Approximation Algorithm for Norm Multiway Cut

Charlie Carlson
University of Colorado Boulder, CO, USA

Jafar Jafarov
Toyota Technological Institute at Chicago, IL, USA

Konstantin Makarychev
Northwestern University, Evanston, IL, USA

Yury Makarychev
Toyota Technological Institute at Chicago, IL, USA

Liren Shan
Northwestern University, Evanston, IL, USA

Abstract

We consider variants of the classic Multiway Cut problem. Multiway Cut asks to partition a graph G into k parts so as to separate k given terminals. Recently, Chandrasekaran and Wang (ESA 2021) introduced ℓ_p-norm Multiway Cut, a generalization of the problem, in which the goal is to minimize the ℓ_p norm of the edge boundaries of k parts. We provide an $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation algorithm for this problem, improving upon the approximation guarantee of $O(\log^{3/2} n \log^{1/2} k)$ due to Chandrasekaran and Wang.

We also introduce and study Norm Multiway Cut, a further generalization of Multiway Cut. We assume that we are given access to an oracle, which answers certain queries about the norm. We present an $O(\log^{1/2} n \log^{7/2} k)$ approximation algorithm with a weaker oracle and an $O(\log^{1/2} n \log^{5/2} k)$ approximation algorithm with a stronger oracle. Additionally, we show that without any oracle access, there is no $n^{1/4-\varepsilon}$ approximation algorithm for every $\varepsilon > 0$ assuming the Hypergraph Dense-vs-Random Conjecture.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Facility location and clustering

Keywords and phrases Multiway cut, Approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ESA.2023.32

Related Version *Full Version*: `https://arxiv.org/abs/2308.08373`

Funding *Konstantin Makarychev*: supported by NSF Awards CCF-1955351, CCF-1934931, EECS-2216970.
Yury Makarychev: supported by NSF CCF-1955173, CCF-1934843, and ECCS-2216899.
Liren Shan: supported by NSF Awards CCF-1955351, CCF-1934931, and EECS-2216970.

1 Introduction

In this paper, we consider a variant of the classic combinatorial optimization problem, Minimum Multiway Cut. Given an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_{\geq 0}$ and k terminals $t_1, \ldots, t_k \in V$, the Minimum Multiway Cut problem asks to partition graph G into k parts $P_1, \ldots, P_k$ so that P_i contains terminal t_i. The Multiway Cut objective is to minimize the number or total weight of cut edges. For $k = 2$, the problem is equivalent to the minimum st-Cut problem. Dahlhaus, Johnson, Papadimitriou, Seymour, and Yannakakis proved that it is NP-complete and APX-hard for every $k > 2$ [9]. They also gave a simple combinatorial $(2 - 2/k)$-approximation algorithm. Later Călinescu, Karloff, and Rabani [7] showed how to obtain a $3/2$ approximation using linear programming. This result was improved in a series of papers by Karger, Klein, Stein, Thorup, and Young [11], Buchbinder, Naor and Schwartz [5], and Sharma and Vondrák [13] (see also [6]). The currently best known approximation factor is 1.2965 [13]. The best known LP integrality gap and Unique Games Conjecture hardness is 1.20016 due to Bérczi, Chandrasekaran, Király, and Madan [4] (see also [2, 10, 12]).

In 2004, Svitkina and Tardos [14] introduced the Min-Max Multiway Cut problem. In this problem, as before, we need to partition graph G into k parts $P_1, \dots, P_k$ so that each P_i contains one terminal t_i. However, the objective function is different: Min-Max Multiway Cut asks to minimize the maximum of edge boundaries of sets P_i i.e., minimize $\max_i \delta(P_i)$, where $\delta(P_i)$ is the total weight of edges crossing the cut $(P_i, V \setminus P_i)$. Svitkina and Tardos [14] gave an $O(\log^3 n)$ approximation algorithm for the problem. Later, Bansal, Feige, Krauthgamer, Makarychev, Nagarajan, Naor, and Schwartz [3] provided an $O(\sqrt{\log n \log k})$-approximation algorithm. Also, Ahmadi, Khuller, and Saha [1] studied a related Min-Max Multicut problem.

Recently, Chandrasekaran and Wang [8] proposed a common generalization of the Min Multiway Cut and Min-Max Multiway Cut problems, which they called Minimum ℓ_p-norm Multiway Cut. This problem asks to minimize the ℓ_p norm of the edge boundaries of parts $P_1, \dots, P_k$. In other words, the objective is to

$$\textbf{minimize:}\ \Big(\sum_{i=1}^{k} \delta(P_i)^p\Big)^{1/p}.$$

Note that this problem is equivalent to Min Multiway Cut when $p = 1$ and to Min-Max Multiway Cut when $p = \infty$. Chandrasekaran and Wang [8] gave an $O(\log^{3/2} n \log^{1/2} k)$ approximation for the problem. Further, they proved that the problem is NP-hard for every $p \geq 1$ and $k \geq 4$. Moreover, it does not admit an $O(k^{1-1/p-\varepsilon})$-approximation for every $\varepsilon > 0$ assuming the Small Set-Expansion Conjecture; a natural convex program for the problem has the intgerality gap of $\Omega(k^{1-1/p})$.

In this paper, we provide an improved $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation algorithm. We note that for $p = \infty$, our approximation guarantee matches the approximation of the algorithm due to Bansal et al. [3].[1] For $p = 2$, our approximation guarantee is $O(\log^{1/2} n \log k)$, which is $\Theta(\log n/\sqrt{\log k})$ times better than the approximation guarantee of the algorithm due to Chandrasekaran and Wang [8]. We also consider variants of Multiway Cut with norms other than the ℓ_p norm.

1.1 Our Results

We now formally state our results. First, we present an approximation algorithm for the ℓ_p-norm Multiway Cut problem. We show that our algorithm achieves an $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation for every $p > 1$.

▶ **Theorem 1.** *There exists a polynomial-time randomized algorithm that given a graph with n vertices, k terminals, and $p > 1$, finds an $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation for ℓ_p-norm Multiway Cut with high probability.*

Further, we provide approximation algorithms for Norm Multiway Cut with an arbitrary monotonic norm, a further generalization of ℓ_p-norm Multiway Cut. The monotonic norm is defined as follows.

[1] Our algorithm is stated only for the case where p is finite. However, we can solve an instance with $p = \infty$ by running the algorithm with $p = \log k$. Since $\|\cdot\|_{\log k}$ is within a constant factor of $\|\cdot\|_\infty$ for vectors in $\mathbb{R}^k$, this approach yields an $O(\sqrt{\log n} \log^{1/2+1/\log k} k) = O(\sqrt{\log n \log k})$-approximation.

▶ **Definition 2.** *A norm $\|\cdot\|$ on $\mathbb{R}^d$ is monotonic if for any $x, y \in \mathbb{R}^d$ with $|x_i| \leq |y_i|$ for all $i \in [d]$, it holds $\|x\| \leq \|y\|$.*

We consider two oracles to the monotonic norm used in the Norm Multiway Cut: (1) minimization oracle; (2) ordering oracle. For a set $A \subseteq [d]$, let $\mathbb{1}_A \in \{0,1\}^d$ denote the indicator vector of A, i.e., the i-th coordinate $(\mathbb{1}_A)_i = 1$ if $i \in A$; otherwise, $(\mathbb{1}_A)_i = 0$.

