Fortgeschrittene Netzwerk- und Graph-Algorithmen

Dr. Hanjo Täubig

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

Wintersemester 2007/08

Übersicht

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 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)
Die Minimum Cuts können durch folgenden Baum T_G repräsentiert werden:

- 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 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 $P_1, \ldots, P_z$

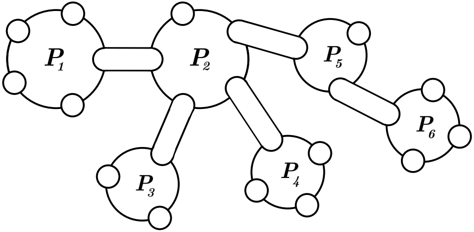
- Betrachte alle zirkulären Partitionen als Menge von Mengen
- Konstruiere den Kaktus, der die Circular Partition Cuts repräsentiert:
  - Die Knoten jeder Menge $F \in \mathcal{F}_{\boldsymbol{P_1} \cup \ldots \cup \boldsymbol{P_z}}$ werden einem Knoten zugeordnet.
  - Zwei Knoten sind verbunden, wenn für ihre Mengen F_1 und F_2 gilt: $w(F_1, F_2) > 0$.

  $\Rightarrow$ Jede zirkuläre Partition erzeugt einen Kreis in C_G.
  - Da alle zirkulären Partitionen paarweise kompatibel sind, sind die Kreise durch Kanten verbunden, die nicht Teil eines Kreises sind.
  - Der Kaktus C_G ist jetzt ein baumartiger Graph.

Kaktus-Repräsentation aller Minimum Cuts

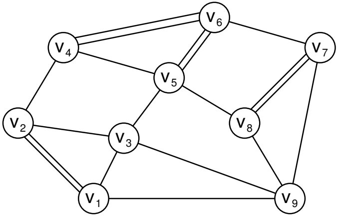
(Fortsetzung 3. Fall: Graph mit zirkulären Partitionen $P_1, \ldots, P_z$)

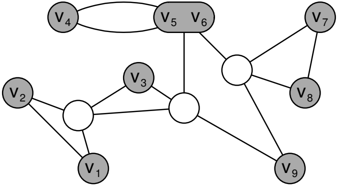
- Bsp.: Kaktus für die Circular Partition Cuts von 6 Circular Partitions



- 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





Cycle-type normal cactus representations

Definition

In einem Kaktus $\mathcal{R}$ nennen wir einen Kreis mit h Knoten einen h-Kreis. Einen Knoten, der zu genau k Kreisen gehört, nennt man k-Verzweigungsknoten.

Eine Kaktus-Repräsentation heißt *normal*, wenn sie keinen leeren 2-Verzweigungsknoten enthält, der zu einem 2-Kreis gehört.

Eine Kaktus-Repräsentation heißt *cycle-type*, wenn sie keine leere 3-Verzweigung enthält.

Eine *cycle-type normal cactus representation* nennt man auch kurz CNCR.

CNCRs

Lemma (Nagamochi/Kameda)

Angenommen es gibt für eine Teilmenge $\mathcal{C}' \subseteq \mathcal{C}(G)$ der MinCuts eines Graphen G eine Kaktus-Repräsentation $(\mathcal{R}, \varphi)$. Dann gilt:

- *Es gibt eine CNCR für $\mathcal{C}'$.*
- *Die CNCR für $\mathcal{C}'$ ist eindeutig.*
- *Jede CNCR für $\mathcal{C}'$ hat höchstens $|V(G)|$ leere Knoten.*
- *$(\mathcal{R}, \varphi)$ kann in eine CNCR für $\mathcal{C}'$ konvertiert werden in Zeit und Platz linear in der Größe von $(\mathcal{R}, \varphi)$.*

