

Abstract

The Dynamic Minimum Spanning Tree Problem (DMSTP) is that of maintaining a minimum spanning tree (MST) of a dynamically changing graph, where these changes (or operations) can be insertions and deletions of vertices, insertions or deletions of edges, and modifications of edge weights. The problem is said to be *fully dynamic* if both insertion and deletion operations are allowed (or if the edge weights can increase or decrease). Otherwise, the problem is said to be *partially dynamic* or *semi dynamic* if only one kind of operation is allowed (either edge deletions or insertions, either weight increases or decreases). Also, the problem is said to be *on-line* if the dynamic changes must be processed in real time (i.e. there is no preprocessing and updates are performed one by one).

The study of dynamic graph algorithms, in particular those for maintaining a minimum spanning tree of a dynamically changing graph, is motivated by both practical and theoretical reasons. Dynamic algorithms and data structures can be used in a wide range of real-life problems, e.g. in network-related problems (computer, telephony and cable-TV networks), metaheuristics and local search heuristics.

In this work, we make a step toward the experimental evaluation of algorithms to update a minimum spanning tree after edge weight changes. Such algorithms are particularly helpful in the implementation of metaheuristics and local search heuristics for solving broadcast optimization and design problems in communication networks, similar to the algorithms involving dynamic shortest path problems studied by Buriol et al. [6, 7, 8, 9] in the context of the weight setting problem in OSPF/IS-IS routing. Complementary, we propose and evaluate both a new algorithm and a new data structure specifically designed for the edge weight updating variant of the DMSTP. The new algorithm is quite simple to implement and can be used with any data structure for dynamic trees representation.

Keywords: Minimum spanning trees, minimum weight spanning trees, dynamic graphs, experimental analysis of algorithms, computational complexity.