

Reoptimierung am Beispiel des TSP

Seminararbeit Theoretische Informatik 6. Semester

Florian Müller, flm@fastmailfm, ZHAW

Betreuer: H.-J. Böckenhauer, hjb@inf.ethz.ch

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen	5
2.1	TSP- Travelling Salesman Problem	5
2.2	Hamilton-Pfad und Hamilton-Kreis	5
2.3	Polynomialzeit-Reduktion	5
2.4	Metrisches TSP	6
3	Reoptimierung des TSP	7
3.1	Definitionen	7
3.2	Ziele	8
3.3	Die Schwere von $LM - TSP$	8
3.4	$LM - \triangle_\beta - TSP$	11
3.4.1	Schwere	11
3.4.2	Ein Approximationsalgorithmus	13

1 Einleitung

In dieser Arbeit wird anhand des Travelling Salesman Problems (TSP) aufgezeigt, inwiefern eine bereits gefundene optimale Lösung wiederverwendet werden kann, um eine minimal veränderte Probleminstance zu approximieren. Die Motivation dahinter ist die Tatsache, dass in vielen Bereichen der Optimierung wie z.B. bei Routing-Problemen oft eine Lösung bereits bekannt ist und sich in einem Schritt nur wenig verändert. Wie etwa in einem Busnetz, dem eine neue Haltestelle hinzugefügt wird.

Die Arbeit baut hauptsächlich auf zwei Arbeiten von H.-J. Böckenhauer et al. auf: Reusing Optimal TSP Solutions for Locally Modified Input Instances (Extended Abstract) ([1]) und On the Hardness of Reoptimization ([2]). Alle Beweise sind daraus entnommen und um zusätzliche Zwischenschritte und Illustrationen ergänzt, um die Materie einem breiteren Zielpublikum näher bringen zu können.

Die Arbeit behandelt zwei Typen des TSP:

- Das allgemeine TSP
- Das $\triangle_\beta - TSP$, welches die abgeschwächte bzw. verstärkte Dreiecksungleichung erfüllt

Beide Typen werden in eine modifizierte Instanz $LM - TSP$ bzw. $LM - \triangle_\beta - TSP$ überführt, indem die Kosten genau einer Kante verändert werden, alles andere jedoch gleich bleibt. Formal gibt es immer einen Graphen G mit den Knoten V und den Kanten E , eine originale Kostenfunktion $c_O : E \rightarrow \mathbb{Q}^+$, die jeder Kante eine rationale Zahl – die Kosten – zuordnet und eine neue Funktion $c_N : E \rightarrow \mathbb{Q}^+$, welche für genau eine Kante einen anderen Wert zurück liefert. Ausserdem gehen wir davon aus, dass für G unter der Kostenfunktion c_O bereits eine optimale Tour \overline{C} gefunden wurde. Das ganze Problem ist durch das n-Tupel $(G, c_O, c_N, \overline{C})$ vollständig spezifiziert.

Wir haben natürlich die Hoffnung, dass das Problem viel einfacher wird, weil wir bereits im Besitz einer optimalen Lösung sind und sich nur wenig ändert. Viel einfacher kann heissen:

1. Es gibt einen Algorithmus, der das Problem in Polynomialzeit exakt löst.
2. Und falls es den nicht gibt, so doch wenigstens einen Algorithmus, welcher das Problem in Polynomialzeit mit einer konstanten relativen Güte approximiert und der besser ist als ein Algorithmus, welcher die alte optimale Tour nicht kennt.

Als erstes werden wir zeigen, dass leider weder für $LM - TSP$ noch $LM - \Delta_\beta - TSP$ ein Polynomialzeit-Algorithmus, welcher das Problem exakt löst (ausser $P = NP$), existiert. Und dass es für $LM - TSP$ sogar nicht mal einen polynomiellen Approximationsalgorithmus mit konstanter relative Güte gibt (ausser $P = NP$).

Für das Problem $LM - \Delta_\beta - TSP$ lässt sich jedoch ein Algorithmus finden, der eine konstante relative Güte in Abhängigkeit von β garantiert. Im Spezialfall $LM - \Delta_1 - TSP$ (das metrische TSP) liefert der Algorithmus eine Approximation von $\frac{7}{5} = 1.4$, was besser ist, als der Algorithmus von Christofides mit einer relativen Güte von $\frac{3}{2} = 1.5$. Auch im fast-metrischen Bereich von $1 < \beta < 3.34899$ liefert der Algorithmus eine bessere Approximationsrate als alle anderen bekannten Algorithmen.

Die Arbeiten von H.-J. Böckenhauer et al. behandeln noch eine weitere TSP-Variante: das metrische TSP mit Deadlines (Zeitschranken) $DL - \Delta TSP$. Bei dieser Art sind zusätzlich gewisse Knoten als Deadlines markiert und erhalten durch eine Funktion $d_O : D \rightarrow \mathbb{Q}^+$ einen Wert – die Deadline – zugeordnet. Die Summe der Kosten aller Kanten vom Startknoten s bis zu den Deadline-Knoten darf deren Deadline nicht überschreiten. Die Modifikation in diesem Fall ist die Änderung genau einer Deadline. Es wird eine neue Funktion $c_N : D \rightarrow \mathbb{Q}^+$ definiert, die genau für eine Deadline einen anderen Wert annimmt.

Für das Original-Problem $DL - \Delta TSP$ ist die beste zu erreichende Approximationsrate 2. In einem spannenden Beweis wird gezeigt, dass dies auch für $LM - DL - \Delta TSP$ gilt. Der Beweis nimmt dafür an, dass es einen $2 - \epsilon$ Algorithmus gibt mit $\epsilon > 0$ und gelangt zum Ergebnis, dass ein solcher Algorithmus ein anderes Problem RHP^1 , von dem bekannt ist, dass es NP-vollständig ist, in Polynomialzeit exakt lösen kann. Dies ist ein Widerspruch. Daraus folgt, dass wir nicht darauf hoffen können, einen besseren Approximationsalgorithmus für $LM - DL - \Delta TSP$ zu finden, als der Algorithmus mit relativer Güte 2. Die Kenntnis einer optimalen Tour bringt uns bei diesem Problem nichts.

