MATCHING IN DYNAMIC GRAPHS

MANOJ GUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI

DECEMBER 2014

MATCHING IN DYNAMIC GRAPHS

by
Manoj Gupta
Department of Computer Science and Engineering

Submitted
in fulfilment of the requirements of the degree of
Doctor of Philosophy
to the

Indian Institute of Technology Delhi
December 2014

Certificate

This is to certify that the thesis titled **"Matching in Dynamic Graphs"** being submitted by **Mr. Manoj Gupta** to the Indian Institute of Technology Delhi, for the award of the degree of Doctor of Philosophy, is a record of bonafide research carried out by him under our supervision. The results obtained in this thesis have not been submitted to any other university or institute for the award of any other degree or diploma.

Dr. Sandeep Sen
Professor
Dept. of Computer Science and Engineering
IIT Delhi
Delhi 110016

Dr. Surender Baswana
Associate Professor
Dept. of Computer Science and Engineering
IIT Kanpur
Kanpur 208016

December 2014, New Delhi

Acknowledgements

I am indebted to my thesis advisor Sandeep Sen and Surender Baswana for my thesis. I started my research work while doing M.Tech at IIT Kanpur under the guidance of Surender Baswana. He initiated me to research problem which eventually let to this thesis. I wish to thank him for giving me full support right through my thesis work. I also wish to thank Sandeep Sen for constantly giving me support - both academic and monetary - during my stay at IIT Delhi. I have always looked to Naveen Garg if I faced any bureaucracy hurdle at IIT Delhi and he always gave his valuable time to listen to me.

I wish to thank Microsoft Research India for awarding me Microsoft Ph.D Research Fellowship. I interned twice at Microsoft Research under Navin Goyal and I wish to thank him for initiating me to a beautiful problem which I have still not managed to solve. I also wish to thank IMPECS (Indo-German Max Planck Center for Computer Science) for sponsoring my trip to Max Planck Institute (MPI) in Saarbrucken, Germany. This trip provided me the opportunity to meet new people and talk about beautiful problems. I also wish to thank Richard Peng with whom I worked online (using Skype) on a problem. His enthusiasm was pivotal in solving that problem.

I wish to thank my colleagues of the research scholar room for listening to my vague and useless talk for these four years. This includes, but is not limited to, Nisha Jain, Swati Sharma, Syamantak Das, Shibashis Guha, Chinmay Narayan, Anuj Gupta, Arindam Pal, Anamitra Roy Choudhary, Yamuna Prasad.

During my stay at Kanpur and Delhi, I have been very lucky to view gems of cinema. I wish to thank the masters of cinema for educating me via these films. This includes, but is not limited to, Akira Kurosawa, Garry Bardin, Ritwik Ghatak, Andrei Tarkovsky, Elem Klimov, Shyam Benegal and others.

Last, but not the least, I wish to thank my parents, for supporting me right through my study days.

Abstract

We consider the problem of maintaining matching in a graph under addition and deletion of edges. We show the following results:

- *Unweighted Graph*

    1. *Maximal Matching*

       We design a randomized algorithm [5] that maintains a maximal matching and takes expected amortized $O(\log n)$ time for each edge update. While there is a trivial $O(n)$ algorithm for edge update, the previous best known result for this problem was due to Ivković and Llyod [22]. For a graph with n vertices and m edges, they give an $O((n+m)^{0.7072})$ update time algorithm which is sublinear only for a sparse graph. For the related problem of maximum matching, Onak and Rubinfeld [27] designed a randomized data structure that achieves amortized $O(\log^2 n)$ time for each update for maintaining a c-approximate maximum matching for some unspecified large constant c.

    2. $(1+\epsilon)$-approximate matching

       We present the first data structures [18] that maintains a near optimal maximum cardinality matching on sparse graphs in sublinear (in m) time per update. Our main result is a data structure that maintains an $(1+\epsilon)$ approximation of maximum matching under edge insertions/deletions in worst case $O(\sqrt{m}\epsilon^{-2})$ time per update. This improves the $3/2$ approximation algorithm designed by Neiman and Solomon [26] which runs in similar time.

- *Weighted Graph*

    1. *4.9108-approximate matching*

       Using the maximal matching algorithm in unweighted dynamic graph, we design an algorithm [3] which maintains an expected 4.9108 approximation of maximum weighted matching under addition/deletion of edges. The algorithm achieves an expected amortized $O(\log n \log N)$ time per edge insertion or deletion, where the weight of an edge in the graph is in the range [1,N]. We design a deterministic version of the above algorithm,i.e., an algorithm that maintains a $4.9108+\epsilon$ approximate maximum weight matching in $O(\log n \log N/\epsilon)$ worst case update time.

    2. $(3+\epsilon)$ *approximate matching*

       Using the $(1+\epsilon)$ approximate matching in unweighted dynamic graph, we design an algorithm [18] which maintains an $(3+\epsilon)$ approximate maximum weighted

matching under addition/deletion of edges. For any $\epsilon < 1/2$, our algorithm takes $O(\sqrt{m} \log N \epsilon^{-3})$ time per update in the worst case.

3. $(1 + \epsilon)$ *approximate matching*

   We design an algorithm [18] that maintains an $(1+\epsilon)$-approximate maximum weighted matching under addition/deletion of edges. For any $\epsilon < 1/2$, our algorithm takes $O(\sqrt{m}\epsilon^{-2-O(1/\epsilon)} \log N)$ time per update in the worst case. Note that the algorithm has an exponential dependence on $(1/\epsilon)$ in its running time while the $(3 + \epsilon)$ approximate algorithm has none.

Contents