

Algorithms for Stable and Perturbation-Resilient Problems

Haris Angelidakis
TTIC
Chicago, IL, USA

Konstantin Makarychev
Northwestern University
Evanston, IL, USA

Yury Makarychev
TTIC
Chicago, IL, USA

ABSTRACT

We study the notion of stability and perturbation resilience introduced by Bilu and Linial [7] and Awasthi, Blum, and Sheffet [2]. A combinatorial optimization problem is α -stable or α -perturbation-resilient if the optimal solution does not change when we perturb all parameters of the problem by a factor of at most α . In this paper, we give improved algorithms for stable instances of various clustering and combinatorial optimization problems. We also prove several hardness results.

- We give an exact algorithm for 2-perturbation resilient instances of clustering problems with natural center-based objectives. The class of clustering problems with natural center-based objectives includes such problems as k -means, k -median, and k -center. Our result improves upon the result of Balcan and Liang [4], who gave an algorithm for clustering $1 + \sqrt{2} \approx 2.41$ perturbation-resilient instances. Our result is tight in the sense that no polynomial-time algorithm can solve $(2 - \epsilon)$ -perturbation resilient instances of k -center unless $NP = RP$, as was shown by Balcan, Haghtalab, and White [3].
- We give an exact algorithm for $(2 - 2/k)$ -stable instances of Minimum Multiway Cut with k terminals, improving the previous result of Makarychev, Makarychev, and Vijayaraghavan [13], who gave an algorithm for 4-stable instances. We also give an algorithm for $(2 - 2/k + \delta)$ -weakly stable instances of Minimum Multiway Cut.
- Finally, we show that there are no *robust* polynomial-time algorithms for $n^{1-\epsilon}$ -stable instances of Set Cover, Minimum Vertex Cover, and Min 2-Horn Deletion (unless $P = NP$).

CCS CONCEPTS

• **Theory of computation** \rightarrow **Facility location and clustering**:

KEYWORDS

perturbation resilience and stability, k -median, k -means

ACM Reference format:

Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. 2017. Algorithms for Stable and Perturbation-Resilient Problems. In *Proceedings of 49th Annual ACM SIGACT Symposium on the Theory of Computing, Montreal, Canada, June 2017 (STOC'17)*, 14 pages. DOI: 10.1145/3055399.3055487

The third author was supported by NSF awards CAREER CCF-1150062 and IIS-1302662. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. STOC'17, Montreal, Canada

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-4528-6/17/06...\$15.00
DOI: 10.1145/3055399.3055487

1 INTRODUCTION

The notion of stability and perturbation resilience was proposed by Bilu and Linial [7] and Awasthi, Blum, and Sheffet [2]. Informally, an instance is Bilu–Linial stable or perturbation-resilient if the optimal solution remains the same when we perturb the instance. The definition was introduced in the context of beyond-the-worst-case analysis and aims to capture a wide class of real-life instances that are computationally easier than worst-case instances. As several authors argue, in instances arising in practice, the optimal solution is often significantly better than all other solutions and thus does not change if we slightly perturb the instance [6, 7].

Definition 1.1. Consider an instance $\mathcal{I} = (G, w)$ of a graph partitioning problem with a set of vertex or edge weights w_i . An instance (G, w') , with weights w'_i , is an α -perturbation ($\alpha \geq 1$) of (G, w) if $w_i \leq w'_i \leq \alpha w_i$ for every vertex/edge i ; that is, an α -perturbation is an instance obtained from the original one by multiplying each weight by a number from 1 to α (the number may depend on i).

Now, consider an instance $\mathcal{I} = (V, d)$ of a clustering problem, where V is a set of points and d is a metric on V . An instance (V, d') is an α -perturbation of (V, d) if $d(u, v) \leq d'(u, v) \leq \alpha d(u, v)$; here, d' does not have to be a metric. If, in addition, d' is a metric, then d' is an α -metric perturbation of (V, d) .

An instance \mathcal{I} of a graph partitioning or clustering problem is α -stable or α -perturbation-resilient if it has a unique optimal solution and every α -perturbation of \mathcal{I} has the same unique optimal solution/clustering as \mathcal{I} . We will refer to α as the stability or perturbation resilience parameter. Adhering to the literature, we call α -stable instances of graph partitioning problems “ α -Bilu–Linial stable” or simply “ α -stable” and α -stable instances of clustering problems “ α -perturbation-resilient”.

It was shown that stable/perturbation-resilient instances of such problems as k -center, k -means, k -median, clustering problems with center-based and min-sum objectives, Max Cut, Minimum Multiway Cut, and TSP – with a sufficiently large value of the stability/resilience parameter α – can be solved exactly in polynomial time [2–4, 6, 7, 13, 14]; meanwhile, the worst-case instances of these problems are NP-hard. Further, for two problems – k -center [3] and Max Cut [13] – tight or almost tight lower and upper bounds on α are known; in addition, for Max k -Cut and Minimum Multicut, strong hardness or non-integrality results are known [13]. However, for many other problems, known lower and upper bounds are not tight, and many interesting questions in the area remain open. Following are some of the most important ones.

PROBLEM 1. *Get a better upper bound on the perturbation resilience parameter for k -median and k -means (arguably the most popular clustering problems). Balcan and Liang [4] showed that $(1 + \sqrt{2})$ -perturbation-resilient instances of k -means and k -median can be solved exactly in polynomial time. Balcan, Haghtalab, and White [3] showed that it is possible to solve 2-perturbation-resilient instances*

of k -center in polynomial time. Is it possible to get a similar result for k -means and k -median, and possibly a unifying algorithm that solves all 2-perturbation-resilient instances of these 3 problems?

PROBLEM 2 (POSED BY ROUGHGARDEN IN HIS LECTURE NOTES [17] AND TALKS [16, 18]). *Get a better upper bound and any lower bound on the stability parameter for Minimum Multiway Cut. The only currently known result [13] states that 4-stable instances of Minimum Multiway Cut can be solved in polynomial time.*

PROBLEM 3. *Algorithmic results for stable/perturbation-resilient instances are especially interesting when the stability/perturbation resilience parameter α is close to 1, since these results are more likely to be relevant in practice. Currently, there are algorithms for 1.8-stable instances of TSP [14], and 2-stable instances of k -center [3]. Find an NP-hard problem such that its α -stable instances can be solved in polynomial time for some $\alpha < 1.8$.*

PROBLEM 4. *Understand whether stable instances of other combinatorial optimization problems can be solved in polynomial time.*

In this paper, we address said problems. We design a simple algorithm for solving 2-perturbation-resilient instances of k -means, k -median, and other clustering problems with *natural center-based objectives*. We show how to solve $(2 - 2/k)$ -stable instances of Minimum Multiway Cut (where k is the number of terminals), as well as weakly stable instances (see below for the definitions). Specifically, we prove that the standard LP relaxation for Minimum Multiway Cut [9] is integral when the instance is $(2 - 2/k)$ -stable. This result also addresses Problem 3, showing that $4/3$ -stable instances of 3-Multiway Cut can be solved in polynomial time. On the other hand, we show that there are $(\frac{4}{3+1/(k-1)} - \epsilon)$ -stable instances of Minimum Multiway Cut, for which the standard LP relaxation is not integral. Finally, we show that there are no robust polynomial-time algorithms even for $n^{1-\epsilon}$ -stable instances of Set Cover, Minimum Vertex Cover, and Min 2-Horn Deletion unless $P = NP$. In the following subsections, we discuss our results in more detail.

