Comp 7720 - Online Algorithms

Assignment 4: Bin Packing (cntd.) & Graph Algorithms

Shahin Kamalli
University of Manitoba - Fall 2019

Due: December 2nd at 23:59pm

‘If the hurt comes, so will the happiness - be patient ...’ Rupi Kaur

Please pay attention to the followings when preparing/submitting your assignment:

• All problems are written problems. There are four problems with 35 + 10 marks in total.
• If there is a result in the slides that can be used in your answers, you are welcome to refer to that.
• If you have any question related to the assignment, you are encouraged to post it on Piazza. Note that you can submit anonymously (in case you are shy). Do not post any answer or hint on Piazza (for the obvious reason).
• You are welcome to discuss the problems with your friends. But you should write your answers individually. Please pay attention to this rule; it is important.
• Submit your answers electronically using Crowdmark.

Problem 1 Renting Servers in the Cloud [6 marks]

In the class, we visited Move-To-Front algorithm for renting servers in the cloud. In fact, any list-update algorithm $A$ can be used to choose the next bin to place an item. For that, one can arrange bins in a list. Upon arrival of an item $x$, place the item in the first bin in the list which has enough space. If no such bin exists, open a new bin. After placing an item into a bin, re-arrange the list using algorithm $A$ (a newly opened bin is considered an item which is accessed, for the first time, at the end of the list).

Consider the following instance of server renting problem (note that no item arrives at time 3).

$$\sigma = \langle (0.9, 1, 9), (0.5, 2, 9), (0.7, 4, 9), (0.35, 5, 10), (0.05, 6, 9), (0.11, 7, 9), (0.15, 8, 10) \rangle$$

(a) Trace TimeStamp algorithm for serving the above sequence. You need to show the packing of the algorithm at the following time steps: I) after placing the forth item (of size 0.35). II) after placing the sixth item (of size 0.11). III) after packing the whole sequence.

No justification is needed; just show the packings.

(b) Indicate the cost for each bin, and the total renting cost associated with serving $\sigma$ using TimeStamp algorithm.
Problem 2 ReserveCritical Analysis [14 + 7 marks]

(a) Prove that ReserveCritical has the claimed competitive ratio of 1.5.

**Hint:** consider two separate cases based on whether a tiny bin exists in the final packing of the algorithm, and prove the upper bound for each case separately.

(b) [Bonus] Consider a generalization of the ReserveCritical algorithm with a parameter $\alpha \in (1/2, 1)$. Huge items are defined as those larger than $\alpha$, critical items are those in $(1/2, \alpha]$, small items are in $(1 - \alpha, 1/2]$, and tiny items are those smaller than $1 - \alpha$. The algorithm treats items in each class as ReserveCritical does, that is, huge items are placed in their own bins, small items in pairs, critical items in a reserved space of size $\alpha$, and tiny items are placed using First-Fit in the non-reserved space of critical bins (and if required in tiny bins). Note that for $\alpha = 2/3$, we get the same algorithm as discussed in the class. Provide an upper bound for the competitive ratio of the generalized ReserveCritical as a function of $\alpha$.

Problem 3 Online Edge Coloring [5 marks]

In the class, we saw a lower-bound of $\frac{2\Delta - 1}{\Delta} \approx 2$ for competitive ratio of any online algorithm, that is, we proved that the number of colors used by an online algorithm cannot be less than twice that of Opt in the worst case.

In case of lower bounds (negative results) like this for graph problems, researchers tend to consider specific graph families. One might say it is true that we cannot do better than 2 times Opt for general graphs; but what about specific graph families? This is a standard direction of research for graph problems.

A clever student, named Bob, claims that there is an online algorithm with competitive ratio strictly better than $\frac{2\Delta - 1}{\Delta + 1}$ when the input graph is a complete bipartite graph (that is, after all edges have arrived, the graph is a complete bipartite graph). Prove or disprove Bob’s claim.

Problem 4 Renting Servers with Advice [10 marks]

Use a reduction from Binary Guessing problem to make a statement like the following:

In order to achieve a competitive ratio better than $X$ for renting servers in the cloud, advice of size $\Omega(n)$ is required.

To get the full mark, you should indicate how the instance of the renting-servers-in-the-cloud is formed from a given binary string, and find the lower bound for competitive ratio (that is, the value of $X$ in the above statement).

**Hint:** in your reduction, all items can have size 0.5, and at some timesteps, no item might arrive (as in Problem 1). You might assume $\mu$ is an integer larger than 3.