Einführende Beispiele – Approximation mit relativer Gütegarantie

Marvin Schiller

Basierend auf R. Wanka: *Approximationsalgorithmen* (2006),
V. Vazirani: *Approximation Algorithms* (2001)

Studienstiftung des deutschen Volkes

Sommerakademie Görlitz, 2.-15. September 2007

Übersicht

Definitionen

Sei Π Optimisierungsproblem, A Approximationsalgorithmus für Π.

- Die relative Güte von A auf Eingabe I ist $\rho_A(I) = \max\{\frac{A(I)}{OPT(I)}, \frac{OPT(I)}{A(I)}\}$
- Die relative worst-case-Güte von A ist die Funktion $\rho_A^{wc}(n) = max\{\rho_A(I) \mid I \in \mathcal{D}, |I| \leq n\}$
- Gegeben Funktion $\rho_A : \mathbb{N} \to \mathbb{N}$, A garantiert relative Güte von $\rho_A(n)$, falls für alle n gilt: $\rho_A^{wc}(n) \leq \rho_A(n)$

Definitionen(II)

- Relativer Fehler von A bei Eingabe I:
  $\epsilon_A(I) = \frac{|A(I)-OPT(I)|}{OPT(I)} = |\frac{A(I)}{OPT(I)} - 1|$
- A garantiert einen relativen Fehler von höchstens $\epsilon_A(n)$, falls für alle $I \in \mathcal{D}$ mit $|I| \leq n$ gilt: $\epsilon_A(I) \leq \epsilon_A(n)$
- Gegeben Funktion $\rho'_A : \mathbb{N} \rightarrow \mathbb{N}$, hat Algorithmus A hat relative Abweichung von $\rho'_A(n)$, falls für unendlich viele n:
  $\rho_A^{wc}(n) \geq \rho'_A(n)$.
- Eine unendlich grosse Menge $\mathcal{D}'$ so dass $\mathcal{D}' \subseteq \mathcal{D}$ heisst $\rho'_A(n)$-Zeugenmenge gegen A, wenn $\forall I \in \mathcal{D}' : \rho_A(I) \geq \rho'_A(|I|)$ (einzelne solche Eingabe I heisst " $\rho'_A(n)$-Zeuge").

Schnitt-Probleme – Überblick

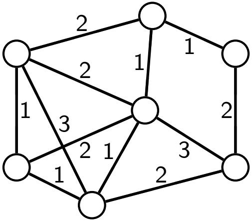
- Definition Schnitt
- Mehrfachschnitt
- Approximationsalgorithmus mit relativer Approximationsgüte
- Beweis + Zeugenmenge

Schnitt

Definitionen

Gegeben: ungerichteter Graph $G = (E, V)$
Zuordnung $w : E \rightarrow R^+$ (Kantengewichte)

Ein Schnitt (cut) wird durch eine Partition von V in zwei disjunkte Mengen V' und $V - V'$ definiert. Daraus ergibt sich eine durch den Schnitt gegebene Menge aller Kanten mit jeweils einem Endpunkt in jeder der Mengen V' und $V - V'$.

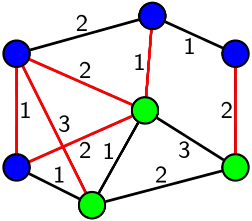


Schnitt

Definitionen

Gegeben: ungerichteter Graph $G = (E, V)$
Zuordnung $w : E \rightarrow R^{+}$ (Kantengewichte)

Ein Schnitt (cut) wird durch eine Partition von V in zwei disjunkte Mengen V' und $V - V'$ definiert. Daraus ergibt sich eine durch den Schnitt gegebene Menge aller Kanten mit jeweils einem Endpunkt in jeder der Mengen V' und $V - V'$.

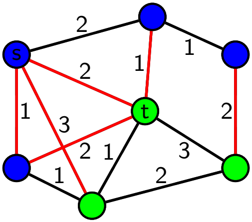


Schnitt

Definitionen

Gegeben: ungerichteter Graph $G = (E, V)$
Zuordnung $w : E \longrightarrow R^+$ (Kantengewichte)

Ein Schnitt (cut) wird durch eine Partition von V in zwei disjunkte Mengen V' und $V - V'$ definiert. Daraus ergibt sich eine durch den Schnitt gegebene Menge aller Kanten mit jeweils einem Endpunkt in jeder der Mengen V' und $V - V'$.

