

## Exercise 3: May 17, 2022

Lecturer: Prof. Yossi Azar

Write short but full and accurate answers. Each solution should appear on a **separate** page and each of its parts should not exceed a page.

1. We are given a connected graph  $G = (V, E)$ . All edges have unit capacity. At step  $i$  we receive a request to allocate a bandwidth  $p_i$  on a specific path  $Q_i$ . If we allocate the bandwidth on the path we receive benefit of  $b_i$  otherwise, we receive no benefit. Also for some known value  $\mu$  and for all  $i$ ,  $1 \leq b_i/p_i \leq \mu$  and  $p_i \leq \frac{1}{\log(2|V|\mu+2)}$ . Our goal is to maximize the total benefit while maintaining the capacity constraints. Design an  $O(\log(|V|\mu))$  competitive algorithm.
2. Consider admission control for the edge disjoint paths problem on an extended star (a center with  $k$  paths each of  $n$  edges). The goal is to maximize the number of accepted paths.
  - (a) Design a deterministic preemptive algorithm which is  $O(\log n)$  competitive.
  - (b) Design a randomized non-preemptive algorithm which is  $O(\log n)$  competitive.
3. Suppose we are given one machine and a set of jobs that arrive over time. The machine can process one job at a time and may preempt jobs. The duration of a job is known at its release time and the benefit of a job is equal to its duration. In order to get the benefit of a job it must be processed immediately at its release time and should not be preempted until its completion.
  - (a) Design a 4 competitive algorithm for the problem.
  - (b) Show a lower bound of  $4 - \epsilon$  for any  $\epsilon > 0$ .
4. Consider scheduling of packets problem over time. Request  $i$  arrives at time  $a_i$  has deadline  $d_i$ , size  $w_i$  slots (all numbers are integers) and a value  $v_i$ . To get the value  $v_i$  for the packet  $i$  one need to transmit the packet on  $w_i$  consecutive slots before the deadline. A packet can be preempted (at integer times) but once preempted it is lost for ever (we cannot continue or restart the transmission).
  - (a) Design a constant competitive preemptive algorithm if  $w_i = k$  for all  $i$
  - (b) Modify the algorithm and the proof if all packets are of sizes of at least  $k$  and at most  $2k$ .
  - (c) Design  $O(\log k)$  randomized competitive algorithm if  $1 \leq w_i \leq k$  for all  $i$ .
5. Suppose we are given one machine and  $m$  clients. At any time each client can hold at most one job. All jobs have unit size and unit value. The machine can process a job from one of the clients at any time step. If there are no jobs the machine is idle. At each time step a job may be created at each of the clients. A client may keep the new job if it does not hold a previous job. Otherwise the new job is discarded. We assume that at each time step, jobs are first created at clients and then the machine decides to process one of the client jobs. The goal is to maximize the number of processed jobs.
  - (a) Show that any on-line algorithm that processes any job if exists, and keeps new jobs at the clients if possible is 2-competitive. Hint: partition the time line into segments where each consists of maximum consecutive times steps in which the on-line algorithm processed jobs. Transform the sequence (while possibly improving the optimum) to a new sequence such that before the beginning of a new segment the optimum does not process a job.
  - (b) Show a lower bound of  $2 - \frac{1}{m}$  for the competitive ratio of any deterministic algorithm.
  - (c) Consider the case where the jobs have values between 1 and  $\alpha$  for some known  $\alpha > 1$ . The goal is to maximize the total value of processed jobs. A client must discard a new job if it holds a previous job, but may also discard a new job even if it does not hold a previous job. Note that if a client decides to keep a job upon its creation, it will have to discard all newer jobs until the execution of that job (even if newer jobs have higher values). Design an  $O(\log \alpha)$  randomized competitive algorithm for the decisions of the clients and the machine.

**Exercise # 3 is due Jun 3, 2022 at 10AM .**