Fortgeschrittene Netzwerk- und Graph-Algorithmen

Dr. Hanjo Täubig

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

Wintersemester 2009/10

Übersicht

Kantenzusammenhangsalgorithmen

- Der Kantenzusammenhang λ kann in ungerichteten ungewichteten Graphen ebenfalls mit der MaxFlow-Prozedur gelöst werden.
- Ersetze dafür jede ungerichtete Kante durch zwei antiparallele gerichtete Kanten mit Kapazität 1 und berechne dann den lokalen Zusammenhang zwischen der entsprechenden Quelle s und der Senke t.
- Da das resultierende Netzwerk ein Unit Capacity Network vom Typ 1 ist, ist die Komplexität $\mathcal{O}(m \cdot \min\{m^{1/2}, n^{2/3}\})$.

Einfache Kantenzusammenhangsalgorithmen

- Trivialer Algorithmus: mit $n(n-1)/2$ MaxFlow-Aufrufen den lokalen Kantenzusammenhang aller Knotenpaare berechnen

- Besser: einen Knoten s festzuhalten und dann für alle anderen Knoten t die lokalen Zusammenhangszahlen $\lambda(s,t)$ berechnen

  Mindestens einer dieser Knoten muss auf der anderen Seite eines MinCuts liegen. Deshalb ist das Minimum aller dieser $(n-1)$ lokalen Zusammenhangszahlen gleich dem (globalen) Kantenzusammenhang λ des Graphen.

  Gesamtkomplexität: $\mathcal{O}(nm \cdot \min\{n^{2/3}, m^{1/2}\})$

λ-Covering

- Der Algorithmus funktioniert auch, wenn man statt der ganzen Knotenmenge nur eine Teilmenge verwendet, die zwei Knoten s, t enthält, deren lokaler Zusammenhang $\lambda(s, t)$ gleich dem globalen Zusammenhang λ ist.
- Eine solche Teilmenge heißt λ-Covering.

$\Rightarrow$ Versuche, die Kardinalität dieser Knotenmenge zu reduzieren.

Komponentengröße

Lemma

Sei S ein Minimum Edge-Cut eines Graphen $G = (V, E)$ und sei $L, R \subset V$ eine Partition der Knotenmenge, so dass L and R durch S separiert werden.
Wenn $\lambda(G) < \delta(G)$, dann besteht jede Komponente von $G - S$ aus mehr als $\delta(G)$ Knoten, d.h. es gilt $|L| > \delta(G)$ und $|R| > \delta(G)$.

Komponentengröße

Beweis.

- Seien $l_1, \ldots, l_k$ die Elemente von L und sei $E[L] = E(G[L])$ die Menge der durch L induzierten Kanten in G.
- Es gilt:

$$\begin{aligned} \delta_G \cdot k &\leq \sum_{i=1}^{k} d_G(l_i) \\ &\leq 2 \cdot |E[L]| + |S| \\ &\leq 2 \cdot \frac{k(k-1)}{2} + |S| \\ &< k(k-1) + \delta_G \end{aligned}$$

- Aus $\delta_G \cdot (k-1) < k(k-1)$ folgt $|L| = k > 1$ und $|L| = k > \delta_G$ (sowie $|R| > \delta(G)$).

$\square$

Komponentengröße

Folgerung

Wenn gilt $\lambda_G < \delta_G$, dann enthält jede Komponente von $G - S$ einen Knoten, der mit keiner Kante in S inzident ist.

Kantenzusammenhang λ

Lemma

Sei $\lambda_G < \delta_G$ und sei T ein Spannbaum von G.
Dann enthalten alle Komponenten von $G - S$ mindestens einen Knoten, der kein Blatt in T ist, d.h. die inneren Knoten von T bilden ein λ-Cover.

Beweis.

