

Näherungsalgorithmen (Approximationsalgorithmen)

WiSe 2012/13 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

30. Januar 2013

Näherungsalgorithmen

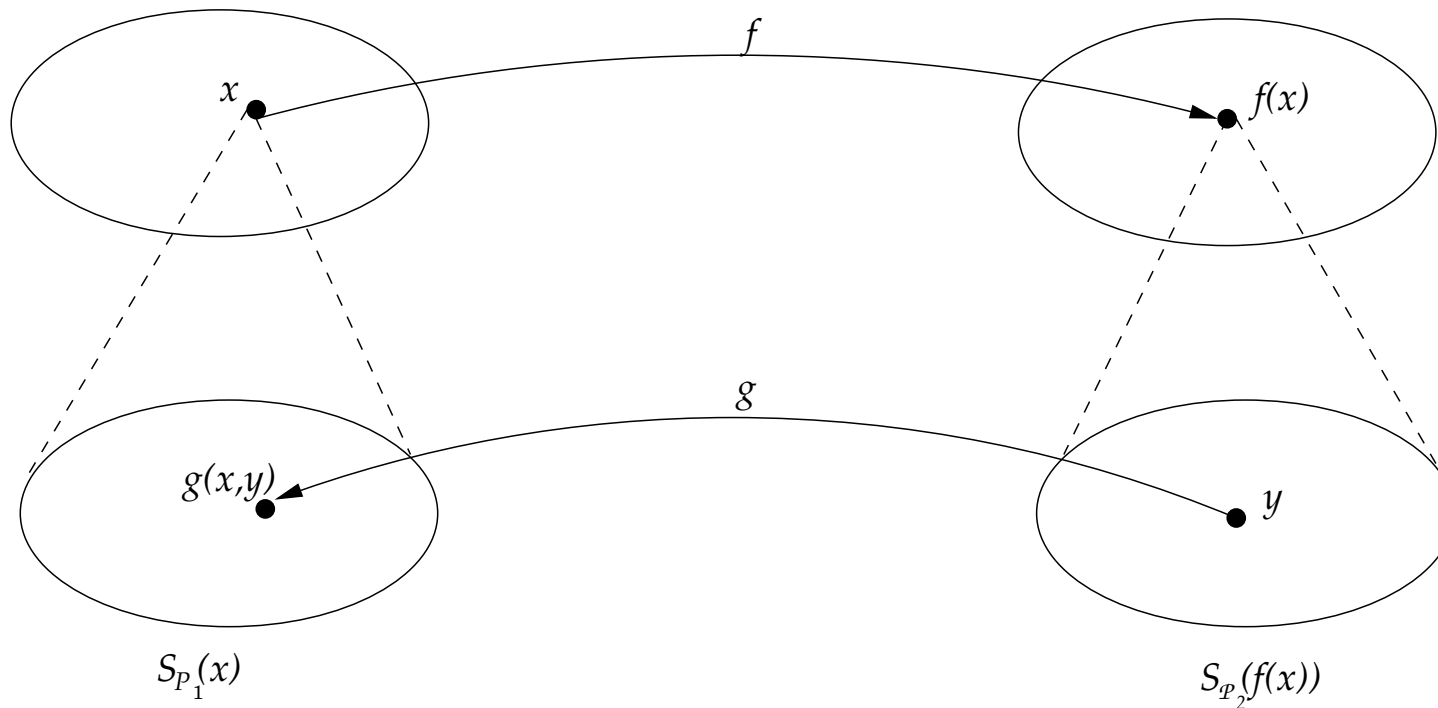
Gesamtübersicht

- Organisatorisches
- Einführung / Motivation
- Grundtechniken für Näherungsalgorithmen
- Approximationsklassen (Approximationstheorie)

Approximationstheorie

- Absolute Approximation
- Relative Approximation: die Klasse APX
- Polynomzeit-Approximationsschemata PTAS
- Zwischen APX und NPO
- Zwischen PTAS und APX
- Approximationsklassen und Reduktionen

