CSE 400: Fundamentals of Probability in Computing
Lecture 10: Randomized Min-Cut Algorithm
Group No.: S1 G17
Date: February 5, 2026

1 Min-Cut Problem

1.1 Why Use Min-Cut

Min-cut algorithms are used in applications related to network connectivity, reliability, and optimization. They are applied in:

- Network design to improve communication efficiency and optimize network flow by identifying the minimum capacity cut.
- Communication networks to analyze vulnerability to failures and build robust, fault-tolerant systems.
- VLSI design for partitioning circuits into smaller components to reduce interconnectivity complexity.

1.2 What is Min-Cut

Definition 1 (Cut-Set). *A cut-set in a graph is a set of edges whose removal breaks the graph into two or more connected components.*

Definition 2 (Minimum Cut). *Given a graph $G = (V, E)$ with n vertices, the minimum cut (min-cut) problem is to find a minimum cardinality cut-set in G.*

Min-cut algorithms such as Karger's algorithm are randomized and can be sensitive to the initial choice of edges. If critical edges are contracted early, the algorithm may find a smaller cut.

1.3 Edge Contraction

The main operation in the min-cut algorithm is *edge contraction*.

Definition 3 (Edge Contraction). *In contracting an edge (u, v), the two vertices u and v are merged into one vertex, all edges connecting u and v are eliminated, and all other edges are retained. The resulting graph may contain parallel edges but no self-loops.*

1.4 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.

1.5 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.

2 Max-Flow Min-Cut Theorem

Theorem 1 (Max-Flow Min-Cut Theorem)**.** *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.*

2.1 Related Definitions

- Capacity of a cut: The sum of the capacities of the edges in the cut that are oriented from a vertex in set X to a vertex in set Y.
- Minimum cut: The cut in the network with 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.

3 Deterministic Min-Cut Algorithm

3.1 Stoer–Wagner Min-Cut Algorithm

Proposition 1. *Let s and t be two vertices of a graph G. Let $G/\{s,t\}$ be the graph obtained by merging s and t. A minimum cut of G is the smaller of:*

- *a minimum s–t cut of G, and*
- *a minimum cut of $G/\{s,t\}$.*

3.2 Pseudocode

Algorithm 1 (MinimumCutPhase(G, a))**.** *1. $A \leftarrow \{a\}$*

2. while $A \neq V$ do

- *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 $|V| \geq 1$ do*

- *choose any $a \in V$*
- *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

4 Randomized Min-Cut Algorithm

4.1 Why Randomized Algorithms

Randomized algorithms provide a probabilistic guarantee of success and can give a more accurate estimate of the minimum cut with fewer iterations. Properties include efficiency, parallelization, approximation guarantees, avoidance of worst-case instances, heuristic nature, and robustness.

4.2 Karger's Randomized Algorithm

Karger's algorithm repeatedly contracts randomly selected edges until only two vertices remain. The remaining edges form a cut.

4.3 Pseudocode

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

1. *Input: Undirected multigraph G with n vertices, integer constant $\alpha > 0$*
2. *Output: A cut C of G*
3. *If $n \leq 3$, compute a min-cut of G using brute force and set C*
4. *Else:*
   - *for $i = 1$ to α:*
     - *$G' \leftarrow$ graph obtained by applying $n - \lceil n/\sqrt{\alpha} \rceil$ random contractions to G*
     - *$C' \leftarrow$ Recursive-Randomized-Min-Cut(G', α)*
     - *if $i = 1$ or $|C'| < |C|$, set $C \leftarrow C'$*
5. *return C*

5 Comparison: Deterministic vs Randomized Min-Cut

The choice of algorithm depends on the specific problem.

5.1 Deterministic Min-Cut

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

5.2 Randomized Min-Cut

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

6 Theorem for Min-Cut Set

Theorem 2. *The randomized min-cut algorithm outputs a min-cut set with probability at least* $\dfrac{2}{n(n-1)}$.