

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

- 1 Zusammenhang
 - k -Kanten-Zusammenhangskomponenten

Slicings

Definition

Die geordnete Partition der Kanten des Graphen G , $Z = (C_1, C_2, \dots, C_m)$, ist ein **Slicing** von G falls für jedes Element C_i gilt:

$$C_i \text{ ist ein Cut } (A_i, \bar{A}_i) \text{ von } \begin{cases} G & \text{für } i = 1 \\ G - \bigcup_{j=1}^{i-1} C_j & \text{für } i \in \{2, \dots, m\} \end{cases}$$

Ein Element C_i des Slicings heißt auch *Cut des Slicings*.

Ein Slicing $Z = (C_1, C_2, \dots, C_m)$ von G , für das es keine echte Unterpartition gibt, die ein Slicing von G ist, ist ein *minimales Slicing* von G . (Jeder Cut C_i eines minimalen Slicings muss ein minimaler Cut einer Komponente von $G - \bigcup_{j=1}^{i-1} C_j$ sein.)

Slicings

Definition

Z ist ein *Narrow Slicing* von G , falls jeder Cut C_i ein MinCut einer Komponente von $G - \bigcup_{j=1}^{i-1} C_j$ ist.

Dynamische Interpretation:

- Sequenz von nicht-leeren Cuts, die G in isolierte Knoten teilen
- Bei Minimalen/Narrow Slicings werden nur Minimale/Minimum Cuts in jedem Schritt verwendet.
- Jedes Narrow Slicing ist auch ein Minimales Slicing. (Das gilt umgekehrt nicht.)

Slicings

- Es ist nützlich, ein Slicing als sukzessive Zerlegung der Knotenmenge zu betrachten.
- Der i -te Cut $C_i = (A_i, \bar{A}_i)_i$ des Slicings Z von G ist ein Cut von $G - \bigcup_{j=1}^{i-1} C_j$.
- Da $A_i \cup \bar{A}_i = V(G - \bigcup_{j=1}^{i-1} C_j) = V(G)$, betrifft $(A_i, \bar{A}_i)_i$ die gleiche Partition der Knoten von G wie der Cut (A_i, \bar{A}_i) von G .
- Es gilt

$$(A_i, \bar{A}_i)_i = (A_i, \bar{A}_i) - \bigcup_{j=1}^{i-1} C_j$$

- Also enthält $(A_i, \bar{A}_i)_i$ nicht unbedingt alle Kanten des Cuts (A_i, \bar{A}_i) von G für $i > 1$.
- Man beachte, dass die Kantenmenge (A_i, \bar{A}_i) implizit von G abhängt und dass die die Kantenmenge $(A_i, \bar{A}_i)_i$ implizit von G und Z abhängt. Die Knotenpartition ist jedoch explizit und in beiden Fällen gleich!

Knotenpartitionen

- Die Repräsentation der Cuts des Slicings Z als Knotenpartition von $V(G)$ bietet eine nützliche Charakterisierung von jedem Graphen $G - \bigcup_{j=1}^{i-1} C_j$ im Sinne einer Vereinigung von induzierten Teilgraphen von G .
- Da C_1 die Knotenmenge A_1 von \bar{A}_1 separiert, und damit $G - C_1 = G[A_1] \cup G[\bar{A}_1]$, gilt somit:

$$G - (C_1 \cup C_2) = G[A_1 \cap A_2] \cup G[A_1 \cap \bar{A}_2] \cup G[\bar{A}_1 \cap A_2] \cup G[\bar{A}_1 \cap \bar{A}_2]$$

Satz

Induktiv folgt: Sei $Z = (C_1, C_2, \dots, C_m)$ ein Slicing von G mit $C_i = (A_i, \bar{A}_i)$ für $i \in \{1, \dots, m\}$. Dann gilt für $j \in \{1, \dots, m\}$:

$$G - \bigcup_{k=1}^j C_k = \bigcup \left\{ G \left[\bigcap_{i \in s} A_i \cap \bigcap_{i \notin s} \bar{A}_i \right] : s \subset \{1, 2, \dots, j\} \right\}$$

