Fortgeschrittene Netzwerk- und Graph-Algorithmen

Prof. Dr. Hanjo Täubig

Lehrstuhl für Effiziente Algorithmen
(Prof. Dr. Ernst W. Mayr)
Institut für Informatik
Technische Universität München

Wintersemester 2010/11

Weitere Sätze

Satz

In einem Graphen $G = (V, E)$ gibt es für jede einem Crossing Cut entsprechende Partition P von V in 4 disjunkte Mengen eine zirkuläre Partition in G, die eine Verfeinerung von P ist.

Satz

Ein Graph $G = (V, E)$ hat $\mathcal{O}\left(\binom{|V|}{2}\right)$ viele Minimum Cuts (und diese Schranke ist scharf).

Kaktus-Vorbereitung: Laminare Mengen

Definition

Eine Menge $\mathcal{S}$ von Mengen heißt laminar, wenn für jedes Paar von Mengen $S_1, S_2 \in \mathcal{S}$ gilt, dass entweder S_1 und S_2 disjunkt sind, oder eine der beiden Menge die andere enthält.

- Jede laminare Menge $\mathcal{S}$ kann als Baum repräsentiert werden.
- Jeder Knoten repräsentiert eine Menge in $\mathcal{S}$.
- Die Blätter des Baums repräsentieren die Mengen von $\mathcal{S}$, die keine anderen Mengen enthalten.
- Der Vater eines zur Menge T gehörigen Knotens repräsentiert die (eindeutige) kleinste Übermenge von T.
- Die Konstruktion liefert eine Menge von Bäumen, deren Wurzelknoten Mengen repräsentieren, die in keiner anderen Menge von $\mathcal{S}$ enthalten sind.

Kaktus-Vorbereitung: Laminare Mengen

- Hinzufügen eines künstlichen Wurzelknotens, der mit allen eigentlichen Wurzeln verbunden ist, liefert einen Baum.

⇒ Die Knoten eines Baums repräsentieren alle Mengen von $\mathcal{S}$, wobei die Wurzel die zugrundeliegende Gesamtmenge (Vereinigung aller Mengen) repräsentiert.

- Wenn diese Vereinigung n Elemente enthält, kann der Baum höchstens n Blätter bzw. $2n - 1$ Knoten haben.

Kaktus

Definition

Ein zusammenhängender Graph heißt Kaktus, falls jedes Paar von einfachen Kreisen höchstens einen Knoten gemeinsam hat.
Ein Graph bestehend aus einem einzelnen Knoten wird als *trivialer Kaktus* bezeichnet.

Ein Graph ist ein Kaktus genau dann, wenn jede Zweifachzusammenhangskomponente entweder ein einfacher Kreis oder eine einzelne Kante (Brücke) ist.

Hinweis: Oft wird in der Definition auch einfach verlangt, dass jede Kante zu genau einem (knoten-disjunkten) Kreis gehört, wobei eine Doppelkante zwischen zwei Knoten einen Kreis der Länge 2 darstellt.
Man kann einen Kaktus nach dieser Definition erhalten, indem man die Brücken durch Doppelkanten ersetzt.

Kaktus-Repräsentation von Minimum Cuts

- Betrachte
  - einen Graph G,
  - einen ungewichteten Kaktus $\mathcal{R}$ und
  - eine Abbildung φ der Graphknoten in die Menge der Kaktusknoten $\varphi: \ V(G) \to V(\mathcal{R})$.

- Die Menge $V(\mathcal{R})$ kann einen Knoten x enthalten, der nicht in der Bildmenge der Abbildung φ enthalten ist, für den also $V(G)$ keinen Knoten v mit $\varphi(v) = x$ enthält.
  Ein solcher Knoten wird *leerer Knoten* genannt.

- Für jeden nichttrivialen (ungewichteten) Kaktus $\mathcal{R}$ gilt $\lambda(\mathcal{R}) \leq 2$ bzw. $\lambda(\mathcal{R}) = 2$ (je nach Definition).

Kaktus-Repräsentation von Minimum Cuts

Sei $\mathcal{C}(\mathcal{R})$ die Menge aller Minimum Cuts vom Kaktus $\mathcal{R}$.
D.h. $\{S, V(\mathcal{R}) \setminus S\} \in \mathcal{C}(\mathcal{R})$ gilt genau dann, wenn $E(S, V(\mathcal{R}) \setminus S; R)$ eine Menge von zwei Kanten ist, die zum gleichen Kreis in $\mathcal{R}$ gehören.

