

CSE 400 Lecture Scribe



L10_S1.pdf

PDF

Task CSE 400: A Professional Lecture Scribe of CSE 400: Fundamentals of Probability in Computing Lecture 10 (Randomized Min-Cut Algorithm).

I. Academic Status and Purpose Professional academic scribe. This is a determinable and it must be an allusion of close-note tests. Employment: Priority Strict faithfulness to L10s1.pdf. Do not create, or present intuitions/analogies that were not covered in the slides (do not especially introduce non-academic filler).

II. Introduction to Compulsory Technical Material.

Definitions Have formal definitions cut-set, min-cut problem and edge contraction mechanisms (parallel edges and self-loops). Theorems: State Max-Flow Min-Cut Theorem capacity/flow definition and the success probability of the min-cut set (no less than $2/n(n-1)$) Theorems. Node-Specific Reconstruction: In Successful Run (Slide 19) and the Unsuccessful Run (Slide 22), the order of the merging the nodes (e.g. 1: Nodes 3 and 4 are merged into supernode $\{3,4\}$) is displayed in the order so that the logic can be re-created without the image. Phase and Minimum Cut based on the StoerWagner logic Phase and Minimum Cut Algorithms Phase and Minimum Cut Algorithms 1 (Phase) and 2 (Minimum Cut). Critical Correction: There is a case where $n = \alpha^3$ which is the base case in the Algorithm 3 (Recursive Randomized Min-Cut). It is only so in the case of Slide 35. Complexity: Does it scale up to the same complexity $O(VE + V^2 \log V)$ as $O(V^2)$ does?

III. Negative Constraints are hampered.

* Forbidden Language: One should not use such words as for example and like this. No Fluff: Do not give an introductory or a conclusion summary. The scribe must halt at the intersection of the technical contents.

Output formatting: It is compulsory that the customers should merely place the complete source code of the LaTeX containing the documentclass and the end of

document.

IV. LaTeX Template The template should have the following format:

```
\documentclass[11pt]{article}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage{amsmath,amssymb}
\usepackage{enumitem} \usepackage{fancyhdr}
\pagestyle{fancy} \fancyhf{} \lhead{CSE 400:
Fundamentals of Probability in Computing}
\rhead{Lecture 10 Scribe Report} \cfoot{\thepage}
\title{ \normalsize School of Engineering and Applied
Science (SEAS), Ahmedabad University \ \vspace{0.2cm}
\textbf{CSE 400: Fundamentals of Probability in
Computing}} \Large Lecture 10 Scribe: Randomized
Min-Cut Algorithm } \author{} \date{}
\begin{document} \maketitle \vspace{-2cm}
\begin{center} \begin{tabular}{ll} \textbf{Group No.:} & &
\{S1 G17 \hspace{2.5in}} \ [1.5ex] \textbf{Domain:} & &
\{ITS \hspace{2.5in}} \ [1.5ex] \textbf{Date:} & &
\{February
5, 2026 \hspace{2.2in}} \end{tabular} \end{center}
\hrule \vspace{0.5cm} % INSERT HIERARCHICAL
LECTURE CONTENT HERE \end{document}
```

latex

```
\documentclass[11pt]{article}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage{amsmath,amssymb}
\usepackage{enumitem}
\usepackage{fancyhdr}

\pagestyle{fancy}
\fancyhf{}
\lhead{CSE 400: Fundamentals of Probability in Computing}
\rhead{Lecture 10 Scribe Report}
\cfoot{\thepage}

\title{
\normalsize School of Engineering and Applied Science (SEAS), Ahmedabad University\
\vspace{0.2cm}
\textbf{CSE 400: Fundamentals of Probability in Computing}}\
\Large Lecture 10 Scribe: Randomized Min-Cut Algorithm
```

```

}
\author{}
\date{}

\begin{document}
\maketitle
\vspace{-2cm}

\begin{center}
\begin{tabular}{ll}
\textbf{Group No.:} & {S1 G17 \hspace{2.5in}} \\\[1.5ex]
\textbf{Domain:} & {ITS \hspace{2.5in}} \\\[1.5ex]
\textbf{Date:} & {February 5, 2026 \hspace{2.2in}}
\end{tabular}
\end{center}

\hrule
\vspace{0.5cm}

```

\section*{Min-Cut Problem}

\subsection*{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 .

