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

Lösungsvorschläge – Blatt 11

Zürich, 23. Mai 2024

Lösung zu Aufgabe 14

Wir definieren die Phasen P_i genau wie in der Analyse der Kompetitivität von RMark. Das heisst, Phase P_1 beginnt mit dem ersten Seitenfehler von RMark. In Phase P_i werden k unterschiedliche Seiten angefragt und die Phase endet unmittelbar bevor die $(k+1)$-ste unterschiedliche Seite angefragt wird. Die Anzahl neuer Seiten in jeder Phase ist immer $\ell = 1$, weil es insgesamt nur $k+1$ Seiten gibt. Die erwartete Anzahl Seitenfehler von RMark berechnen wir genau gleich wie bisher, sie beträgt also in jeder Phase $\ell \cdot H_k = H_k$.

Wir können jetzt die Seitenfehler des optimalen Offline-Algorithmus LFD zählen, der jeweils die Seite im Cache ersetzt, deren nächste Anfrage am weitesten in der Zukunft liegt. Wir können davon ausgehen, dass der Cache in beiden Algorithmen zu Beginn aus den Seiten $s_1, \ldots, s_k$ besteht. Das heisst, wenn zu Beginn von Phase P_1 die Seite s_{k+1} angefragt wird, machen beide Algorithmen einen Seitenfehler. In Phase P_1 werden neben s_{k+1} noch $k-1$ der Seiten $s_1, \ldots, s_k$ angefragt. Das heisst, LFD wird die einzige Seite entfernen, die in dieser Phase nicht angefragt wird. Diese Seite wird aber zu Beginn von Phase P_2 angefragt und somit macht LFD auch zu Beginn von Phase P_2 einen Seitenfehler. Dieses Argument können wir beliebig wiederholen und stellen fest, dass LFD tatsächlich zu Beginn jeder Phase einen Seitenfehler machen muss. Da RMark in jeder Phase im Erwartungswert H_k Fehler macht, ist seine Kompetitivität also tatsächlich H_k.

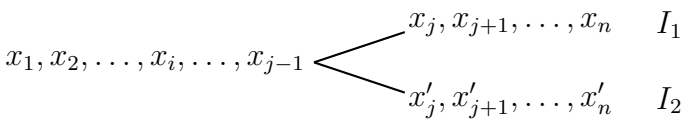
Lösung zu Aufgabe 15

(a) Wir beweisen im Folgenden, dass ein 1-kompetitiver Online-Algorithmus existiert, wenn der Gegenspieler gleichverteilt eine Instanz $I \in \mathcal{I}$ zieht, wobei I mit einer Wahrscheinlichkeit von $\text{Prob}_m(I) = 1/m$ ausgewählt wird.

Seien $I_1, I_2, \ldots, I_m$ die Instanzen in $\mathcal{I}$ und sei $\textsc{Opt}(I_i)$ die optimale LFD-Lösung für I_i mit $1 \leq i \leq m$. Beachten Sie, dass die Instanzen bekannt sind. Wir betrachten den deterministischen Online-Algorithmus Alg, der zunächst o. B. d. A. $\textsc{Opt}(I_1)$ simuliert. Wenn I_1 die tatsächlich ausgewählte Instanz ist, ist die von Alg berechnete Lösung optimal. Falls Alg hingegen in einem Zeitschritt feststellen sollte, dass die gewählte Instanz nicht I_1 ist, simuliert er von diesem Moment an $\textsc{Opt}(I_2)$. Zu diesem Zweck ersetzt Alg seinen Cacheinhalt, sodass er dem von $\textsc{Opt}(I_2)$ entspricht (dies

nehmen wir nur der Einfachheit halber an, tatsächlich kann ALG seinen Cache nach und nach an den optimalen anpassen).

Wir behaupten, dass $\text{OPT}(I_1)$ und $\text{OPT}(I_2)$ dieselbe Anzahl von Seitenfehlern auf dem gemeinsamen Präfix von I_1 und I_2 verursachen. Bezeichne j den Index der ersten Position, in der I_1 und I_2 sich unterscheiden, wie im folgenden Bild dargestellt.