Definition

Für eine gegebene Teilmenge $\mathcal{C}' \subseteq \mathcal{C}(G)$ von Minimum Cuts, nennt man ein Paar $(\mathcal{R}, \varphi)$, das aus einem Kaktus $\mathcal{R}$ und einer Knotenabbildung φ besteht, Kaktus-Repräsentation für $\mathcal{C}'$ falls folgende Bedingungen erfüllt sind:

1. Für einen beliebigen Kaktus-MinCut $\{S, V(\mathcal{R}) \setminus S\} \in \mathcal{C}(\mathcal{R})$ gehört der Cut $\{X, \bar{X}\}$ mit $X = \{u \in V(G) \mid \varphi(u) \in S\}$ und $\bar{X} = \{u \in V(G) \mid \varphi(u) \in V(\mathcal{R}) \setminus S\}$ zu $\mathcal{C}'$.
2. Für jeden MinCut $\{X, \bar{X}\} \in \mathcal{C}'$ existiert ein Kaktus-MinCut $\{S, V(\mathcal{R}) \setminus S\} \in \mathcal{C}(\mathcal{R})$ mit $X = \{u \in V(G) \mid \varphi(u) \in S\}$ und $\bar{X} = \{u \in V(G) \mid \varphi(u) \in V(\mathcal{R}) \setminus S\}$.

Kaktus-Repräsentation aller Minimum Cuts

Fallunterscheidung:

1. Graph ohne zirkuläre Partitionen
2. Graph mit genau einer zirkulären Partition
3. Graph mit mehreren zirkulären Partitionen $P_1, \ldots, P_z$

Kaktus-Repräsentation aller Minimum Cuts

1. Fall: Graph ohne zirkuläre Partitionen

- Wenn es Crossing (Minimum) Cuts geben würde, müsste es laut Lemma auch eine zirkuläre Partition geben, die eine Verfeinerung der 4 entsprechenden disjunkten Mengen ist.

$\Rightarrow$ Es kann in diesem Fall keine Crossing Cuts geben.

$\Rightarrow$ Die Minimum Cuts (repräsentiert durch die jeweils kleinere Knotenmenge) sind laminar.

$\Rightarrow$ Die Minimum Cuts können durch einen Baum T_G repräsentiert werden.

Kaktus-Repräsentation aller Minimum Cuts

(Fortsetzung 1. Fall: Graph ohne zirkuläre Partitionen)
Repräsentation der Minimum Cuts durch folgenden Baum T_G:

- Betrachte die jeweils kleinere Knotenmenge jedes Minimum Cuts und bezeichne die Menge dieser Knotenmengen mit Λ
(wähle bei gleicher Kardinalität irgendeine von beiden).
- Repräsentiere jede Menge von Λ durch einen Knoten in T_G.
- Zwei Baumknoten, die zu den MinCut-Mengen A und B im Graphen gehören, sollen genau dann verbunden sein, wenn $A \subset B$ gilt und es keinen MinCut C mit $A \subset C \subset B$ gibt (der echte Obermenge von A und echte Teilmenge von B ist).
- Die Wurzeln der resultierenden Bäume repräsentieren die MinCuts in Λ, die in keiner anderen MinCut-Menge von Λ enthalten sind.
- Hinzufügen eines künstlichen Wurzelknotens und Verbinden mit den Wurzeln aller Bäume resultiert in einem Baum (T_G).

Kaktus-Repräsentation aller Minimum Cuts

(Fortsetzung 1. Fall: Graph ohne zirkuläre Partitionen)

- Definiere Abbildung:
  - Jeder Knoten des Graphen G wird auf den Knoten des Baums T_G abgebildet, der zu dem MinCut mit kleinster Kardinalität gehört, der diesen Knoten enthält.
  - Jeder nicht zugeordnete Knoten wird der Wurzel zugeordnet.
- Für jeden Minimum Cut S von G werden die Knoten von S einer Menge X von Baumknoten zugeordnet, so dass es eine Kante gibt, die beim Entfernen die Baumknoten X vom Rest des Baums trennt.
- Andererseits zerfällt beim Entfernen einer Kante aus T_G die Menge der Baumknoten so in zwei Teile, dass die Menge der Knoten, die in dem einen Teil zugeordnet werden, die eine Seite eines Minimum Cuts bilden.

$\Rightarrow$ Wenn der Graph keine zirkulären Partitionen enthält, dann ist der Baum T_G der Kaktus C_G des Graphen G und die Anzahl seiner Knoten ist durch $2|V| - 1$ beschränkt.