Prior work. Awasthi, Blum, and Sheffet [2] initiated the study of perturbation-resilient instances of clustering problems. They offered the definition of a separable center-based objective (s.c.b.o.) and introduced an important center proximity property (see Definition 2.6). They presented an exact algorithm for solving 3-perturbation-resilient instances of clustering problems with s.c.b.o.; they also gave an algorithm for $(2 + \sqrt{3})$ -perturbation-resilient instances of clustering problems with s.c.b.o. that have Steiner points. Additionally, they showed that for every $\alpha < 3$, k -median instances with Steiner points are NP-hard under the α -center proximity property mentioned above (see Definition 2.6), which includes all α -perturbation-resilient instances of the problem. Ben-David and Reyzin [5] showed that, under the same center proximity condition, for every $\epsilon > 0$, k -median, k -center, and k -means instances with no Steiner points that satisfy $(2 - \epsilon)$ -center proximity are NP-hard. Later, Balcan and Liang [4] designed an exact algorithm for $(1 + \sqrt{2})$ -perturbation-resilient instances of problems with s.c.b.o., improving the result of Awasthi, Blum, and Sheffet. Balcan and Liang also studied clustering with the min-sum objective and (α, ϵ) -perturbation resilience (a weaker notion of perturbation resilience, which we do not discuss in this paper). Recently, Balcan, Haghtalab,

and White [3] designed an algorithm for 2-perturbation-resilient instances of symmetric and asymmetric k -center and showed that there is no polynomial-time algorithm for $(2 - \epsilon)$ -perturbation resilient instances of k -center unless $NP = RP$. They also gave an algorithm for 2-perturbation-resilient instances of problems with s.c.b.o. that satisfy a strong additional condition of cluster verifiability. To summarize, in the setting where there are no Steiner points, the best known algorithm for arbitrary s.c.b.o. requires that the instance be $1 + \sqrt{2} \approx 2.4142$ perturbation-resilient [4]; the best known algorithm for k -center requires that the instance be 2-perturbation-resilient, and the latter result cannot be improved [3].

There are several results for stable instances of graph partitioning problems. Bilu and Linial [7] designed an exact polynomial-time algorithm for $O(n)$ -stable instances of Max Cut. Bilu, Daniely, Linial, and Saks [6] improved the result, showing that $O(\sqrt{n})$ -stable instances can be solved in polynomial time. Then, Makarychev, Makarychev, and Vijayaraghavan [13] gave a polynomial-time algorithm for $O(\sqrt{\log n} \cdot \log \log n)$ -stable instances of Max Cut and 4-stable instances of Minimum Multiway Cut (as well as for weakly stable instances of these problems). They also showed that the results for Max Cut are essentially tight, and proved lower bounds for Max k -Cut (for $k \geq 3$) and Minimum Multicut (see [13] for details).

1.1 Our Results for Clustering Problems

In a clustering problem, we are given a metric space (X, d) and an integer parameter k ; our goal is to partition X into k clusters C_1, \dots, C_k so as to minimize the objective function $\mathcal{H}(C_1, \dots, C_k; d)$ (which depends on the problem at hand). The most well-studied and, perhaps, most interesting clustering objectives are k -means, k -median, and k -center. These objectives are defined as follows. Given a clustering C_1, \dots, C_k , the objective is equal to the minimum over all choices of centers $c_1 \in C_1, \dots, c_k \in C_k$ of the following functions:

$$\begin{aligned} \mathcal{H}_{\text{means}}(C_1, \dots, C_k; d) &= \sum_{i=1}^k \sum_{u \in C_i} d(u, c_i)^2; \\ \mathcal{H}_{\text{median}}(C_1, \dots, C_k; d) &= \sum_{i=1}^k \sum_{u \in C_i} d(u, c_i); \\ \mathcal{H}_{\text{center}}(C_1, \dots, C_k; d) &= \max_{i \in \{1, \dots, k\}} \max_{u \in C_i} d(u, c_i). \end{aligned}$$

Note that in the optimal solution each cluster C_i consists of the vertices u that are closer to c_i than to other centers c_j ; i.e. (C_1, \dots, C_k) is the Voronoi partition of X with centers c_1, \dots, c_k . We refer to objectives satisfying this property as *center-based* objectives. We study two closely related classes of center-based objectives – *separable* center-based objectives and *natural* center-based objectives (which we discuss below and formally define in Section 2). We note that k -means, k -median, and k -center are separable and natural center-based objectives.

Metric Perturbation Resilience. The standard definition of perturbation resilience previously considered in the literature (see Definition 1.1) does not require that the perturbation d' be a metric (d' does not have to satisfy the triangle inequality). It is more natural to consider only *metric* perturbations of \mathcal{I} – those perturbations in which d' is a metric. In this paper, we give the definition of

metric perturbation resilience, in which we do require that d' be a metric (see Definition 2.5). Note that every α -perturbation-resilient instance is also α -metric perturbation-resilient.

We define a class of clustering problems with *natural center-based objectives*; an objective is a natural center-based objective if it is representable in the following form. For some functions f_c and $g_u(r)$ (f_c is a function of c , $g_u(r)$ is a function of u and r ; intuitively, f_{c_i} is the cost of having a center at c_i and $g_u(r)$ is the cost of connecting u to a center at distance r from u), we have

$$\mathcal{H}(C_1, \dots, C_k; d) = \min_{c_1 \in C_1, \dots, c_k \in C_k} \sum_{i=1}^k \left(f_{c_i} + \sum_{u \in C_i} g_u(d(u, c_i)) \right) \quad (1)$$

or $\mathcal{H}(C_1, \dots, C_k; d) =$

$$\min_{c_1 \in C_1, \dots, c_k \in C_k} \max \left(\max_{i \in \{1, \dots, k\}} f_{c_i}, \max_{\substack{i \in \{1, \dots, k\} \\ u \in C_i}} g_u(d(u, c_i)) \right). \quad (2)$$

This class includes such problems as k -means, k -median (sum objectives with $f_c = 0$, $g_u(r)$ equal r^2 and r , respectively), and k -center (a max-objective with $f_c = 0$ and $g_u(r) = r$). It also includes a special version of a metric facility location problem, in which the set of facilities \mathcal{F} is a subset of the points that we want to cluster, i.e. $\mathcal{F} \subseteq X$, and each point $c \in \mathcal{F}$ is associated with an opening cost f_c . (Observe that, as stated above, the objective function asks for k facilities, while, in general, we are allowed to open any number of facilities; to resolve this, we simply guess the optimal number and then use the above objective function.)

We present a polynomial-time algorithm for 2-metric perturbation-resilient instances of clustering problems with natural center based objectives; thus, we improve the known requirement on the perturbation resilience parameter α from $\alpha \geq 1 + \sqrt{2} \approx 2.4142$ to $\alpha \geq 2$ and relax the condition on instances from a stronger α -perturbation resilience condition to a weaker and more natural α -metric perturbation resilience condition. In particular, our result improves the requirement for k -median and k -means from $\alpha \geq 1 + \sqrt{2}$ to $\alpha \geq 2$. Our result is optimal for some natural center-based objectives, since $(2 - \epsilon)$ -perturbation-resilient instances of k -center cannot be solved in polynomial time unless $NP = RP$ [3].

THEOREM 1.2. *There exists a polynomial-time algorithm that given any 2-metric perturbation-resilient instance $((X, d), \mathcal{H}, k)$ of a clustering problem with a natural center-based objective, returns the (exact) optimal clustering of X .*

Our algorithm is quite simple. It first runs the single-linkage algorithm to construct the minimum spanning tree on the points of X and then partitions the minimum spanning tree into k clusters using dynamic programming. We note that Awasthi, Blum, and Sheffet [2] also used the single-linkage algorithm together with dynamic programming to cluster 3-perturbation-resilient instances. However, their approach is substantially different from ours: They first find a hierarchical clustering of X using the single-linkage algorithm and then pick k optimal clusters from this hierarchical clustering. This approach fails for α -perturbation-resilient instances with $\alpha < 3$ (see [2]). That is why we do not use the single-linkage

hierarchical clustering in our algorithm, and, instead, partition the minimum spanning tree.

