Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

20. Oktober 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 1

Aufgabe 1: Zur asymptotischen Notation:

1. Beweisen oder widerlegen Sie die folgenden Behauptungen:

$$\begin{aligned} 2^{n+c} &\in O(2^n) \\ 2^{cn} &\in O(2^n), \\ 2^{n^2} &\in O(2^{kn}) \quad \text{für ein geeignetes } k \end{aligned}$$

2. Sei $\alpha : \mathbb{N} \to \mathbb{N}$ eine beliebige schwach monoton wachsende, aber nicht schließlich konstante Funktion, d.h. $\alpha(n+1) \geq \alpha(n)$ und für jedes n gibt es ein k derart dass $\alpha(n+k) > \alpha(n)$ ist. Zeigen Sie, dass $\alpha(n)^{\log n}$ *superpolynomial* ist: für jedes d ist $n^d \in O(\alpha(n)^{\log n})$, aber andererseits ist $\alpha(n)^{\log n} \notin n^{O(1)}$.

Aufgabe 2: Sei $G = (V, E)$ ein ungerichteter Graph. Eine Färbung von G mit k Farben ist eine Funktion $\chi : V \to \{1, \ldots, k\}$ mit der Eigenschaft

$$\{x, y\} \in E \implies \chi(x) \neq \chi(y) \quad \text{für alle } x, y \in V\,.$$

Das Entscheidungsproblem k-COLOR ist wie folgt definiert: Eine Instanz ist ein ungerichteter Graph G, gefragt ist, ob es eine Färbung von G mit k Farben gibt.

1. 2-COLOR ist in **P**. Geben Sie einen Algorithmus an, der dies zeigt. [*Hinweis*: Welche Konsequenzen hat die Zuweisung einer Farbe an einen Knoten?]

2. Zeigen Sie, dass 3-COLOR **NP**-vollständig ist. [*Hinweis*: Geben Sie eine Reduktion von 3-SAT auf 3-COLOR an. Dreiecke sind für die Konstruktion nützlich.]

Abgabetermin: Freitag, 27. Oktober, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

27. Oktober 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 2

Aufgabe 3: Betrachten Sie das folgende Entscheidungsproblem: eine Instanz besteht aus einem Graphen G und einer natürlichen Zahl $k \in \mathbb{N}$, und gefragt ist, ob das viertkleinste *vertex cover* in G höchstens k Ecken enthält.

Geben Sie ein Optimierungsproblem P an, so dass das zugehörige Entscheidungsproblem P_D gerade das oben Beschriebene ist, und zeigen Sie, dass P in **NPO** ist.

Aufgabe 4: Ein Problem P heißt **NP**-*leicht*, falls es ein Entscheidungsproblem Q in **NP** gibt mit $P \leq_T Q$.

Zeigen Sie, dass das Optimierungsproblem Minimum Traveling Salesperson **NP**-leicht ist. [*Hinweis*: Betrachten Sie das Entscheidungsproblem, dessen Instanzen Tripel (G, w, k) aus einem Graphen G, einem Weg w in G und einer natürlichen Zahl k sind, und bei dem gefragt wird, ob w zu einer Tour mit Kosten $\leq k$ erweitert werden kann.]

Abgabetermin: Freitag, 3. November, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

3. November 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 3

Aufgabe 5: Sei MAXIMUM FRACTIONAL KNAPSACK die folgende Variante des Rucksack-Problems:

Instanz:	Menge $X = \{x_1, \ldots x_n\}$,		
	für jedes $x_i \in X$:	Gewicht	$a_i \in \mathbb{N}$
		Wert	$p_i \in \mathbb{N}$
	Kapazität $b \in \mathbb{N}$		
Lösung:	Vektor $\alpha \in \mathbb{Q}^n$ mit $0 \leq \alpha_i \leq 1$, so dass $\sum \alpha_i a_i \leq b$		
Maß:	Gesamtwert $\sum \alpha_i p_i$		

Zeigen Sie, dass ein Greedy-Algorithmus stets eine optimale Lösung findet. Das Problem MAXIMUM FRACTIONAL KNAPSACK ist also in **PO**.

Aufgabe 6: Für eine Instanz x von MAXIMUM KNAPSACK bezeichne $m_H(x)$ den Wert der vom modifizierten Greedy-Algorithmus gefundenen Lösung,

$$m_H(x) = \max(m_{Greedy}(x), p_{max}) .$$

Konstruieren Sie für jedes $\epsilon > 0$ eine Instanz x, für die $m^*(x)/m_H(x) > 2 - \epsilon$ ist. Die in der Vorlesung gegebene Analyse des Algorithmus kann also nicht verbessert werden.

