Randomized Minimum Cut Problem

Group members

Md. Taki Tahmid Saad – 2207022
Saleh Sadid Mir – 2207024
Ahmed Kaif – 2207025

Report:

This algorithm solves the minimum cut problem , which is used to determine the smallest set of edges that disconnects an undirected graph into two fragments. The input is an undirected, unweighted (sometimes with parallel edges) connected graph. The algorithm is a randomized approximation method that may not always return the exact minimum cut. The core idea is contracting edge that means , repeatedly pick a random edge (using rand function) and merge its endpoints into a new combined supernode. Before combining the nodes to form a supernode, Disjoint set Union is used to find respective parents. If parents aren't similar, they are combined. After each contraction, self-loops should be removed but parallel edges are kept, reducing the number of vertices by one. The process continues until there are only two supernodes remaining in the graph and the number of edges between these two supernodes represents a cut. A single run has low probability of finding the true minimum cut, so the algorithm is repeated many times to increase accuracy. Compared to deterministic max-flow based methods, this approach is simpler and faster for large graphs.

Discussion:

This algorithm solves the minimum cut problem using a randomized approach.
It repeatedly selects a random edge and contracts its two vertices into a single vertex, removing self-loops while keeping parallel edges. This contraction process continues until only two vertices remain in the graph. The number of edges between these two remaining vertices represents a possible cut value. Since the algorithm is randomized, it is run multiple times, and the smallest cut found is taken as the minimum cut.