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

Aufzählung in lexikographischer Reihenfolge

- Idee: während der Generierung der aktuellen Ausgabe wird zusätzliche Arbeit in die Erzeugung lexikographisch größerer Cliquen investiert

$\Rightarrow$ Diese werden in einer Priority Queue gespeichert, die dann u.U. exponentiell viele Cliquen enthält und damit auch exponentiell viel Speicherplatz braucht.

Aufzählung in lexikographischer Reihenfolge

Algorithmus 3 : Alg. von Johnson, Papadimitriou und Yannakakis

U_0 := erste (lexikographisch kleinste) maximale Clique;
Füge U_0 in Priority Queue Q ein;
while Q *ist nicht leer* **do**
  U :=ExtractMin(Q);
  Ausgabe U;
  foreach *Knoten* v_j *von* G*, der zu einem Knoten* $v_i \in U$ *mit* $i < j$ *nicht adjazent ist* **do**
    $U_j := U \cap \{v_1, \ldots, v_j\}$;
    if $(U_j - \overline{N}(v_j)) \cup \{v_j\}$ *ist eine maximale Clique in* $G[\{v_1, \ldots, v_j\}]$ **then**
      Sei T die lexikographisch kleinste maximale Clique, die $(U_j - \overline{N}(v_j)) \cup \{v_j\}$ enthält;
      Füge T in Q ein

Aufzählung in lexikographischer Reihenfolge

Satz

Der Algorithmus von Johnson, Papadimitriou und Yannakakis zählt alle maximalen Cliquen eines Graphen mit n Knoten in lexikographischer Reihenfolge und mit Delay $\mathcal{O}(n^3)$ auf.

Aufzählung in lexikographischer Reihenfolge

Beweis.

Korrektheit:

- Menge T ist beim Einfügen in Q (bei Betrachtung von U) lexikographisch größer als U
  (denn es wird ja aus U zumindest Knoten v_i entnommen während lediglich v_j hinzukommt und es gilt $i < j$)

$\Rightarrow$ Es werden nur Mengen in der Queue gespeichert, die erst nach U ausgegeben werden dürfen.

$\Rightarrow$ Die ausgegebenen maximalen Cliquen sind lexikographisch aufsteigend.

Aufzählung in lexikographischer Reihenfolge

Beweis.

Vollständigkeit:

- Falls U die lexikographisch kleinste noch auszugebende Clique ist, dann ist U in Q.
- Induktionsanfang: Für $U = U_0$ ist das korrekt.
- Induktionsschritt: Sei U lexikographisch größer als U_0.
  - Sei j der größte Index, dass $U_j = U \cap \{v_1, \ldots, v_j\}$ *keine* maximale Clique in $G[\{v_1, \ldots, v_j\}]$.
    (Muss existieren, weil sonst $U = U_0$. Außerdem muss $j < n$ sein, weil U eine maximale Clique des Gesamtgraphen G ist.)
  - Aufgrund der Maximalität von j muss gelten $v_{j+1} \in U$.
  - $\exists$ Menge S: $U_j \cup S$ ist maximale Clique in $G[\{v_1, \ldots, v_j\}]$

Aufzählung in lexikographischer Reihenfolge

Beweis.

- Induktionsschritt (Fortsetzung):
  - Aufgrund der Maximalität von j ist v_{j+1} nicht adjazent zu allen Knoten in S.

  $\Rightarrow$ $\exists$ maximale Clique U', die zwar $U_j \cup S$, aber nicht v_{j+1} enthält
  - $U' \leq U$, weil sie sich in S unterscheiden
  - U' wurde schon ausgegeben (Ind.voraussetzung)
  - Als U' ausgegeben wurde, wurde festgestellt, dass v_{j+1} nicht adjazent ist zu einem Knoten $v_i \in U'$ mit $i < j+1$
  - $(U'_{j+1} \setminus \bar{N}(v_{j+1})) \cup \{v_{j+1}\} = U_{j+1}$
    und U_{j+1} ist maximale Clique in $G[\{v_1, \ldots, v_{j+1}\}]$

  $\Rightarrow$ Die lexikographisch kleinste maximale Clique, die U_{j+1} enthält, wurde in Q eingefügt.

