Seminar
Exakte Algorithmen für NP-schwere Probleme
Handout vom 05.12.2005

Haobo Song

9. Januar 2006

1 Kontext

Bei dieser Arbeit studieren wir zwei Algorithmen für MAX-2-SAT Problem, die auch für MAX-CUT und MAX-XSAT einsetzbar sind. Diese Probleme sind wichtige MAX-SNP-hard Probleme [G.Woeginger. Exact algorithms for NP-hard Problems:A survey. Springer,2003]. Während der zweite Algorithmus eine time complexity $O^*(2^{m/5.217})$ mit polynomial space besitzt, mit dem wir gerade in der letzter Diskussion angefangen haben, hat der erste auch relativ einfachere Algorithmus $O^*(2^{m/5})$ time complexity.

MAX-SAT ist eine Generalisierung von SAT. Gegeben eine Formel in CNF, so entsteht das Problem, welches complete für nicht nur NP, sondern auch MAX-SNP ist, um eine maximale Anzahl von gleichzeitig erfüllten Klausen zu bestimmen. Im Fall von MAX-2-SAT gibt es in jeder Klause höchstens zwei Literale, und dieses Problem kann man in Graphen umwandeln und effizient lösen.
F ist eine Formula in 2-CNF,

$$\text{die Variablen in F} \longrightarrow \text{Knoten in V}$$

Der entstehende Graph ist $G_F = (V, E)$ wenn

E={{x,y} | x und y treten zusammen in einer Klause auf, x,y sind verschieden}

Es macht keinen Unterschied, in wie vielen Klausen zwei Variablen erscheinen oder ob eine Variable negiert oder nicht ist.
Wir haben auch in der vorletzten Diskussion über treewidth gesprochen. Die Probleme in Graphen sind so zu lösen, dass man die Graphen in Bäumen mit kleinster treewidth zerlegt, und es existieren sehr viele schöne Algorithmen, die nur für Bäume anwendbar sind!

ERSTER ALGORITHMUS

`Ein Algorithmus mit nur einem Reduktionsregel.`

Zuerst haben wir eine sehr einfache Reduktionsregel gezeigt, die die Knoten mit Grad eins von dem Graph entfernt. Dann sind die Begriffe:leg, hot dog, 4-spider, 3-spider und Potenzial eingeführt worden. Am Ende diser Diskussion zeigt es, wie man einen Graph um nicht mehr als m/5 Knoten in einem speziellen Fall von einem partial 2 tree reduziert, so dass der resultierende Baum treewidth höchstens m/5+2 hat.
Definition 1.
R(G) ist der Graph nach der wiederholender Löschung von Knoten mit Grad 1.
Lemma 1.
$D = \{v_1, v_2, ..., v_k\}$ ist eine Menge von Knoten von V, dann
$R(G - D) = R(R(...(R(R(G - v_1) - v_2) - v_3)... - v_{k-1}) - v_k)$
Lemma 2.
$tw(G) = tw(R(G))$
Definition 2.
Ein leg ist ein Pfad, der mindestens eine Länge eins zwischen zwei möglichen identischen Knoten, wenn alle Knoten ausser den beiden einen Grad von zwei haben.
Ein hot dog Graph besteht aus Knoten $v_1, ...v_k$, so dass v_i und v_{i+1} mit beliebig viele legs verbunden sind. v_k und v_1 können auch auf dieser Weise verbunden werden.
Definition 3.
Ein 4-spider ist ein Subgraph, das aus einem head $h \in V$ mit Grad vier, drei oder vier verschiedene feet $u_1, ..., u_l \in V \backslash \{h\}$ von mindestens Grad drei, und vier verschiedene legs binden Kopf und feet besteht.
Ein 3-spider ist ähnlich definiert, ein head mit Grad 3 und exakt drei verschiedene feet, die werden mit drei verschiedenen legs verbunden.
Lemma 3.
$G = (V, E)$ ist ein zusammenhängender Graph, wessen Knoten alle einen Grad zwischen zwei und vier haben. G ist ein hot dog Graph iff er kein 3- oder 4-spider enthält.

2 Definition 4.

Sei G=(V,E) ein Graph, $v \in V$*, und*
$deg_3(v) = |\{v \in V | u \text{ und } v \text{ sind verbindet mit einem leg, und } deg(u) \geq 3\}|$.
Potential Funktion $\varphi : V \to N$ *und* $\Psi : \varsigma \to N$ *wie folgende:*

$$\varphi(v) = \begin{cases} 0 & \text{if } deg(v) \leq 2 \\ 0 & \text{if } deg(v)=3 \text{ und } deg_3(v) = 1 \\ 5/4 & \text{if } deg(v)=3 \text{ und } deg_3(v) > 1 \\ 2 & \text{if } deg(v) \geq 4 \end{cases}$$

und $\Psi(G) = \Sigma_{v \in V} \varphi(v)$.

