1 Further results

1. We looked at online matching for bipartite graphs where the vertices arrive. We found that (a) deterministic algorithms always have competitive ratio $\leq 2$, (b) there is a randomized algorithms with competitive ratio $\frac{e-1}{e}$, and (c) $\frac{e-1}{e}$ is the best one can do. What about general graphs? What if edges arrive? Gamlash et al. [?] showed that (a) for vertex arrivals in general graphs there is a randomized algorithm with competitive ratio $(\frac{1}{2} + \Omega(1))$, and (b) for edge arrivals randomization does not help.

2. Role-matchmaking is a problem where players of different skills levels arrive and must be assigned to a team as soon as they arrive. The goal is to have the teams be balanced so that no team dominates. This can get very complicated since different skills is not 1-dimensional. For example, in soccer a team may need a good Goalkeeper more than a great midfielder. This problem has immediate applications to many popular online videogames where such as League of Legends and Dota 2. Alman & McKay [?] view this as a dynamic data structers problem. The show (a) assuming the 3SUM conjecture, any data structure for this problem requires $n^{1-o(1)}$ time per insertion or $n^{2-o(1)}$ time per query, and (b) there is an approximation algorithm that takes $O(\log n)$ per operation.

2 Chapter suggestions

1. p330 Theorem 18.4.3 uses a different definition of competitive ratio than what is given at the beginning of the chapter. I believe these are reciprocals of each other. A suggestion would be to stick with the convention being used in the Theorem since it seems to be most commonly used in current research.

2. p327: Under Yao’s Lemma. The second enumerated item has $\max_{x \in A}$ but it should read $x \in X$. Same thing below in the lemma environment.

   (a) backwards quote right under the Lemma at the top of p328. Should have a “the worst input in . . . .

   (b) Suggested edit for the intuition of Yao’s Lemma: cost under the worst input in $p \geq$ cost of the best deterministic algorithm w.r.t.
$p$. Note the case used for the distribution $p$. Also, the brief explanation should mention we are comparing costs. Otherwise it isn’t clear if the $\leq$ means less cost or better.