▶ **Definition 3.** *Given a monotonic norm $\|\cdot\|$ on $\mathbb{R}^d$, for any $i \in [d]$, the minimization oracle efficiently finds a set $A_i \subseteq [d]$ that minimizes the norm of indicator vectors among all subsets with size i, i.e.*

$$A_i = \operatorname*{arg\,min}_{A \subseteq [n], |A| = i} \|\mathbb{1}_A\|.$$

▶ **Definition 4.** *Given a monotonic norm $\|\cdot\|$ on $\mathbb{R}^d$, for any vector $x \in \mathbb{R}^d$, the ordering oracle efficiently finds an ordering of the vector x that minimizes the norm, i.e.*

$$\pi_x = \operatorname*{arg\,min}_{\pi \in S_d} \|x^\pi\|,$$

where x^π denotes the ordering of x regarding the permutation π.

Assuming that they are given access to either a "minimization oracle" or a stronger "ordering oracle", our algorithms give $O(\log^{1/2} n \log^{7/2} k)$ and $O(\log^{1/2} n \log^{5/2} k)$ approximation, respectively. We remark that the oracles only answer queries about the norm and, in particular, there is an ordering oracle for the ℓ_p-norm, weighted ℓ_p-norm, and many other natural norms. Thus, our result implies an $O(\log^{1/2} n \log^{5/2} k)$ approximation for *weighted* ℓ_p-norm Multiway Cut. We prove the following theorems in the full version of the paper.

▶ **Theorem 5.** *There exists a polynomial-time algorithm that for every monotonic norm with a minimization oracle, given a graph with n vertices and k terminals, finds an $O(\log^{1/2} n \log^{7/2} k)$ approximation for the Norm Multiway Cut with high probability.*

▶ **Theorem 6.** *There exists a polynomial-time algorithm that for every monotonic norm with an ordering oracle, given a graph with n vertices and k terminals, finds an $O(\log^{1/2} n \log^{5/2} k)$ approximation for the Norm Multiway Cut with high probability.*

Finally, we show that the problem becomes very hard if we are not given access to a norm minimization oracle. The proof is given in the full version of the paper.

▶ **Theorem 7.** *Consider the Norm Multiway Cut problem with a monotonic norm. Assume that the norm is given by a formula (in particular, we can easily compute the value of the norm; however, we are not given a minimization oracle for it). Then, assuming the Hypergraph Dense-vs-Random Conjecture, there is no polynomial-time algorithm for Norm Multiway Cut with approximation factor $\alpha(n) \leq n^{1/4-\varepsilon}$ for every $\varepsilon > 0$.*

1.2 Proof Overview

We first describe our algorithm for the ℓ_p-norm Multiway Cut. Our algorithm consists of three procedures: (1) covering procedure, (2) uncrossing procedure, and (3) aggregation procedure.

In the covering procedure, we generate a collection of subsets of the vertex set, $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$. We generate these sets iteratively by using a bi-criteria approximation algorithm for Unbalanced Terminal Cut by Bansal et al. [3] and a multiplicative weight update method. See Section 2.1 and Algorithm 1 for details. Each set in $\mathcal{S}$ contains at most one terminal. These sets are not disjoint. While, these sets cover the entire graph, which means the union of all sets in $\mathcal{S}$ contains all vertices. The number of sets in $\mathcal{S}$ is at most $O(k \log n)$. We show that the ℓ_p norm and ℓ_1 norm of the edge boundaries of sets in $\mathcal{S}$ is at most $O(\log^{1/p} n \cdot \alpha)\,\mathrm{OPT}$ and $O(\log n \cdot k^{1-1/p} \cdot \alpha)\,\mathrm{OPT}$ respectively, where $\alpha = \sqrt{\log n \log k}$ and OPT is the cost of the optimal solution. This covering procedure follows the approach by Bansal et al. [3] for Min-Max Multiway Cut. Chandrasekaran and Wang [8] also use a similar covering procedure for ℓ_p-norm Multiway Cut. Their algorithm finds a cover $\mathcal{S}$ that satisfies the above properties except for the ℓ_1 norm bound. We get this ℓ_1 norm bound on the edge boundaries of sets in $\mathcal{S}$ by picking proper measure constraints for the Unbalanced Terminal Cut algorithm. This ℓ_1 norm bound is important in the aggregation procedure to get an improved approximation.

Note that the sets in $\mathcal{S}$ are not disjoint. We use the uncrossing procedure to create a partition of the graph with at most $O(k \log k)$ sets. Our uncrossing procedure first sample $O(k \log k)$ sets from $\mathcal{S}$ uniformly at random. Then, we run an iterative uncrossing process given by Bansal et al. [3] over sampled sets until all sets are disjoint and have small boundaries. We show that all sampled sets cover almost the entire graph. The set of uncovered vertices does not contain terminals and has a small boundary with high probability. Next, we use the aggregation procedure to merge these $O(k \log k)$ sets into a k partition. We assign k sets containing one terminal to k parts. For other sets without terminals, we assign them to k parts almost uniformly such that each part has almost the same ℓ_1 norm over assigned sets. After the uncrossing procedure, the ℓ_p norm and ℓ_1 norm of edge boundaries is at most $O(\log^{1/p} k \cdot \alpha)\,\mathrm{OPT}$ and $O(k^{1-1/p} \cdot \alpha)\,\mathrm{OPT}$ respectively. We upper bound the sets containing one terminal and the sets with the largest edge boundary in each part by the above ℓ_p norm bound. For the remaining sets, by the ℓ_1 norm bound and the uniform assignment, we upper bound the ℓ_p norm for these sets by $O(\alpha)\,\mathrm{OPT}$. Chandrasekaran and Wang [8] achieve an $O(\log n \cdot \alpha)$ where the $O(\log n)$ factor is due to their aggregation procedure. We use the ℓ_1 norm bound in the covering procedure and a new aggregation procedure to reduce $O(\log n)$ extra factor to $O(\log^{1/p} n)$. We use the sampling in the uncrossing procedure to further reduce the extra factor from $O(\log^{1/p} n)$ to $O(\log^{1/p} k)$.

We now describe our algorithm for Norm Multiway Cut. We use the same framework with covering, uncrossing, and aggregation procedures. While, unlike the ℓ_p norm, the general monotonic norm may not be permutation invariant. For each terminal, we first compute a minimum cut that separates this terminal from other terminals. Then, we can remove all terminals and assign the remaining vertices freely among k parts. We mainly use a bucketing idea to modify our algorithm. We partition k coordinates into $\log_2 k$ buckets with exponentially increasing size 2^i such that the coordinates with large boundaries in the optimal solution are assigned to small buckets. Differing from the previous covering procedure, the new covering procedure uses the Unbalanced Terminal Cut algorithm with parameters related to each bucket. The cover $\mathcal{S}$ contains $O(2^i \log n \log k)$ sets in each bucket i. The boundary of every set in each bucket is relatively small, at most $O(\alpha)$ times the boundary of the optimal part in this bucket. We then run the uncrossing and aggregation procedure to create a multiway cut. We still sample each set in $\mathcal{S}$ with probability $\log_2 k / \log_2 n$. Thus, we have $O(2^i \log^2 k)$ sets in each bucket i after the uncrossing procedure. For bucket $0 \le i \le \log_2 k$, we find a set of 2^i coordinates $I_i \in [k]$ that minimizes the norm of the indicator vector through the minimization oracle. We then assign $O(2^i \log^2 k)$ sets in each bucket to coordinates in I_i such that each coordinate has $O(\log^2 k)$ sets in bucket i. Thus, we achieve an $O(\log^2 k \cdot \alpha)$ approximation for each bucket. Since these sets of coordinates I_i may overlap, we lose an additional $O(\log k)$ factor for $\log_2 k$ buckets. Suppose we have a stronger oracle that finds the best ordering for any given vector that minimizes the norm. Then, we provide an assignment for each bucket to avoid the large overlapping among buckets. Therefore, we avoid the extra $O(\log k)$ factor loss due to the overlapping.