We note that the definitions of *separable* and *natural* center-based objectives are different. However, in Appendix A we consider a slightly strengthened definition of s.c.b.o. and show that every s.c.b.o., under this new definition, is also a natural center-based objective; thus, our result applies to it. We are not aware of any non-pathological objective that satisfies the definition of s.c.b.o. but is not a natural center-based objective.

Finally, we consider clustering with s.c.b.o. and show that the optimal solution for every α -metric perturbation-resilient instance satisfies the α -center proximity property; previously, that was only known for α -perturbation-resilient instances [2]. Our result implies that the algorithms by Balcan and Liang [4] and Balcan, Hagh-talab, and White [3] for clustering with s.c.b.o. and k -center, respectively, apply not only to α -perturbation-resilient but also to α -metric perturbation-resilient instances.

1.2 Our Results for Minimum Multiway Cut

We show that $(2 - 2/k)$ -stable and $(2 - 2/k + \delta)$ -weakly stable instances of Minimum Multiway Cut can be solved in polynomial time (for every $\delta > 0$).

Definition 1.3. In the Minimum Multiway Cut problem, we are given a graph $G = (V, E, w)$ with positive edge weights $w(e)$ and a set of terminals $T = \{s_1, \dots, s_k\} \subset V$. Our goal is to partition the vertices into k sets P_1, \dots, P_k such that $s_i \in P_i$, so as to minimize the total weight of cut edges.

To obtain our results, we extend the framework for solving stable instances of graph partitioning problems developed in [13]. Consider an LP relaxation for a graph partitioning problem. In [13], it was shown that if there is a rounding scheme satisfying two conditions, which we call the approximation and co-approximation conditions, with certain parameters α and β , then the LP relaxation is integral for $(\alpha\beta)$ -stable instances of the problem. In particular, there is an exact polynomial-time algorithm for $(\alpha\beta)$ -stable instances: solve the LP relaxation; if it is integral, output the integral solution corresponding to the LP solution; otherwise, output that the instance is not stable. The algorithm is robust in the sense of Raghavan and Spinrad [15]: if the instance is $(\alpha\beta)$ -stable, the algorithm returns the optimal solution; if it is not, the algorithm either returns the optimal solution or certifies that the instance is not $(\alpha\beta)$ -stable.¹

To use this framework for solving the Minimum Multiway Cut, we have to construct a rounding scheme for the LP relaxation by Călinescu, Karloff, and Rabani [9]. However, rounding this relaxation is a highly non-trivial task; see papers by Sharma and Vondrák [19] and Buchbinder, Schwartz, and Weizman [8] for the state-of-the-art rounding algorithms. Our key observation is that in order to apply the results from [13], it is sufficient to design a rounding scheme that only rounds LP solutions that are very close to integral solutions. We present such a rounding scheme with parameters α and β satisfying $\alpha\beta = 2 - 2/k$ and obtain our results.

¹We do not claim that our algorithms for clustering problems are robust.

1.3 Our Hardness Results

We show that there are $(\frac{4}{3+1/(k-1)} - \epsilon)$ -stable instances of Minimum Multiway Cut, for which the LP relaxation is not integral. Then we prove that there are no *robust* polynomial-time algorithms for $n^{1-\epsilon}$ -stable instances of Set Cover, Minimum Vertex Cover, and Min 2-Horn Deletion (unless $P = NP$). We note that robustness is a very desirable property of algorithms for stable instances. We discuss it in Section 2 (see Definition 2.8). This result, particularly, implies that there are no polynomial-time solvable convex relaxations (e.g., LP or SDP relaxations) that are integral for $n^{1-\epsilon}$ -stable instances of Set Cover, Minimum Vertex Cover, and Min 2-Horn Deletion (unless $P = NP$).

1.4 Overview

In Section 2, we formally define key notions used in this paper. Then, in Section 3, we prove that the optimal solution for every α -metric perturbation-resilient instance satisfies the α -center proximity property. We use this result later in the analysis of our algorithm; also, as noted above, it is of independent interest and implies that previously known algorithms from [3, 4] work under the metric perturbation resilience assumption. In Section 4, we present our algorithm for solving 2-perturbation-resilient instances of problems with natural center-based objectives.

In Section 5, we describe our algorithmic results for Minimum Multiway Cut. In Section 6, we prove our hardness results for Set Cover, Minimum Vertex Cover, Min 2-Horn Deletion, and Minimum Multiway Cut. Finally, in Appendix A, we state our result that if a s.c.b.o. satisfies some additional properties, then it is a natural center-based objective; we prove this result in the full version of the paper.

2 PRELIMINARIES

In this section, we formally define key notions used in this paper: clustering problems, α -metric perturbation resilience, separable center-based and natural center-based objectives.

Definition 2.1. An instance of a *clustering problem* is a tuple $((X, d), \mathcal{H}, k)$ of a metric space (X, d) , objective function \mathcal{H} , and integer number $k > 1$. The objective \mathcal{H} is a function that, given a partition of X into k sets C_1, \dots, C_k and a metric d on X , returns a nonnegative real number, which we call the cost of the partition.

Given an instance of a clustering problem $((X, d), \mathcal{H}, k)$, our goal is to partition X into disjoint (non-empty) sets C_1, \dots, C_k , so as to minimize $\mathcal{H}(C_1, \dots, C_k; d)$. Awasthi et al. [2] gave the following definition for center-based and separable center-based objectives.

Definition 2.2 (Awasthi et al. [2]). A clustering objective is *center-based* if the optimal solution can be defined by k points c_1, \dots, c_k in the metric space, called centers, such that every data point is assigned to its nearest center. Such a clustering objective is *separable* if it further satisfies the following two conditions:

- The objective function value of a given clustering is either a (weighted) sum or the maximum of the individual cluster scores.
- Given a proposed single cluster, its score can be computed in polynomial time.

Formally, this definition does not impose any constraints on the points c_1, \dots, c_k other than the requirement that every $p \in C_i$ is closer to c_i than to c_j for every $j \neq i$. However, in the paper [2], Awasthi et al. (implicitly) assume that c_1, \dots, c_k satisfy an extra condition: Each point c_i must be the *optimal center* for the cluster C_i . In the proof of Fact 2.2, they write: “Furthermore, since the distances within C_i^* were all changed by the same constant factor, c_i^* will still remain an *optimal center* of cluster i ” [emphasis added]. In Definition 2.3, we formally introduce the notion of the optimal center of a cluster C . The optimal center of C should only depend on the cluster C and the metric induced on C ; it should not depend on other clusters. For instance, in k -means, the optimal center of a cluster C_i is the point c that minimizes the objective $\sum_{p \in C_i} d(p, c)^2$. Note that often the optimal center is not unique e.g. if the cluster C consists of two points u and v , then both u and v are optimal centers of C . We denote this set by $\text{center}(C, d|_C)$.

Definition 2.3. We say that \mathcal{H} is a *center-based* objective function if for every metric d on X , there exists an optimal clustering C_1, \dots, C_k of X (i.e., a clustering that minimizes $\mathcal{H}(C_1, \dots, C_k; d)$) satisfying the following condition: there exists sets of optimal centers $\{\text{center}(C_i, d|_{C_i})\}_{i=1, \dots, k}$, such that every data point $p \in C_i$ is closer to any optimal center $c_i \in \text{center}(C_i, d|_{C_i})$ than to any optimal center $c_j \in \text{center}(C_j, d|_{C_j})$ ($i \neq j$). The value of $\text{center}(C, d|_C)$ may depend only on C and $d|_C$.