¹Das eingeschränkte Hamilton-Pfad Problem: Sei $G = (V, E)$ ein Graph, der einen Hamilton-Pfad von s nach u enthält, enthält er noch einen anderen Hamilton-Pfad von s nach $v \neq u$

2 Grundlagen

Da die Zielgruppe dieser Arbeit Studenten sind, werden gewisse Grundlagen, welche danach für das Verständnis der Beweisführungen hilfreich sein können, in diesem Kapitel nochmals kurz aufgefrischt.

2.1 TSP- Travelling Salesman Problem

Das Travelling Salesman Problem (TSP) besteht aus einer Probleminstanz (G, c) , wobei $G = (V, E)$ ein vollständiger ungerichteter Graph und $c = E \rightarrow \mathbb{Q}^+$ eine Kostenfunktion ist, welche jeder Kante in E eine rationale Zahl zuordnet. Gesucht wird die Tour $T = (v_1, \dots, v_n, v_1)$ von einem Startknoten v_1 aus durch alle anderen Knoten zurück zum Startknoten mit minimalen Kosten.

2.2 Hamilton-Pfad und Hamilton-Kreis

Ein Hamilton-Pfad HP ist ein Pfad durch eine Menge von Knoten $V = v_1, \dots, v_n$, welcher genau einmal durch jeden Knoten geht ($HP = (v_1, \dots, v_n)$).

Ein Hamilton-Kreis HC geht wie der Hamilton-Pfad genau einmal durch jeden Knoten, muss jedoch im selben Knoten enden, bei dem er gestartet ist ($HC = (v_1, \dots, v_n, v_1)$).

2.3 Polynomialzeit-Reduktion

Eine Polynomialzeit-Reduktion (\prec) ist der Prozess, bei dem jede Probleminstanz eines Problems P_A in Polynomialzeit in eine gültige Probleminstanz des Problems P_B umgewandelt werden kann. Das heisst, die Reduktion ist eine Funktion $f(I_A) \rightarrow I_B$. Zudem gilt, dass falls eine Lösung für I_A existiert, auch eine Lösung für I_B existiert und umgekehrt.

In dieser Arbeit wird die Reduktion von NP-vollständigen Problemen auf Probleme der Reoptimierung benutzt, um zu beweisen, dass diese Probleme NP-schwer sind.

Eine Reduktion in dieser Arbeit hat immer die Form

$Problem_{NP-vollständig} \prec Problem_{zu\ beweisen\ NP-schwer}$

und eine erfolgreiche Reduktion sagt aus, dass $Problem_{zu\ beweisen\ NP-schwer}$ mindestens so schwer ist wie $Problem_{NP-vollständig}$.

Damit werden wir beweisen, dass auch Reoptimierungsprobleme NP-schwer sind und daher kein exakter Polynomialzeit-Algorithmus dafür existiert (ausser $P = NP$).

2.4 Metrisches TSP

Das metrische TSP ist ein Spezialfall des TSP, welches die Dreiecksungleichung erfüllt. Diese besagt, dass der direkte Weg von einem Knoten zu einem anderen immer kleiner oder gleich ist, wie der Umweg über zwei andere Knoten. Formal:

$$c(u, w) \leq c(u, v) + c(v, w)$$

wobei c die Kostenfunktion $c : E \rightarrow \mathbb{Q}^+$ ist, welche jeder Kante Kosten zuordnet.

Dies entspricht den geometrischen Gegebenheiten. Man muss sich das vorstellen, wie in Abbildung 2.1. Die Kante c kann nur grösser werden, indem man den Winkel zwischen a und b vergrössert. Dies kann man höchstens so lange machen, bis der Winkel 180° beträgt und $c = a + b$, danach wird c wieder kleiner.



Abbildung 2.1: Dreiecksungleichung

3 Reoptimierung des TSP

3.1 Definitionen

Dieses Kapitel behandelt drei Arten von Travelling Salesman Problemen:

$lm - TSP$ Beschreibt das Problem, das als Eingabe einen vollständigen Graphen $G = (V, E)$, eine originale Kostenfunktion c_O , eine optimale Tour \bar{C} und eine modifizierte Kostenfunktion c_N bekommt und daraus eine neue optimale Tour C ermitteln soll. Die Kostenfunktion c_N ist so konstruiert, dass sie genau für eine Kante e einen anderen und für alle $E \setminus \{e\}$ denselben Wert wie c_O ausgibt. Formal lautet eine Problem Instanz I für $lm - TSP$ demnach $I = (G, c_O, c_N, \bar{C})$.

Die Knoten- und Kantendefinitionen ändern sich dabei nicht.

$lm - \Delta_\beta - TSP$ Ist ein Spezialfall des $lm - TSP$ für Kostenfunktionen c , welche folgende Dreiecksungleichung erfüllen:

$$c(\{u, w\}) \leq \beta \cdot (c(\{u, v\}) + c(\{u, w\}))$$

Für das metrische TSP $\Delta_1 - TSP$ wird als Abkürzung $\Delta - TSP$ verwendet.

Falls $\beta < 1$ wird von einer verschärften, falls $\beta > 1$ von einer abgeschwächten Dreiecksungleichung gesprochen (siehe Abbildung 3.1).

