Das Rundreiseproblem und Stabilität von Approximationsalgorithmen

Florian Forster

Friedrich Alexander Universität Erlangen-Nürnberg

Seminar „Perlen der theoretischen Informatik“, 2008-01-19
`http://verplant.org/uni/perlen/`

Motivation

- Für viele „spezielle“ Probleme existieren gute Approximationsalgorithmen
- Ist es möglich diese Algorithmen auch für „normale“ Instanzen zu nutzen?
- Wie gut sind diese Lösungen?

Gliederung

Das Rundreiseproblem

Definition

Das Rundreiseproblem wird auch „Traveling Sales**person** Problem" (**TSP**) genannt.

- Sei $G = (V, E)$ ein vollständiger, ungerichteter Graph mit Kantengewichten
- Gesucht ist die kürzeste (billigste) Rundreise, so dass jeder Knoten genau einmal besucht wird (Hamiltonkreis)

Das Rundreiseproblem

NP-Vollständigkeit

- TSP ist **stark** NP-vollständig
- Approximationsalgorithmen mit relativer Güte kann es nur geben, wenn $P = NP$
- Es gibt exakte Algorithmen mit Laufzeit $O(n^2 \cdot 2^n) \ll O(n!)$

Das Rundreiseproblem

Varianten

- Asymmetischrs TSP: Hin- und Rückweg haben nicht notwendigerweise die gleichen Kosten
- TSP mit Dreiecksungleichung (Δ**TSP**): $|a \rightarrow c| \leqq |a \rightarrow b \rightarrow c|$
- euklidisches TSP: Knoten haben euklidische Abstände (beinhaltet Dreiecksungleichung)
- ...

Das Rundreiseproblem

Das ΔTSP

- Für alle Knoten gilt die Dreiecksungleichung
  $\Rightarrow$ „Umwege sind teurer als direkt zu gehen“
- Problem immernoch NP-schwer
- Approximationsalgorithmen mit relativer Güte existieren:
  2APPR mit Güte 2,
  Christofides [2] mit Güte $\frac{3}{2}$
- Diese Algorithmen werden im Folgenden kurz vorgestellt

Gliederung

Approximationsalgorithmen für ΔTSP

Approximationsalgorithmen

- Approximationsalgorithmen für Optimierungsprobleme berechnen „möglichst gute“ Lösungen
- Die Abweichung vom Optimum ist nicht beliebig groß
- Laufzeit ist aber (verhältnismäßig) gering (also: polynomiell)

Approximationsalgorithmen für ΔTSP

Approximationsgüte

- Das Verhältnis von schlechtester approximierter zur optimalen Lösung heißt **Approximationsgüte**
- Zum Beispiel:
  - Optimale Lösung/Rundreise: 100
  - Approximierte Lösung/Rundreise: 150
  - $\Rightarrow$ Güte mindestens 1,5

Approximationsalgorithmen für ΔTSP

Approximationsgüte

- Das Verhältnis von schlechtester approximierter zur optimalen Lösung heißt **Approximationsgüte**
- Zum Beispiel:
  - Optimale Lösung/Rundreise: 100
  - Approximierte Lösung/Rundreise: 150
  - $\Rightarrow$ Güte mindestens 1,5

Approximationsalgorithmen für ΔTSP

Approximationsgüte

- Das Verhältnis von schlechtester approximierter zur optimalen Lösung heißt **Approximationsgüte**
- Zum Beispiel:
  - Optimale Lösung/Rundreise: 100
  - Approximierte Lösung/Rundreise: 150
  - $\Rightarrow$ Güte mindestens 1,5

Approximationsalgorithmen für ΔTSP

Approximationsgüte

- Das Verhältnis von schlechtester approximierter zur optimalen Lösung heißt **Approximationsgüte**
- Zum Beispiel:
  - Optimale Lösung/Rundreise: 100
  - Approximierte Lösung/Rundreise: 150
  - $\Rightarrow$ Güte mindestens 1,5

Approximationsalgorithmen für ΔTSP

Approximationsgüte

- Das Verhältnis von schlechtester approximierter zur optimalen Lösung heißt **Approximationsgüte**
- Zum Beispiel:
  - Optimale Lösung/Rundreise: 100
  - Approximierte Lösung/Rundreise: 150
  - $\Rightarrow$ Güte mindestens 1,5

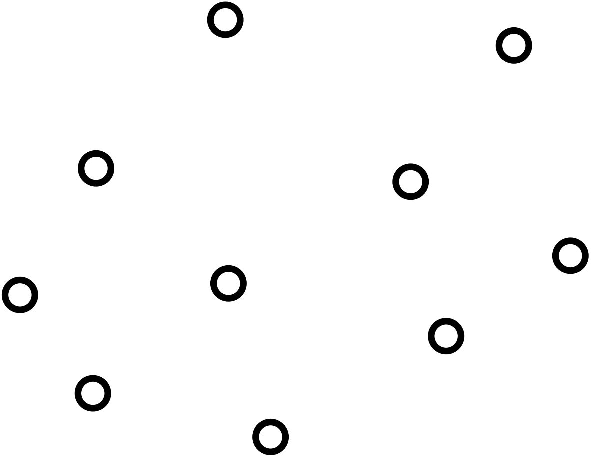
Approximationsalgorithmen für ΔTSP

2APPR

- Sei H ein minimaler Spannbaum von G
- Berechne eine Eulertour auf $H \uplus H$; überspringe doppelte Knoten
- Approximationsgüte 2 folgt aus der Minimalität des Spannbaums und der Dreiecksungleichung

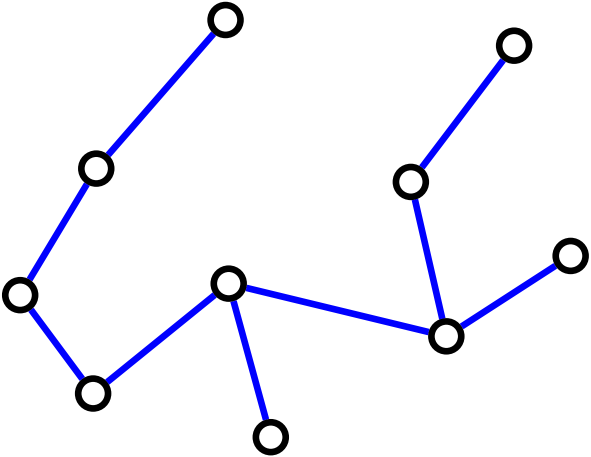
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



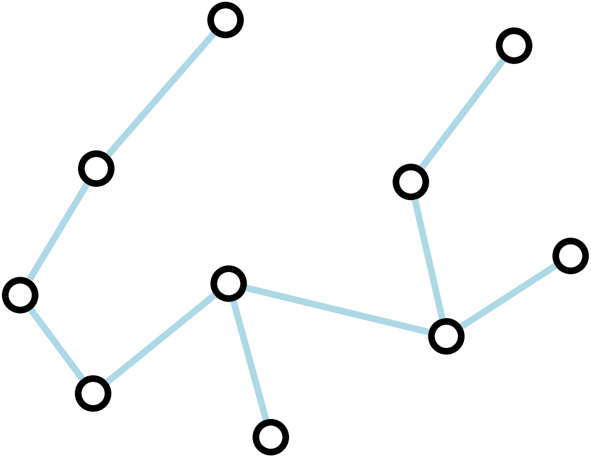
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



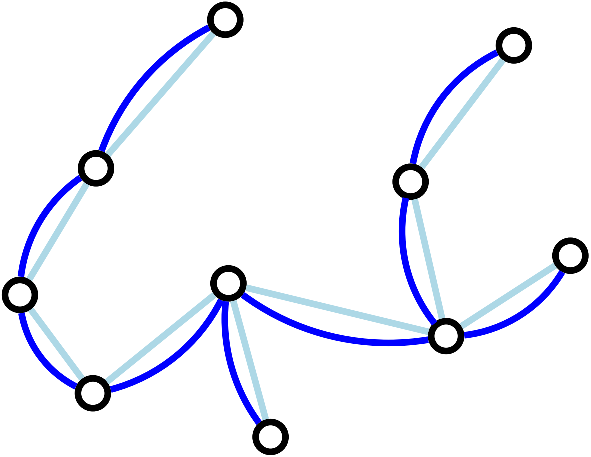
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



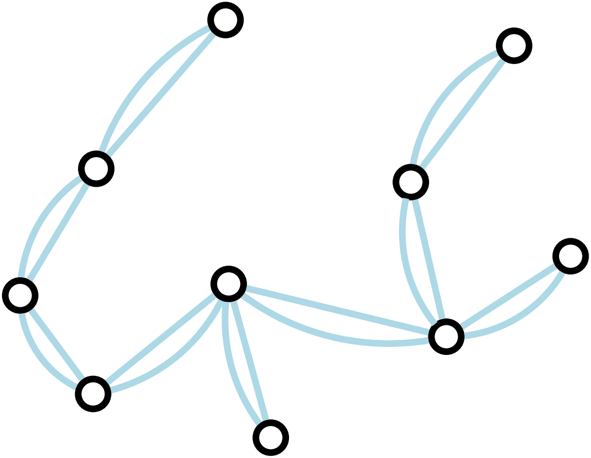
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



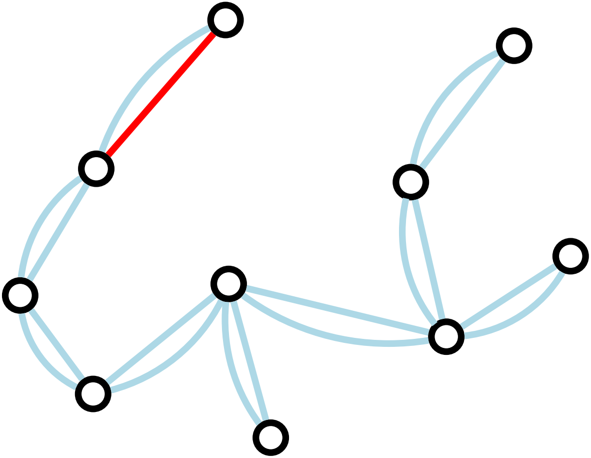
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



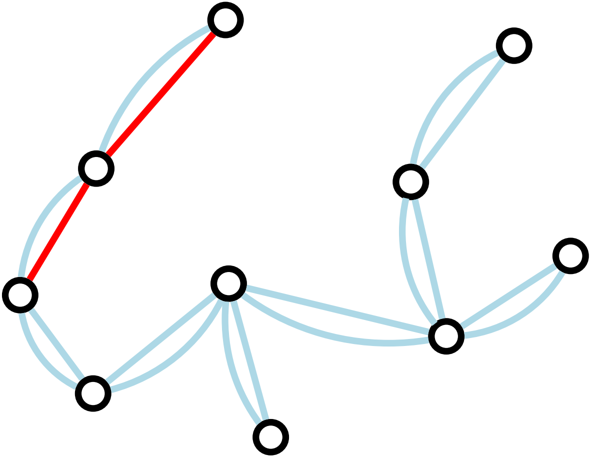
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