Nehmen wir an, dass eine Seite im Präfix $x_1, x_2, \ldots, x_{j-1}$ angefragt wird, die einen Seitenfehler für $\text{OPT}(I_1)$ verursacht, aber nicht für $\text{OPT}(I_2)$; sei x_i die erste solche Seite. $\text{OPT}(I_1)$ hat x_i in einem vorherigen Zeitschritt ersetzt, da es diejenige war, die am weitesten in der Zukunft wieder angefragt werden würde, nämlich im i-ten Zeitschritt. Da dies allerdings im gemeinsamen Präfix von I_1 und I_2 passiert sein muss (wegen $i \leq j-1$), hat $\text{OPT}(I_2)$ dieselbe Seite ersetzt, was zu einem Widerspruch führt.

Sollte I_2 die tatsächlich ausgewählte Instanz sein, so folgt, dass die Kosten von ALG höchstens $\text{cost}(\text{OPT}(I_2)) + k$ sind, wegen der zusätzlichen Kosten für das Wechseln von $\text{OPT}(I_1)$ zu $\text{OPT}(I_2)$.

Sollte ALG feststellen, dass die gewählte Instanz nicht I_2 ist, so können wir das obige Argument iterieren. In diesem Fall simuliert ALG also $\text{OPT}(I_3)$, usw. Jede solche Änderung verursacht Kosten von k. Sei I_d mit $1 \leq d \leq m$ die tatsächlich ausgewählte Instanz. Wir erhalten

$$\mathbb{E}_{\text{ADV}}[\text{cost}(\text{ALG}(\mathcal{I}))] \leq \text{cost}(\text{OPT}(I_d)) + (d-1)k \leq \text{cost}(\text{OPT}(I_d)) + (m-1)k \,,$$

und da weder m noch k von der Eingabelänge abhängen, ist ALG 1-kompetitiv (im Erwartungswert).

(b) Falls der Gegenspieler eine andere bekannte Wahrscheinlichkeitsverteilung als die Gleichverteilung über $\mathcal{I}$ wählt, gibt dies ALG nur einen Vorteil, da er in diesem Fall die Instanzen ihrer Wahrscheinlichkeit nach sortieren und dann die optimalen Lösungen bezüglich dieser Sortierung simulieren kann. Es folgt, dass es im Fall einer bekannten Wahrscheinlichkeitsverteilung eine beste Strategie für den Gegenspieler ist, die Eingaben gleichverteilt zu wählen.

Sollte die Verteilung nicht bekannt sein, funktioniert die Argumentation aus Aufgabenteil (a) noch immer, da es weiterhin maximal $m-1$ Wechsel gibt und der Zeitpunkt für einen Wechsel noch immer analog erkannt werden kann.

(c) Die Aufgabe scheint zunächst im Widerspruch zum Resultat aus der Vorlesung zu stehen. Dort hatten wir eine Familie von Instanzen $\mathcal{I}$ und eine Wahrscheinlichkeitsverteilung Prob_{ADV} gesehen, sodass

$$\mathbb{E}[\text{cost}(\text{ALG}(\mathcal{I}))] \geq H_k \cdot \text{cost}(\text{OPT}(I))$$

für *jeden* determinstischen Algorithmus ALG, also auch für den aus Teilaufgabe (b).

In dieser Teilaufgabe wird aber eine endliche Anzahl von m Eingaben verwendet. Damit wir mit damit Yaos Prinzip anwenden könnten, müsste die Kompetitivität

strikt sein, was hier nicht der Fall ist, wegen der haben additiven Konstante von $(m-1)\cdot k$. Unser Beweis der unteren Schranke für das Paging-Problem ging ausserdem von einer unendlichen Familie von Instanzen aus. Anders gesagt, die Anzahl m von möglichen Instanzen wuchs dort mit der Länge der Instanzen, das heisst, die Aussage aus der Lösung von Teilaufgabe (a) „da weder m noch k von der Eingabelänge abhängen“ trifft ebenfalls nicht mehr zu.