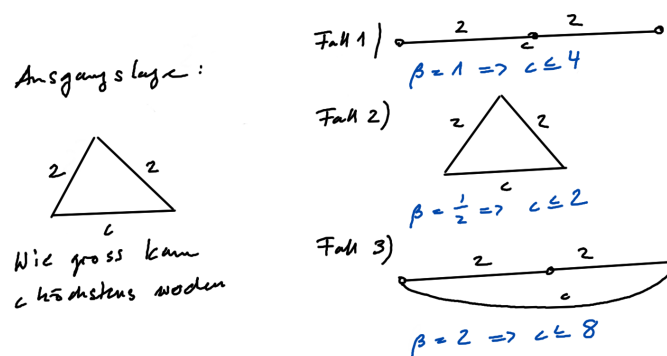


Abbildung 3.1: Dreiecksungleichung mit verschiedenen β

3.2 Ziele

Es ist allgemein bekannt, dass falls $P \neq NP$, es keinen polynomiellen Approximationsalgorithmus mit konstanter relativer Güte r für das allgemeine TSP (ohne Einschränkungen) gibt. Das heisst, eine Garantie, dass das Verhältnis $g = \frac{Solution_{found}}{Solution_{optimal}}$ für jede Eingabe kleiner gleich einem bestimmten konstanten Wert r ist (wie z.B. $\frac{3}{2}$).

Wir zeigen anhand eines Beweises von H.-J.-Böckenhauer et. al. ([1]), dass dies auch für $lm - TSP$ gilt. Das heisst, das Problem wird nicht leichter, obwohl die Änderung nur eine einzige Kante betrifft und eine optimale Lösung für das Originalproblem bereits ermittelt wurde.

Des weiteren wird gezeigt, dass das Problem einfacher wird, wenn wir fordern, dass die β -Dreiecksungleichung eingehalten werden muss. Daraus ergibt sich das Problem $lm - \Delta_\beta - TSP$. H.-J. Böckenhauer et. al. ([1], [2]) konnten für eine Teilklasse dieser Probleme mit $\beta > \frac{1}{2}$ zeigen, dass dieses Problem weiterhin NP-schwer ist, dass jedoch ein polynomieller Approximationsalgorithmus mit konstanter relativer Güte existiert, welche nur vom Parameter β abhängt. Die relative Güte verbessert sich im metrischen Fall $lm - \Delta_1 - TSP = lm - \Delta - TSP$ gegenüber der bisher durch den Algorithmus von Christofides erreichten 1.5 auf 1.4 und schlägt auch im Bereich von $1 < \beta < 3.34899$ alle bisher bekannte Algorithmen ([1] S. 253).

Alle Beweise basieren auf den Arbeiten [1] und [2] von H.-J. Böckenhauer et. al.

3.3 Die Schwere von $LM - TSP$

Theorem 1. *Es gibt für $LM - TSP$ keinen polynomiellen Approximationsalgorithmus (ausser $P = NP$).*

Beweis. Dazu führen wir eine Reduktion vom Hamilton-Kreis HC Problem auf $LM - TSP$ durch.

$$HC \prec LM - TSP$$

Hamilton-Kreis-Problem Es ist ein ungerichteter Graph $G = (V, E)$ mit $V = \{v_1, \dots, v_n\}$ und $E = \{e_1, \dots, e_k\}$ ohne Kantengewichte gegeben. Gesucht wird die Entscheidung, ob G ein Hamilton-Kreis besitzt oder nicht.

Reduktion Der ungerichtete Graph G wird zuerst in einen ungerichteten vollständigen Graphen $G_O = (V', E')$ mit einer Kostenfunktion c_O umgewandelt. Dieser Graph wird so konstruiert, dass eine optimale Tour \bar{C} offensichtlich ist. Danach werden die Kosten einer einzigen Kante so verändert (neue Kostenfunktion c_N), dass die Tour \bar{C} nicht mehr

optimal ist. Eine weitere optimale Tour entspräche nun genau einem Hamilton-Kreis in G und umgekehrt: ein Hamilton-Kreis in G entspräche genau einer optimalen Tour in G_O unter der neuen Kostenfunktion c_A .

Schritt 1 Für jeden Knoten $v_i \in V$ wird ein sogenannter Diamant-Graph erstellt. Alle Kantenkosten sind dabei genau 1. (Abbildung 3.2). Die Bezeichner N_i, E_i, S_i, W_i stehen dabei für north, east, south und west. Das Spezielle an diesen Diamantgraphen ist, dass sie so konstruiert sind, dass es genau zwei Wege durch sie hindurch gibt, möchte man jeden Knoten genau einmal besuchen. 1) West \rightleftharpoons Ost 2) Nord \rightleftharpoons Süd (Abbildung 3.3). Diese Tatsache wird hier nicht bewiesen.

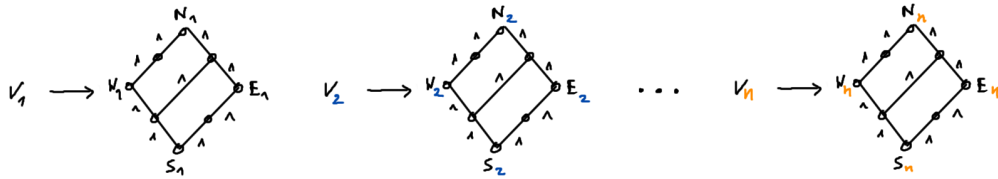


Abbildung 3.2: Im ersten Schritt werden alle Knoten aus $V = \{v_1, \dots, v_n\}$ in einen Diamant-Graphen mit allen Kantenkosten = 1 umgewandelt

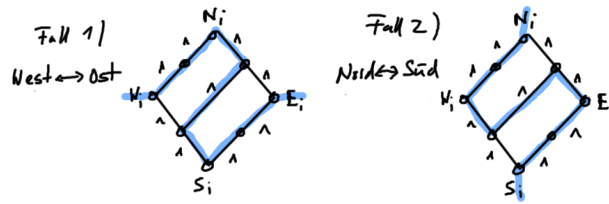


Abbildung 3.3: Es existieren genau zwei Wege durch den Diamanten hindurch, möchte man jeden Knoten genau einmal besuchen

Schritt 2 Wir definieren $M := n \cdot 2^n + 1$ in Abhängigkeit der Anzahl Knoten in G $|V| = n$. Die einzelnen Diamant-Graphen werden wie auf Abbildung 3.4 zu einem vollständigen Graphen $G_O = (V', E')$ zusammengeführt. Alle Kanten $\{E_i, E_{i+1}\}$ und ausserdem die Kante $\{E_n, E_1\}$ erhalten die Kosten 1.