Aufgabe 7: Finden Sie für jede Konstante $c > 1$ und jedes $n \geq 4$ eine Instanz $x_{c,n}$ von MINIMUM TRAVELING SALESPERSON mit n Städten, derart dass für die vom Algorithmus *Nearest Neighbor* gefundene Lösung mit den Kosten $m_{NN}(x_{c,n})$ gilt: $m_{NN}(x_{c,n})/m^*(x_{c,n}) > c$.

Abgabetermin: Freitag, 10. November, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

10. November 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 4

Aufgabe 8: Zeigen Sie, dass die in der Vorlesung gezeigte Schranke für die Regel *LPT* beim Problem MINIMUM SCHEDULING ON IDENTICAL MACHINES,

$$m_{LPT}(x)/m^*(x) \leq (\frac{4}{3} - \frac{1}{3p})$$

für Instanzen x mit $p = 2$ scharf ist.

Aufgabe 9: Betrachten Sie die folgende Variante *LPT'* der Regel *LPT*: die Aufträge $t_1, \ldots t_n$ seien sortiert, so dass $\ell_1 \geq \ell_2 \geq \ldots \geq \ell_n$. Zunächst werden t_1 und t_2 dem Prozessor 1 zugewiesen, danach wird *List Scheduling* mit der Regel *LPT* auf die verbleibenden Aufträge $t_3, \ldots, t_n$ angewandt. Die so gefundene Lösung wird mit der durch *LPT* gefundenen verglichen und die bessere ausgewählt.

Bezeichne $m_{LPT'}(x)$ die makespan der Lösung, die auf diese Weise gefunden wird. Geben Sie eine obere Schranke für $m_{LPT'}(x)/m^*(x)$ für Instanzen x mit $p = 2$ an, die besser ist als die für *LPT* angegebene $(m_{LPT}(x)/m^*(x) \leq \frac{7}{6})$.

Aufgabe 10: Betrachten Sie den Algorithmus *Best Fit Decreasing* für das Problem MINIMUM BIN PACKING.

> Sortiere A so dass $a_1 \geq a_2 \geq \ldots \geq a_n$
> $k := 1$; $B_1 := \emptyset$
> Für $i := 1$ bis n
>   finde $1 \leq j \leq k$ mit $\Sigma B_j + a_i \leq 1$ and ΣB_j maximal
>   falls ein solches j existiert, setze $B_j := B_j \cup \{a_i\}$
>   sonst setze $k := k + 1$; $B_k := \{a_i\}$

Beispiel 2.6 im Buch zeigt, dass dieser Algorithmus für manche Instanzen besser ist als *First Fit Decreasing*. Finden Sie eine Instanz, für die *First Fit Decreasing* weniger bins verwendet als *Best Fit Decreasing*.

Abgabetermin: Freitag, 17. November, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

17. November 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 5

Aufgabe 11: Betrachten Sie die Maximierungs-Variante von MINIMUM BIN PACKING, bei der zusammen mit den items $a_1, \dots, a_n \in \mathbb{Q}$, mit $0 \leq a_i \leq 1$, die Zahl m der bins, mit $\sum_i a_i \leq m$, eingegeben wird. Dabei soll die Anzahl der items a_i, die in den m bins untergebracht werden können, maximiert werden.

Beschreiben Sie einen sequentiellen Algorithmus für dieses Problem, analog zu *First Fit*, und zeigen Sie, dass dieser eine Lösung findet, in der mindestens $n/2$ der a_i verpackt werden.

Aufgabe 12: Konstruieren Sie für jedes hinreichend große n einen 3-färbbaren Graphen mit n Knoten, für den der Algorithmus *Smallest Last* $\Omega(n)$ Farben benötigt.

Aufgabe 13: Betrachten Sie die Variante von MAXIMUM CUT, bei der nur Instanzen $G = (V, E)$ mit $|V|$ gerade betrachtet werden, und wo in der Partition $V = V_0 \cup V_1$ beide Hälften gleich groß sein sollen, also $|V_0| = |V_1|$ gelten muss.

Entwerfen Sie einen Algorithmus, der diese Problemvariante in polynomialer Zeit mit Lokaler Suche löst, und geben Sie eine möglichst scharfe obere Schranke an dessen performance ratio an.

Abgabetermin: Freitag, 24. November, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

24. November 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 6

Aufgabe 14: Betrachten Sie das Problem MINIMUM HITTING SET:

Instanz:	Familie $\mathcal{C} = \{C_1, \dots, C_k\}$ von Teilmengen einer endlichen Menge S.
Lösung:	Teilmenge $S' \subseteq S$ so dass $S' \cap C_i \neq \emptyset$ für alle $C_i \in \mathcal{C}$.
Maß:	Mächtigkeit $\|S'\|$

Entwerfen sie einen Primal-Dual-Algorithmus für dieses Problem, der eine Lösung findet, deren Maß höchstens das k-fache des Optimums beträgt, wobei $k := \max_i |C_i|$.

