CSE400: Fundamentals of Probability in Computing

Lecture 10: Randomized Min-Cut Algorithm

Swayam Prajapati, AU2440087

Group 17

February 06, 2026

Outline

- Min-Cut Problem
- Why use min-cut?
- What is min-cut?
- Successful min-cut run
- Unsuccessful min-cut run
- Max-Flow Min-Cut Theorem
- Deterministic Min-Cut Algorithm
- Stoer–Wagner Minimum Cut Algorithm
- Pseudocode
- Randomized Min-Cut Algorithm
- Why randomized algorithm?
- Karger's Randomized Algorithm
- Pseudocode
- Comparison: Deterministic Min-Cut vs Randomized Min-Cut Algorithm
- Theorem for min-cut set
- Python Simulation

1 Min-Cut Problem

1.1 Why use min-cut?

We use the min-cut algorithm in various applications to solve problems related to network connectivity, reliability, and optimization.

- **Network Design:** Min cut helps in improving the efficiency of communication and optimizing network flow. The algorithm is used in network design to find the minimum capacity cut.
- **Communication Networks:** For understanding the vulnerability of the networks

1.3 Edge Contraction

The main operation in the algorithm is **edge contraction**, which is an operation that removes an edge from a graph while simultaneously merging the two vertices.

In contracting an edge (u, v):

- Merge vertices u and v into one vertex
- Eliminate all edges connecting u and v
- Retain all other edges in the graph

The new graph may have parallel edges but no self-loops.

2 Successful Min-Cut Run

A successful min-cut run refers to the success in the outcome of an algorithm designed to find the minimum cut in a graph.

Figure: Successful Run

3 Unsuccessful Min-Cut Run

An unsuccessful min-cut run refers to an iteration of a min-cut algorithm where the algorithm fails to correctly identify the minimum cut of a given graph.

Figure: Unsuccessful Run

4 Max-Flow Min-Cut Theorem

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."

Definitions

- **Capacity of a cut:** The sum of the capacities of the edges in the cut that are oriented from a vertex $\in X$ to a vertex $\in Y$
- **Minimum cut:** The cut in the network that has the smallest possible capacity
- **Minimum cut capacity:** The capacity of the minimum cut
- **Maximum flow:** The largest possible flow from source S to sink T

5 Deterministic Min-Cut Algorithm

5.1 Stoer–Wagner Min-Cut Algorithm

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

Then, a minimum cut of G can be obtained by taking the smaller of:

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

The theorem holds because:

- Either a minimum cut of G separates s and t, in which case the minimum s-t cut is a minimum cut of G
- Or there is none, in which case a minimum cut of $G/\{s,t\}$ is sufficient

5.2 Pseudocode

Algorithm 1: MinimumCutPhase(G, a)

1. $A \leftarrow \{a\}$
2. While $A \neq V$:
    - Add to A the most tightly connected vertex
3. Return the cut weight as the "cut of the phase"

Algorithm 2: MinimumCut(G)

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

6 Randomized Min-Cut Algorithm

6.1 Why Randomized Algorithm?

Randomized algorithms provide a probabilistic guarantee of success. It provides a more accurate estimate of the minimum cut with fewer iterations.

- Efficiency
- Parallelization
- Approximation Guarantees
- Avoidance of Worst-Case Instances
- Heuristic Nature
- Robustness

6.2 Karger's Randomized Algorithm

An example run of Karger's randomized algorithm shows that when random edges are picked in an unfavorable order, the output cut may not be minimal.

6.3 Pseudocode

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

- **Input:** An undirected multigraph G with n vertices, and an integer constant $\alpha > 0$
- **Output:** A cut C of G

1. If $n \leq \alpha^3$:
   - $C \leftarrow$ a min-cut of G found using brute force (exhaustive) search
2. Else:
   (a) For $i = 1$ to α:
      - $G' \leftarrow$ multigraph obtained by applying $n - \left\lceil \frac{n}{\sqrt{\alpha}} \right\rceil$ random contraction steps on G
      - $C' \leftarrow$ Recursive-Randomized-Min-Cut(G', α)
      - If $i = 1$ or $|C'| < |C|$, then $C \leftarrow C'$
3. Return C

7 Comparison: Deterministic vs Randomized Min-Cut

Any specific problem is responsible for deciding the approach.

Deterministic Min-Cut

- Always guarantees an exact minimum cut
- May have higher time complexity for large graphs
- Stoer–Wagner algorithm complexity: $O(V \cdot E + V^2 \log V)$

Randomized Min-Cut

- Provides an approximate minimum cut with high probability
- Karger's algorithm complexity: $O(V^2)$

8 Theorem for Min-Cut Set

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

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

End of Lecture