Kaktus-Repräsentation aller Minimum Cuts

2. Fall: Graph mit genau einer zirkulären Partition $V_1, \ldots, V_k$.

- Die Circular Partition Cuts können durch einen Kreis mit k Knoten repräsentiert werden.
- Die Knoten jedes Partitionsteils V_i $(1 \leq i \leq k)$ werden so durch einen Knoten N_i des Kreises repräsentiert, dass zwei Teile V_i und V_{i+1} durch zwei adjazente Knoten repräsentiert werden.
- Bemerkung: Für jeden Minimum Cut S, der kein Circular Partition Cut ist, ist entweder S oder $\bar{S}$ eine echte Teilmenge eines Teils V_i (folgt direkt aus der Definition).

Kaktus-Repräsentation aller Minimum Cuts

(Fortsetzung 2. Fall: Graph mit genau einer zirkulären Partition)

- Man kann den Baum $T_{(V_i,E)}$ für alle Minimum Cuts konstruieren, die Teilmenge von V_i sind, aber mit der Beschränkung, dass nur die Knoten von V_i diesem Baum zugeordnet werden.
- Die Wurzel von $T_{(V_i,E)}$ entspricht genau der Menge V_i.

$\Rightarrow$ Knoten N_i des Kreises kann mit der Wurzel von $T_{(V_i,E)}$ verschmolzen werden ($\forall i : 1 \leq i \leq k$).

- Dieser mit allen Bäumen verbundene Kreis ist der Kaktus C_G für G.
- Anzahl Knoten: Summe der Anzahl der Knoten aller Bäume

$\Rightarrow$ wieder beschränkt durch $2|V| - 1$ und wieder Korrespondenz zwischen Minimum Cuts in G und Separation in C_G.

Kaktus-Repräsentation aller Minimum Cuts

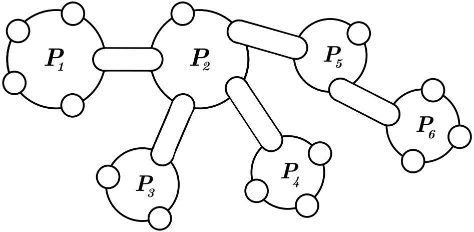
3. Fall: Graph mit zirkulären Partitionen $P_1, \ldots, P_z$

- Betrachte alle zirkulären Partitionen als Menge von Mengen
- Konstruiere den Kaktus, der die Circular Partition Cuts repräsentiert:
- Jede zirkuläre Partition wird durch einen Kreis repräsentiert.
- Die Knoten des Kreises entsprechen dabei den jeweiligen disjunkten Mengen.
- Berührungspunkte ergeben sich entsprechend der Definition der Kompatibilität zirkulärer Partitionen.
- Es entsteht eine baumartige Struktur aus verbundenen Kreisen.

Kaktus-Repräsentation aller Minimum Cuts

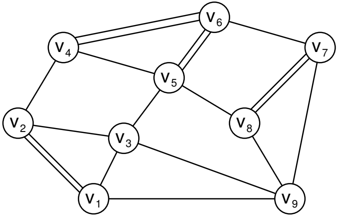
(Fortsetzung 3. Fall: Graph mit zirkulären Partitionen $P_1, \ldots, P_z$)

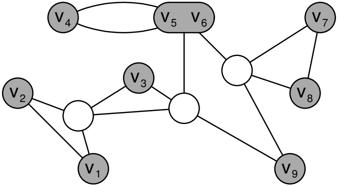
- Bsp.: Kaktus für die Circular Partition Cuts von 6 Circular Partitions



- Repräsentiere die Minimum Cuts, die nicht Teil einer zirkulären Partition sind (analog zum 2. Fall)
- Man erhält den Kaktus T_C von G.
- Die Anzahl Knoten ist wieder beschränkt durch $2|V| - 1$

Beispiel für Kaktus-Repräsentation





Berechnung eines globalen MinCuts

- 1994 Algorithmus publiziert von M. Stoer und F. Wagner
- benutzt keine MaxFlow-Technik
- sehr einfach
- arbeitet in $n-1$ Phasen
- Phase hat starke Ähnlichkeit zu den Algorithmen von Prim (minimale Spannbäume) bzw. Dijkstra (kürzeste Wege)
- Pro Phase Komplexität $\mathcal{O}(m + n \log n)$
- Gesamtkomplexität $\mathcal{O}(mn + n^2 \log n)$

Der Stoer/Wagner-Algorithmus

Algorithmus 13 : MinCut Berechnung nach Stoer & Wagner

Input : Ungerichteter Graph $G = (V, E)$
Output : Ein MinCut C_{min} entsprechend $\lambda(G)$

