An approximation scheme for planar graph TSP

Ahmet Altinbüken

Abstract

In diesem Paper wird ein Approximationsschema für das *planare Traveling Salesman Problem* präsentiert. Dies soll letzten Endes als Einstieg für weitere komplexere Probleme gelten. Ein solches Problem ist z.B. das euklidische TSP, welches auch mindestens NP-Vollständig ist. Man könnte mit genauen Handgriffen das hier vorgestellte Schema so bearbeiten, sodass es auch eine Lösung für das Euklidische TSP approximiert.

Contents

1 Einführung

1.1 Traveling Salesman Problem

Das Traveling Salesman Problem (TSP) ist ein grundlegendes Problem der Informatik, genauer dem Feld der Graphentheorie. Die Frage, die es stellt ist folgende:
Gegeben sei ein Graph mit Kosten auf den Kanten; wie kann ich einen Rundgang in diesem Graphen durchführen, sodass ich jeden Knoten mindestens einmal besuche und wieder am Startknoten auskomme?
In diesem Paper beschäftigen wir uns hauptsächlich mit dem planaren TSP, ein TSP welches auf einem planaren Graphen arbeitet.

1.2 PTAS

Das im Abstract angesprochene Approximationsschema ist genauer gesagt ein *Polynomialzeit-Approximationsschema (PTAS)*. Hierbei approximiert das PTAS eine Lösung, die dem Idealwert um einen gewissen multiplikativen Faktor abweicht und es in $n^{\mathcal{O}(1/\epsilon)}$ löst. Genauer: Sei *Optimum* der ideale Lösungswert, dann ist ein Problem α-approximierbar, wenn $\alpha > 1$ und die gefundene Lösung höchstens $\alpha \cdot Optimum$ ist. Und weiter, wenn ein Problem mit n Knoten und gewünschter Fehlerschwelle ϵ in $n^{\mathcal{O}(1/\epsilon)}$ $(1+\epsilon)\cdot Optimum$ -approximierbar ist, dann hat es ein PTAS.

1.3 Erforderliche Strukturen

Diese folgenden Strukturen kommen im Verlauf zu Gebrauch:

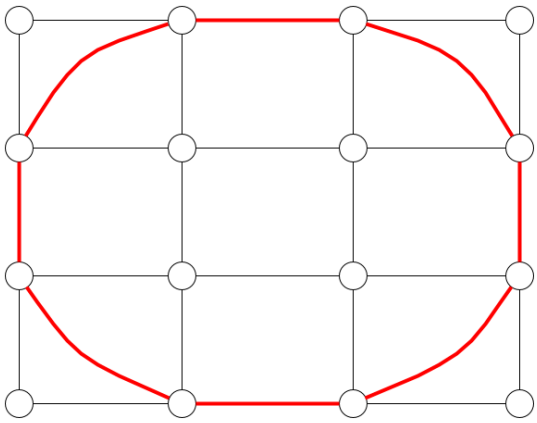
Face-edges

Gegeben sei ein eingebetteter Graph mit beliebig vielen faces. Eine face-egde ist eine Kante, die durch solch eine face verläuft. Diese Kante muss im ursprünglichen Graphen nicht existiert haben.

Cycle Separator

Gegeben sei ein eingebetteter Graph mit face-edges. Cycle separators werden benutzt um gegebenen Graphen, eventuell mit Hilfe von face-edges, zu unterteilen. Die resultierenden Teile sind der innere und äußere Graph, die wichtig für die Zerlegung des Algorithmus sind.

Figure 1: Cycle Separator in rot gemalt



In dieser Figur erkennt man, dass der rote Kreis ein separator ist. Hier sind die vier äußeren *Kurven* des Kreises face-edges des eingebetteten Graphen. Der innere Teil besteht trivialerweise aus den Knoten die im Kreis liegen und analog besteht der äußere Teil aus den Knoten die außerhalb des Kreises liegen.

1.4 Ziel

Zusammengefasst ist unser Ziel das Aufstellen eines PTAS für das Planargraph TSP. Gegeben ist uns ein planarer Graph G mit n Knoten und einer gewünschten Fehlerschwelle $\epsilon > 0$, sodass das TSP mit Laufzeit $n^{\mathcal{O}(1/\epsilon)}$ gelöst wird. Ebenso darf die gefundene Lösung den Wert $(1 + \epsilon) \cdot Optimum$ nicht überschreiten.
Weiterführend ist noch zu erwähnen, dass das TSP NP-Vollständig ist und der hier vorgestellte PTAS ein Ansatz zum Aufstellen eines PTAS für das Euklidische TSP ist.

