Problem Summary: The *arboricity* of a graph G, denoted $a(G)$, is defined to be the minimum number of edge-disjoint forests into which G can be decomposed. While efficient algorithms exist to compute arboricity both on general graphs and on special classes of graphs, nearly all are designed to operate on static graphs. Do efficient algorithms exist to maintain graph arboricity and/or minimal forest decompositions on dynamic graphs?

Previous Work: Surprisingly little work has been done in this area. In the literature, arboricity is often examined in conjunction with special orientations of undirected graphs. Define a *d-bounded orientation* of G to be an orientation in which all vertices of G have out-degree at most d. It is easily shown that any graph G has an $a(G)$-bounded orientation [2]. It is slightly less straightforward to show that if G has a d-bounded orientation, then we can compute, in $\mathcal{O}(m)$ time, a decomposition of G into $2d-1$ forests [2]. Therefore, graphs with bounded arboricity also have d-bounded orientations for bounded d, and vice-versa. If we only wish to compute loose bounds on the arboricity, it thus suffices to maintain orientations with small maximum out-degree.

Brodal and Fagerberg in [1] present a set of algorithms to maintain a Δ-bounded orientation for a dynamic graph G, where Δ is bounded by some small constant multiple of $a(G)$; these algorithms can be used to approximate $a(G)$ within roughly a factor of 4. They achieve $\mathcal{O}(1)$ amortized time for edge insertions and $\mathcal{O}(c + \log n)$ amortized time for edge deletions. In [2], Eppstein provides algorithms to compute a 2-approximation for $a(G)$ in $\mathcal{O}(m)$ time for static graphs. The fastest known algorithm for computing $a(G)$ exactly, rather than approximately, runs in $\mathcal{O}(a(G)n\sqrt{m + a(G)n\log n})$ time [3]. Faster algorithms are known for special classes of graphs; most notably, planar graphs can be decomposed into three forests in $\mathcal{O}(n \log n)$ time [4].

Open Problems: Several open questions naturally arise from the existing work. The first of the following was posed by Brodal and Fagerberg; the others do not seem to appear in the literature.

- Can Brodal and Fagerberg's algorithms be modified to provide equivalent *non-amortized* time bounds? Can they be improved to handle edge deletions in $\mathcal{O}(1)$ (possibly amortized) time?

- The algorithms in [1] maintain a Δ-bounded orientation of the graph, for some fixed $\Delta \geq 2a(G) + 2$. Eppstein shows in [2] shows that a d-bounded orientation of some graph G can be used to produce a decomposition of G into $2d$ forests in $\mathcal{O}(m)$ time. Can such a forest decomposition be maintained dynamically with update time sublinear in m? Alternatively, can we dynamically maintain a forest decomposition (or bounded orientation) that *better* approximates the arboricity?

- Brodal and Fagerberg prove that maintaining a forest decomposition of G with size $a(G)$ would require $\Omega(n/a(G)^2)$ edge operations, even considering amortized algorithms; however, achieving this bound would likely yield a significant improvement over recomputing the arboricity from scratch. Can this bound be achieved? Furthermore, they prove that an $(a(G)+1)$-bounded orientation of G can be maintained using a logarithmic number of edge reorientations per update; could an *approximately* optimal forest decomposition be maintained efficiently?

References:

[1] Gerth Stølting Brodal and Rolf Fagerberg. *Dynamic Representations of Sparse Graphs.* In Proc. 6th International Workshop on Algorithms and Data Structures, volume 1663 of Lecture Notes in Computer Science, 342-351. Springer Verlag, Berlin, 1999.

[2] David Eppstein. *Arboricity and Bipartite Subgraph Listing Algorithms.* Information Processing Letters, 51(4):207-211, 1994.

[3] Harold N. Gabow and Herbert H. Westermann. *Forests, Frames, and Games: Algorithms for Matroid Sums and Applications.* Algorithmica, 7:465-497, 1992.

[4] Grossi and Lodi. *Simple Planar Graph Partition into Three Forests.* Discrete Applied Mathematics, 84:121 - 132, 1998.