ETH zürich
Departement Informatik

Approximations- und Online-Algorithmen

Dr. Hans-Joachim Böckenhauer
Prof. Dr. Dennis Komm
Dr. Manuel Wettstein

`courses.csedu.inf.ethz.ch/approx_online_alg_24`

Übungsaufgaben – Blatt 5

Zürich, 21. März 2024

Aufgabe 6

Sei $U \in \text{NPO}$ ein Optimierungsproblem, bei dem die Menge $\{\text{cost}(y) \mid y \in \mathcal{M}(I)\}$ der möglichen Kosten einer Lösung zu einer Instanz I nur aus positiven ganzen Zahlen besteht und für die Kosten der optimalen Lösung Opt zu einer Instanz I gilt, dass $\text{cost}(\text{Opt}(I)) \leq p(|I|, \text{Max-Int}(I))$ für ein von I unabhängiges Polynom p.

Zeigen Sie, dass es kein FPTAS für U geben kann, wenn die Schwellenwertsprache $Lang_U$ stark NP-schwer und $\text{P} \neq \text{NP}$ ist. **10 Punkte**

Aufgabe 7

Wir möchten jetzt ein Optimierungsproblem finden, dessen Schwellenwertsprache *nicht* stark NP-schwer ist, das aber trotzdem keinen FPTAS besitzt, sofern $\text{P} \neq \text{NP}$.

Wir betrachten dazu das *zweidimensionale* Rucksackproblem, bei der Gegenstände einen Wert, ein Gewicht und eine Grösse haben und der Wert des Rucksacks maximiert werden soll, ohne das Maximalgewicht und die Maximalgrösse zu überschreiten:

Optimierungsproblem 2D-Knapsack

Eingabe: Positive ganze Zahlen B_w und B_s, sowie positive ganze Zahlen $c_1, \ldots, c_n$, $w_1, \ldots, w_n$ und $s_1, \ldots, s_n$.

Zulässige Lösungen:

$$\mathcal{M}(B_w, B_s, c_1, \ldots, c_n, w_1, \ldots, w_n, s_1, \ldots, s_n) = \{T \subseteq \{1, \ldots, n\} \mid \sum_{i \in T} w_i \leq B_w, \sum_{i \in T} s_i \leq B_s\}$$

Kosten: Für eine Menge $T \in \mathcal{M}(B_w, B_s, c_1, \ldots, c_n, w_1, \ldots, w_n, s_1, \ldots, s_n)$ ist

$$cost(T, B_w, B_s, c_1, \ldots, c_n, w_1, \ldots, w_n, s_1, \ldots, s_n) = \sum_{i \in T} c_i \,.$$

Ziel: Maximierung.

Wir benötigen ausserdem das folgende NP-schwere Entscheidungsproblem, in dem festgestellt werden soll, ob eine Menge von Zahlen in zwei gleich grosse Mengen mit derselben Summe aufgeteilt werden kann:

Entscheidungsproblem Exact Partition

Eingabe: Positive ganze Zahlen $a_1, \ldots, a_n$ für ein gerades n.

Ausgabe:

"Ja"	falls es ein $T \subseteq \{1, \ldots, n\}$ gibt mit $\lvert T \rvert = n/2$ und $\sum_{i \in T} a_i = \sum_{i \in \{1,\ldots,n\} \setminus T} a_i$
"Nein"	sonst

(a) Konstruieren Sie für eine Eingabe I für Exact Partition eine Eingabe I' für 2D-Knapsack, sodass I' genau dann eine Lösung der Grösse mindestens $n/2$ hat, wenn I eine Ja-Instanz ist.

Hinweis: Verwenden Sie Grössen der Form $s_i = \max_{1 \leq j \leq n}\{a_j\} - a_i$.

(b) Schliessen Sie daraus, dass es kein FPTAS für 2D-Knapsack gibt, ausser wenn P=NP gilt.

(c) Beschreiben Sie, wie ein pseudopolynomieller Algorithmus für 2D-Knapsack aussehen könnte und argumentieren Sie kurz, warum er die gewünschte Laufzeit erreicht.

10 Punkte

Abgabe: Am 28. März zu Beginn der Übungsstunde.