2 ℓ_p-norm Multiway Cut

In this section, we present our algorithm for ℓ_p-norm Multiway Cut. We prove the following theorem. Our algorithm consists of three parts: covering procedure, uncrossing procedure, and aggregation procedure. We describe and analyze the covering procedure in Section 2.1, the uncrossing and aggregation procedures in Section 2.2.

▶ **Theorem 1.** *There exists a polynomial-time randomized algorithm that given a graph with n vertices and k terminals, and $p > 1$, finds an $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation for the ℓ_p-norm Multiway Cut with high probability.*

2.1 Covering Procedure

We first present and analyze a covering procedure in our algorithm. The covering procedure takes a undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_{\geq 0}$ and k terminals $T = \{t_1, \dots, t_k\} \subset V$ as input and outputs a collection of sets $\mathcal{S} = \{S_1, \dots, S_m\}$ where $S_i \subset V$ for all i. All sets $S_i \in \mathcal{S}$ covers the entire graph, $\bigcup_{i=1}^m S_i = V$. Each set $S_i \in \mathcal{S}$ contains at most one terminal. For each subset $S \subseteq V$, we use $\partial(S) = E(S, V \setminus S)$ to denote all edges crossing the cut $(S, V \setminus S)$. We use $\delta(S) = \sum_{e \in \partial(S)} w(e)$ to denote the edge boundary of set S, which is the total weight of all edges crossing $(S, V \setminus S)$. We prove the following upper bounds on the ℓ_1-norm and ℓ_p-norm of the edge boundaries of these sets in $\mathcal{S}$, which is crucial for our approximation guarantee.

▶ **Lemma 8.** *Given a graph $G = (V, E)$ with n vertices and k terminals $T \subset V$, the covering procedure shown in Algorithm 1 returns $m = O(k \log n)$ sets $\mathcal{S} = \{S_1, \dots, S_m\}$ that satisfies*

1. $|S_i \cap T| \leq 1$ *for all* $i \in [m]$,
2. $\bigcup_{i=1}^{m} S_i = V$,
3. $\sum_{t=1}^{m} \delta(S_t)^p \leq \log n \cdot O(\alpha^p) \cdot \mathrm{OPT}^p$,
4. $\sum_{t=1}^{m} \delta(S_t) \leq \log n \cdot O(\alpha) \cdot k^{1-1/p} \cdot \mathrm{OPT}$,

where $\alpha = \sqrt{\log n \log k}$ and OPT is the objective value of the optimal ℓ_p-norm Multiway Cut.

Our algorithm relies on an intermediate problem, Unbalanced Terminal Cut that we introduce now.

▶ **Definition 9** (Unbalanced Terminal Cut). *The input to this problem is a tuple $\langle G, w, \mu, \rho, T \rangle$, where $G = (V, E)$ is a graph with edge weights $w : E \to \mathbb{R}_{\geq 0}$, a measure $\mu : V \to \mathbb{R}_{\geq 0}$, a parameter $\rho \in (0, 1]$, and a set of terminals T. The goal is to find $S \subseteq V$ of minimum cost $\delta(S)$ satisfying:*

1. $|S \cap T| \leq 1$,
2. $\mu(S) \geq \rho \cdot \mu(V)$.

Bansal, Feige, Krauthgamer, Makarychev, Nagarajan, Naor, and Schwartz [3] gave a bi-criteria approximation algorithm for Unbalanced Terminal Cut that we state in the following theorem.

Algorithm 1 Covering Procedure.

Set $t = 1$, and $\mu_1(v) = 1$ for all $v \in V$. Let $\mathcal{S} = \varnothing$.
while $\sum_{v \in V} \mu_t(v) \geq \frac{1}{n}$ **do**
  Let P_i^* be a set as stated in Lemma 14.
  Guess $\mu_t(P_i^*)$.
  Let $S_t \subseteq V$ be the solution for Unbalanced Terminal Cut instance $\langle G, w, \mu_t, \max\{\frac{1}{2k}, \frac{\mu_t(P_i^*)}{\mu_t(V)}\}, T\rangle$.
  Let $\mathcal{S} = \mathcal{S} \cup \{S_t\}$.
  for $v \in V$ **do**
    Set $\mu_{t+1}(v) = \begin{cases} \mu_t(v)/2, & \text{if } v \in S_t, \\ \mu_t(v), & \text{if } v \notin S_t. \end{cases}$
  Set $t = t + 1$
return $\mathcal{S}$.

▶ **Theorem 10.** *There exists a polynomial-time algorithm that given an instance $\langle G, w, \mu, \rho, T\rangle$ of Unbalanced Terminal Cut, finds a set $S \subseteq V$ satisfying $|S \cap T| \leq 1$, $\mu(S) \geq \Omega(\rho) \cdot \mu(V)$, and $\delta(S) \leq \alpha \cdot \mathrm{OPT}_{\langle G,w,\mu,\rho,T\rangle}$ where $\alpha = O\left(\sqrt{\log n \log 1/\rho}\right)$.*

Our covering procedure relies on the multiplicative weights update method and is inspired by the algorithm in [3]. It initializes the measure of each vertex to one. At each iteration t, the algorithm guesses the measure $\mu_t(P_i^*)$ of a particular set P_i^* in an optimal solution and computes S_t of measure $\mu_t(S_t) \approx \mu_t(P_i^*)$ using the Unbalanced Terminal Cut algorithm in Theorem 10. The existence of such a P_i^* is shown in Lemma 14. Once S_t is computed, the algorithm decreases the measure of the vertices covered by S_t by a factor of 2. The algorithm terminates when the total measure of vertices is less than $1/n$.

We guess $\mu_t(P_i^*)$ as follows: For any set $S \subseteq V$, its measure $\mu_t(S)$ lies in the range $[\mu_t(u), n \cdot \mu_t(u)]$, where $u = \arg\max_{v \in S} \mu_t(v)$ is the heaviest vertex in S. Thus $\mu_t(P_i^*)$ can be well approximated by the set $A = \left\{2^i \cdot \mu_t(v) : v \in V, i = 0, \ldots, \lfloor \log_2 n \rfloor\right\}$ of size $O(n \log n)$. For each candidate $a \in A$ we compute a set $S(a)$ using the Unbalanced Terminal Cut algorithm with a parameter a and choose $S_t = \arg\min_{a \in A} \delta(S(a))$ with the smallest cost. We give a pseudo-code for this algorithm in Algorithm 1. We remark that one can think of this algorithm as of multiplicative weight update algorithm for solving a covering LP with constraints from Lemma 8.