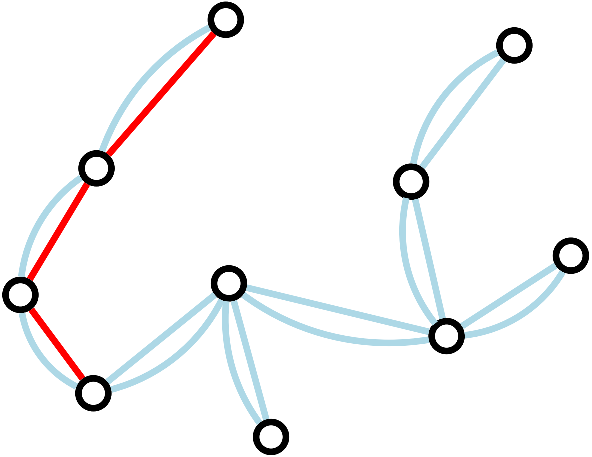
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



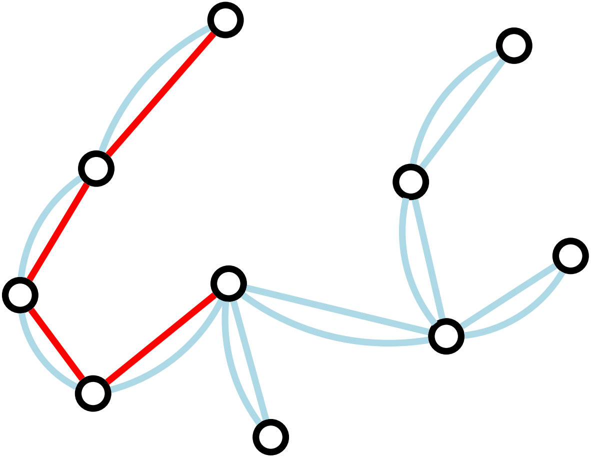
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



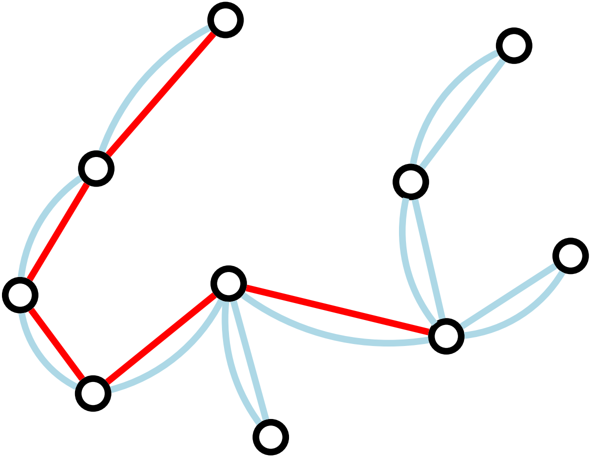
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



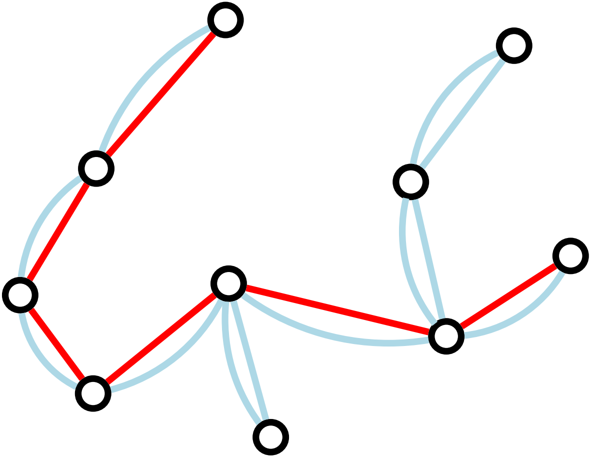
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



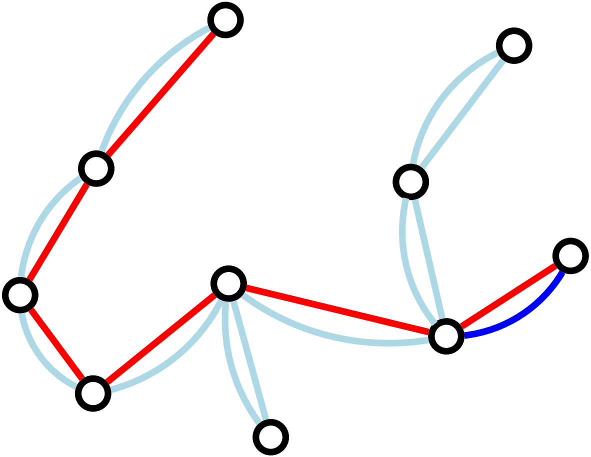
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



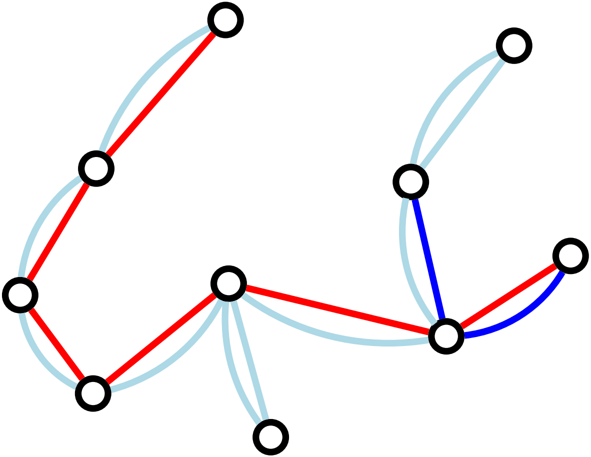
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



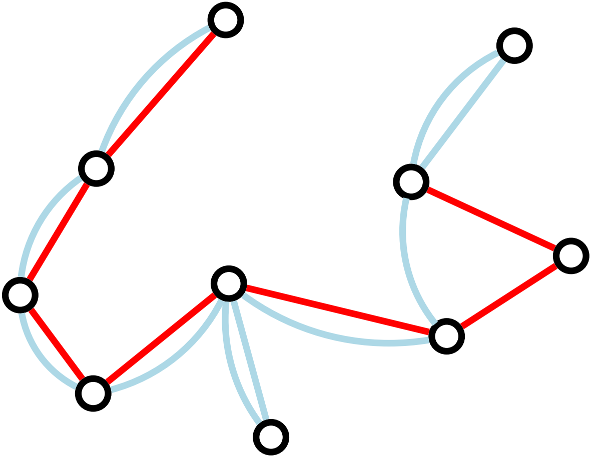
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



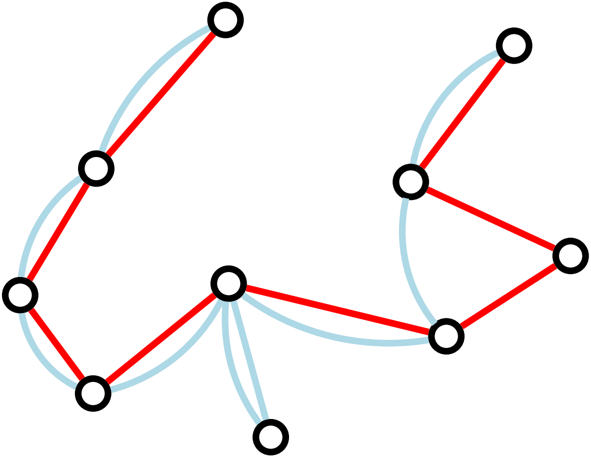
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



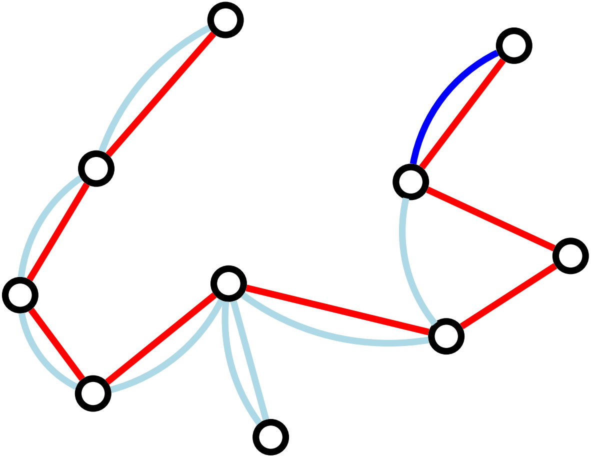
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



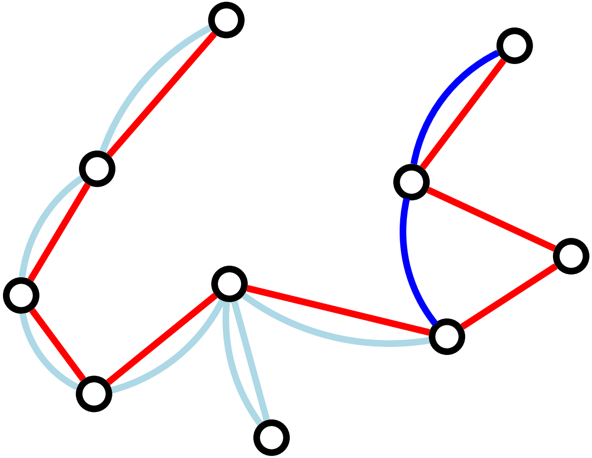
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



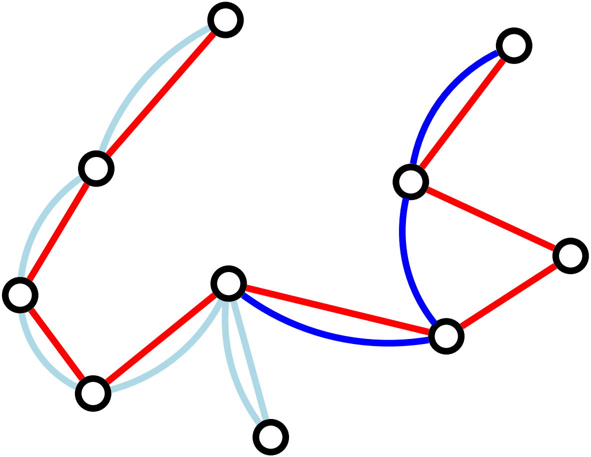
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



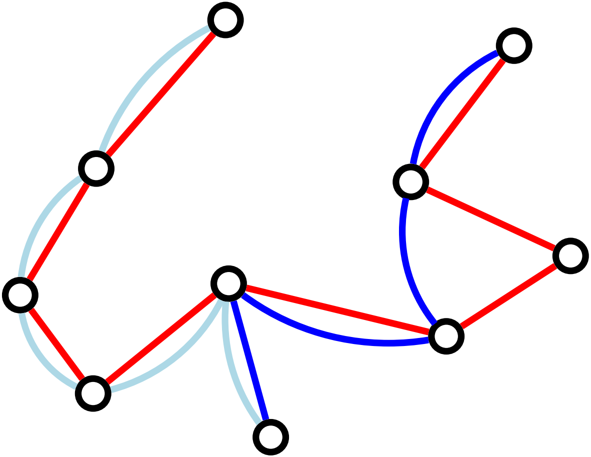
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



