Übung1

Moritz Haslhofer (e0025125)

2012-06-03

Aufgabe 1

Angabe

Approximative Algorithmen:

1. Was bedeutet es, wenn ein algorithmus A für ein *Maximierungsproblem* Gütegarantie ϵ besitzt?
2. Was bedeutet es, wenn ein algorithmus A für ein *Minimierungsproblem* Gütegarantie ϵ besitzt?

Lösung

Sei $c_A(P)$ die Lösung des Algorithmus A für die Probleminstanz $P \in \Pi$.
Weiters sei $c_{opt}(P)$ der optimale Wert für P.

1. Falls A ein Maximierungsproblem löst und: $\frac{c_A(P)}{c_{opt}(P)} \leq \epsilon$ ($\forall P \in \Pi, \exists \epsilon > 0$) dann ist A ein ϵ-approximativer Algorithmus, und ϵ die Gütegarantie von A
2. Analog dazu, wenn A ein Minimierungsproblem löst und $\frac{c_A(P)}{c_{opt}(P)} \geq \epsilon$ ($\forall P \in \Pi, \exists \epsilon > 0$) dann ist A ein ϵ-approximativer Algorithmus, und ϵ die Gütegarantie von A

Aufgabe 2

Angabe

Das symmetrische Traveling Salesman Problem:

Angenommen die Distanzwerte zwischen den Knoten im TSP sind alle paarweise ver- schieden, ist die mittels der Christophides-Heuristik erzeugte Lösung immer eindeutig?

Begründen Sie Ihre Antwort!

Lösung

Die einzelnen Schritte analysiert

1. **MST** - ist eindeutig
2. **min. Perfect Match** - muss nicht eindeutig sein
3. **MST U PF** - muss nicht endeutig sein (aus 2)

Aus zwei unterschiedlichen minimal perfect matches können unterschiedliche Graphen entstehen. Daher liefert die Heuristik keine eindeutige Lösung.

Aufgabe 3

Angabe

Gegeben ist die Nearest-Neighbor-Heuristik (NN) für das symmetrische TSP:

1. Starte von einem beliebigen Knoten s und markiere ihn als besucht.
2. Finde die kürzeste Kante zu einem unbesuchten, adjanzenten Knoten v.
3. Wenn alle Knoten besucht sind, kehre zum Anfangsknoten zurück und terminiere
4. Setze v als aktuellen Knoten, mariere ihn als besucht und gehe zu (2). Geben Sie die Laufzeit der NN-Heuristik in Σ - Notation an. Welche Approximationsgüte besitzt die NN-Heuristik?

Lösung

Die Laufzeit ist durch $\Theta(|V|^2)$ nach oben beschränkt. Die Heuristik hat keine beerechenbare Approximationsgüte, da sich für jedes Problem P mit ϵ ein Problem P' konstruieren lässt mit $\epsilon' > \epsilon$.

Aufgabe 4

Angabe

Betrachten Sie den Branch-and-Bound-Algorithmus für das asymmetrische Traveling-Salesman-Problem aus dem Skriptum bzw. der Vorlesung und die folgende Distanzmatrix:

$$D = \begin{pmatrix} \infty & 7 & 3 & 2 \\ 4 & \infty & 3 & 5 \\ 8 & 3 & \infty & 2 \\ 5 & 2 & 3 & \infty \end{pmatrix}$$

1. Führen Sie zuerst Zeilen- und danach Spaltenreduktion durch. Geben Sie die vollständige resultierende Matrix nach beiden Schritten an. Welche untere Schranke L für die Tourlänge können Sie aus den Reduktionen ableiten?
2. Zeichnen Sie den Nulldigraphen nach den obigen Reduktionen. Welche Schlussfolgerung können Sie aus diesem Nulldigraphen ziehen?