Alle Kanten $\{S_i, N_j\}$ (alle Verbindung zwischen dem S -Knoten des einen Diamanten mit dem N -Knoten eines anderen Diamanten) erhalten die Kosten 1 genau dann, wenn $\{v_i, v_j\} \in E$, das heisst, wenn diese Kante bereits im Originalgraphen $G = (V, E)$ vorhanden war. Falls nicht, erhalten sie die Kosten M .

Da G_O ein vollständiger Graph ist, existieren noch viele weitere Kanten (jeder Knoten

muss mit jedem anderen Knoten verbunden werden); jede dieser Kanten erhält auch die Kosten M .

Die optimale Tour \bar{C} ist in Abbildung 3.4 eingetragen. Die Kosten $c_O(\bar{C})$ betragen genau $8 \cdot n$ ($7 \cdot n$ in jedem Diamanten plus jeweils die Verbindung zwischen den Diamanten).

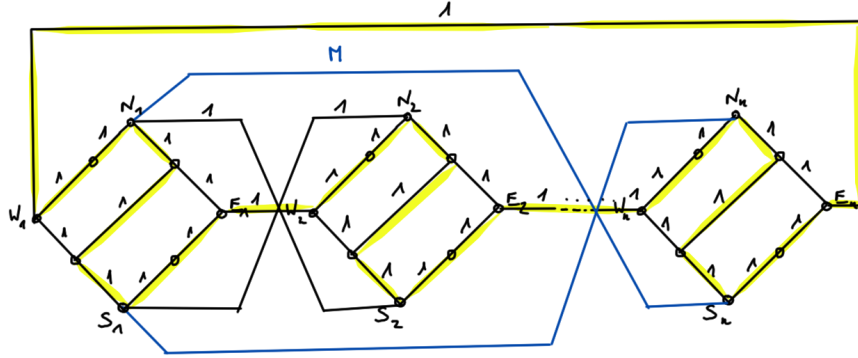


Abbildung 3.4:

Schritt 3 Damit ein Problem der Form $LM - TSP$ geschaffen werden kann, werden die Kosten genau einer Kante angepasst. Kante $\{E_N, W_1\}$ erhält neu die Kosten M . Die bisherige Lösung hat nun die Kosten $c_A(\bar{C}) = 8 \cdot n + M - 1$. Eine optimale Lösung hätte erneut auch unter der Kostenfunktion c_N die Kosten $8 \cdot n$ und würde sich jeweils von Nord nach Süd durch die Diamanten bewegen wie in Abbildung 3.5 eingezeichnet. Diese Tour hätte jedoch nur Kosten $8 \cdot n$, falls sich ein Hamilton-Kreis im Originalgraphen $G = (V, E)$ befindet (ansonsten wäre es nicht möglich, nur über Kanten mit Kosten 1 von einem Diamanten zum nächsten zu gelangen).

$$RHC \prec LM - \Delta_\beta - TSP$$

RHC Gegeben ist ein Graph $G = (V, E)$ mit $V = \{v_1, \dots, v_n\}$ und ein Hamilton-Pfad $P = (v_1, \dots, v_n)$, der nicht zu einem Hamilton-Kreis vervollständigt werden kann, indem man einfach die Endpunkte zusammenhängt (wie in Abbildung 3.6 dargestellt). Gesucht wird nun eine Antwort auf die Frage, ob der Graph G auch einen Hamilton-Kreis enthält. Dieses Problem ist NP-vollständig und wenn uns eine Polynomialzeit-Reduktion auf $LM - \Delta_\beta - TS$ mit $\beta > \frac{1}{2}$ gelingt, wissen wir, dass auch $LM - \Delta_\beta - TS$ mindestens NP-schwer ist.

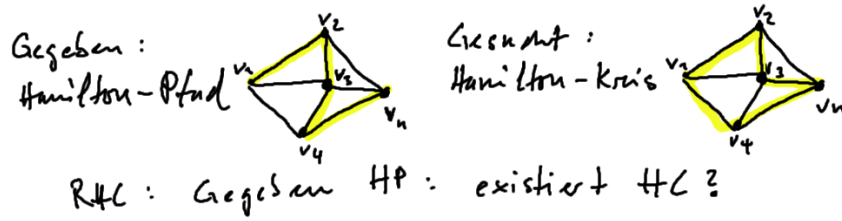


Abbildung 3.6:

Schritt 1 Wir konstruieren aus $G = (V, E)$ eine Eingabe $I = (G_O, c_N, c_O, \bar{C})$ für $LM - \Delta_\beta - TSP$. Sei $G_O = (V, \tilde{E})$ ein vollständiger Graph (mit den gleichen Knoten wie G) und c_O eine Kostenfunktion, welchen einer Kante e den Wert 1 zuordnet, falls e bereits eine Kante in G war ($e \in E$) oder falls $e = \{v_n, v_1\}$ ist (die Kante, welche im Originalgraphen noch gefehlt hätte, um aus dem Pfad trivialerweise einen Kreis zu machen). Alle anderen Kanten erhalten den Wert $2 \cdot \beta$ (bei $\beta > \frac{1}{2}$ ist dieser Wert immer grösser als 1).

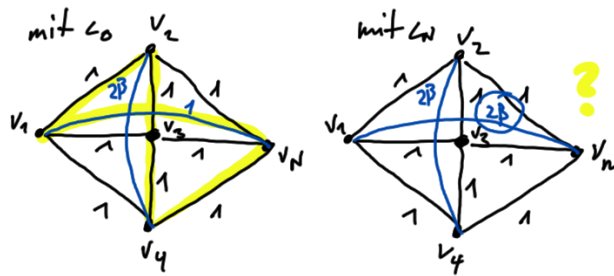


Abbildung 3.7:

Schritt 2 Die Kostenfunktion c_N unterscheidet sich in genau einer Kante und ordnet der Kante $\{v_n, v_1\}$ anstatt den Wert 1 den Wert $2 \cdot \beta$ zu. Eine optimale Tour \bar{C} bezüglich

c_O ist $\overline{C} = (v_1, v_2, \dots, v_n, v_1)$ (wie in Abbildung 3.7 links).