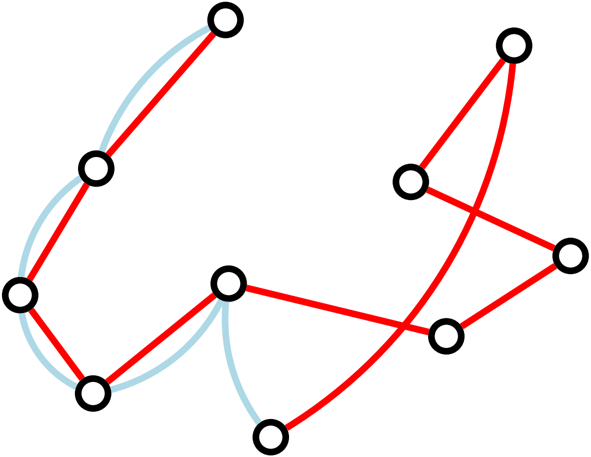
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



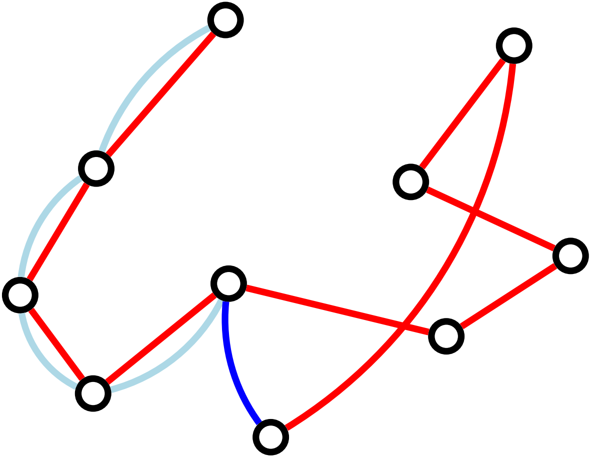
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



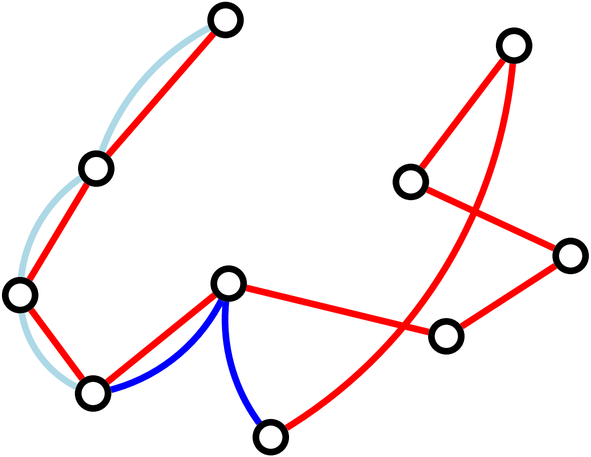
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



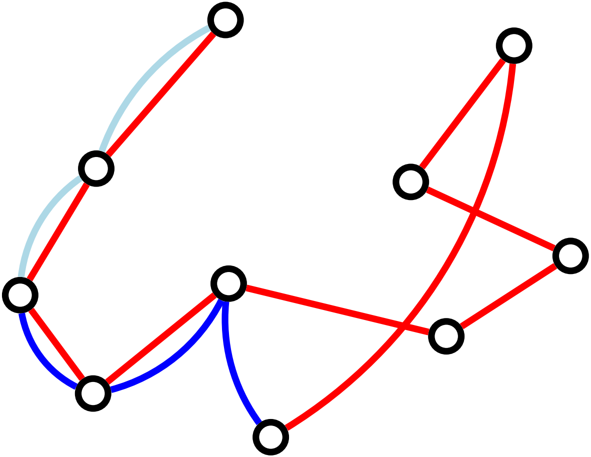
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



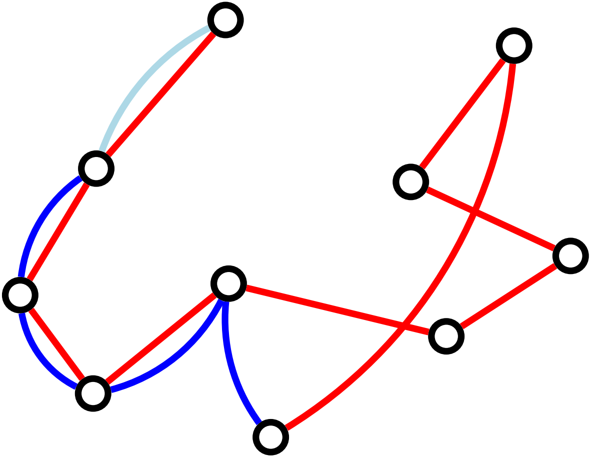
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



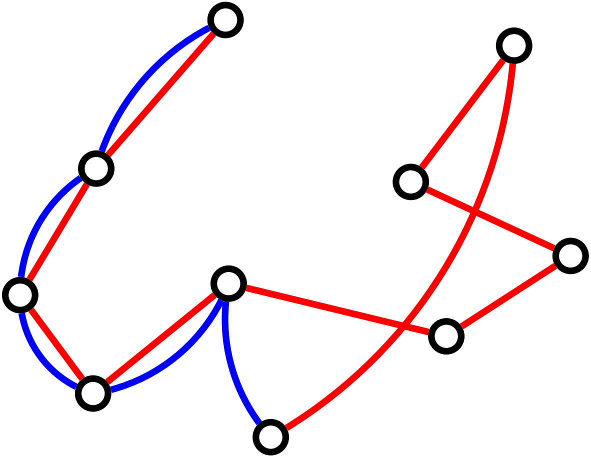
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



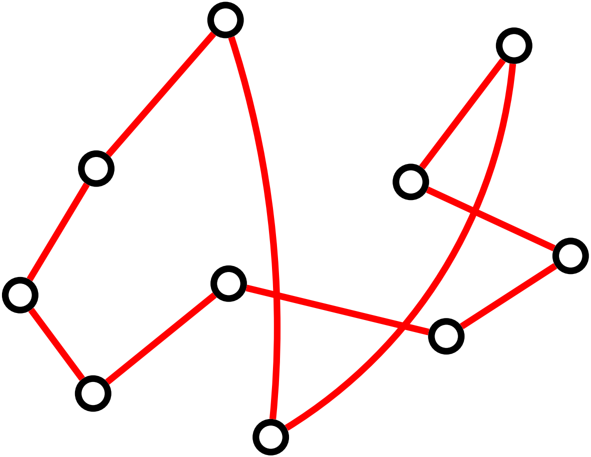
Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



Approximationsalgorithmen für ΔTSP

Beispiel für 2APPR



Approximationsalgorithmen für ΔTSP

Beweis für die Güte von 2APPR

2APPR besitzt eine relative Güte von 2:

- Die Kosten des minimalen Spannbaum sind kleiner als die kosten der optimalen Rundreise R_{OPT}:

$$\mathrm{cost}(H) \leqq \left(1 - \frac{1}{|V|}\right) \cdot \mathrm{cost}(R_{\mathrm{OPT}})$$

- Durch das überspringen von Kanten wird die Strecke nicht länger (Dreickesungleichung):

$$\mathrm{cost}\left(A(x)\right) \leqq 2 \cdot \mathrm{cost}(H) \lneqq 2 \cdot \mathrm{cost}(R_{\mathrm{OPT}})$$

Approximationsalgorithmen für ΔTSP

Beweis für die Güte von 2APPR

2APPR besitzt eine relative Güte von 2:

- Die Kosten des minimalen Spannbaum sind kleiner als die kosten der optimalen Rundreise R_{OPT}:

$$\mathrm{cost}(H) \leqq \left(1 - \frac{1}{|V|}\right) \cdot \mathrm{cost}(R_{\mathrm{OPT}})$$

- Durch das überspringen von Kanten wird die Strecke nicht länger (Dreickesungleichung):

$$\mathrm{cost}\,(A(x)) \leqq 2 \cdot \mathrm{cost}(H) \lneqq 2 \cdot \mathrm{cost}(R_{\mathrm{OPT}})$$

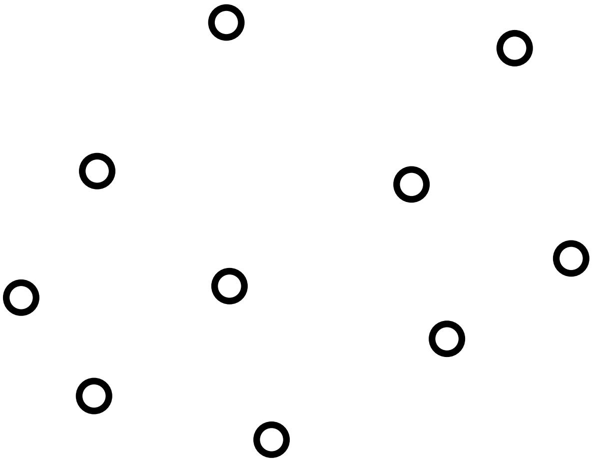
Approximationsalgorithmen für ΔTSP

Christofides

- Sei H ein minimaler Spannbaum von G
- Sei V_u die Menge aller Knoten in H mit ungeradem Grad
- Sei M ein leichtestes Matching aller Knoten in V_u
- Berechne eine Eulertour auf $H \cup M$; überspringe doppelte Knoten
- Approximationsgüte $\frac{3}{2}$ folgt aus der Minimalität des Spannbaums und des leichtesten Matchings sowie der Dreiecksungleichung

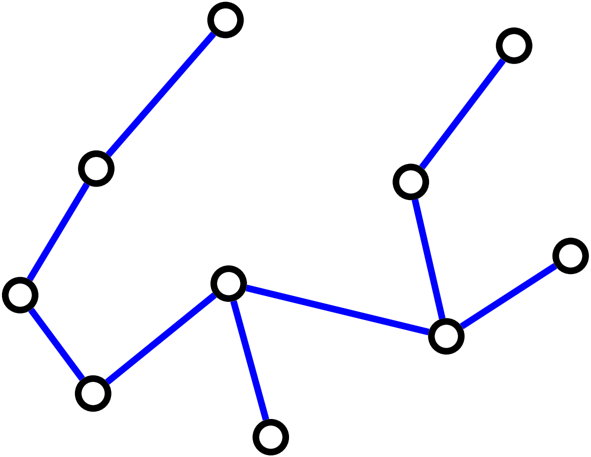
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