Lösung

1. Zeilenreduktion:

$$D = \begin{pmatrix} \infty & 5 & 1 & 0 \\ 1 & \infty & 0 & 2 \\ 6 & 1 & \infty & 0 \\ 3 & 0 & 1 & \infty \end{pmatrix}$$

$$L = 2 + 3 + 2 + 2 = 9$$

Spaltenreduktion:

$$D = \begin{pmatrix} \infty & 5 & 1 & 0 \\ 0 & \infty & 0 & 2 \\ 5 & 1 & \infty & 0 \\ 2 & 0 & 1 & \infty \end{pmatrix}$$

$$L = 9 + 1 + 0 + 0 + 0 = 10$$

2. Nulldigraph:

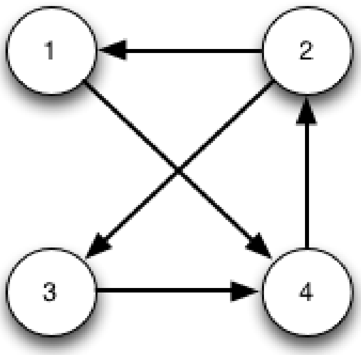

Abb. 1: Nulldigraph zu reduziertem D

Aufgabe 5

Angabe

Setzen Sie den Branch-and-Bound-Algorithmus aus der vorigen Aufgabe fort.

1. Führen Sie den Branching-Schritt aus. Wählen Sie Kante (3,4) zum Definieren der neuen Teilprobleme. Geben Sie die Matrizen der neuen Teilprobleme an.
2. Führen Sie auf den Matrizen der neuen Teilprobleme wiederum Zeilen- und Spaltenreduktion durch und geben Sie die resultierenden Matrizen an. Nennen Sie die neuen unteren Schranken für die Tourlänge.
3. Zeichnen Sie für beide Teilprobleme den Nulldigraphen nach den obigen Reduktionen. Welche Schlussfolgerungen können Sie aus diesen Nulldigraphen ziehen?

Lösung

1. Wir schauen uns die beiden Unterprobleme:

P = {(3,4)},{},{1,2,4},{1,2,3},D',10,\infty}

   und:

P = {},{(3,4)},{1,2,3,4},{1,2,3,4},D'',10,\infty}

   an. Die Adjazentenmatrizen schauen wie folgt aus:

$$D' = \begin{pmatrix} \infty & 5 & 1 \\ 0 & \infty & 0 \\ 2 & 0 & \infty \end{pmatrix}$$

Abb. 2: D' für erzwungene Kante (3,4)

$$D'' = \begin{pmatrix} \infty & 5 & 1 & 0 \\ 0 & \infty & 0 & 2 \\ 5 & 1 & \infty & \infty \\ 2 & 0 & 1 & \infty \end{pmatrix}$$

Abb. 3: D'' für entfernte Kante (3,4)

2.

$$D' = \begin{pmatrix} \infty & 4 & 0 \\ 0 & \infty & 0 \\ 2 & 0 & \infty \end{pmatrix}$$

$$L = 10 + 1 = 11$$

Abb. 4: D' reduziert

$$D'' = \begin{pmatrix} \infty & 5 & 1 & 0 \\ 0 & \infty & 0 & 2 \\ 4 & 0 & \infty & \infty \\ 2 & 0 & 1 & \infty \end{pmatrix}$$

$$L = 10 + 1 = 11$$

Abb. 5: D'' reduziert

3.

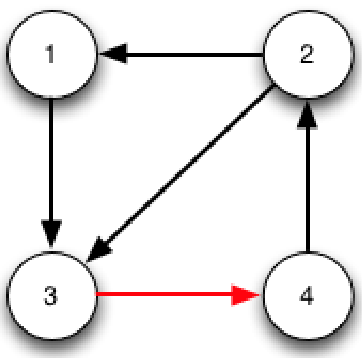

Abb. 6: Nulldigraph zu D'

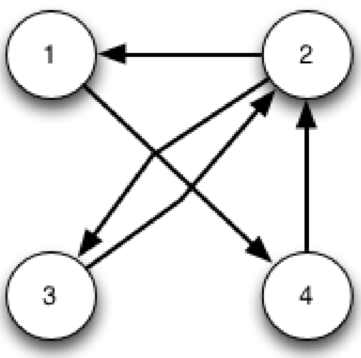

Abb. 7: Nulldigraph zu D''

Aufgabe 6

Angabe

Lösung

Aufgabe 7

Angabe

Lösung

Aufgabe 8

Angabe

Überlegen Sie sich zum Euklidischen TSP ein Beispiel, in dem die Verbesserungsheuristik 2-OPT eine Tour nicht weiter verbessern kann, obwohl diese noch nicht global optimal ist.