Schritt 3 - Konklusion Man kann leicht sehen, dass im Graph G_O unter der Kostenfunktion c_N genau dann eine Tour C mit Kosten n existiert, wenn im ursprünglichen Graph ein Hamilton-Kreis existiert und umgekehrt. Denn nur Kanten, welche im Originalgraphen bereits vorgekommen sind, haben die Kosten 1, alle anderen Kanten haben Kosten > 1 . \square

3.4.2 Ein Approximationsalgorithmus

Zuerst stellen wir etwas in einem Lemma fest, welches wir danach im Beweis brauchen werden:

Lemma 1. Sei $G = (V, E)$ ein Graph. Seien c_1 und c_2 Kostenfunktionen auf diesem Graphen, die in genau einer Kante e unterschiedlich sind und für die gilt $c_1(e) \leq c_2(e)$. Seien $\beta_1, \beta_2 \geq 1$ durch c_1 und c_2 auf E gegeben. Dann gilt, jede Kante, die direkt an e anschliesst hat mindestens die Kosten $\frac{c_2(e) - \beta_1 \beta_2 c_1(e)}{\beta_1 \beta_2 + \beta_2}$. Im Spezialfall $\beta_1 = \beta_2 = 1$ sind die Kosten mindestens $\frac{1}{2} \cdot (c_2(e) - c_1(e))$.

Beweis. Sei $G = (V, E)$ ein Graph. Seien c_1 und c_2 Kostenfunktionen auf diesem Graphen, die in genau einer Kante e unterschiedlich sind. Wähle zwei an e anschliessende Kanten $\{f, f'\}$ mit $f \neq f'$. Seien $\beta_1, \beta_2 \geq 1$ durch c_1 und c_2 auf E gegeben.

Dann gilt

$$c_2(e) \leq \beta_2 \cdot (c_2(f) + c_2(f'))$$

und

$$c(f') \leq \beta_1 \cdot (c_1(f) + c_1(e))$$

Daraus folgt:

$$\begin{aligned} c_2(e) &\leq \beta_2 \cdot [c_2(f) + \beta_1 \cdot (c_1(f) + c_1(e))] = c_2(e) \leq \beta_2 c_2(f) + \beta_1 \beta_2 c_1(f) + \beta_1 \beta_2 c_1(e) = c_2(e) - \\ &\beta_1 \beta_2 c_1(e) \leq \beta_2 c_2(f) + \beta_1 \beta_2 c_1(f) = \frac{c_2(e) - \beta_1 \beta_2 c_1(e)}{\beta_2} \leq c_2(f) + \beta_1 c_1(f) = \frac{c_2(e) - \beta_1 \beta_2 c_1(e)}{\beta_2} \leq \\ c_2(f)(1 + \beta_1) &= \frac{c_2(e) - \beta_1 \beta_2 c_1(e)}{\beta_2(1 + \beta_1)} \leq c_2(f) = \frac{c_2(e) - \beta_1 \beta_2 c_1(e)}{\beta_2 + \beta_1 \beta_2} \leq c_2(f) \end{aligned}$$

und somit:

$$c_2(f) \geq \frac{c_2(e) - \beta_1 \beta_2 c_1(e)}{\beta_2 + \beta_1 \beta_2}$$

der gleiche Beweis kann auf dieselbe Weise für c_1 geführt werden. \square

Ergebnis

Obwohl für $LM - \Delta_\beta - TSP$ kein Algorithmus existiert, der das Problem in Polynomialzeit exakt löst, können wir immer noch auf einen polynomiellen Approximationsalgorithmus mit konstanter relativer Güte hoffen. Tatsächlich haben H.-J. Böckenhauer et. al. ([1], [2]) gezeigt, dass es einen solchen Algorithmus gibt, welcher das Problem in Abhängigkeit von β (und nicht in Abhängigkeit der Anzahl Knoten n) approximiert. Er garantiert eine relative Güte von

$$\beta_L \beta_H \cdot \frac{15\beta_L^2 + 5\beta_L - 6}{10\beta_L^2 + 3\beta_L \beta_H + 3\beta_H - 6}$$

für Eingaben $I = (G, c_O, c_N)$ mit $G = (V, E)$, so dass c_O die Δ_{β_O} -Ungleichung und c_N die Δ_{β_N} -Ungleichung auf E erfüllt und $\beta_L := \min(\beta_O, \beta_N)$ und $\beta_H := \max(\beta_O, \beta_N)$.

Das heisst, die Kostenfunktion c_O ist so konstruiert, dass $c_O(\{u, w\}) = \beta_O \cdot (c_O(\{u, v\}) + c_O(\{v, w\}))$. c_N unterscheidet sich in genau einer Kante und erfüllt die Dreiecksungleichung $c_N(\{u, w\}) = \beta_N \cdot (c_N(\{u, v\}) + c_N(\{v, w\}))$. Beide Graphen G_O und G_N enthalten dieselben Knoten und Kante V und E .

Algorithmus 1

Der Algorithmus erhält als Eingabe den Graphen $G = (V, E)$, eine originale Kostenfunktion c_O , eine optimale Tour unter c_O \bar{C} und eine neue Kostenfunktion c_N .