\subsection*{Edge Contraction Mechanism}

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

Contracting an edge (u,v) consists of:

- \begin{itemize}[leftmargin=1.2cm]
- \item Merging vertices u and v into a single supernode.
- \item Removing all edges directly connecting u and v .
- \item Retaining all remaining edges.
- \end{itemize}

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

\section*{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:

\begin{enumerate}[leftmargin=1.2cm]

\item Nodes 3 and 4 are merged into supernode $\{3,4\}$.

\item Nodes 2 and $\{3,4\}$ are merged into supernode $\{2,3,4\}$.

\item Nodes 1 and $\{2,3,4\}$ are merged into supernode $\{1,2,3,4\}$.

\item The final remaining partition corresponds to the minimum cut.

\end{enumerate}

\section*{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:

\begin{enumerate}[leftmargin=1.2cm]

\item Nodes 1 and 2 are merged into supernode $\{1,2\}$.

\item Nodes 3 and 4 are merged into supernode $\{3,4\}$.

\item Nodes $\{1,2\}$ and $\{3,4\}$ are merged into a single supernode.

\item The resulting cut is not minimum.

\end{enumerate}

\section*{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:

\begin{itemize}[leftmargin=1.2cm]

\item Capacity of a cut: the sum of capacities of edges directed from partition X to partition Y .

\item Minimum cut: a cut with minimum possible capacity.

\item Maximum flow: the largest feasible flow from S to T .

\end{itemize}

$\text{\section*{Deterministic Min-Cut: Stoer--Wagner Algorithm}}$

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:

$\text{\begin{itemize}[leftmargin=1.2cm]}$

\item A minimum s - t cut of G .

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

$\text{\end{itemize}}$

$\text{\subsection*{Algorithm 1: MinimumCutPhase}(G,a)}$

$\text{\begin{itemize}[leftmargin=1.2cm]}$

\item Initialize $A \leftarrow a$.

\item While $A \neq V$, add the most tightly connected vertex to A .

\item Return the cut weight of the phase.

$\text{\end{itemize}}$

$\text{\subsection*{Algorithm 2: MinimumCut}(G)}$

$\text{\begin{itemize}[leftmargin=1.2cm]}$

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

\item Select any vertex $a \in V$.

\item Execute $\text{MinimumCutPhase}(G,a)$.

\item Update the current minimum cut if the phase cut is lighter.

\item Shrink G by merging the last two added vertices.

\item Return the minimum cut.

$\text{\end{itemize}}$

$\text{\section*{Randomized Min-Cut Algorithm}}$

$\text{\subsection*{Karger's Randomized Algorithm}}$

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

The edges between the two remaining supernodes form a cut.

$\text{\subsection*{Algorithm 3: Recursive-Randomized-Min-Cut}(G,\alpha)}$

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

Output: A cut C of G .

```
\begin{itemize}[leftmargin=1.2cm]
```

```
\item If  $n \leq 3$ :
```

```
\begin{itemize}
```

```
\item Compute a min-cut of  $G$  using exhaustive search.
```

```
\end{itemize}
```

```
\item Else:
```

```
\begin{itemize}
```

```
\item For  $i = 1$  to  $\alpha$ :
```

```
\item Obtain  $G'$  by performing  $n - \lceil \frac{n}{\sqrt{\alpha}} \rceil$  random  
contractions on  $G$ .
```

```
\item Recursively compute  $C' = \text{Recursive-Randomized-Min-Cut}(G', \alpha)$ .
```

```
\item If  $i=1$  or  $|C'| < |C|$ , update  $C \leftarrow C'$ .
```

```
\end{itemize}
```

```
\item Return  $C$ .
```

```
\end{itemize}
```

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

$\text{\section*{Theorem: Success Probability}}$

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

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

$\text{\section*{Complexity Comparison}}$

```
\begin{itemize}[leftmargin=1.2cm]
```

```
\item Stoer--Wagner deterministic min-cut algorithm runs in time  $O(VE + V^2 \log V)$ .
```

```
\item Karger's randomized min-cut algorithm runs in time  $O(V^2)$  per execution.
```

```
\end{itemize}
```

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

```
\end{document}
```

