Abstract

We study the problem of computing a minimum cut in a simple, undirected graph and give a deterministic $O(m \log^2 n \log\log^2 n)$ time algorithm. This improves both on the best previously known deterministic running time of $O(m \log^{12} n)$ (Kawarabayashi and Thorup STOC'2015) and the best previously known randomized running time of $O(m \log^3 n)$ (Karger STOC'1996) for this problem, though Karger's algorithm can be further applied to weighted graphs. Our approach is using the Kawarabayashi and Thorup graph compression technique, which repeatedly finds low-conductance cuts. To find these cuts they use a diffusion-based local algorithm. We use instead a flow-based local algorithm and suitably adjust their framework to work with our flow-based subroutine. Both flow and diffusion based methods have a long history of being applied to finding low conductance cuts. Diffusion algorithms have several variants that are naturally local while it is more complicated to make flow methods local. Some prior work has proven nice properties for local flow based algorithms with respect to improving or cleaning up low conductance cuts. Our flow subroutine, however, is the first that is both local and produces low conductance cuts. Thus, it may be of independent interest.