

**CSE 400: Fundamentals of Probability in Computing**  
Lecture 10 Scribe: Randomized Min-Cut Algorithm

**Group No.:** S1 G17

**Domain:** ITS

**Date:** February 5, 2026

---

## Min-Cut Problem

### Cut-Set and Min-Cut Definitions

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

Given an undirected graph  $G = (V, E)$  with  $|V| = n$ , the minimum cut problem is defined as the task of finding a cut-set of minimum cardinality in  $G$ .

### Edge Contraction Mechanism

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

Contracting an edge  $(u, v)$  consists of:

- Merging vertices  $u$  and  $v$  into a single supernode.
- Removing all edges directly connecting  $u$  and  $v$ .
- Retaining all remaining edges.

The resulting graph may contain parallel edges and does not contain self-loops.

## Successful Min-Cut Run

A successful min-cut run is defined as an execution of the algorithm that returns a cut-set whose cardinality equals the minimum cut of the graph.

Node contraction sequence corresponding to the successful run:

1. Nodes 3 and 4 are merged into supernode  $\{3, 4\}$ .
2. Nodes 2 and  $\{3, 4\}$  are merged into supernode  $\{2, 3, 4\}$ .
3. Nodes 1 and  $\{2, 3, 4\}$  are merged into supernode  $\{1, 2, 3, 4\}$ .
4. The final remaining partition corresponds to the minimum cut.

## Unsuccessful Min-Cut Run

An unsuccessful min-cut run refers to an execution in which the algorithm contracts at least one edge belonging to a minimum cut, resulting in a cut-set with cardinality strictly larger than the minimum cut.

Node contraction sequence corresponding to the unsuccessful run:

1. Nodes 1 and 2 are merged into supernode  $\{1, 2\}$ .
2. Nodes 3 and 4 are merged into supernode  $\{3, 4\}$ .
3. Nodes  $\{1, 2\}$  and  $\{3, 4\}$  are merged into a single supernode.
4. The resulting cut is not minimum.

## Max-Flow Min-Cut Theorem

The Max-Flow Min-Cut Theorem states:

In a flow network, the maximum amount of flow from a source vertex  $S$  to a sink vertex  $T$  is equal to the total capacity of a minimum cut separating  $S$  and  $T$ .

Formal definitions:

- Capacity of a cut: the sum of capacities of edges directed from partition  $X$  to partition  $Y$ .
- Minimum cut: a cut with minimum possible capacity.
- Maximum flow: the largest feasible flow from  $S$  to  $T$ .

## Deterministic Min-Cut: Stoer–Wagner Algorithm

Let  $s$  and  $t$  be two vertices of a graph  $G$ .

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

A minimum cut of  $G$  is obtained as the minimum of:

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

### Algorithm 1: MinimumCutPhase( $G, a$ )

- Initialize  $A \leftarrow \{a\}$ .
- While  $A \neq V$ , add the most tightly connected vertex to  $A$ .
- Return the cut weight of the phase.

### Algorithm 2: MinimumCut( $G$ )

- While  $|V| \geq 1$ :
- Select any vertex  $a \in V$ .
- Execute MinimumCutPhase( $G, a$ ).
- Update the current minimum cut if the phase cut is lighter.
- Shrink  $G$  by merging the last two added vertices.
- Return the minimum cut.

## Randomized Min-Cut Algorithm

### Karger's Randomized Algorithm

The algorithm repeatedly contracts randomly chosen edges until only two supernodes remain.

The edges between the two remaining supernodes form a cut.

### Algorithm 3: Recursive-Randomized-Min-Cut( $G, \alpha$ )

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

Output: A cut  $C$  of  $G$ .

- If  $n \leq 3$ :
  - Compute a min-cut of  $G$  using exhaustive search.
- Else:
  - For  $i = 1$  to  $\alpha$ :
  - Obtain  $G'$  by performing  $n - \lceil \frac{n}{\sqrt{\alpha}} \rceil$  random contractions on  $G$ .
  - Recursively compute  $C' = \text{Recursive-Randomized-Min-Cut}(G', \alpha)$ .
  - If  $i = 1$  or  $|C'| < |C|$ , update  $C \leftarrow C'$ .
- Return  $C$ .

The base case  $n \leq 3$  applies only in the context specified on Slide 35.

### Theorem: Success Probability

For a graph with  $n$  vertices, Karger's randomized min-cut algorithm outputs a minimum cut with probability at least:

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

### Complexity Comparison

- Stoer–Wagner deterministic min-cut algorithm runs in time  $O(VE + V^2 \log V)$ .
- Karger's randomized min-cut algorithm runs in time  $O(V^2)$  per execution.

The randomized algorithm does not scale to the same complexity as  $O(VE + V^2 \log V)$  and remains asymptotically distinct.