We then analyze this covering algorithm in Algorithm 1. Let $\mathcal{S} = \{S_1, \cdots, S_m\}$ denote the collection of m sets output by Algorithm 1. By Theorem 10, every set S_i contains at most one terminal. First, for any fixed vertex $v \in V$, we give a lower bound on the number of sets containing v.

▷ Claim 11. For a vertex $v \in V$, let $N_v = |\{S_t | \, v \in S_t\}|$ denote the number of sets containing v. Then $N_v \geq \Omega(\log n)$.

Proof. Recall that initially $\mu_1(v) = 1$ and after iteration m its measure becomes $\mu_{m+1}(v) = \left(\frac{1}{2}\right)^{N_v}$. Due to the stopping condition of our algorithm we have $\mu_{m+1}(V) < \frac{1}{n}$. Thus, $\frac{1}{2^{N_v}} < \frac{1}{n}$ and the claim follows. ◁

Next, we bound the number of sets in $\mathcal{S}$. In the following claim we give an upper bound on the total normalized measure of the sets produced by our algorithm.

▷ Claim 12. $\sum_{t=1}^{m} \frac{\mu_t(S_t)}{\mu_t(V)} \leq 4 \ln n + 1$.

Proof. Observe that the total measure at iteration t can be described as follows:

$$\mu_t(V) = \mu_{t-1}(V) - \frac{\mu_{t-1}(S_{t-1})}{2} = \mu_{t-1}(V) \cdot \left(1 - \frac{\mu_{t-1}(S_{t-1})}{2\mu_{t-1}(V)}\right).$$

Since $\mu_m(V) \geq \frac{1}{n}$, we have

$$\frac{1}{n} \leq \mu_m(V) = \mu_1(V) \cdot \prod_{t=1}^{m-1} \left(1 - \frac{\mu_t(S_t)}{2\mu_t(V)}\right) \leq n \cdot \prod_{t=1}^{m-1} e^{-\frac{\mu_t(S_t)}{2\mu_t(V)}} = n \cdot e^{-\frac{1}{2} \cdot \sum_{t=1}^{m-1} \frac{\mu_t(S_t)}{\mu_t(V)}},$$

which implies $\sum_{t=1}^{m-1} \frac{\mu_t(S_t)}{\mu_t(V)} \leq 4 \ln n$. Since $\frac{\mu_m(S_m)}{\mu_m(V)} \leq 1$, we get the desired result. ◁

We obtain an upper-bound on the number of sets in $\mathcal{S}$ which immediately follows from Claim 12 and the fact that $\mu_t(S_t) \geq \Omega(1/k)\mu_t(V)$ for all t.

▶ **Corollary 13.** *The cover $\mathcal{S}$ returned by Algorithm 1 contains $m = O(k \log n)$ sets.*

We prove the existence of a set in an optimal solution with large measure and small cut value.

▶ **Lemma 14.** *Let $\mathcal{P}^* = (P_1^*, \dots, P_k^*)$ be an optimal solution to an ℓ_p-norm Multiway Cut instance and let* OPT *denote the ℓ_p-norm of $\mathcal{P}^*$. For any measure $\mu : V \to \mathbb{R}_{\geq 0}$ on vertices such that $\mu(V) \neq 0$, there exists an $i \in [k]$ such that the following three conditions hold:*

1. $\delta(P_i^*)^p \leq 5 \cdot \mathrm{OPT}^p \cdot \frac{\mu(P_i^*)}{\mu(V)}$
2. $\delta(P_i^*) \leq 5k^{1-1/p} \cdot \mathrm{OPT} \cdot \frac{\mu(P_i^*)}{\mu(V)}$
3. $\mu(P_i^*) \geq \frac{\mu(V)}{2k}$

Proof. Let

$$J = \{j \in [k] : \delta(P_j^*)^p \leq 5 \cdot \mathrm{OPT}^p \cdot \mu(P_j^*)/\mu(V), \delta(P_j^*) \leq 5k^{1-1/p} \cdot \mathrm{OPT} \cdot \mu(P_j^*)/\mu(V)\}$$

be the indices of sets in $\mathcal{P}^*$ that satisfies conditions 1 and 2 in Lemma 14. It is sufficient to show that $\sum_{j\in[k]\setminus J} \mu(P_j^*) < \mu(V)/2$. If $\sum_{j\in[k]\setminus J} \mu(P_j^*) < \mu(V)/2$, then there exists a $j \in J$ such that $\mu(P_j^*) \geq \mu(V)/2k$, which implies this set P_j^* satisfies all three conditions.

We now show that $\sum_{j\in[k]\setminus J} \mu(P_j^*) < \mu(V)/2$. Let $\bar{J}_1 = \{j \in [k] \setminus J : \delta(P_j^*)^p > 5 \cdot \mathrm{OPT}^p \cdot \mu(P_j^*)/\mu(V)\}$ be the indices of sets P_j^* that does not satisfy condition 1. Let $\bar{J}_2 = \{j \in [k] \setminus J : \delta(P_j^*) > 5k^{1-1/p} \cdot \mathrm{OPT} \cdot \mu(P_j^*)/\mu(V)\}$ be the indices of sets P_j^* that does not satisfy condition 2. Note that $[k] \setminus J = \bar{J}_1 \cup \bar{J}_2$. Then, we have

$$\begin{aligned}
\sum_{j\in[k]\setminus J} \mu(P_j^*) &\leq \sum_{j\in\bar{J}_1} \mu(P_j^*) + \sum_{j\in\bar{J}_2} \mu(P_j^*) \\
&\leq \sum_{j\in\bar{J}_1} \mu(V) \cdot \frac{\delta(P_j^*)^p}{5\,\mathrm{OPT}^p} + \sum_{j\in\bar{J}_2} \mu(V) \cdot \frac{\delta(P_j^*)}{5k^{1-1/p}\,\mathrm{OPT}} \\
&\leq \mu(V) \cdot \sum_{j\in[k]} \left(\frac{\delta(P_j^*)^p}{5\,\mathrm{OPT}^p} + \frac{\delta(P_j^*)}{5k^{1-1/p}\,\mathrm{OPT}}\right).
\end{aligned}$$

Since $\mathcal{P}^*$ is a partition with an optimal cost, we have

$$\sum_{i=1}^{k} \delta(P_i^*)^p = \mathrm{OPT}^p.$$

Similarly, we have

$$k^{-1/p} \cdot \mathrm{OPT} = \left(\sum_{i=1}^{k} \frac{1}{k} \cdot \delta(P_i^*)^p\right)^{1/p} \geq \frac{1}{k} \sum_{i=1}^{k} \delta(P_i^*),$$

where the inequality follows from Jensen's inequality. Thus, we have

$$\sum_{j \in [k] \setminus J} \mu(P_j^*) \leq \mu(V) \cdot \sum_{j \in [k]} \frac{\delta(P_j^*)^p}{5\,\mathrm{OPT}^p} + \frac{\delta(P_j^*)}{5k^{1-1/p}\,\mathrm{OPT}} \leq \mu(V) \cdot \frac{2}{5} < \frac{\mu(V)}{2}. \qquad \blacktriangleleft$$

We now prove the main lemma in this section. Specifically, we give two upper-bounds on the ℓ_1-norm and ℓ_p-norm of the cut values of the sets produced by the covering procedure in Algorithm 1, respectively.

