

Advanced Algorithms, Spring 2021

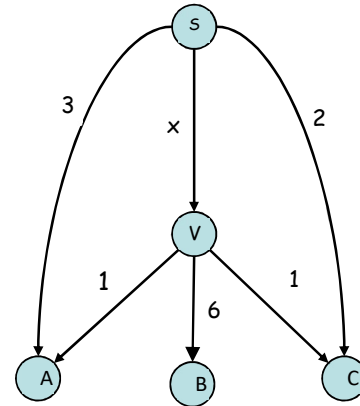
Homework 5. Due date: Sunday 20/6/20

The HW guidelines published in Homework 1 are valid for all the assignments in this course.

Question 1. (20 pts) Three selfish players A,B,C, need to send messages from s to the bottom nodes (one message to each target, the target of player j is node j , for $j \in \{A,B,C\}$). What is the social optimum cost and what are the Nash equilibrium routings in the **fair cost-sharing** model?

Your answer should distinguish between different ranges of x (≥ 0). What is the price of stability and the price of anarchy for each range?

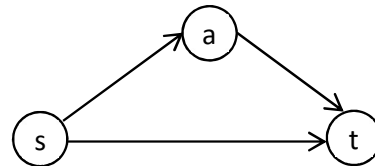
Note: There are 5 different ranges.



Question 2. (20 pts.):

Consider the following **network congestion game**.

The edge costs are $c_{sa}(k) = k$, $c_{at}(k) = 2k$, and $c_{st}(k) = 3k$. For example, if 4 players are using the edge (st) then each of them pays 12.



Assume that 16 players need to use the network.

6 players need a path from s to a , and 10 players need a path from s to t . Formally, for 6 players the only strategy is $\{sa\}$ and for 10 players there are two strategies: $\{sa, at\}$, $\{st\}$.

2.1. Complete the following computation of a Nash Equilibrium profile in this game:

A profile is characterized by the number of $(s-t)$ -players that choose the upper path.

For $0 \leq x \leq 10$, denote by P_x the profile in which x $(s-t)$ -players choose the upper path.

In the profile P_x , the cost of the strategy $\{sa, at\}$ is _____, and the cost of the strategy $\{st\}$ is _____. The strategy P_x is a NE if (fill in condition 1 – players on top path are stable) _____ and (fill in condition 2 – players on lower path are stable) _____.

That is (write the profiles that are NE of the game) _____.

2.2. What is the social optimum for this game (with respect to the objective of minimizing the total players' cost)? Explain your answer.

2.3. What is the price of Stability of this game? Explain your answer.

Question 3. (25 pts.) Consider the following local search heuristic for solving $P2 || C_{\max}$ (load balancing on two machines). It is known that there is no single job for which $p_j > \sum_i p_i / 2$.

a. Assign the jobs on the machines arbitrarily.

b. While there exists a job j such that moving j from M_1 to M_2 or from M_2 to M_1 decreases the absolute gap between the current loads, $|C_1 - C_2|$, then perform this move.

Warm up 1: Verify to yourself that the condition in step b is equivalent to j reducing its cost in the corresponding **job scheduling game** (slides 73+ in the lecture notes).

Warm up 2: Note that if the algorithm is applied on the schedule in slide 79 (jobs of lengths 2,2 on one machine and jobs of lengths 1,1 on the other machine), then no job will migrate in step b.

3.1. Prove that when the algorithm terminates $C_1/2 \leq C_2 \leq 2C_1$. What is the resulting approximation ratio of the algorithm? Explain

3.2. Prove that if in each iteration you select the longest job that can migrate and improve the balance (if there is more than one candidate job), then each job will be moved at most once, and therefore the algorithm terminates after at most n moves.

Note: Questions 4 and 5 consider Online Algorithms – we will cover the relevant material in the

Question 4. (15 pts.) Consider the following cow's algorithm for the Hole in the Fence problem. The difference from the algorithm presented in class is that in this algorithm d is multiplied by two only after the cow walked distance d (and back) **on both sides**.

1. $d=1$; current side = right
2. repeat:
 - i. Walk distance d on current side
 - ii. if find hole then exit
 - iii. else return to starting point
 - iv. Flip current side
 - v. Walk distance d on current side
 - vi. if find hole then exit
 - vii. else return to starting point
 - viii. $d = 2d$
 - ix. Flip current side

Analyze the competitive ratio of the algorithm.

Question 5. (20 pts.) Recall that a matching in a graph $G=(V,E)$ is a set of edges that share no vertices.

A matching M is maximal if there is no matching N such that $M \subseteq N$. A matching M is maximum if for every matching N , $|M| \geq |N|$.

5.1 Prove that a maximal matching is a 2-approximation for the maximum matching.

5.2 Let $G=(U,V,E)$ be a bipartite graph. In the online bipartite matching problem the vertices of V are known in advance, but those of U arrive in an online fashion. When a vertex $u \in U$ arrives, the edges incident to u are revealed. Give a 2-competitive algorithm for the online bipartite matching problem.

5.3 Prove that every deterministic algorithm for the online bipartite matching problem has competitive ratio at least 2.