Die Potenziale, die hier zugeordnet sind, sind nicht beliebig angegeben. Diese

Funktion wird benutzt, um später bei der Reduktion die Anzahl der verschwundenen Kanten mit Potenzial zu vergleichen.

3 Lemma 4.

Sei $G = (V, E)$ ein Graph. Dann $\Psi(G) \leq |E|$

Der Beweis ist direkt und klar.

$|E| = \frac{1}{2} \sum_{v \in V} deg(v)$
$= \frac{1}{2} \sum_{i=1}^{|V|-1} \sum_{v \in V, deg(v)=i} i$
$\geq \frac{1}{2} (\sum_{v \in V, deg(v)=3} 3 + \sum_{v \in V, deg(v) \geq 4} 4)$
$\geq \sum_{v \in V, deg(v)=3} \frac{5}{4} + \sum_{v \in V, deg(v) \geq 4} 2 \geq \Psi(G). \diamondsuit$

4 Lemma 5.

Sei G=(V,E) ein Graph, und alle Knoten aus V haben einen Grad zwischen zwei und vier. Wenn G einen 4-spider mit head h enthält, dann

$$\Psi(R(G - h)) \leq \Psi(G) - 5$$

Der Beweis läßt sich in ein paar Fällen durchdiskutieren:
Sei S ist der 4-spider mit head h.
Fall 1: S hat vier verschiedene feet $u_1, ..., u_4$ mit $3 \leq deg(u_i) \leq 4$.
Entfernen von dem head verursacht eine Potenzial Senkung um 2 beim head und mindestens um 3/4 bei jedem foot, also insgesamt mindestens 5.
Fall 2: S hat nur drei verschiedene feet u_1, u_2, u_3, zwei legs sind mit u_1, und die andere zwei jeweils mit u_2, u_3 verbunden.
Wenn $deg(u_1) = 4$ ist, Entfernen von dem head verursacht eine Potenzial Senkung um 2 jeweils beim head und bei u_1, mindestens um 3/4 jeweils bei u_2, u_3, also insgesamt mehr als 5.
Wenn $deg(u_1) = 3$ ist, u_1 hat einen leg, der weder mit dem head noch mit anderem foot verbunden ist und mit z endet, Entfernen von dem head verursacht eine Potenzial Senkung um 2 beim head, um 5/4 bei u_1, und mindestens um 3/4 jeweils bei z, u_2, u_3, insgesamt wieder mehr als 5.
Wenn z, u_2, u_3 nicht alle verschieden sind, z.B. $z = u_3 \neq u_2$, dann ist die Senkung des Potenzials um 2 bei head, um 5/4 bei u_1, mindestens um 5/4 bei $z = u_3$ und mindestens um 3/4 bei u_2, es ist insgesamt mehr als 5.

5 Lemma 6.

Sei G=(V,E) ein Graph, und alle Knoten aus V haben einen Grad zwischen zwei und vier. Wenn G einen 3-spider mit head h aber *keinen 4-spider enthält, dann*

$$\Psi(R(G-h)) \leq \Psi(G) - 5$$

Der Beweis ist sehr ähnlich wie bei Lemma 5, allerdings wird es hier nicht ausführlich besprochen.

6 Theorem 1.

Sei G=(V,E) ein Graph. Es existiert eine Menge $D \subseteq V$, so dass R(G-D) ein hot dog Graph ist und

$$|D| \leq |E|/5$$

Seien A,B,D drei leere Knotenmengen.
Wir entfernen zuerst die Knoten mit Grad am mindesten 5 und fügen die in A ein. Führen wir die Reduktion durch, dann haben wir einen Graph $G_A = R(G-A)$.
Anschließend entfernen wir die heads von den 4-spiders und führen die Reduktion durch, dann das Gleiche bei 3-spiders. Am Ende fügen wir die heads in B ein und haben wir $G_{AB} = R(G_A - B)$
Setze $D = A \cup B$. Offensichtlich ist R(G-D) ein hot dog Graph.
Nach Lemma 4,5 und 6:
$0 \leq \Psi(G_{AB}) \leq \Psi(G_A) - 5|B| \leq |E_A| - 5|B| \leq |E| - 5|A| - 5|B|$
$\Rightarrow |D| \leq |E|/5$

7 Theorem 2.

Das treewidth von dem Graph G=(V,E) ist am höchsten $|E|/5 + 2$.

