The Karger-Stein Algorithm Is Optimal for k-Cut

Technical Summary — CP -2 — Rameez Wasif — Zain Hatim

Problem and Contribution

The paper addresses the fundamental problem in graph theory of finding a **minimum k-cut** in an edge-weighted graph. The k-cut problem asks for the least-weight set of edges whose removal breaks the given graph into at least k connected components. This problem generalizes the well-known global min-cut problem (where $k = 2$).

The main contribution of this paper is proving that **the original randomized Karger-Stein algorithm is optimal for solving the k-cut problem for any fixed** $k \geq 2$. Specifically, the authors show that the Karger-Stein algorithm outputs any fixed minimum k-cut with a probability of at least $\hat{O}(n^{-k})$, where $\hat{O}(\cdot)$ hides a quasi-logarithmic factor of $2^{O(\ln \ln n)^2}$. This result has significant implications, leading to two important corollaries:

- An **extremal bound of $\hat{O}(n^k)$ on the number of minimum k-cuts** in an n-vertex graph. This improves upon the previous best bound of $n^{(1.98+o(1))k}$. The bound is also shown to be almost tight, as a cycle on n vertices has $\Omega(n^k)$ minimum k-cuts.
- A **randomized algorithm to compute a minimum k-cut in $\hat{O}(n^k)$ time with high probability**. This improves upon existing running times, including $n^{(1.98+o(1))k}$ for general weighted graphs and $n^{(1+o(1))k}$ for unweighted graphs. The achieved runtime is also considered almost tight under the hypothesis that solving the Max-Weight $(k-1)$-Clique problem requires $\tilde{\Omega}(n^{(1-o(1))k})$ time.

Algorithmic Description

The core of the paper's result lies in a refined analysis of the original **Karger-Stein algorithm**. The algorithm is as follows:

- **Input:** An edge-weighted graph $G = (V, E, w)$ and an integer $k \geq 2$.
- **Output:** A set of edges whose removal disconnects G into at least k components, ideally a minimum such set.
- **High-Level Logic:**
  1. While the number of vertices $|V|$ is greater than k:
     (a) Sample an edge $e \in E$ with probability proportional to its weight $w(e)$.
     (b) Contract the two vertices incident to e, merging them into a single vertex. Remove any self-loops that may result.
  2. When $|V| = k$, consider the set of edges that would separate these k remaining (contracted) vertices in the original graph. Return this set of edges as a candidate for the minimum k-cut.

The paper's analysis introduces an **"exponential clock" view** of this process. Instead of contracting one edge at a time, each edge e is imagined to independently sample a random variable $x(e)$ from an exponential distribution with a mean related to the minimum k-cut value λ_k. An edge is considered for contraction by time t with a certain probability derived from this distribution. This perspective allows for analyzing the probability of each edge being contracted independently over infinitesimal timesteps.

The proof strategy involves two main components:

1. **Fine-grained algorithmic analysis of the Karger-Stein algorithm:** This analysis tracks how the graph shrinks and how the average degree of vertices evolves under the exponential clock process. A key insight is that as the process continues, more vertices are expected to have a degree significantly higher than λ_k, leading to a faster rate of contraction than a worst-case analysis would suggest. This analysis leverages an extremal bound on the number of "small" cuts in the graph.
2. **Extremal result bounding the number of "small" cuts:** The paper proves that for any $\gamma < 2$, the number of cuts with weight less than $\gamma\lambda_k$ is at most $(\max(\frac{1}{2-\gamma}, k))^{O(k)} n$. This proof relies on techniques from extremal

set theory, particularly the **Sunflower Lemma** and its refinements. The connection between graph cuts and sunflowers of sets of vertices is established, showing that a large number of small cuts with certain intersection properties would lead to a k-cut with a weight less than the minimum k-cut, resulting in a contradiction.

Comparison with Existing Approaches

Historically, finding a minimum k-cut was known to be solvable in polynomial time for fixed k, with the first such algorithm by Goldschmidt and Hochbaum achieving a runtime of $O(n^{(1/2-o(1))k^2})$. The original randomized minimum-cut algorithm by Karger and Stein could be adapted to solve k-Cut in $\tilde{O}(n^{2(k-1)})$ time. Deterministic algorithms, such as those based on tree-packing results, also achieved $O(n^{(2-o(1))k})$ bounds.

Recent progress had yielded algorithms with improved running times for specific cases, such as $O(n^{(1.98+o(1))k})$ for general k-cut, and more efficient algorithms for graphs with polynomial integer weights or unweighted graphs.

The key novelty and improvement of this work are:

- It provides a tight (up to quasi-logarithmic factors) bound on the probability of the original Karger-Stein algorithm finding a minimum k-cut.
- It establishes an almost tight extremal bound on the number of minimum k-cuts in a graph.
- It presents a randomized algorithm based on the Karger-Stein approach that achieves a runtime of $\hat{O}(n^k)$ for finding a minimum k-cut in general weighted graphs. This significantly improves upon previous general algorithms and matches the conjectured lower bound, suggesting optimality.

Data Structures and Techniques

The paper utilizes a variety of mathematical tools and algorithmic techniques:

- **Probability Theory:** The analysis relies heavily on probability concepts, including independent events, conditional probability, expectation, Markov's inequality (implicitly), and concentration bounds like **Freedman's inequality** for martingales.
- **Exponential Distribution:** The "exponential clock" view leverages the properties of the exponential distribution
- **Graph Contraction:** The fundamental operation of the Karger-Stein algorithm is the contraction of edges.
- **Extremal Set Theory:** The proof of the bound on the number of small cuts employs the **Sunflower Lemma** and related concepts like **Venn diagrams** and **set systems**.
- **Differential Equations:** The expected behavior of the graph shrinkage process is modeled and analyzed using **ordinary differential equations**.
- **Graph Sparsification:** The **Nagamochi-Ibaraki theorem** is used to reduce the number of edges in the graph to at most $\lambda_k n$ while preserving all k-cuts of size $\leq \lambda_k$, which is beneficial for the concentration analysis.

Implementation Outlook

Implementing an algorithm based on this work would present several technical challenges:

- **Graph Contraction Implementation:** The vertex contraction operation needs to be implemented carefully, ensuring correct merging of edges and removal of self-loops. Maintaining the edge weights during contraction is crucial.
- **Estimating λ_k:** The efficient algorithm relies on having a sufficiently accurate estimate of the minimum k-cut value λ_k. While approximations exist, the impact of the accuracy of this estimation on the algorithm's performance in practice needs to be considered.
- **Implementation of Sunflower Lemma Related Techniques:** The extremal result relies on the Sunflower Lemma, which is primarily used for the theoretical analysis of the Karger-Stein algorithm's probability of success. Directly implementing algorithms based on finding sunflowers might be complex and computationally expensive in practice.