Theoretische Informatik II

Prof. Christoph Kreitz / Nuria Brede
Universität Potsdam, Theoretische Informatik — Sommersemester 2009

Übungsblatt 12 (Version 1) — Abgabetermin: –

Zur Vorbereitung ist es sinnvoll, sich mit folgenden Fragen auseinanderzusetzen:

- Wie wird die $\mathcal{NP}$-Vollständigkeit des Rucksack-Problems gezeigt?
- Was für Möglichkeiten gibt es, mit $\mathcal{NP}$-vollständigen Problemen in der Praxis umzugehen?
- Was ist ein Approximationsalgorithmus? Wie beschreibt man Optimierungsprobleme? Was ist die Güte eines Aproximationsalgorithmus und wie kann man sie bestimmen? Welche positiven und negativen Resultate bzgl. der Approximation $\mathcal{NP}$-vollständiger Probleme wurden in der Vorlesung vorgestellt?
- Was ist ein probabilistischer Algorithmus? Welche probabilistischen Berechnungsmodelle und Spachklassen gibt es? Wie kann die Fehlerwahrscheinlichkeit bei probabilistischen Algorithmen verkleinert werden? Welche Beispiele wurden in der Vorlesung besprochen?
- *Beginnen Sie rechtzeitig mit der Wiederholung der Themen der Theoretischen Informatik II!!!*

Kurzquiz 12

(1)	Polynomielle Approximationsalgorithmen können optimale Lösungen für Optimierungsvarianten $\mathcal{NP}$-vollständiger Probleme finden, falls $\mathcal{P} \neq \mathcal{NP}$.	**ja**	**nein**
(2)	Seien $W(x, A(x))$ und $OPT_L(x)$ wie in Vorlesungsfolien definiert. Dann ist die Güte eines Approximationsalgorithmus A mit Eingabe x bei einem Minimierungsproblem das Ergebnis der Division von $W(x, A(x))$ durch $OPT_L(x)$.	**ja**	**nein**
(3)	Bei der Optimierungsvariante des Rucksack-Problems ist es möglich, für einen konstanten Wert k einen Algorithmus anzugeben, so dass der Differenzbetrag aus dem Wert einer optimalen Lösung des Problems und dem Wert der durch den Algorithmus bestimmten Lösung kleiner gleich k ist.	**ja**	**nein**
(4)	Die Fehlerwahrscheinlichkeit bei randomisierten polynomiellen Algorithmen kann durch wiederholte Ausführung unter jede beliebige Schranke gesenkt werden.	**ja**	**nein**
(5)	Las-Vegas-Algorithmen geben mit einer Wahrscheinlichkeit von $\frac{1}{2}$ eine korrekte Antwort.	**ja**	**nein**

Aufgabe 12.1 *($\mathcal{NP}$-Vollständigkeit)*

Sei $F = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4) \wedge (x_4 \vee \overline{x_3} \vee x_1) \wedge (\overline{x_2} \vee x_3 \vee x_4)$.

Wenden Sie die in der Problem-Reduktion *3SAT* $\leq_p$ *KP* benutzte Transformation auf F an.

Aufgabe 12.2 *(Approximation)*

Ein kombinatorisches Optimierungsproblem lässt sich durch vier Komponenten charakterisieren [1]:

- $\mathcal{D}$: Die Menge der Probleminstanzen (Eingaben)

- $L(x)$ für $x \in \mathcal{D}$: Die Menge der zu einer Eingabe x zulässigen Lösungen.
- Die Bewertungs-Funktion $W : L(x) \to \mathbb{N}^+$
- $ziel \in \{min, max\}$: Handelt es sich um eine Maximierung oder Minimierung?

(a) Charakterisieren Sie das Optimierungsproblem *KP*$_{OPT}$ (siehe Einheit 6.4, Folie 3) anhand der oben genannten vier Komponenten.