Lösung

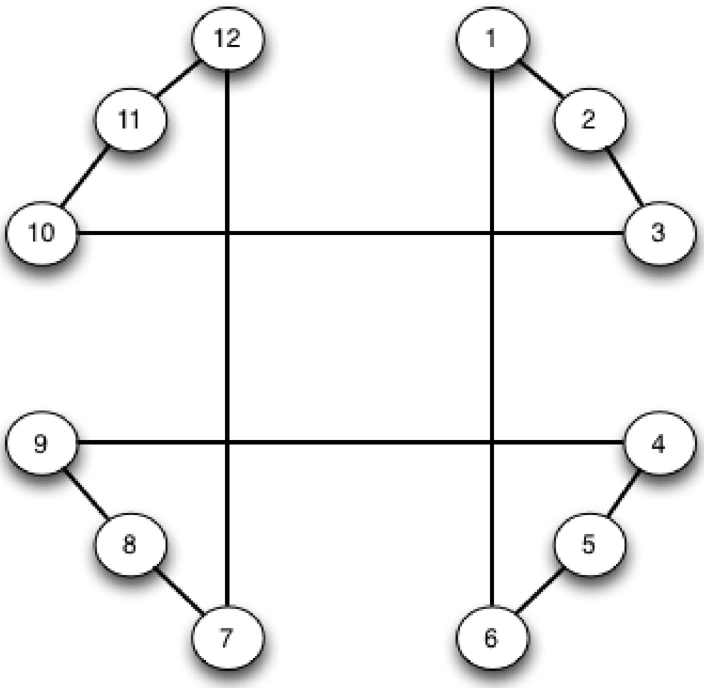

Abb. 8: 2-opt "resistenter" Graph

Aufgabe 9

Angabe

Gegeben ist der folgende vollständige, ungerichtete Graph G von fünf Städten und den jeweiligen Distanzen. Wenden Sie die Christophides-Heuristik an, um eine approximative Lösung für das Traveling-Salesman-Problem in diesem Graphen zu finden.

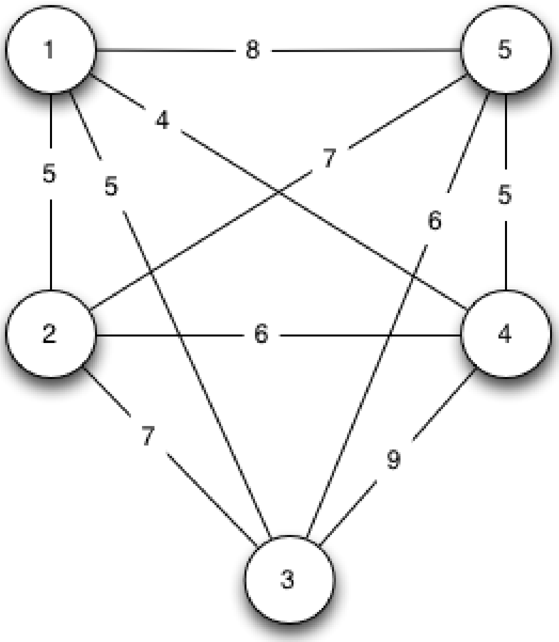

Abb. 9: Graph

1. Markieren Sie jene Kanten, die einen minimalen Baum B aufspannen und geben Sie die Menge W der Knoten mit ungeraden Grad an.
2. Geben Sie das perfekte Matching kleinsten Gewichts M im von W induzierten Untergraphen von G an.
3. Zeichnen Sie den Graphen G_2 mit den Kanten $B \cup M$.
4. Bestimmen Sie eine Euler-Tour F im Graphen G_2

