Kap. 7.1 Heuristiken
Kap. 7.2 Approximative Algorithmen und Gütegarantien

Professor Dr. Petra Mutzel

Lehrstuhl für Algorithm Engineering, LS11

Fakultät für Informatik, TU Dortmund

23. VO DAP2 SS 2008 14. Juli 2009

Überblick

- Hausaufgabenbesprechung: „schlechte Instanzen finden für Greedy-Heuristiken“ für TSP, Bin Packing, Rucksackproblem

- Kap. 7.2: Approximative Algorithmen
  - Gütegarantien für NN-Heuristik (TSP), FF-Heuristik (Bin Packing), Greedy-Heuristik (Rucksackproblem)
  - Approximative Algorithmen mit Gütegarantie für TSP: ST-Heuristik, CH-Heuristik

Das Travelling Salesman Problem

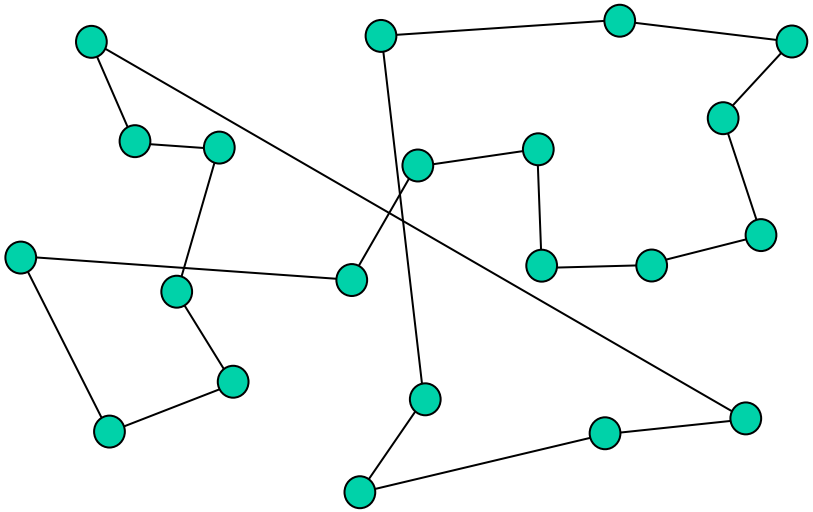
Auch: Handlungsreisendenproblem, Rundreiseproblem, TSP

- **Gegeben:** Vollständiger ungerichteter Graph $G=(V,E)$ mit Kantenkosten c_e
- **Gesucht:** Tour T (Kreis, der jeden Knoten genau einmal enthält) mit minimalen Kosten $c(T)=\sum c_e$

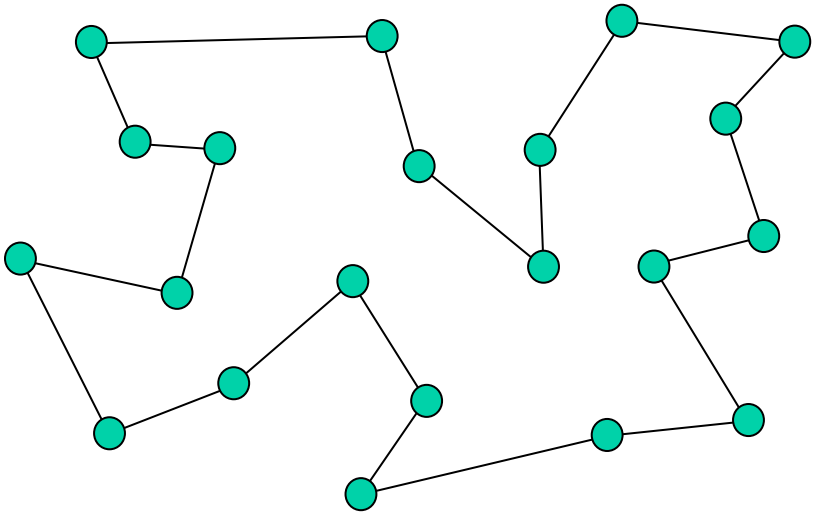
Nearest-Neighbor Heuristik für TSP

- Beginne mit leerer Tour $T:=\emptyset$
- Beginne an einem Knoten $v=v_0$: Ende der Tour
- Solange noch „freie“ Knoten existieren:
  - Suche den nächsten freien (noch nicht besuchten) Knoten zu v (d.h. die billigste Kante (v,w)) und
  - addiere Kante (v,w) zu T.
- Addiere die Kante (v,v_0)