Proof of Lemma 8. We already show the number of sets in $\mathcal{S}$ is at most $m = O(k \log n)$ in Corollary 13. By Theorem 10 and Claim 11, we have every S_i contains at most one terminal and all sets in $\mathcal{S}$ covers the entire graph. Thus, it is sufficient to prove the two bounds on the ℓ_1-norm and ℓ_p-norm of the cut values of the sets in $\mathcal{S}$ as shown in Conditions 3 and 4 in the lemma.

Due to Lemma 14 at each iteration t, there exists a set P_i^* in an optimal solution with a measure $\mu_t(P_i^*) \geq \frac{\mu_t(V)}{2k}$ such that

$$\delta(P_i^*) \leq 5 \cdot \min\left\{\left(\frac{\mu_t(P_i^*)}{\mu_t(V)}\right)^{1/p}, k^{1-1/p} \cdot \frac{\mu_t(P_i^*)}{\mu_t(V)}\right\} \cdot \mathrm{OPT}\,.$$

Thus, at each iteration t of Algorithm 1, we have

$$\delta(S_t) \leq O(\alpha) \cdot \min\left\{\left(\frac{\mu_t(P_i^*)}{\mu_t(V)}\right)^{1/p}, k^{1-1/p} \cdot \frac{\mu_t(P_i^*)}{\mu_t(V)}\right\} \cdot \mathrm{OPT}\,.$$

Note that each set S_t is computed by the Unbalanced Terminal Cut algorithm in Theorem 10. Thus, we have $\mu_t(S_t) \geq \Omega(\mu_t(P_i^*))$. Since $\mu_t(S_t) \geq \Omega(\mu_t(P_i^*))$ holds, we obtain (1) $\delta(S_t)^p \leq O(\alpha^p) \cdot \mathrm{OPT}^p \cdot \frac{\mu_t(S_t)}{\mu_t(V)}$; and (2) $\delta(S_t) \leq O(\alpha) \cdot k^{1-1/p} \cdot \mathrm{OPT} \cdot \frac{\mu_t(S_t)}{\mu_t(V)}$. These provide

$$\sum_{t=1}^{m} \delta(S_t)^p \leq O(\alpha^p) \cdot \mathrm{OPT}^p \cdot \sum_{t=1}^{m} \frac{\mu_t(S_t)}{\mu_t(V)},$$
$$\sum_{t=1}^{m} \delta(S_t) \leq O(\alpha) \cdot k^{1-1/p} \cdot \mathrm{OPT} \cdot \sum_{t=1}^{m} \frac{\mu_t(S_t)}{\mu_t(V)}.$$

By Claim 12, we have $\sum_{t=1}^{m} \frac{\mu_t(S_t)}{\mu_t(V)} = O(\log n)$. Then, we get the desired upper bounds on $\sum_{t=1}^{m} \delta(S_t)^p$ and $\sum_{t=1}^{m} \delta(S_t)$. ◀

2.2 Uncrossing and Aggregation Procedures

In this section, we provide procedures that transform the cover of the graph $\mathcal{S}$ produced by the covering procedure into a partition of the graph $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$. Each set P_i in $\mathcal{P}$ contains exactly one terminal in T. With a positive probability, this solution is an $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation for the ℓ_p-Norm Multiway Cut.

▶ **Theorem 15.** *Given a graph $G = (V, E)$ and k terminals $T \subset V$, there exists a polynomial-time algorithm that returns a partition of the graph $\mathcal{P} = \{P_1, P_2 \ldots, P_k\}$ such that with probability at least $3/4 - 1/k$*

1. $|P_i \cap T| = 1$ *for all* $i \in [k]$,
2. $\left(\sum_{i=1}^{k} \delta(P_i)^p\right)^{1/p} \leq O(\log^{1/2} n \log^{1/2+1/p} k) \cdot \mathrm{OPT}$.

Note that the sets in the cover $\mathcal{S}$ are not disjoint. We first use the uncrossing procedure to generate a $m' = O(k \log k)$ partition of the graph $\mathcal{P}' = \{P'_1, P'_2, \ldots, P'_{m'}\}$ from the cover $\mathcal{S}$ produced by the covering procedure. We sample $O(k \log k)$ sets from $\mathcal{S}$ uniformly at random. These sampled sets cover a large fraction of the graph. Then, we generate disjoint sets from these sampled sets by using the uncrossing step in [3]. The uncrossing procedure is shown in Algorithm 2. We then merge these sets in $\mathcal{P}'$ to get a k-partition $\mathcal{P}$ using the aggregation procedure in Algorithm 3.

In the aggregation procedure, we assign all sets in $\mathcal{P}'$ into k parts to get a k-partition. Since $\mathcal{P}' = \{P'_1, P'_2, \ldots, P'_{m'}\}$ is a partition of the graph and each set P'_i contains at most one terminal, there are exactly k sets containing one terminal in $\mathcal{P}'$. Suppose $P'_1, P'_2, \cdots, P'_k$ are these sets containing one terminal. We initially assign these sets $P'_1, P'_2, \cdots, P'_k$ to k parts $P_1, P_2, \ldots, P_k$. Let $\mathcal{Q} = \mathcal{P}' \setminus \{P_1, P_2, \cdots, P_k\}$ be the sets in $\mathcal{P}'$ that does not contain any terminals. We assign all sets in $\mathcal{Q}$ into k parts in a round-robin approach. We sort the sets in $\mathcal{Q}$ by the cut values in descending order and denote it by $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_{m'-k}\}$. We then partition all sets in $\mathcal{Q}$ into k buckets $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ as follows. Consider every k consecutive sets $\{Q_{jk+1}, Q_{jk+2}, \cdots, Q_{(j+1)k}\}$ in $\mathcal{Q}$ for $0 \leq j \leq \lfloor {m'-k}/k \rfloor$. If $jk + i > n$ for $j = \lfloor {m'-k}/k \rfloor$ and some $i \in [k]$, then let $Q_{jk+i} = \varnothing$. For every $i \in [k]$, we assign the set Q_{jk+i} to the bucket $\mathcal{Q}_i$. Finally, we assign each bucket $\mathcal{Q}_i$ to part P_i and set $P_i = P_i \cup (\bigcup_{Q_j \in \mathcal{Q}_i} Q_j)$.

Algorithm 2 Uncrossing Procedure.

Sample $m'' - 1 = 12k \ln k$ sets $\mathcal{S}' = (S'_1, S'_2, \ldots, S'_{m''-1})$ from $\mathcal{S}$ uniformly at random.
Sort sets in $\mathcal{S}'$ in a random order.
Set $P'_i = S'_i \setminus \cup_{j<i} S'_j$ for all $i = 1, 2 \ldots, m'' - 1$.
while there exists a set P'_i such that $\delta(P'_i) > 2\delta(S'_i)$ **do**
    Set $P'_i = S'_i$ and for all $j \neq i$, $P'_j = P'_j \setminus S'_i$.
Set the set $P'_{m''} = V \setminus \cup_{i=1}^{m''-1} P'_i$.
return all non-empty sets P'_i.