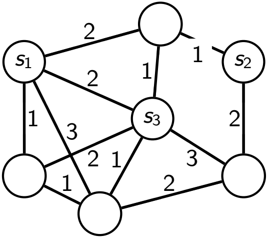


Ein *s*-*t*-Schnitt ist ein Schnitt, der zwei Knoten s und t in zwei unterschiedliche Partitionen trennt.

Definitionen (II)

Mehrfachschnitt (Multiway Cut)

Gegeben $S = \{s_1, s_2, ..., s_k\} \subseteq V$. Ein Mehrfachschnitt definiert eine Kantenmenge, bei deren Entfernen alle Knoten von S (die sog. "Terminale") voneinander getrennt werden.



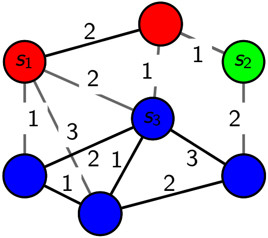
Mehrfachschnitt-Problem

Finde einen Mehrfachschnitt mit minimaler Summe der Kantengewichte (ist $\mathcal{NP}$-schwer für gegebenes $k \geq 3$).

Definitionen (II)

Mehrfachschnitt (Multiway Cut)

Gegeben $S = \{s_1, s_2, ..., s_k\} \subseteq V$. Ein Mehrfachschnitt definiert eine Kantenmenge, bei deren Entfernen alle Knoten von S (die sog. "Terminale") voneinander getrennt werden.



Mehrfachschnitt-Problem

Finde einen Mehrfachschnitt mit minimaler Summe der Kantengewichte (ist $\mathcal{NP}$-schwer für gegebenes $k \geq 3$).

Ein Approximationsalgorithmus für das Mehrfachschnitt-Problem

Definition: Isolierender Schnitt

Gegeben Knoten s_i, so erzeugt ein isolierender Schnitt eine Kanten-Menge, deren Entfernen s_i von allen (anderen) Terminalen trennt.

Algorithmus

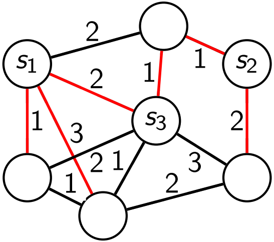
(i) Für alle $i = 1, \ldots, k$, finde einen minimalen isolierenden Schnitt für s_i, nenne ihn C_i.

(ii) Entferne den schwersten dieser Schnitte und gebe die Vereinigung der übrigen Schnitte aus, nenne das Ergebnis C.

Satz

Der Algorithmus erzielt eine relative Approximationsgüte von $2 - \frac{2}{k}$ [d.h. $w(C) \leq (2 - \frac{2}{k}) \cdot w(OPT)$].

Beweis



- Nehme an, S^* sei optimaler Mehrfachschnitt in Graph G.
- Betrachte S^* als Vereinigung von k Schnitten S_i, die jeweils ein Terminal s_i vom Rest abtrennen (=isolierender Schnitt). Es gilt: $S^* = \bigcup_{i=1}^{k} S_i$
- Jede Kante von S^* verbindet jeweils zwei verbundene Komponenten, und ist in jeweils zwei der Schnitte S_i, d.h. $\sum_{i=1}^{k} w(S_i) = 2w(S^*)$ (*).

Beweis (II)

Algorithmus (Erinnerung)

(i) Für alle $i = 1, \ldots, k$, finde einen minimalen isolierenden Schnitt für s_i, nenne ihn C_i.
(ii) Entferne den schwersten dieser Schnitte und gebe die Vereinigung der übrigen Schnitte aus, nenne das Ergebnis C.

- C_i (aus dem Algorithmus) ist minimaler isolierender Schnitt für s_i, daher $w(C_i) \leq w(S_i)$.
- Damit gilt:

$$w(C) = \sum_{i=1}^{k} w(C_i) \leq \sum_{i=1}^{k} w(S_i) \stackrel{(*)}{=} 2w(S^*)$$

(d.h. wir haben einen Faktor-2 Approximationsalgorithmus). $\square$

- Aber: es gibt eine engere obere Schranke (warum: wir haben den schwersten Schnitt C_i entfernt!)