TSP - Vergleich der Lösungen



Greedy



optimal

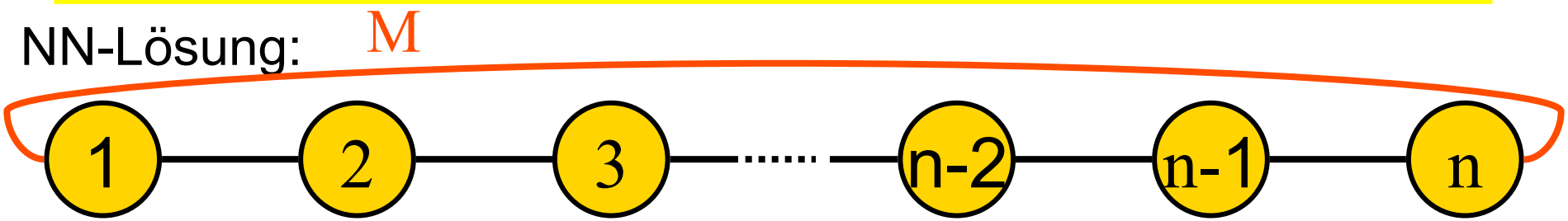
Wie gut ist die Nearest-Neighbor Heuristik (NN)?

IHRE ERGEBNISSE ☺

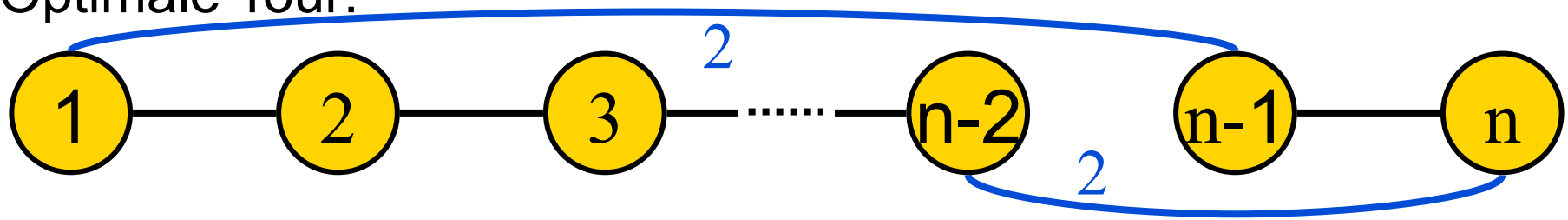
Wie gut ist die Nearest-Neighbor Heuristik (NN)?

- Kann beliebig schlechte Ergebnisse liefern,
- z.B. für $c(i,i+1)=1$ für $i=1,..,n-1$ und $c(n,1)=M$ (für eine sehr große Zahl M) und $c(i,j)=2$ sonst

NN-Lösung:



Optimale Tour:



Lösungswert NN: $(n-1)+M \gg$ Optimalwert: $(n-2)+4=n+2$

Bin-Packing / Packen von Kisten

- **Geg.:** Gegenstände $1,\ldots,N$ der Größe w_i und beliebig viele Kisten der Größe K.

- **Gesucht:** Finde die kleinste Anzahl von Kisten, die alle Gegenstände aufnehmen.

First-Fit Heuristik: Jeder Gegenstand wird in die erstmögliche Kiste gelegt, in die er paßt.

Wie gut ist die First-Fit Heuristik?

IHRE ERGEBNISSE ☺

First-Fit Heuristik für Bin-Packing

- Gegeben sind beliebig viele Kisten der Größe 101 und 37 Gegenstände der Größen:

  7x Größe 6, 7x Größe 10, 3x Größe 16,

  10x Größe 34, 10x Größe 51

Kiste 1:	7x Größe 6	5x Größe 10	Summe=92
Kiste 2:	2x Größe 10	3x Größe 16	Summe=68
Kisten 3-7:	2x Größe 34		Summe=68
Kisten 8-17:	1x Größe 51		FF-Lösung: 17 Kisten

Optimale Lösung: Insgesamt: 10 Kisten

Kisten 1-3:	1x Größe 51	1x Größe 34	1x Größe 16	
Kisten 4-10:	1x Größe 51	1x Größe 34	1x Größe 10	1x Gr. 6

Das 0/1-Rucksackproblem

- **Geg.:** N Gegenstände mit Gewicht (Größe) w_i und Wert c_i, und ein Rucksack der Größe K.

- **Gesucht:** Menge der in den Rucksack gepackten Gegenstände mit maximalem Gesamtwert; dabei darf das Gesamtgewicht den Wert K nicht überschreiten.

Greedy-Heuristik für das 0/1-Rucksackproblem

- Sortiere die N Gegenstände nach ihrem Nutzen:= Wert c_i, / Gewicht w_i
- Für alle Gegenstände i in der sortierten Reihenfolge:
  - Falls i noch in den Rucksack paßt: füge es hinzu.

Greedy-Heuristik an Beispiel für das 0/1-Rucksackproblem

K=17

Gegenstand	a	b	c	d	e	f	g	h
Gewicht	3	4	4	6	6	8	8	9
Wert	3	5	5	10	10	11	11	13
Nutzen	1	1,25	1,25	1,66	1,66	1,37	1,37	1,44

- Sortierung nach Nutzen:= Wert c_i / Gewicht w_i ergibt: d,e,h,f,g,b,c,a
- Wir packen also in den Rucksack:

d (6) | e (6) | b(4)

Wert=25

Wie gut ist die Greedy-Heuristik für das 0/1-Rucksackproblem?