1. Sei $e \in E$ die Kante im Graphen wo $c_O(e) \neq c_N(e)$ (die Kante, deren Kosten sich verändert haben). Wir stellen nun eine Menge P zusammen, welche ungeordnete Paare von Kanten $\{f, f'\}$ enthält. f und f' sind dabei Kanten, welche direkt an e angehängt sind (siehe Abbildung 3.8). Dazu unterscheiden wir zwei Fälle:

Fall 1: $c_O(e) < c_N(e)$ Die Kosten von e sind gestiegen. Wir bilden Paare $\{f, f'\}$ nur unter Kanten, welche sich auf der gleichen Seite von e befinden. Formal: $f \cap f' \cap e$ enthält nur genau 1 Element (z.B. $\{v_1, v_2\} \cap \{v_1, v_3\} \cap v_1, v_7 = \{v_1\}$ $|\{v_1\}| = 1$).

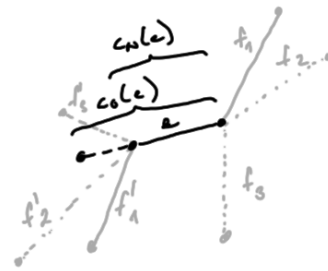
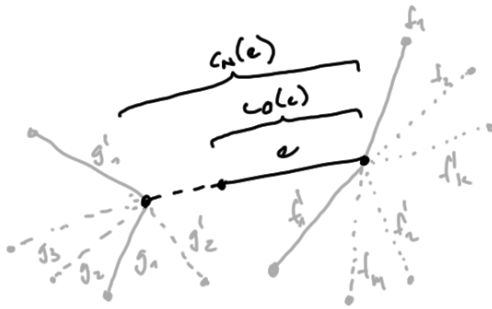
In Abbildung 3.8 bilden wir im linken Bild nur Paare unter den f und f' Kanten und unter der g und g' Kanten.

Fall 2: $c_O(e) > c_N(e)$ Wir bilden Paare $\{f, f'\}$ nur zwischen Kanten, die sich an entgegengesetzten Enden von e befinden und zusammen kein Dreieck bilden. Formal: $f \cup f' = \emptyset$ (z.B. $\{v_1, v_2\} \cap \{v_4, v_7\} = \emptyset$).

Dies ist in Abbildung 3.8 auf der rechten Seite dargestellt.

Fall 1 $c_O(e) < c_N(e)$

Fall 2 $c_O(e) > c_N(e)$



Alle mögl. Paare $\{f_i, f'_j\}$ bzw. $\{g_i, g'_j\}$
werden berücksichtigt.

Abbildung 3.8:

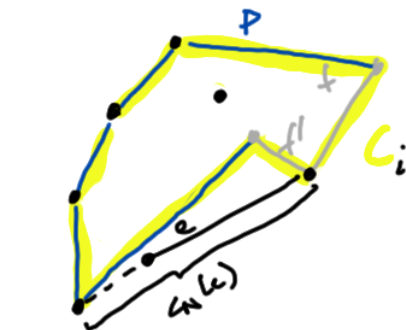
2. Danach wird für jedes Paar $\{f, f'\}$ aus P ein Hamilton-Pfad Q zwischen den Knoten aus f und f' , welche nicht auch in e vorkommen (formal $(f \cup e) \setminus e$), auf dem Graphen unter Benutzung der Knoten $V \setminus (e \cap (f \cup f'))$ berechnet. Das sind alle Knoten, ausser die, welche f, f' und e gemeinsam haben. Der Pfad Q wird durch die Kanten f und f' und – falls $c_O(e) > c_N(e)$ (Fall 2) – durch e ergänzt und bildet so einen Hamilton-Kreis $C_{\{f, f'\}}$.

Für die Berechnung des Pfades wird ein Algorithmus von Forlizzi et al. verwendet ([3]). Dieser Algorithmus approximiert den Pfad mit einer relativen Güte von $\frac{5}{3}\beta^2$.

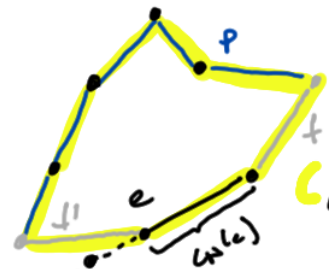
Der Vorgang ist in Abbildung 3.9 dargestellt.

Fall 1 $c_O(e) < c_N(e)$

Fall 2 $c_O(e) > c_N(e)$



e wird ausgelassen



e wird eingelassen

Beste Tour aller C_i auswählen.

Abbildung 3.9:

- Im letzten Schritt wählt der Algorithmus die Tour C mit den geringsten Kosten aus der Menge $\{\bar{C}\} \cup \{C_{\{f,f'\}} | \{f, f'\} \in P\}$ (die Menge aller berechneten Touren inklusive der optimalen Tour unter c_O aus).

Der Algorithmus gibt den Hamilton-Kreis C aus.

Beweis

Beweis. Sei $\{c_1, c_2\} = \{c_O, c_N\}$ so dass $c_1(e) \leq c_2(e)$. Das heisst, falls die Kante e kürzer wird, ist $c_1 = c_N$, falls sie grösser wird ist $c_1 = c_O$.

Wir können annehmen, dass Touren in G unter c_N die Kante e genau dann benutzen, wenn $c_1 = c_N$, das heisst, wenn die Kosten von e kleiner geworden sind. Falls sie das nicht tun würden, hätte e überhaupt keinen Einfluss auf das Geschehen und \bar{C} wäre weiterhin eine optimale Tour.

Wir wählen eine solche optimale Tour C_{OPT} in G unter c_N aus und nehmen an, dass die Tour die an e angrenzenden Kanten $\{f, f'\}$ benutzt (falls sie durch e geht, muss sie zwingend ein solches paar von Kanten nutzen).

Sei nun der Pfad P die Tour C_{OPT} ohne die Kanten f und f' und ohne e , falls $c_N(e) >$

$c_O(e)$. Dadurch ergibt sich eine relative Approximations-Garantie von

$$\frac{c_N(C_{OPT}) - c_N(P) + \frac{5}{3}\beta_L^2 c_N(P)}{OT_{G_{c_N}}}$$

(mit $OT_{G_{c_N}}$ = die Länge der optimalen Tour in G bezüglich c_N).