Beweis (II)

Algorithmus (Erinnerung)

(i) Für alle $i = 1, \ldots, k$, finde einen minimalen isolierenden Schnitt für s_i, nenne ihn C_i.

(ii) Entferne den schwersten dieser Schnitte und gebe die Vereinigung der übrigen Schnitte aus, nenne das Ergebnis C.

- C_i (aus dem Algorithmus) ist minimaler isolierender Schnitt
- für s_i, daher $w(C_i) \leq w(S_i)$.

$$w(C) \leq (1-\frac{1}{k}) \sum_{i=1}^{k} w(C_i) \leq (1-\frac{1}{k}) \sum_{i=1}^{k} w(S_i) = 2 \cdot (1-\frac{1}{k}) \cdot w(S^*)$$

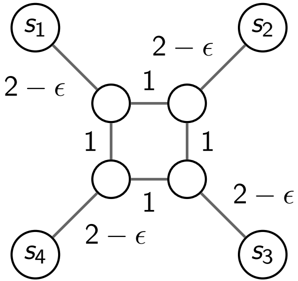
$\square$

Ein Zeuge für die relative Approximationsgüte $2 - \frac{2}{k}$

Frage: Kann die Analyse des Algorithmus verbessert werden? (bessere Gütegarantie?)
Suche nach unendlicher Beispielmenge, die das Gegenteil "bezeugt".

Ein Zeuge für die relative Approximationsgüte $2 - \frac{2}{k}$

Graph mit $2k$ Knoten, bestehend aus einem k-Kreis, und unterschiedlichen Terminalen verbunden mit jeweils einem der Kreisknoten. Sei ϵ eine kleine Zahl, $\epsilon > 0$.



- Für jedes Terminal s_i, ist der isolierende Schnitt mit minimalem Gewicht für s_i durch die Kante an s_i gegeben.
- Folglich erzeugt der Algorithmus einen Schnitt mit Gewicht $(k-1)(2-\epsilon)$

Das Handlungsreisendenproblem – Überblick

- Definition von TSP, metrischem TSP
- Einfüge-Heuristiken für TSP
- Minimal Spanning Tree (MST)
- Ein MST-basierter Algorithmus für metrisches TSP
- Relative Gütegarantie + Beweis
- Ein Unmöglichkeitsergebnis für TSP

Das Handlungsreisenden-Problem (Traveling Salesman Problem, TSP)

- $\mathcal{D} = \{\langle K_n, c\rangle | K_n$ vollständiger Graph mit n Knoten, $c : E \rightarrow \mathbb{N}$ Kantengewichtung$\}$.
- $S(\langle K_n, c\rangle) = \{C | C = (v_{i_1}, v_{i_2}, \ldots, v_{i_n}, v_{i_1})$ ist ein Hamilton-Kreis$\}$ Hamiltonkreis
- $f(C) = c(v_{i_n}, v_{i_1}) + \sum_{j=1}^{n-1} c(v_{i_j}, v_{i_{j+1}})$
- Finde eine Lösung $s \in S$ so dass $f(s)$ minimal wird.

Metrisches TSP (ΔTSP): Dreiecksgleichung gilt für Funktion c.
d.h. $\forall u, v, w \in V : c(u, v) \leq c(u, w) + c(w, v)$

überspringe Hamiltonkreis

Hamilton-Kreis

Weg

Gegeben Graph G=(V,E). $W = (v_1, \ldots v_n)$ mit jeweils $(v_i, v_{i+1}) \in E$ (für $i \in \{1, \ldots, n-1\}$) ist ein Weg in G.

Pfad

Weg, dessen Knoten voneinander verschieden sind .

Kreis

Weg, bei dem Start- und Endknoten identisch sind, alle anderen Knoten sind verschieden.

Hamilton-Kreis

Kreis, der alle Knoten eines gegebenen Graphen beinhaltet.

zurück

Einfüge-Heuristiken

Idee: Eine unfertige Tour $C = (u_1, u_2, ..., u_k, u_{k+1})$ wird schrittweise durch Einfügen von weiteren Knoten (zwischen zwei benachbarte Knoten der bestehenden Tour) erweitert.