Schema einer AP-Reduktion



Approximationserhaltene Reduktionen

Betrachte $\mathcal{P}_1, \mathcal{P}_2 \in \text{NPO}$, \mathcal{P}_1 heißt *näherungserhaltend* auf \mathcal{P}_2 *reduzierbar*, kurz \mathcal{P}_1 ist *AP-reduzierbar* (AP bedeutet ausgeschrieben „approximation preserving“) auf \mathcal{P}_2 , in Zeichen $\mathcal{P}_1 \leq_{\text{AP}} \mathcal{P}_2$, wenn es zwei Abbildungen f, g gibt und eine Konstante $\alpha \geq 1$ derart, dass folgende Bedingungen erfüllt sind:

1. $\forall x \in I_{\mathcal{P}_1} \forall r \in \mathbb{Q} \cap (1, \infty) : f(x, r) \in I_{\mathcal{P}_2}$.
2. $\forall x \in I_{\mathcal{P}_1} \forall r \in \mathbb{Q} \cap (1, \infty) : S_{\mathcal{P}_1}(x) \neq \emptyset \rightarrow S_{\mathcal{P}_2}(x)(f(x, r)) \neq \emptyset$.
3. $\forall x \in I_{\mathcal{P}_1} \forall r \in \mathbb{Q} \cap (1, \infty) \forall y \in S_{\mathcal{P}_2}(f(x, r)) : g(x, y, r) \in S_{\mathcal{P}_1}(x)$.
4. f, g sind durch Algorithmen $\mathcal{A}_f, \mathcal{A}_g$ berechenbar, deren Laufzeit polynomiell ist für jedes feste $r \in \mathbb{Q} \cap (1, \infty)$.
5. $\forall x \in I_{\mathcal{P}_1} \forall r \in \mathbb{Q} \cap (1, \infty) \forall y \in S_{\mathcal{P}_2}(f(x, r)) :$

$$R_{\mathcal{P}_2}(f(x, r), y) \leq r \rightarrow R_{\mathcal{P}_1}(x, g(x, y, r)) \leq 1 + \alpha(r - 1)$$

Ein einfaches **Beispiel** für eine AP-Reduktion liefern MAXCLIQUE und MAX-IS durch Übergang auf den Komplementgraphen; die Clique wird so zur unabhängigen Menge.

Satz: Betrachte $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3 \in \text{NPO}$.

1. Gilt $\mathcal{P}_1 \leq_{\text{AP}} \mathcal{P}_2$ und $\mathcal{P}_2 \leq_{\text{AP}} \mathcal{P}_3$, so auch $\mathcal{P}_1 \leq_{\text{AP}} \mathcal{P}_3$ (Transitivität)
2. Gilt $\mathcal{P}_1 \leq_{\text{AP}} \mathcal{P}_2$ und $\mathcal{P}_2 \in \text{APX}$, so folgt $\mathcal{P}_1 \in \text{APX}$.
3. Gilt $\mathcal{P}_1 \leq_{\text{AP}} \mathcal{P}_2$ und $\mathcal{P}_2 \in \text{PTAS}$, so folgt $\mathcal{P}_1 \in \text{PTAS}$.

Wegen dem Satz ist die folgende Definition sinnvoll: Es sei $C \subseteq \text{NPO}$.
Ein Problem $\mathcal{P} [\in \text{NPO}]$ heißt *C-hart*, wenn für jedes $\mathcal{P}' \in C$ gilt:

$$\mathcal{P}' \leq_{\text{AP}} \mathcal{P}.$$

Ein C-hartes Problem heißt *C-vollständig*, wenn es in C liegt.

In der Literatur werden verschiedene Reduktionsbegriffe für Approximationsprobleme betrachtet. Entsprechend gibt es auch verschiedene Härte- und Vollständigkeitsbegriffe. Näheres dazu im Buch von Ausiello et al., Kapitel 8. Im Folgenden werden wir noch einige konkrete AP-Vollständigkeitsbegriffe diskutieren. Dadurch wird auch der Umgang mit AP-Reduktionen geübt.

