Lecture Scribe: Randomized Min-Cut Algorithm

Course: CSE400 – Fundamentals of Probability in Computing
Instructor: Dhaval Patel, PhD
Based strictly on Lecture Slides and Textbook Context

1 Min-Cut Problem

1.1 Motivation: Why Use Min-Cut?

The minimum cut (min-cut) algorithm is used to solve problems related to:

- Network connectivity
- Network reliability
- Optimization problems in networks

These applications arise because the min-cut identifies the smallest set of edges whose removal disconnects the network.

Applications

(1) Network Design

The min-cut helps improve communication efficiency and optimize network flow. It is used to find the minimum capacity cut, which identifies bottlenecks in the network.

(2) Communication Networks

The min-cut helps understand network vulnerability to failures. It enables the construction of robust and fault-tolerant networks.

(3) VLSI Design

In Very Large Scale Integration (VLSI), the min-cut algorithm helps partition circuits into smaller components. This reduces interconnectivity complexity.

2 Definition of Cut and Minimum Cut

2.1 Graph Notation

Let:

$$G = (V, E)$$

Where:

- V = set of vertices
- E = set of edges
- $n = |V|$ = number of vertices

2.2 Definition: Cut-Set

A cut-set is defined as a set of edges whose removal breaks the graph into two or more connected components.

Thus, removing these edges disconnects the graph.

2.3 Definition: Minimum Cut (Min-Cut)

Given a graph $G = (V, E)$, the minimum cut problem is defined as finding a cut-set with minimum cardinality.

This means finding the smallest number of edges whose removal disconnects the graph.

2.4 Sensitivity of Randomized Min-Cut Algorithms

Min-cut algorithms such as Karger's algorithm are randomized and sensitive to edge selection.

If the algorithm contracts critical edges early, it may produce a different cut.

Thus, the algorithm's outcome depends on random choices.

3 Edge Contraction Operation

3.1 Definition: Edge Contraction

Edge contraction is the primary operation used in randomized min-cut algorithms.

Edge contraction removes an edge and merges its two endpoints into a single vertex.

3.2 Formal Description

When contracting an edge (u, v):

- Step 1: Merge vertices u and v into a single vertex
- Step 2: Remove the edge (u, v)
- Step 3: Retain all other edges
- Step 4: Remove self-loops
- Step 5: Parallel edges may remain

4 Successful and Unsuccessful Min-Cut Runs

Since randomized algorithms involve randomness, different runs produce different results.

4.1 Successful Min-Cut Run

A successful min-cut run occurs when the algorithm correctly identifies the minimum cut of the graph.

Characteristics:

- Critical edges are not contracted prematurely
- Final result equals actual minimum cut

4.2 Unsuccessful Min-Cut Run

An unsuccessful run occurs when the algorithm fails to identify the minimum cut.

Characteristics:

- Critical edges were contracted early
- Final cut is not minimum

5 Max-Flow Min-Cut Theorem

5.1 Theorem Statement

The Max-Flow Min-Cut theorem states:

> "In a flow network, the maximum amount of flow passing from the source to the sink is equal to the total weight of the edges in a minimum cut."

5.2 Definitions Related to Theorem

Capacity of a cut:

$$\text{Capacity of cut} = \sum \text{capacities of edges from } X \text{ to } Y$$

Where X and Y form the partition of vertices.

Minimum cut:

The cut with smallest possible capacity.

Minimum cut capacity:

Capacity of the minimum cut.

Maximum flow:

Largest possible flow from source S to sink T.

5.3 Result of Theorem

$$\text{Maximum Flow} = \text{Minimum Cut Capacity}$$

6 Deterministic Min-Cut Algorithm (Stoer–Wagner Algorithm)

6.1 Theorem Used

Let s and t be vertices of graph G.

Let $G/s,t$ be the graph obtained by merging s and t.

Then the minimum cut of G is the smaller of:

- Minimum s–t cut of G, or
- Minimum cut of $G/s,t$

6.2 Explanation

Case 1: Minimum cut separates s and t

Then minimum s–t cut is minimum cut.

Case 2: Minimum cut does not separate s and t

Then minimum cut exists in contracted graph $G/s,t$.

6.3 Pseudocode: MinimumCutPhase(G, a)

Step 1: Initialize

$$A \leftarrow \{a\}$$

Step 2: Repeat until $A \neq V$

Step 3: Add most tightly connected vertex to A

Step 4: Return cut weight

6.4 Pseudocode: MinimumCut(G)

Step 1: While $|V| \geq 1$

Step 2: Choose vertex a

Step 3: Run MinimumCutPhase(G, a)

Step 4: If cut is smaller than current minimum cut, store cut

Step 5: Merge last two added vertices

Step 6: Repeat

Step 7: Return minimum cut

7 Randomized Min-Cut Algorithm

7.1 Why Randomized Algorithm?

Randomized algorithms provide:

- Probabilistic guarantee of success
- Efficient computation
- Robust performance
- Parallelization capability

8 Karger's Randomized Min-Cut Algorithm

8.1 Main Idea

The algorithm repeatedly performs edge contraction randomly until only two vertices remain. The edges between these two vertices form the cut.

8.2 Algorithm Steps

Step 1: Start with graph G

Step 2: Randomly select an edge

Step 3: Contract the edge

Step 4: Repeat until only two vertices remain

Step 5: Remaining edges form the cut

9 Recursive Randomized Min-Cut Algorithm

Input:

Graph G with n vertices

Constant $\alpha > 0$

Output:

Cut C

If $n \leq \alpha^3$, compute minimum cut using brute force.

Else repeat for $i = 1$ to α:

Create graph G' by random contraction

Compute $C' = \text{RecursiveRandomizedMinCut}(G', \alpha)$

If $|C'| < |C|$, update $C \leftarrow C'$

Return C.

10 Comparison: Deterministic vs Randomized Min-Cut

10.1 Deterministic Min-Cut

Time complexity:

$$O(VE + V^2 \log V)$$

Guarantees exact minimum cut.

10.2 Randomized Min-Cut

Time complexity:

$$O(V^2)$$

Provides minimum cut with high probability.

11 Probability of Finding Min-Cut

$$\text{Probability} \geq \frac{2}{n(n-1)}$$

Where n = number of vertices.

12 Summary of Key Concepts

- Min-cut: smallest set of edges disconnecting graph
- Core operation: edge contraction
- Deterministic algorithm: Stoer–Wagner algorithm
- Randomized algorithm: Karger's algorithm
- Max-flow min-cut theorem: Maximum Flow = Minimum Cut Capacity
- Probability of success $\geq \frac{2}{n(n-1)}$