- Nehmen wir an, dass es nicht so wäre
  (d.h. alle Knoten in L sind Blätter von T).
- Also für keine Kante von T sind beide Endknoten in L, d.h. $|L| = |S|$.
- Aus dem vorhergehenden Lemma ($\lambda_G < \delta_G \Rightarrow |L| > \delta_G$ und $|R| > \delta_G$) folgt, dass $\lambda_G = |S| = |L| > \delta_G$
  (Widerspruch zur Annahme).

$\square$

Spannbaum-Berechnung (Esfahanian & Hakimi)

Algorithmus von Esfahanian & Hakimi:

- Berechne Spannbaum des gegebenen Graphen.
- Wähle beliebigen inneren Knoten v des Baums.
- Berechne für jeden anderen Knoten w, der kein Blatt ist, den lokalen Kantenzusammenhang $\lambda(v, w)$.
- Das Minimum dieser Wertemenge, zusammen mit δ_G, ergibt genau den Kantenzusammenhang λ_G.
- Ein Baum mit möglichst vielen Blättern wäre vorteilhaft, aber die Konstruktion eines optimalen Baums ist $\mathcal{NP}$-hart.

Spannbaum-Berechnung (Esfahanian & Hakimi)

Esfahanian & Hakimi:

- Algorithmus zur Berechnung eines Spannbaums T von G, so dass beide Mengen, L und R eines minimalen Kantenseparators mindestens ein Blatt von T enthalten, und nach dem letzten Lemma mindestens einen inneren Knoten.

Spannbaum-Berechnung (Esfahanian & Hakimi)

Algorithmus 12 : Spannbaum-Berechnung (Esfahanian & Hakimi)

Input : Ungerichteter Graph $G = (V, E)$
Output : Spannbaum T mit einem Blatt und einem inneren Knoten in L bzw. R

Wähle $v \in V$;
$T \leftarrow$ alle Kanten inzident mit v;
while $|E(T)| < n - 1$ **do**
  Wähle ein Blatt w in T, so dass für alle Blätter r in T gilt: $|N(w) \cap (V - V(T))| \geq |N(r) \cap (V - V(T))|$;
  $T \leftarrow T \cup \{(w, x) \in E : x \in (V - V(T))\}$
return T;

Kantenzusammenhang λ (Esfahanian & Hakimi)

Algorithmus 13 : Kantenzusammenhang λ (Esfahanian & Hakimi)

Input : Ungerichteter Graph $G = (V, E)$
Output : $\lambda(G)$

Konstruiere einen Spannbaum T (voriger Algorithmus);
Sei P die kleinere der beiden Mengen
(entweder die Blätter oder die inneren Knoten von T);
Wähle einen Knoten $u \in P$;
$c \leftarrow \min\{\lambda_G(u, v) : v \in P \setminus \{u\}\}$;
$\lambda \leftarrow \min(\delta(G), c)$;
return λ;

Kantenzusammenhang λ (Esfahanian & Hakimi)

- Da P als kleinere der beiden Mengen (Blätter / innere Knoten) gewählt wird, benötigt der Algorithmus höchstens $n/2$ Berechnungen für lokale Zusammenhangszahlen.

$\Rightarrow$ Gesamtzeitkomplexität: $\mathcal{O}(\lambda mn)$

Kantenzusammenhang λ (Matula)

Definition

Ein Dominating Set für einen Graphen $G = (V, E)$ ist eine Knotenteilmenge V' von V, so dass jeder Knoten, der nicht in V' ist, mit mindestens einem Knoten in V' über eine Kante verbunden ist.

Lemma

Im Fall $\lambda(G) < \delta(G)$ ist jedes Dominating Set von G auch ein λ-covering von G.