EXP-APX-Vollständigkeit

Der Unterschied zwischen EXP-APX und NPO besteht darin, dass für NPO-Probleme \mathcal{P} die Menge $S_{\mathcal{P}}(x)$ für jede Instanz x NP-hart sein darf (und in den vorigen Beispielen für NPO-vollständige Probleme auch war).

Dadurch, dass man „künstlich“ die Menge der zulässigen Lösungen trivialisiert, ohne die optimalen Lösungen zu verändern, kann man sich Probleme definieren, für die man EXP-APX-Vollständigkeit beweisen kann.

Der Nachteil der hier betrachteten NPO- bzw. der daraus abgeleiteten EXP-APX-vollständigen Probleme ist ihre „Künstlichkeit“.

Wir werden abschließend APX-vollständige „natürliche“ Probleme kennen lernen.

APX-Vollständigkeit

Mit Hilfe der uns hier nicht zur Verfügung stehenden PCP-Kennzeichnung von NP lässt sich zeigen (s. das Buch von Ausiello et al.).

Mitteilung: MAX 3-SAT ist vollständig für die Klasse der Maximierungsprobleme in APX.

Daraus folgt zusammen mit dem folgenden Satz, dass MAX 3-SAT tatsächlich APX-vollständig ist.

Satz: Zu jedem Minimierungsproblem $\mathcal{P} \in \text{APX}$ gibt es ein Maximierungsproblem $\mathcal{P}' \in \text{APX}$ mit $\mathcal{P} \leq_{\text{AP}} \mathcal{P}'$.

Beweis: (Skizze) Es sei \mathcal{A} eine r -Approximation von \mathcal{P} . Definiere \mathcal{P}' wie \mathcal{P} bis auf die Maßfunktion, die wie folgt definiert ist:

$$m_{\mathcal{P}'}(x, y) = \begin{cases} ((\lceil r \rceil + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)) - (\lceil r \rceil \cdot m_{\mathcal{P}}(x, y))), & m_{\mathcal{P}}(x, y) \leq m_{\mathcal{P}}(x, \mathcal{A}(x)) \\ m_{\mathcal{P}}(x, \mathcal{A}(x)), & \text{sonst} \end{cases}$$

$$\leadsto m_{\mathcal{P}}(x, \mathcal{A}(x)) \leq m_{\mathcal{P}'}^*(x) \leq ((\lceil r \rceil + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)))$$

→ \mathcal{A} ist $(\lceil r \rceil + 1)$ -Approximation für das Maximierungsproblem \mathcal{P}' .

Die AP-Reduktion ist wie folgt definiert:

1. Für jede Instanz x setze $f(x) = x$.
2. Für jede Instanz x und für jede Lösung y von $f(x)$:

$$g(x, y) = \begin{cases} y, & m_{\mathcal{P}}(x, y) \leq m_{\mathcal{P}}(x, \mathcal{A}(x)) \\ \mathcal{A}(x), & \text{sonst} \end{cases}$$

3. $\alpha = \lceil r \rceil + 1$.

Um nachzuweisen, dass diese Reduktion wirklich eine AP-Reduktion ist, betrachte eine Lösung y mit $R_{\mathcal{P}'}(x, y) = m_{\mathcal{P}'}^*(x)/m_{\mathcal{P}'}(x, y) \leq r'$.

Zu zeigen ist:

$$R_{\mathcal{P}}(x, g(x, y)) \leq 1 + \alpha(r' - 1)$$

Dies kann durch Unterscheidung der Fälle $m_{\mathcal{P}}(x, y) \leq m_{\mathcal{P}}(x, \mathcal{A}(x))$ bzw. $m_{\mathcal{P}}(x, y) > m_{\mathcal{P}}(x, \mathcal{A}(x))$ durch einige Abschätzungen hergeleitet werden. □

Einzelheiten...

