Effiziente Approximation unabhängiger Mengen in Graphen

Dipl.-Inf. Matthias Baumgart

Diplomarbeit an der TU-Chemnitz

Gliederung

1. Einleitung

2. Der Algorithmus von Feige
   - Beweis der Korrektheit
   - Laufzeitanalyse

3. Eine Approximationsgüte von $O(n(\log\log n)^2/(\log n)^3$

Gliederung

Gliederung

Einleitung

- Entwicklung eines Algorithmus von Boppana und Halldórsson basierend auf der Ramsey-Theorie
- Idee: Graph mit $R(k,l)$ Knoten enthält eine Clique der Größe k oder eine unabhängige Menge der Größe l
- Entferne gefundene Clique und suche weiter
- Analoges gilt für die Approximation von Cliquen

- Idee von U. Feige: Entferne nicht nur unabhängige Mengen sondern auch Untergraphen mit „wenig“ Kanten

Definition

Eine schwache Knotenmenge S in einem Graphen $G = (V, E)$, der eine Clique C der Kardinalität $|C| \geq n/k$ enthält, ist eine Knotenmenge, so dass der auf S induzierte Untergraph von G keine Clique C_S enthält für die gilt

$$|C_S| \geq \frac{|S|}{2k} .$$

- Idee von U. Feige: Entferne nicht nur unabhängige Mengen sondern auch Untergraphen mit „wenig“ Kanten

Definition

Eine schwache Knotenmenge S in einem Graphen $G = (V, E)$, der eine Clique C der Kardinalität $|C| \geq n/k$ enthält, ist eine Knotenmenge, so dass der auf S induzierte Untergraph von G keine Clique C_S enthält für die gilt

$$|C_S| \geq \frac{|S|}{2k} .$$

Satz

Sei $G = (V, E)$ ein Graph mit einer Clique C der Kardinalität

$$|C| \geq \frac{n}{k} .$$

Seien $S_1, \ldots, S_l$ beliebige disjunkte schwache Knotenmengen von G und $G' = (V', E')$ der auf $V \setminus \{S_1 \cup \cdots \cup S_l\}$ induzierte Untergraph von G.

Dann gilt

$$|V'| \geq \frac{n}{2k}$$

und G' enthält eine Clique C' der Kardinalität

$$|C'| \geq \frac{|V'|}{k} .$$

Beweis.

Zu zeigen ist

1. für V' gilt $|V'| \geq n/(2k)$
   - Vereinigung $S = \bigcup_{i=1}^{l} S_i$ ist schwache Knotenmenge auf höchstens n Knoten
   - G enthält eine Clique C der Größe $|C| \geq n/k$
   - mindestens $n/(2k)$ Knoten der Clique C müssen in G' sein

Beweis.

Zu zeigen ist

1. für V' gilt $|V'| \geq n/(2k)$
2. für C' gilt $|C'| \geq |V'|/k$

   Annahme: G' enthält nur Cliquen C' mit $|C| < |V'|/k$

   - dann befinden sich jedoch mindestens

     $$\frac{|V|}{k} - \frac{|V'|}{k} = \frac{|S|}{k}$$

     viele Knoten der Clique C in der Knotenmenge S
   - Widerspruch, da S schwache Knotenmenge ist $\square$

Der Algorithmus von Feige

Der Algorithmus von Feige liefert nach Eingabe eines Graphen $G = (V, E)$, welcher eine Clique der Größe $|V|/k$ enthält, eine Clique C der Kardinalität

$$|C| \geq t \cdot \log_{3k}(|V|/t - 3) \ .$$

- Einteilung des Algorithmus in Phasen und Iterationen
- Phasen bestehen aus einzelnen Iterationen

Der Algorithmus von Feige liefert nach Eingabe eines Graphen $G = (V, E)$, welcher eine Clique der Größe $|V|/k$ enthält, eine Clique C der Kardinalität

$$|C| \geq t \cdot \log_{3k}(|V|/t - 3) .$$

- Einteilung des Algorithmus in Phasen und Iterationen
- Phasen bestehen aus einzelnen Iterationen

- Jede Phase arbeitet auf einem Graphen $G' = (V', E')$, welcher eine Clique der Größe $|V'|/k$ enthält.
- Nach Abarbeitung einzelner Iterationen endet eine Phase, so dass eine der folgenden zwei Bedingungen erfüllt ist:

1. Eine Clique C der Kardinalität

$$|C| \geq t \cdot \log_{3k}(|V'|/(6kt))$$

   wurde gefunden.
2. Ein schwacher Untergraph $G'' = (V'', E'')$ wurde gefunden.