IHRE VORSCHLÄGE ☺

Beispielinstanz für 0/1-Rucksackproblem

- N Gegenstände mit Gewichten und Werten $w_i = c_i = 1$ für $i=1,\ldots,N-1$, $c_N = K-1$ und $w_n = K = MN$, wobei M eine sehr große Zahl ist; der Rucksack hat Größe K (also sehr groß).

- Greedy packt die ersten $N-1$ Gegenstände in den Rucksack; kein Platz mehr für N-tes Element. Greedy-Lösungswert: $N-1$

- Optimale Lösung packt nur N-tes Element ein; Lösungswert: $K-1=MN-1$

Zusammenfassung

- **TSP:** NN-Heuristik: der berechnete Lösungswert kann beliebig weit vom optimalen Lösungswert entfernt sein.
- **Bin-Packing:** die von uns berechnete FF-Heuristik berechnete einen Lösungswert von 17, während der Optimalwert 10 war.
- **Rucksackproblem:** der von der Greedy-Heuristik berechnete Wert kann beliebig weit von der Optimallösung entfernt sein

Frage: Haben wir für Bin-Packing einfach keine „besonders schlechte“ Instanz gefunden? Oder ist Greedy hierfür immer relativ nah am Optimum?

Vertrauen in die Lösung

- Professor: Bei meiner Lösung entstehen Produktionskosten von 1.000.000 EUR.
- Praktiker: Gibt es keine bessere Lösung?
- Professor: Wir haben tagelang gerechnet und keine bessere gefunden.
- Praktiker:  (Skepsis)
- Professor: Aber ich kann garantieren, dass es keine für weniger als 950.000 EUR geben kann.
- Praktiker:  (höchstens 5% daneben!)

Motivation für Approximative Algorithmen

Kap. 7.2: Approximative Algorithmen und Gütegarantien

- Approximative Algorithmen sind Heuristiken, die (im vorhinein) eine Gütegarantie für die gefundene Lösung geben können.
- Z.B. der Art: „Die gefundene Lösung ist um höchstens $x\%$ schlechter als der Wert der optimalen Lösung.“

Approximative Algorithmen und Gütegarantien

- Sei A ein Algorithmus, der für jede Probleminstanz P eines Optimierungsproblems Π eine zulässige Lösung mit positivem Wert liefert. Dann def. wir:
- $c_A(P)$ als den Wert der Lösung des Algorithmus A für Probleminstanz $P \in \Pi$
- $c_{opt}(P)$ sei der optimale Wert für P.

- **Für Minimierungsprobleme gilt:**
- Falls $c_A(P) / c_{opt}(P) \leq \varepsilon$ für **alle** Probleminstanzen P und ein $\varepsilon > 0$, dann heißt A ein ε-approximativer Algorithmus und die Zahl ε heißt Gütegarantie von Algorithmus A.

Approximative Algorithmen und Gütegarantien

- Sei A ein Algorithmus, der für jede Probleminstanz P eines Optimierungsproblems Π eine zulässige Lösung mit positivem Wert liefert. Dann def. wir:
- $c_A(P)$ als den Wert der Lösung des Algorithmus A für Probleminstanz $P \in \Pi$
- $c_{opt}(P)$ sei der optimale Wert für P.

- **Für Max imierungsprobleme gilt:**
- Falls $c_A(P) / c_{opt}(P) \geq \varepsilon$ für **alle** Probleminstanzen P und ein $\varepsilon > 0$, dann heißt A ein ε-approximativer Algorithmus und die Zahl ε heißt Gütegarantie von Algorithmus A.

Approximative Algorithmen

- Für Minimierungsprobleme gilt: $\varepsilon \geq 1$
- Für Maximierungsprobleme gilt: $\varepsilon \leq 1$
- $\varepsilon = 1 \Leftrightarrow A$ ist exakter Algorithmus (berechnet immer den optimalen Wert)

- **Für Minimierungsprobleme gilt:**
- Falls $c_A(P) / c_{opt}(P) \leq \varepsilon$ für **alle** Probleminstanzen P und ein $\varepsilon > 0$, dann heißt A ein ε-approximativer Algorithmus und die Zahl ε heißt Gütegarantie von Algorithmus A.