Fall 1: $m_{\mathcal{P}}(x, y) \leq m_{\mathcal{P}}(x, \mathcal{A}(x))$

$$\begin{aligned}
 m_{\mathcal{P}}(x, y) &= \frac{([\![r]\!] + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)) - m_{\mathcal{P}'}(x, y)}{[\![r]\!]} && \leq \frac{([\![r]\!] + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)) - m_{\mathcal{P}'}^*(x)/r'}{[\![r]\!]} \\
 &= \frac{([\![r]\!] + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)) - m_{\mathcal{P}'}^*(x)/(1 + (r' - 1))}{[\![r]\!]} && \leq \frac{([\![r]\!] + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)) - (1 - (r' - 1))m_{\mathcal{P}'}^*(x)}{[\![r]\!]} \\
 &= \frac{([\![r]\!] + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)) + m_{\mathcal{P}'}^*(x) + (r' - 1)m_{\mathcal{P}'}^*(x)}{[\![r]\!]} && \leq m_{\mathcal{P}}^*(x) + \frac{r' - 1}{[\![r]\!]} m_{\mathcal{P}'}^*(x) \\
 &\leq m_{\mathcal{P}}^*(x) + \frac{r' - 1}{[\![r]\!]} ([\![r]\!] + 1)m_{\mathcal{P}}(x, \mathcal{A}(x)) && \leq m_{\mathcal{P}}^*(x) + (r' - 1)([\![r]\!] + 1) \frac{m_{\mathcal{P}}(x, \mathcal{A}(x))}{r} \\
 &\leq (1 + \alpha(r' - 1))m_{\mathcal{P}}^*(x)
 \end{aligned}$$

$$\rightsquigarrow R_{\mathcal{P}}(x, g(x, y)) = R_{\mathcal{P}}(x, y) \leq 1 + \alpha(r' - 1).$$

Fall 2: $m_{\mathcal{P}}(x, y) > m_{\mathcal{P}}(x, \mathcal{A}(x))$:

$$R_{\mathcal{P}}(x, g(x, y)) = R_{\mathcal{P}}(x, \mathcal{A}(x)) = R_{\mathcal{P}'}(x, y) \leq r' \leq 1 + \alpha(r' - 1).$$

L-Reduktionen—Motivation

Wie kann man nun APX-Vollständigkeit für „natürliche“ Probleme beweisen? Hierbei ist es manchmal hilfreich, sich „alte“ bekannte NP-Vollständigkeitsbeweise für die betreffende Entscheidungsprobleme anzusehen.

So wird für das Kantenschnittproblem üblicherweise (siehe das Buch von Ausiello et al.) der Weg von 3-SAT über MAX2SAT_D , NOT-ALL-EQUAL-3SAT zu MAXCUT_D gegangen.

Hierbei ist es häufig hilfreich, eine andere als die bislang diskutierte AP-Reduktion zu betrachten, da sich dabei NP-Reduktionen leichter übertragen:

L-Reduktionen–Definition

Es seien \mathcal{P}_1 und \mathcal{P}_2 zwei NPO-Probleme. \mathcal{P}_1 heißt *L-reduzierbar* auf \mathcal{P}_2 , i. Z. $\mathcal{P}_1 \leq_L \mathcal{P}_2$, wenn zwei Funktionen f, g und zwei positiven Konstanten β, γ existieren, sodass:

1. $\forall x \in I_{\mathcal{P}_1} : f(x) \in I_{\mathcal{P}_2}$ ist in Polynomzeit berechenbar.
2. $\forall x \in I_{\mathcal{P}_1} : S_{\mathcal{P}_1}(x) \neq \emptyset \rightarrow S_{\mathcal{P}_2}(f(x)) \neq \emptyset$.
3. $\forall x \in I_{\mathcal{P}_1} \forall y \in S_{\mathcal{P}_2}(f(x)) : g(x, y) \in S_{\mathcal{P}_1}(x)$ ist in Polynomzeit berechenbar.
4. $\forall x \in I_{\mathcal{P}_1} : m_{\mathcal{P}_2}^*(f(x)) \leq \beta m_{\mathcal{P}_1}^*(x)$.
5. $\forall x \in I_{\mathcal{P}_1} \forall y \in S_{\mathcal{P}_2}(f(x)) :$
$$|m_{\mathcal{P}_1}^*(x) - m_{\mathcal{P}_1}(x, g(x, y))| \leq \gamma \cdot |m_{\mathcal{P}_2}^*(f(x)) - m_{\mathcal{P}_2}(f(x), y)|$$

(f, g, β, γ) heißt auch *L-Reduktion* von \mathcal{P}_1 auf \mathcal{P}_2 .