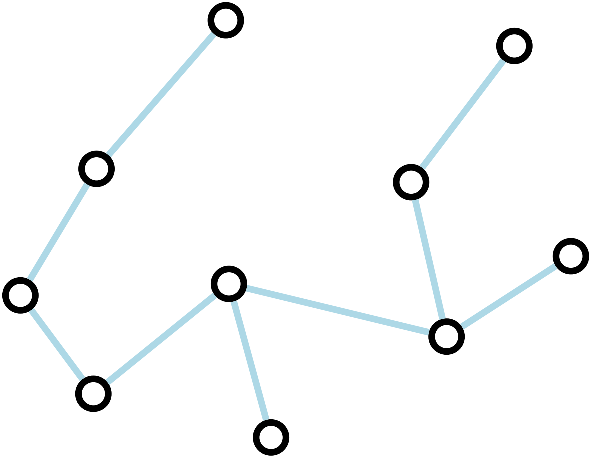
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



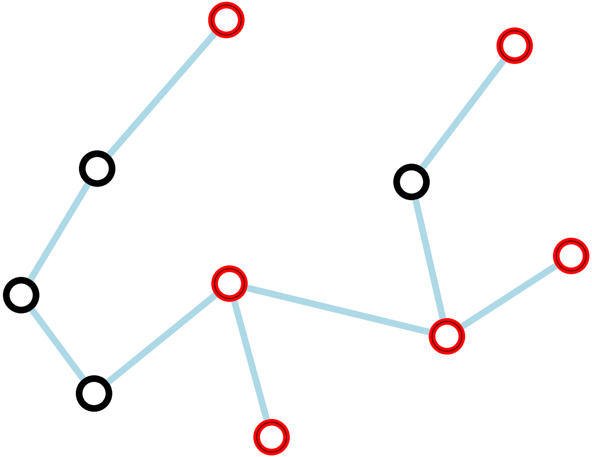
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



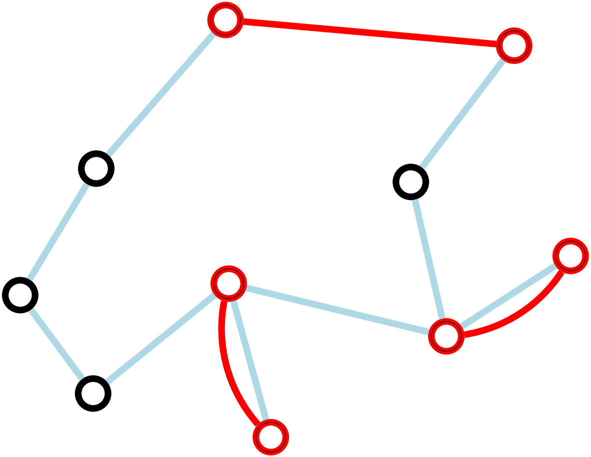
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



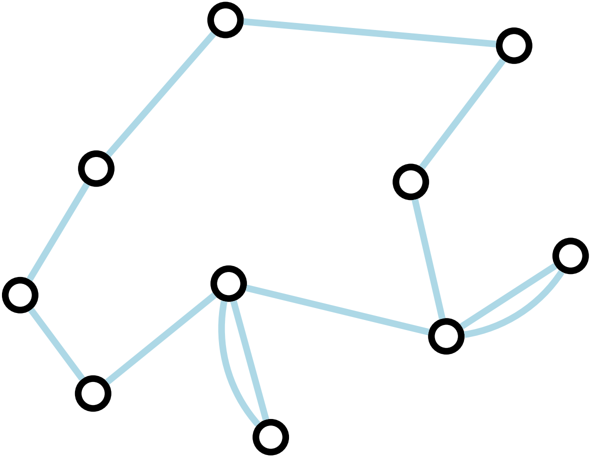
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



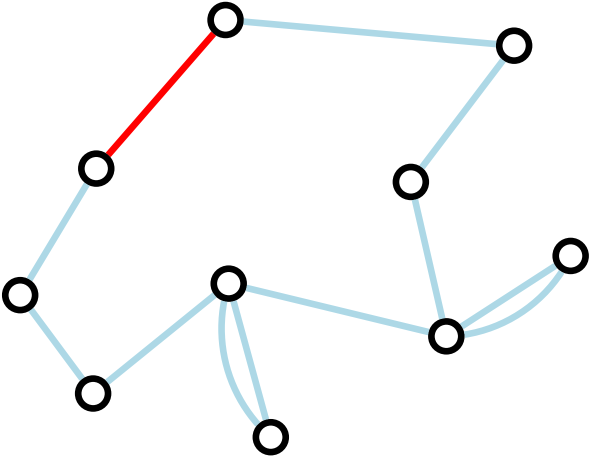
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



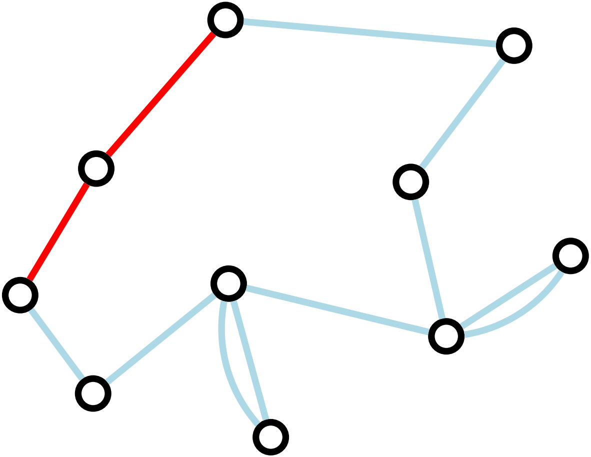
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



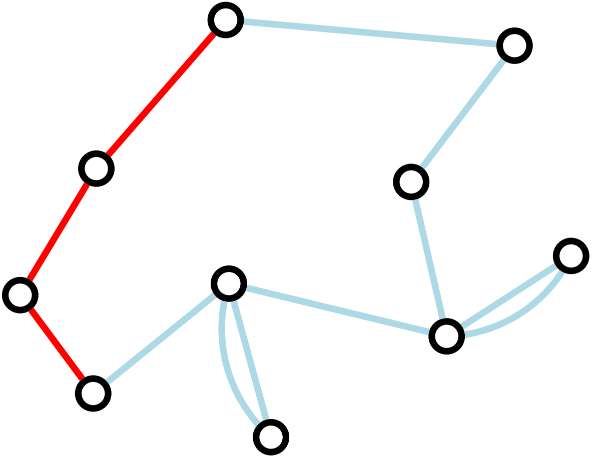
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



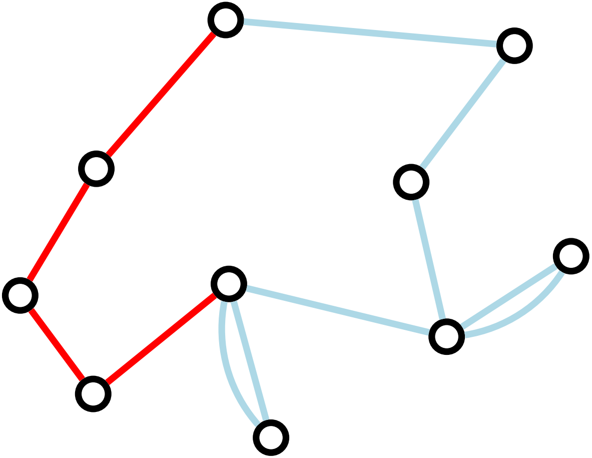
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



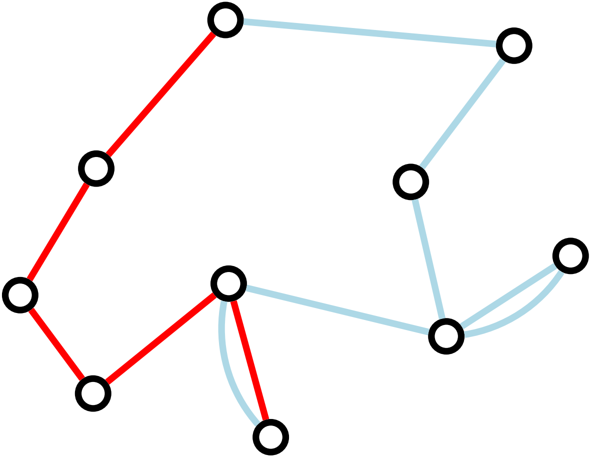
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



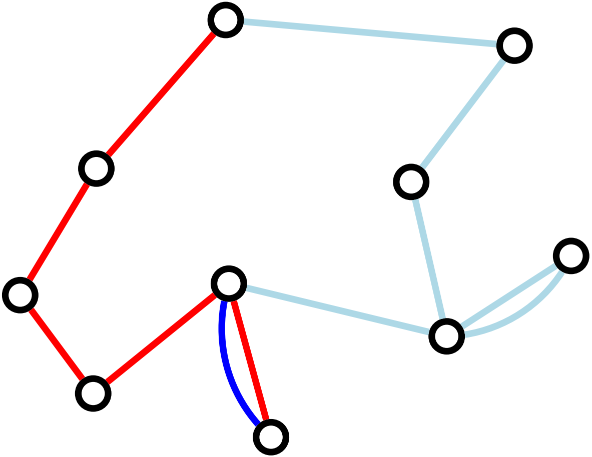
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



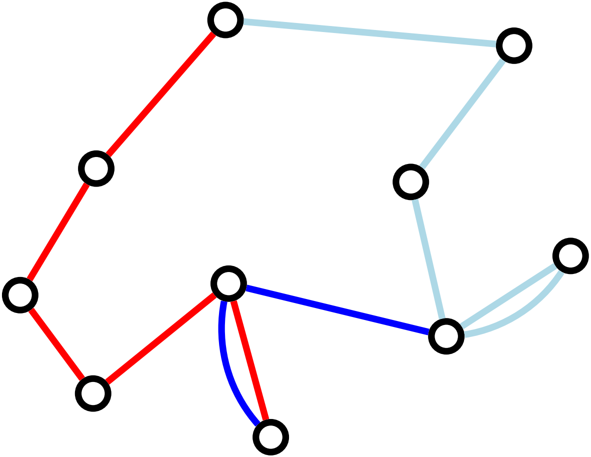
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



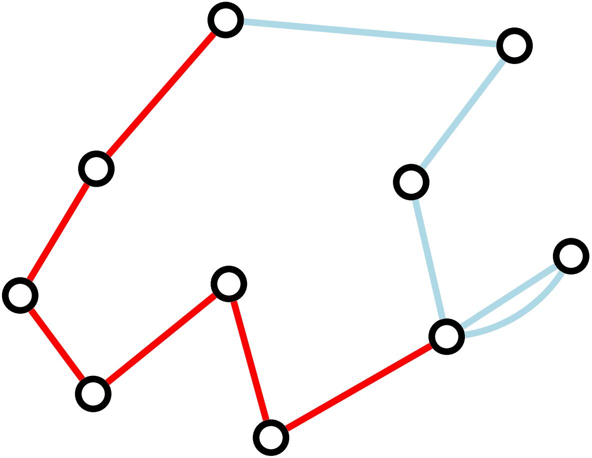
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