[Hinweis: MINIMUM HITTING SET ist eine Verallgemeinerung von MINIMUM VERTEX COVER!]

Aufgabe 15: Konstruieren Sie ein Approximationsschema für das Problem MINIMUM SCHEDULING ON IDENTICAL MACHINES im Fall, wo die Zahl p der Prozessoren fest (also nicht Bestandteil der Eingabe) ist. Verwenden Sie dazu Dynamic Programming mit der *fixed partitioning* Technik.

Abgabetermin: Freitag, 1. Dezember, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

1. Dezember 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 7

Aufgabe 16: Sei $P \in \mathbf{NPO}$ ein Minimierungsproblem, und k eine Konstante so dass das Problem, zu entscheiden ob für $x \in I_P$ $m^*(x) \leq k$ ist, $\mathbf{NP}$-hart ist. Zeigen Sie, dass P nicht r-approximierbar ist, für $r < \frac{k+1}{k}$, falls $\mathbf{P} \neq \mathbf{NP}$.

Aufgabe 17: Ein Optimierungsproblem $P \in \mathbf{NPO}$ heisst *einfach*, falls für jede Konstante $k \in \mathbb{N}$ das Problem, zu entscheiden ob für $x \in I_P$ der Wert $m^*(x)$ besser als k ist, in $\mathbf{P}$ ist.

1. Zeigen Sie, dass MAXIMUM CLIQUE einfach ist, MINIMUM GRAPH COLORING jedoch nicht, falls $\mathbf{P} \neq \mathbf{NP}$.
2. Zeigen Sie, das jedes Problem in **PTAS** einfach ist.

Aufgabe 18: Ein Optimierungsproblem $P \in \mathbf{NPO}$ heisst *beschränkt*, falls es eine Konstante b und einen Algorithmus A gibt mit:

- Eingabe: $x \in I_P$ und $c \in \mathbb{N}$, Ausgabe: $A(x, c) \in S_P(x)$
- Maß $m(x, A(x, c)) \leq m^*(x) + bc$
- Laufzeit $O(|x|^d)$, wobei d nur von $m(x, A(x, c))/c$ abhängt.

Zeigen Sie:

1. MAXIMUM KNAPSACK ist beschränkt.
2. Ein Optimierungsproblem $P \in \mathbf{NPO}$ ist in **PTAS** genau dann, wenn es einfach und beschränkt ist.

Abgabetermin: Freitag, 8. Dezember, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

8. Dezember 2000

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme
Blatt 8

Aufgabe 19: Betrachten Sie das Problem MINIMUM EDGE COLORING:

Instanz:	Graph $G = (V, E)$
Lösung:	Kantenfärbung $\chi : E \to \{1, \ldots, k\}$ so dass $\chi(e) = \chi(e')$ nur falls $e \cap e' = \emptyset$, d.h. e und e' haben keinen gemeinsamen Endpunkt.
Maß:	Anzahl der Farben k

Zeigen Sie unter Verwendung der Lückentechnik, dass MINIMUM EDGE COLORING nicht in **PTAS** ist.

Hinweis: Bezeichne Δ den maximalen Grad eines Knoten in V. Offensichtlich braucht man mindestens Δ Farben. Andererseits sagt ein Satz der Graphentheorie, dass $\Delta + 1$ Farben stets ausreichen.

Aufgabe 20: Zeigen Sie, dass $\mathbf{PTAS}^\infty \subsetneq \mathbf{APX}$ ist, falls $\mathbf{P} \neq \mathbf{NP}$.

Aufgabe 21: Die Klasse $\mathbf{PCP}_{c,s}[r(n), q(n)]$ ist definiert als die Klasse aller Entscheidungsprobleme P, für die es einen $(r(n), q(n))$-beschränkten Verifikator V gibt mit

$$x \in Y_P \quad \to \quad \exists \pi \;\; \mathbb{P}_\rho[V(x, \rho, \pi) = \text{ja}] \geq c$$

$$x \in N_P \quad \to \quad \forall \pi \;\; \mathbb{P}_\rho[V(x, \rho, \pi) = \text{nein}] \geq 1 - s$$

Das **PCP**-Theorem lautet also in dieser Notation $\mathbf{NP} = \mathbf{PCP}_{1,0.5}[O(\log n), O(1)]$. Folgende schärfere Charakterisierungen von **NP** sind bekannt:

$$\begin{aligned} \mathbf{NP} &= \mathbf{PCP}_{1,0.32}[O(\log n), 9] \\ &= \mathbf{PCP}_{1,0.76}[O(\log n), 3] \\ &= \mathbf{PCP}_{0.99,0.51}[O(\log n), 3] \end{aligned}$$

Welche Schranken folgen daraus jeweils an die Approximierbarkeit von MAXIMUM 3-SATISFIABILITY ?