Mitteilung: Es seien $\mathcal{P}_1, \mathcal{P}_2 \in \text{NPO}$ mit $\mathcal{P}_1 \leq_L \mathcal{P}_2$. $\mathcal{P}_1 \in \text{APX} \rightarrow \mathcal{P}_1 \leq_{\text{AP}} \mathcal{P}_2$.

Als eine Anwendung und erstes Glied obiger Reduktionskette zeigen wir:

Satz: $\text{MAX3SAT} \leq_L \text{MAX2SAT}$.

Beweis: Wir geben einer L-Reduktion $(f, g, 13, 1)$ an.

Es sei φ eine Instanz von MAX3SAT mit m Klauseln $a_i \vee b_i \vee c_i, 1 \leq i \leq m$, wobei a_i, b_i und c_i Literale sind.

Jeder dieser Klauseln ordnen wir die folgenden zehn neuen Klauseln zu (jede hat höchstens zwei Literale!):

$$(*) \quad \begin{array}{cccccccccc} a_i, & b_i, & c_i, & d_i, & \bar{a}_i \vee \bar{b}_i, & \bar{a}_i \vee \bar{c}_i, & \bar{b}_i \vee \bar{c}_i, & a_i \vee \bar{d}_i, & b_i \vee \bar{d}_i, & c_i \vee \bar{d}_i \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array}$$

Dabei ist d_i eine neue Variable.

$f(\varphi) = \varphi'$ sei die sich so ergebene Instanz von MAX2SAT.

$$(*) \quad a_i, \quad b_i, \quad c_i, \quad d_i, \quad \overline{a_i} \vee \overline{b_i}, \quad \overline{a_i} \vee \overline{c_i}, \quad \overline{b_i} \vee \overline{c_i}, \quad a_i \vee \overline{d_i}, \quad b_i \vee \overline{d_i}, \quad c_i \vee \overline{d_i}$$

Die folgende Tabelle zeigt, dass jede Belegung höchstens 7 der 10 Klauseln einer Gruppe der Form (*) erfüllt (symmetrische Fälle sind nicht aufgeführt)

a_i	b_i	c_i	d_i	1	2	3	4	5	6	7	8	9	10	Σ
0	0	0	0					1	1	1	1	1	1	6
0	0	0	1				1	1	1	1				4
0	0	1	0			1		1	1	1	1	1	1	7←
0	0	1	1			1	1	1	1	1			1	6
0	1	1	0		1	1		1	1		1	1	1	7←
0	1	1	1		1	1	1	1	1			1	1	7←
1	1	1	0	1	1	1					1	1	1	6
1	1	1	1	1	1	1	1				1	1	1	7←

Für jede erfüllende Belegung der ursprünglichen 3-SAT-Klausel $a_i \vee b_i \vee c_i$ gibt es eine Belegung von (*), die sieben Klauseln erfüllt; diese sind durch Pfeile in der Tabelle markiert. Wegen der ersten Zeile der Tabelle gibt es im Falle einer $a_i \vee b_i \vee c_i$ nicht-erfüllenden Belegung eine Belegung von d_i , die 6 Klauseln von (*) erfüllt. Somit können wir schließen:

$$m^*(\varphi') = 6m + m^*(\varphi) \leq 12m^*(\varphi) + m^*(\varphi) = 13m^*(\varphi).$$

Die Ungleichung gilt, da zu einer KNF-Formel stets eine Belegung existiert, die wenigstens die Hälfte der Klausel erfüllt (siehe Analyse der Greedy-Heuristik für MAXSAT in früherer Vorlesung).