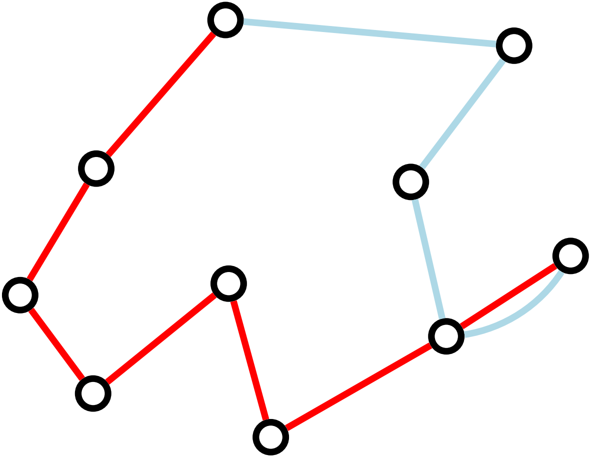
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



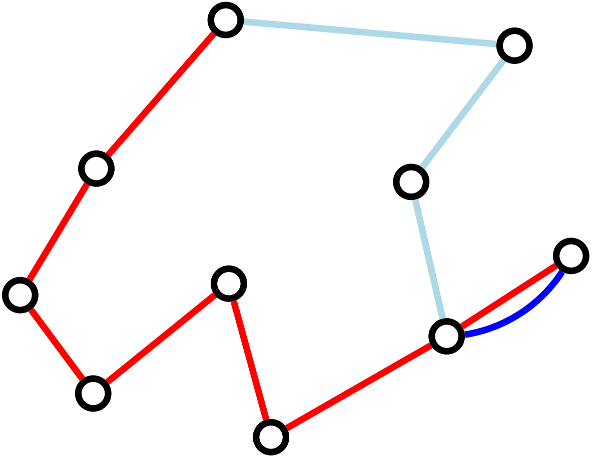
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



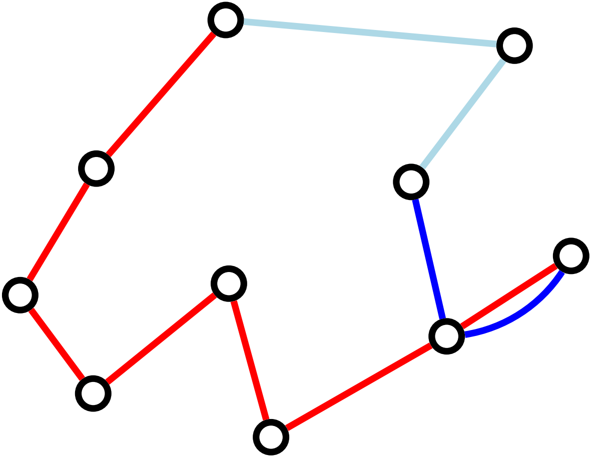
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



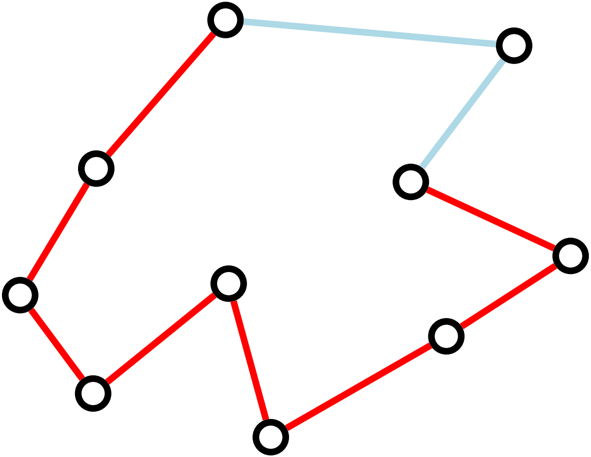
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



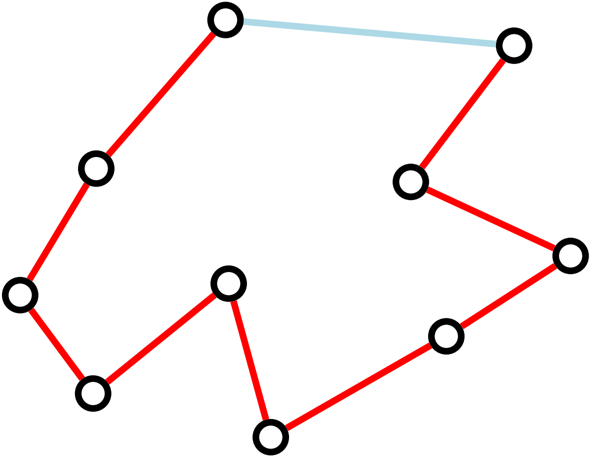
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



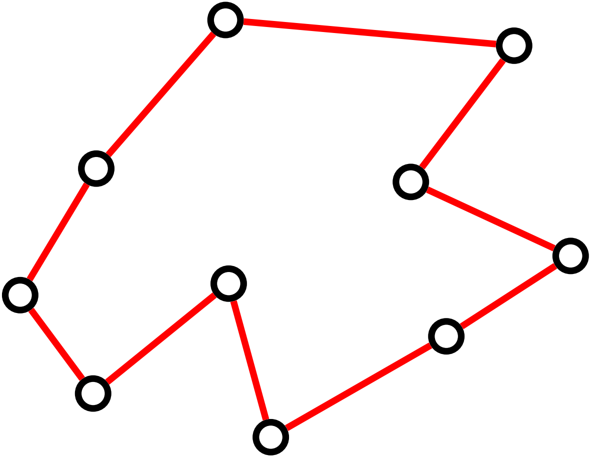
Approximationsalgorithmen für ΔTSP

Beispiel für Christofides



Approximationsalgorithmen für ΔTSP

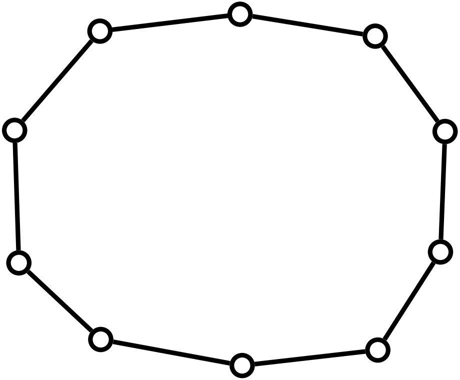
Beispiel für Christofides



Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

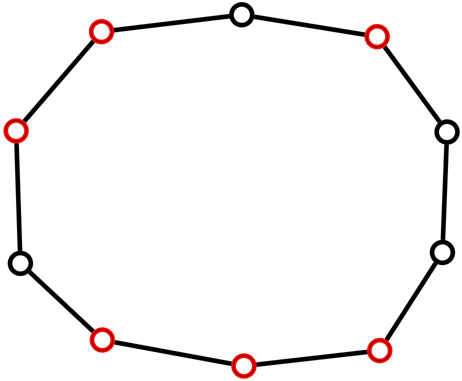
Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$:



Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

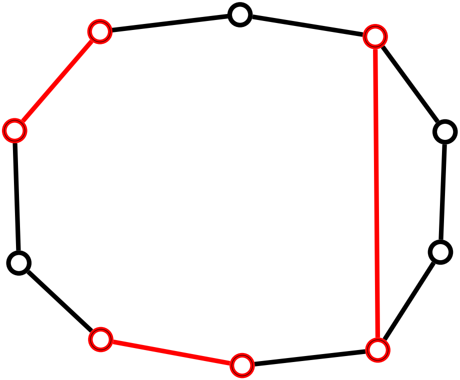
Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$:



Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

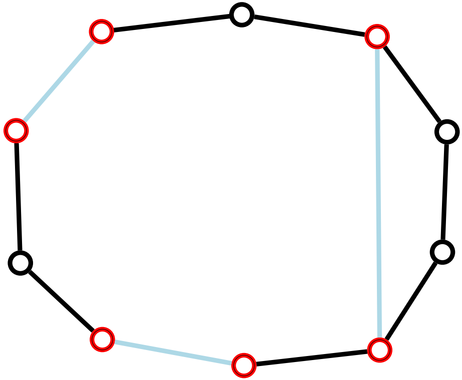
Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$:



Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

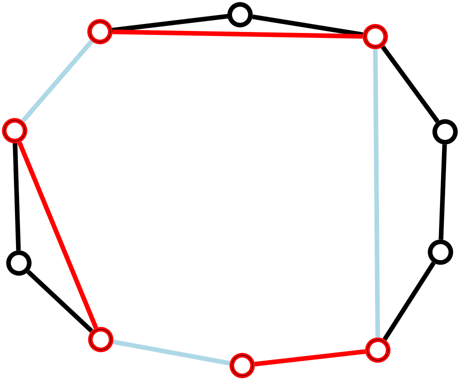
Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$:



Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$:



Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

Christofides besitzt eine relative Güte von $\frac{3}{2}$:

- Die Kosten des minimalen Spannbaum sind kleiner als die kosten der optimalen Rundreise R_{OPT}
- Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$
- Durch das überspringen von Kanten wird die Strecke nicht länger (Dreickesungleichung):

$$\mathrm{cost}\,(A(x)) \leq \mathrm{cost}(H) + \mathrm{cost}(M) \lneqq \frac{3}{2} \cdot \mathrm{cost}(R_{\mathrm{OPT}})$$

Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

Christofides besitzt eine relative Güte von $\frac{3}{2}$:

- Die Kosten des minimalen Spannbaum sind kleiner als die kosten der optimalen Rundreise R_{OPT}
- Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$
- Durch das überspringen von Kanten wird die Strecke nicht länger (Dreickesungleichung):

$$\mathrm{cost}\,(A(x)) \leq \mathrm{cost}(H) + \mathrm{cost}(M) \lneqq \frac{3}{2} \cdot \mathrm{cost}(R_{\mathrm{OPT}})$$

Approximationsalgorithmen für ΔTSP

Beweis für die Güte von Christofides

Christofides besitzt eine relative Güte von $\frac{3}{2}$:

- Die Kosten des minimalen Spannbaum sind kleiner als die kosten der optimalen Rundreise R_{OPT}
- Die Kosten des leichtesten Matchings sind kleiner als $\frac{1}{2} \cdot R_{\mathrm{OPT}}$
- Durch das überspringen von Kanten wird die Strecke nicht länger (Dreickesungleichung):

$$\mathrm{cost}\,(A(x)) \leqq \mathrm{cost}(H) + \mathrm{cost}(M) \lneqq \frac{3}{2} \cdot \mathrm{cost}(R_{\mathrm{OPT}})$$

Approximationsalgorithmen für ΔTSP

Wichtige Bemerkung

- Die Algorithmen selbst haben die Dreiecksungleichung **nicht** verwendet. Lediglich die Abschätzung der Approximationsgüte verwendet die Δ-Ungleichung!