The objective is *separable* if, additionally, we can define individual cluster scores so that the following holds.

- The cost of the clustering is either the sum (for separable sum-objectives) or maximum (for separable max-objectives) of the cluster scores.
- The score $H(C, d|_C)$ of each cluster C depends only on C , $\text{center}(C, d|_C)$ and $d|_C$, and can be computed in polynomial time (this implies that each set $\text{center}(C, d|_C)$ can be computed in polynomial time as well).

In this paper, we consider a slightly narrower class of *natural* center-based objectives (which we described in the introduction). The class contains most important center-based objectives: k -center, k -means, and k -median, as well as the special variant of the metric facility location objective, as explained in Section 1.1. We are not aware of any reasonable center-based objective that is not a natural center-based objective. Now, we formally define natural center-based objectives.

Definition 2.4. We say that \mathcal{H} is a *natural center-based* objective function for a ground set X , if there exist functions $f : X \rightarrow \mathbb{R}$ and $g : X \times \mathbb{R} \rightarrow \mathbb{R}$ such that $\mathcal{H}(C_1, \dots, C_k; d)$ satisfies Equation (1) or (2) (see Section 1.1). We require that the functions f and g be computable in polynomial time, and that g_u be non-decreasing for every $u \in X$. We call the points c_i that minimize the objective the centers of the clustering.

Now, we formally define metric perturbation and metric perturbation resilience. Since we do not require that the objective \mathcal{H} is homogeneous as a function of the metric d , we introduce two perturbation resilience parameters α_1 and α_2 in the definition, which specify by how much the distances can be contracted and expanded, respectively, in the perturbed instances.

Definition 2.5. Consider a metric space (X, d) . We say that a metric d' is an (α_1, α_2) -metric perturbation of (X, d) , for $\alpha_1, \alpha_2 \geq 1$, if $\alpha_1^{-1}d(u, v) \leq d'(u, v) \leq \alpha_2 d(u, v)$ for every $u, v \in X$. An instance $((X, d), \mathcal{H}, k)$ is (α_1, α_2) -metric perturbation-resilient if for every (α_1, α_2) -metric perturbation d' of d , the unique optimal clustering for $((X, d'), \mathcal{H}, k)$ is the same as for $((X, d), \mathcal{H}, k)$. We say that an instance $((X, d), \mathcal{H}, k)$ is α -metric perturbation-resilient if it is $(\alpha, 1)$ -metric perturbation-resilient.

Note that in the case of a s.c.b.o., the centers of clusters in the optimal solutions for $((X, d), \mathcal{H}, k)$ and $((X, d'), \mathcal{H}, k)$ may differ.

Observe that if the instance $((X, d), \mathcal{H}, k)$ is (α_1, α_2) -metric perturbation-resilient, then $((X, \lambda d), \mathcal{H}, k)$ is $(\lambda\alpha_1, \lambda^{-1}\alpha_2)$ -metric perturbation-resilient for $\lambda \in [\alpha_1^{-1}, \alpha_2]$. Particularly, if $((X, d), \mathcal{H}, k)$ is (α_1, α_2) -metric perturbation-resilient, then $((X, \alpha_2 d), \mathcal{H}, k)$ is $(\alpha_1\alpha_2, 1)$ -metric perturbation-resilient and the optimal solution for $((X, \alpha_2 d), \mathcal{H}, k)$ is the same as for $((X, d), \mathcal{H}, k)$. Thus, to solve an (α_1, α_2) -metric perturbation-resilient instance $((X, d), \mathcal{H}, k)$, it suffices to solve $\alpha = (\alpha_1\alpha_2)$ metric perturbation-resilient instance $((X, \alpha_2 d), \mathcal{H}, k)$. Consequently, we will only consider α -metric perturbation-resilient instances in this paper.

We recall the definition of the α -center proximity property for s.c.b.o., introduced in [2].

Definition 2.6. We say that a clustering C_1, \dots, C_k of X with centers c_1, \dots, c_k satisfies the α -center proximity property if for all $i \neq j$ and $p \in C_i$, we have $d(p, c_j) > \alpha d(p, c_i)$.

Now, we give the definition of weak stability for Multiway Cut [13]. Unlike the definition of stability, this definition does not require that the optimal partition P remain the same when we perturb the instance; instead, it, loosely speaking, requires that the optimal partition to the perturbed instance be “close enough” to P . Let N be a set of partitions that are close to P in some sense; e.g. it may be the set of partitions that can be obtained from P by moving at most a δ fraction of the vertices among sets P_1, \dots, P_k . Then we formally define (γ, N) -stability as follows.

Definition 2.7. Consider an instance $\mathcal{I} = (G, T)$ of Minimum Multiway Cut, where $G = (V, E, w)$ is a weighted graph and $T = \{s_1, \dots, s_k\} \subset V$ is the set of terminals. Denote the optimal multiway cut in G by $P = (P_1, \dots, P_k)$. Let N be a set of partitions that contains P and $\gamma \geq 1$ be a parameter. We say that \mathcal{I} is a (γ, N) -weakly stable instance of Minimum Multiway Cut if for every γ -perturbation $G' = (V, E, w')$ of G , and every partition $P' \notin N$, partition P has a strictly smaller cost than P' in G' .

Note that the notion of weak stability generalizes the notion of stability: an instance is γ -stable if and only if it is $(\gamma, \{P\})$ -weakly stable. Finally, we define the notion of a robust algorithm [15].

Definition 2.8. A robust algorithm for γ -stable instances of a combinatorial optimization problem Π is a polynomial time algorithm that satisfies the following property, when ran on an instance \mathcal{I}_Π :

- If the instance is γ -stable, then it returns the unique optimal solution of \mathcal{I}_Π .
- If the instance is not γ -stable, then it either returns an optimal solution of \mathcal{I}_Π or it reports that the instance is not γ -stable.

It is very desirable to have robust algorithms for stable instances of combinatorial optimization problems. A robust algorithm will

never return a solution which is not optimal even if our instance is not stable (in which case, it may report that the instance is not stable). Thus, we can safely use robust algorithms on instances that are likely to be stable but may as well be non-stable.

3 CENTER PROXIMITY FOR METRIC PERTURBATION RESILIENCE

In this section, we prove that the (unique) optimal solution to an α -metric perturbation-resilient clustering problem satisfies the α -center proximity property. Our proof is similar to the proof of Awasthi, Blum, and Sheffet, who showed that the optimal solution to a (non-metric) α -perturbation-resilient clustering problem satisfies the α -center proximity property.

