Algorithms Group **Summer 2022**
Department of Computer Science - IBR
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Michael Perk

Online Algorithms
Homework Assignment 1
May 09, 2022

Solutions are due on Monday, the 16th of May, 2022, until 15:00 in the homework cupboard. You can also hand in your solution in person before the big tutorial begins or via e-mail to `kramer@ibr.cs.tu-bs.de` with CC to `mperk@ibr.cs.tu-bs.de`.

Exercise 1 (The BahnCard Problem: Optimal Offline Algorithm):
In the big tutorial, we considered the BahnCard Problem $BC(C, \beta, T)$ with cost C, cost reduction β and validity duration T. We proved that, in the worst case, no online algorithm can perform better than $2 - \beta$ times the cost of an optimal offline algorithm. Construct an optimal offline algorithm that, for a given sequence σ consisting of n chronologically ordered ticket requests $(t_1, c_1), \ldots, (t_n, c_n)$, produces an optimal solution in $O(n)$ time.
You may make use of the following two facts:

- The optimal offline algorithm never has to buy a BahnCard while it still owns one.
- The optimal offline algorithm never has to buy a BahnCard at a time point that is not the time point of some ticket request.

(20 pts.)

Algorithm 1: Online algorithm SUM for the Bahncard problem

Input: Sequence $\sigma = ((t_i, c_i))_{1 \le i \le n}$ of travel requests, T, β, C
Output: $\gamma = (\gamma_i)_{1 \le i \le n} \in \{0, 1\}^n$, where $\gamma_i = 1$ means buying a BC at request i

if *We already own a BC at request* i **then**
  Output $\gamma_i = 0$
else
  if *The cost of all regular requests in* $(t_i - T, t_i]$ *is at least* c^* **then**
    Output $\gamma_i = 1$
  else
    Output $\gamma_i = 0$

Exercise 2 (The BahnCard Problem: Online Algorithm SUM):
For the BahnCard problem $BC(C, \beta, T)$, we presented the online algorithm SUM. Recall that a request is called a *reduced* request if SUM possesses a BahnCard for that request and regular otherwise, and the *break-even price* c^* is $\frac{C}{1-\beta}$
Let $\sigma = (t_1, c_1), \ldots, (t_n, c_n)$ be a sequence of travel requests. Moreover, let $\tau_1, \ldots, \tau_k$ be the times where the optimal offline solution buys a BahnCard and consider the phases $[0, \tau_1), [\tau_1, \tau_2), \ldots, [\tau_k, \infty)$. We prove that SUM is $(2-\beta)$-competitive by proving $c_{SUM} \leq (2 - c_{OPT})$ for each phase individually.

a) Recall that we call a time interval $I = [b, e)$ expensive if the sum of all costs for travel requests with time $t_i \in I$ is at least c^*, and cheap otherwise. Prove that for each phase $[\tau_i, \tau_{i+1})$ with $1 \leq i \leq k$, the interval $[\tau_i, \tau_i + T)$ is expensive. Moreover, let $\tau_{k+1} := \infty$. Prove that any subinterval of $[\tau_i + T, \tau_{i+1})$ of length at most T is cheap.

b) Prove that, for the first phase $I = [0, \tau_1)$, $c_{SUM} \leq c_{OPT}$

c) Prove that $c_{SUM} \leq (2 - \beta) \cdot c_{OPT}$ for a phase $I = [\tau_i, \tau_{i+1})$ if SUM does not buy a BahnCard in phase I

d) Finally, prove that $c_{SUM} \leq (2 - \beta) \cdot c_{OPT}$ for a phase $I = [\tau_i, \tau_{i+1})$ if SUM buys a BahnCard in phase I. Hint: Decompose I into three intervals I_1, I_2, I_3 based on the time until which SUM possesses a BahnCard from the last phase and the time where SUM decides to buy a new BahnCard.

(3 + 4 + 8 + 10 pts.)

Exercise 3 (Potential Functions and Amortized Analysis):
Consider an abstract online problem where an online algorithm A faces an online sequence $r = r_1 r_2 \ldots r_n$ of *requests*. As a response to each request r_i, A has to perform an action $A(i)$ without knowing the next request r_{i+1}. Each such action incurs a cost $c_A(i) \in \mathbb{R}$. Analogously, the optimal offline algorithm OPT performs actions as response to requests with costs $c_{\text{OPT}}(i)$.
In the analysis of online algorithms, it is often impossible to bound the cost of an online algorithm by proving $c_A(i) \leq c \cdot c_{\text{OPT}}(i)$ for each request i. Therefore, we need a way to distribute the costs of an expensive action of A across several requests.
One way of doing this is by considering a *potential function* $\Phi_r : \{1, 2, \ldots, n\} \to \mathbb{R}_{\geq 0}$ with $\Phi_r(0) = 0$. This potential function acts as a savings account that is not allowed to become negative and that accumulates saved costs to pay for later expensive actions.

(a) Prove the following. If for every request sequence r, there is a potential function Φ_r such that
$$c_A(i) + \Phi_r(i) - \Phi_r(i-1) \leq c \cdot c_{\text{OPT}}(i),$$
then A is c-competitive, i.e., $\sum_{i=1}^{n} c_A(i) \leq c \sum_{i=1}^{n} c_{\text{OPT}}(i)$.

(b) Consider the problem READ INTO BUFFER, where we want to read a non-empty stream s of unknown length into a buffer that is stored in memory as contiguous array of size at most $2|s|$. Reading a symbol from s into the buffer has a cost of 1.

The optimal offline algorithm allocates an array of size $|s|$ once and thus has a cost of $|s|$.

In the online scenario, if the buffer is full, it has to be reallocated and the old contents have to be copied to the new buffer. For every symbol already in the buffer, this incurs an additional cost of 1. Thus, reading the kth symbol from s costs either 1 (not full) or k (buffer full).

Devise a 3-competitive algorithm for Read Into Buffer, using a potential function to prove the competitive ratio.

(5+15 pts.)