Gütegarantie-Überlegungen der FF-Heuristik für Bin-Packing

- Unsere Bin-Packing Beispielinstanz P:
- Lösung der FF-Heuristik: $c_A(P)=17$ Kisten
- Optimale Lösung: $c_{opt}(P)=10$ Kisten
- $c_A(P) / c_{opt}(P) = 17 / 10 = 1,7$
- FF-Heuristik ist vielleicht ein 1,7-approximativer Algorithmus (nur wenn ≤ 1,7 für alle Instanzen gilt)
- ε kann auf keinen Fall kleiner als 1,7 sein

Gütegarantie der FF-Heuristik

- **Theorem:** Die First-Fit Heuristik für Bin-Packing besitzt asymptotisch eine Gütegarantie von 2.
- Es gilt: $c_{FF}(P) / c_{opt}(P) \leq 2+1/c_{opt}(P)$ für alle $P \in \Pi$

- **Beweis:** Offensichtlich gilt: Jede FF-Lösung füllt alle bis auf eine der belegten Kisten mindestens bis zur Hälfte (sonst hätten wir diese zusammenlegen können). Daraus folgt für alle $P \in \Pi$:

$$K/2\ (c_{FF}(P) - 1) \leq \sum_{j=1..N} w_j \leq c_{opt}(P)\ K$$

Größe der verteilten Gegenstände

Gesamtvolumen: in c_{opt} Kisten paßt alles rein

Gütegarantie der FF-Heuristik

- **Theorem:** Die First-Fit Heuristik für Bin-Packing besitzt asymptotisch eine Gütegarantie von 2.
- Es gilt: $c_{FF}(P) / c_{opt}(P) \leq 2+1/c_{opt}(P)$ für alle $P \in \Pi$

- **Beweis:** Offensichtlich gilt: Jede FF-Lösung füllt alle bis auf eine der belegten Kisten mindestens bis zur Hälfte (sonst hätten wir diese zusammenlegen können). Daraus folgt für alle $P \in \Pi$:

$$K/2\ (c_{FF}(P) - 1) \leq \sum_{j=1..N} w_j \leq c_{opt}(P)\ K$$

$$\Leftrightarrow \quad c_{FF}(P) \leq 2\, c_{opt}(P) + 1$$

Gütegarantie der FF-Heuristik

- **Theorem (ohne Beweis):** Die First-Fit Heuristik besitzt asymptotisch eine Gütegarantie von 17/10.
- Es gilt: $c_{FF}(P) / c_{opt}(P) < 17/10+2/c_{opt}(P)$ für alle $P \in \Pi$

- Unsere Beispielinstanz zeigt auch, dass die FF-Heuristik keinen besseren Approximationsfaktor als 17/10 besitzen kann.

Gütegarantie der Greedy-Heuristik für Rucksackproblem

Optimalwert: $MN-1$ ≫ Lösungswert Greedy: $N-1$

- Unsere Beispielinstanz zeigt, dass die Greedy-Heuristik keinen besseren Approximationsfaktor als $(N-1) / (MN-1)$ besitzen kann.
- Da M beliebig groß sein kann, existiert kein festes $\varepsilon>0$ für die eine Güte festgelegt werden kann.
- Greedy-Heuristik für das Rucksackproblem kann also beliebig schlecht werden.

Eine kleine Änderung führt zu 2-approximativem Algorithmus

Gütegarantie der Nearest-Neighbor Heuristik für TSP

Lösungswert NN: $(n-1)+M \gg$ Optimalwert: $(n-2)+4=n+2$

- Unsere Beispielinstanz zeigt, dass die NN-Heuristik keinen besseren Approximationsfaktor als $(n-1+M) / (n+2)$ besitzen kann.
- Da M beliebig groß sein kann, existiert kein festes $\varepsilon>0$ für die eine Güte festgelegt werden kann.
- NN-Heuristik für das TSP kann also beliebig schlecht werden.

Spanning-Tree Heuristik für TSP

Idee:

- Bestimme einen MST von G
- generiere daraus eine Eulertour
- generiere daraus eine zulässige Tour