Aufzählung in lexikographischer Reihenfolge

Beweis.

- Induktionsschritt (Fortsetzung):
  - Aufgrund der Maximalität von j stimmen U und T in den ersten $j+1$ Knoten überein.
  - Annahme: $U \neq T$
    Sei k der kleinste Index eines Knoten v_k, der in genau einer der beiden Mengen ist.
  - $k > j+1$
  - Da $T \leq U$, gilt $v_k \in T$ und $v_k \notin U$.

  $\Rightarrow$ U_k ist nicht maximale Clique in $G[\{v_1, \ldots, v_k\}]$
  (Widerspruch zur Maximalität von j)

  $\Rightarrow$ $U = T$

  $\Rightarrow$ U ist in der Priority Queue Q

$\square$

Aufzählung in lexikographischer Reihenfolge

Komplexität:

- Extraktion der lexikographisch kleinsten maximalen Clique aus Q: $\mathcal{O}(n \log C)$
- n Berechnungen von maximalen Cliquen, die eine gegebene Menge enthalten: $\mathcal{O}(m+n)$ pro Menge
- Einfügen einer maximalen Clique in Q: $\mathcal{O}(n \log C)$ pro Clique
- Da $C \leq 3^{\lceil \frac{n}{3} \rceil}$, folgt dass die Verzögerung $\mathcal{O}(n^3)$ ist.

Plex

Relaxiere Cliquenbegriff, so dass konstant viele Verbindungen zu Gruppenmitgliedern bei jedem Knoten fehlen dürfen.

Definition

Sei $G = (V, E)$ ein ungerichteter Graph und sei $k \in \{1, \ldots, n-1\}$ eine natürliche Zahl.
Dann wird ein Subset $U \subseteq V$ als k-Plex bezeichnet, falls $\delta(G[U]) \geq |U| - k$.

k-Plex-Eigenschaften

- Jede Clique ist ein 1-Plex.
- Jedes k-Plex ist auch ein $(k+1)$-Plex.
- Ein *maximales k-Plex* ist in keinem größeren k-Plex echt enthalten.
- Ein *Maximum k-Plex* hat maximale Kardinalität unter allen k-Plexen in G.
- Jeder induzierte Teilgraph eines k-Plex ist auch ein k-Plex, d.h. die k-Plex-Eigenschaft ist abgeschlossen unter Exklusion.

k-Plex-Durchmesser

Satz

Sei $k \in \{1, \dots, n-1\}$ eine natürliche Zahl und sei $G = (V, E)$ ein ungerichteter Graph auf n Knoten. Dann gilt:

1. *Wenn V ein k-Plex mit $k < \frac{n+2}{2}$ ist, dann gilt $diam(G) \leq 2$. Falls zusätzlich $n \geq 4$, dann ist G zweifach kantenzusammenhängend.*
2. *Wenn V ein k-Plex mit $k \geq \frac{n+2}{2}$ und G zusammenhängend ist, dann gilt $diam(G) \leq 2k - n + 2$.*

k-Plex-Durchmesser

Beweis.

Fall $k < \frac{n+2}{2}$:

- Seien $u, v \in V$ nicht adjazente Knoten ($u \neq v$) (adjazente Knoten haben $d(u,v) = 1$)
- Annahme: $d(u,v) \geq 3 \Rightarrow N(u) \cap N(v) = \emptyset$, d.h.

$$n-2 \geq |N(u) \cup N(v)| \geq 2\delta(G) \geq 2(n-k) > \dots$$

$$\dots > 2\left(n - \frac{n+2}{2}\right) = n-2$$

(Widerspruch)

$\Rightarrow\ d(u,v) \leq 2 \quad \Rightarrow \quad \mathrm{diam}(G) \leq 2$

k-Plex-Durchmesser

Beweis.

