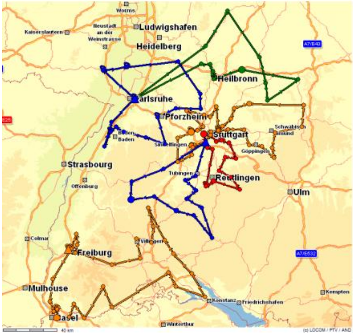
An approximation scheme for planar graph TSP

Ahmet Altinbüken

13. Juli 2018

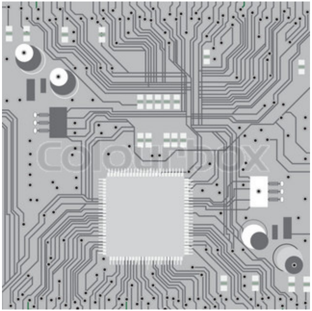
Warum brauchen wir das?

- Tourenplanung



Warum brauchen wir das?

- Tourenplanung
- Mikrochips



Warum brauchen wir das?

- Tourenplanung
- Mikrochips
- DNA-Sequenzierung (in abgewandelter Form)

TSP

kurze Wiederholung

- kürzester Durchlauf durch einen Graph der alle Knoten besucht

TSP

kurze Wiederholung

- kürzester Durchlauf durch einen Graph der alle Knoten besucht
- NP-Vollständig

TSP

kurze Wiederholung

- kürzester Durchlauf durch einen Graph der alle Knoten besucht
- NP-Vollständig
- hier: planarer Graph, auf dem der kürzeste Rundgang gesucht wird, der wieder am Startknoten auskommt

Approximation - der Gedanke

Gegeben sei ein Minimierungsproblem und der dazugehörige optimale Lösungswert. Dann:

Approximation - der Gedanke

Gegeben sei ein Minimierungsproblem und der dazugehörige optimale Lösungswert. Dann:

- ist das Problem bei einem $\alpha > 1$ mit Lösungswert höchstens $\alpha \cdot$ *Optimum* α-**approximierbar**

Approximation - der Gedanke

Gegeben sei ein Minimierungsproblem und der dazugehörige optimale Lösungswert. Dann:

- ist das Problem bei einem $\alpha > 1$ mit Lösungswert höchstens $\alpha \cdot$ *Optimum* α-**approximierbar**
- hat das Problem ein **PTAS**, wenn es in $n^{\mathcal{O}(1/\epsilon)}$ mit Lösungswert $(1 + \epsilon) \cdot$ *Optimum* approximiert werden kann

Ziel

Finde einen **PTAS** für das Planargraph TSP:

Ziel

Finde einen **PTAS** für das Planargraph TSP:
Gegeben sei ein planarer Graph G mit n Knoten und einem Parameter $\epsilon > 0$, sodass gelten muss:

- Laufzeit von $n^{\mathcal{O}(1/\epsilon)}$

Ziel

Finde einen **PTAS** für das Planargraph TSP:
Gegeben sei ein planarer Graph G mit n Knoten und einem Parameter $\epsilon > 0$, sodass gelten muss:

- Laufzeit von $n^{\mathcal{O}(1/\epsilon)}$
- eine Lösung des TSP mit Länge höchstens $(1+\epsilon) \cdot \mathit{Optimum}$ finden

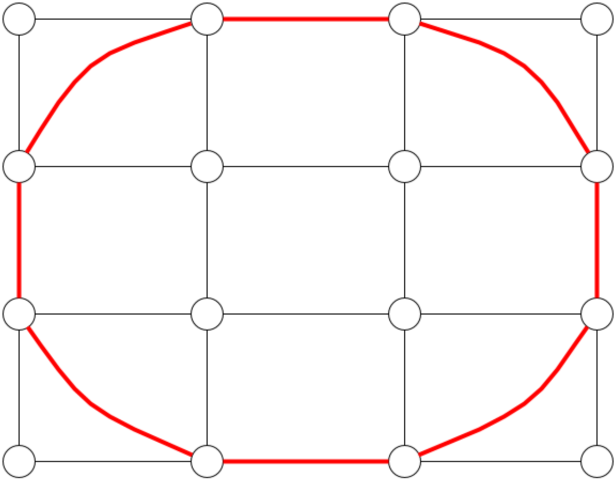
Cycle Separator

- wird benutzt um einen Graphen in die Teile A, B und C zu unterteilen

Cycle Separator

- wird benutzt um einen Graphen in die Teile A, B und C zu unterteilen
- A ist der innere Teil
- B der äußere
- C ist ein Kreis

Cycle Separator



Face edges

Eine Face edge ist eine Kante, die durch ein face geht. Diese können ebenso in einem Kreis verwendet werden.