Eigenschaften von Kaktus-Repräsentationen

- Angenommen wir haben zwei Kaktus-Repräsentationen $(\mathcal{R}_1, \varphi_1)$ und $(\mathcal{R}_2, \varphi_2)$ für Teilmengen $\mathcal{C}_1, \mathcal{C}_2$ von $\mathcal{C}(G)$.
- Nagamochi/Kameda haben gezeigt, dass es bei Existenz von Knoten $z_1 \in V(\mathcal{R}_1)$ und $z_2 \in V(\mathcal{R}_2)$ mit
$$\varphi_1^{-1}(z_1) \cup \varphi_2^{-1}(z_2) = V(G)$$
eine Kaktus-Repräsentation $(\mathcal{R}, \varphi)$ für $\mathcal{C}_1 \cup \mathcal{C}_2$ gibt, wo man $\mathcal{R}$ durch Identifikation von Knoten z_1 mit Knoten z_2 (zum neuen Knoten z) erhält und die
Abbildung $\varphi: \ V(G) \to V(\mathcal{R}_1) \cup V(\mathcal{R}_2) \cup \{z\} \setminus \{z_1, z_2\}$ wie folgt definiert ist:
$$\begin{aligned}
\varphi^{-1}(z) &= \varphi_1^{-1}(z_1) \cap \varphi_2^{-1}(z_2) \\
\varphi^{-1}(x_1) &= \varphi_1^{-1}(x_1) \text{ für alle Knoten } x_1 \in V(\mathcal{R}_1) \setminus z_1 \\
\varphi^{-1}(x_2) &= \varphi_2^{-1}(x_2) \text{ für alle Knoten } x_2 \in V(\mathcal{R}_2) \setminus z_2
\end{aligned}$$

Eigenschaften von Kaktus-Repräsentationen

- Die so definierte Repräsentation bezeichnen wir mit $(\mathcal{R}_1, \varphi_1) \oplus (\mathcal{R}_2, \varphi_2) = (\mathcal{R}, \varphi)$ und die entsprechenden Knoten z_1, z_2 heißen verbundene Knoten.
- Der neue Knoten z ist immer ein Artikulationsknoten in $\mathcal{R}$. Er ist genau dann leer in $(\mathcal{R}, \varphi)$, wenn $\varphi_1^{-1}(z_1) \cap \varphi_2^{-1}(z_2) = \emptyset$ gilt.

Kritische Kanten und (s, t)-MC-Partition

Eine Kante $e = (s, t)$ heißt *kritisch*, falls $c_G(e) > 0$ und $\lambda_G(s, t) = \lambda_G$.

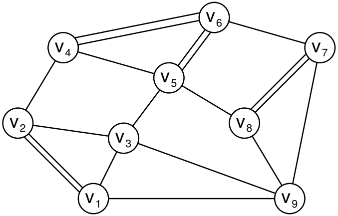
Lemma (Karzanov/Timofeev)

Für jede kritische Kante $e = (s, t)$ sind beliebige MinCuts, die s und t separieren, keine Crossing Cuts.
D.h. es gibt eine geordnete Partition $(V_1, \ldots, V_2)$ von $V(G)$, so dass die Menge der Cuts der Form $\{V_1 \cup V_2 \cup \ldots \cup V_i, V_{i+1} \cup \ldots \cup V_r\}$ gleich der Menge der MinCuts in $\mathcal{C}(G)$ ist, die s und t separieren.

Solch eine geordnete Partition nennt man (s, t) minimum cut o-partition oder kurz (s, t)-MC-Partition.

Beispiel für (s,t)-MC-Partition

Beispiel:



Die (s,t)-MC-Partition für $(s,t) = (v_1, v_9)$ ist $\{V_1 = \{v_1\}, V_2 = \{v_2\}, V_3 = \{v_3\}, V_4 = \{v_4, v_5, v_6\}, V_5 = \{v_7, v_8\}, V_6 = \{v_9\}$.

Berechnung der (s, t)-MC-Partition

Lemma

Sei (s, t) eine kritische Kante in einem Graphen.
Wenn ein beliebiger Maximum Flow zwischen s und t gegeben ist, kann die (s, t)-MC-Partition in Zeit und Platz $\mathcal{O}(m + n)$ berechnet werden.

(Beweis: Karzanov/Timofeev, Naor/Vazirani)

Mit Partition π kompatible und unteilbare Cuts

Sei π eine Partition $\{V_1, V_2, \dots, V_r\}$ oder eine geordnete Partition $(V_1, V_2, \dots, V_r)$ von $V(G)$.
Wir nennen einen Cut $\{X, \bar{X}\}$ kompatibel mit π, falls

$$X = \bigcup_{i \in I} V_i \text{ für ein } I \subset \{1, 2, \dots, r\}$$

Wir nennen einen Cut $\{X, \bar{X}\}$ unteilbar mit π, falls

$$X \subset V_i \text{ für ein } i \in \{1, 2, \dots, r\}$$

Ein Cut $\{X, \bar{X}\}$ kreuzt eine Partition $\{V_1, V_2, \dots, V_r\}$ oder eine geordnete Partition $(V_1, V_2, \dots, V_r)$, falls ein V_i existiert, so dass keine der Teilmengen $X \cap V_i$, $X \setminus V_i$ und $V_i \setminus X$ leer ist.
Jeder Cut, der π nicht kreuzt, ist entweder kompatibel oder unteilbar bezüglich π. Wir bezeichnen die entsprechenden Mengen von MinCuts mit $\mathcal{C}_{\text{comp}}(\pi)$ und $\mathcal{C}_{\text{indv}}(\pi)$

