Institut für Informatik
Prof. Dr. Heiko Röglin
Anna Großwendt

Online-Algorithmen
Sommersemester 2014
Abgabe: 14.05.2014, 10:15 Uhr

Übungsblatt 5

Aufgabe 5.1

6 Punkte

Es sei $G = (V, E)$ ein Graph mit einer Metrik $d\colon V \times V \to \mathbb{R}_{\geq 0}$. Mit

$$\delta(G) = \frac{\max_{u \neq v} d(u, v)}{\min_{u \neq v} d(u, v)}$$

bezeichnen wir die *Aspect Ratio* von G. Wir betrachten die natürliche Übertragung von Markierungsalgorithmen auf das k-Server-Problem: Knoten entsprechen verschiedenen Seiten und Anfragen an Knoten, auf denen zur Zeit kein Server steht, entsprechen Seitenfehlern. Daraus ergibt sich eine Phaseneinteilung und der Begriff eines markierten Knotens analog zum Paging-Problem.

Zeigen Sie, dass jeder Markierungsalgorithmus δk-kompetitiv ist, wobei $\delta = \delta(G)$ die Aspect Ratio von G ist.

Aufgabe 5.2

6 Punkte

Wir betrachten die natürliche Übertragung von FIFO und LRU auf das k-Server-Problem. Zeigen Sie, dass es für jeden Wert $\delta \geq 1$ und jede Zahl $k \in \mathbb{N}$ einen gewichteten Graphen G mit Aspect Ratio δ gibt, auf dem FIFO und LRU keinen kompetitiven Faktor $r < \delta k$ erreichen.

Aufgabe 5.3

6 Punkte

Wir betrachten die Reduktion des k-Server-Problems auf das Min-Cost-Flow-Problem. Zeigen Sie, dass der maximale Fluss f mit minimalen Kosten alle Kanten der Form (σ_j, σ'_j) voll auslastet.

Hinweis: Sie können annehmen, dass f ganzzahlig ist.

Aufgabe 5.4

6 Zusatzpunkte

In Theorem 2.14 haben wir gezeigt, dass kein randomisierter Online-Algorithmus für das Paging-Problem einen besseren kompetitiven Faktor als H_k erreicht, nicht einmal auf Instanzen, die nur auf $k + 1$ verschiedene Seiten zugreifen. Dazu haben wir für eine Sequenz σ eine Wahrscheinlichkeitsverteilung $(p_1, \ldots, p_{k+1})$ verwendet, wobei p_i die Wahrscheinlichkeit dafür ist, dass nach Bearbeitung der Sequenz σ Seite i nicht im Cache ist. Die Sequenz wurde sukzessive gemäß diesen Wahrscheinlichkeiten generiert.

Die naheliegendste Konstruktionsvorschrift ist, zu jedem Zeitpunkt auf die Seite zuzugreifen, für die die Wahrscheinlichkeit am größten ist, dass sie zu diesem Zeitpunkt nicht im Cache ist. Wir wollen zeigen, dass diese Vorschrift zu einer deutlich schlechteren unteren Schranke als H_k führt.

Geben Sie für eine beliebig kleine reelle Zahl $\varepsilon > 0$ einen randomisierten Algorithmus A an, der auf der gemäß obiger Vorschrift gegen ihn konstruierten Sequenz σ_A erwartete Kosten von

$$\mathbf{E}\left[w_A(\sigma_A)\right] \leq (1 + \varepsilon) \cdot \mathrm{OPT}(\sigma_A) + \tau$$

für eine geeignete Konstante τ verursacht.