Ansatz

Theorem

Sei ein zusammenhängender planarer Graph H gegeben. Dieser hat n Knoten mit jeweiligem Gewichten. Wähle ein f, sodass $1 \leq f \leq n$ gilt. Es gibt nun einen planaren Kreis C, der höchstens f *face edges* besitzt, der Rest sind normale Kanten. Außerdem haben die Teile A und B jeweils höchstens $\frac{2}{3}$ des Gesamtgewichts.

Zerlegung: Der Plan

Gegeben sei ein planarer Graph G mit n Knoten.

Zerlegung: Der Plan

Gegeben sei ein planarer Graph G mit n Knoten.
Wähle nun $f = \Theta((\log n)/\epsilon)$.

Zerlegung: Der Plan

Unterteile G in zwei Teilgraphen mithilfe des *separator*, dann:

Zerlegung: Der Plan

Unterteile G in zwei Teilgraphen mithilfe des *separator*, dann:

1. Pfade zusammenziehen

Zerlegung: Der Plan

Unterteile G in zwei Teilgraphen mithilfe des *separator*, dann:

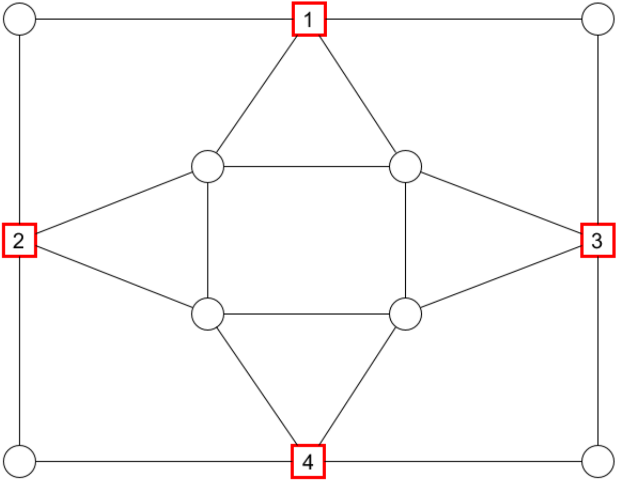
1. Pfade zusammenziehen
2. Zusammengezogenen Knoten das Gewicht $\frac{|G|}{6f}$ geben

Zerlegung: Der Plan

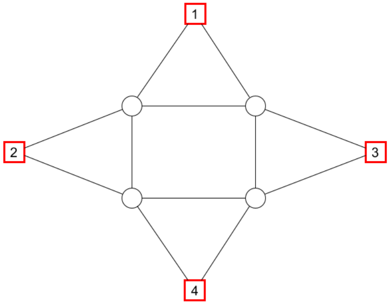
Unterteile G in zwei Teilgraphen mithilfe des *separator*, dann:

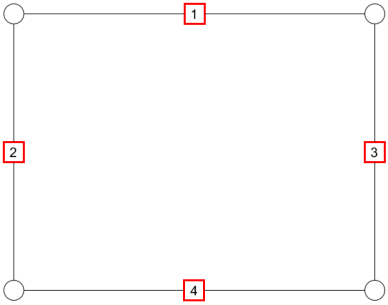
1. Pfade zusammenziehen
2. Zusammengezogenen Knoten das Gewicht $\frac{|G|}{6f}$ geben
3. Wiederhole diese Schritte rekursiv bis Teilgraph klein genug ist

Pfade zusammenziehen



Teilgraphen G_1 und G_2





Der Baum

- G ist Wurzel

Der Baum

- G ist Wurzel
- Kinder von G sind G_1 und G_2

Der Baum

- G ist Wurzel
- Kinder von G sind G_1 und G_2
- Blätter haben "Größe" $\leq S = \Theta(f^2)$

Einzelheiten des Baums

- Kind wiegt höchstens $\frac{5}{6}$ des Elternknotens

Einzelheiten des Baums

- Kind wiegt höchstens $\frac{5}{6}$ des Elternknotens
- alle Teilgraphen haben maximal $5f$ zusammengezogene Knoten