Gliederung

Stabilität von Approximationsalgorithmen

Definition 1/2

- Definition nach [1]
- Sei L_ϕ eine Spezialisierung des (Optimierungs)Problems L
- Sei $A_\phi(x)$ ein Approximationsalgorithmus für L_ϕ mit relativer Güte δ_ϕ
- Sei $h_\phi(x)$ eine **Abstandsfunktion** für die gilt:
  - $h_\phi(x) = 0$ für $x \in L_\phi$
  - $h_\phi(x)$ effizient berechenbar
- Sei $L_{\phi,h,r}$ die Menge aller Probleminstanzen, deren Abstand (bezüglich h_ϕ) kleiner oder gleich r ist:

$$L_{\phi,h,r} = \{x \in L : h_\phi(x) \leq r\}$$

Stabilität von Approximationsalgorithmen

Definition 1/2

- Definition nach [1]
- Sei L_ϕ eine Spezialisierung des (Optimierungs)Problems L
- Sei $A_\phi(x)$ ein Approximationsalgorithmus für L_ϕ mit relativer Güte δ_ϕ
- Sei $h_\phi(x)$ eine **Abstandsfunktion** für die gilt:
  - $h_\phi(x) = 0$ für $x \in L_\phi$
  - $h_\phi(x)$ effizient berechenbar
- Sei $L_{\phi,h,r}$ die Menge aller Probleminstanzen, deren Abstand (bezüglich h_ϕ) kleiner oder gleich r ist:

$$L_{\phi,h,r} = \{x \in L : h_\phi(x) \leq r\}$$

Stabilität von Approximationsalgorithmen

Definition 1/2

- Definition nach [1]
- Sei L_ϕ eine Spezialisierung des (Optimierungs)Problems L
- Sei $A_\phi(x)$ ein Approximationsalgorithmus für L_ϕ mit relativer Güte δ_ϕ
- Sei $h_\phi(x)$ eine **Abstandsfunktion** für die gilt:
  - $h_\phi(x) = 0$ für $x \in L_\phi$
  - $h_\phi(x)$ effizient berechenbar
- Sei $L_{\phi,h,r}$ die Menge aller Probleminstanzen, deren Abstand (bezüglich h_ϕ) kleiner oder gleich r ist:

$$L_{\phi,h,r} = \{x \in L : h_\phi(x) \leq r\}$$

Stabilität von Approximationsalgorithmen

Definition 1/2

- Definition nach [1]
- Sei L_ϕ eine Spezialisierung des (Optimierungs)Problems L
- Sei $A_\phi(x)$ ein Approximationsalgorithmus für L_ϕ mit relativer Güte δ_ϕ
- Sei $h_\phi(x)$ eine **Abstandsfunktion** für die gilt:
  - $h_\phi(x) = 0$ für $x \in L_\phi$
  - $h_\phi(x)$ effizient berechenbar
- Sei $L_{\phi,h,r}$ die Menge aller Probleminstanzen, deren Abstand (bezüglich h_ϕ) kleiner oder gleich r ist:

$$L_{\phi,h,r} = \{x \in L : h_\phi(x) \leq r\}$$

Stabilität von Approximationsalgorithmen

Definition 1/2

- Definition nach [1]
- Sei L_ϕ eine Spezialisierung des (Optimierungs)Problems L
- Sei $A_\phi(x)$ ein Approximationsalgorithmus für L_ϕ mit relativer Güte δ_ϕ
- Sei $h_\phi(x)$ eine **Abstandsfunktion** für die gilt:
  - $h_\phi(x) = 0$ für $x \in L_\phi$
  - $h_\phi(x)$ effizient berechenbar
- Sei $L_{\phi,h,r}$ die Menge aller Probleminstanzen, deren Abstand (bezüglich h_ϕ) kleiner oder gleich r ist:

$$L_{\phi,h,r} = \{x \in L : h_\phi(x) \leqq r\}$$

Stabilität von Approximationsalgorithmen

Definition 2/2

- $A_\phi(x)$ heißt *p*-**stabil bezüglich** h_ϕ, wenn für jedes $r, 0 \lneqq r \leqq p$ ein $\delta_{\phi,r} \in \mathbb{R}^{>1}$ existiert, so dass $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $\delta_{\phi,r}$ ist.
- $A_\phi(x)$ heißt **stabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **alle** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt **instabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **kein** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt $(r, f_r(n))$-**quasistabil bezüglich** h_ϕ, wenn $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $f_r(n)$ ist.

Stabilität von Approximationsalgorithmen

Definition 2/2

- $A_\phi(x)$ heißt p-**stabil bezüglich** h_ϕ, wenn für jedes $r, 0 \lneqq r \leqq p$ ein $\delta_{\phi,r} \in \mathbb{R}^{>1}$ existiert, so dass $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $\delta_{\phi,r}$ ist.
- $A_\phi(x)$ heißt **stabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **alle** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt **instabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **kein** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt $(r, f_r(n))$-**quasistabil bezüglich** h_ϕ, wenn $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $f_r(n)$ ist.

Stabilität von Approximationsalgorithmen

Definition 2/2

- $A_\phi(x)$ heißt ***p*-stabil bezüglich** h_ϕ, wenn für jedes $r, 0 \lneqq r \leqq p$ ein $\delta_{\phi,r} \in \mathbb{R}^{>1}$ existiert, so dass $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $\delta_{\phi,r}$ ist.
- $A_\phi(x)$ heißt **stabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **alle** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt **instabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **kein** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt $(r, f_r(n))$**-quasistabil bezüglich** h_ϕ, wenn $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $f_r(n)$ ist.

Stabilität von Approximationsalgorithmen

Definition 2/2

- $A_\phi(x)$ heißt *p*-**stabil bezüglich** h_ϕ, wenn für jedes $r, 0 \lneqq r \leqq p$ ein $\delta_{\phi,r} \in \mathbb{R}^{>1}$ existiert, so dass $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $\delta_{\phi,r}$ ist.
- $A_\phi(x)$ heißt **stabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **alle** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt **instabil bezüglich** h_ϕ, wenn $A_\phi(x)$ für **kein** $p \in \mathbb{R}^+$ p-stabil bezüglich h_ϕ ist.
- $A_\phi(x)$ heißt $(r, f_r(n))$-**quasistabil bezüglich** h_ϕ, wenn $A_\phi(x)$ ein Approximationsalgorithmus für $L_{\phi,h,r}$ mir relativer Güte $f_r(n)$ ist.

Abstandsfunktionen für ΔTSP

Definition

- Funktion $\textsc{Dist}_{\text{three}} =$

$$\max\left\{0, \max\left\{\frac{\text{cost}(u,v)}{\text{cost}(u,w)+\text{cost}(w,v)} - 1, u,v,w \in V\right\}\right\}$$

- Sei $W_{u,v} = \{w_1 = u, w_2, \ldots, w_m = v\}$ ein einfacher Pfad von u nach v.
  Funktion $\textsc{Dist}_{\text{path}} =$

$$\max\left\{0, \max\left\{\frac{\text{cost}(u,v)}{\text{cost}(W_{u,v})} - 1, u,v \in V\right\}\right\}$$

Abstandsfunktionen für ΔTSP

Definition

- Funktion $\textsc{Dist}_{\text{three}} =$

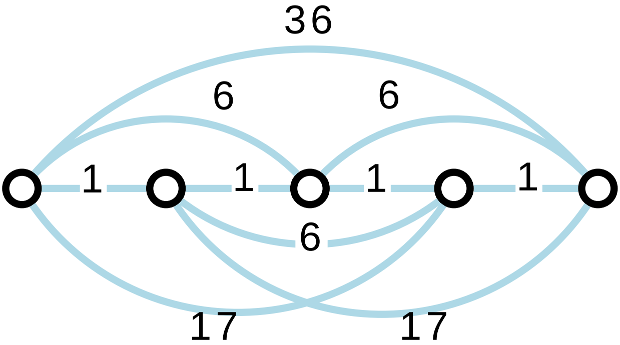
$$\max\left\{0, \max\left\{\frac{\text{cost}(u,v)}{\text{cost}(u,w)+\text{cost}(w,v)} - 1, u,v,w \in V\right\}\right\}$$

- Sei $W_{u,v} = \{w_1 = u, w_2, \dots, w_m = v\}$ ein einfacher Pfad von u nach v.
  Funktion $\textsc{Dist}_{\text{path}} =$

$$\max\left\{0, \max\left\{\frac{\text{cost}(u,v)}{\text{cost}(W_{u,v})} - 1, u,v \in V\right\}\right\}$$

Abstandsfunktionen für ΔTSP

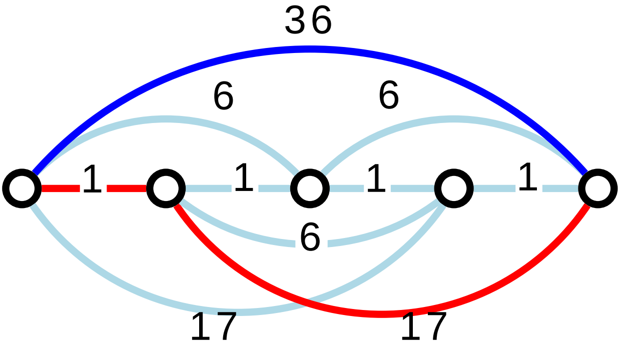
Beispiel: Abstandsfunktion $\textsc{Dist}_{\text{three}}$



$\textsc{Dist}_{\text{three}} = 0$

Abstandsfunktionen für ΔTSP

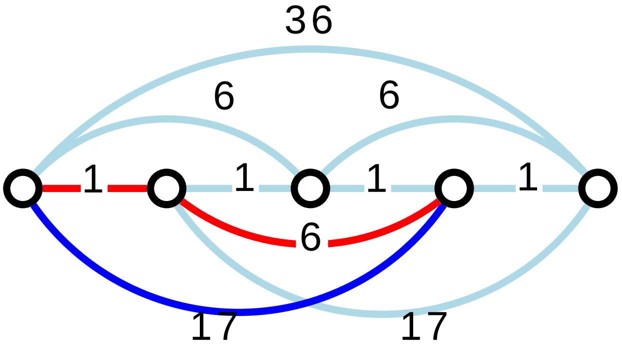
Beispiel: Abstandsfunktion $\text{DIST}_{\text{three}}$



$$\text{DIST}_{\text{three}} = \frac{36}{18} - 1 = 1$$

Abstandsfunktionen für ΔTSP

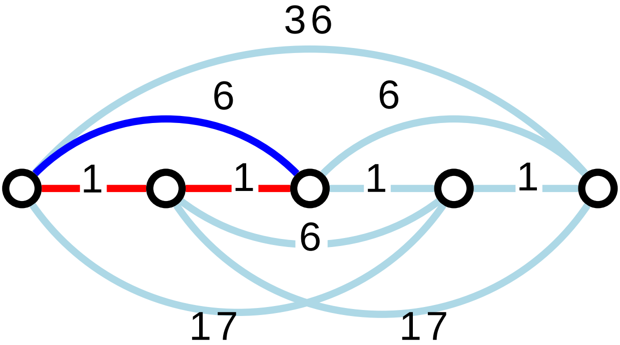
Beispiel: Abstandsfunktion $\mathrm{DIST}_{\mathrm{three}}$