Ist τ' eine Belegung der Variablen von φ' , so bezeichne $\tau = g(\varphi, \tau')$ die Restriktion von τ' auf die in φ vorkommenden Variablen. Durch Betrachten obiger Tabelle ergibt sich

$$m(\varphi, \tau) \geq m(\varphi', \tau') - 6m.$$

Damit haben wir:

$$\begin{aligned} |m^*(\varphi) - m(\varphi, g(\varphi, \tau'))| &= m^*(\varphi) - m(\varphi, \tau) \\ &= m^*(\varphi') - 6m - m(\varphi, \tau) \\ &\leq m^*(\varphi') - 6m - m(\varphi', \tau') + 6m \\ &= 1 \cdot |m^*(\varphi') - m(\varphi', \tau')|. \square \end{aligned}$$

Facetten der Härte für Approximationen

Diese Darstellung folgt:

D.S. Johnson: The NP-Completeness Column: The Many Limits on Approximation. ACM Trans. Alg., Band 2, S. 473–489, 2006.

Der “Goldstandard” für Härteresultate:

Ist betrachtetes Optimierungsproblem “leicht” (das hängt von der konkreten Klasse ab, die man betrachtet), so folgt $P\text{TIME} = NP$.

Der “Goldstandard” für Härteresultate lässt Lücken

Beispiel: Dinur und Safra konnten 2002 (ECCC, dann SODA 2005) zeigen:
Falls $P\text{TIME} \neq NP$, so lässt sich das Knotenüberdeckungsproblem nicht besser als 1.36067 approximieren.

Diese Aussage lässt noch eine beträchtliche Lücke, denn das beste bekannte positive Approximationsergebnis stammt von Karakostas (ICALP 2005):

Das Knotenüberdeckungsproblem für n -Knotengraphen lässt sich in der Leistungsgüte $2 - \Theta\left(\frac{1}{\sqrt{\log(n)}}\right)$ approximieren.

Unter schwächeren Annahmen lassen sich manchmal stärkere Aussagen gewinnen.

Intuitive Ähnlichkeit von Problemen

Bekannt: Das Cliquesproblem.

Feige et al. konnte 2006 zeigen (J.ACM Band 43, Seiten 268–292):

Mitteilung: *MAXCLIQUE lässt sich nicht mit einem Faktor $2^{(\log(n))^{1-\epsilon}}$ approximieren für irgendein $\epsilon > 0$, falls nicht $NP \subseteq DTIME(n^{\text{polylog}(n)})$.*

Unter stärkeren Voraussetzungen (Goldstandard) kann man eine schwächere Aussage zeigen:

Mitteilung: Gilt $P \neq NP$, so lässt sich MAXCLIQUE für kein $\epsilon > 0$ auf einen Faktor von $n^{1-\epsilon}$ approximieren.

Gelten ähnliche Schranken für ähnliche Probleme ?

Leider nicht immer...

Intuitive Ähnlichkeit von Problemen

Sehr ähnlich zu dem Problem des Findens großer Cliques scheint doch das Problem zu sein, möglichst große ausgeglichene Bicliquen zu finden.

Ein vollständiger bipartiter Graph $K_{k,\ell}$ heißt auch *Biclique*. Eine Biclique heißt *ausgeglichen*, falls auf beiden Seiten der Bipartition gleichviele Knoten liegen.

Das Problem, möglichst große ausgeglichene Bicliquen in einem (nicht notwendig bipartiten) Graphen zu finden, ist bekanntermaßen *NP*-hart (auch bekannt als BCBS: balanced complete bipartite subgraph).

BCBS-Approximation

Oft begegnen wir Approximationsschranken, die randomisierte Komplexitätsklassen betreffen.

BP steht für “bounded error probabilistic”.

Details zu dieser und vielen anderen Komplexitätsklassen finden Sie unter [Komplexitäts-Wiki in Stanford, früher am Caltech](#).

