Algorithms Under Uncertainty : Homework 2
Full Marks : 75
Due : 09/10/2023

Solutions must be done in teams of at most two. Clearly mention the name of your partner at the top. Also, both the partners are required to submit their individual solution copies on Google Classroom. All solutions to be written in latex. No other format is acceptable. You are free to consult online resources, but must mention them. If found hiding your sources, it will be considered an act of plagiarism.

Problem 1. (15 points) Consider the k-server problem on the cycle graph on n vertices.

a. (5 points) Consider the following online algorithm: remove an edge of the cycle graph. Now it becomes a line graph on n vertices. Run the double cover algorithm for line graphs. Show that the competitive ratio of this algorithm is at least $\Omega(n)$.

b. (10 points) Now consider a randomized version of the above algorithm. Pick an edge uniformly at random and remove it. Now run the double cover algorithm on the resulting line. Show that the expected competitive ratio of this algorithm is $O(k)$.

Problem 2. (15 points) You are given a line graph on n vertices, and at each time a request in the form of an interval in this line arrives. The goal is to pick a maximum cardinality subset of intervals which are mutually disjoint (i.e., do not share an edge).

a. (10 points) Consider a special case when there is an integer ℓ such that all intervals have length in the range $[2^\ell, 2^{\ell+1})$ (length of an interval is the number of edges in it). Give a constant-competitive algorithm in this case.

   We follow a greedy strategy. We maintain a S set of disjoint intervals. Initially S is empty. When a new interval arrives, we check if intersects with any interval in S. If it does, we do not select it; otherwise we select it and add it to S. The set S is our solution. Show the the greedy algorithm is constant competitive for this special instance.

b. (5 points) Use the above to give a randomized algorithm for arbitrary instances with competitive ratio $\mathcal{O}(\log n)$

Problem 3. (15 points) Recall the fractional set-cover problem done in class. We have a universe of elements $U = \{e_1, e_2, \cdots e_n\}$ and a family of subsets $S_1, S_2 \cdots S_m$ of U. The elements are arriving online. At each round t, the online algorithm needs to be maintain fractional values x_S for each set S in the family such that for any element e arrived so far $\sum_{S:e\in S} x_S \geq 1$. The goal is to maintain this while minimizing $\sum_S x_S$. In class, we designed an $\mathcal{O}(\log m)$ competitive algorithm for this problem.

a. (10 points) Modify that algorithm and show that one can actually prove a competitiveness of $\mathcal{O}(\log f_{\max})$ where $f_{\max}$ is the *maximum* number of subsets to which any element in U can belong.

b. (5 points) Design an online rounding scheme for the integral version of the above problem and show that the overall competitive ratio is $\mathcal{O}(f_{\max} \log f_{\max})$. (**Hint:** You do not need randomized rounding any more. Think of when you can safely round a variable to 1 so that the cost is not too much and the rounded solution makes sure that you always have a feasible cover. If it makes thinking easier, you can assume that the algorithm knows the value of $f_{\max}$.)

Problem 4. (20 points) Consider the following cat and mouse game. You are given a *complete* graph on n vertices. At time 0, the cat and mouse are at different vertices. Now imagine a chasing game as follows. At any time $t, 1 \leq t \leq T$, both the cat chooses a vertex c_t *without knowing the current location of the mouse*. The mouse sees c_t and chooses a vertex $m_t \neq c_t$ as its location in order to avoids the cat. Note that at any iteration, both the cat and the mouse might choose to stay at their respective previous locations. The goal is to design an online strategy for the mouse so that it can always avoid the cat (that is $m_t \neq c_t$ for any t) while *minimizing* its total movement, where, the cost of movement is 1 if and only if $m_t \neq m_{t-1}, t = 1, 2, \cdots T$.

a. (5 points) Design a deterministic algorithm which is n-competitive.

b. (5 points) Show that no deterministic algorithm can be better than n-competitive.

c. (10 points) Design a randomized $\mathcal{O}(\log n)$-competitive algorithm

(**Hint**: The main thing which you have to do is relate this to a problem you have seen in class. Once you do that, the rest should not be very difficult).

Problem 5. (10 points) The online vertex cover problem is defined as follows. We are given a graph $G = (V, E)$ which you can assume you know fully. At any time step t, one $e_t \in E$ arrives. The task of the online algorithm is to maintain a subset of vertices S such that every edge arrived so far has at least one of its endpoints in the set. Consider the following algorithm for online (unweighted) vertex cover. Let S denote the vertex cover maintained by the algorithm. When a new edge $e = (u, v)$ arrives, and the set S does not contain either u or v, we pick one of the vertices in $\{u, v\}$ uniformly at random, and add it to S. Prove the the expected competitive ratio of this algorithm is 2.

(**Hint:** One of the end-points of every edge must be in the optimal solution. The solution to this problem can be found online. So you are not allowed to look up the internet. If I realize you have done this, you will get a zero in this entire homework.)