

CS381 Assignment 5

Due November 17 2023 12:00 PM AoE

Most of the questions of this assignment have to do with the *Online Minimum Weight Metric Matching Problem*, ie the parking problem from class. In this problem, the algorithm oversees n parking spots, which are used to handle incoming visitors. Sequentially, visitors arrive and must be immediately matched to an available parking spot. The goal is to match these spots in such a way that visitors have to walk as little as possible to their intended location from where they park. The cost of an assignment is the sum of distances from each driver's assigned parking spot to their intended location. We rank the performance of algorithms by analyzing their competitive ratio, where the competitive ratio of an algorithm A is given by

$$\max_{\text{instances } I} \frac{|A(I)|}{|\text{OPT}(I)|},$$

where $|A(I)|$ denotes the cost of A 's output on I and $|\text{OPT}(I)|$ denotes the cost of the optimal solution on I .

In the dynamic pricing version of the game, instead of assigning drivers to parking spots, the algorithm can instead place prices over the servers. In this model, we assume incoming drivers choose spots that minimize the sum of the price of the spot and the distance from the spot to their intended location. Sequentially, after each spot is taken, prices may be updated as desired by the algorithm.

1. Recall the algorithm HARMONIC defined for inputs restricted to a single street (modeled via the real line): if a driver arrives between two spots s_L and s_R with distance d_L to the left and d_R to the right, then assign the driver to the spot on the left with probability $\frac{d_R}{d_L+d_R}$ and to the right with probability $\frac{d_L}{d_L+d_R}$. First, **do either (a) or (b)**:
 - (a) Prove that HARMONIC has competitive ratio $\Omega(n)$ ¹. Hint: do the exact same sequence that we showed in class the GREEDY algorithm is $2^k - 1$ competitive on.
 - (b) Show experimentally that HARMONIC has competitive ratio $\Omega(n)$. Hint: do the exact same sequence that we showed in class the GREEDY algorithm is $2^k - 1$ competitive on.
2. Read the abstract and sections 1 and 2 of this paper: <https://arxiv.org/pdf/1504.01093.pdf> **Everyone needs to read it, it's 4 pages**. For **either** the problem metrical task systems or the k -server problem, describe the problem in plain English and work through an example of your choosing using the natural GREEDY algorithm and determine any lowerbound on its competitive ratio.

¹Note that Ω is just the corresponding version of O for lowerbounds. So $\Omega(n)$ just means that the function is *at least* linear.

3. In the above definition of the *Online Minimum Weight Matching Problem*, we defined the objective cost of an assignment to be the sum of distances between each drivers' assigned spot to their intended location. This is called the *utilitarian* objective: we are trying to make everyone happy in a somewhat equal way. Suppose, instead, I measure the objective cost of an assignment by only taking the maximal distance any one person walks: this is called the *egalitarian* objective. One way of thinking about this is that I'm trying to minimize the unhappiness of the unhappiest player. Let's see how this objective changes things:

- (a) Show that the GREEDY algorithm is $\Omega(2^n)$ competitive.
- (b) Show that the PERMUTATION algorithm is $\Omega(n)$ competitive. Hint: stick with the line: start with parking spots at $1, 2, \dots, n$, have drivers arrive from left to right. Make sure they arrive in such a way that the maximal distance anyone walks is small *until the final driver*.
- (c) When we thought about general lowerbounds for the utilitarian objective, we realized the following:
 - For any deterministic algorithm, we can come up with an instance on which it is at least linearly competitive.
 - For any randomized algorithm, we can come up with an instance on which it is at least logarithmically competitive.

So now let's try and do the same. In fact, it'll turn out that any randomized algorithm must also be linearly competitive².

- i. Imagine the following instance: the parking spots are at $1, 2, \dots, n$. The first $n - 1$ drivers intend to go to $.5, 1.5, 2.5, \dots, n - .5$. Let A be the parking spots from 1 to $\frac{n}{2}$ (rounded down, if n is odd) and let B be the rest. Since $n - 1$ drivers have arrived, there must be exactly one parking spot left. Convince me that at least one of A and B have that final spot with probability at least $\frac{1}{2}$.
 - ii. Based on whichever of A and B is more likely to have that final spot, where should I put that final driver to make the algorithm look bad?
 - iii. Put it together: convince me that for any (randomized) algorithm, I can specify an input on which it is at least linearly competitive in expectation.
4. **Extra Credit:** In class, we said an algorithm for the online minimum weight matching problem on a tree is mimicable with prices if and only if it is *monotone*. Recall that for deterministic algorithms, monotonicity just means that if a driver arriving at a particular location v would be assigned to a particular spot s , then if that driver had instead arrived at any location on the path from v to s it still would have been assigned to s . I am *pretty sure* that the PERMUTATION algorithm defined in class isn't monotone, but to date it hasn't been confirmed yet. This is probably not "automatic A" level of extra credit, but I'd be happy to help you publish the result.

²This proof is actually my first ever publication, which I did my Junior(? maybe Senior) year of undergrad.