Online Algorithms **Summer 2024**

Prof. Dr. Sándor P. Fekete
Peter Kramer

Due: 27.05.2024
Discussion: 03.06.2024

Sheet 2

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 (k-Server Problem: Greedy algorithm): (15 points)
In this exercise, we consider the k-Server Problem in the Euclidean plane. In this problem, we start with k servers at given points $s_1, \ldots, s_k \in \mathbb{R}^2$. We are given a sequence of n requests at locations $\sigma_j \in \mathbb{R}^2$. Each request σ_j must be handled immediately by moving one of the servers from its current position p to σ_j. The cost for moving a server from p to σ_j corresponds to the Euclidean distance $d(\sigma_j, p) = ||\sigma_j - p||_2$. The goal is to minimize the total distance traveled by the servers. The algorithm Greedy serves each request using the server closest to the request.

What is the competitive ratio c of Greedy for $k = 2$ servers?

Exercise 2 (k-Server Problem: Double Coverage): (10+10+10 points)
In this exercise, we consider the k-Server Problem in the Euclidean line. In this problem, we start with k servers at given points $s_1, \ldots, s_k \in \mathbb{R}$. We are given a sequence of n requests at $\sigma_j \in \mathbb{R}$. Each request σ_j must be handled immediately by moving one of the servers from its current position p to σ_j. The cost for moving a server from p to σ_j corresponds to the Euclidean distance $d(\sigma_j, p) = |\sigma_j - p|$. The goal is to minimize the total distance traveled by the servers.

We want to prove that the algorithm Double Coverage (DC) is k-competitive: If a request falls outside the convex hull of the servers' current locations, DC serves it with the nearest server. Otherwise, the request is between two adjacent servers. In this case, DC moves both servers toward the request at equal speed until (at least) one of the servers reaches it.

We denote the cost incurred in response to σ_i by $c_{DC}(i)$ and $c_{OPT}(i)$ for DC and an optimal offline algorithm OPT, respectively. Let $M_{\min}$ denote the minimum cost matching between the servers in OPT and DC, and let $\Sigma_{DC} = \sum_{i<j} d(s_i, s_j)$ be the sum of all interpoint distances between DC's servers. Let $\Phi(i) = k \cdot M_{\min} + \Sigma_{DC}$ be a potential function. From the previous homework we know that DC is k-competitive if $c_{DC}(i) + \Phi(i) - \Phi(i-1) \leq k \cdot c_{OPT}(i)$. In this exercise we show that $\Phi(i) - \Phi(i-1) \leq k \cdot c_{OPT}(i) - c_{DC}(i)$.

a) Both the moves of DC and OPT have an influence on the change of potential $\Phi(i) - \Phi(i-1)$ at request σ_i. We split the change of potential in two steps and first consider the influence of OPT, i.e. the change of potential after OPT serves request σ_i and *before* DC serves the request. Show that OPT's move only changes the value of $M_{\min}$ with a difference lower or equal to $k \cdot c_{OPT}(i)$.

b) We now consider DC's move after OPT's potential change was applied. As DC behaves differently in response to some request position σ_i, we split it's behavior into two cases. For the first case, DC changes the position of a single server. Show that in this case,

$$\Phi(i) \leq \Phi(i-1) + k \cdot c_{OPT}(i) - c_{DC}(i).$$

c) In the second case, DC changes the position of two servers. Show that in this case $\Phi(i) \leq \Phi(i-1) + k \cdot c_{OPT}(i) - c_{DC}(i)$, which concludes the proof.

Exercise 3 (k-Server Problem: DC-Tree): **(20 points)**

In this exercise, we consider the k-Server Problem in a tree network. The distance between two points on the tree is the length of the simple path between them.

We say that a server s_j is a *neighbor* to a request σ_i exactly if there is no other server on the path from s_j to σ_i. The DC-Tree algorithm answers each request by moving all neighboring servers at constant speed towards the request. DC-Tree is k-competitive. Let G be any N-node graph. Let T be the minimum spanning tree of G. Show that DC-Tree is $k(N-1)$-competitive.

(Hint: Compare the optimal offline algorithm for T and the optimal offline algorithm for G.)