Fall $k < \frac{n+2}{2}$, $n \geq 4$:

- Annahme: $\exists$ Brücke e (d.h. $G - \{e\}$ enthält zwei Zusammenhangskomponenten V_1 und V_2)
- Jeder kürzeste Pfad von einem Knoten in V_1 zu einem Knoten in V_2 muss diese Brücke benutzen.

$\Rightarrow$ Da $\mathrm{diam}(G) \leq 2$, muss eine Komponente ein einzelner Knoten sein. Dieser Knoten hat Grad 1.

- Da V ein k-Plex mit $n \geq 4$ Knoten ist, gilt für den Grad dieses Knotens v: $\deg(v) \geq n - k > n - \frac{n+2}{2} = \frac{n-2}{2} \geq 1$ (Widerspruch)

$\Rightarrow$ Es existiert keine Brücke in G bzw.
G ist 2-fach kantenzusammenhängend.

k-Plex-Durchmesser

Beweis.

Fall $k \geq \frac{n+2}{2}$:

- Sei $\{v_0, v_1, \ldots, v_r\}$ ein längster kürzester Pfad, also ein Pfad mit $d(v_0, v_r) = r = \text{diam}(G)$.
- Wir können annehmen, dass $r \geq 4$.
- Da es keinen kürzeren Pfad zwischen v_0 und v_r gibt, ist v_i zu keinem der Knoten $v_0, \ldots, v_{i-2}, v_{i+2}, \ldots, v_r$ auf dem Pfad adjazent, außer zu seinen Pfad-Nachbarknoten v_{i-1} und v_{i+1}
- Außerdem kann kein Knoten existieren, der gleichzeitig zu v_0 und zu v_3 adjazent ist.

$\Rightarrow$ $\{v_0\} \uplus \{v_2, v_3, \ldots, v_r\} \uplus (N(v_3) \setminus \{v_2, v_4\}) \subseteq \overline{N}(v_0)$

$\Rightarrow$ $1 + (r-1) + d_G(v_3) - 2 \leq k \quad \Rightarrow \quad r + n - k - 2 \leq k$

$\Rightarrow$ $\text{diam}(G) = r \leq 2k - n + 2$

k-Plex Problem

- (variables) Entscheidungsproblem **Plex** mit Eingabe
  - Graph G,
  - Größenparameter ℓ und
  - Plex-Parameter k

  ist $\mathcal{NP}$-vollständig, da das **Clique**-Problem sich darauf reduzieren lässt (mit $k = 1$)

$\Rightarrow$ Betrachte das Problem für feste Plex-Parameter c

Problem

Problem:	c-***Plex***
Eingabe:	*Graph* G, *Parameter* $\ell \in \mathbb{N}$
Frage:	*Existiert ein* c-*Plex der Kardinalität* $\geq \ell$ *in* G?

c-Plex

Genau wie 1-**Plex**=**Clique** sind auch alle anderen Probleme für einen festen Plex-Parameter c $\mathcal{NP}$-vollständig:

Satz

*c-**Plex** ist $\mathcal{NP}$-vollständig für alle $c \in \mathbb{N}$.*

Beweis: durch Reduktion von **Clique**, siehe z.B. Brandes/Erlebach (Eds.): Network Analysis (S. 128).

Cores

Relaxiere Cliquenbegriff, so dass konstant viele Verbindungen zu Gruppenmitgliedern bei jedem Knoten mindestens vorhanden sein müssen.

Definition

Sei $G = (V, E)$ ein ungerichteter Graph und sei $k \in \{1, \ldots, n-1\}$ eine natürliche Zahl.
Dann wird ein Subset $U \subseteq V$ als k-Core (k-Kern) bezeichnet, falls $\delta(G[U]) \geq k$.
Ein *maximaler k-Core* ist in keinem größeren k-Core echt enthalten.
Ein *Maximum k-Core* hat maximale Kardinalität unter allen k-Cores in G. Jeder Maximum-k-Core ist auch ein maximaler k-Core (was umgekehrt nicht unbedingt gilt).

k-Core-Eigenschaften

- Jeder Graph G ist ein $\delta(G)$-Core, jede Clique ist ein $n-1$-Core.
- Jeder $(k+1)$-Core ist auch ein k-Core.
- Jeder k-Core ist ein $(n-k)$-Plex.
- Wenn U und U' k-Cores sind, dann ist auch $(U \cup U')$ ein k-Core.