Kanten-Verdopplung

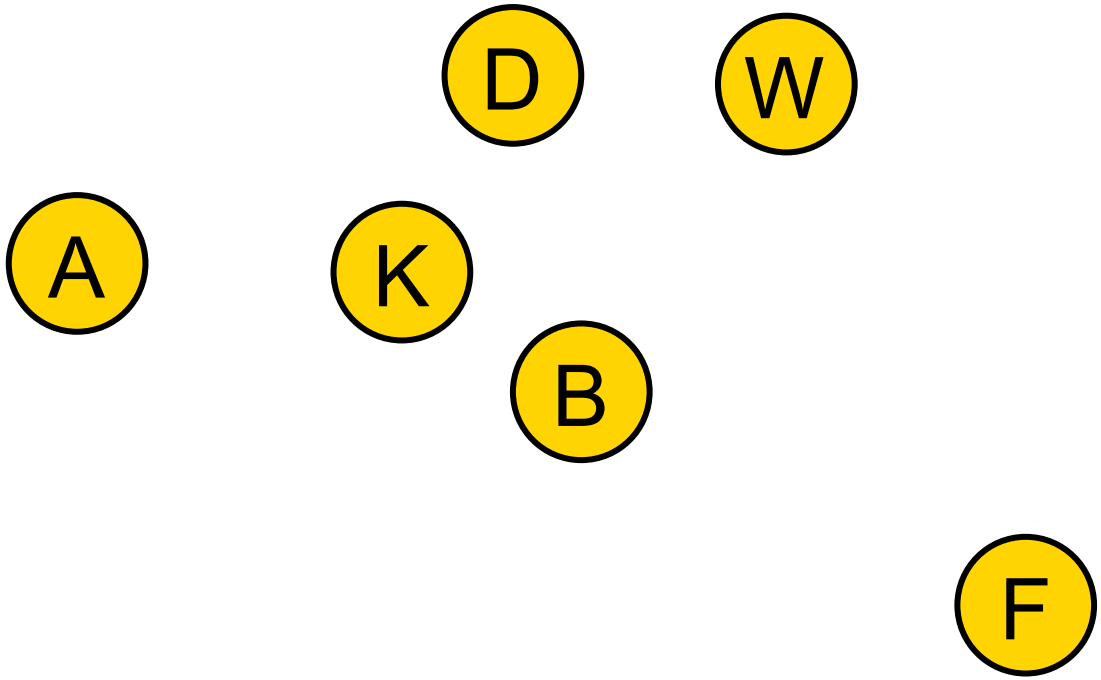
Abkürzungen

Definition:

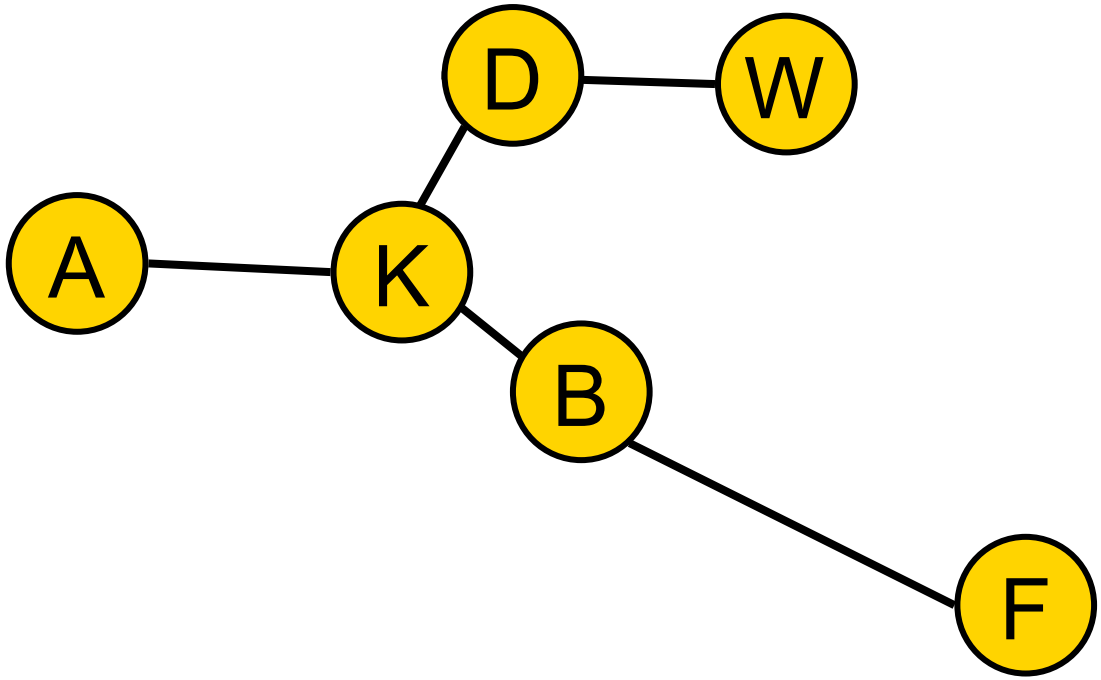
- Eine Eulertour ist ein geschlossener Kantenzug, der jede Kante des Graphen genau einmal enthält.

Es gilt: Ein Graph enthält eine Eulertour genau dann wenn jeder Knoten geraden Knotengrad hat (ohne Beweis).