Mit Partition $\pi_{(s,t)}$ kompatible und unteilbare Cuts

Lemma

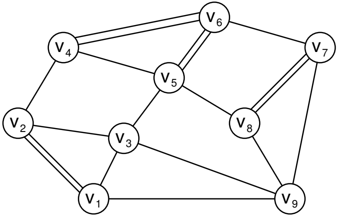
Sei (s,t) eine kritische Kante in einem Graph G und $\pi_{(s,t)}$ die (s,t)-MC-Partition über $\mathcal{C}(G)$. Dann ist jeder MinCut $\{X, \bar{X}\} \in \mathcal{C}(G)$ entweder kompatibel oder unteilbar bezüglich $\pi_{(s,t)}$, d.h. $\mathcal{C}(G) = \mathcal{C}_{comp}(\pi_{(s,t)}) \cup \mathcal{C}_{indv}(\pi_{(s,t)})$.

Man beachte, dass $\mathcal{C}_{\text{comp}}(\pi_{(s,t)})$ einen MinCut enthalten kann, der s und t nicht separiert.

Satz (Nagamochi/Kameda)

Sei (s,t) eine kritische Kante eines Graphen G und $\pi_{(s,t)}$ die (s,t)-MC-Partition.
Dann existiert eine Kaktus-Repräsentation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ für alle MinCuts in $\mathcal{C}_{comp}(\pi_{(s,t)})$, die man (s,t)-Kaktus-Repräsentation nennt.
Weiterhin kann für gegebenes $\pi_{(s,t)}$ die cycle-type normal (s,t)-cactus representation ((s,t)-CNCR) in $\mathcal{O}(m+n)$ Zeit und

Mit Partition $\pi_{(s,t)}$ kompatible und unteilbare Cuts



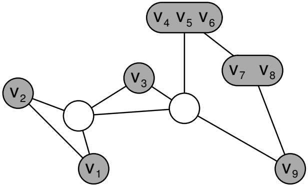


Abbildung: Cycle-type normal (s,t)-cactus representation mit $(s,t) = (v_1, v_9)$ über $\mathcal{C}(G)$

Berechnungsgrundlage

Grundlage für den NNI-Algorithmus:

Satz

In einem Graphen G kann man eine Kante $E = (s, t)$ mit $c_G(e) > 0$, die die folgenden zwei Bedingungen erfüllt, in Zeit $\mathcal{O}(m + n \log n)$ und Platz $\mathcal{O}(m + n)$ berechnen:

1. *$\lambda_G(s, t)$ kann in Zeit $\mathcal{O}(m + n \log n)$ und Platz $\mathcal{O}(m + n)$ berechnet werden.*
2. *Wenn $\lambda_G(s, t) = \lambda_G$, dann kann die (s, t)-CNCR in Zeit $\mathcal{O}(m + n \log n)$ und Platz $\mathcal{O}(m + n)$ berechnet werden.*

Maximum Adjacency Ordering

Definition

Eine totale Ordnung $v_1, v_2, \ldots, v_n$ aller Knoten in $V(G)$ bezeichnet man als Maximum Adjacency Ordering (MAO), falls für alle i, j mit $1 \leq i \leq j \leq n-1$ gilt:

$$w(\{v_1, v_2, \ldots, v_{i-1}\}, v_i) \geq w(\{v_1, v_2, \ldots, v_{i-1}\}, v_j)$$

Verbal: Die Anbindung des Knotens v_i an seine Vorgänger $v_1, v_2, \ldots, v_{i-1}$ ist mindestens so groß wie die Anbindung jedes einzelnen nachfolgenden Knotens $v_{i+1}, \ldots, v_n$ an die Vorgänger von v_i.

Maximum Adjacency Ordering

Lemma

Ein MAO eines Graphen G kann in Zeit $\mathcal{O}(m + n \log n)$ und Platz $\mathcal{O}(m + n)$ berechnet werden.
Für ein MAO $v_1, v_2, \ldots, v_n$ in G gilt für die letzten beiden Knoten v_{n-1}, v_n, dass $\lambda_G(v_{n-1}, v_n) = w(\{v_n\}, V_G \setminus \{v_n\})$.

Beweis des letzten Satzes

Beweis.