Mitteilung: BCBS lässt sich nicht mit einem Faktor n^δ approximieren für ein bestimmtes $\delta > 0$, falls nicht $NP \subseteq \bigcap_{\epsilon > 0} \text{BPTIME}(2^{n^\epsilon})$.

Eine ähnliche Approximationsschranke konnte Feige 2002 unter einer anderen Annahme zeigen, nämlich unter der sogenannten *Durchschnittsfalhärte* von 3SAT. Auch hier kommt aber wieder eine Randomisierung ins Spiel.

Label Cover: ein hübsches Hilfsproblem

Eine Instanz wird beschrieben durch:

einen bipartiten Graphen $G = (V_1, V_2; E)$, $E \subseteq V_1 \times V_2$,

natürliche Zahlen N, M, B ,

Abbildungen $\pi_{u,v} : \{1, \dots, M\} \rightarrow \{1, \dots, N\}$ für jede Kante $(u, v) \in E$.

Frage: Gibt es Knotenmarkierungen $L_1 : V_1 \rightarrow \{1, \dots, M\}$ und $L_2 : V_2 \rightarrow \{1, \dots, N\}$, sodass mindestens B Kanten *abgedeckt* sind ?

Hierbei ist eine Kante (u, v) abgedeckt durch die Markierung, falls $\pi_{u,v}(L_1(u)) = L_2(v)$.

Arora und Lund konnten 1997 zeigen: Die Maximierungsvariante von Label Cover ist hart zu approximieren ist für jeden konstanten Faktor, wenn nicht PTIME = NP.

Label Cover: Mehr Einzelheiten

Arora und Lund konnten genauer folgendes zeigen:

Zu jedem $0,5 > \epsilon > 0$ gibt es eine Konstante $k(\epsilon)$, sodass für alle Instanzen von Label Cover mit $M, N \leq k(\epsilon)$ die nachstehend beschriebenen zwei Fälle *NP*-hart zu unterscheiden sind:

- (1) Es gibt eine Markierung, die alle Kanten abdeckt.
- (2) Es gibt keine Markierung, die mehr als $\epsilon \cdot |E|$ viele Kanten abdeckt.

Die so entstandene Lücke wird im Beweis der in der vorigen Folie gezeigten Behauptung verwendet.

Unique Game Conjecture (UGC) ist ein seltsamer Name für eine Einschränkung von Label Cover; tatsächlich wurden Label Cover und die Unique Game Conjecture zunächst in Termini von Spielen definiert.

Vermutung (UGC): Für alle $\epsilon, \delta \in (0; 1/2)$ gibt es eine Konstante $k(\epsilon, \delta)$, sodass für Label Cover Instanzen mit $M = N = k(\epsilon, \delta)$, in denen darüber hinaus die Abbildungen $\pi_{u,v}$ sämtlich Permutationen sind, es keinen Polynomzeit-Algorithmus gibt, der unterscheiden kann zwischen

- (1) Es gibt eine Markierung, die wenigstens $(1 - \delta)|E|$ viele Kanten abdeckt und
- (2) Es gibt keine Markierung, die mehr als $\epsilon \cdot |E|$ viele Kanten abdeckt.

Viele Forscher bezweifeln (noch) die Gültigkeit der UGC.

Konsequenzen der UGC:

Khot und Regev konnten 2003 zeigen:

Mitteilung: Gilt die UGC, so gibt es keinen Algorithmus, der das Knotenüberdeckungsproblem auf einen konstanten Faktor besser als 2 approximiert.

Noch überraschender ist das Ergebnis für MaxCut:

Hierbei wird für einen gegebenen Graphen $G = (V, E)$ nach einer Knotenmenge S gesucht, die die Zahl der Kanten mit genau einem Endpunkt in S maximiert.

Goemans und Williamson haben 1995 einen (komplizierten) α -Faktor-Approximationsalgorithmus gefunden mit $\alpha = \min_{0 < \theta \leq \pi} (\pi(1 - \cos(\theta)) / (2\theta))$.

Unter der Annahme der UGC ist dieser Faktor bestmöglich !