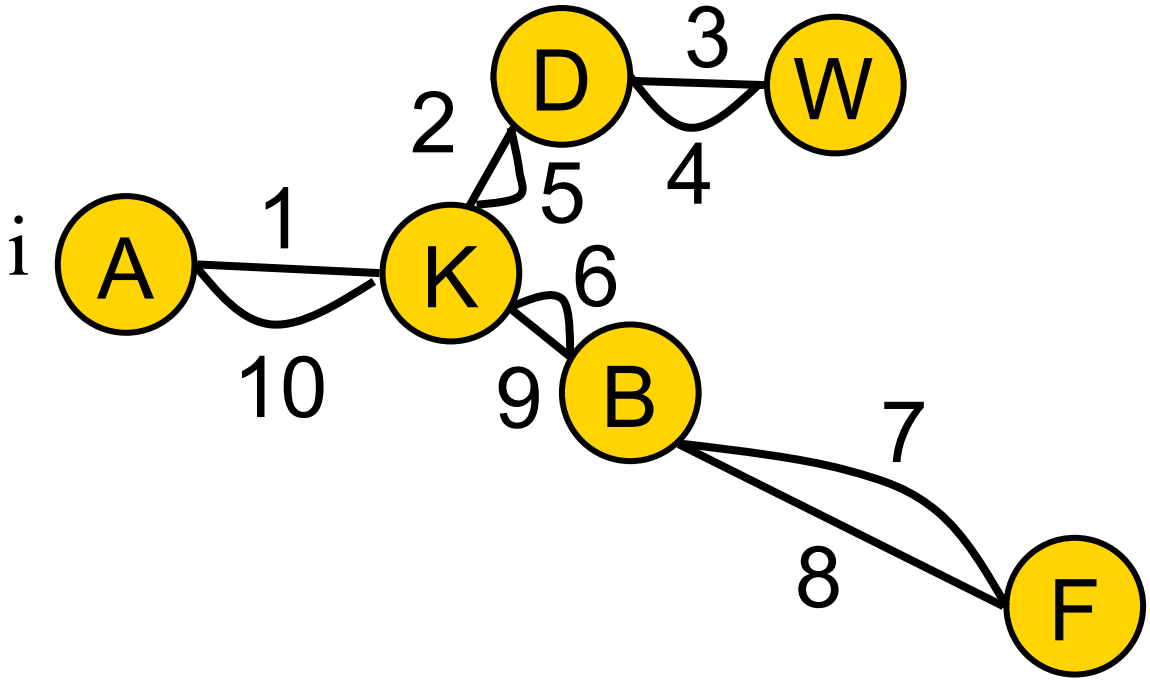
Beispiel für Spanning-Tree Heuristik



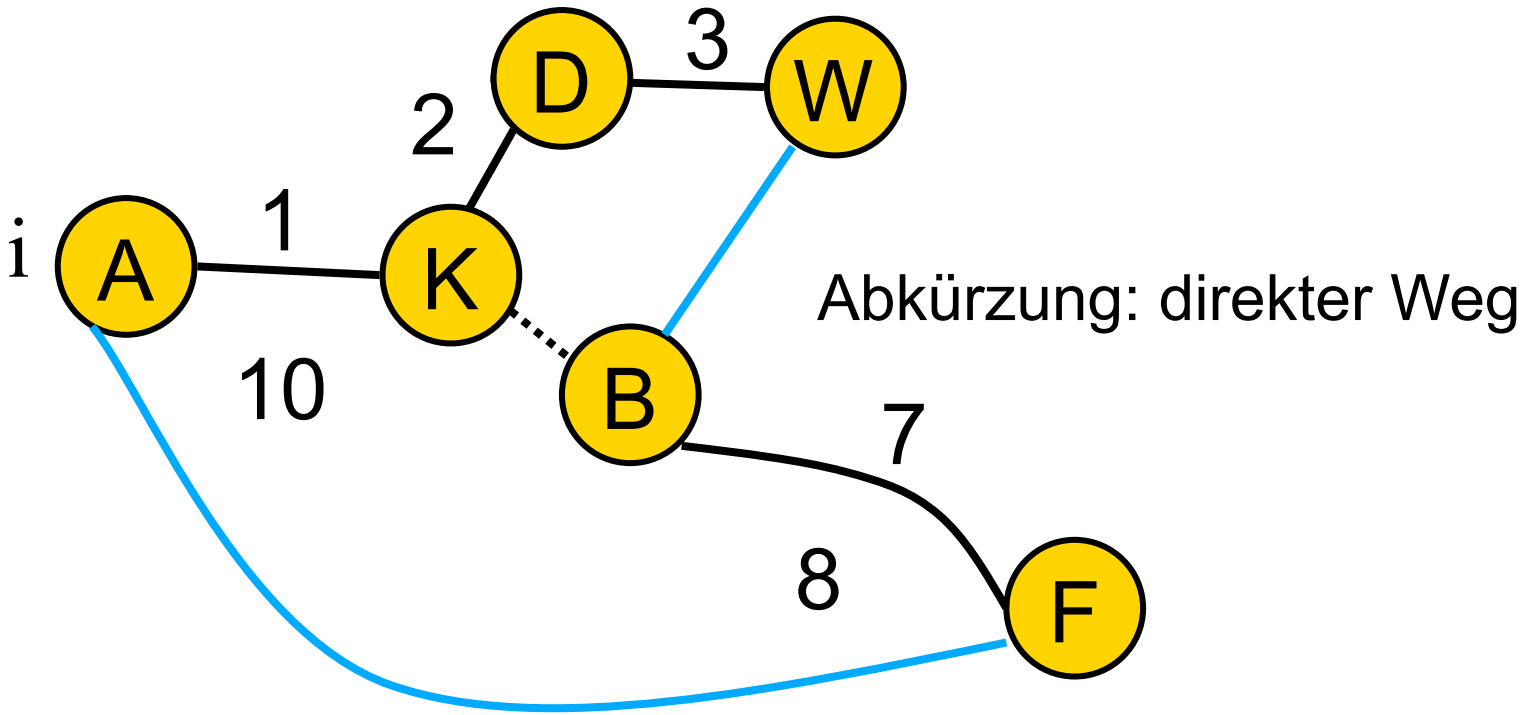
Beispiel für Spanning-Tree Heuristik



Beispiel für Spanning-Tree Heuristik



Beispiel für Spanning-Tree Heuristik



Spanning-Tree Heuristik für TSP

1. Bestimme einen MST B von G
2. Verdopple alle Kanten aus $B \rightarrow$ Graph $G_2:=(V,B_2)$
3. Bestimme eine Eulertour C in G_2 (Tour, die jede Kante genau einmal enthält)
4. Gib der Eulertour C eine Orientierung
5. Wähle einen Knoten $s \in V$, markiere s, setze $p:=s$ und $T:=\emptyset$
6. **Solange** noch unmarkierte Knoten existieren **do {**
7. Laufe von p entlang der Orientierung von C bis ein
8. unmarkierter Knoten q erreicht ist.
9. Setze $T:=T \cup \{(p,q)\}$, markiere q, $p:=q$
10. **}**
11. Setze $T:=T \cup \{(p,s)\} \rightarrow$ STOP; T ist Tour.

Diskussion der Gütegarantie

- Auch für die Spanning-Tree Heuristik gibt es eine „schlechte“ Instanz, bei der Lösungen produziert werden, die beliebig weit vom optimalen Lösungswert entfernt sind.

- Man kann zeigen: Das Problem, das TSP-Problem für beliebiges $\varepsilon>1$ zu approximieren ist NP-schwierig.

- Aber: wenn man nur spezielle TSP-Instanzen betrachtet, dann kann man eine Gütegarantie finden.

Das Metrische TSP-Problem

- Ein TSP heißt **metrisch**, wenn für die gegebene Distanzmatrix alle $c_{ii}=0$ sind und die Dreiecksungleichung erfüllt ist, d.h. für alle Knoten i,j,k gilt: $c_{ik} \leq c_{ij} + c_{jk}$

- „Der direkte Weg von i nach k kann nicht länger sein als der Weg von i nach k über j."