THEOREM 3.1. *Consider an α -metric perturbation-resilient clustering problem $((X, d), \mathcal{H}, k)$ with a center-based objective. Let C_1, \dots, C_k be the unique optimal solution, and let c_1, \dots, c_k be a set of centers of C_1, \dots, C_k (that is, each $c_i \in \text{center}(C_i, d|_{C_i})$). Then, the following α -center proximity property holds: for all $i \neq j$ and $p \in C_i$, we have $d(p, c_j) > \alpha d(p, c_i)$.*

PROOF. Suppose that for some $i \neq j$ and $p \in C_i$, we have that $d(p, c_j) \leq \alpha d(p, c_i)$. Let $r^* = d(p, c_i)$. Define a new metric d' as follows. Consider the complete graph on X . Assign length $len(u, v) = d(u, v)$ to each edge (u, v) other than (p, c_j) . Assign length $len(p, c_j) = r^*$ to the edge (p, c_j) . Let metric $d'(u, v)$ be the shortest path metric on the complete graph on X with edge lengths $len(u, v)$. Note that $d(p, c_j) \geq d(p, c_i) = r^*$ since $p \in C_i$ and C_1, \dots, C_k is an optimal clustering. Hence, for every (u, v) : $len(u, v) \leq d(u, v)$ and $d'(u, v) \leq d(u, v)$. It is easy to see that

$$d'(u, v) = \min(d(u, v), d(u, p) + r^* + d(c_j, v), d(v, p) + r^* + d(c_j, u)).$$

Observe that since the ratio $d(u, v)/len(u, v)$ is at most $d(p, c_j)/r^* \leq \alpha$ for all edges (u, v) , we have $d(u, v)/d'(u, v) \leq \alpha$ for all u and v . Hence, $d(u, v) \leq \alpha d'(u, v) \leq \alpha d(u, v)$, and consequently, d' is an $(\alpha, 1)$ -metric perturbation of d .

We now show that d' is equal to d within the cluster C_i and within the cluster C_j .

LEMMA 3.2. *For all $u, v \in C_i$, we have $d(u, v) = d'(u, v)$, and for all $u, v \in C_j$, we have $d(u, v) = d'(u, v)$.*

PROOF. I. Consider two points u, v in C_i . We need to show that $d(u, v) = d'(u, v)$. It suffices to prove that

$$d(u, v) \leq \min(d(u, p) + r^* + d(c_j, v), d(v, p) + r^* + d(c_j, u)).$$

Assume without loss of generality that $d(u, p) + r^* + d(c_j, v) \leq d(v, p) + r^* + d(c_j, u)$. We have

$$d(u, p) + r^* + d(c_j, v) = d(u, p) + d(p, c_i) + d(c_j, v) \geq d(u, c_i) + d(c_j, v).$$

Since $v \in C_i$, we have $d(v, c_i) \leq d(v, c_j)$, and thus

$$d(u, p) + r^* + d(c_j, v) \geq d(u, c_i) + d(c_i, v) \geq d(u, v).$$

II. Consider two points u, v in C_j . Similarly to the previous case, we need to show that $d(u, v) \leq d(u, p) + r^* + d(c_j, v)$. Since now

$u \in C_j$, we have $d(u, c_j) \leq d(u, c_i)$. Thus,

$$\begin{aligned} d(u, p) + r^* + d(c_j, v) &= (d(u, p) + d(p, c_i)) + d(c_j, v) \\ &\geq d(u, c_i) + d(c_j, v) \geq d(u, c_j) + d(c_j, v) \\ &\geq d(u, v). \end{aligned} \quad \square$$

By the definition of α -metric perturbation resilience, the optimal clusterings for metrics d and d' are the same. By Lemma 3.2, the distance functions d and d' are equal within the clusters C_i and C_j . Hence, the centers of C_i and C_j w.r.t. metric d' are also the points c_i and c_j , respectively (see Definition 2.3). Thus, by the definition of center-based objective, and since the clustering is unique, we must have $d'(c_i, p) < d'(c_j, p)$, and, consequently, $d(c_i, p) = d'(c_i, p) < d'(c_j, p) = r^* = d(c_i, p)$. We get a contradiction. \square

COROLLARY 3.3. *Consider a 2-metric perturbation-resilient instance. Let C_1, \dots, C_k be an optimal clustering with centers c_1, \dots, c_k . Then each point $u \in C_i$ is closer to c_i than to any point $v \notin C_i$.*

PROOF. Suppose that $v \in C_j$ for some $j \neq i$. By the triangle inequality, $d(u, v) \geq d(u, c_j) - d(v, c_j)$. By Theorem 3.1, $d(u, c_j) > \alpha d(u, c_i)$ and $d(v, c_j) < d(v, c_i)/\alpha$. Thus,

$$\begin{aligned} d(u, v) &\geq d(u, c_j) - d(v, c_j) > \alpha d(u, c_i) - d(v, c_i)/\alpha \\ &\geq \alpha d(u, c_i) - (d(u, c_i) + d(u, v))/\alpha. \end{aligned}$$

In the last inequality, we used the triangle inequality $d(v, c_i) \leq d(u, c_i) + d(u, v)$. Rearranging the terms, we get that $d(u, v) > (\alpha - 1)d(u, c_i)$. Plugging in $\alpha = 2$, we get the desired inequality. \square

4 CLUSTERING ALGORITHM

In this section, we present our algorithm for solving 2-metric perturbation-resilient instances of clustering problems with natural center-based objectives. Our algorithm is based on single-linkage clustering: first, we find a minimum spanning tree on the metric space X (e.g., using Kruskal's algorithm) and then run a dynamic programming algorithm on the spanning tree to find the clusters. We describe the two steps of the algorithm in Sections 4.1 and 4.2. We note that the algorithm only relies on the 2-center proximity property, and thus, it is optimal for this broader class of problems, unless $P = NP$ (see Ben-David and Reyzin [5]).

4.1 Minimum Spanning Tree

At the first phase of the algorithm, we construct a minimum spanning tree on the points of the metric space using Kruskal's algorithm. Kruskal's algorithm maintains a collection of trees. Initially, each tree is a singleton point. At every step, the algorithm finds two points closest to each other that belong to different trees and adds an edge between them. The algorithm terminates when all points belong to the same tree. Let T be the obtained spanning tree.

Let C_1, \dots, C_k be the optimal clustering. The key observation is that each cluster C_i forms a subtree of the spanning tree T .

LEMMA 4.1. *Each cluster C_i in the optimal solution forms a subtree of the spanning tree T . In other words, the unique path between every two vertices $u, v \in C_i$ does not leave the cluster C_i .*

PROOF. Let c_i be the center of C_i . We show that the (unique) path p from u to c_i lies in C_i , and, therefore, the Lemma holds. Let u' be the next vertex after u on the path p . Consider the step at

which Kruskal's algorithm added the edge (u, u') . At that step, u and c_i were in distinct connected components (as p is the only path connecting u and c_i). Thus, $d(u, u') \leq d(u, c_i)$ as otherwise the algorithm would have added the edge (u, c_i) instead of (u, u') . By Corollary 3.3, the inequality $d(u, u') \leq d(u, c_i)$ implies that u' belongs to C_i . Proceeding by induction we conclude that all vertices on the path p belong to C_i . \square

4.2 Dynamic Programming Algorithm

At the second phase, we use dynamic programming to compute the optimal clustering. (We only describe the DP for objectives that satisfy Equation (1), but it is straightforward to make it work for objectives satisfying Equation (2).) We root the tree T at an arbitrary vertex. We denote the subtree rooted at u by T_u . We first assume that the tree is binary. Later, we explain how to transform any tree into a binary tree by adding dummy vertices.

The algorithm partitions the tree into (non-empty) subtrees P_1, \dots, P_k and assigns a center $c_i \in P_i$ to all vertices in the subtree P_i so as to minimize the objective:

$$\sum_{i=1}^k f c_i + \sum_{i=1}^k \sum_{u \in P_i} g_u(d(u, c_i)). \quad (3)$$

Lemma 4.1 implies that the optimal partitioning of X is the solution to this problem.

Let $\text{cost}_u(k', c)$ be the minimum cost of partitioning the subtree T_u into k' subtrees $P_1, \dots, P_{k'}$ and choosing k' centers $c_1, \dots, c_{k'}$ (the cost is computed using the formula (3) with $k = k'$) so that the following conditions hold:

- (1) $u \in P_1$ and $c_1 = c$ (we denote the tree that contains u by P_1 and require that its center be c),
- (2) if $c_1 \in T_u$, then $c_1 \in P_1$ (if the center c_1 of P_1 lies in T_u , then it must be in P_1),
- (3) $c_i \in P_i$ for $i > 1$ (the center c_i of every other tree P_i lies in P_i).

