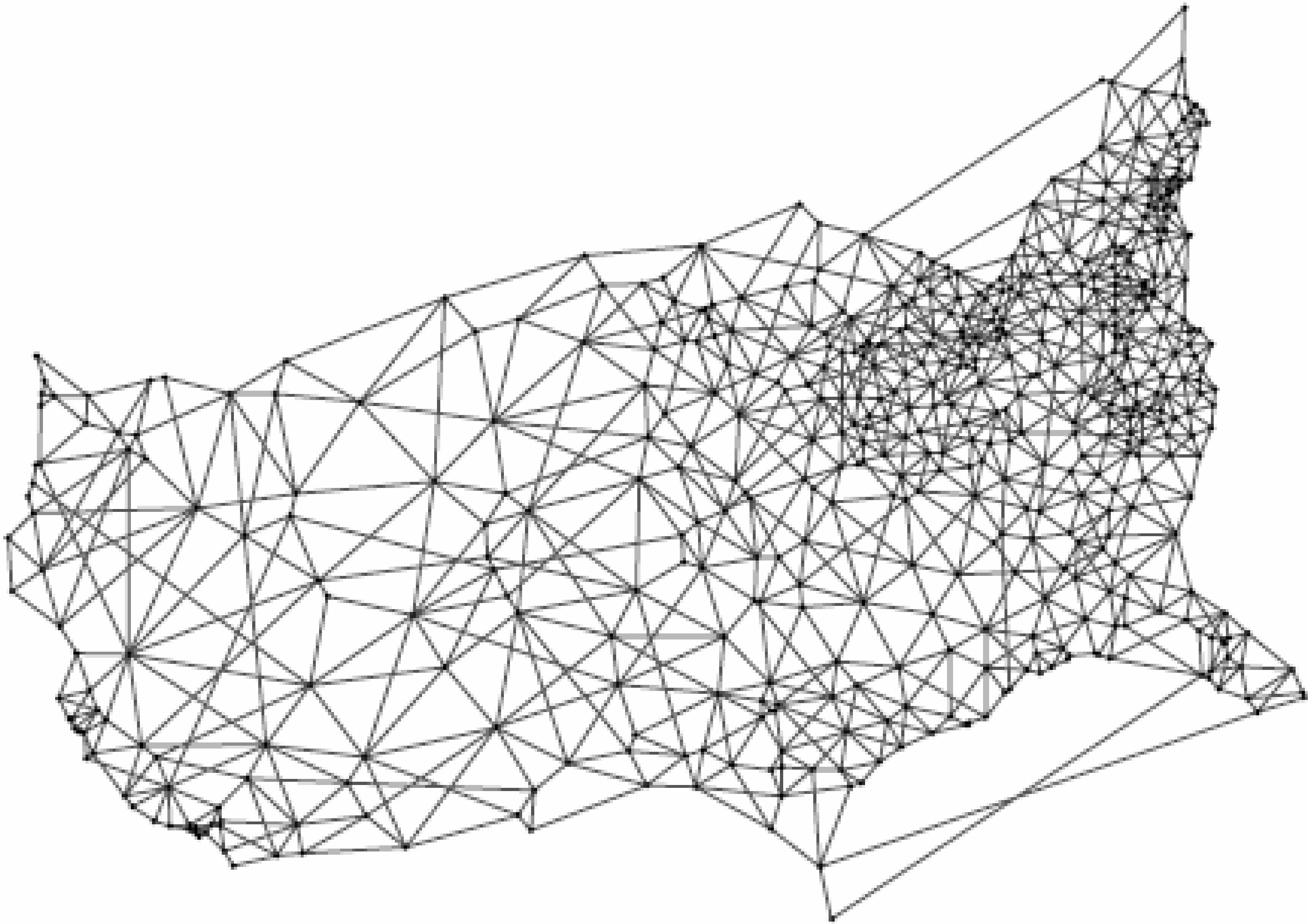


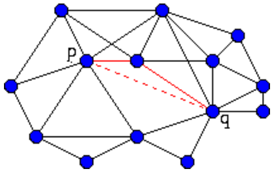


Beispiel: 1.5-spanner



Beispiel: 1.2-spanner





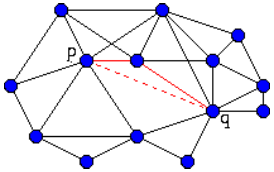
Spanner Problem

Sei S eine Menge von n Punkten aus dem R^d.

Und t > 1 eine reelle Zahl.

- Hauptfragen:
  - Existiert ein t-spanner für S mit maximal $c_{td}\, n$ Kanten?
  - Wieviel Zeit wird benötigt um diesen t-spanner zu finden?

- Weitere Fragen:
  - Existiert ein t-spanner für S
    - dessen Grad nur von t und d abhängt?
    - dessen Gewicht linear zum Gewicht des MST ist?
    - mit geringem Durchmesser
  - Lassen sich t-spanner in $O(n \log n)$ konstruieren?



GREEDY-Algorithmus

Eingabe: Punktemenge S und reele Zahl $t > 1$

Eingabe: t-spanner für S

PATHGREEDY(S, t)

sortiere Kanten und speichere sie in einer Liste L

$E := \emptyset$;

$G := (S, E)$;

for each $\{p, q\} \in L$

do

δ := Länge des kürzesten Weges in G von p nach q;

if $\delta > t\ |pq|$ **then**

$E := E \cup \{\{p, q\}\}$;

$G := (S, E)$;