- TSP-Probleme aus der Praxis sind sehr oft metrisch (z.B. Wegeprobleme) – sogar **euklidisch**: d.h. die Städte besitzen Koordinaten im 2-dim. Raum und die Distanzmatrix ist durch die euklidischen Distanzen gegeben.

Gütegarantie für ST-Heuristik

- Für metrische TSP und für die Spanning-Tree Heuristik gilt: $c_{ST}(P) / c_{opt}(P) \leq 2$ für alle $P \in \Pi$

- Beweis: $c_{ST}(P) \leq c_{B2}(P) = 2\ MST(P) \leq 2\ c_{opt}(P)$

wegen Dreiecksungleichung: ST-Lösung läuft direkt

denn: in TSP-Lösung müssen u.a. alle Knoten miteinander verbunden sein; der billigste Weg hierfür ist MST

Christophides-Heuristik (CH)

Idee: Ähnlich wie die Spanning-Tree Heuristik, verzichte jedoch auf die Verdopplung der Kanten

Problem: Es existiert keine Eulertour in T.

Lösung: Füge Kanten hinzu, so dass eine Eulertour existiert.

Definition: Ein **perfektes Matching** M ist eine Kantenmenge, die jeden Knoten genau einmal enthält. Sie ordnet also jedem Knoten einen eindeutigen Partnerknoten zu.

Christophides-Heuristik (CH)

Lösungsidee:

- Füge die Kanten eines **perfekten Matchings** zwischen den ungeraden Knoten zum MST hinzu.
- Nun haben alle Knoten geraden Grad und eine Eulertour existiert.

- Um eine gute Lösungsgarantie zu erhalten, bestimmen wir ein **perfektes Matching** M mit **kleinstem Gewicht**, d.h. die Summe aller Kantengewichte c_e über die Kanten in M ist minimal unter allen perfekten Matchings.

- **Ein minimales perfektes Matching** M kann in polynomieller Zeit berechnet werden (o.Bw.)

Christophides-Heuristik (CH)