That is, we assume that u belongs to the first subtree P_1 and that c is the center for P_1 . Every center c_i must belong to the corresponding set P_i except for c_1 . However, if $c_1 \in T_u$, then $c_1 \in P_1$.

Denote the children of vertex u by l_u and r_u (recall that we assume that the tree is binary). The cost $\text{cost}_u(k', c)$ is computed using the following recursive formulas: If $c \notin T_{l_u} \cup T_{r_u}$, then

$$\text{cost}_u(j, c) = f c + g_u(d(c, u)) + \min \left(\begin{aligned} &\min \left\{ \text{cost}_{l_u}(j', c') + \text{cost}_{r_u}(j'', c'') : \right. \\ &\quad \left. j' + j'' = j - 1, c' \in T_{l_u}, c'' \in T_{r_u} \right\}, \end{aligned} \quad (4)$$

$$\min \left\{ \text{cost}_{l_u}(j', c') + \text{cost}_{r_u}(j'', c) - f c : \right. \\ \left. j' + j'' = j, c' \in T_{l_u} \right\}, \quad (5)$$

$$\min \left\{ \text{cost}_{l_u}(j', c) + \text{cost}_{r_u}(j'', c'') - f c : \right. \\ \left. j' + j'' = j, c'' \in T_{r_u} \right\}, \quad (6)$$

$$\min \left\{ \text{cost}_{l_u}(j', c) + \text{cost}_{r_u}(j'', c) - 2f c : \right. \\ \left. j' + j'' = j + 1 \right\}. \quad (7)$$

If $c \in T_{l_u}$, then we remove lines (4) and (5) from the formula.

If $c \in T_{r_u}$, then we remove lines (4) and (6) from the formula.

The first term $f_c + g_u(d(c, u))$ is the cost of opening a center in c and assigning u to c . The lines (4–7) correspond to the following cases:

- (4) neither l_u nor r_u is in P_1 ; they are assigned to (trees P_i and P_j with centers) c' and c'' ,
- (5) l_u is not in P_1 , but r_u is in P_1 ; they are assigned to c' and c ,
- (6) l_u is in P_1 , but r_u is not in P_1 ; they are assigned to c and c'' ,
- (7) both l_u and r_u are in P_1 ; they are assigned to c .

For leaves, we set $\text{cost}_u(1, c) = f_c + g_u(d(u, c))$ and $\text{cost}_u(j, c) = \infty$ for $j > 1$. Note that if we want to find a partitioning of T into at most k subtrees, we can use a slightly simpler dynamic program. It is easy to verify that the formulas above hold. The cost of the optimal partitioning of T into k subtrees equals $\min_{c \in X} \text{cost}_{\text{root}}(k, c)$, where root is the root of the tree.

We now explain how to transform the tree T into a binary tree. If a vertex u has more than two children, we add new dummy vertices between u and its children by repeating the following procedure: take a vertex u having more than two children; pick any two children of u : u_1 and u_2 ; create a new child v of u and rehang subtrees T_{u_1} and T_{u_2} to the vertex v . We forbid opening centers in dummy vertices v by setting the opening cost f_v to be infinity. We set the assignment costs g_v to be 0. Note that Lemma 4.1 still holds for the new tree if we place every dummy vertex in the same part P_i as its parent.

5 MINIMUM MULTIWAY CUT

In this section, we present our algorithmic results for Minimum Multiway Cut (recall Definition 1.3). Consider the LP relaxation by Călinescu, Karloff, and Rabani [9] for Minimum Multiway Cut. For every vertex u , there is a vector $\bar{u} = (u_1, \dots, u_k)$ in the LP. In the integral solution corresponding to a partition (P_1, \dots, P_k) , $u_i = 1$, if $u \in P_i$; and $u_i = 0$, otherwise. That is, $\bar{u} = e_i$ (the i -th standard basis vector) if $u \in P_i$.

$$\text{minimize } \frac{1}{2} \sum_{e=(u,v) \in E} w(e) \cdot \|\bar{u} - \bar{v}\|_1$$

subject to

$$\begin{aligned} \bar{s}_j &= e_j && \text{for every } j \in \{1, \dots, k\}, \\ \sum_{i=1}^k u_i &= 1 && \text{for every vertex } u, \\ u_j &\geq 0 && \text{for every vertex } u \text{ and } j \in \{1, \dots, k\}. \end{aligned}$$

Let $d(u, v) = \frac{1}{2} \|\bar{u} - \bar{v}\|_1$. A randomized rounding scheme R is a randomized algorithm that given the instance and an LP solution, outputs a feasible partition $P = (P_1, \dots, P_k)$ of V (we denote the cluster P_i which vertex u belongs to by $P(u)$). We say that a randomized rounding scheme is an (α, β) -rounding if for every LP solution $\{\bar{u}\}$ the following conditions hold:

- **Approximation Condition:** $\Pr(P(u) \neq P(v)) \leq \alpha d(u, v)$.

- **Co-approximation Condition:**

$$\Pr(P(u) = P(v)) \geq \beta^{-1}(1 - d(u, v)).$$

Note that an (α, β) -rounding is a randomized α -approximation algorithm for Minimum Multiway Cut and a β^{-1} -approximation for the complement problem (Maximum Multiway Uncut), the problem whose objective is to maximize the number of uncut edges.

It is shown in [13] that if there is an (α, β) -rounding then the optimal LP solution for every (α, β) -stable instance of Minimum Multiway Cut is integral. Further, [13] presents a $(2, 2)$ -rounding for Minimum Multiway Cut.

In this paper, we show that the conditions on (α, β) -rounding can be substantially relaxed. We say that an LP solution is ε -close to an integral solution if every vertex u is ε -close to some terminal s_i ; that is, for every u there exists i such that $d(u, s_i) \leq \varepsilon$ (here, d is the distance defined by the LP solution). A rounding scheme is an ε -local (α, β) -rounding if for every LP solution that is ε -close to an integral one the approximation and co-approximation conditions hold (but the conditions do not necessarily hold for an LP solution that is not ε close to an integral). Clearly, every (α, β) -rounding is also ε -local (α, β) -rounding. We prove a counterpart of the result from [13] for ε -local (α, β) -rounding.

THEOREM 5.1. *Assume that there exists an ε -local (α, β) -rounding for some $\varepsilon = \varepsilon(n, k) > 0$ (ε may depend on n and k). Then the optimal LP solution for an (α, β) -stable instance of Minimum Multiway Cut is integral.*

As a corollary of Theorem 5.1, we get that the existence of an ε -local (α, β) -rounding scheme implies the existence of a robust algorithm for (α, β) -stable instances of Minimum Multiway Cut. Further, we prove the following theorem for weakly stable instances.

THEOREM 5.2. *Assume that there is a polynomial-time ε -local (α, β) -rounding for some $\varepsilon = \varepsilon(n, k) > 1/\text{poly}(n) > 0$, and further that the support of the distribution of multiway cuts generated by the rounding has polynomial size². Let $\delta > 1/\text{poly}(n) > 0$. Then there is a polynomial-time algorithm for $(\alpha\beta + \delta, N)$ -weakly stable instances of Minimum Multiway Cut. Given an $(\alpha\beta + \delta, N)$ -weakly stable instance, the algorithm finds a partition $P' \in N$ (the algorithm does not know the set N).*

The proofs of Theorems 5.1 and 5.2 are overall similar to the proofs of their counterparts in [13]. The crucial difference, however, is that we do not apply the rounding scheme to the optimal LP solution $\{\bar{u}\}$ (which may be far from an integral solution), but rather we take a convex combination of $\{\bar{u}\}$ and an appropriately chosen integral solution (with weights ε and $1 - \varepsilon$) and get a fractional solution that is ε -close to this integral solution. Then, we apply the (α, β) -rounding to it and proceed essentially in the same way as in [13]. We give proofs of Theorems 5.1 and 5.2 in Appendix B.