Algorithm 3 Aggregation Procedure.

Set $\mathcal{P} = \{P'_i \in \mathcal{P}' : P'_i \cap T \neq \varnothing\} = \{P_1, P_2, \ldots, P_k\}$.
Set $\mathcal{Q} = \mathcal{P}' \setminus \mathcal{P}$.
Sort the sets in $\mathcal{Q} = \{Q_1, \ldots, Q_{m'-k}\}$ by the cut value in descending order.
Partition the sets in $\mathcal{Q}$ into k buckets $\mathcal{Q}_1, \cdots, \mathcal{Q}_k$, where

$$\mathcal{Q}_i = \{Q_j \in \mathcal{Q} : (j-1) \mod k = i - 1\}.$$

Set $P_i = (\bigcup_{Q_j \in \mathcal{Q}_i} Q_j) \cup P_i$ for all $i = 1, \ldots, k$.
return all sets $P_1, P_2, \ldots, P_k$.

We first prove the following lemma on the partition $\mathcal{P}'$ returned by the uncrossing procedure.

▶ **Lemma 16.** *Let $\mathcal{S}$ denote the collection of sets produced by the covering procedure for a graph $G = (V, E)$ and k terminals T. Given $\mathcal{S}$ as input, the uncrossing procedure as shown in Algorithm 2 generates a $m' = O(k \log k)$ partition of the graph $\mathcal{P}' = \{P_1', P_2', \ldots, P_{m'}'\}$ such that with probability at least $3/4 - 1/k$*

1. $|P_i' \cap T| \leq 1$ *for all* $i \in [m']$,
2. $\sum_{i=1}^{m'} \delta(P_i')^p \leq O(\log k \cdot \alpha^p) \cdot \mathrm{OPT}^p$,
3. $\sum_{i=1}^{m'} \delta(P_i') \leq O(k^{1-1/p} \cdot \alpha) \cdot \mathrm{OPT}$,

where $\alpha = \sqrt{\log n \log k}$.

Proof. We consider all sets $P_1', P_2', \ldots, P_{m''}'$ generated in the uncrossing procedure (Algorithm 2), including those empty sets that are not returned. If the set P_i' is empty, we take $\delta(P_i') = 0$. It is easy to see that these sets $P_1', P_2', \ldots, P_{m''}'$ are disjoint, and $\bigcup_{i=1}^{m''} P_i' = V$.

We first show that Algorithm 2 terminates in polynomial time. In graph G, we assume that the ratio between the largest non-infinite edge weight w_{max} and the smallest non-zero edge weight w_{min} is at most $w_{max}/w_{min} \leq n^2/\varepsilon$ for a small constant $\varepsilon > 0$. If the graph does not satisfy this assumption, then we transform it into an instance satisfying this condition as follows. We guess the largest weight of the cut edge in the optimal solution, denoted by W. There are at most $O(n^2)$ different edge weights. Then, we construct a new graph G' with the same vertex set V and edge set E. For every edge $e \in E$, we assign its weight $w'(e)$ in G' to be $w(e)$ if $\varepsilon W/n^2 \leq w(e) \leq W$, $w'(e) = 0$ if $w(e) < \varepsilon W/n^2$, and $w'(e) = \infty$ if $w(e) \geq W$. Thus, the new graph G' satisfies the assumption that $w_{max}/w_{min} \leq n^2/\varepsilon$. Let OPT' be the optimal value of ℓ_p multiway cut on graph G'. We know that $\mathrm{OPT}' \leq \mathrm{OPT}$ since the optimal multiway cut on graph G has a smaller value on graph G'. Suppose we find an α-approximation for ℓ_p multiway cut on graph G'. Then, the same partition on the original graph G has an objective value at most $\alpha \cdot \mathrm{OPT}' + \varepsilon W \leq (\alpha + \varepsilon)\,\mathrm{OPT}$. Hence, this α-approximation solution on G' provides an $(\alpha + \varepsilon)$-approximation on G.

Consider any iteration of Algorithm 2. Let P_i' be the partition of V before the current uncrossing iteration. Suppose we pick a set P_i' such that $\delta(P_i') > 2\delta(S_i')$. For any two subsets $A, B \subseteq V$, we use $\delta(A, B)$ to denote the total weight of edges crossing A and B. Then, we have the ℓ_1-norm of the cut values after this iteration is

$$\begin{aligned}
\delta(S_i') + \sum_{j \neq i} \delta(P_j' \setminus S_i') &\leq \delta(S_i') + \sum_{j \neq i} \delta(P_j') - \delta(P_j', S_i' \setminus P_j') + \delta(S_i', P_j' \setminus S_i') \\
&\leq \delta(S_i') - \delta(P_i') + \delta(S_i') + \sum_{j \neq i} \delta(P_j') \\
&\leq 2\delta(S_i') - 2\delta(P_i') + \sum_{j} \delta(P_j') \leq \sum_{j} \delta(P_j') - 2w_{min},
\end{aligned}$$

where the last inequality is due to $\delta(P_i') > 2\delta(S_i')$ and the minimum non-zero edge weight is w_{min}. Thus, the ℓ_1-norm of the cut values decreases by $2w_{min}$ after each iteration. Since the largest ℓ_1-norm of the cut values is at most $w_{max} n^2$, the total number of iterations is polynomial in n.

We then show that the partition returned by Algorithm 2 satisfies two conditions in the Lemma. We first show that each set P_i' contains at most one terminal. Note that for every $i = 1, 2, \ldots, m'' - 1$, the set P_i' is a subset of $S_i' \in \mathcal{S}$. By Lemma 8, we have

$|P'_i \cap T| \leq |S'_i \cap T| \leq 1$ for all $i = 1, 2, \ldots, m'' - 1$. By Claim 11, every vertex $u \in V$ is covered by at least $\log_2 n$ sets in $\mathcal{S}$. By Corollary 13, the cover $\mathcal{S}$ contains at most $6k \log_2 n$ sets. Thus, a random set in $\mathcal{S}$ covers u with probability at least $1/6k$. For each vertex $u \in V$, the probability that u is not covered by any set in $\mathcal{S}'$ is at most

$$\mathbb{P}\{u \notin \cup_{i=1}^{m''-1} S'_i\} \leq \left(1 - \frac{1}{6k}\right)^{12k \ln k} \leq \frac{1}{k^2}. \tag{1}$$

By the union bound over all terminals, all terminals are covered by $\mathcal{S}'$ with probability at least $1 - 1/k$. Thus, the set $P'_{m''}$ contains no terminal with probability at least $1 - 1/k$.

We now bound the ℓ_1-norm of the cut values of sets $P'_1, P'_2, \ldots, P'_{m''}$. The first two steps in Algorithm 2 can be implemented equivalently by sorting sets in $\mathcal{S}$ in a random order and picking the first $m'' - 1 = 12k \ln k$ sets as $\mathcal{S}'$. Let $S_1, S_2, \ldots, S_m$ be the sets in $\mathcal{S}$ in a random order. Let $\tilde{P}_i = S'_i \setminus \cup_{j<i} S'_j$ for $i = 1, 2, \ldots, m$. Then, for any $i = 1, 2, \ldots, m'' - 1$, the set $\tilde{P}_i$ corresponds to the set P'_i before running the while loop in Algorithm 2.