Algorithmus EH

$C_1 := (v_1, v_1)$ für einen beliebigen Knoten v_1;
for j:= 2 to n do
  wähle einen Knoten v_j, der nicht in C_{j-1} ist;
  $C_j :=$ EINFÜGE(C_{j-1}, v_j)
done

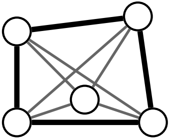
Einfügen

EINFÜGEN(C,v)

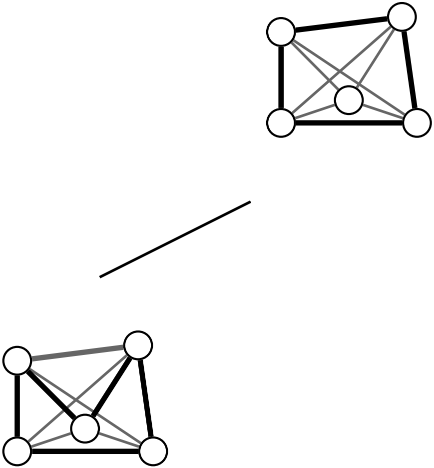
(i) Bestimme i, so dass $c(u_i, v) + c(v, u_{i+1}) - c(u_i, u_{i+1})$

(ii) Neuer Kreis: $(u_1, \ldots, u_i, u_{i+1}, \ldots, u_k, u_1)$

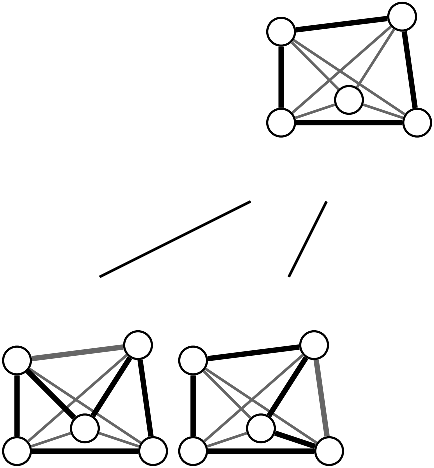
Einfügen – ein Beispiel



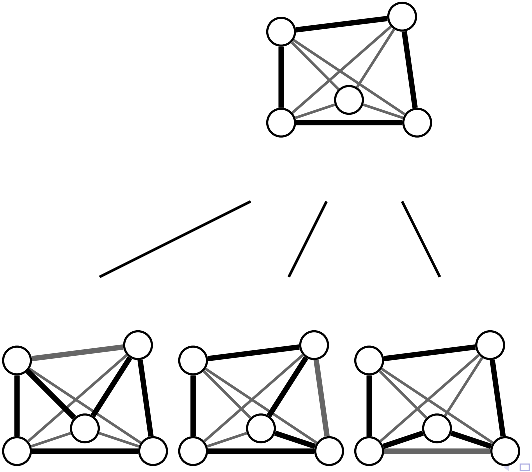
Einfügen – ein Beispiel



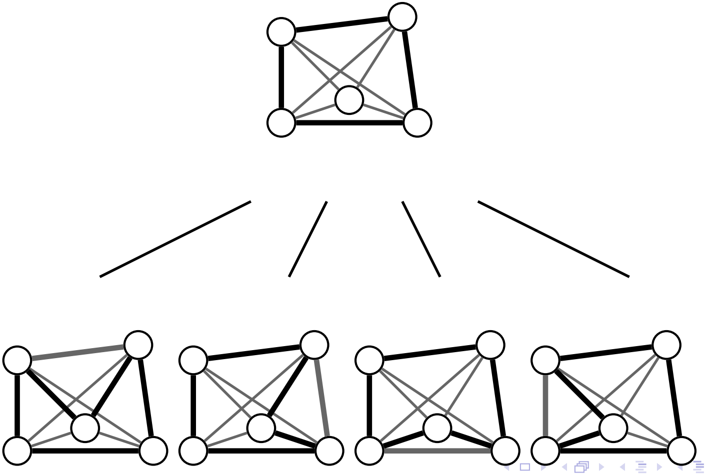
Einfügen – ein Beispiel



Einfügen – ein Beispiel



Einfügen – ein Beispiel



Gütegarantie

Satz

Jede Einfügeheuristik garantiert
$EH(\langle K_n, c\rangle) = \lceil \log n + 1\rceil \cdot \mathrm{OPT}(\langle K_n, c\rangle)$

Beweisidee

Betrachte optimale Tour. Bilde $\lceil \log\ n\rceil + 1$ Mal disjunkte Gruppen von Knoten Z_I, die beim Einfügen durch EH nicht mehr als OPT(I) Kosten erzeugen. Dies beinhaltet alle Knoten.