- Jede Phase arbeitet auf einem Graphen $G' = (V', E')$, welcher eine Clique der Größe $|V'|/k$ enthält.
- Nach Abarbeitung einzelner Iterationen endet eine Phase, so dass eine der folgenden zwei Bedingungen erfüllt ist:

1. Eine Clique C der Kardinalität

$$|C| \geq t \cdot \log_{3k}(|V'|/(6kt))$$

   wurde gefunden.
2. Ein schwacher Untergraph $G'' = (V'', E'')$ wurde gefunden.

- Jede Phase arbeitet auf einem Graphen $G' = (V', E')$, welcher eine Clique der Größe $|V'|/k$ enthält.
- Nach Abarbeitung einzelner Iterationen endet eine Phase, so dass eine der folgenden zwei Bedingungen erfüllt ist:

1. Eine Clique C der Kardinalität
$$|C| \geq t \cdot \log_{3k}(|V'|/(6kt))$$
wurde gefunden.
2. Ein schwacher Untergraph $G'' = (V'', E'')$ wurde gefunden.

- Jede Phase arbeitet auf einem Graphen $G' = (V', E')$, welcher eine Clique der Größe $|V'|/k$ enthält.
- Nach Abarbeitung einzelner Iterationen endet eine Phase, so dass eine der folgenden zwei Bedingungen erfüllt ist:

1. Eine Clique C der Kardinalität
$$|C| \geq t \cdot \log_{3k}(|V'|/(6kt))$$
wurde gefunden.
2. Ein schwacher Untergraph $G'' = (V'', E'')$ wurde gefunden.

Jede Iteration arbeitet auf einem Graphen $G'' = (V'', E'')$ und führt folgende Schritte aus:

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus
2. Teile V'' in disjunkte Knotenmengen P_i mit $|P_i| = 2kt$
3. Betrachte alle t-elementigen Teilmengen S von P_i:
4. Sei $N(S)$ die Menge aller Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind, dann bezeichne S als gut, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ gilt
5. Falls eine gute Knotenmenge S gefunden wird, dann setze $C = C \cup S$ und starte neue Iteration mit dem auf $N(S)$ induzierten Untergraphen von G''
6. Sonst bezeichne V'' als schwach und beende Phase

Jede Iteration arbeitet auf einem Graphen $G'' = (V'', E'')$ und führt folgende Schritte aus:

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus
2. Teile V'' in disjunkte Knotenmengen P_i mit $|P_i| = 2kt$
3. Betrachte alle t-elementigen Teilmengen S von P_i:
4. Sei $N(S)$ die Menge aller Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind, dann bezeichne S als gut, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ gilt
5. Falls eine gute Knotenmenge S gefunden wird, dann setze $C = C \cup S$ und starte neue Iteration mit dem auf $N(S)$ induzierten Untergraphen von G''
6. Sonst bezeichne V'' als schwach und beende Phase

Jede Iteration arbeitet auf einem Graphen $G'' = (V'', E'')$ und führt folgende Schritte aus:

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus
2. Teile V'' in disjunkte Knotenmengen P_i mit $|P_i| = 2kt$
3. Betrachte alle t-elementigen Teilmengen S von P_i:
4. Sei $N(S)$ die Menge aller Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind, dann bezeichne S als gut, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ gilt
5. Falls eine gute Knotenmenge S gefunden wird, dann setze $C = C \cup S$ und starte neue Iteration mit dem auf $N(S)$ induzierten Untergraphen von G''
6. Sonst bezeichne V'' als schwach und beende Phase

Jede Iteration arbeitet auf einem Graphen $G'' = (V'', E'')$ und führt folgende Schritte aus:

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus
2. Teile V'' in disjunkte Knotenmengen P_i mit $|P_i| = 2kt$
3. Betrachte alle t-elementigen Teilmengen S von P_i:
4. Sei $N(S)$ die Menge aller Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind, dann bezeichne S als gut, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ gilt
5. Falls eine gute Knotenmenge S gefunden wird, dann setze $C = C \cup S$ und starte neue Iteration mit dem auf $N(S)$ induzierten Untergraphen von G''
6. Sonst bezeichne V'' als schwach und beende Phase

Jede Iteration arbeitet auf einem Graphen $G'' = (V'', E'')$ und führt folgende Schritte aus:

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus
2. Teile V'' in disjunkte Knotenmengen P_i mit $|P_i| = 2kt$
3. Betrachte alle t-elementigen Teilmengen S von P_i:
4. Sei $N(S)$ die Menge aller Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind, dann bezeichne S als gut, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ gilt
5. Falls eine gute Knotenmenge S gefunden wird, dann setze $C = C \cup S$ und starte neue Iteration mit dem auf $N(S)$ induzierten Untergraphen von G''
6. Sonst bezeichne V'' als schwach und beende Phase