Kantenzusammenhang λ (Matula)

Verbesserter Algorithmus von Matula:

- Analog zur Spannbaum-Methode, wird λ hier berechnet, indem man
  - ein Dominating Set D von G berechnet,
  - einen beliebigen Knoten $u \in D$ auswählt und
  - den lokalen Kantenzusammenhang $\lambda(u, v)$ für alle anderen Knoten $v \in D \setminus \{u\}$ berechnet.
- Das Minimum der Wertemenge (wieder zusammen mit δ_G) ist der Kantenzusammenhang.
- Obwohl die Berechnung eines Dominating Sets minimaler Kardinalität $\mathcal{NP}$-hart ist, kann man zeigen, dass der Algorithmus in Zeit $\mathcal{O}(mn)$ läuft, wenn man das Dominating Set wie im folgenden Algorithmus konstruiert.

Kantenzusammenhang λ (Matula)

Algorithmus 14 : Berechnung eines Dominating Sets (Matula)

Input : Ungerichteter Graph $G = (V, E)$
Output : Dominating Set D

Wähle $v \in V$;
$D \leftarrow \{v\}$;
while $V \setminus (D \cup N(D)) \neq \emptyset$ **do**
    Wähle einen Knoten $w \in V \setminus (D \cup N(D))$;
    $D \leftarrow D \cup \{w\}$;
return D;

k-Kanten-Zusammenhangskomponenten

David W. Matula: k-Components, Clusters, and Slicings in Graphs

- gegeben: ungewichteter, ungerichteter Graph $G = (V, E)$
- Wdh.: Eine k-Kanten-(Zusammenhangs-)Komponente von G ist ein maximaler k-kanten-zusammenhängender Teilgraph von G.
- Da wir in diesem Abschnitt nur über *Kanten*zusammenhangskomponenten sprechen, werden wir diese einfach als k-Komponenten bezeichnen (auch wenn dieser Begriff sich sonst auf *Knoten*zusammenhang bezieht)
- Teilgraphen bestehend aus einem einzelnen isolierten Knoten ($k = 0$) nennen wir *triviale* Komponenten, aber diese werden hier nicht als k-Komponenten angesehen.

Vereinigung von Teilgraphen

Lemma

Seien $G_1, G_2, \ldots, G_n$ Teilgraphen von G, so dass $\bigcup_{i=1}^{n} G_i$ zusammenhängend ist, dann gilt:

$$\lambda\left(\bigcup_{i=1}^{n} G_i\right) \geq \min_{1 \leq i \leq n} \{\lambda(G_i)\}$$

Vereinigung von Teilgraphen

Beweis.

- Wenn $\bigcup_{i=1}^{n} G_i$ aus einem einzigen Knoten besteht, gilt die Behauptung (denn beide Seiten sind Null).
- Sei anderenfalls $C = (A, \bar{A})$ ein MinCut von $\bigcup_{i=1}^{n} G_i$.
- Dieser MinCut muss mindestens eine Kante enthalten, da $\bigcup_{i=1}^{n} G_i$ zusammenhängend und nicht trivial ist.
- Falls C eine Kante eines Teilgraphen G_j enthält, enthält sowohl A als auch $\bar{A}$ jeweils mindestens einen Knoten aus $V(G_j)$, d.h. C muss einen Cut für G_j enthalten.

$\Rightarrow$ Für mindestens ein $j \in \{1, \ldots, n\}$ gilt: $\lambda\left(\bigcup_{i=1}^{n} G_i\right) \geq \lambda(G_j)$

$\square$

Vereinigung von Teilgraphen

- Für die Teilgraphen $G_1, G_2, \ldots, G_n$ von G muss jeder Cut von $G\left[\bigcup_{i=1}^n V(G_i)\right]$ einen Cut von $\bigcup_{i=1}^n G_i$ enthalten. Also gilt:

$$\lambda\left(G\left[\bigcup_{i=1}^{n} V(G_i)\right]\right) \geq \lambda\left(\bigcup_{i=1}^{n} G_i\right)$$

Folgerung

Wenn $G_1, G_2, \ldots, G_n$ Teilgraphen von G sind, so dass $\bigcup_{i=1}^n G_i$ zusammenhängend ist, dann gilt:

$$\lambda\left(G\left[\bigcup_{i=1}^{n} V(G_i)\right]\right) \geq \min_{1\leq i\leq n}\{\lambda(G_i)\}$$

Vereinigung von Teilgraphen

- Man beachte, dass in der Folgerung die Bedingung, dass $\bigcup_{i=1}^{n} G_i$ zusammenhängend ist, nicht durch die abgeschwächte Forderung, dass $G\left[\bigcup_{i=1}^{n} V(G_i)\right]$ zusammenhängend ist, ersetzt werden kann.