Einzelheiten des Baums

- Kind wiegt höchstens $\frac{5}{6}$ des Elternknotens
- alle Teilgraphen haben maximal $5f$ zusammengezogene Knoten
- Baum hat polynomielle Tiefe: $\mathcal{O}(log n)$

Einzelheiten des Baums

- Kind wiegt höchstens $\frac{5}{6}$ des Elternknotens
- alle Teilgraphen haben maximal $5f$ zusammengezogene Knoten
- Baum hat polynomielle Tiefe: $\mathcal{O}(logn)$
- jeder Schritt der Zerlegung ist in polynomieller Zeit durchführbar

Idee

Nun wird der Baum benutzt, um approximierte Lösungen für die Blätter zu finden.

Idee

Nun wird der Baum benutzt, um approximierte Lösungen für die Blätter zu finden.
Die Lösungen von Kindern *merged* man dann, während man den Baum weiter hoch läuft.

Vorbereitung

Sei H ein zusammenhängender Graph, dann:

- wähle Teilmengen X aller Knoten von H

Vorbereitung

Sei H ein zusammenhängender Graph, dann:

- wähle Teilmengen X aller Knoten von H
- diese Teilmengen müssen gerade Anzahl Elemente enthalten

Lösungen speichern

- (H, X)-Lösung sammelt Pfade

Lösungen speichern

- (H, X)-Lösung sammelt Pfade
- Falls $X = \emptyset$, gib TSP aus

Lösungen speichern

- (H, X)-Lösung sammelt Pfade
- Falls $X = \emptyset$, gib TSP aus
- Tabelle $T[H, X]$ speichert Lösungen

Lösung auf Blättern

Blätter sind Spezialfälle.

Lösung auf Blättern

Blätter sind Spezialfälle. Diese können mit Fehler $\epsilon/4$ gelöst werden.

Lösung auf Blättern

Blätter sind Spezialfälle. Diese können mit Fehler $\epsilon/4$ gelöst werden. Wiederhole für alle X.

Mergen

Gegeben sei Graph H mit Kindern H_1 und H_2:

Mergen

Gegeben sei Graph H mit Kindern H_1 und H_2:

- Definiere X' als *mod-2 contraction* von X

Mergen

Gegeben sei Graph H mit Kindern H_1 und H_2:

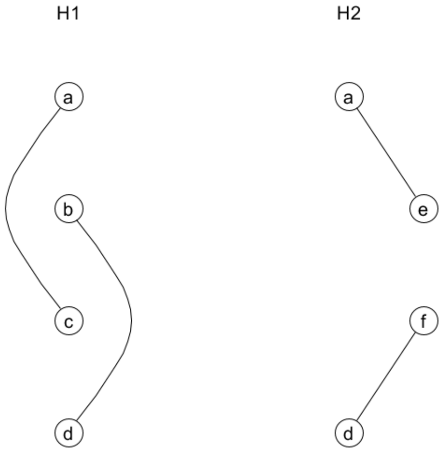
- Definiere X' als *mod-2 contraction* von X
- Definiere weiter X_1 und X_2 für ihre jeweiligen Graphen

Mergen

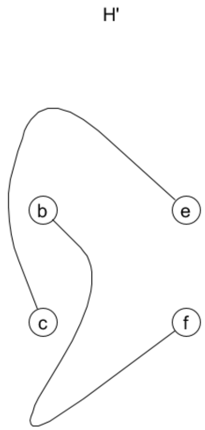
Gegeben sei Graph H mit Kindern H_1 und H_2:

- Definiere X' als *mod-2 contraction* von X
- Definiere weiter X_1 und X_2 für ihre jeweiligen Graphen
- X' ist symmetrische Differenz von X_1 und X_2

Mergen



Mergen



Mergen

- Wähle minimierendes Y^* für beiden Kinder

Mergen

- Wähle minimierendes Y^* für beiden Kinder
- Verbinde Kinder in H'

Mergen

- Wähle minimierendes Y^* für beiden Kinder
- Verbinde Kinder in H'
- möglich entstandene Kreise aufbrechen

Mergen

- Wähle minimierendes Y^* für beiden Kinder
- Verbinde Kinder in H'
- möglich entstandene Kreise aufbrechen
- Lösung (H', X') erweitern auf (H, X)

Mergen

- Wähle minimierendes Y^* für beiden Kinder
- Verbinde Kinder in H'
- möglich entstandene Kreise aufbrechen
- Lösung (H', X') erweitern auf (H, X)
- Wiederhole bis $X = \emptyset$

Blätter - Der Basisfall

Gegeben sei ein zusammenhängender planarer Graph H.