Dies kann zu

$$\frac{c_N(C_{OPT})}{OT_{G_{c_N}}} - \frac{c_N(P)}{OT_{G_{c_N}}} + \frac{\frac{5}{3}\beta_L^2 c_N(P)}{OT_{G_{c_N}}}$$

und schliesslich zu

$$1 - \frac{c_N(P)}{OT_{G_{c_N}}} + \frac{5}{3}\beta_L^2 \frac{c_N(P)}{OT_{G_{c_N}}}$$

vereinfacht werden.

Es werden nun zwei Variablen eingeführt:

$$\alpha = \frac{c_N(P)}{OT_{G_{c_N}}}$$

und damit wir nicht so viel schreiben müssen

$$\gamma = \beta_L \beta_H \cdot \frac{15\beta_L^2 + 5\beta_L - 6}{10\beta_L^2 + 3\beta_L \beta_H + 3\beta_H - 6} \quad (3.1)$$

damit vereinfachen wir den obigen Term auf:

$$\begin{aligned} 1 - \frac{c_N(P)}{OT_{G_{c_N}}} + \frac{5}{3}\beta_L^2 \frac{c_N(P)}{OT_{G_{c_N}}} \\ = 1 - \alpha + \frac{5}{3}\beta_L^2 \alpha \end{aligned}$$

Durch Auflösen der Ungleichung

$$1 - \alpha + \frac{5}{3}\beta_L^2 \alpha \leq \gamma$$

nach α finden wir heraus, dass sofern nicht gilt

$$\alpha > \frac{\gamma - 1}{\frac{5}{3}\beta_L^2 - 1} \quad (3.2)$$

bereits eine γ -Approximation vorliegt und wir fertig sind. Daher nehmen wir an, dass 3.2 gilt.

Wegen Lemma 1 wissen wir, dass jede an e angrenzende Kante mindestens die Länge $\frac{c_2(e) - \beta_1\beta_2c_1(e)}{\beta_1\beta_2 + \beta_2}$ haben. Ausserdem gilt $\frac{c_2(e) - \beta_1\beta_2c_1(e)}{\beta_1\beta_2 + \beta_2} \geq \frac{c_2(e) - \beta_L\beta_Hc_1(e)}{\beta_L\beta_H + \beta_H}$. Da beide Kanten aus $\{f, f'\}$ mindestens diese Länge haben müssen, können wir davon auch die Kleinste nehmen und es gilt immer noch:

$$\min\{c(f), c(f')\} \geq \frac{c_2(e) - \beta_L\beta_Hc_1(e)}{\beta_L\beta_H + \beta_H}$$

Daraus folgt

$$1 - \alpha \geq \frac{2}{OT_{G_{c_N}}} \cdot \frac{c_2(e) - \beta_L\beta_Hc_1(e)}{\beta_L\beta_H + \beta_H}$$

Dafür muss man sich daran erinnern, dass

$$1 - \alpha = 1 - \frac{c_N(P)}{OT_{G_{c_N}}} = \frac{OT_{G_{c_N}} - c_N(P)}{OT_{G_{c_N}}} = \frac{c_N(f) + c_N(f')(\underbrace{+c_N(e)})}{OT_{G_{c_N}}}$$

wobei $c_N(e)$ dabei ist oder nicht.

In jedem Fall gilt, dass $c_N(f) + c_N(f')(\underbrace{+c_N(e)})$ grösser oder gleich zweimal die Kosten der kleineren angrenzenden Kante $\min\{c(f), c(f')\}$ ist.

Zusammen mit 3.2 ergibt das

$$\begin{aligned} 1 - \frac{\gamma - 1}{\frac{5}{3}\beta_L^2 - 1} &\geq \frac{2}{OT_{G_{c_N}}} \cdot \frac{c_2(e) - \beta_L\beta_Hc_1(e)}{\beta_L\beta_H + \beta_H} \\ 1 - \frac{2}{OT_{G_{c_N}}} \cdot \frac{c_2(e) - \beta_L\beta_Hc_1(e)}{\beta_L\beta_H + \beta_H} &\geq \frac{\gamma - 1}{\frac{5}{3}\beta_L^2 - 1} \\ \frac{\gamma - 1}{\frac{5}{3}\beta_L^2 - 1} &\leq 1 - \frac{2}{OT_{G_{c_N}}} \cdot \frac{c_2(e) - \beta_L\beta_Hc_1(e)}{\beta_L\beta_H + \beta_H} \end{aligned}$$

durch Multiplikation von $\beta_L\beta_H + \beta_H$, durch Teilen durch 2 auf beiden Seiten erhalten wir

$$\begin{aligned} \frac{(\gamma - 1) \cdot (\beta_L\beta_H + \beta_H)}{\frac{10}{3}\beta_L^2 - 2} &\leq \frac{\beta_L\beta_H + \beta_H}{2} - \frac{c_2(e) - \beta_L\beta_H c_1(e)}{OT_{G_{c_N}}} \\ &(\dots \text{ und noch etwas umformen } \dots) \\ \frac{c_2(e) - \beta_L\beta_H c_1(e)}{OT_{G_{c_N}}} &\leq \frac{\beta_L\beta_H + \beta_H}{2} - \frac{(\gamma - 1) \cdot (\beta_L\beta_H + \beta_H)}{\frac{10}{3}\beta_L^2 - 2} \end{aligned}$$

