ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2015 Prof. R. Wattenhofer / G. Bachmeier, S. Brandt, K.-T. Foerster, J. Seidel

Discrete Event Systems

Exercise Sheet 9

1 Competitive Analysis

In this exercise, we analyze algorithms for cellular networks such as GSM. In such networks, the area is segmented into cells, each of which contains a base station. Due to interference, base-stations in neighboring (adjacent) cells cannot use the same carrier frequencies, but frequencies may be reused in non-interfering cells (i.e., cells that are not neighboring). In this exercise, we use the idealized hexagonal grid for modelling these cells (cf. Figure 1).

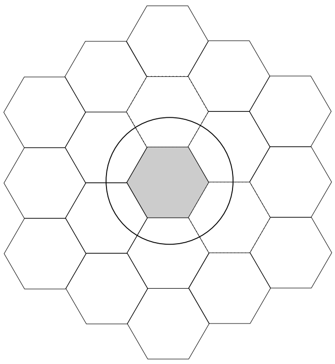


Figure 1: The hexagonal grid modelling the cells. Only a finite part of the infinite grid is shown.

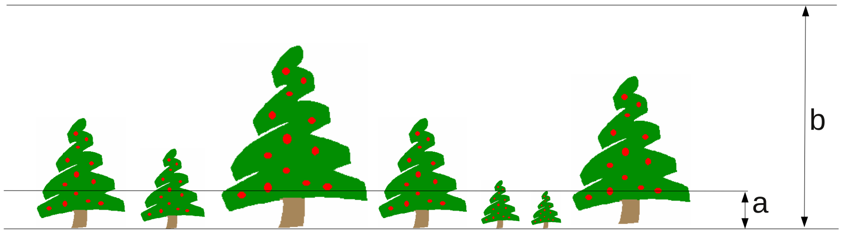
The number of frequencies (channels) in real GSM networks is limited. If there are more calls than channels, some calls must be rejected. In this exercise, we make the simplifying assumption that there is only a *single channel* shared by all base stations. That is, every base station can accept exactly one call. Furthermore, if a call is accepted by a base station in a certain cell, no calls can be accepted in neighboring cells due to interference.

Assume that calls arrive in an online fashion, that is, one after another in an input sequence σ (where σ is essentially a list of cells). We need to accept or reject calls such that there is at most 1 call in a cell and its 6 neighboring cells. The *benefit* of the algorithm is the number of calls we accept.

a) Assume that calls have *infinite duration*. Once a call is accepted, it remains active forever. Describe a natural greedy algorithm for the problem. Is your algorithm c-competitive for some fixed constant c? If so, what is the smallest possible value of c?

b) Assume that every call can have an arbitrary duration, but base stations are not allowed to preempt accepted calls. Does there exist a competitive online algorithm for this scenario? Prove your claim!

2 Competitive Christmas



Roger has been instructed by his children to buy a Christmas tree as large as possible. Nearby his house, a Christmas market sells Christmas trees in a street. Since Roger additionally has to get a lot of presents for Christmas, he does not really have time for searching for a large Christmas tree. But as he passes through this particular street *exactly once* on his optimised shopping route anyway, he intends to get a tree on the way. Being in a hurry, Roger forgot his glasses at home and because he is short-sighted, he can only see the tree right in front of him. Hence, Roger walks through the street and has to decide (online) when to buy a tree (he cannot turn back because then he would not have enough time to get a present for his wife – a disaster). Roger knows from biology classes that the height of fir trees is in the interval $[a, \ldots, b]$ and he further knows when he has arrived at the last tree in the street.

Help Roger! Give as good a deterministic algorithm as possible that tells Roger when to buy a Christmas tree. How good is your algorithm? What is its competitive ratio?

3 The Best Secretary

The Distributed Computing group needs a new secretary because the current one is about to retire after decades of impeccable service. Hence, Roger needs to find a new one with equally good qualifications.

Roger has received n applications and invited all n candidates for an interview. He invites them into his office in a definable order. After each interview, Roger rates the current candidate with a grade $[0, \ldots, \infty]$ and has to decide immediately and irrevocably whether he wants to hire the candidate or not.

We assume that Roger's algorithm is allowed to use randomization for choosing the order in which the candidates are interviewed and for his selection process. Further, we assume that an adversary knows Roger's algorithm, but not the outcome of the random bits. This adversary chooses the set of candidates in a worst-case fashion, knowing the rating that Roger will give to each candidate.

a) Describe an algorithm that with probability $p \geq 1/4$ selects the *best* candidate.

Note: Roger is very ambitious, hence he only wants to hire the *best* candidate. If he does not get the best one, he does not care about the quality of the candidate at all.

b) Can Roger do better?