Online Algorithms **Summer 2024**

Prof. Dr. Sándor P. Fekete
Peter Kramer

Due: 06.05.2024
Discussion: 13.05.2024

Sheet 1

Please submit your individual solutions using the boxes in front of IZ338, before the exercise timeslot on the due date above. Your homework submission may be handwritten using proper ink (no pencil, no red ink) or printed.

Exercise 1 (BahnCard Problem): **(20 points)**

We consider instances of the BahnCard Problem $BC(C, \beta, T)$ with cost C, cost reduction β and validity period T as introduced in the exercise. We already proved that no online algorithm can guarantee a cost lower than $2 - \beta$ times that of an optimal offline algorithm.

Construct an optimal offline algorithm OPT 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 $\mathcal{O}(n)$ time.

You may make use of the following two facts:

- OPT never has to buy a BahnCard while it still owns one.
- OPT never has to buy a BahnCard at a point in time that is not contained in a ticket request.

Exercise 2 (BahnCard Problem: The SUM Algorithm): **(3 + 4 + 8 + 10 points)**

For the BahnCard Problem $BC(C, \beta, T)$, we introduced the online algorithm SUM:

Input sequence $\sigma = (t_1, c_1), \ldots, (t_n, c_n)$ of travel requests, as well as C, β, and T.
Output sequence $\gamma = \gamma_1, \ldots, \gamma_n \in \{0, 1\}^n$, of purchase decisions.
function SUM(t_i, c_i)
    if we already own a BC at request i **then**
        return $\gamma_i = 0$ ▷ Do not purchase.
    else
        if the cost of all regular requests in $(t_i - T, t_i]$ is at least c^* **then**
            return $\gamma_i = 1$ ▷ Make a purchase.
        else
            return $\gamma_i = 0$ ▷ Do not purchase.

Recall that a request is called *reduced* if SUM possesses a BahnCard for that request and *regular* otherwise, and that the *break-even price* c^* is $\frac{C}{1-\beta}$. Let $\sigma = (t_1, c_1), \ldots, (t_n, c_n)$ be the sequence of travel requests. Moreover, let $\tau_1, \ldots, \tau_k$ be the moments in time when OPT buys a BahnCard.

We prove that SUM is $(2 - \beta)$-competitive by considering the phases $[0, \tau_1), [\tau_1, \tau_2), \ldots, [\tau_k, \infty)$ and 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 costs for travel requests with time $t_i \in I$ is at least c^*, and cheap otherwise. Moreover, let $\tau_{k+1} := \infty$.

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. 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.)

Exercise 3 (Potential Functions and Amortized Analysis): **(5 + 15 points)**
Consider an abstract online problem where an online algorithm A faces a sequence r of online requests $r_1, r_2, \ldots, r_n$. In 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 $OPT(i)$ with cost $c_{OPT}(i)$.

In the analysis of online algorithms, it is often impossible to bound the cost of an online algorithm by proving $c_A(i) \leq c \cdot c_{OPT}(i)$ 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 so-called *potential function*, which we define as

$$\Phi_r : \{1, 2, \ldots, n\} \to \mathbb{R}_{\geq 0} \quad \text{with} \quad \Phi_r(0) = 0.$$

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 $c_A(i) + \Phi_r(i) - \Phi_r(i-1) \leq c \cdot c_{OPT}(i)$, then A is c-competitive, i.e., $\sum_{i=1}^{n} c_A(i) \leq c \sum_{i=1}^{n} c_{OPT}(i)$.

b) Consider the problem Read Into Buffer: 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 its content 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 either costs 1 (not full) or k (buffer full).

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