$\Rightarrow$ Maximale k-Cores sind einzigartig.

- Nicht jeder induzierte Teilgraph eines k-Cores ist auch wieder ein k-Core, d.h. die k-Core-Eigenschaft ist *nicht* abgeschlossen unter Exklusion.
  Bsp.: Jeder Kreis ist ein 2-Core, aber jeder echte Teilgraph enthält nmindestens einen Knoten vom Grad < 2.
- k-Cores sind auch nicht geschachtelt. (Nicht jeder k-Core der Größe n enthält einen k-Core der Größe $n-1$.)
- k-Cores müssen nicht unbedingt zusammenhängend sein.

Maximale zusammenhängende k-Cores

Fakt

Sei $G = (V, E)$ ein ungerichteter Graph und $k > 0$ eine natürliche Zahl. Wenn U und U' zwei maximale zusammenhängende k-Cores in G mit $U \neq U'$ sind, dann gibt es keine Kante zwischen U und U'.

Folgerung

- *Der einzige Maximum k-Core eines Graphen ist die Vereinigung seiner maximalen zusammenhängenden k-Cores.*
- *Der Maximum 2-Core eines zusammenhängenden Graphen ist zusammenhängend.*
  (Wenn er es nicht wäre, könnte man die Teile verbinden und alle neuen Knoten hätten mindestens Grad 2.)
- *Ein Graph ist genau dann ein Wald, wenn er keine 2-Cores enthält.*

Der Maximum k-Core

Satz

Sei $G = (V, E)$ ein ungerichteter Graph und $k > 0$ eine natürliche Zahl. Wenn man wiederholt alle Knoten mit Grad $< k$ (und alle inzidenten Kanten) entfernt, dann entspricht die verbleibende Knotenmenge U genau dem Maximum k-Core.

Beweis.

- U ist ein k-Core, d.h. wir müssen nur noch Maximum zeigen.
- Annahme: U ist nicht Maximum k-Core.

$\Rightarrow$ $\exists$ nichtleere Menge $T \subseteq V$, so dass $U \cup T$ Maximum k-Core

- Als der erste Knoten von T aus G entfernt wurde, muss er Grad $< k$ gehabt haben. Das kann aber nicht sein, da er mindestens k Nachbarn in $U \cup T$ hat und alle anderen Knoten noch im Graph enthalten waren. (Widerspruch)

Core-Zerlegung

Definition

Die Core-Zahl $\xi_G(v)$ eines Knotens $v \in V$ ist die höchste Zahl k, so dass v Teil eines k-Cores im Graphen G ist, d.h.

$$\xi_G(v) = \max\{k : \exists k\text{-Core } U \text{ in } G \text{ mit } v \in U\}$$

Algorithmus zur Berechnung arbeitet nach folgendem Prinzip:

- Jeder Graph G ist ein $\delta(G)$-Core.
- Jeder Nachbarknoten mit geringerem Grad verringert die potentielle Core-Zahl eines Knotens.

Berechnung der Core-Zahlen

Algorithmus 4 : Berechnung der Core-Zahlen

Input : Graph $G = (V, E)$
Output : Array ξ_G mit den Core-Zahlen aller Knoten in G

Berechne die Grade aller Knoten und speichere sie in D;
Sortiere V in aufsteigender Grad-Folge D;
foreach $v \in V$ in sortierter Reihenfolge **do**
  $\xi_G(v) := D[v]$;
  foreach *Knoten u adjazent zu v* **do**
    if $D[u] > D[v]$ **then**
      $D[u] := D[u] - 1$;
      Sortiere V in aufsteigender Grad-Reihenfolge nach D

Berechnung der Core-Zahlen

Komplexität:

