Lecture Scribe Notes: Randomized Min-Cut Algorithm

Course: CSE400 – Fundamentals of Probability in Computing
Lecturer: Dhaval Patel, PhD

February 5, 2026

1 Introduction to the Min-Cut Problem

1.1 Definitions and Terminology

Cut-set: A set of edges in a graph whose removal breaks the graph into two or more connected components. +2

Min-Cut Problem: Given a graph $G = (V, E)$ with n vertices, the problem is to find a cut-set with minimum cardinality. +1

Edge Contraction: The fundamental operation used in min-cut algorithms. It involves:

- Removing an edge (u, v) from the graph. +3
- Merging the two vertices u and v into a single vertex. +1
- Eliminating all edges connecting u and v (no self-loops).
- Retaining all other edges, which may result in parallel edges.

1.2 Applications

Min-cut algorithms are utilized for network connectivity, reliability, and optimization: +2

- **Network Design:** To find the minimum capacity cut and optimize network flow and communication efficiency. +1
- **Communication Networks:** To understand network vulnerability to failures and build fault-tolerant systems. +1
- **VLSI Design:** To partition circuits into smaller components, reducing interconnectivity complexity.

2 Max-Flow Min-Cut Theorem

The theorem establishes the relationship between flow and cuts in a network.

Statement: In a flow network, the maximum amount of flow passing from the source (S) to the sink (T) is equal to the total weight of the edges in a minimum cut. +2

Key Components:

- **Capacity of a cut:** The sum of capacities of edges oriented from a vertex in set X to a vertex in set Y.
- **Minimum cut:** The network cut with the smallest possible capacity.
- **Maximum flow:** The largest possible flow from S to T.

3 Deterministic Min-Cut: Stoer-Wagner Algorithm

3.1 Theorem for Deterministic Approach

Let s and t be two vertices of graph G. Let $G/\{s,t\}$ be the graph obtained by merging them. A minimum cut of G is the smaller of:

- A minimum s–t-cut of G (if the min-cut separates s and t).
- A minimum cut of $G/\{s,t\}$ (if the min-cut does not separate s and t).

+1

3.2 Stoer-Wagner Pseudocode

Algorithm 1: MinimumCutPhase(G, a)

- Start with set $A = \{a\}$.
- While $A \neq V$, add the most tightly connected vertex to A.
- Return the cut weight of the last phase.

Algorithm 2: MinimumCut(G)

- While $|V| \geq 1$:
- Choose any vertex a from V.
- Run MinimumCutPhase(G, a).
- If the cut-of-the-phase is lighter than the current minimum cut, store it as the new minimum cut.
- Shrink G by merging the two vertices added last in the phase.
- Return the minimum cut found.

4 Randomized Min-Cut: Karger's Algorithm

4.1 Rationale for Randomization

Randomized algorithms provide a probabilistic guarantee of success and can provide accurate estimates with fewer iterations. Benefits include efficiency, parallelization, approximation guarantees, and avoidance of worst-case instances. +3

4.2 Karger's Algorithm Mechanism and Example

The algorithm is sensitive to the initial choice of edges. If critical edges are contracted early, the algorithm may fail to find the true minimum cut. +1

Example of an Unsuccessful Run:

Input Graph: Vertices $\{0, 1, 2, 3\}$ with edges a, b, c, d, e. +2

Step 1: Pick edge b, remove it, and fuse corners into vertex $\{0, 2\}$. +1

Step 2: Pick edge d, remove it, and fuse corners into vertex $\{1, 3\}$. +1

Outcome: The output cut is $\{a, c, e\}$, which is not minimal.

Actual Minimal Cuts: Either $\{b, e\}$ or $\{a, d\}$.

4.3 Karger's Recursive Pseudocode

Algorithm 3: RECURSIVE-RANDOMIZED-MIN-CUT(G, α)

Input: Undirected multigraph G with n vertices; integer constant $\alpha > 0$.

Base Case: If $n \leq \alpha^3$, find min-cut via brute force.

Recursive Step:

- For $i = 1$ to α:
- Create multigraph G' by applying $n - \lceil n/\alpha \rceil$ random contraction steps.
- Recursively call the algorithm on G' to get cut C'.
- If $i = 1$ or $|C'| < |C|$, update $C = C'$.

Output: Cut C.

4.4 Success Probability Theorem

The algorithm outputs a min-cut set with a probability of at least:

$$\frac{2}{n(n-1)}$$

5 Comparison: Deterministic vs. Randomized

Feature	**Deterministic (Stoer-Wagner)**	**Randomized (Ka**
Guarantee	Always guarantees an exact minimum cut. +1	Provides an approximate min-cut
Time Complexity	$O(VE + V^2 \log V)$. +1	$O(V^2)$.
Scalability	May have higher complexity for large graphs. +1	Generally more efficient fo