2 Algorithmus

Das Problem sieht wie folgt aus: Gegeben sei ein zusammenhängender planarer Graph G mit n Knoten, der eingebettet ist bzw. werden kann. Nun wählt man sich eine Konstante $\epsilon > 0$, die als Fehlerschwelle fungiert. Gewünscht ist nun eine Tour T zu finden, sodass diese ein Gewicht um höchstens einen Faktor von $1 + \epsilon$ des Optimums hat. Es genügt ebenso innerhalb eines additiven Fehlers von ϵn zu bleiben.
Der hier vorgeführte Algorithmus basiert auf dem *divide and conquer* -Prinzip und wird deshalb auch in zwei Teilen erklärt. Im ersten Teil wird der Graph in mehrere kleine Untergraphen zerlegt. Diese werden dann weiter behandelt zu einem Baum und über diesen approximiert man dann Lösungen im zweiten Teil.

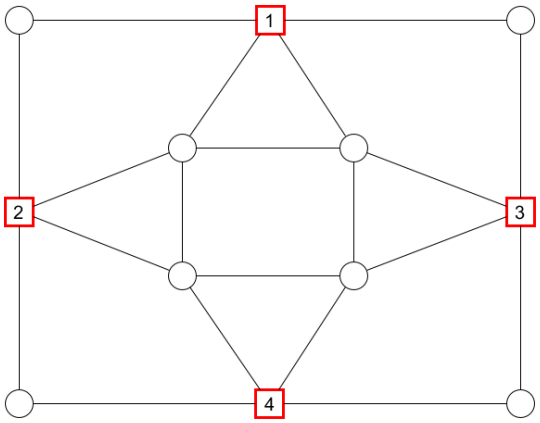
2.1 Teil 1: Zerlegung

Um den Graphen G zu zerlegen brauchen wir einen cycle separator, der genau das macht was wir hier brauchen. Dazu machen wir Gebrauch von folgendem Theorem, einer Anpassung von Millers Theorem (3) an unsere Bedürfnisse:

Theorem 1. (2) Sei ein zusammenhängender planarer Graph H gegeben. Dieser hat n Knoten mit jeweiligem Gewichten. Wähle ein f, sodass $1 \leq f \leq \sqrt{n}$ gilt. Es gibt nun einen planaren Kreis C, der höchstens f *face-edges* besitzt, der Rest sind normale Kanten. Außerdem haben der resultierende innere bzw. äußere Graph jeweils höchstens $\frac{2}{3}$ des Gesamtgewichts. Solch ein C kann in polynomieller Zeit gefunden werden.

Nun wird Theorem 1 iterativ angewendet, beginnend beim Ursprungsgraph G. Wähle zunächst das $f = \Theta((logn)/\epsilon)$. Zu betrachten ist hierbei, dass f abhängig von n und ϵ ist. Wähle ebenso $S = \Theta(f^2)$ als Abbruchbedingung, sodass nicht mehr weiter zerlegt wird wenn der zu betrachtende Graph S Knoten oder weniger besitzt. Das bedeutet, dass die nachher Blätter des noch zu konstruierenden Baumes höchstens S Knoten enthalten.
Betrachtet man nun Figure 1, so erkennt man das auf dem roten Kreis offensichtlich Pfade zwischen Knoten bestehen. Der erste Schritt der Zerlegung ist es diese Pfade zusammenzuziehen, aber nur solche, die auch im ursprünglichen Graphen vorhanden waren. Somit werden Pfade, die durch face-edges erzeugt wurden, nicht zusammengezogen. Es resultiert folgender zusammengezogener Graph G':

Figure 2: Zusammengezogener Graph G'



Wie man sieht behalten diese zusammengezogen Knoten ihre ein- und ausgehenden Kanten aus dem Ursprungsgraph. Der nächste Schritt trennt nun G' in die Untergraphen G_1 und G_2 auf. Hierbei wird vom separator Gebrauch gemacht, der den Graphen eben auch in Inneres und Äußeres

unterteilt. Zu erwähnen ist hier noch, dass sich beide Teile die zusammengezogenen Knoten auf dem Kreis teilen. Also in diesem Fall wären es die rot markierten Knoten in Figure 2.
Somit resultieren G_1 und G_2:

