RWTH Aachen **SS 2010**
Lehrgebiet Theoretische Informatik Übungsblatt 10
Kneis—Langer—Rossmanith 07.07.2010

Übung zur Vorlesung Effiziente Algorithmen

Tutoraufgabe 24

Betrachten Sie das FPTAS für das KNAPSACK-Problem, welches auf Runden basiert. In der Vorlesung wurde gezeigt, daß dieses höchstens $O(n^3/\varepsilon)$ Schritte benötigt. Zeigen Sie durch Angabe einer entsprechenden Instanz, daß tatsächlich sogar $\Theta(n^3/\varepsilon)$ gilt.

Lösung:
Betrachte eine Instanz, auf der Gewicht und Nutzen gleich sind. Wähle $p_1, \ldots, p_n$, so daß für jedes $1 \leq i \leq n$ gilt $p_i = i$, und $\epsilon = 1$.
Sei wie im Algorithmus $L := \max\{p_1, \ldots, p_n\} = n$ und für jedes $1 \leq i \leq n$

$$q_i = \left\lfloor \frac{p_i}{\varepsilon L/n} \right\rfloor \varepsilon L/n = i.$$

Mit Hilfe eines leichten Induktionsbeweises über i sieht man leicht ein, daß man für jedes $0 \leq i \leq n$ und den Werten $\{q_j \mid j \leq i\}$ genau

$$S^i = \{0, \ldots, i(i+1)/2\}$$

und damit $|S^i| = \Omega(i^2/\varepsilon)$ erhält.
Zusammen ergibt sich die Gesamtlaufzeit von mindestens

$$\sum_{i=1}^{n} |S^i| = \Omega(1/\varepsilon \sum_{i=1}^{n} i^2) = \Omega(n^3/\varepsilon)$$

mit der bekannten Formel $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$.

Hausaufgabe 20 (10 Punkte)

Betrachten Sie eine Eingabe von KNAPSACK mit n Gegenständen, so daß für jedes $1 \leq i \leq n$ gilt $c_i = p_i = 2^i$, wobei c_i das Gewicht und p_i der Nutzen dieses Gegenstandes ist. Welche Laufzeiten ergeben sich für beliebiges $\varepsilon > 0$ bei den zwei FPTAS, die auf Runden bzw. auf Intervallpartitionierung basieren?

Lösung: Sei wie im Algorithmus $L := \max\{p_1, \ldots, p_n\} = 2^n$ und für jedes $1 \leq i \leq n$

$$2^i \geq q_i = \left\lfloor \frac{p_i}{\varepsilon L/n} \right\rfloor \varepsilon L/n = \left\lfloor \frac{2^i}{\varepsilon 2^n/n} \right\rfloor \varepsilon 2^n/n$$

Für $i < n - \log(n/\varepsilon)$ ist $q_i = 0$, da $\frac{2^i}{\varepsilon 2^n/n} = \frac{n}{\varepsilon 2^{n-i}} < 1$ ist. Für $i \geq n - \log(n/\varepsilon)$ gilt jedoch

$$2^i \geq q_i = \left\lfloor \frac{2^i}{\varepsilon 2^n/n} \right\rfloor \varepsilon 2^n/n > 2^{i-1}.$$

Für beliebiges $n - \log(n/\varepsilon) < i \leq n$ ist daher

$$\sum_{k=0}^{i-2} q_k = \sum_{k=n-\log(n/\varepsilon)}^{i-2} q_k < \sum_{k=n-\log(n/\varepsilon)}^{i-2} 2^{k-1} < 2^{i-1}$$

und somit $|S_i| \geq 2|S_{i-2}|$ bzw. $|S_i| \geq 2^{(i-\log(n/\varepsilon))/2}$. Insgesamt erhalten wir

$$\begin{aligned}
\sum_{i=0}^{n} |S_i| &= \sum_{i=0}^{n-\log(n/\varepsilon)-1} |\{0\}| + \sum_{i=n-\log(n/\varepsilon)}^{n} |S_i| \\
&\geq n - \log(n/\varepsilon) + \sum_{i=0}^{\log(n/\varepsilon)} 2^{i/2} \\
&= n - \log(n/\varepsilon) + \frac{\sqrt{2}^{\log(n/\varepsilon)+1} - 1}{\sqrt{2} - 1} \\
&= \Omega(n + \sqrt{n/\varepsilon}).
\end{aligned}$$

