Technische Universität Chemnitz

Diplomarbeit

Effiziente Approximation unabhängiger Mengen in Graphen

Matthias Baumgart

Gliederung

1. Einleitung

2. Der Algorithmus von Feige

   1.1. Korrektheit

   1.2. Laufzeit

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

Einleitung

- Boppana und Halldórsson entwickelten basierend auf der Ramsey-Theorie einen Algorithmus zur Bestimmung unabhängiger Mengen.
- Idee: Ein Graph mit $R(k, l)$ Knoten enthält eine Clique der Größe k oder eine unabhängige Menge der Größe l (oder beides).
- Findet der Algorithmus eine große Clique (und nur eine kleine unabhängige Menge), dann entferne diese Clique und suche auf dem Restgraphen weiter nach einer unabhängigen Menge.
- Analoges gilt auch für die Approximation der Cliquenzahl, das heißt statt Cliquen müssen nun unabhängige Mengen entfernt werden.

Idee von U. Feige: Entferne nicht nur unabhängige Mengen, sondern auch Untergraphen, welche nur „wenige“ Kanten enthalten.

Definition 1 *Sei $G = (V, E)$ ein Graph mit einer Clique C der Kardinalität $|C| \geq n/k$. Eine Knotenmenge S in G heißt schwach, wenn der auf S induzierte Untergraph von G keine Clique C_S der Kardinalität $|C_S| \geq |S|/(2k)$ enthält.*

Satz 2 *Sei $G = (V, E)$ ein Graph mit einer Clique C der Kardinalität $|C| \geq n/k$. Seien $S_1, \ldots, S_l$ beliebige disjunkte schwache Knotenmengen von G. Sei $G' = (V', E')$ der auf $V \setminus \{S_1 \cup \ldots \cup S_l\}$ induzierte Untergraph von G. Die Anzahl Knoten in G' ist $|V'| \geq n/(2k)$. Außerdem enthält G' eine Clique C' der Kardinalität $|C'| \geq |V'|/k$.*

Beweis:

- Vereinigung $S = \bigcup_{i=1}^{l} S_i$ ist schwache Knotenmenge auf höchstens n Knoten und hat folglich keine Clique der Größe $n/(2k)$
- Voraussetzung: G eine Clique C der Kardinalität $|C| \geq n/k$

$\Longrightarrow$ mindestens $n/(2k)$ Knoten der Clique C müssen in G' sein

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

$\Longrightarrow$ dann befinden sich jedoch mindestens

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

Knoten der Clique C in der Knotenmenge S

$\Longrightarrow$ 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.
- Jede Phase kann aus mehreren einzelnen Iterationen bestehen.

- 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

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

   wurde gefunden.

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

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

1. Falls $|V''| < 6kt$, dann beende Phase und gib C aus.
2. Partitioniere V'' in disjunkte Knotenmengen P_i der Größe $2kt$.
3. Betrachte alle möglichen t-elementigen Teilmengen S von P_i.
4. Sei $N(S)$ alle Knoten in $V'' \setminus S$, die mit jedem Knoten aus S in G'' verbunden sind. Bezeichne S als *gut*, falls S eine Clique ist und $|N(S)| \geq |V''|/(2k) - t$ erfüllt.
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.

Der Algorithmus von Feige – Korrektheit

Satz 3 *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:* $V'' = P_1 \cup \ldots \cup P_l$ enthält Clique C mit

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

$\Longrightarrow$ es gibt ein P_i, dass t Knoten von C enthält (Schubfachprinzip)

$\Longrightarrow$ es gibt ein S mit der Eigenschaft *gut* $\square$

Satz 4 *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 $G' = (V', E')$ und damit auch in $G = (V, E)$ bilden.

Beweis:

- die Menge C ist offensichtlich eine Clique
- jede – außer der letzten – Iteration fügt t Knoten zu C hinzu
- Bestimmung einer unteren 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

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

- die Anzahl Knoten der $i+1$'ten Iteration ist mindestens

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

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

$$\begin{aligned} & \frac{|V'|}{(3k)^x} &<& \quad 6kt \\ \iff & \frac{|V'|}{6kt} &<& \quad (3k)^x \\ \iff & \log_{3k}\frac{|V'|}{6kt} &<& \quad x \end{aligned}$$

- es werden mindestens $\log_{3k}(|V'|/(6kt))$ Iterationen durchlaufen, in denen jeweils t Knoten zu C hinzufügen werden, folglich gilt:

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

Der Algorithmus von Feige – Laufzeit

- 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 $O(n(\log\log n)^2/(\log n)^3)$

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

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

Fall 1: $k \geq (\log n)^3$

- durch Ausgabe eines beliebigen Knotens $v \in V$ erreicht man eine Approximationsgüte von $O(n/(\log n)^3)$

Fall 2: $k \leq \log n/(2\log\log n)$

- Algorithmus *IndependentSetRemoval* von Boppana und Halldórsson

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

Fall 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 $O(n(\log\log n)^2/(\log n)^3)$

- müssen also einen Faktor von $\Omega(\log\log n)$ einsparen

$\Longrightarrow$ modifiziere Algorithmus von Feige

$\Longrightarrow$ ändere Definition einer *guten* 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

  $\Longrightarrow$ erhält man eine Clique der Größe $(\log n_{test})^3/(6\log\log n_{test})$, dann fügt man diese Clique zu C hinzu und gibt C aus

  $\Longrightarrow$ sonst ist S eine *schwache* Knotenmenge

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

Literatur

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

[2] U. Feige. *Approximating Clique by Removing Subgraphs*. Manuskript, erscheint in SIAM Journal on Discrete Mathematics, 2002.

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