Figure 3: Untergraph G_1

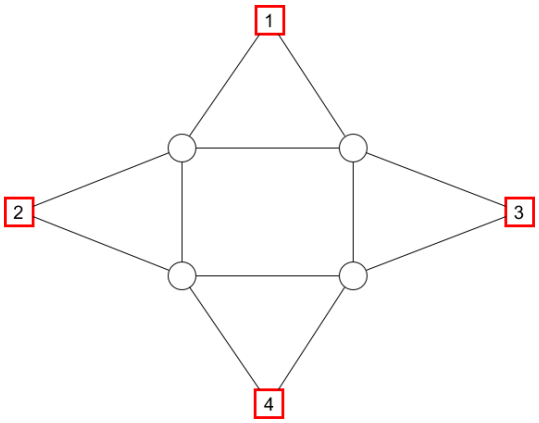
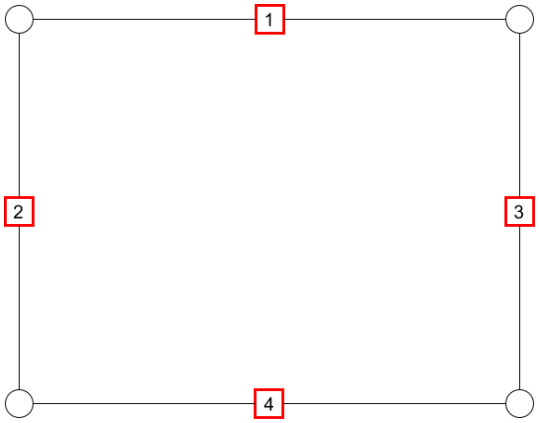


Figure 4: Untergraph G_2



Diese Untergraphen sind nun die Kinder von G im binären Zerlegungsbaum. Wie zuvor angesprochen werden die neu erzeugten Graphen weiter zerlegt bis sie klein genug sind. Hierzu ist noch zu beachten, dass jeder zusammengezogene Knoten aus dem aktuell betrachteten Graphen H das Gewicht $W(H)/6f$ bekommt. Dies schließt ebenso ältere Knoten dieser Art ein.
Betrachtet man nun den erzeugten Baum, so fallen einem besondere Einzelheiten auf. Zum einen hat das Kind im Baum höchsten 5/6 des Gewichts des Elternknotens. Dies lässt sich wie folgt zeigen:
Sei H_i Kind von H, dann gilt $W(H_i) \leq \frac{2}{3}W(H) + f \cdot \frac{W(H)}{6f} = \frac{5}{6}W(H)$. Der erste Teil folgt aus Theorem 1, und zwar dass die neuen Teile jeweils 2/3 des Gewicht des ursprünglichen Graphen haben. Der zweite Teil lässt sich aus der Tatsache folgern, dass es höchstens f viele zusammengezogene Knoten im resultierenden Graphen gibt und diese jeweils mit $W(H)/6f$ gewichtet sind.
Daraus folgert man weiter, dass jeder Knoten im Baum höchstens $5f$ zusammengezogene Knoten hat und da $S > 5f$, dass jedes innere Kind im Baum Knoten besitzt, die noch nicht zusammengezogen sind. Also haben diese Knoten weiterhin Gewicht 1 und der Baum somit eine Tiefe $D = log_{(6/5)}n < 4lgn$ unabhängig von ϵ und polynomiell groß. Die Zerlegung läuft in polynomieller Zeit durch.

2.2 Teil 2: Approximation

Gegeben sei nun ein binärer Baum mit Wurzel G und Blättern, die jeweils höchstens S Knoten haben. Alle Kinder und Blätter seien so zerlegt wie in Teil 1 beschrieben. Der Grundgedanke hier ist es von unten nach oben zu arbeiten und vorhandene Lösungen von Kindern zu einer Lösung des Elternknotens zu verschmelzen. Hierbei nehmen wir zunächst einmal an, dass die Lösungen der Blätter bereits gegeben sind.
Jetzt definieren wir uns eine (H, X)-Lösung folgendermaßen: Sei H ein zusammenhängender Graph und X eine gerade Teilmenge