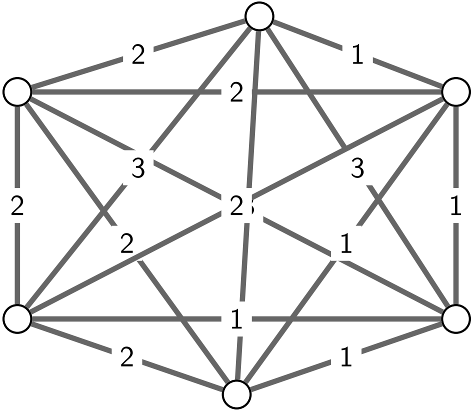
Christofides' Algorithmus

Definitionen

- Ein Matching M eines kantengewichteten Graphen G ist ein Teilgraph von G dessen Knoten maximalen Grad 1 haben.
- Gewicht eines Matchings M: Summe der Kanten in M.
- Sei G vollständiger Graph mit n Knoten (n gerade), dann gibt es perfekte Matchings: Matchings in denen alle Knoten genau Grad 1 haben (d.h. es sind genau $n/2$ Kanten in M).
- Leichtestes Matching: Perfektes Matching mit minimalem Gewicht.
- Berechnung eines leichtesten Matchings in $O(n^{2.5}(log n)^4)$ [Vaidya 1989].

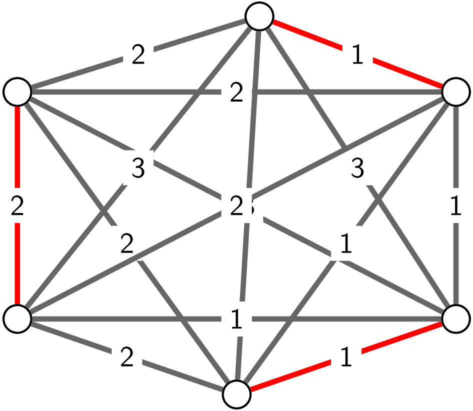
Matching

Beispiel



Matching

Beispiel



Weitere Definitionen

Multi-Graph: Graph, bei dem zwei Knoten durch mehrere Kanten miteinander verbunden sein können.

Euler-Pfad: Pfad $(u_1, u_2, ..., u_{|E|+1})$ in einem (Multi-) Graphen $G = (V, E)$ mit $u_i \in V$, in dem jede Kante genau einmal vorkommt. Die u_i müssen nicht notwendigerweise voneinander verschieden sein. Falls $u_1 = u_{|E|+1}$, handelt es sich um einen Euler-Kreis.

Euler-Kreis

Satz von Euler

Wenn alle Knoten in einem Graph G geraden Grad haben, dann hat G einen Euler-Kreis.

In diesem Fall kann der Kreis in $O(|V| + |E|)$ berechnet werden.

Christofides' Algorithmus

(i) Gegeben $I = \langle K_n, c \rangle$, berechne minimalen Spannbaum T_{CH} (mit Prims Algorithmus in $O(|E| + |V| \cdot log|V|))$.
Erinnerung MST

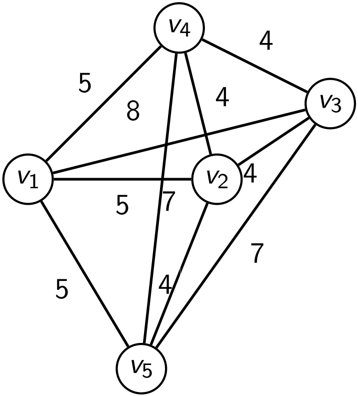
(ii) Sei $S := \{v \in T_{CH} | \deg_{T_{CH}}(v) \text{ ist ungerade}\}$.

(iii) Bilde den durch S induzierten Teilgraph von K_n und berechne darauf ein leichtestes Matching M_{CH} (möglich in $O(n^{2.5}(\log\, n)^4)$ [Vaidya 1989]).