Knotenpartitionen

- Jeder Graph G kann geschrieben werden als $G = G[P_1] \cup G[P_2] \cup \dots \cup G[P_n]$, wobei jedes $G[P_i]$ eine Komponente von G ist.
- Sei dann $\{P_1, P_2, \dots, P_n\}$ die Komponenten-Knoten-Partition von G .
- Falls G nicht zusammenhängend ist, können die induzierten Teilgraphen auf der rechten Seite im letzten Satz unzusammenhängend sein.
- Sei $Z = (C_1, C_2, \dots, C_m)$ ein Slicing von G , wobei G die Komponenten-Knoten-Partition $\{P_1, P_2, \dots, P_n\}$ hat. Dann hat $G - \bigcup_{k=1}^j C_k$ die Komponenten-Knoten-Partition $G - \bigcup_{k=1}^j C_k =$

$$\bigcup \left\{ G \left[P_l \cap \bigcap_{i \in s} A_i \cap \bigcap_{i \notin s} \bar{A}_i \right] : 1 \leq l \leq n, s \subset \{1, 2, \dots, j\} \right\}$$

Knotenpartitionen

- Man beachte, dass die Komponenten-Knoten-Partition von $G - \bigcup_{k=1}^j C_k$ eine Subpartition der Komponenten-Knoten-Partition von $G - \bigcup_{k=1}^{j-1} C_k$ für $j \in \{1, \dots, m\}$ ist.
- D.h., das Slicing $Z = (C_1, C_2, \dots, C_m)$ bewirkt eine geschachtelte Sequenz von $m + 1$ Knotensubpartitionen von der Komponenten-Knoten-Partition $\{P_1, P_2, \dots, P_n\}$ bis hinunter zur minimalen Partition bestehend aus lauter einzelnen Knoten.
- Ein Cut C_i des Slicings $Z = (C_1, C_2, \dots, C_m)$ kann einige der Komponenten von $G - \bigcup_{j=1}^{i-1} C_j$ intakt lassen. Es ist daher sinnvoll zu spezifizieren, welche Komponenten von C_i wirklich zertrennt werden.

Knotenpartitionen

- Die Subgraphen G_1, G_2, \dots, G_m von G sind die *durch das Slicing $Z = (C_1, C_2, \dots, C_m)$ zertrennten Teilgraphen*, wenn jedes G_i genau die Komponenten von $G - \bigcup_{j=1}^{i-1} C_j$ enthält, deren Knoten durch C_i separiert werden.
- Jeder vom Slicing Z zertrennte Teilgraph G_i ist dann eine Vereinigung von induzierten Teilgraphen von G .
- Da ein minimales Slicing nur sukzessive Cuts von individuellen Komponenten beinhaltet, gilt das folgende Korollar.

Knotenpartitionen

Folgerung

Die durch ein minimales Slicing $Z = (C_1, C_2, \dots, C_m)$ zertrennten Teilgraphen sind alle zusammenhängende induzierte Teilgraphen von G .

- D.h., die geschachtelte Sequenz von Komponenten-Knoten-Partitionen für ein minimales Slicing ist derart, dass jede Subpartition aus der vorhergehenden durch Aufspaltung von genau einem Teil hervorgeht, und entspricht demzufolge einer Maximalen Kette im Verband der Komponenten-Knoten-Partitionen.

Länge von Slicings

Definition

Sei die **Länge** $\ell(Z)$ eines Slicings Z des Graphen G die Anzahl der Cuts des Slicings (also die Kardinalität der Kantenpartition Z)

- Jeder Cut des Slicings erhöht die Anzahl der Zusammenhangskomponenten im Graphen.
- Diese Erhöhung ist genau dann immer gleich Eins, wenn die Cuts minimal sind (bzw. das Slicing ein minimales Slicing ist).

Satz

*Für jeden Graphen G mit $|E(G)| \geq 1$ gilt:
 $\max\{\ell(Z) \mid Z \text{ ist ein Slicing von } G\} = |V(G)| - \#\text{Komponenten}(G)$
und dieses Maximum wird genau von den minimalen Slicings erreicht. (Die Größe rechts heißt auch Cocycle-Rang von G .)*

Länge von Slicings

Definition

Ein Graph ist k -partit, wenn die Knotenmenge $V(G)$ in k Mengen V_1, \dots, V_k partitioniert werden kann, so dass jeder induzierte Graph $G[V_i]$ ($i \in \{1, \dots, k\}$) keine Kante enthält.