Jetzt wird auf beiden Seiten $(\beta_L\beta_H - 1) \cdot \frac{c_1(e)}{OT_{G_{c_N}}}$ dazu addiert. Das ergibt

$$\frac{c_2(e) - c_1(e)}{OT_{G_{c_N}}} \leq \frac{\beta_L\beta_H + \beta_H}{2} - \frac{(\gamma - 1) \cdot (\beta_L\beta_H + \beta_H)}{\frac{10}{3}\beta_L^2 - 2} + (\beta_L\beta_H - 1) \cdot \frac{c_1(e)}{OT_{G_{c_N}}}$$

und weil $\frac{c_1(e)}{OT_{G_{c_N}}}$ immer kleiner als 1 ist, können wir dafür den Maximalwert 1 annehmen und die Ungleichung stimmt trotzdem noch. Damit bekommen wir

$$\begin{aligned} \frac{c_2(e) - c_1(e)}{OT_{G_{c_N}}} &\leq \frac{\beta_L\beta_H + \beta_H}{2} - \frac{(\gamma - 1) \cdot (\beta_L\beta_H + \beta_H)}{\frac{10}{3}\beta_L^2 - 2} + (\beta_L\beta_H - 1) \cdot 1 \\ &(\dots(\beta_L\beta_H - 1) \text{ mit 2 erweitern } \dots) \\ &\leq \frac{\beta_L\beta_H + \beta_H}{2} + \frac{2 \cdot \beta_L\beta_H - 2}{2} - \frac{(\gamma - 1) \cdot (\beta_L\beta_H + \beta_H)}{\frac{10}{3}\beta_L^2 - 2} \\ &\leq \frac{3\beta_L\beta_H + \beta_H - 2}{2} - \frac{(\gamma - 1) \cdot (\beta_L\beta_H + \beta_H)}{\frac{10}{3}\beta_L^2 - 2} \\ &\leq \frac{3}{2} \cdot \beta_L\beta_H + \frac{1}{2} \cdot \beta_H - 1 - \frac{(\gamma - 1) \cdot (\beta_L\beta_H + \beta_H)}{\frac{10}{3}\beta_L^2 - 2} \end{aligned}$$

Nun setzen wir den Wert aus 3.1 für γ ein und erhalten dadurch:

$$\begin{aligned}
\frac{c_2(e) - c_1(e)}{OT_{G_{c_N}}} &\leq \frac{3}{2} \cdot \beta_L \beta_H + \frac{1}{2} \cdot \beta_H - 1 - \frac{(\gamma - 1) \cdot (\beta_L \beta_H + \beta_H)}{\frac{10}{3} \beta_L^2 - 2} \\
&= \frac{\beta_L \beta_H + \beta_H}{2} - \frac{3}{2} \cdot \beta_L \beta_H + \frac{1}{2} \cdot \beta_H - 1 - \frac{(\beta_L \beta_H \cdot \frac{15\beta_L^2 + 5\beta_L - 6}{10\beta_L^2 + 3\beta_L \beta_H + 3\beta_H - 6} - 1) \cdot (\beta_L \beta_H + \beta_H)}{\frac{10}{3} \beta_L^2 - 2}
\end{aligned}$$

... nach mühsamen Berechnungen erhalten wir schliesslich ...

$$= \dots = \beta_L \beta_H \cdot \frac{15\beta_L^2 + 5\beta_L - 6}{10\beta_L^2 + 3\beta_L \beta_H + 3\beta_H - 6} - 1 = \gamma - 1$$

und damit

$$\begin{aligned}
\frac{c_2(e) - c_1(e)}{OT_{G_{c_N}}} &\leq \gamma - 1 \\
\Rightarrow 1 + \frac{c_2(e) - c_1(e)}{OT_{G_{c_N}}} &\leq \gamma \\
\Rightarrow \frac{OT_{G_{c_N}}}{OT_{G_{c_N}}} + \frac{c_2(e) - c_1(e)}{OT_{G_{c_N}}} &\leq \gamma \\
\Rightarrow \frac{OT_{G_{c_N}} + (c_2(e) - c_1(e))}{OT_{G_{c_N}}} &\leq \gamma
\end{aligned}$$

Dieses Ergebnis sagt uns, dass, falls die Kosten für die Kante e gestiegen sind (dann ist $c_2 = c_N$) und wir einfach weiterhin diese Kante benutzen und dadurch höchstens $c_2(e) - c_1(e)$ über dem Optimum sind, da wir ja bereits von einer optimalen Lösung ausgehen, ist unsere Approximierungsrate trotzdem noch kleiner als γ . Damit erreicht der Algorithmus in jedem Fall eine Approximierungsrate von $\gamma = \beta_L \beta_H \cdot \frac{15\beta_L^2 + 5\beta_L - 6}{10\beta_L^2 + 3\beta_L \beta_H + 3\beta_H - 6}$ \square

Im Spezialfall des metrischen TSP $LM - \triangle_1 - TSP = LM - \triangle - TSP$ betragen $\beta_L = \beta_H = 1$ und somit

$$\begin{aligned}
\gamma &= \beta_L \beta_H \cdot \frac{15\beta_L^2 + 5\beta_L - 6}{10\beta_L^2 + 3\beta_L \beta_H + 3\beta_H - 6} \\
&= 1 \cdot \frac{15 + 5 - 6}{10 + 3 - 6} \\
&= \frac{14}{10 + 3 + 3 - 6} = \frac{14}{10} = \frac{7}{5} = 1.4
\end{aligned}$$

und erzielt somit für das metrische TSP eine bessere relative Güte als der beste bekannte Algorithmus von Christofides, der nur $\frac{3}{2} = 1.5$ schafft

Zudem schlägt er alle bekannten Algorithmen für das $\triangle_\beta - TSP$ auf dem Intervall $1 < \beta < 3.34899$.

Literaturverzeichnis

- [1] H.-J. Böckenhauer, L. Forlizzi, J. Hromkovič, J. Kneis, J. Kupke, G. Proietti, P. Widmayer Reusing Optimal TSP Solutions for Locally Modified Input Instances (Extended Abstract), International Federation for Information Processing, Volume 209, Fourth IFIP International Conference of Theoretical Computer Science-TCS 2006, 2006, S. 251-270
- [2] H.-J. Böckenhauer, J. Hromkovič, T. Mömke, P. Widmayer, On the Hardness of Reoptimization, SOFSEM 2008, LNCS 4910, 2008, S. 50-65
- [3] L. Forlizzi, J. Hromkovič, G. Proietti, S. Seibert, On the stability of approximation for Hamiltonian path problems, Algorithmis Operations Research 1, 2006, S. 31-45