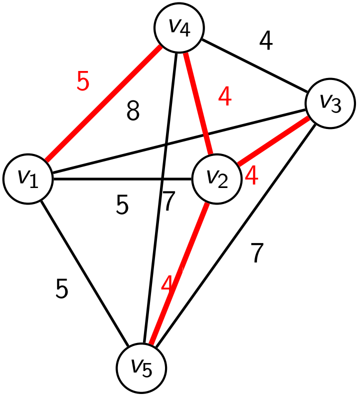
(iv) berechne einen Euler-Kreis $E = (u_1, u_2, \ldots)$ auf $T_{CH} \uplus M_{CH}$ [$T_{CH} \uplus M_{CH}$ kann Multi-Graph sein; alle Knoten haben geraden Grad]

(v) entferne Wiederholungen von Knoten in E, und gebe den Kreis aus.

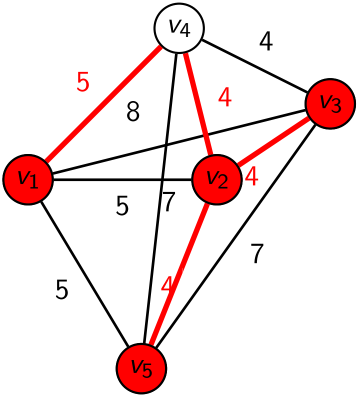
Gegeben: $I = \langle K_n, c \rangle$



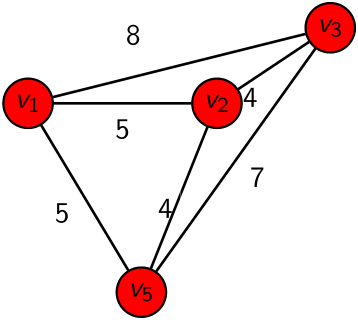
1. Berechne minimalen Spannbaum T_{CH}



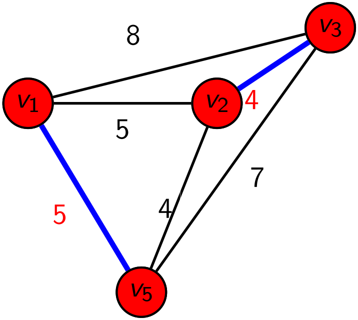
2. Sei $S := \{v \in T_{CH} | \deg_{T_{CH}}(v) \text{ist ungerade}\}$.



3. Bilde den durch S induzierten Teilgraph von K_n und berechne darauf ein leichtestes Matching M_{CH}.

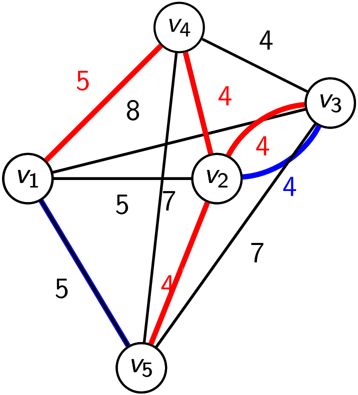


3. Bilde den durch S induzierten Teilgraph von K_n und berechne darauf ein leichtestes Matching M_{CH}.



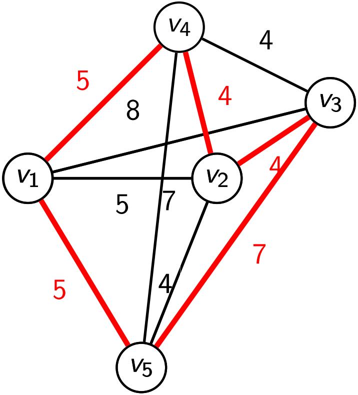
4. berechne einen Euler-Kreis $E = (u_1, u_2, \ldots)$ auf $T_{CH} \dot{\cup} M_{CH}$

[$T_{CH} \dot{\cup} M_{CH}$ kann Multi-Graph sein; alle Knoten haben geraden Grad]



Hier z.B.: $E = \{v_1, v_4, v_2, v_3, v_2, v_5, v_1\}$

5. entferne Wiederholungen von Knoten in E, und gebe den Kreis aus.