(b) Betrachten Sie folgende Instanz von *KP*$_{OPT}$:
∘ Gewichte: $g_1 = 5,\ g_2 = 3,\ g_3 = 5,\ g_4 = 7,\ g_5 = 4,$
∘ Nutzwert: $a_1 = 3,\ a_2 = 3,\ a_3 = 8,\ a_4 = 2,\ a_5 = 6$
∘ Mindestnutzen: $A = 12$
Bestimmen Sie den Wert einer optimalen Lösung für diese Instanz.

(c) Angenommen, das Problem *KP* wäre in polynomieller Zeit entscheidbar. Zeigen Sie, dass es dann auch einen Polynomialzeit-Algorithmus für die Optimierungsvariante *KP*$_{OPT}$ gäbe. Gehen Sie in zwei Schritten vor: Berechnen Sie zunächst das optimale Gewicht und bestimmen Sie erst dann die zugehörige Bepackung. (Ein informaler Algorithmus genügt.)

Aufgabe 12.3 *(Randomisierte Approximation)*

MAX-SAT ist eine Optimierungsvariante des Erfüllbarkeits-Problems *SAT*, bei der nach einer Belegung gesucht wird, die möglichst viele der Klauseln einer aussagenlogischen Formel in KNF erfüllt.

(a) Charakterisieren Sie *MAX-SAT* anhand der in Aufgabe 12.2 genannten vier Komponenten eines Optimierungsproblems.

(b) Betrachten Sie folgenden Algorithmus `RANDOM-ASSIGN`:

1. Bei Eingabe n
2. Für $i := 1 \ldots n$ $\begin{cases} \text{mit Wahrscheinlichkeit } \frac{1}{2} \text{ belege } x_i := wahr \\ \text{mit Wahrscheinlichkeit } \frac{1}{2} \text{ belege } x_i := falsch \end{cases}$
3. Gib $b = (x_1, \ldots, x_n)$ aus.

Angenommen, wir betrachten eine aussagenlogische Formel F in KNF mit $F = c_1 \wedge \cdots \wedge c_m$ und Variablen $\{x_1, \ldots, x_n\}$. Sei b eine mit Hilfe von `RANDOM-ASSIGN` gefundene Belegung für F. Wie groß ist die Wahrscheinlichkeit, dass eine Klausel c_j ($1 \leq j \leq n$) mit k Literalen von b erfüllt wird?

Hinweis: Die Wahrscheinlichkeit $Pr[x_i = wahr]$, dass eine Variable x_i mit wahr belegt wird, ist gleich der Wahrscheinlichkeit $Pr[x_i = falsch]$. Da für $i \neq j$ die Wahrscheinlichkeiten $Pr[x_i = b_i]$ und $Pr[x_j = b_j]$ unabhängig sind, gilt $Pr[(x_i = b_i) \wedge (x_j = b_j)] = Pr[x_i = b_i] \cdot Pr[x_j = b_j]$ (mit $b_i, b_j \in \{wahr,\ falsch\}$).

(c) Wie groß ist die Wahrscheinlichkeit, dass eine von `RANDOM-ASSIGN` erzeugte Belegung eine *3SAT*-Formel mit 5 Klauseln erfüllt? Wie gross ist die Wahrscheinlichkeit, dass 4 der 5 Klauseln erfüllt werden?

Hintergrund: Normalerweise betrachtet man bei Approximationsalgorithmen die *relative Güte* des Algorithmus (siehe Einheit 6.4, Folie 5). Bei randomisierten Approximationsalgorithmen wird dieser Begriff angepasst, indem für den Wert der Ausgabe des Algorithmus in der Güte-Berechnung der *Erwartungswert* eingesetzt wird. So erhält man die *erwartete relative Güte* eines solchen Algorithmus.
Durch weiterführende Überlegungen lässt sich zeigen, dass der oben betrachtete Algorithmus für aussagenlogische Formeln in KNF eine erwartete relative Güte von $\frac{3}{2}$ garantiert.

Literatur

[1] Rolf Wanka. Approximationsalgorithmen – Eine Einführung. 1. Auflage. Wiesbaden : Teubner, 2006.