der Knoten von H. Solch eine Lösung beinhaltet eine Menge von Pfaden, die jeden Knoten von H besuchen und dass die Endpunkte dieser Pfade in X liegen. Jeder Knoten aus X muss exakt einmal Endpunkt eines Pfades sein. Sollte X leer sein, so gibt es die Lösung einen Rundgang aus. Weiterhin definiere $c^*(H, X)$ als minimale Summe alle Pfade der (H, X)-Lösung.
Betrachtet man nun jedes H des Baumes mit Kindern H_1 und H_2. Wählt X so, dass es gerade *Teilmengen der zusammengezogenen Knoten* enthält, sodass man eine (H, X)-Lösung mit Kosten approximierbar zu $c^*(H, X)$ finden kann. Hierzu definiert man sich nun ein X' zu jedem X, welches die Elemente aus X beinhaltet, die vor dem Zusammenziehen einer ungeraden Anzahl von Knoten entsprechen. Sei nun X_1 die Menge der Knoten aus X', die sich auch in H_1 befinden. Und analog dazu X_2. Nun werden die Pfade aus H_1 und H_2 so zusammengeschlossen, dass im neu resultierenden H' die neuen Endpunkte aus $(X_1 \cap X_2) \cup (X_2 \cap X_1)$ sind.

Figure 5: Kindergraphen H_1 und H_2 mit eigenen Endpunkten

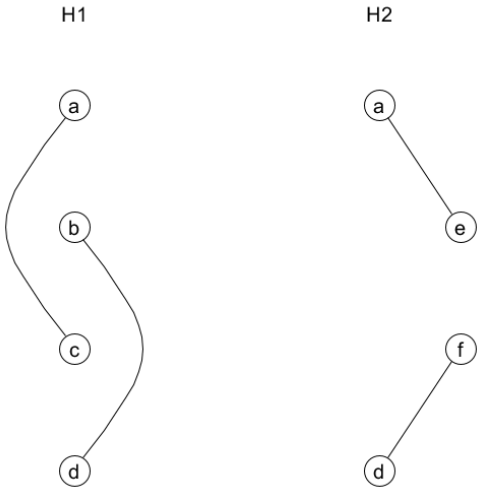
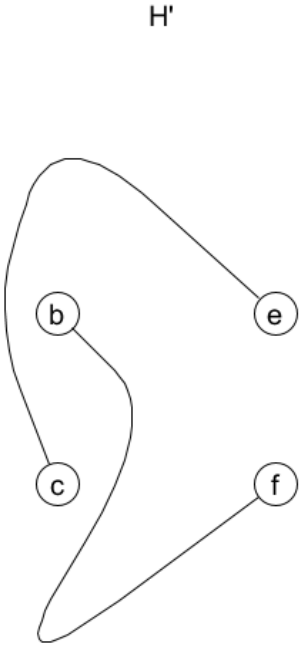


Figure 6: H' mit neuen Endpunkten



Wie in Figure 6 zu sehen, wurden die Pfade aus Figure 5 zusammengehangen und es bleiben die Endpunkte bestehen, die in mindestens einem der Kindergraphen aber nicht in beiden vorhanden waren. Es ist hierbei zu beachten, dass die alten Endpunkte a und d, welche nichts weiter sind als zusammengezogene Knoten, weiterhin auf dem Pfad existieren. Es kann passieren, dass bei diesen Schritten ein Kreis erzeugt wird, diesen kann man aber einfach mit wenigen zusätzlichen Kanten aufbrechen und zu anderen Pfaden verbinden.
Was man bisher geschaffen hat ist eine (H', X')-Lösung, die es nun gilt auf eine (H, X)-Lösung zu erweitern. Hierzu werden die zusammengezogenen Pfade verdoppelt um einen Rundgang zu generieren. Dieser Rundgang wird nun in die Lösung von H' eingepflegt. Es wird zumindest ein Knoten dieses Rundgangs in H' besucht.

So trifft man die Endpunkte aus X.
Wiederholt wird das Ganze bis eine Lösung zu $(G, \emptyset)$ ausgegeben werden kann.