Wähle beliebigen Startknoten a;
$C_{\text{min}} \leftarrow$ undefiniert; $V' \leftarrow V$;
while $|V'| > 1$ **do**
  $A \leftarrow \{a\}$;
  while $A \neq V'$ **do**
    Füge zu A den am meisten angebundenen Knoten hinzu;
    Aktualisiere die Kapazitäten zwischen A und den Knoten in $V' \setminus A$;
  $C :=$ Schnitt von V', der den zuletzt zu A hinzugefügten Knoten vom Rest des Graphen trennt;
  if $C_{\text{min}} = \text{undefiniert}$ *or* $w(C) < w(C_{\text{min}})$ **then**
    $C_{\text{min}} \leftarrow C$;
  Verschmelze die zwei Knoten, die zuletzt zu A hinzugefügt wurden;
return C_{min};

Beschreibung des Stoer/Wagner-Algorithmus

- Wahl eines beliebigen Startknotens a
- Algorithmus verwaltet in jeder Phase eine Knotenteilmenge A
  - initialisiert mit a,
  - wird immer wieder um einen Knoten erweitert,
  - und zwar um einen, der gerade maximale Anbindung (Summe der Kantenkapazitäten) an die aktuellen Knoten in A hat.

$\Rightarrow$ Maximum Adjacency Ordering (MAO)

- Nach Einfügen aller Knoten in A:
  Cut, der nur den zuletzt hinzugefügten Knoten t vom Rest abtrennt, heißt 'Cut of the Phase'.

Beschreibung des Stoer/Wagner-Algorithmus

- Nach jeder Phase kommt der 'Cut of the Phase' als globaler MinCut in Frage.
- Verschmelzung der beiden zuletzt behandelten Knoten s, t:
  - Wenn zwischen s und t eine Kante existiert, wird sie einfach gelöscht.
  - Alte Kanten von einem anderen Knoten $\{x, s\}$ und $\{x, t\}$ werden durch eine neue Kante mit der Summe der alten Gewichte $w(s, x) + w(t, x)$ ersetzt.
- Dann: erneute MAO-Phase mit verschmolzenen Knoten ...
- 'Cut of the Phase' mit kleinster Kapazitätssumme ist globaler MinCut

Korrektheit des Stoer/Wagner-Algorithmus

Lemma

Der 'Cut of the Phase' ist ein Minimum (s, t)*-Cut für die beiden zuletzt in* A *eingefügten Knoten* s *und* t.

Korrektheit des Stoer/Wagner-Algorithmus

Beweis.

- Betrachte für die letzten zwei Knoten s und t einen beliebigen (also evt. auch Minimum) s-t-Cut C.
- Ein Knoten $v \neq a$ heißt aktiv, falls sich v und sein unmittelbarer Vorgänger bezüglich des Hinzufügens zu A auf entgegengesetzten Seiten von C befinden.
- Für einen Knoten v sei
  A_v: die Menge von Knoten, die in A sind bevor v hinzugefügt wird,
  C_v: der durch C in $A_v \cup \{v\}$ induzierte Cut.
- Sei $w(S, v)$ für eine Knoten(teil)menge S die Kapazitätssumme aller Kanten zwischen v und den Knoten in S.

Korrektheit des Stoer/Wagner-Algorithmus

Beweis.

- Beweisidee: (Induktion über die aktiven Knoten)

  Für jeden aktiven Knoten v gilt:
  Die Anbindung (Adjazenz) zu den Knoten, die zuvor eingefügt wurden (also A_v), überschreitet nicht das Gewicht des durch C in $A_v \cup \{v\}$ induzierten Cuts (also C_v).

- Also zu zeigen:

$$w(A_v, v) \leq w(C_v)$$

- Induktionsanfang: (1. aktiver Knoten)
  Im Basisfall ist die Ungleichung erfüllt, weil für den ersten aktiven Knoten die Werte auf beiden Seiten gleich sind.

Korrektheit des Stoer/Wagner-Algorithmus

Beweis.

- Induktionsvoraussetzung:
  Das Lemma stimmt für alle aktiven Knoten bis zum aktiven Knoten v.

⇒ Der Wert für den nächsten aktiven Knoten u ist

$$\begin{aligned} w(A_u, u) &= w(A_v, u) + w(A_u \setminus A_v, u) \\ &\leq w(A_v, v) + w(A_u \setminus A_v, u) \\ &\leq w(C_v) + w(A_u \setminus A_v, u) \qquad \text{(Ind.voraussetzung)} \\ &\leq w(C_u) \end{aligned}$$

- Die letzte Zeile folgt, weil alle Kanten zwischen $A_u \setminus A_v$ und u ihr Gewicht zu $w(C_u)$ beitragen, aber nicht zu $w(C_v)$.
- Da t durch den Cut C von seinem unmittelbaren Vorgänger s getrennt wird, ist t immer ein aktiver Knoten.
- Es folgt $w(A_t, t) \leq w(C_t)$ (mit $C_t = C$).

□