Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 89–98

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

Fully-Dynamic Graph Algorithms with Sublinear Time Inspired by Distributed Computing

Leonid Barenboim and Tzalik Maimon*

Open University of Israel
leonidb@openu.ac.il ; tzali.tm@gmail.com

Abstract

We study dynamic graphs in the fully-dynamic *centralized* setting. In this setting the vertex set of size n of a graph G is fixed, and the edge set changes step-by-step, such that each step either adds or removes an edge. The goal in this setting is maintaining a solution to a certain problem (e.g., maximal matching, edge coloring) after each step, such that each step is executed efficiently. The running time of a step is called *update-time*. One can think of this setting as a dynamic network that is monitored by a central processor that is responsible for maintaining the solution. Currently, for several central problems, the best-known deterministic algorithms for general graphs are the naive ones which have update-time $O(n)$. This is the case for maximal matching and proper $O(\Delta)$-edge-coloring. The question of existence of sublinear in n update-time deterministic algorithms for dense graphs and general graphs remained wide open. We address this question by devising sublinear update-time deterministic algorithms for maximal matching in *graphs with bounded neighborhood independence* $o(n/\log^2 n)$, and for proper $O(\Delta)$-edge-coloring in *general graphs*. The family of bounded neighborhood independence is a very wide family of dense graphs that represent very well various networks. For graphs with constant neighborhood independence, our maximal matching algorithm has $\tilde{O}(\sqrt{n})$ update-time. Our $O(\Delta)$-edge-coloring algorithms has $\tilde{O}(\sqrt{\Delta})$ update-time for general graphs.

In order to obtain our results we employ a novel approach that adapts certain distributed algorithms of the $\mathcal{LOCAL}$ setting to the centralized fully-dynamic setting. This is achieved by optimizing the work each processors performs, and efficiently simulating a distributed algorithm in a centralized setting. The simulation is efficient thanks to a careful selection of the network parts that the algorithm is invoked on, and by deducing the solution from the additional information that is present in the centralized setting, but not in the distributed one. Our experiments on various network topologies and scenarios demonstrate that our algorithms are highly-efficient in practice. We believe that our approach is of independent interest and may be applicable to additional problems.

Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: Dynamic Graphs, Large-Scale Networks, Simulation, Maximal Matching, Coloring

*This research has been supported by the Open University of Israel's Research Fund, and by the Israeli Academy of Science, grant 724/15. Part of this work has been performed while Leonid Barenboim was affiliated with the joint program of the Simons institute at UC Berkeley and Weizmann institute of science.

1877-0509
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.098

1 Introduction

1.1 Problems and Results

We consider the Dynamic Graphs (centralized) setting in which the number of vertices n is fixed, and in each step an edge is either removed or added to a graph $G = (V, E)$. Thus, the number of edges m changes in each step. This setting attracted a lot of attention recently, due to its relevance to various fields such as communication networks, molecular biology (e.g., metabolic network modeling), neural networks, VLSI design, and computer graphics, in which graphs change rapidly. In the field of networks, for instance, a central processor may have to monitor the entire network for changes, and update a certain solution accordingly after each change. Sublinear-time algorithms are crucial in this context. Numerous graph problems have been studied in this setting. Notable examples include maximal matching and edge coloring. In particular, they are useful for link scheduling in wireless networks that may change frequently.

A *matching* is a subset of edges $M \subseteq E$, such that no vertex of V belongs to more than one edge in M. Vertices that do not belong to edges of M are called *free vertices*. A *maximal matching* is a matching M, such that there is no edge in $E \setminus M$ with both its endpoints free. (It is well known that a maximal matching is not necessarily a maximum cardinality matching, but it is a 2-approximation of it.) The goal of the dynamic maximal matching problem is maintaining a maximal matching after each step of addition or removal of an edge, where initially the graph consists of n vertices and no edges. A *centralized dynamic algorithm* defines the sequential operations that have to be performed on addition and removal of an edge in order to preserve the desired solution. The running time[1], called *update-time*, is the maximum of the running times of an addition operation and a removal operation.

The dynamic maximal matching and approximate maximum matching problems have been very intensively studied in the last years [5, 7, 8, 12, 15, 16] (SODA'2016 alone has four papers on the subject). Nevertheless, no deterministic centralized algorithm for maximal matching in general graphs that beats the naive[2] update-time of $O(n)$ is currently known. (For randomized algorithms better running times are known, and this problem can be solved with $O(\log n)$ expected amortized update-time [5]. Such algorithms, however, are less suitable when every step has to be executed as efficiently as possible.) In STOC'2013 Neiman and Solomon [15] achieved a significant progress by devising a deterministic dynamic maximal matching algorithm with update-time $O(\sqrt{m})$, i.e., obtaining a sublinear in n time for sparse graphs. Since then, several papers studied the problem on sparse graphs [7, 16]. However, the question of existence of sublinear in n deterministic algorithm for general graphs, and in particular for dense graphs remains wide open.

In the current paper we make a major step towards settling this question, and devise deterministic algorithms with sublinear in n update-time, for a very wide family of dense graphs. Specifically, our algorithms are applicable to graphs with *bounded neighborhood independence*. A graph has neighborhood independence c if each vertex has at most c independent neighbors (i.e., at most c neighbors with no edges between them). A family of constant neighborhood independence includes line graphs, unit disk graphs, unit ball graphs, graphs of bounded growth, and many other families of graphs. Note that there are graphs in these families that have

[1] All running times mentioned in the current paper are worst-case, unless stated otherwise. In particular, the running times of all our algorithms are worst-case running times.

[2] The naive maximal matching algorithm works as follows. On addition of an edge, it is added to the matching if its endpoints are free. On removal of an edge that belongs to the matching, its endpoints become free, and a search for substitutions is performed on edges adjacent on the two endpoints. Consequently, at most two new edges may be added to the matching. On removal of an edge that does not belong to the matching, no additional operations are performed.