$$\mathrm{DIST}_{\mathrm{three}} = \frac{17}{7} - 1 = 1\frac{3}{7}$$

Abstandsfunktionen für ΔTSP

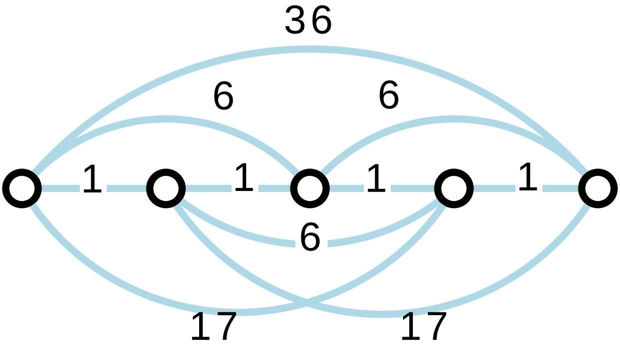
Beispiel: Abstandsfunktion $\textsc{Dist}_{\text{three}}$



$$\textsc{Dist}_{\text{three}} = \frac{6}{2} - 1 = 2$$

Abstandsfunktionen für ΔTSP

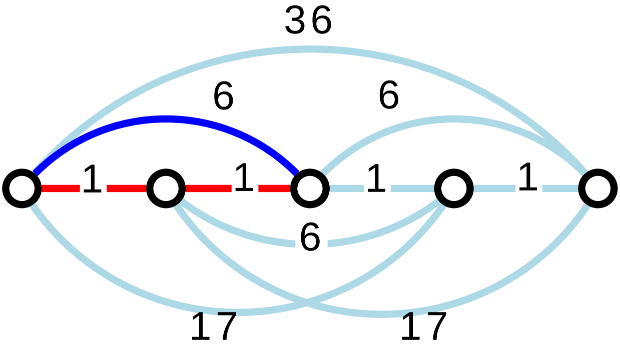
Beispiel: Abstandsfunktion $\textsc{Dist}_{\text{path}}$



$\textsc{Dist}_{\text{path}} = 0$

Abstandsfunktionen für ΔTSP

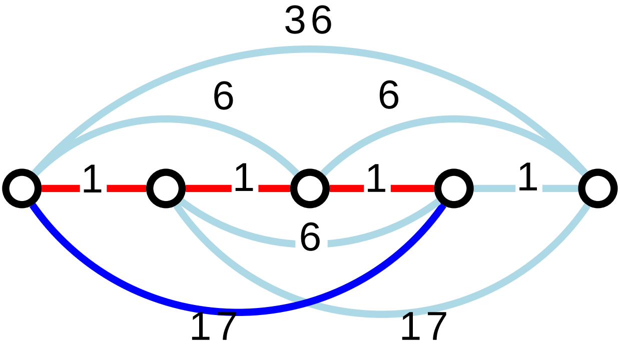
Beispiel: Abstandsfunktion $\textsc{Dist}_{\text{path}}$



$$\textsc{Dist}_{\text{path}} = \frac{6}{2} - 1 = 2$$

Abstandsfunktionen für ΔTSP

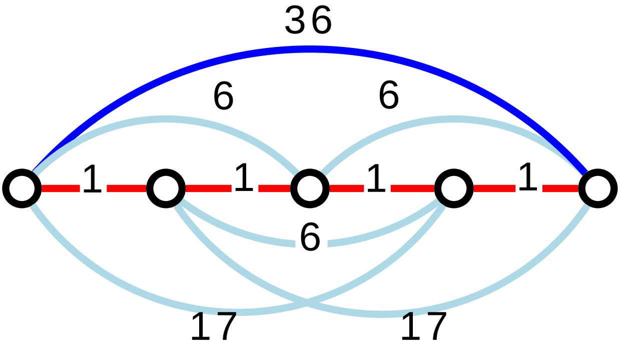
Beispiel: Abstandsfunktion $\textsc{Dist}_{\text{path}}$



$$\textsc{Dist}_{\text{path}} = \frac{17}{3} - 1 = 4\frac{2}{3}$$

Abstandsfunktionen für ΔTSP

Beispiel: Abstandsfunktion $\textsc{Dist}_{\text{path}}$



$$\textsc{Dist}_{\text{path}} = \frac{36}{4} - 1 = 8$$

Stabilität von 2APPR und Christofides

Stabilität bezüglich $\textsc{Dist}_{\text{path}}$

- 2APPR und Christofides sind **stabil bezüglich** $\textsc{Dist}_{\text{path}}$
- Die Kosten des berechneten Hamiltonkreises $v_0 \to v_1 \to \ldots \to v_n \to v_0$ sind maximal $(1+r)$ mal die Kosten von $v_0 \to_p v_1 \to_p \ldots \to_p v_n \to_p v_0$ wobei $u \to_p v$ der kurzeste **Pfad** von u nach v in G ist.
- 2APPR ist ein $(2\,(1+r))$-Approximationsalgorithmus für TSP
- Christofides ist ein $\left(\frac{3}{2}\,(1+r)\right)$-Approximationsalgorithmus für TSP

Stabilität von 2APPR und Christofides

Stabilität bezüglich $\text{DIST}_{\text{path}}$

- 2APPR und Christofides sind **stabil bezüglich** $\text{DIST}_{\text{path}}$
- Die Kosten des berechneten Hamiltonkreises $v_0 \rightarrow v_1 \rightarrow \ldots \rightarrow v_n \rightarrow v_0$ sind maximal $(1 + r)$ mal die Kosten von $v_0 \rightarrow_p v_1 \rightarrow_p \ldots \rightarrow_p v_n \rightarrow_p v_0$ wobei $u \rightarrow_p v$ der kurzeste **Pfad** von u nach v in G ist.
- 2APPR ist ein $(2\,(1 + r))$-Approximationsalgorithmus für TSP
- Christofides ist ein $\left(\frac{3}{2}\,(1 + r)\right)$-Approximationsalgorithmus für TSP

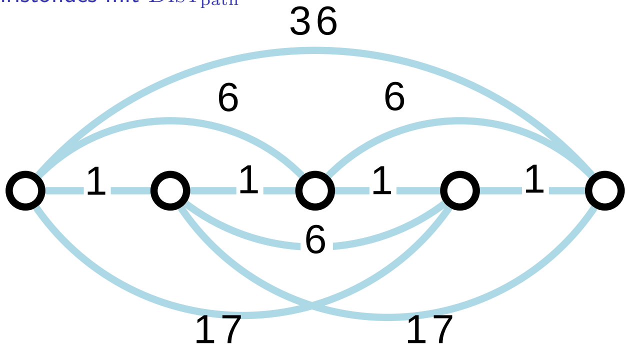
Stabilität von 2APPR und Christofides

Stabilität bezüglich $\mathrm{DIST}_{\mathrm{path}}$

- 2APPR und Christofides sind **stabil bezüglich** $\mathrm{DIST}_{\mathrm{path}}$
- Die Kosten des berechneten Hamiltonkreises $v_0 \to v_1 \to \ldots \to v_n \to v_0$ sind maximal $(1 + r)$ mal die Kosten von $v_0 \to_p v_1 \to_p \ldots \to_p v_n \to_p v_0$ wobei $u \to_p v$ der kurzeste **Pfad** von u nach v in G ist.
- 2APPR ist ein $(2\,(1 + r))$-Approximationsalgorithmus für TSP
- Christofides ist ein $\left(\frac{3}{2}\,(1 + r)\right)$-Approximationsalgorithmus für TSP

Stabilität von 2APPR und Christofides

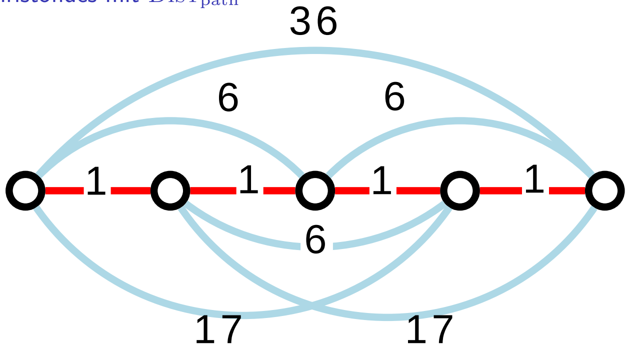
Beispiel für Christofides mit $\text{DIST}_{\text{path}}$



Abstand bzgl. $\text{DIST}_{\text{path}}$: 8

Stabilität von 2APPR und Christofides

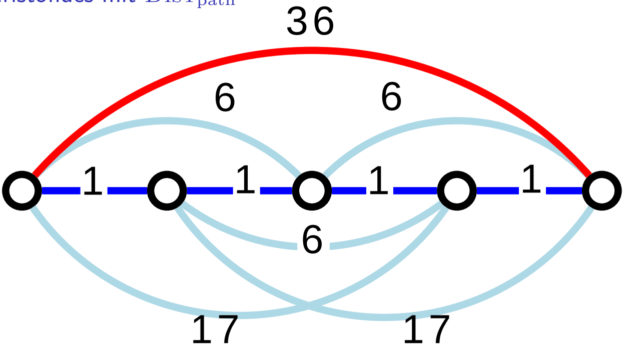
Beispiel für Christofides mit $\textsc{Dist}_{\text{path}}$



Abstand bzgl. $\textsc{Dist}_{\text{path}}$: 8

Stabilität von 2APPR und Christofides

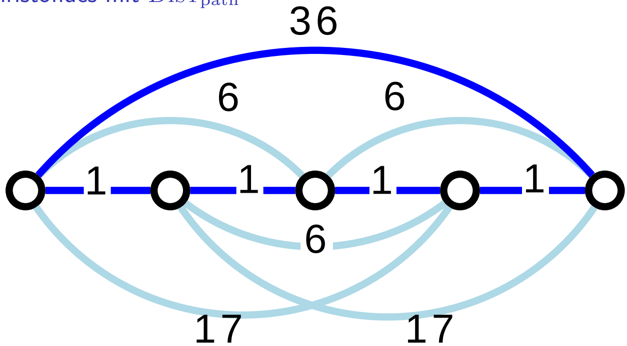
Beispiel für Christofides mit $\textsc{Dist}_{\text{path}}$



Abstand bzgl. $\textsc{Dist}_{\text{path}}$: 8

Stabilität von 2APPR und Christofides

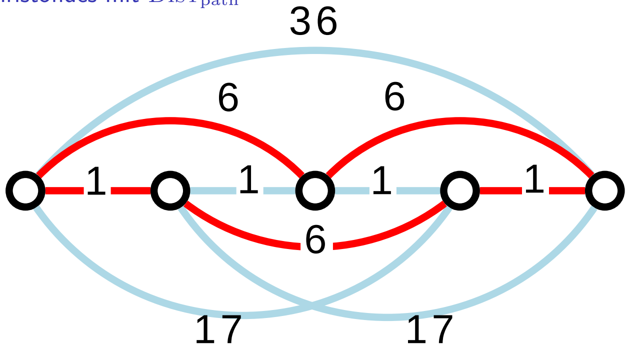
Beispiel für Christofides mit $\textsc{Dist}_{\text{path}}$



Abstand bzgl. $\textsc{Dist}_{\text{path}}$: 8
Lösung von Christofides: 40

Stabilität von 2APPR und Christofides

Beispiel für Christofides mit $\textsc{Dist}_{\text{path}}$



Abstand bzgl. $\textsc{Dist}_{\text{path}}$: 8
Lösung von Christofides: 40
Optimale Rundreise: 20

Stabilität von 2APPR und Christofides

Stabilität bezüglich $\text{DIST}_{\text{three}}$

- Jeder Pfad p mit m Kanten in G kann durch eine einzige Kante e mit

$$\operatorname{cost}(e) \leqq (1+r)^{\lceil \log_2 m \rceil} \cdot \operatorname{cost}(p)$$

ersetzt werden.