Lösung

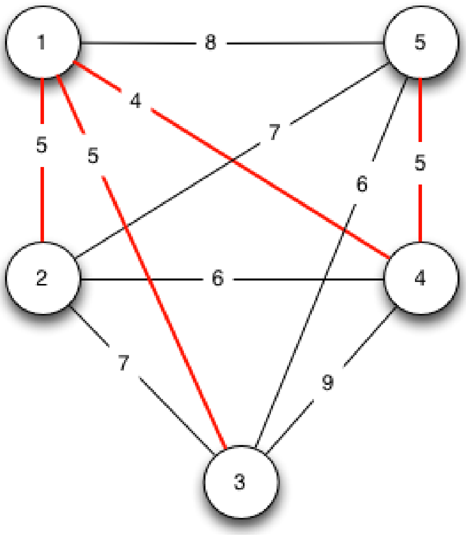

Abb. 10: Minimal Spanning Tree

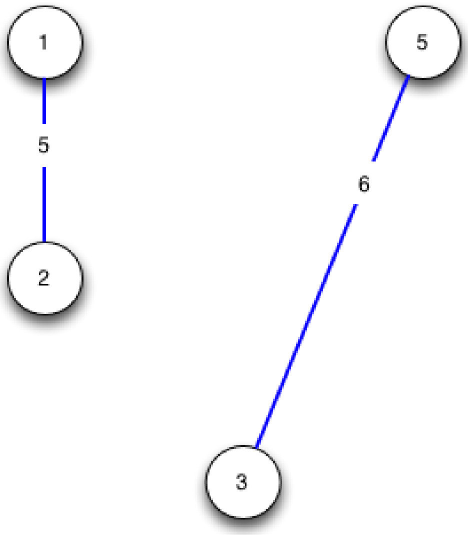

Abb. 11: Perfect Match

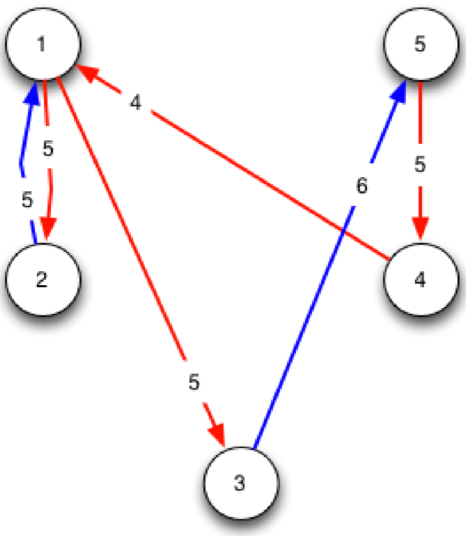

Abb. 12: Minimal Spanning Tree + Perfect Match

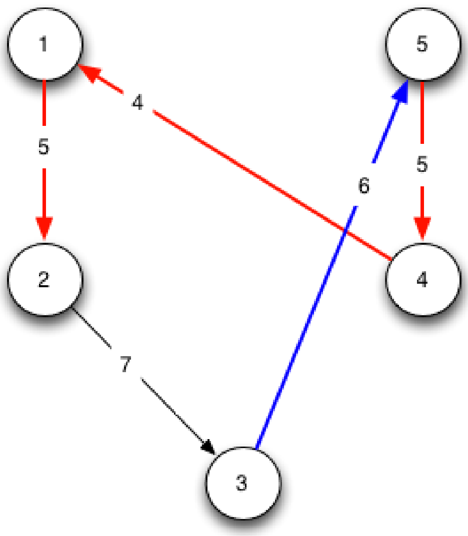

Abb. 13: Hamilton

Aufgabe 10

Angabe

Tabu-Suche: Die Wahl der Länge einer Tabu-Liste ist häufig kritisch. Was können die Probleme bei zu kurzer bzw. zu langer Tabu-Liste sein?

Lösung

Eine zu kleine Tabuliste führt zu Zyklen im Algorithmus - (der Algorithmus bleibt in einem lokalen Optimum hängen)

Eine zu grosse Tabuliste:

- verbraucht mehr Zeit bei der Prüfung, ob ein Wert in der Liste vorkommt
- beschränkt den Algorithmus zu stark, und führt ihn von einer Optimalen suche weg