Approximationsgüte von Christofides Algorithmus

Der Algorithmus garantiert eine relative Güte von $p_{CH} \leq \frac{3}{2} - \frac{1}{n}$ für n Knoten. Die Laufzeit ist $O(n^{2.5}(\log n)^4)$.

Beweis überspringen

Beweis

Sei R^* die optimale Rundreise für Probleminstanz I mit n Knoten. Wir müssen zeigen, dass $CH(I) \leq (\frac{3}{2} - \frac{1}{|V|}) \cdot R^*$.

1. Kosten von T_{CH}

In R^* gibt es mindestens eine Kante ê mit Länge $\frac{c(R^*)}{n}$ ("Durchschnittliche" Länge). Entferne ê aus Hamiltonkreis und erhalte Spannbaum T'. Da T_{CH} minimaler Spannbaum, gilt $c(T_{CH}) \leq c(T')$. Genauer:

$$c(T_{CH}) \leq (1 - \frac{1}{n}) \cdot c(R^*)$$

Beweis (II)

2. Kosten von leichtestem Matching M_{CH} auf durch S induziertem Graphen

- Nenne $R^* = \{u_1, u_2, ..., u_n, u_1\}$
- Schreibe $S = \{u_{i_1}, ..., u_{i_{|S|}}\}$ mit $i_1 < ... < i_{|S|}$
- Sei $H = \{u_{i_1}, ..., u_{i_{|S|}}, u_{i_1}\}$ Kreis (Teilsequenz von R^*, und daran angelehnt)
- Wegen Dreiecksungleichung: $c(H) \leq c(R^*)$ [siehe Tafel]
- Da $|S|$ gerade: H zerlegbar in zwei perfekte Matchings M_1, M_2. Sei OBdA $c(M_1) \leq c(M_2)$, so gilt:

$$c(M_{CH}) \overset{\substack{\text{leichtestes}\\ \text{Matching}}}{\leq} c(M_1) \leq \frac{1}{2} \cdot (c(M_1) + c(M_2)) \leq \frac{1}{2} c(H) \leq \frac{1}{2} c(R^*)$$

Beweis (III)

3. Kosten der Euler-Tour

- Jeder Knoten in $M_{CH} \uplus T_{CH}$ hat geraden Grad $\Rightarrow$ es gibt eine Euler-Tour E.
-

$$
\begin{aligned}
c(E) &= c(M_{CH} \uplus T_{CH}) \\
&\leq (1 - \frac{1}{n}) \cdot c(R^*) + \frac{1}{2} c(R^*) \\
&= (\frac{3}{2} - \frac{1}{n}) \cdot OPT(I)
\end{aligned}
$$

$\square$

Ein Unmöglichkeitsergebnis für TSP

- Voriger Beweis beruht auf der Dreiecksungleichung
- Ohne Dreiecksungleichung: Wenn es einen polynomiellen Approximationsalgorithmus für das volle TSP mit relativer Güte r gibt, dann $\mathcal{P} = \mathcal{NP}$.
- Beweis: Reduktion eines $\mathcal{NP}$-vollständigen Problems (Entscheidungsproblem Hamilton-Kreis) auf TSP mit relativer Güte r: HAMILTON $\leq_p$ TSP[r] ($\leq_p$ Reduktion in Polynomialzeit).
- Technik: gap amplification

Beweis

- Nehme an, es gibt einen polynomiellen Approximationsalgorithmus A für TSP mit relativer Güte r.
- Dann benutze folgenden Algorithmus, um HAMILTON (in $\mathcal{NP}$) mithilfe von A (in $\mathcal{P}$) zu entscheiden:

(i) berechne eine Probleminstanz I_G für TSP (Details folgen!)
(ii) verwende A um eine kürzeste Rundreise zu approximieren
(iii) *If* $A(I_G) > r \cdot |V|$ *then* $\top$ *else* $\bot$

Konstruktion von I_G

Sei $n = |V|$. Setze $I_G = \langle K_n, c \rangle$ (K_n ist vollständiger Graph) mit