Stabilität von 2APPR und Christofides

Stabilität bezüglich $\textsc{Dist}_{\text{three}}$

- Jeder Pfad p mit m Kanten in G kann durch eine einzige Kante e mit

$$\operatorname{cost}(e) \leqq (1+r)^{\lceil \log_2 m \rceil} \cdot \operatorname{cost}(p)$$

ersetzt werden.

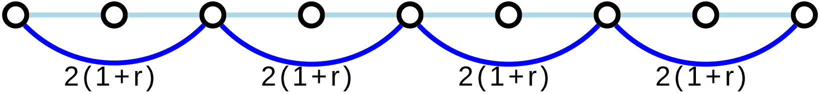
Stabilität von 2APPR und Christofides

Stabilität bezüglich $\textsc{Dist}_{\text{three}}$

- Jeder Pfad p mit m Kanten in G kann durch eine einzige Kante e mit

$$\mathrm{cost}(e) \leqq (1+r)^{\lceil \log_2 m \rceil} \cdot \mathrm{cost}(p)$$

ersetzt werden.



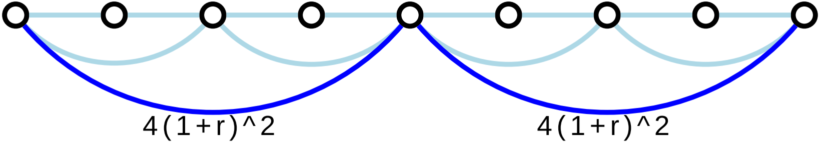
Stabilität von 2APPR und Christofides

Stabilität bezüglich $\textsc{Dist}_{\text{three}}$

- Jeder Pfad p mit m Kanten in G kann durch eine einzige Kante e mit

$$\mathrm{cost}\,(e) \leqq (1+r)^{\lceil \log_2 m \rceil} \cdot \mathrm{cost}\,(p)$$

ersetzt werden.



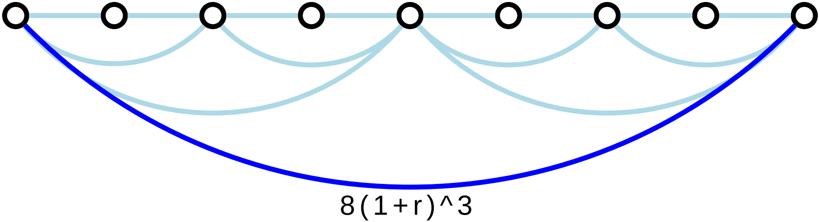
Stabilität von 2APPR und Christofides

Stabilität bezüglich $\text{DIST}_{\text{three}}$

- Jeder Pfad p mit m Kanten in G kann durch eine einzige Kante e mit

$$\mathrm{cost}(e) \leqq (1+r)^{\lceil \log_2 m \rceil} \cdot \mathrm{cost}(p)$$

ersetzt werden.



Stabilität von 2APPR und Christofides

Stabilität bezüglich $\textsc{Dist}_{\text{three}}$

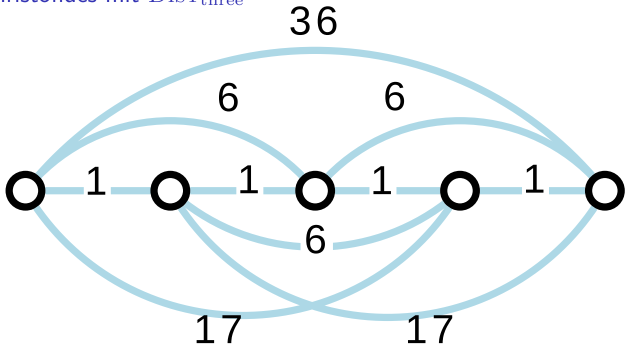
- Jeder Pfad p mit m Kanten in G kann durch eine einzige Kante e mit

$$\mathrm{cost}\,(e) \leqq (1+r)^{\lceil \log_2 m \rceil} \cdot \mathrm{cost}\,(p)$$

  ersetzt werden.
- 2APPR ist $\left(r, 2\,(1+r)^{\lceil \log_2 n \rceil}\right)$-quasistabil bezüglich $\textsc{Dist}_{\text{three}}$
- Christofides ist $\left(r, \frac{3}{2}\,(1+r)^{\lceil \log_2 n \rceil}\right)$-quasistabil bezüglich $\textsc{Dist}_{\text{three}}$
- 2APPR und Christofides sind **instabil bezüglich Dist$_{\text{three}}$**

Stabilität von 2APPR und Christofides

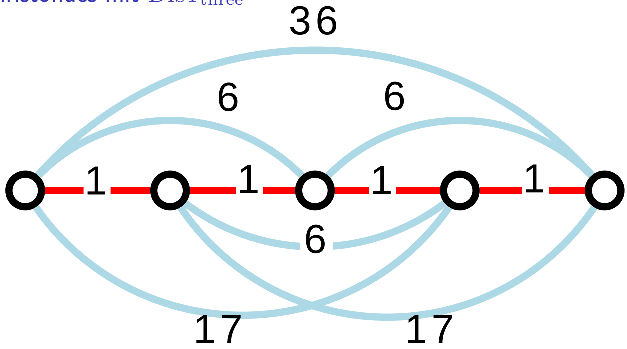
Beispiel für Christofides mit $\textsc{Dist}_{\text{three}}$



Abstand bzgl. $\textsc{Dist}_{\text{three}}$: 2

Stabilität von 2APPR und Christofides

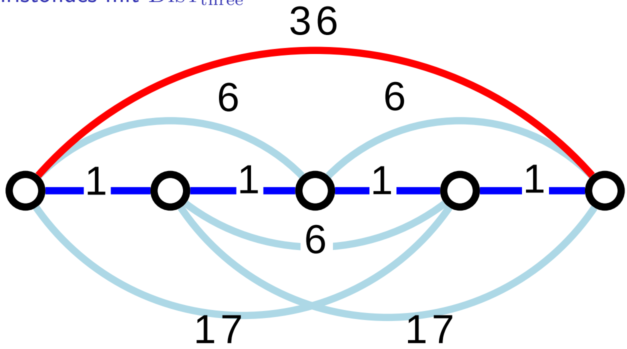
Beispiel für Christofides mit $\textsc{Dist}_{\text{three}}$



Abstand bzgl. $\textsc{Dist}_{\text{three}}$: 2

Stabilität von 2APPR und Christofides

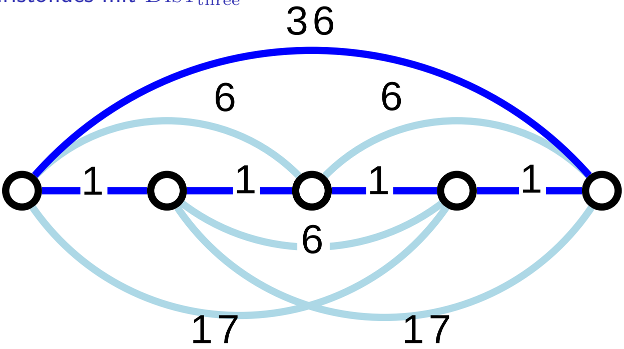
Beispiel für Christofides mit $\textsc{Dist}_{\text{three}}$



Abstand bzgl. $\textsc{Dist}_{\text{three}}$: 2

Stabilität von 2APPR und Christofides

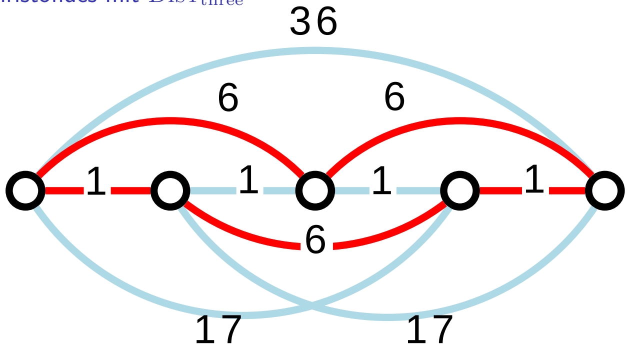
Beispiel für Christofides mit $\textsc{Dist}_{\text{three}}$



Abstand bzgl. $\textsc{Dist}_{\text{three}}$: 2
Lösung von Christofides: 40

Stabilität von 2APPR und Christofides

Beispiel für Christofides mit $\mathrm{DIST}_{\mathrm{three}}$



Abstand bzgl. $\mathrm{DIST}_{\mathrm{three}}$: 2
Lösung von Christofides: 40
Optimale Rundreise: 20

Der PMCA-Algorithmus

Ein stabiler Algorithmus bezüglich $\textsc{Dist}_{\text{three}}$

- In [1] wird der PMCA-Algorithmus vorgestellt
- Der Algorithmus ist stabil bezüglich $\textsc{Dist}_{\text{three}}$
- Erreicht wird das durch ein „Pfad-Matching" und trickreiche „Abkürzungen"
- Die Approximationsgüte ist $\frac{3}{2}(1+r)^2$, die Laufzeit $O(n^3)$

Fazit 1/2

Das Rundreiseproblem

- Das allgemeine TSP kann man nicht gut approximieren
- Für das ΔTSP gibt es gute Approximationsalgorithmen
- Die Algorithmen verwenden die Dreiecksungleichung **nicht**, lediglich die Abschätzung braucht sie

Fazit 2/2

Stabilität von Approximationsalgorithmen

- 2APPR und Christofides sind stabil bezüglich $\textsc{Dist}_{\text{path}}$ aber instabil bezüglich $\textsc{Dist}_{\text{three}}$
- 2APPR und Christofides sind quasistabil bezüglich $\textsc{Dist}_{\text{three}}$
- Mit PMCA existiert ein Approximationsalgorithmus für ΔTSP der stabil bezüglich $\textsc{Dist}_{\text{three}}$ ist

Literatur

Hans-Joachim Böckenhauer, Juraj Hromkovic, Ralf Klasing, Sebastian Seibert, and Walter Unger.
Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem.
In *CIAC '00: Proceedings of the 4th Italian Conference on Algorithms and Complexity*, pages 72–86, London, UK, 2000. Springer-Verlag.

Nicos Christofides.
Worst-case analysis of a new heuristic for the travelling salesman problem.
In *Algorithms and Complexity: New Directions and Recent Results*, page 441. Academic Press, 1976.