2.3 Spezialfall: Blätter

Schauen wir uns nun den Basisfall, das Lösen der Blätter. Sei das Blatt ein zusammenhängender planarer Graph H mit Anzahl Knoten höchstens $S = \Theta(f^2)$ und maximal $5f$ zusammengezogenen Knoten. Annahme: $1/\epsilon \leq logn$.
Auf den Blättern wird der bekannte Algorithmus ausgeführt, aber mit einigen Änderungen. Und zwar ist das neue Limit $S' = \Theta(f)$, das heißt es wird so lange zerlegt bis die resultierenden Graphen höchstens so viele Knoten besitzen. Eine weitere Änderung ist das Gewicht der Knoten. Diesen sind nun immer 1, ob zusammengezogen oder nicht. Weiterhin wird zur Wahl des cycle separators nicht mehr ein fixes f gewählt. Stattdessen sucht man nun in jedem jeweiligen Graphen K nach cycles, die höchstens $\sqrt{|K|}$ face-edges und $\mathcal{O}(\sqrt{|K|})$ Knoten haben.
Es wird nun also wieder ein Baum gebaut, dessen Kinder mit der Tiefe deutlich kleiner werden. Hierbei stellt man fest, dass die Anzahl der zusammengezogenen Knoten im Baum sich auf $\mathcal{O}(\epsilon|H|/\sqrt{c'})$, mit c' als große Konstante, beschränkt.
Sobald der Baum aufgestellt ist, werden wieder Lösungen von unten nach oben approximiert. Die neuen Blätter sind zusammenhängende planare Graphen mit Größe S' zu denen man mit einem beliebigen X eine (K, X)-Lösung sucht. Hierzu genügt es alle Möglichkeiten von Multimengen auszuprobieren. Hat man die Blätter gelöst werden diese Lösungen wie zuvor gemerged und es wird eine Lösung für die Wurzel ausgegeben, welches unser Blatt vom ursprünglichen Baum war.
Die ausgegebene Lösung weicht vom Optimum um Faktor $\epsilon/4$ ab. Hierbei kann X beliebig sein. Es gibt $3^{\mathcal{O}(|K|)} = n^{\mathcal{O}(1/\epsilon)}$ viele Multimengen. Dies folgt aus der Anzahl der Kanten von K, eingeschränkt durch die Planarität von K.

3 Analyse

Laufzeit

Die Laufzeit des Algorithmus beschränkt sich auf $n^{\mathcal{O}(1/\epsilon)}$. Jeder Schritt der Zerlegung lässt sich in $n^{\mathcal{O}(1)}$ ausführen. Die Approximation hingegen ist langsamer, aber immer noch in unserem vereinbarten Limit. Sowohl ein Approximationsschritt eines Blattes als auch inneren Knotens benötigt $n^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1/\epsilon)} = n^{\mathcal{O}(1/\epsilon)}$. Die Laufzeit der kompletten Approximation lautet $n^{\mathcal{O}(1)} \cdot n^{\mathcal{O}(1/\epsilon)} = n^{\mathcal{O}(1/\epsilon)}$.

Fehler

Es liegen drei Fehlerquellen vor: der Blatt-, Merge- und Erweiterungsfehler.
Beim Blattfehler, dem Basisfall, wird eine Lösung mit Fehler $\epsilon/4$ ausgegeben.
Beim Zusammenführen der Kinderlösungen müssen eventuell entstandene Kreise aufgebrochen und neu verbunden werden. Dies ist mit $f/2$ Kanten zu bewerkstelligen. Ebenso weitere $\mathcal{O}(f)$, da die Endpunkte zweimal besucht werden, obwohl eben nur einmal genügen würde. Dies addiert sich zusammen zu einem Fehler von $2\mathcal{O}(f)$.
Beim Erweitern von H' auf H werden bis zu $\mathcal{O}(|H|/f)$ neue Kanten benutzt.
Durch bestimmte Fehlerkorrekturen beläuft sich der additive Fehler auf ϵn, wie gewünscht.

4 Diskussion

Zusammengefasst wurde hier das PTAS von Grigni, Koutsoupias und Papadimitriou für den Planargraph TSP vorgestellt. Der Ursprungsgraph wird solange unterteilt, bis die neu entstandenen Graphen höchstens Größe S haben. Danach werden Kinderlösungen von unten nach oben zusammengeführt bis letzten Endes eine Lösung für den Ursprungsgraph ausgegeben werden kann. Dieser gibt für das Problem eine Lösung mit Abweichung um Faktor ϵn in Zeit $n^{\mathcal{O}(1/\epsilon)}$ an.
Das Ergebnis von Grigni führte letzten Endes dazu, dass Sanjeev Arora und Joseph S. B. Mitchell einen PTAS (1) für das euklidische TSP entdecken konnten. Die Laufzeit dazu beläuft sich auf $\mathcal{O}(n(\log n)^{(\mathcal{O}(c\sqrt{d}))^{d-1}})$.

References

1 Arora, S. : Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. In: *Journal of the ACM (JACM)* 45 (1998), Nr. 5, S. 753–782

2 Grigni, M. ; Koutsoupias, E. ; Papadimitriou, C. : An approximation scheme for planar graph TSP. In: *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on* IEEE, 1995, S. 640–645

3 Miller, G. L.: Finding small simple cycle separators for 2-connected planar graphs. In: *Journal of Computer and system Sciences* 32 (1986), Nr. 3, S. 265–279