  Bsp.: wenn G_1 und G_2 zwei disjunkte Kreise sind, die in G durch eine einzelne Kante verbunden sind, ist die Ungleichung in der Folgerung nicht erfüllt.
- Eine weitere Konsequenz des Lemmas ist die Tatsache, dass die k-Kanten-Komponenten jedes Graphen disjunkt sind. (Beweis siehe Abschnitt 'Fundamentale Sätze' am Anfang des Kapitels)

Die Kohäsion / Zusammenhangsfunktion

Definition

Für jedes Element (Knoten oder Kante) $x \in V(G) \cup E(G)$ eines Graphen G ist die Kohäsion (cohesiveness) bzw. die Zusammenhangsfunktion $h(x)$ definiert als maximaler Wert des Kantenzusammenhangs von allen Teilgraphen von G, die x enthalten.

Das Maximum $\sigma(G)$ aller Kohäsionswerte des Graphen G wird als Stärke (strength) des Graphen bezeichnet, also $\sigma(G) = \max\{\lambda(G') : G' \text{ ist Teilgraph von } G\}$.

Die Kohäsionsmatrix

- Die Zusammenhangsfunktion kann durch die (symmetrische) Kohäsionsmatrix dargestellt werden.
- Zeilen und Spalten werden durch die Knoten des Graphen indiziert, und der Eintrag für Position v_i, v_j ist jeweils die Kohäsion der Kante $\{v_i, v_j\}$, falls sie existiert, und sonst Null.
- Die Kohäsion eines Knotens ist dann das Maximum der entsprechenden Zeile oder Spalte.
- Die Stärke von G ist das Maximum aller Matrixeinträge.

Die Kohäsionsmatrix

- Für $v \in V(G)$ gilt: $0 \leq h(v) \leq \deg(v)$.
- Falls $\{v_i, v_j\} \in E(G)$, gilt:
  $1 \leq h(\{v_i, v_j\}) \leq \min\{\deg(v_i), \deg(v_j)\}$
- $h(x) = 0$ gilt also nur, wenn x ein isolierter Knoten ist.
- $\sigma(G) = 0$ gilt nur, wenn G keine Kante enthält.
- $\sigma(G) = 1$ gilt genau dann, wenn G ein Wald mit mindestens einer Kante ist, denn jeder Kreis würde $\sigma(G) \geq 2$ implizieren.

Eindeutige Komponentenzuordnung

- Für $x \in V(G) \cup E(G)$ und $h(x) \geq 1$ muss es für jedes $k \in \{1, \ldots, h(x)\}$ einen k-kanten-zusammenhängenden Teilgraph geben, der x enthält.
- Insbesondere muss es auch einen maximalen solchen Teilgraph (also eine k-Kanten-Komponente) geben.
- Da die k-Kanten-Komponenten sich nicht überschneiden ist dieser maximale Teilgraph (die Komponente) eindeutig.

Folgerung

Für jeden Graphen G, jedes Element $x \in V(G) \cup E(G)$ und jede Zusammenhangszahl $k \in \{1, \ldots, h(x)\}$ existiert eine eindeutige k-Kanten-Zusammenhangskomponente in G, die x enthält.