Jede Iteration arbeitet auf einem Graphen $G'' = (V'', E'')$ und führt folgende Schritte aus:

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus
2. Teile V'' in disjunkte Knotenmengen P_i mit $|P_i| = 2kt$
3. Betrachte alle t-elementigen Teilmengen S von P_i:
4. Sei $N(S)$ die Menge aller Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind, dann bezeichne S als gut, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ gilt
5. Falls eine gute Knotenmenge S gefunden wird, dann setze $C = C \cup S$ und starte neue Iteration mit dem auf $N(S)$ induzierten Untergraphen von G''
6. Sonst bezeichne V'' als schwach und beende Phase

Jede Iteration arbeitet auf einem Graphen $G'' = (V'', E'')$ und führt folgende Schritte aus:

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus
2. Teile V'' in disjunkte Knotenmengen P_i mit $|P_i| = 2kt$
3. Betrachte alle t-elementigen Teilmengen S von P_i:
4. Sei $N(S)$ die Menge aller Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind, dann bezeichne S als gut, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ gilt
5. Falls eine gute Knotenmenge S gefunden wird, dann setze $C = C \cup S$ und starte neue Iteration mit dem auf $N(S)$ induzierten Untergraphen von G''
6. Sonst bezeichne V'' als schwach und beende Phase

Beweis der Korrektheit

Satz

Wenn eine Knotenmenge V'' vom Algorithmus von Feige als schwach erkannt wird, dann enthält der auf V'' induzierte Untergraph von $G = (V, E)$ tatsächlich keine Clique C der Kardinalität

$$|C| \geq \frac{|V''|}{2k} \, .$$

Beweis.

Annahme: $V'' = P_1 \cup \cdots \cup P_l$ enthält Clique C mit

$$|C| \geq \frac{|V''|}{2k} = t \cdot l$$

- nach dem Schubfachprinzip gibt es ein P_i, das t Knoten der Clique C enthält
- es gibt ein S mit der Eigenschaft gut

Beweis.

Annahme: $V'' = P_1 \cup \cdots \cup P_l$ enthält Clique C mit

$$|C| \geq \frac{|V''|}{2k} = t \cdot l$$

- nach dem Schubfachprinzip gibt es ein P_i, das t Knoten der Clique C enthält
- es gibt ein S mit der Eigenschaft gut $\square$

Satz

Endet eine Phase mit der Ausgabe einer Menge C, dann enthält diese mindestens

$$|C| \geq t \cdot \log_{3k} \frac{|V'|}{6kt}$$

Knoten, welche eine Clique in $G = (V, E)$ bilden.

Beweis.

- Menge C ist offensichtlich eine Clique
- Pro Iteration vergrößert sich C um t Knoten
- Bestimmung unterer Schranke für die Anzahl Iterationen

Beweis.

- Menge C ist offensichtlich eine Clique
- Pro Iteration vergrößert sich C um t Knoten
- Bestimmung unterer Schranke für die Anzahl Iterationen
  - die erste Iteration startet mit V' vielen Knoten
  - eine neue Iteration startet mit wenigstens $|V''|/(2k) - t$
  - es gilt $|V''| \geq 6kt$, also $t \leq |V''|/(6k)$
  - die Anzahl Knoten einer neuen Iteration ist mindestens

$$\frac{|V''|}{2k} - \frac{|V''|}{6k} = \frac{|V''|}{3k}$$

Beweis.

- Menge C ist offensichtlich eine Clique
- Pro Iteration vergrößert sich C um t Knoten
- Bestimmung unterer Schranke für die Anzahl Iterationen
  - die erste Iteration startet mit V' vielen Knoten
  - eine neue Iteration startet mit wenigstens $|V''|/(2k) - t$
  - es gilt $|V''| \geq 6kt$, also $t \leq |V''|/(6k)$
  - die Anzahl Knoten einer neuen Iteration ist mindestens

$$\frac{|V''|}{2k} - \frac{|V''|}{6k} = \frac{|V''|}{3k}$$

Beweis.

- Menge C ist offensichtlich eine Clique
- Pro Iteration vergrößert sich C um t Knoten
- Bestimmung unterer Schranke für die Anzahl Iterationen
  - die Anzahl Knoten der $i+1$'ten Iteration ist mindestens

$$\frac{|V'|}{(3k)^i}$$

  - Wann ist die Anzahl Knoten einer Iteration kleiner $6kt$?

$$\frac{|V'|}{(3k)^x} < 6kt \iff x > \log_{3k} \frac{|V'|}{6kt}$$

Beweis.