- Berechne zuerst ein MAO $v_1, v_2, \ldots, v_n$ von G.
- Wähle den Knoten v_p mit höchstem Index p, so dass v_p und v_n durch eine Kante (mit pos. Gewicht) verbunden sind.
- Sei $s = v_n$ und $t = v_p$.
- Man beachte, dass $v_1, v_2, \ldots, v_p, v_n$ ein MAO in dem Graphen G' ist, der aus G entsteht, wenn man die Knoten $v_{p+1}, \ldots, v_{n-1}$ löscht.
- D.h. nach dem letzten Lemma gilt: $\lambda_G(s,t) \geq \lambda_{G'}(s,t) = w_{G'}(\{s\}, V(G') \setminus \{s\}) = w_G(\{s\}, V(G) \setminus \{s\})$
- Da $\lambda_G(s,t) \leq w_G(\{s\}, V(G) \setminus \{s\})$, folgt der 1. Teil des Satzes.

Beweis des letzten Satzes

Beweis.

- Annahme: $\lambda_G(s,t) = \lambda_G$ (wie im 2. Teil des Satzes)
- Ein Maximum Flow zwischen den letzten beiden Knoten eines MAO kann in $\mathcal{O}(m)$ Zeit und Platz berechnet werden (Arikati/Mehlhorn).

$\Rightarrow$ MaxFlow zwischen s und t kann in $\mathcal{O}(m)$ gefunden werden.

- Für kritische Kante (s,t) mit gegebenem MaxFlow zwischen s und t kann nach Lemma die (s,t)-MC-Partition in $\mathcal{O}(m+n)$ Zeit und Platz bestimmt werden.
- Nach vorhergehendem Satz existiert für die kritische Kante (s,t) und die (s,t)-MC-Partition $\pi_{(s,t)}$ eine Kaktus-Repräsentation für alle MinCuts in $\mathcal{C}_{\text{comp}}(\pi_{(s,t)})$, genannt (s,t)-Kaktus-Repräsentation, deren CNCR in $\mathcal{O}(m+n)$ Zeit und Platz konstruiert werden kann.
- Die cycle-type normal (s,t)-cactus representation kann in Zeit

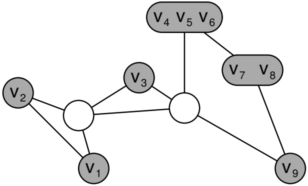
Berechnung der Kaktus-Repräsentation

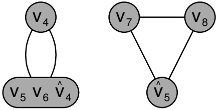
- Sei im folgenden G^* der Eingabe-Graph, für den eine Kaktus-Repräsentation berechnet werden soll.
- Die Konstruktion der Kaktus-Repräsentation von $\mathcal{C}(G^*)$ erfolgt rekursiv auf der Basis des letzten Satzes.
- G/X stellt den Graph dar, den man aus G erhält, wenn man alle Knoten in X zu einem einzigen Knoten kontrahiert.
- Sei $\lambda = \lambda(G^*)$.
- Wir wählen eine Kante (s,t) entsprechend dem letzten Satz.
- Falls $\lambda_G(s,t) > \lambda$, dann kontrahiere Knoten $\{s,t\}$ (kein MinCut in G separiert s und t).
- Falls $\lambda_G(s,t) = \lambda$, dann bestimme (s,t)-MC-Partition $\pi_{(s,t)} = (V_1, \ldots, V_r)$ und (s,t)-Kaktus-Repräsentation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ in G.
- Nach Lemma sind alle mit $\pi_{(s,t)}$ kompatiblen MinCuts durch $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ repräsentiert und jeder mit $\pi_{(s,t)}$ unteilbare MinCut $\{X, V(G) \setminus X\}$ erfüllt $X \subset V_i$ für ein $V_i \in \pi_{(s,t)}$

Berechnung der Kaktus-Repräsentation

- Sei $G_i = G/(V(G) \setminus V_i)$ der Graph, bei dem die Knoten $V(G) \setminus V_i$ $(i = 1, \ldots, r)$ zu einem Knoten $\hat{v}_i$ kontrahiert wurden, wobei gilt: $\lambda_{G_i} \geq \lambda_G$.
- Annahme: für jedes G_i wurde eine Kaktus-Repräsentation $(\mathcal{R}_i, \varphi_i)$ rekursiv berechnet, wobei $(\mathcal{R}_i, \varphi_i)$ der triviale Kaktus sein soll falls $\lambda_{G_i} > \lambda_G$.
- Dann sind alle MinCuts repräsentiert in $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$, $(\mathcal{R}_1, \varphi_1)$,...,$(\mathcal{R}_r, \varphi_r)$.
- Weiterhin können diese Repräsentationen zu einer einzigen vereinigt werden, indem die Knoten, die $\hat{v}_i$ enthalten, als verbundene Knoten betrachtet werden.

