School of Engineering and Applied Science (SEAS), Ahmedabad University

CSE 400: Fundamentals of Probability in Computing

Lecture - 10 scribe

Name: Samridhi Mehrotra

Enrollment No: AU2440033

Email: samridhi.m@ahduni.edu.in

Lecture 10 : Randomized Min-Cut Algorithm

1 Overview

This lecture covers:

- Min-Cut Problem
- Applications of Min-Cut
- Definition of Min-Cut
- Successful and Unsuccessful Min-Cut Runs
- Max-Flow Min-Cut Theorem
- Deterministic Min-Cut Algorithm (Stoer–Wagner)
- Randomized Min-Cut Algorithm (Karger)
- Pseudocode
- Comparison of Deterministic vs Randomized Approaches
- Theorem for Min-Cut Probability
- Python Simulation Reference

2 Definitions and Notation

Let $G = (V, E)$ be an undirected graph with:

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

3 Min-Cut Problem

Given a graph $G = (V, E)$ with n vertices, the **Minimum Cut (Min-Cut)** problem is to find a cut-set of minimum cardinality.

3.1 Cut-Set

A **cut-set** is a set of edges whose removal disconnects the graph into two or more connected components.

3.2 Min-Cut

The **minimum cut** is the cut-set with the smallest number of edges.

4 Applications of Min-Cut

4.1 Network Design

Min-cut helps improve communication efficiency and optimize network flow by identifying the minimum capacity cut.

4.2 Communication Networks

Min-cut identifies vulnerabilities and helps design fault-tolerant networks.

4.3 VLSI Design

In VLSI, min-cut partitions circuits into smaller components to reduce interconnect complexity.

Reference: Section 1.5, *Probability and Computing (2nd Edition).*

5 Edge Contraction

The main operation in Karger's algorithm is **edge contraction**.

5.1 Definition

Contracting an edge (u, v):

- Merge vertices u and v
- Remove the edge (u, v)
- Remove self-loops
- Retain parallel edges

The resulting graph may contain parallel edges but no self-loops.

6 Successful and Unsuccessful Runs

6.1 Successful Min-Cut Run

A successful run correctly identifies the minimum cut after successive contractions.

6.2 Unsuccessful Min-Cut Run

An unsuccessful run occurs when the algorithm contracts an edge belonging to the true minimum cut early in the process.

Since the algorithm is randomized, early contraction of critical edges leads to incorrect results.

7 Max-Flow Min-Cut Theorem

7.1 Statement

In a flow network:

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

7.2 Definitions

- Capacity of a cut: sum of capacities of edges from X to Y
- Minimum cut: cut with smallest capacity
- Maximum flow: largest possible flow from source S to sink T

8 Deterministic Min-Cut Algorithm

8.1 Stoer–Wagner Algorithm

Let s and t be vertices of G. Let $G/\{s,t\}$ be the graph obtained by merging s and t.

A minimum cut of G is the smaller of:

- Minimum s-t cut of G
- Minimum cut of $G/\{s,t\}$

8.2 Proof Idea

Either:

- There exists a minimum cut separating s and t
- Or no such cut exists, and contraction preserves the minimum cut

9 Pseudocode: Stoer–Wagner

Algorithm 1: MinimumCutPhase(G, a)

```
A ← {a}
while A  V do
    add to A the most tightly connected vertex
return cut weight as the cut-of-the-phase
```

Algorithm 2: MinimumCut(G)

```
while |V|  1 do
    choose any vertex a
    MinimumCutPhase(G, a)
    if cut-of-the-phase < current minimum
        update minimum cut
    merge the last two vertices added
return minimum cut
```

9.1 Time Complexity

$$O(V \cdot E + V^2 \log V)$$

10 Randomized Min-Cut Algorithm

10.1 Why Randomized?

- Probabilistic success guarantee
- Efficient for large graphs
- Avoids worst-case deterministic behavior
- Easily parallelizable

11 Karger's Randomized Algorithm

11.1 Algorithm Idea

- Randomly select an edge
- Contract it
- Repeat until only 2 vertices remain
- Remaining edges form the cut

11.2 Example

Example run:

- Pick edge b, contract
- Pick edge d, contract
- Output cut: $\{a, c, e\}$

True minimum cut:

$$\{b, e\} \quad \text{or} \quad \{a, d\}$$

12 Recursive Randomized Min-Cut

Algorithm 3: Recursive-Randomized-Min-Cut(G, α)

```
if n  ˆ3 then
    return brute-force min-cut
else
    for i = 1 to  do
        G’ ← apply random contractions
        C’ ← Recursive-Randomized-Min-Cut(G’, )
        keep smallest cut found
return C
```

12.1 Time Complexity

$$O(V^2)$$

13 Probability Theorem for Min-Cut

The algorithm outputs a minimum cut with probability at least:

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

This probability increases with repeated independent runs.

14 Comparison: Deterministic vs Randomized

14.1 Deterministic (Stoer–Wagner)

- Always exact
- Higher time complexity

14.2 Randomized (Karger)

- Approximate with high probability
- Lower time complexity
- Requires repetition for higher confidence

15 Derivations / Proof Summary

Stoer–Wagner correctness follows from:

- If minimum cut separates s and t, then minimum s-t cut is global minimum.
- Otherwise, contraction preserves minimum cut.

16 Python Simulation

Steps:

- Open Campuswire Post for Lecture 10
- Download provided `.ipynb` file
- Run simulation of Karger's algorithm

17 Summary

- Cut-set: edges whose removal disconnects graph.
- Min-cut: minimum cardinality cut-set.
- Edge contraction: merge vertices, remove self-loops.
- Max-flow equals min-cut capacity.
- Stoer–Wagner: deterministic, $O(VE + V^2 \log V)$.
- Karger: randomized, $O(V^2)$.
- Probability of success $\geq \frac{2}{n(n-1)}$.