Eindeutige Komponentenzuordnung

- Für jedes Element x mit $h(x) \geq 1$ gilt:
  Unter allen Teilgraphen, die x enthalten und die maximalen Kantenzusammenhang (also $h(x)$) haben, hat die eindeutige $h(x)$-Komponente, die x enthält, die meisten Knoten.
  Sie heißt $h(x)$-Komponente selektiert durch x (Symbol: H_x).
- Die Kohäsion eines Elements kann aus dem Wissen über einen beliebigen Teilgraph maximalen Kantenzusammenhangs, der das Element enthält, abgeleitet werden.
- Aus der Kenntnis der k-Kanten-Komponenten von G für alle k kann man $h(x)$ für jedes Element x bestimmen.
- Aber man kann umgekehrt auch mit Hilfe der Zusammenhangsfunktion die Komponente H_x bestimmen.

Komponentenbestimmung

Satz

Sei x ein Element des Graphen G mit $h(x) \geq 1$.
Sei M_x ein maximaler zusammenhängender Teilgraph von G, der x enthält und dessen Elemente alle Kohäsion mindestens $h(x)$ haben.
Dann gilt $M_x = H_x$.

Beweis.

- Für $x \in V(G) \cup E(G)$ mit $h(x) \geq 1$ sei M_x definiert wie in dem Satz.
- Dann haben für jedes $y \in V(M_x) \cup E(M_x)$ alle Elemente von H_y Kohäsion mindestens $h(y) \geq h(x)$, so dass also gilt $H_y \cup M_x = M_x$.

Komponentenbestimmung

Beweis.

- Da jedes Element von M_x in einem H_y ist, gilt:

$$M_x = \bigcup \{H_y : y \in V(M_x) \cup E(M_x)\}$$

- Nach dem Lemma gilt daher

$$\lambda(M_x) \geq h(x)$$

- Da M_x selbst ein Teilgraph von G ist, der x enthält, gilt

$$\lambda(M_x) = h(x)$$

- Damit ist M_x ein $h(x)$-kanten-zusammenhängender Teilgraph von G und muss in der $h(x)$-Komponente selektiert durch x enthalten sein, also in H_x.

Komponentenbestimmung

Beweis.

- Da wegen $M_x = \bigcup\{H_y : y \in V(M_x) \cup E(M_x)\}$ der Teilgraph H_x in M_x enthalten sein muss, gilt $M_x = H_x$.
- Der im Satz definierte Teilgraph M_x ist damit eindeutig und kann bestimmt werden, indem man ausgehend von x alle Elemente anhängt, die von x über einen Pfad erreichbar sind, dessen Elemente alle Kohäsion mindestens $h(x)$ aufweisen.

$\square$

Folgerung

Für jeden Graph G und eine beliebige Zahl $k \in \{1, \ldots, \sigma(G)\}$ bilden die Knoten und Kanten von G mit Kohäsion mindestens k einen Graph, dessen Komponenten die k-Komponenten von G sind.

Komponentenbestimmung

- Für jeden Graph G sind die $h(x)$-Komponenten selektiert durch $x \in V(G) \cup E(G)$ von besonderem Interesse. Es wird nun gezeigt, dass in dieser Menge alle k-Kanten-Komponenten von G (für alle $k \in \{1, \ldots, \sigma(G)\}$) enthalten sind.

Folgerung

Wenn G' eine k-Komponente (für ein $k \geq 1$) des Graphen G ist, dann gibt es ein $x \in V(G) \cup E(G)$, so dass $G' = H_x$ ist.

Komponentenbestimmung

Beweis.

- Sei G' eine k-Komponente von G.
- Dann gilt $1 \leq k \leq \lambda(G')$ und G' ist damit auch eine $\lambda(G')$-Komponente von G.
- Wähle $x \in V(G') \cup E(G')$ so, dass $h(x)$ minimal ist und sei M_x definiert wie im Satz. (Man beachte, dass G' in M_x als Teilgraph enthalten sein muss.)
- $M_x = H_x$ ist eine $h(x)$-Komponente und deshalb k-kanten-zusammenhängend (da $k \leq \lambda(G') \leq h(x)$).
- Da G' ein maximaler k-kanten-zusammenhängender Teilgraph ist, gilt $G' = H_x$.

□