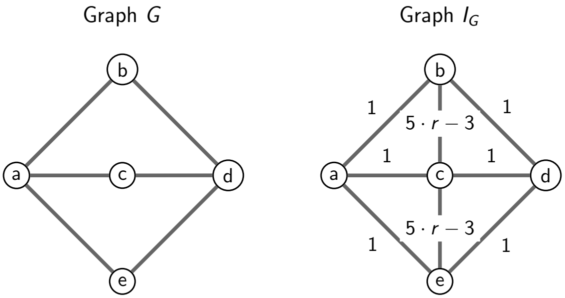
$$c(u,v) = \begin{cases} 1 & \text{falls } \{u,v\} \in E \\ (r-1) \cdot n + 2 & \text{falls } \{u,v)\} \notin E \end{cases}$$

- I_G in Polynomzeit aus G berechenbar
- Kanten mit Gewicht $(r-1) \cdot n + 2$ "lange Kanten"
- Kanten mit Gewicht 1 "kurze Kanten".

Konstruktion von I_G

Sei $n = |V|$. Setze $I_G = \langle K_n, c\rangle$ (K_n ist vollständiger Graph) mit

$$c(u,v) = \begin{cases} 1 & \text{falls } \{u,v\} \in E \\ (r-1)\cdot n + 2 & \text{falls } \{u,v)\} \notin E \end{cases}$$



Konstruktion von I_G

Fall 1: G hat Hamilton-Kreis

$\Rightarrow$ kürzeste Rundreise in I_G hat Länge n (zu jeder Kante im Hamilton-Kreis gibt es eine kurze Kante in I_G).

Fall 2: G hat keinen Hamilton-Kreis

$\Rightarrow$ in jeder Rundreise gibt es mindestens eine lange Kante.

Konsequenz: TSP I_G nimmt nur Lösungen an mit Länge kleiner als $n + 1$ oder grösser als $r \cdot n$ (d.h., wir haben einen gap/ein Loch). D.h., der gegebene Algorithmus entscheidet HAMILTON. $\square$

(i) berechne eine Probleminstanz I_G für TSP

(ii) verwende A um eine kürzeste Rundreise zu approximieren

(iii) *If* $A(I_G) > r \cdot |V|$ *then* $\top$ *else* $\perp$

Resümee

Das haben wir gesehen:

- Verschieden Beispiele für Approximationsalgorithmen mit relativer Gütegarantie.
- Beweise relativer Gütegarantien
- Die Rolle der Zeugenmenge
- Ein Unmöglichkeitsergebnis: es gibt keinen polynomiellen Approximationsalgorithmus für das generelle TSP mit fester relativer Güte r, es sei denn, $\mathcal{P} = \mathcal{NP}$.

Minimaler Spannbaum (MST) – Erinnerung

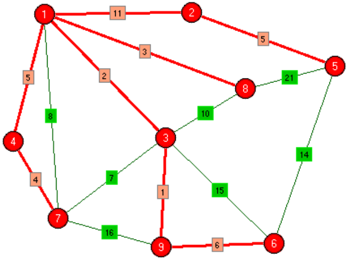
Spannbaum

Gegeben einen zusammenhängenden, ungerichteten Graphen G, so ist ein Spannbaum ein Teilgraph von G, der

(i) ein Baum ist
(ii) alle Knoten von G verbindet

Minimaler Spannbaum

... ist ein Spannbaum mit minimalem Gewicht (für eine gegebene Kantengewichtung von G)



zurück

Flüsse

Gegeben:

- gerichteter Graph
- zwei ausgezeichnete Knoten s (Quelle) und t (Senke).
- Kapazitätsfunktion c, ordnet jeder Kante (u, v) eine reelle Zahl $r \geq 0$ zu (Kapazität)

Fluss

Funktion f, die jeder Kante (u, v) eine nichtnegative reelle Zahl zuordnet, mit der Einschränkung

- Der Fluss auf einer Kante übersteigt nicht deren Kapazität
- Flusserhaltung: der eingehende Fluss eines Knoten entspricht genau dem ausgehenden Fluss (Ausnahmen: Quelle, Senke).

zurück

Flüsse (II)

Komplexität

Der Algorithmus von *Dinic* erreicht $O(|V|^3)$

Maximaler-Fluss Minimaler-Schnitt Theorem

Der maximale Fluss gleicht der Kapazität des minimalen Schnitts. (bewiesen durch P. Elias, A. Feinstein, und C.E. Shannon sowie L.R. Ford, Jr. und D.R. Fulkerson im Jahr 1956)

zurück