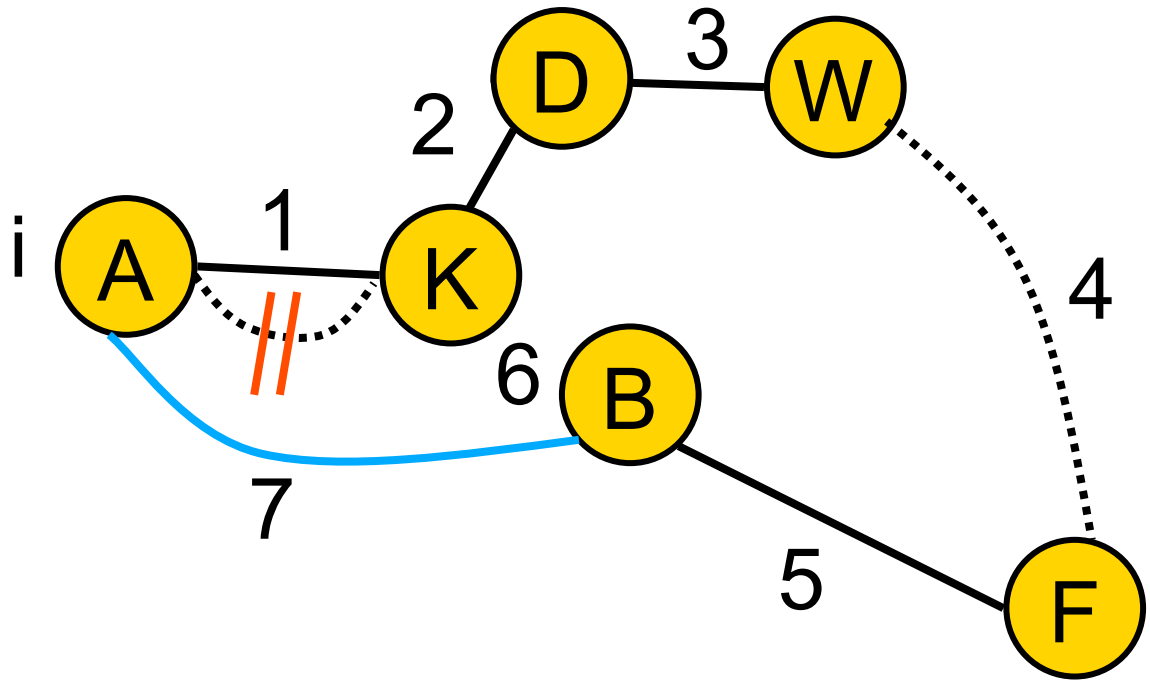
Ersetze in der Spanning-Tree Heuristik die Kanten-verdopplung (Schritt (2)) durch folgende Schritte:

Sei W die Menge der Knoten in (V,B) mit ungeradem Grad und $n=|W|$.

1. Bestimme im von W induzierten Untergraphen von K_n (vollständiger Graph auf n Knoten) ein perfektes Matching M kleinsten Gewichts.
2. Setze $B_2:=B \cup M$.

- Das Matching „geht auf“, denn: $|W|$ ist gerade.

Beispiel für CH-Heuristik



Gütegarantie für CH-Heuristik

- Für das metrische TSP und die CH-Heuristik gilt: $c_{CH}(P) / c_{opt}(P) \leq 3/2$ für alle $P \in \Pi$

- Beweis: Seien $i_1, i_2, \ldots, i_{2M}$ die Knoten von B mit ungeradem Grad so nummeriert, wie sie in einer optimalen Tour T_{opt} vorkommen.
- Sei $M_1 := \{(i_1,i_2),(i_3,i_4),\ldots\}$ und $M_2 := \{(i_2,i_3),(i_4,i_5),\ldots, (i_{2M},i_1)\}$.
- Es gilt: $c_{opt}(P) \geq c_{M1}(P) + c_{M2}(P) \geq c_M(P) + c_M(P)$

wg. Dreiecksungleichung: M nimmt Abkürzung

denn: M ist Matching kleinsten Gewichts

Gütegarantie für CH-Heuristik

- Für das metrische TSP und die CH-Heuristik gilt: $c_{CH}(P) / c_{opt}(P) \leq 3/2$ für alle $P \in \Pi$

- Beweis: Seien $i_1, i_2, \ldots, i_{2M}$ die Knoten von B mit ungeradem Grad so nummeriert, wie sie in einer optimalen Tour T_{opt} vorkommen.
- Sei $M_1 := \{(i_1,i_2),(i_3,i_4),\ldots\}$ und $M_2 := \{(i_2,i_3),(i_4,i_5),\ldots, (i_{2M},i_1)\}$.
- Es gilt: $c_{opt}(P) \geq c_{M1}(P) + c_{M2}(P) \geq c_M(P) + c_M(P)$
- Weiterhin gilt: $c_{CH}(P) \leq c_{B2}(P) = c_B(P) + c_M(P) \leq$

$$\leq c_{opt}(P) + 1/2\, c_{opt}(P) = 3/2\, c_{opt}(P)$$

CH-Heuristik: Bemerkungen

- Die Christophides-Heuristik (1976) war lange Zeit die Heuristik mit der besten Gütegarantie.
- Vor kurzem zeigte Arora (1996): das euklidische TSP kann beliebig nah approximiert werden: die Gütegarantie $\varepsilon>1$ kann mit Laufzeit $O(N^{1/(\varepsilon-1)})$ approximiert werden (PTAS: polynomial time approximation scheme)
- Konstruktionsheuristiken für das symmetrische TSP erreichen in der Praxis meist eine Güte von ca. 10-15% Abweichung von der optimalen Lösung. Die CH-Heuristik liegt bei ca. 14%.