We now present an ε -local (α, β) -rounding for Minimum Multiway Cut with $\alpha\beta = 2 - 2/k$ and $\varepsilon = 1/(10k)$. We assume that the LP solution is ε -close to an integral (as otherwise the algorithm may output any solution). Since the LP solution is ε -close to an integral, for every vertex u there exists a unique j such that $d(u, s_j) \leq \varepsilon$. We denote this j by $j(u)$. Note that, in particular, $u_{j(u)} \geq 1 - \varepsilon$ and $u_{j'} \leq \varepsilon$ for $j' \neq j(u)$.

²If we do not make this assumption, we can still get a randomized algorithm for $(\alpha\beta + \delta, N)$ -weakly stable instances.

Rounding scheme R

let $p = 1/k$, $\theta = 6/(5k)$ (note that $\theta > \varepsilon$)
 choose $r \in (0, \theta)$ uniformly at random
 choose $i \in \{1, \dots, k\}$ uniformly at random
 with probability p apply rule **A** to every u
 with probability $1 - p$ apply rule **B** to every u
 rule **A**: if $u_{j(u)} \geq 1 - r$, add u to $P_{j(u)}$; otherwise, add u to P_i
 rule **B**: if $u_i < r$, add u to $P_{j(u)}$; otherwise, add u to P_i
 return partition $P = (P_1, \dots, P_k)$

THEOREM 5.3. *The algorithm is an ε -local (α, β) -rounding for Minimum Multiway Cut for some α and β , with $\alpha\beta = 2 - 2/k$, and $\varepsilon = 1/(10k)$. The algorithm runs in polynomial-time and generates a distribution of multiway cuts with a domain of polynomial size.*

PROOF. First, we show that the algorithm returns a feasible solution. To this end, we prove that the algorithm always adds $u = s_t$ to P_t . Note that $j(u) = t$. If the algorithm uses rule **A**, then $u_{j(u)} = 1 > 1 - r$, and thus it adds u to $P_{j(u)} = P_t$. If the algorithm uses rule **B**, then $u_i \geq r$ only when $i = j(u)$; thus the algorithm adds u to $P_{j(u)} = P_t$, as required. Let $\theta = 6/(5k)$ and

$$\alpha = \frac{2(k-1)}{k^2\theta} = \frac{5}{3} \left(1 - \frac{1}{k}\right) \quad \text{and} \quad \beta = k\theta = \frac{6}{5}.$$

Now we show that the rounding scheme satisfies the approximation and co-approximation conditions with parameters α and β . Consider two vertices u and v . Let $\Delta = d(u, v)$. We verify that the approximation condition holds for u and v . There are two possible cases: $j(u) = j(v)$ or $j(u) \neq j(v)$. Consider the former case first. Denote $j = j(u) = j(v)$. Note that $P(u) \neq P(v)$ if and only if one of the vertices is added to P_i , the other to P_j , and $i \neq j$. Suppose first that rule **A** is applied. Then, $P(u) \neq P(v)$ exactly when $1 - r \in (\min(u_j, v_j), \max(u_j, v_j)]$ and $i \neq j$. The probability of this event (conditioned on the event that rule **A** is applied) is

$$\begin{aligned} \Pr(i \neq j) \cdot \Pr(1 - r \in (\min(u_j, v_j), \max(u_j, v_j))) &= \\ &= \frac{k-1}{k} \cdot \frac{\max(u_j, v_j) - \min(u_j, v_j)}{\theta} \\ &= \frac{k-1}{k} \cdot \frac{|u_j - v_j|}{\theta} \end{aligned}$$

(here we used that $\max(u_j, v_j) \geq 1 - \varepsilon > 1 - \theta$). Now suppose that rule **B** is applied. Then, we have $P(u) \neq P(v)$ exactly when $r \in (\min(u_i, v_i), \max(u_i, v_i)]$ and $i \neq j$. The probability of this event (conditioned on the event that rule **B** is used) is

$$\frac{1}{k} \sum_{i:i \neq j} \Pr(r \in (\min(u_i, v_i), \max(u_i, v_i))) = \frac{1}{k} \sum_{i:i \neq j} \frac{|u_i - v_i|}{\theta}.$$

Thus,

$$\begin{aligned} \Pr(P(u) \neq P(v)) &= p \frac{k-1}{k} \frac{|u_j - v_j|}{\theta} + (1-p) \frac{1}{k} \sum_{i:i \neq j} \frac{|u_i - v_i|}{\theta} \\ &= \frac{k-1}{k^2\theta} \sum_i |u_i - v_i| = \frac{2(k-1)}{k^2\theta} \Delta = \alpha\Delta. \end{aligned}$$

Now consider the case when $j(u) \neq j(v)$. Then the approximation condition holds simply because $\Pr(P(u) \neq P(v)) \leq 1$ and $\alpha\Delta \geq 1$. Namely, we have $\Delta = d(u, v) \geq d(s_{j(u)}, s_{j(v)}) - d(u, s_{j(u)}) -$

$d(v, s_{j(v)}) \geq 1 - 2\varepsilon \geq 1 - 2/30 = 14/15$ and $\alpha \geq \frac{5}{3} \left(1 - \frac{1}{3}\right) = 10/9$; thus, $\alpha\Delta \geq (10/9) \times (14/15) > 1$.

Let us verify that the co-approximation condition holds for u and v . Assume first that $j(u) = j(v)$. Let $j = j(u) = j(v)$. Then, $\Delta = d(u, v) \leq d(u, s_j) + d(v, s_j) \leq 2\varepsilon \leq 1/15$. As we showed, $\Pr(P(u) \neq P(v)) \leq \alpha\Delta$. We get, $\Pr(P(u) = P(v)) \geq 1 - \alpha\Delta \geq \beta^{-1}(1 - \Delta)$, where the last bound follows from the following inequality $\frac{1-\beta^{-1}}{\alpha-\beta^{-1}} \geq \frac{1/6}{5/3-5/6} = \frac{1}{5} \geq \Delta$.

Assume now that $j(u) \neq j(v)$. Without loss of generality, we assume that $u_{j(u)} \leq v_{j(v)}$. Suppose that rule **A** is applied. Event $P(u) = P(v)$ happens in the following disjoint cases:

- (1) $u_{j(u)} \leq v_{j(v)} < 1 - r$ (then both u and v are added to P_i);
- (2) $u_{j(u)} < 1 - r \leq v_{j(v)}$ and $i = j(v)$.