- Menge C ist offensichtlich eine Clique
- Pro Iteration vergrößert sich C um t Knoten
- Bestimmung unterer Schranke für die Anzahl Iterationen
  - die Anzahl Knoten der $i+1$'ten Iteration ist mindestens

$$\frac{|V'|}{(3k)^i}$$

  - Wann ist die Anzahl Knoten einer Iteration kleiner $6kt$?

$$\frac{|V'|}{(3k)^x} < 6kt \quad \Longleftrightarrow \quad x > \log_{3k} \frac{|V'|}{6kt}$$

Beweis.

- Menge C ist offensichtlich eine Clique
- Pro Iteration vergrößert sich C um t Knoten
- Bestimmung unterer Schranke für die Anzahl Iterationen

$$\text{Anzahl Iterationen} > \log_{3k} \frac{|V'|}{6kt}$$

- Damit erhalten wir

$$|C| \geq t \cdot \log_{3k} \frac{|V'|}{6kt}$$

□

Laufzeitanalyse

- Laufzeit polynomiell in n, wenn $\binom{2kt}{t}$ polynomiell in n
- Wahl des Parameters t beeinflußt Größe der gefundenen Clique sowie die Laufzeit des Algorithmus
- Maximierung der Größe der Clique und eine polynomiell beschränkte Laufzeit erhält man bei

$$t = \Theta\left(\frac{\log n}{\log\log n}\right)$$

- Clique C hat in diesem Fall die Kardinalität

$$|C| = \Omega\left(\left(\frac{\log n}{\log\log n}\right)^2\right)$$

- Laufzeit polynomiell in n, wenn $\binom{2kt}{t}$ polynomiell in n
- Wahl des Parameters t beeinflußt Größe der gefundenen Clique sowie die Laufzeit des Algorithmus
- Maximierung der Größe der Clique und eine polynomiell beschränkte Laufzeit erhält man bei

$$t = \Theta\left(\frac{\log n}{\log\log n}\right)$$

- Clique C hat in diesem Fall die Kardinalität

$$|C| = \Omega\left(\left(\frac{\log n}{\log\log n}\right)^2\right)$$

Eine Approximationsgüte von $O(n(\log\log n)^2/(\log n)^3$

Graph $G = (V, E)$ mit einer Clique der Kardinalität n/k

1. $k \geq (\log n)^3$
   - durch Ausgabe eines beliebigen Knotens $v \in V$ erreicht man eine Approximationsgüte von $O(n/(\log n)^3)$
2. $k \leq \log n/(2\log\log n)$
   - wende Algorithmus *IndependentSetRemoval* an
   - Approximationsgüte von $O(n\log\log n/(\log n)^3)$
3. $\log n/(2\log\log n) < k < (\log n)^3$
   - der Algorithmus von Feige erreicht hier nur eine Approximationsgüte von $O(n(\log\log n)^3/(\log n)^3)$
   - für $k > \log n$ ist die Güte schon $O(n(\log\log n)^2/(\log n)^3)$
   - modifiziere Algorithmus von Feige

Graph $G = (V, E)$ mit einer Clique der Kardinalität n/k

1. $k \geq (\log n)^3$
   - durch Ausgabe eines beliebigen Knotens $v \in V$ erreicht man eine Approximationsgüte von $O(n/(\log n)^3)$
2. $k \leq \log n/(2\log\log n)$
   - wende Algorithmus *IndependentSetRemoval* an
   - Approximationsgüte von $O(n\log\log n/(\log n)^3)$
3. $\log n/(2\log\log n) < k < (\log n)^3$
   - der Algorithmus von Feige erreicht hier nur eine Approximationsgüte von $O(n(\log\log n)^3/(\log n)^3)$
   - für $k > \log n$ ist die Güte schon $O(n(\log\log n)^2/(\log n)^3)$
   - modifiziere Algorithmus von Feige

Graph $G = (V, E)$ mit einer Clique der Kardinalität n/k

1. $k \geq (\log n)^3$
   - durch Ausgabe eines beliebigen Knotens $v \in V$ erreicht man eine Approximationsgüte von $O(n/(\log n)^3)$
2. $k \leq \log n/(2 \log\log n)$
   - wende Algorithmus *IndependentSetRemoval* an
   - Approximationsgüte von $O(n \log\log n/(\log n)^3)$
3. $\log n/(2 \log\log n) < k < (\log n)^3$
   - der Algorithmus von Feige erreicht hier nur eine Approximationsgüte von $O(n(\log\log n)^3/(\log n)^3)$
   - für $k > \log n$ ist die Güte schon $O(n(\log\log n)^2/(\log n)^3)$
   - modifiziere Algorithmus von Feige