Betrachten wir nun die Methode der Intervallpartitionierung. Dann ist, wie man durch eine Induktion leicht einsieht, ist für $i > 2$ im i-ten Schritt stets $P_i = \sum_{j=1}^{i} 2^j = 2^{i+1} - 3$
Für $i < \log(n/\varepsilon)$ ist die Intervallänge kleiner als 1, und daher $|S_i| = \Omega(2^{i+1})$.
Für $i \geq \log(n/\varepsilon)$ verdoppelt sich (ungefähr) die betrachtete Intervallänge mit jedem Schritt: Das Intervall $[0, 2^{i+1} - 3]$ wird im i-ten Schritt in Unterintervalle der Größe $\epsilon(2^{i+1} - 3)/(n-1)$ unterteilt, ist also jeweils ungefähr doppelt so lang wie die Intervalle der Größe $\epsilon(2^i - 3)/(n-1)$.
Sei S_r^i das die Menge, die sich nach der Eliminierung aus S^i ergibt. Da $p_{i+1} = 2^{i+1} > P_i$, erhalten wir stets $|S^{i+1}| = 2|S_r^i|$.
Per Induktion zeigen wir außerdem, daß für $i \geq \lfloor \log(n/\varepsilon) \rfloor$ stets $\Omega(n/\varepsilon)$ Intervalle besetzt sind, also $|S_r^i| = \Omega(n/\varepsilon)$. Für den Induktionsanfang $i = \rfloor \log(n/\varepsilon) \rfloor$ gilt dies nach obigen Ausführungen. Betrachte im Induktionsschritt von $i-1$ auf i ein beliebiges Teilintervall $[a, b)$ des Gesamtintervalls $[0, P^i] = [0, 2^{i+1} - 3]$. Wenn $b \leq 2^i$, dann enthält $[a, b]$ wenigstens ein Intervall des vorherigen $i-1$-ten Schrittes vollständig, das nach IV besetzt war. Wenn $b > 2^i$, betrachte das Intervall $[\max 0, a - 2^i, b - 2^i)]$; auch dieses enthält nach IV ein kleineres Intervall aus dem vorherigen Schritt vollständig, welches wiedrum mit einer Einschränkung mit einem Zielwert von, sagen wir, x besetzt ist. Dann ist $x + p_i \in [a, b]$.
Diese Induktion zeigt uns, daß wir tatsächlich in jedem Schritt i $|S_i| = \Omega(n/\varepsilon)$ erhalten und daher

$$\sum_{i=1}^{n} |S_i| = \Omega(n^2/\varepsilon).$$

Hausaufgabe 21 (10 Punkte)

Beweisen Sie, daß unter der Annahme P $\neq$ NP TSP keinen Approximationsalgorithmus mit einem Approximationsfaktor von 5 zuläßt.

Traveling Salesman (TSP)
Eingabe: Ein gewichteter Graph $G = (V, E, c)$ mit $c\colon E \to \mathbf{R}_{\geq 0}$
Gesucht: Eine geschlossene Tour minimalen Gewichts, die jeden Knoten genau einmal besucht.
Das Gewicht einer Tour $(v_1 e_1 v_2 e_2 \cdots e_{n-1} v_n e_n v_1)$ ist $\sum_{i=1}^{n} c(e_i)$.

Lösung

In dieser Formulierung des TSP-Problems, ist es sogar NP-schwer, ein zulässige Lösung zu finden:

Dazu nehme man eine Instanz $G = (V, E)$ des Hamiltonian Cycle Problems und gebe jeder Kante das Gewicht 1. Jede zulässige Lösung des TSP Problems ist ein Kreis durch alle Knoten in diesem Graphen.

Üblicherweise wird TSP auf vollständigen Graphen definiert. Hier kann man zeigen, daß es keine 5-Approximation gibt, indem man wiederum eine Reduktion von Hamiltonian Cycle durchführt.

Für einen Graphen $G = (V, E)$ konstruieren wir den gewichteten, vollständigen Graphen $G' = (V, E', c)$ mit $c(e) = 1$ falls $e \in E$ und $5|V|$ sonst.

Falls G einen Hamiltonkreis enthält, so ist dieser offensichtlich eine Tour mit Kosten $|V|$ in G'. Falls es keinen solchen Kreis gibt, so kostet jede Tour mindestens $5|V| + |V| - 1$. Ein 5-Approximationsalgorithmus könnte also entscheiden, ob es einen Hamiltonkreis gibt oder nicht.