- Die Länge eines Slicings Z von einem Graphen G kann Eins sein, nämlich dann, wenn G bipartit ist.
- Allgemein hängt der minimale Wert für die Länge eines Slicings eines Graphen mit der minimalen Zahl k zusammen, so dass G ein k -partiter Graph ist.
- Die minimale Anzahl k , so dass G k -partit ist, heißt **chromatische Zahl** $\chi(G)$ des Graphen G .

Länge von Slicings

Satz

Für jeden Graph G mit $|E(G)| \geq 1$ gilt:

$$\min\{\ell(Z) : Z \text{ ist ein Slicing von } G\} = \lceil \log_2 \chi(G) \rceil$$

Beweis.

siehe D. Matula, SIAM J. Appl. Math. 22(3), 1972. □

Weite von Slicings

Definition

Die **Weite** $w(Z)$ eines Slicings Z des Graphen G sei definiert durch

$$w(Z) = \max\{|C| : c \text{ ist ein Cut von } G\}$$

und jeder Cut C von Z mit $|C| = w(Z)$ ist ein **Wide Cut** des Slicings Z .

- Man beachte, dass ein MinCut sich auf den Graph bezieht, während ein Wide Cut sich auf ein bestimmtes Slicing bezieht.

Weite von Slicings

- Da jeder Cut $C_i = (A_i, \bar{A}_i)_i$ eines Slicings Z von G in einem Cut (A_i, \bar{A}_i) von G enthalten ist, und da jedes Slicing mit einem beliebigem Cut von G beginnen kann, muss die maximale Weite eines Cuts gleich der maximalen Anzahl von Kanten in einem beliebigen Cut sein.

Satz

Für jeden Graphen mit $|E(G)| \geq 1$ gilt:

$$\max\{w(Z) : Z \text{ ist Slicing von } G\} = \max\{|C| : C \text{ ist Cut von } G\}$$

Weite von Slicings

- Die minimale Weite eines Slicings ist ähnlich verwandt zu MinCuts, aber nicht vom ganzen Graphen G .

Satz

Für jeden Graphen mit $|E(G)| \geq 1$ gilt:

$$\min\{w(Z) : Z \text{ ist Slicing von } G\} = \sigma(G)$$

- Es ergibt sich damit folgende Dualität zwischen Slicings und Subgraphen:

$$\min_Z \max_C \{|C| : C \text{ ist Cut des Slicings } Z \text{ von } G\} =$$

$$\max_{G'} \min_C \{|C| : C \text{ ist Cut des Teilgraphen } G' \text{ von } G\}$$

Weite von Slicings

Beweis.

$\min\{w(Z) : Z \text{ ist Slicing von } G\} \geq$
 $\max\{\lambda(G') : G' \text{ ist Teilgraph von } G\} = \sigma(G):$

- Sei K ein Cluster von G mit $\lambda(K) = \sigma(K)$.
- Dann gilt für jedes Slicing Z von G , dass es einen ersten Cut C_i gibt, der Knoten von K separiert.
- Dann muss C_i einen Cut für K enthalten, so dass $|C_i| \geq \lambda(K) = \sigma(K)$.
- Also gilt $w(Z) \geq \sigma(Z)$ für jedes Slicing Z .

Weite von Slicings

Beweis.

$$\min\{w(Z) : Z \text{ ist Slicing von } G\} \leq \\ \sigma(G) = \max\{\lambda(G') : G' \text{ ist Teilgraph von } G\} = \sigma(G):$$

- Sei $Z^* = (C_1, \dots, C_m)$ ein Narrow Slicing von G mit Wide Cut C_i .
- Sei G_i der Teilgraph von G , der durch C_i zertrennt wird.
- Da ein Narrow Slicing nur Minimum Cuts benutzt, folgt $\lambda(G_i) = w(Z^*)$.



Weite von Slicings

- Die Weite eines Slicings muss größer oder gleich der durchschnittlichen Anzahl der Kanten in den Cuts des Slicings sein.
- Aus dem letzten Satz und dem Satz über die maximale Länge eines Slicings folgt damit die folgende untere Schranke für die Stärke $\sigma(G)$:

Folgerung

Für jeden Graphen mit $|E(G)| \geq 1$ gilt:

$$\sigma(G) \geq |E(G)| / (|V(G)| - 1) > |E(G)| / |V(G)|$$

Narrow Slicings

- Im Beweis des letzten Satzes war jedes Narrow Slicing ausreichend, um die Ungleichung der zweiten Richtung zu beweisen. Deshalb gelten folgende Korollare:

Folgerung

Für jedes Narrow Slicing Z^ von G gilt: $w(Z^*) = \sigma(G)$.*

Folgerung

Jeder Wide Cut eines Narrow Slicings von G ist ein Minimum Cut eines Subclusters von G .