- naiv: $\mathcal{O}(mn \log n)$
  (teuerste Operationen: Sortieren der Knoten nach dem Grad)
- besser: $\mathcal{O}(m + n)$
  - Knotengrade haben Werte aus dem Intervall $[0, n-1]$
    $\Rightarrow$ Sortieren mit n Buckets in $\mathcal{O}(n)$
  - Nachsortieren kann man sich sparen, indem man sich merkt an welchen Indizes im Array die Knoten des nächsten Grads beginnen
    Beim dekrementieren des Werts eines Knotens kommt der Knoten einfach an den Anfang des alten Intervalls und dann wird die Grenze um eine Stelle verschoben, so dass er nun zum Intervall des kleineren Grads gehört (das geht in $\mathcal{O}(1)$).
  - insgesamt: $\mathcal{O}(n)$ für Initialisierung / Sortieren,
    dann $\mathcal{O}(m)$ für die Schleifendurchläufe (jede Kante wird höchstens zweimal betrachtet), also $\mathcal{O}(m + n)$

Dichte Subgraphen

Definition

Sei $G = (V, E)$ ein ungerichteter Graph mit n Knoten und m Kanten. Die Dichte $\rho(G)$ des Graphen G ist definiert als

$$\rho(G) = \frac{m}{\binom{n}{2}}$$

Ein durch eine Knotenteilmenge $U \subseteq V$ eines Graphen $G = (V, E)$ induzierter Teilgraph heißt η-dicht (für eine reelle Zahl η mit $0 \leq \eta \leq 1$), falls gilt:

$$\rho(G[U]) \geq \eta$$

Aber: selbst (Teil-)Graphen mit hoher Dichte können isolierte Knoten enthalten!

Eigenschaften

- Cliquen sind 1-dichte (Teil-)Graphen.
- Ein k-Plex hat Dichte $1 - \frac{k-1}{n-1}$

$\Rightarrow$ Für $n \rightarrow \infty$ gilt für k-Plexe $\eta \rightarrow 1$) bzw.
$\forall k > 0, 0 \leq \eta \leq 1$: ein k-Plex der Größe mindestens $\frac{k-\eta}{1-\eta}$ ist ein η-dichter (Teil-)Graph.

- Aber: nicht jeder $1 - \frac{k-1}{n-1}$-dichte (Teil-)Graph ist auch ein k-Plex.
- Ein k-Core ist ein $\frac{k}{n-1}$-dichter (Teil-)Graph.
  Die Dichte von k-Cores kann sich für $n \rightarrow \infty$ beliebig nah an 0 annähern.
- η-dichte Graphen sind nicht abgeschlossen unter Exklusion (Knotenausschluss), sie sind aber geschachtelt.

Schachtelung von η-dichten Graphen

Satz

Sei η eine reelle Zahl mit $0 \leq \eta \leq 1$. Dann gilt: Ein η-dichter Teilgraph der Größe ℓ eines Graphen G enthält einen η-dichten Teilgraph der Größe $\ell - 1$ in G.

Schachtelung von η-dichten Graphen

Beweis.

- Sei
  - U ein η-dichter Teilgraph von G mit $|U| = \ell$,
  - m_U die Anzahl der Kanten in $G[U]$,
  - v ein Knoten mit minimalem Grad in $G[U]$ (also $\deg_{G[U]}(v) = \delta(G[U])$)
- $\delta(G[U]) \leq \overline{d}(G[U]) = \frac{2m_U}{\ell} = \rho(G[U])(\ell - 1)$
- Betrachte Knotenteilmenge $U' = U \setminus \{v\}$ mit Kantenanzahl $m_U - \delta(G[U]) \geq \rho(G[U])\binom{\ell}{2} - \rho(G[U])(\ell - 1) = \rho(G[U])\binom{\ell-1}{2}$

$\Rightarrow$ $\rho(G[U']) \geq \rho(G[U]) \geq \eta$

$\Rightarrow$ U' ist ein η-dichter Teilgraph der Größe $\ell - 1$.

□