ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

AS 2007
Prof. Dr. R. Wattenhofer / Roland Flury / Stefan Schmid

Discrete Event Systems
Solution to Exercise 10

1 Competitive Analysis

a) Recall that calls have infinite duration. Therefore, once a cell accepts a call, no neighboring cell can accept a call thereafter. The natural greedy algorithm $\mathcal{A}_{Greedy}$ accepts a call, whenever this is possible. That is, a call in cell C is accepted if no neighboring cell of C has previously accepted a call.

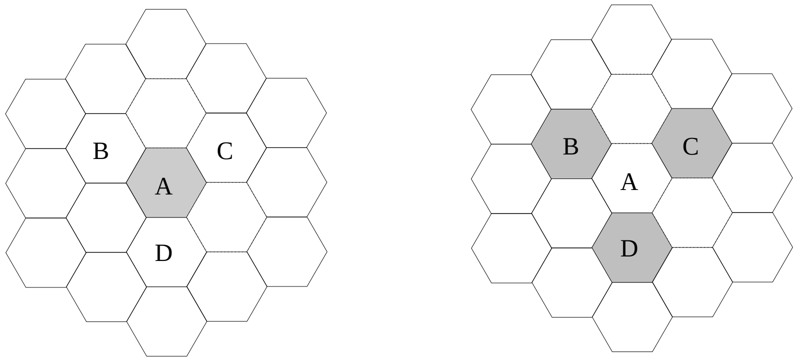

Figure 1: The solutions $\mathcal{A}_{Greedy}$ (left) and $\mathcal{A}_{Opt}$ (right)

By accepting a call, $\mathcal{A}_{Greedy}$ can prevent itself from accepting at most 3 subsequent calls. This is shown in Figure 1. Assume that there are four calls, the first one in A, then three non-interfering ones in neighboring cells B, C, and D. $\mathcal{A}_{Greedy}$ accepts the first and has benefit 1. $\mathcal{A}_{Opt}$ rejects the first call, but accepts the remaining three, resulting in a benefit of 3. The algorithm is 3-competitive.

b) There is no competitive algorithm if calls can have arbitrary durations. Assume that the first call arrives in A and has arbitrary duration. There are two possible actions for an algorithm ALG.

If ALG rejects this call, no further call will arrive and therefore $benefit(ALG) = 0$. The optimal algorithm would have accepted the call, i.e., $benefit(OPT) = 1$. The competitive ratio is infinitely large.

On the other hand, if ALG accepts the call, there will be infinitely many calls coming in state B, each of which has very short duration ϵ. While ALG cannot accept any of these calls (because the call in A has infinite duration), the optimal algorithm rejects the first call and accepts all subsequent calls. This yields $benefit(ALG) = 1$ as opposed to $benefit(OPT) = \beta$, for an arbitrarily large value of β

c) At first sight, it seems that there is no better algorithm than the natural greedy algorithm from part a) of the exercise. After all, the algorithm must accept the first call in order to stay competitive. Accepting the first call, on the other hand, leads to a competitive ratio of 3. However, it can be shown that there exists a *randomized algorithm* with competitive ratio 2.97. This algorithm accepts every call with a certain probability.