- Die nächsten zwei Folgerungen zeigen, dass man ein Narrow Slicing benutzen kann, um die k -Komponenten ($k \geq 1$) und die Zusammenhangsfunktion zu berechnen.

Narrow Slicings und k -Komponenten

Folgerung

Seien $G[A_1], G[A_2], \dots, G[A_m]$ die Teilgraphen, die durch das Narrow Slicing $Z = (C_1, C_2, \dots, C_m)$ getrennt werden.

Dann ist jede k -Komponente von G gleich einem $G[A_i]$ für ein $i \in \{1, \dots, m\}$.

Weiterhin ist ein $G[A_i]$ ($i \in \{1, \dots, m\}$) genau dann eine k -Komponente von G , wenn

$$A_j \supset A_i \text{ für } j < i \Rightarrow |C_j| < |C_i|.$$

Und solch ein $G[A_i]$ ist genau dann ein Cluster von G , wenn auch gilt

$$A_j \subset A_i \text{ für } j > i \Rightarrow |C_j| \leq |C_i|.$$

Narrow Slicings und k -Komponenten

Beweis.

- Seien $G[A_1], \dots, G[A_m]$ die Teilgraphen, die durch das Narrow Slicing $Z = (C_1, C_2, \dots, C_m)$ getrennt werden, so dass $\lambda(G[A_i]) = |C_i|$ ($i \in \{1, \dots, m\}$).
- Sei H eine k -Komponente von G für ein $k \in \{1, \dots, \sigma(G)\}$.
- Dann muss der erste Cut C_n aus Z , der Knoten aus H separiert, einen Cut von H enthalten, also $|C_n| = \lambda(G[A_n]) \geq \lambda(H)$.
- Da H zusammenhängend ist, muss H ein Teilgraph von $G[A_n]$, weil C_n der erste Cut ist, der Knoten aus H separiert.
- Daraus folgt, dass $G[A_n]$ ein $\lambda(H)$ -kanten-zusammenhängender Graph ist ($\lambda(H) \leq \lambda(G[A_n])$).
- Da H ein maximaler $\lambda(H)$ -kanten-zusammenhängender Graph (per Def.) sowie ein Teilgraph von $G[A_n]$ ist, gilt $H = G[A_n]$.

Narrow Slicings und k -Komponenten

Beweis.

- $G[A_i]$ ($i \in \{1, \dots, m\}$) ist in einer $\lambda(G[A_i])$ -Komponente H von G enthalten, und aufgrund des vorangegangenen Arguments gilt $H = G[A_j]$ für ein $j \leq i$.
- Also ist $G[A_i]$ selbst eine $\lambda(G[A_i])$ -Komponente von G genau dann, wenn $A_j \supset A_i$ für $j < i$ impliziert, dass $|C_j| < |C_i|$.
- Falls weiterhin $G[A_i]$ eine $\lambda(G[A_i])$ -Komponente ist, dann handelt es sich genau dann um ein Cluster, wenn $A_j \subset A_i$ für $j > i$ impliziert, dass $|C_j| \leq |C_i|$.
- Also enthalten die Teilgraphen, die durch ein beliebiges Narrow Slicing von G zertrennt werden, alle k -Komponenten von G , und damit alle Cluster von G , die keine isolierten Knoten sind.



Narrow Slicings und Kohäsionsfunktion

Folgerung

Seien $G[A_1], G[A_2], \dots, G[A_m]$ die Teilgraphen, die durch das Narrow Slicing $Z = (C_1, C_2, \dots, C_m)$ getrennt werden.