Graph $G = (V, E)$ mit einer Clique der Kardinalität n/k

1. $k \geq (\log n)^3$
   - durch Ausgabe eines beliebigen Knotens $v \in V$ erreicht man eine Approximationsgüte von $O(n/(\log n)^3)$
2. $k \leq \log n/(2 \log\log n)$
   - wende Algorithmus *IndependentSetRemoval* an
   - Approximationsgüte von $O(n \log\log n/(\log n)^3)$
3. $\log n/(2 \log\log n) < k < (\log n)^3$
   - der Algorithmus von Feige erreicht hier nur eine Approximationsgüte von $O(n(\log\log n)^3/(\log n)^3)$
   - für $k > \log n$ ist die Güte schon $O(n(\log\log n)^2/(\log n)^3)$
   - modifiziere Algorithmus von Feige

Graph $G = (V, E)$ mit einer Clique der Kardinalität n/k

1. $k \geq (\log n)^3$
   - durch Ausgabe eines beliebigen Knotens $v \in V$ erreicht man eine Approximationsgüte von $O(n/(\log n)^3)$
2. $k \leq \log n/(2\log\log n)$
   - wende Algorithmus *IndependentSetRemoval* an
   - Approximationsgüte von $O(n\log\log n/(\log n)^3)$
3. $\log n/(2\log\log n) < k < (\log n)^3$
   - der Algorithmus von Feige erreicht hier nur eine Approximationsgüte von $O(n(\log\log n)^3/(\log n)^3)$
   - für $k > \log n$ ist die Güte schon $O(n(\log\log n)^2/(\log n)^3)$
   - modifiziere Algorithmus von Feige

Modifikation:

- bezeichne eine Knotenmenge S als *gut*, falls S eine Clique ist und $|N(S)| > n_{test} - t$ gilt, wobei n_{test} der größte Wert ist für den gilt
$$n_{test} \leq \frac{\log n_{test}}{2\log\log n_{test}} \cdot \frac{|V''|}{2k}$$
- falls $|V''|/(2k) - t \leq |N(S)| \leq n_{test} - t$ gilt, dann führe den Algorithmus *IndependentSetRemoval* auf $G[S \cup N(S)]$ aus
    - bei einer Clique der Größe $(\log n_{test})^3/(6\log\log n_{test})$ fügen wir diese zu C hinzu und sind fertig
    - sonst ist S eine *schwache* Knotenmenge

Modifikation:

- bezeichne eine Knotenmenge S als *gut*, falls S eine Clique ist und $|N(S)| > n_{test} - t$ gilt, wobei n_{test} der größte Wert ist für den gilt
$$n_{test} \leq \frac{\log n_{test}}{2\log\log n_{test}} \cdot \frac{|V''|}{2k}$$
- falls $|V''|/(2k) - t \leq |N(S)| \leq n_{test} - t$ gilt, dann führe den Algorithmus *IndependentSetRemoval* auf $G[S \cup N(S)]$ aus
  - bei einer Clique der Größe $(\log n_{test})^3/(6\log\log n_{test})$ fügen wir diese zu C hinzu und sind fertig
  - sonst ist S eine *schwache* Knotenmenge

Modifikation:

- bezeichne eine Knotenmenge S als *gut*, falls S eine Clique ist und $|N(S)| > n_{test} - t$ gilt, wobei n_{test} der größte Wert ist für den gilt
$$n_{test} \leq \frac{\log n_{test}}{2\log\log n_{test}} \cdot \frac{|V''|}{2k}$$
- falls $|V''|/(2k) - t \leq |N(S)| \leq n_{test} - t$ gilt, dann führe den Algorithmus *IndependentSetRemoval* auf $G[S \cup N(S)]$ aus
    - bei einer Clique der Größe $(\log n_{test})^3/(6\log\log n_{test})$ fügen wir diese zu C hinzu und sind fertig
    - sonst ist S eine *schwache* Knotenmenge

R. Boppana und M.M. Halldórsson.
Approximating Maximum Independent Sets by Excluding Subgraphs.
BIT, 32(2):180–196, 1992.

U. Feige
Approximating Maximum Clique by Removing Subgraphs.
SIAM Journal on Discrete Math., 18(2):219–225, 2004.

M.M. Halldórsson und J. Radhakrishnan.
A Still Better Performance Guarantee for Approximate Graph Coloring.
Information Processing Letters, 45:19–23, 1993.