Abgabetermin: Mittwoch, 20. Dezember, 13^{00} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

12. Januar 2001

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 9

Aufgabe 22: Zeigen Sie, dass ein Verifikator, der nur $O(\log n)$ Zufallsbits verwendet und nur ein Bit eines Beweises anschauen kann, deterministisch in polynomialer Zeit simuliert werden kann, also $\mathbf{PCP}[O(\log n), 1] = \mathbf{P}$ gilt.

Tatsächlich gilt sogar $\mathbf{PCP}[O(\log n), 2] = \mathbf{P}$. Wer viel Zeit hat, kann versuchen, auch dies zu beweisen.

Aufgabe 23: Zeigen Sie, dass die Inklusionen

$$\mathbf{APX} \subseteq \log\text{-}\mathbf{APX} \subseteq \text{poly-}\mathbf{APX} \subseteq \text{exp-}\mathbf{APX}$$

echt sind, falls $\mathbf{P} \neq \mathbf{NP}$ ist. Konstruieren Sie dazu jeweils ein künstliches Problem, das in der größeren Klasse, aber nicht in der kleineren ist.

Aufgabe 24: Führen Sie den Beweis der Implikation

$$P \leq_L Q \text{ und } P \in \mathbf{APX} \implies P \leq_{AP} Q$$

in dem Fall, wo P ein Maximierungsproblem ist, aus.

[*Hinweis*: Definieren Sie eine Funktion g', die die bessere der Lösungen $g(x, y)$ und $A(x)$ auswählt, wobei A der laut Voraussetzung für P vorhandene Approximationsalgorithmus ist.]

Abgabetermin: Freitag, 19. Januar, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

19. Januar 2001

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 10

Aufgabe 25: Zeigen Sie, dass das Problem MINIMUM TRAVELLING SALESPERSON exp-**APX**-vollständig unter AP-Reduktion ist.

[*Hinweis*: Reduzieren Sie das in der Vorlesung angegebene Problem MINIMUM WEIGHTED 3SATISFIABILITY* darauf. Finden Sie dazu in der Literatur eine geeignete, direkte Reduktion zwischen den Entscheidungsproblemen 3SAT und TSP, z.B. in Papadimitriou "*Computational Complexity*", Addison-Wesley 1994]

Aufgabe 26: Definieren Sie einen approximationserhaltenden Reduktionsbegriff $\leq_\mathsf{F}$, derart dass **FPTAS** unter $\leq_\mathsf{F}$ abgeschlossen ist,

$$\mathrm{Q} \in \textbf{FPTAS} \text{ und } \mathrm{P} \leq_\mathsf{F} \mathrm{Q} \implies \mathrm{P} \in \textbf{FPTAS}\,,$$

und finden Sie ein Problem, dass **PTAS**-vollständig unter $\leq_\mathsf{F}$ ist.

Abgabetermin: Freitag, 26. Januar, 11^{30} Uhr.

Institut für Informatik
der Universität München
Dr. J. Johannsen

WS 2000/2001

26. Januar 2001

Übungen zur Vorlesung
Approximationsverfahren für Optimierungsprobleme

Blatt 11

Aufgabe 27: Geben Sie ein Approximationsverfahren (mit konstanter performance ratio) für das Problem MAXIMUM NOT-ALL-EQUAL 3-SATISFIABILITY an. Verwenden Sie dazu den *Greedy*-Ansatz.

Aufgabe 28: MAXIMUM 3-SATISFIABILITY(b) ist die Einschränkung des Problems MAXIMUM 3-SATISFIABILITY auf Instanzen, in denen jede Variable höchstens b mal vorkommt. In der Vorlesung wird gezeigt, dass MAXIMUM 3-SATISFIABILITY(3) **APX**-vollständig ist.

Geben Sie einen Algorithmus an, der zeigt dass MAXIMUM 3-SATISFIABILITY(2) in **PO** ist.

Aufgabe 29: Wählen Sie aus dem *Compendium of NP Optimization Problems* ein **APX**-vollständiges Problem aus, dessen Vollständigkeit aber nicht im Buch oder der Vorlesung bewiesen wurde, und weisen sie die **APX**-Vollständigkeit nach.

Abgabetermin: Freitag, 2. Februar, 11^{30} Uhr.