Dann gilt für jedes Graphenelement $x \in V(G) \cup E(G)$:

$$h(x) = \begin{cases} 0, & \text{falls } x \text{ ein isolierter Knoten in } G \text{ ist,} \\ \max_i \{|C_i| : x \in A_i \cup E(G[A_i])\}, & \text{sonst} \end{cases}$$

- Wenn $G[A_1], \dots, G[A_m]$ die Subgraphen sind, die durch ein Narrow Slicing $Z = (C_1, \dots, C_m)$ von G zertrennt werden, dann müssen nicht alle m Subgraphen $G[A_i]$ k -Komponenten sein.
- Insbesondere, wenn $G[A_j]$ ein Cluster mit $\lambda(G[A_j]) \geq 2$ ist, dann sind mindestens $\lambda(G[A_j]) - 1$ der Teilgraphen $G[A_j]$ mit $j > i$ echte Teilgraphen von $G[A_j]$ und können demzufolge keine k -Komponenten von G für ein $k \geq 1$ sein.

Berechnung der Zusammenhangsfunktion

- Der Algorithmus basiert auf der sequentiellen Bestimmung von globalen Minimum Cuts für höchstens $|V(G)| - 1$ induzierte Teilgraphen von G .

Narrow Slicing Algorithmus

Algorithmus 15 : Berechnung eines Narrow Slicings (Matula)

Input : Graph $G = (V, E)$ mit N Komponenten und mindestens einer Kante

Output : Ein Narrow Slicing Z

Repräsentiere G als Vereinigung seiner Komponenten $G = \bigcup_{j=1}^N G[P_{1,j}]$;

$i \leftarrow 1$;

while $i < |V(G)| - N$ **do**

Wähle $G_i = G[P_{i,k}]$ für ein k , so dass $|P_{i,k}| \geq 2$;

Finde einen MinCut $C_i = (A, \bar{A})$ von G_i . (Beachte $A \cup \bar{A} = V(G_i)$);

Definiere neue Komponenten-Knoten-Partition $\{P_{i+1,j}\}$ ($j \in \{1, \dots, N+i\}$):

$$P_{i+1,j} = \begin{cases} P_{i,j} & \text{für } 1 \leq j < k, \\ A & \text{für } j = k, \\ \bar{A} & \text{für } j = k + 1, \\ P_{i,j-1} & \text{für } k + 1 < j \leq N + i, \end{cases}$$

$$\text{mit } G = \bigcup_{n=1}^i C_n = \bigcup_{j=1}^{N+i} G[P_{i+1,j}];$$

$i \leftarrow i + 1$;

$|V(G)| - N$

$G = \bigcup_{n=1}^{|V(G)| - N} C_n$ besteht jetzt nur noch aus isolierten Knoten und

$Z = (C_1, \dots, C_{|V(G)| - N})$ ist ein Narrow Slicing;

return Z ;

Narrow Slicing Algorithmus

- Ursprünglich wurde von Matula vorgeschlagen, den MinCut mit Hilfe des Cut Trees von Gomory/Hu zu berechnen. Hier kann man aber auch andere Algorithmen einsetzen, z.B. den Stoer/Wagner-Algorithmus.
- Der Narrow Slicing Algorithmus bestimmt explizit ein Narrow Slicing $Z = (C_1, \dots, C_{|V(G)|-M})$, eine Liste $G_1, \dots, G_{|V(G)|-M}$ von induzierten Graphen, die durch das Slicing zertrennt werden, sowie die durch Z verursachte geschachtelte Sequenz von Komponenten-Knoten-Partitionen.
- Daraus können dann entsprechend den früheren Sätzen sehr einfach die Zusammenhangsfunktion, die k -Komponenten und die Cluster von G bestimmt werden.

Narrow Slicing Algorithmus

- $G_i = G[P_{i,k}]$ hat höchstens $|V(G)| + 2 - N - i$ Knoten.
- Also müssten bei Verwendung der Cut Tree Methode höchstens

$$\sum_{i=1}^{|V(G)|-N} (|V(G)| + 1 - N - i) = \frac{1}{2}(|V(G)| - N)(|V(G)| - N + 1)$$

Flussprobleme einfacher Art gelöst werden.

- Für einen zusammenhängenden Graphen wären das höchstens $\binom{|V(G)|}{2}$ Flussprobleme.
- Matula gibt die Komplexität grob mit n^4 bis n^5 an.
- Das entspricht auch ungefähr der Größenordnung, die man bei Verwendung des Stoer/Wagner-Algorithmusses errechnen würde ($\mathcal{O}(m_i n_i + n_i^2 \log n_i)$ pro Teilproblem mit n_i Knoten und m_i Kanten, summiert für $n_i = n - 1, \dots, 1$)