Online Bipartite Matching in Offline Time[⋆] (Abstract)

Piotr Sankowski

University of Warsaw, Warszawa, Poland
sank@mimuw.edu.pl

Abstract. I will present our (with Bartłomiej Bosek, Dariusz Leniowski and Anna Zych) recent results on the problem of maintaining maximum size matchings in incremental bipartite graphs (FOCS'14). In this problem a bipartite graph G between n clients and n servers is revealed online. The clients arrive in an arbitrary order and request to be matched to a subset of servers. In our model we allow the clients to switch between servers and want to maximize the matching size between them, i.e., after a client arrives we find an augmenting path from a client to a free server. Our goals in this model are twofold. First, we want to minimize the number of times clients are reallocated between the servers. Second, we want to give fast algorithms that recompute such reallocation.

As for the number of changes, we propose a greedy algorithm that chooses an augmenting path π that minimizes the maximum number of times each server in π was used by augmenting paths so far. We show that in this algorithm each server has its client reassigned $O(\sqrt{n})$ times. This gives an $O(n^{3/2})$ bound on the total number of changes, what gives a progres towards the main open question risen by Chaudhuri *et al.* (INFOCOM'09) who asked to prove $O(n \log n)$ upper bound. Next, we argue that the same bound holds in the decremental case. Moreover, we show incremental and decremental algorithms that maintain $(1-\varepsilon)$-approximate matching with total of $O(\varepsilon^{-1}n)$ reallocations, for any $\varepsilon > 0$.

Finally, we address the question of how to efficiently compute paths given by this greedy algorithm. We show that by introducing proper amortization we can obtain an incremental algorithm that maintains the maximum size matching in total $O(\sqrt{n}m)$ time. This matches the running time of one of the fastest static maximum matching algorithms that was given by Hopcroft and Karp (SIAM J. Comput '73). We extend our result to decremental case where we give the same total bound on the running time. Additionally, we show $O(\varepsilon^{-1}m)$ time incremental and decremental algorithms that maintain $(1-\varepsilon)$-approximate matching for any $\varepsilon > 0$. Observe that this bound matches the running time of the fastest approximate static solution as well.

[⋆] This work was partially supported by ERC StG project PAAl 259515, FET IP project MULTIPEX 317532, NCN grant 2013/11/D/ST6/03100 and N206 567940.

G.F. Italiano et al. (Eds.): SOFSEM 2015, LNCS 8939, p. 45, 2015.