We first bound the expected ℓ_1-norm of the cut values of sets $\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_{m''-1}$. Note that we have

$$\sum_{i=1}^{m''-1} \delta(\tilde{P}_i) \leq \sum_{i=1}^{m} \delta(\tilde{P}_i).$$

We assign each cut edges $(u, v) \in \partial \tilde{P}_i$ into the following two types: (1) edge (u, v) is cut by a set $\tilde{P}_j$ for $j < i$; (2) edge (u, v) is first cut by the set $\tilde{P}_i$. Let E_i be the set of cut edges that first cut by the set $\tilde{P}_i$. Let $w(E_i) = \sum_{e \in E_i} w(e)$ be the total weight of edges in E_i. Each cut edge is counted twice in $\sum_{i=1}^{m} \delta(\tilde{P}_i)$, while each cut edge is counted exactly once in $\sum_{i=1}^{m} w(E_i)$. Thus, we have

$$\sum_{i=1}^{m} \delta(\tilde{P}_i) = 2 \sum_{i=1}^{m} w(E_i).$$

Note that $E_i \subseteq \partial S_i$ is a subset of edges cut by S_i. Each edge $(u, v) \in \partial S_i$ is a cut edge in E_i after uncrossing if and only if S_i is the first set among all sets that contain node u or node v in the uncrossing sequence. Suppose S_i only contains node u. Then, the probability that (u, v) is contained in E_i is at most the probability that S_i is the first set among all the sets that contain node u in the uncrossing sequence. If a set in $\mathcal{S}$ that contains node v is before set S_i, then this edge (u, v) is not count in E_i. By Claim 11, we have

$$\mathbb{P}\{(u, v) \in E_i\} \leq \mathbb{P}\{S_i \text{ is the first set that contains } u\} \leq \frac{1}{\log_2 n}.$$

Therefore, we have the expected ℓ_1-norm of the cut values of sets $\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_m$ is at most

$$\begin{aligned}\mathbf{E}\left[\sum_{i=1}^{m} \delta(\tilde{P}_i)\right] &= 2 \sum_{i=1}^{m} \mathbf{E}[w(E_i)] = 2 \sum_{i=1}^{m} \sum_{e \in \partial S_i} w(e) \cdot \mathbb{P}\{e \in E_i\} \\ &\leq \frac{2}{\log_2 n} \sum_{i=1}^{m} \delta(S_i) \leq k^{1-1/p} \cdot O(\alpha) \cdot \mathrm{OPT},\end{aligned}$$

where the last inequality is from Lemma 8. At every iteration of the while loop, the ℓ_1-norm of the cut values of sets $P'_1, P'_2, \cdots, P'_{m''-1}$ only decreases. Thus, we have

$$\mathbf{E}\left[\sum_{i=1}^{m''-1} \delta(P'_i)\right] \leq \mathbf{E}\left[\sum_{i=1}^{m''-1} \delta(\tilde{P}_i)\right] \leq k^{1-1/p} \cdot O(\alpha) \cdot \mathrm{OPT}\,.$$

Thus, the expected ℓ_1-norm of the cut values of sets $P'_1, P'_2, \ldots, P'_{m''}$ is

$$\mathbf{E}\left[\sum_{i=1}^{m''} \delta(P'_i)\right] \leq 2 \cdot \mathbf{E}\left[\sum_{i=1}^{m''-1} \delta(P'_i)\right] \leq k^{1-1/p} \cdot O(\alpha) \cdot \mathrm{OPT}\,.$$

To bound the ℓ_p-norm of edge boundaries, we then bound the edge boundary of the last set $P'_{m''}$. We only consider the subsampling process in the uncrossing procedure. We sample $O(k \log k)$ sets from the cover $\mathcal{S}$ uniformly at random. Consider every edge (u, v) in the boundary of sets in cover $\mathcal{S}$. If this edge (u, v) is a cut edge crossing $P'_{m''}$ and $v \in P'_{m''}$, then one of the sets $S_i \in \mathcal{S}$ that contains node u is sampled and node v is not covered by sampled sets. Each set $S_i \in \mathcal{S}$ is sampled with probability $O(\log k/\log n)$. Suppose the set $S_i \in \mathcal{S}$ cuts this edge (u, v) and contains node u. Similar to Equation (1), the probability that node $v \in P'_{m''}$ conditioned on $S_i \in \mathcal{S}'$ is at most $2/k^2$. Thus, we have

$$\begin{aligned}
\mathbf{E}[\delta(P'_{m''})] &= \mathbf{E}\left[w\left\{(u,v) \in \bigcup_{i=1}^{m''-1} \partial S'_i : u \notin P'_{m''} \text{ and } v \in P'_{m''}\right\}\right] \\
&\leq \sum_{i=1}^{m} \sum_{(u,v)\in\partial S_i} w(u,v) \cdot \mathbb{P}\{u \in S_i, S_i \in \mathcal{S}', v \in P'_{m''}\} \\
&\leq O\left(\frac{1}{k^2} \cdot \frac{\log k}{\log n}\right) \cdot \sum_{i=1}^{m} \delta(S_i) \leq O(\alpha) \cdot \mathrm{OPT},
\end{aligned}$$

where the last inequality is due to condition 4 in Lemma 8.

After the while loop, we have $\delta(P'_i) \leq 2\delta(S'_i)$ for all $i = 1, 2, \ldots, m''-1$. Since $\mathbf{E}[\delta(P'_{m''})] \leq O(\alpha) \cdot \mathrm{OPT}$, by Markov's Inequality, we have with probability at least $7/8$ that $\delta(P'_{m''}) \leq O(\alpha) \cdot \mathrm{OPT}$. Since we subsample a fraction $O(\log k/\log n)$ of sets in the cover S uniformly at random, we have

$$\mathbf{E}\left[\sum_{i=1}^{m''-1} \delta(S'_i)^p\right] \leq O\left(\frac{\log k}{\log n}\right) \sum_{i=1}^{m} \delta(S_i)^p.$$

When $\delta(P'_{m''}) \leq O(\alpha) \cdot \mathrm{OPT}$, we have

$$\begin{aligned}
\mathbf{E}\left[\sum_{i=1}^{m''} \delta(P'_i)^p\right] &\leq 2^p \cdot \mathbf{E}\left[\sum_{i=1}^{m''-1} \delta(S'_i)^p\right] + \mathbf{E}\delta(P'_{m''})^p \\
&\leq 2^p \cdot O\left(\frac{\log k}{\log n}\right) \sum_{i=1}^{m} \delta(S_i)^p + O(\alpha^p) \cdot \mathrm{OPT}^p \leq O(\log k \cdot \alpha^p) \cdot \mathrm{OPT}^p,
\end{aligned}$$

where the third inequality is from the condition 3 in Lemma 8. Therefore, we have the conditions 2 and 3 in this lemma hold in expectation with probability at least $7/8$. By Markov's Inequality, we have the conditions 2 and 3 in the lemma hold simultaneously with probability at least $3/4$. Since the condition 1 hold with probability at least $1 - 1/k$, we have all conditions hold with probability at least $3/4 - 1/k$. ◀

Next, we analyze the aggregation procedure, which merges these sets to get a k partition of the graph.