Beweis:
Sei D die Menge von Theorem 1. R-Reduktion, so wird das treewidth nicht verändert, und G-D enthält einen Zyklus. G-D ist ein hot dog Graph und ein hot dog Graph konstituiert einen speziellen Fall von series-parallel Graphen, wessen treewidth am höchsten zwei ist. Im Spiel Cops & Räuber für Bestimmung von treewith haben wir gesehen, wie das Spiel läuft. Für jeden head von spiders oder für jeden Knoten mit Grad mindestens 5 braucht es insgesamt einen Cop.

$tw(G) \leq |D| + 2 = |E|/5 + 2$

8 Eine zweite Regel vereinfacht den Algorithmus

Mit dem ersten Algorithmus haben wir einen zusammenhängenden Graph auf einen hot dog Graph reduziert. Ein hot dog Graph hat i.d.R. viele legs, also viele Knoten mit Grad zwei. In dieser Situation ist eine zweite Reduktionsregel eingeführt worden: ersetze den Pfad (u,v,w) mit deg(v)=2 durch den Pfad (u,w), und wir nennen die Operation contracting v. Dann wird ein hot dog Graph in einen trivialen Graph ohne einzige Kante reduziert. Letztlich damit der Algorithmus auch für das Problem MAX-CUT anwendbar ist, wird dies in MAX-2-SAT transformiert, in dem man die Kante { x,y} durch zwei Klausen { x,y } und { $\bar{x}$, $\bar{y}$ } ersetzt. Die Anzahl von Kanten wird die Anzahl von Klausentypen.

9 Definition 5.

Sei G=(V,E) ein Graph und v∈V. $R'(G)$ ist der Graph reduziert vom G, in dem wir wiederholend die Grad eins Knoten entfernen und Grad zwei Knoten contracten bis keine solche Operation möglich ist. Wenn eine contract zu eine Doppelkante führt, wird eine davon gelöscht. Wir definieren auch:

$$R'_v(G) := R'(G - v)$$

Neue Reduktionsregel beinhaltet die alte.

10 Lemma 7.

Sei G=(V,E) ein Graph mit Minimum Grad drei und Maximum Grad vier. Wenn v∈V und deg(v)=4, dann

$$\Psi(R'_v(G)) \leq \Psi(G) - 5$$

Beweis:
Eine Potenzial Senkung um 2 bei v und jeweils eine Senkung um mindestens 3/4 bei jedem seinen Nachbar ergibt, insgesamt mehr als 5.

11 Lemma 8.

Sei G=(V,E) ein 3-regular Graph. Es gilt für jedes v∈V

$$\Psi(R'_v(G)) \leq \Psi(G) - 5$$

Ein 3-regular Graph ist ein Graph, in dem alle Knoten Grad drei haben. Jeder Knoten hat ein Potenzial von 5/4, und die Potenzial Senkung beim Entfernen von jedem Knoten ergibt genau 5.

12 Algorithmus

Algorithm A
Input: A graph G=(V,E)
Output: $D \in V, |D| \leq |E|/5$, such that $R'(G-D)$ has no edges
$D \leftarrow \emptyset$;
while there is a node v with deg(v)$\geq$ 3 do
choose a node v with maximum degree;
$D \leftarrow D \cup \{v\}; G \leftarrow R'_v(G)$ **od**;
return D

13 Theorem 3.

Der reduzierte Graph $R'(G-D)$ von Algorithmus A hat keine Kanten und $|D| \leq m/5 = |E|/5$.

Beweis:
Der Algorithmus reduziert zuerst die Knoten mit Grad mindestens 5 ($R'_v(G)$), und verursacht eine Verlust von mindestens 5 Kanten für jeden solchen Knoten. Dann reduziert er die Knoten mit Grad 4 und weiter mit Grad 3. Nach Lemma 7 und Lemma 8 wird das Potenzial des Graphes um mindestens 5 für jeden solchen Knoten vermindert. Mit der Reduktionsregel gibt der Algorithmus am Ende einen Graph ohne einige Kante! Und insgesamt $|D| \leq m/5 = |E|/5$.

14 Fazit

Bis jetzt haben wir über zwei Reduktionsregeln diskutiert. Beide Regeln basieren auf Graphen Potenzial, welches bei Graphenreduktion den Kantenverlust verglichen und eine Obergrenze von treewidth ($|E|/5 + 2$) bestimmt wird. d.h. der Algorithmus für MAX-2-SAT (umgewandelt in Graphen) hat time complexity $O^*(2^{m/5})$. Später werden wir einen modifizierten Algorithmus mit niedrigere time complexity $O^*(2^{m/5.217})$ sehen.