Blätter - Der Basisfall

Gegeben sei ein zusammenhängender planarer Graph H.

Dieser hat Größe höchstens $S = \Theta(f^2)$ mit maximal $5f$ zusammengezogenen Knoten.

Blätter - Der Basisfall

Gegeben sei ein zusammenhängender planarer Graph H.

Dieser hat Größe höchstens $S = \Theta(f^2)$ mit maximal $5f$ zusammengezogenen Knoten.

Annahme: $1/\epsilon \leq \log n$

Weitere Zerlegung

Ähnlich wie zuvor, mit folgenden Änderungen:

Weitere Zerlegung

Ähnlich wie zuvor, mit folgenden Änderungen:

- neues Limit $S' = \Theta(f)$

Weitere Zerlegung

Ähnlich wie zuvor, mit folgenden Änderungen:

- neues Limit $S' = \Theta(f)$
- **alle** Knoten haben Gewicht 1

Weitere Zerlegung

Ähnlich wie zuvor, mit folgenden Änderungen:

- neues Limit $S' = \Theta(f)$
- **alle** Knoten haben Gewicht 1
- f wird nicht mehr benutzt

Der Vorgang

- Baue wieder einen Baum von H

Der Vorgang

- Baue wieder einen Baum von H
- Kinder werden mit Tiefe immer kleiner

Der Vorgang

- Baue wieder einen Baum von H
- Kinder werden mit Tiefe immer kleiner
- Beschränke Anzahl zusammengezogener Knoten auf $\mathcal{O}(\epsilon|H|/\sqrt{c'})$

Der Vorgang

- Baue wieder einen Baum von H
- Kinder werden mit Tiefe immer kleiner
- Beschränke Anzahl zusammengezogener Knoten auf $\mathcal{O}(\epsilon|H|/\sqrt{c'})$
- Teste alle Möglichkeiten von Multimengen um Lösung zu finden

Laufzeit

- Zerlegung: wird $n^{\mathcal{O}(1)}$ mal ausgeführt, zusammengefasst immer noch in $n^{\mathcal{O}(1)}$

Laufzeit

- Zerlegung: wird $n^{\mathcal{O}(1)}$ mal ausgeführt, zusammengefasst immer noch in $n^{\mathcal{O}(1)}$
- Approximation: für Blätter und Kinder jeweils $n^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1/\epsilon)}$, für den ganzen Baum $n^{\mathcal{O}(1/\epsilon)}$

Laufzeit

- Zerlegung: wird $n^{\mathcal{O}(1)}$ mal ausgeführt, zusammengefasst immer noch in $n^{\mathcal{O}(1)}$
- Approximation: für Blätter und Kinder jeweils $n^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1/\epsilon)}$, für den ganzen Baum $n^{\mathcal{O}(1/\epsilon)}$

komplette Laufzeit des Algorithmus: $n^{\mathcal{O}(1/\epsilon)}$

Fehler

3 Fehlerarten gibt es bei diesem Algorithmus:

1. Blattfehler mit $\epsilon/4$

Fehler

3 Fehlerarten gibt es bei diesem Algorithmus:

1. Blattfehler mit $\epsilon/4$
2. Mergefehler mit $2\mathcal{O}(f)$

Fehler

3 Fehlerarten gibt es bei diesem Algorithmus:

1. Blattfehler mit $\epsilon/4$
2. Mergefehler mit $2\mathcal{O}(f)$
3. Erweiterungsfehler mit $\mathcal{O}(|H|/f)$ neuen Kanten

Fehler

Es lassen sich hier und da einige Fehlerkorrekturen durchführen, was zur Folge hat:

Fehler

Es lassen sich hier und da einige Fehlerkorrekturen durchführen, was zur Folge hat:

der additive Fehler beläuft sich auf ϵn

Danke fürs Zuhören.

Danke fürs Zuhören.

Anmerkung

Sanjeev Arora und Joseph S. B. Mitchell wurden 2010 mit dem Gödel Preis ausgezeichnet.
Grund dafür ist ihr PTAS für das Euklidische TSP mit Laufzeit $\mathcal{O}(n(\log n)^{(\mathcal{O}(c\sqrt{d}))^{d-1}})$.