Proof of Theorem 15. By Lemma 16, the partition $P'_1, P'_2, \ldots, P'_{m'}$ returned by the uncrossing procedure (Algorithm 2) satisfies the following three conditions with probability at least $3/4 - 1/k$:

1. $|P'_i \cap T| \le 1$ for all $i \in [m']$,
2. $\sum_{i=1}^{m'} \delta(P'_i)^p \le O(\log k \cdot \alpha^p) \cdot \mathrm{OPT}^p$,
3. $\sum_{i=1}^{m'} \delta(P'_i) \le O(k^{1-1/p} \cdot \alpha) \cdot \mathrm{OPT}$.

We now assume the partition $\mathcal{P}' = \{P'_1, P'_2, \dots, P'_{m'}\}$ given by the uncrossing procedure satisfies these three conditions. Then, we use the aggregation procedure as shown in Algorithm 3 on this partition $\mathcal{P}' = \{P'_1, P'_2, \dots, P'_{m'}\}$ to get a k-partition $\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$. Since each part P_i has exactly one set P'_i containing one terminal, we have $|P_i \cup T| = 1$ for all $i \in [k]$.

We now bound the ℓ_p-norm of the cut values. Let $Q'_i = \bigcup_{j>k, Q_j \in \mathcal{Q}_i} Q_j$ be the union of sets in bucket $\mathcal{Q}_i$ excluding the set with the largest cut in that bucket. Thus, we have each part $P_i = Q'_i \cup Q_i \cup P'_i$ for all $i \in [k]$. By the triangle inequality, we have

$$\left(\sum_{i=1}^{k} \delta(P_i)^p\right)^{1/p} \le \left(\sum_{i=1}^{k} \delta(Q'_i)^p\right)^{1/p} + \left(\sum_{i=1}^{k} \delta(Q_i)^p\right)^{1/p} + \left(\sum_{i=1}^{k} \delta(P'_i)^p\right)^{1/p}.$$

By Lemma 16, the ℓ_p-norm of the cut values of sets $P'_1, P'_2, \dots, P'_k$ is

$$\left(\sum_{i=1}^{k} \delta(P'_i)^p\right)^{1/p} \le O(\log^{1/p} k \cdot \alpha) \cdot \mathrm{OPT}\,.$$

Similarly, we have the ℓ_p-norm of the cut values of sets $Q_1, Q_2, \dots, Q_k$ is

$$\left(\sum_{i=1}^{k} \delta(Q_i)^p\right)^{1/p} \le O(\log^{1/p} k \cdot \alpha) \cdot \mathrm{OPT}\,.$$

We then bound the ℓ_p-norm of the cut values of sets $Q'_1, Q'_2, \dots, Q'_k$. We first bound the cut value of each set Q'_i. Since Q_i are sorted by the cut value in descending order, we have

$$\delta(Q'_i) \le \sum_{j>k, Q_j \in \mathcal{Q}_i} \delta(Q_j) \le \sum_{Q_j \in \mathcal{Q}_k} \delta(Q_j) \le \frac{1}{k} \sum_{Q_j \in \mathcal{Q}} \delta(Q_j),$$

where the second inequality is due to $\delta(Q_{i+zk}) \le \delta(Q_{zk})$ for $z \ge 1$ and the third inequality is because $\mathcal{Q}_k$ contains the smallest cut set for every k consecutive sets. By Lemma 16, we have

$$\delta(Q'_i) \le \frac{1}{k} \cdot \sum_{Q_j \in \mathcal{Q}} \delta(Q_j) \le O(k^{-1/p} \cdot \alpha) \cdot \mathrm{OPT}\,.$$

Therefore, we have ℓ_p-norm of the cut values of sets $Q'_1, Q'_2, \dots, Q'_k$ is at most

$$\left(\sum_{i=1}^{k} \delta(Q'_i)^p\right)^{1/p} \le \left(k \cdot O(k^{-1} \cdot \alpha^p) \cdot \mathrm{OPT}^p\right)^{1/p} = O(\alpha) \cdot \mathrm{OPT}\,.$$

Combining three parts, we get the conclusion. ◀

By Theorem 15, given a graph with n vertices and k terminals, our algorithm finds an $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation for the ℓ_p-Norm Multiway Cut with probability at least $3/4 - 1/k$. We can repeat this algorithm $O(\log 1/\varepsilon)$ times to find an $O(\log^{1/2} n \log^{1/2+1/p} k)$ approximation for the ℓ_p-Norm Multiway Cut with probability at least $1 - \varepsilon$, which proves Theorem 1.

References

1 Saba Ahmadi, Samir Khuller, and Barna Saha. Min-max correlation clustering via multicut. In *Proceedings of the Integer Programming and Combinatorial Optimization*, pages 13–26, 2019.

2 Haris Angelidakis, Yury Makarychev, and Pasin Manurangsi. An improved integrality gap for the Călinescu–Karloff–Rabani relaxation for multiway cut. In *Proceedings of Integer Programming and Combinatorial Optimization*, pages 39–50. Springer, 2017.

3 Nikhil Bansal, Uriel Feige, Robert Krauthgamer, Konstantin Makarychev, Viswanath Nagarajan, Joseph Seffi, and Roy Schwartz. Min-max graph partitioning and small set expansion. *SIAM Journal on Computing*, 43(2):872–904, 2014.

4 Kristóf Bérczi, Karthekeyan Chandrasekaran, Tamás Király, and Vivek Madan. Improving the integrality gap for multiway cut. *Mathematical Programming*, 183(1-2):171–193, 2020.

5 Niv Buchbinder, Joseph Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the Symposium on Theory of Computing*, pages 535–544, 2013.

6 Niv Buchbinder, Roy Schwartz, and Baruch Weizman. Simplex transformations and the multiway cut problem. In *Proceedings of the Symposium on Discrete Algorithms*, pages 2400–2410, 2017.

7 Gruia Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. In *Proceedings of the Symposium on Theory of Computing*, pages 48–52, 1998.

8 Karthekeyan Chandrasekaran and Weihang Wang. ℓ_p-norm multiway cut. *Algorithmica*, 84(9):2667–2701, 2022.

9 Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.

10 Ari Freund and Howard Karloff. A lower bound of $8/(7 + \frac{1}{k-1})$ on the integrality ratio of the Călinescu–Karloff–Rabani relaxation for multiway cut. *Information Processing Letters*, 75(1-2):43–50, 2000.

11 David R Karger, Philip Klein, Cliff Stein, Mikkel Thorup, and Neal E Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In *Proceedings of the Symposium on Theory of Computing*, pages 668–678, 1999.

12 Rajsekar Manokaran, Joseph Naor, Prasad Raghavendra, and Roy Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. In *Proceedings of the Symposium on Theory of Computing*, pages 11–20, 2008.

13 Ankit Sharma and Jan Vondrák. Multiway cut, pairwise realizable distributions, and descending thresholds. In *Proceedings of the Symposium on Theory of Computing*, pages 724–733, 2014.

14 Zoya Svitkina and Éva Tardos. Min-max multiway cut. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization*, pages 207–218, 2004.