Berechnung der Kaktus-Repräsentation





Kaktus-Repräsentationen $(\mathcal{R}_i, \varphi_i)$, wobei sich in den Fällen $i = \{1, 2, 3, 6\}$ der triviale Kaktus ergibt.
Auf dem Bild sieht man $(\mathcal{R}_4, \varphi_4)$ und $(\mathcal{R}_5, \varphi_5)$.

Berechnung der Kaktus-Repräsentation

- Wenn CACTUS(G', V^{old}) für einen Graph G' während der Ausführung von CACTUS(G'', V^{old}) für einen Graph G'' aufgerufen wird, nennen wir G' ein Kind von G'' und G'' den Vater von G'.
- Die Vater-Kind-Beziehung induziert einen Baum $\mathcal{T}$, der am Eingabe-Graph G^* gewurzelt ist (Aufrufbaum).
- Bemerkung: Falls $w(V_i, V(G) \setminus V_i) = \lambda$, dann bleibt der Cut $\{V_i, \hat{v}_i\}$ ein MinCut in einem Kind G_i, obwohl der Cut schon in seinem Vater G erkannt wurde.
- Derselbe MinCut kann in einem Nachfolger von G_i sein.
- Ein MinCut ist alt in einem Graph G' (also wurde schon in einem Vorfahren entdeckt), nur wenn ein einzelner Knoten v von $V(G') \setminus \{v\}$ separiert wird.
- D.h. wir können testen, ob die (s,t)-Kaktus-Repräsentation neue MinCuts enthält, indem wir die Knoten $v \in V(G')$ als 'alt' markieren, falls v ein kontrahierter Knoten $\hat{v}_i$ im Vorgänger ist.

NNI-Alg. für Kaktus-Repräsentation

Algorithmus 7 : CACTUS(G, V^{old})

Input : Graph G, Teilmenge $V^{\text{old}} \subset V(G)$
Output : Kaktus-Repr. $(\mathcal{R}, \varphi)$ für eine Menge $\mathcal{C}'$ von MinCuts, so dass
$\mathcal{C}(G) \setminus \{\{\bar{v}, V(G) \setminus \{\bar{v}\}\} | \bar{v} \in V^{\text{old}}\} \subseteq \mathcal{C}' \subseteq \mathcal{C}(G)$

if $|V(G)| = 1$ **then return** *Trivial-Kaktus* $(\mathcal{R}, \varphi)$;
else
- Wähle Kante $e = (s,t) \in E(G)$ mit $w_G(e) > 0$;
- **if** $\lambda_G(s,t) > \lambda$ *oder* (s,t)*-Kaktus Repr.* $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ *repräsentiert keinen anderen Cut außer die folgenden:* $\{\bar{v}, V(G) \setminus \{\bar{v}\}\}, \bar{v} \in V^{old}$ **then**
  - $G = G/\{s,t\}$; $V^{\text{old}} = V^{\text{old}} \setminus \{s,t\}$;
  - **return** *CACTUS*(G, V^{old})
- **else**
  - **foreach** V_i *in der* (s,t)*-MC-Partition* $\pi_{(s,t)} = (V_1, \dots, V_r)$ **do**
    - $G_i = G/(V(G) \setminus V_i)$, wobei $\bar{v}_i$ den Knoten bezeichnet, der durch die Kontraktion von $V(G) \setminus V_i$ entsteht;
    - $V_i^{\text{old}} = (V^{\text{old}} \cap V_i) \cup \{\bar{v}_i\}$;
    - $(\mathcal{R}_i, \varphi_i) =$CACTUS$(G_i, V_i^{\text{old}})$
  - $(\mathcal{R}, \varphi) = (\mathcal{R}_{(s,t)}, \varphi_{(s,t)}) \oplus (\mathcal{R}_1, \varphi_1) \oplus \dots \oplus (\mathcal{R}_r, \varphi_r)$;
  - **return** $(\mathcal{R}, \varphi)$ *nach Konvertierung in CNCR*

NNI-Alg. für Kaktus-Repräsentation

Algorithmus 8 : CONSTRUCT

Input : Graph G^*

Output : CNCR $(\mathcal{R}, \varphi)$ für $\mathcal{C}(G)$

Compute $\lambda = \lambda(G)$;
$V^{\text{old}} = \emptyset$;
$(\mathcal{R}, \varphi) =$CACTUS(G^*, V^{old})

Satz

Der vorgestellte Algorithmus berechnet eine Kaktus-Repräsentation für alle Minimum Cuts in Zeit $\mathcal{O}(mn + n^2 \log n)$ und Platz $\mathcal{O}(m + n)$.