The probabilities that the above happen are $(1 - v_{j(v)})/\theta$ and $(v_{j(v)} - u_{j(u)})/\theta \times (1/k)$, respectively. Note that $d_u \equiv d(u, s_{j(u)}) = \frac{1}{2} (1 - u_{j(u)} + \sum_{t:t \neq j(u)} u_t) = 1 - u_{j(u)}$, since we have $\sum_{t:t \neq j(u)} u_t = 1 - u_{j(u)}$. Similarly, $d_v \equiv d(v, s_{j(v)}) = 1 - v_{j(v)}$. We express the total probability that one of the two cases happens in terms of d_u and d_v (using that $\Delta \geq d(s_{j(u)}, s_{j(v)}) - d_u - d_v = 1 - d_u - d_v$):

$$\begin{aligned} (d_v + (d_u - d_v)/k)/\theta &= ((k-1)d_v + d_u)/(\theta k) \\ &\geq (d_u + d_v)/(\theta k) \geq (1 - \Delta)/(\theta k) \\ &= \beta^{-1}(1 - \Delta). \end{aligned}$$

Now, suppose that rule **B** is applied. Note that if $u_i \geq r$ and $v_i \geq r$, then both u and v are added to P_i , and thus $P(u) = P(v)$. Therefore,

$$\begin{aligned} \Pr(P(u) = P(v) | \text{rule B}) &\geq \Pr(u_i \geq r, v_i \geq r) = \frac{1}{k} \sum_{i=1}^k \frac{\min(u_i, v_i)}{\theta} \\ &= \frac{1}{k\theta} \sum_{i=1}^k \frac{u_i + v_i - |u_i - v_i|}{2} \\ &= \frac{1}{k\theta} (1 - \Delta) = \beta^{-1}(1 - \Delta). \end{aligned}$$

We conclude that

$$\begin{aligned} \Pr(P(u) = P(v)) &\geq p\beta^{-1}(1 - \Delta) + (1-p)\beta^{-1}(1 - \Delta) \\ &= \beta^{-1}(1 - \Delta). \end{aligned}$$

We have verified that both conditions hold for $\alpha = 2(k-1)/(k^2\theta)$ and $\beta = k\theta$. As required, $\alpha\beta = 2 - 2/k$.

The algorithm clearly runs in polynomial-time. Since the algorithm generates only two random variables i and r , and additionally makes only one random decision, the size of the distribution of P is at most $2 \times k \times (nk) = 2k^2n$. \square

From Theorems 5.1, 5.2, and 5.3 we get the main theorem of this section.

THEOREM 5.4. *The optimal LP solution for a $(2 - 2/k)$ -stable instance of Minimum Multiway Cut is integral. Consequently, there is a robust polynomial-time algorithm for solving $(2 - 2/k)$ -stable instances. Further, there is a polynomial-time algorithm that given a $(2 - 2/k + \delta, N)$ -weakly stable instance of Minimum Multiway Cut finds a solution $P' \in N$ (for every $\delta \geq 1/\text{poly}(n) > 0$).*

6 NEGATIVE RESULTS

6.1 Minimum Multiway Cut

In this subsection, we present a lower bound for integrality of stable instances for the LP relaxation for Minimum Multiway Cut by Călinescu, Karloff, and Rabani [9] (which we will refer to as the CKR relaxation). For that, we first make two claims regarding the construction of stable instances and the use of integrality gap examples as lower bounds for integrality of stable instances. We state both claims in the setting of Minimum Multiway Cut, but they can be easily applied to other partitioning problems as well.

CLAIM 6.1. *Given an instance $G = (V, E, w)$, $w : E \rightarrow \mathbb{R}_{\geq 0}$, of Minimum Multiway Cut with terminals $T = \{s_1, \dots, s_k\}$, and an optimal solution $E^* \subseteq E$, for every $\gamma > 1$ and every $\varepsilon \in (0, \gamma - 1)$, the instance $G_{E^*, \gamma} = (V, E, w_{E^*, \gamma})$, where $w_{E^*, \gamma}(e) = \frac{w(e)}{\gamma}$ for $e \in E^*$, and $w_{E^*, \gamma}(e) = w(e)$ for $e \notin E^*$, is a $(\gamma - \varepsilon)$ -stable instance (whose unique optimal solution is E^*).*

PROOF. First, it is easy to see that for every $\gamma > 1$, E^* is the unique optimal solution for $G_{E^*, \gamma}$. We will now prove that $G_{E^*, \gamma}$ is $(\gamma - \varepsilon)$ -stable, for every $\varepsilon \in (0, \gamma - 1)$. For that, we consider any $(\gamma - \varepsilon)$ -perturbation of $G_{E^*, \gamma}$. More formally, this is a graph $G' = (V, E, w')$, where $w'(e) = f(e) \cdot w_{E^*, \gamma}(e)$, and $f(e) \in [1, \gamma - \varepsilon]$ for all $e \in E$. Let $\bar{E} \neq E^*$ be any feasible solution of G' . We have

$$\begin{aligned} w'(\bar{E}) &= \sum_{e \in \bar{E} \cap E^*} f(e)w_{E^*, \gamma}(e) + \sum_{e \in \bar{E} \setminus E^*} f(e)w_{E^*, \gamma}(e) \\ &\geq w'(E^*) - \sum_{e \in E^* \setminus \bar{E}} f(e)w_{E^*, \gamma}(e) + \sum_{e \in \bar{E} \setminus E^*} w_{E^*, \gamma}(e) \\ &> w'(E^*) - \sum_{e \in E^* \setminus \bar{E}} w(e) + \sum_{e \in \bar{E} \setminus E^*} w(e) \quad (\text{since } \gamma > f(e)) \\ &= w'(E^*) - \sum_{e \in E^*} w(e) + \sum_{e \in \bar{E}} w(e) \\ &\geq w'(E^*) \quad (\text{since } \bar{E} \text{ is feasible for } G). \end{aligned}$$

Thus, E^* is the unique optimal solution for every $(\gamma - \varepsilon)$ -perturbation of $G_{E^*, \gamma}$, and so $G_{E^*, \gamma}$ is $(\gamma - \varepsilon)$ -stable. \square

We will now use the above claim to show how an integrality gap example for Minimum Multiway Cut can be converted to a certificate of non-integrality of stable instances.

CLAIM 6.2. *Let G be an instance of Minimum Multiway Cut, such that $OPT/OPT_{LP} = \alpha > 1$, where OPT is the value of an optimal integral Multiway Cut, and OPT_{LP} is the value of an optimal fractional solution. Then, for every $\varepsilon \in (0, \alpha - 1)$, we can construct an $(\alpha - \varepsilon)$ -stable instance such that the CKR relaxation is not integral for that instance.*

PROOF. Let $G = (V, E, w)$ be an instance of Minimum Multiway Cut such that $OPT/OPT_{LP} = \alpha > 1$. Let $\gamma = \alpha - \delta$, for any fixed $\delta \in (0, \alpha - 1)$. Let E^* be an optimal integral solution, i.e. $OPT = \sum_{e \in E^*} w(e)$. By Claim 6.1, for every $\varepsilon' \in (0, \gamma - 1)$, $G_{E^*, \gamma}$ is a $(\gamma - \varepsilon')$ -stable instance whose unique optimal solution is E^* . Let $\{\bar{u}\}_{u \in V}$ be an optimal LP solution for G . We define $d_e = d(u, v) = \frac{1}{2} \|\bar{u} - \bar{v}\|_1$, for every $e = (u, v) \in E$, and we have $OPT_{LP} = \sum_{e \in E} w(e)d_e$. Note that $\{\bar{u}\}_{u \in V}$ is a feasible fractional solution for $G_{E^*, \gamma}$, and we claim

that its cost in $G_{E^*, \gamma}$ is strictly smaller than the (integral) cost of the optimal solution E^* in $G_{E^*, \gamma}$. For that, we have

$$\begin{aligned} w_{E^*, \gamma}(E^*) &= \sum_{e \in E^*} w_{E^*, \gamma}(e) = \frac{1}{\gamma} \sum_{e \in E^*} w(e) = \frac{\alpha}{\alpha - \delta} \sum_{e \in E} w(e)d_e \\ &> \sum_{e \in E} w(e)d_e \geq \sum_{e \in E} w_{E^*, \gamma}(e)d_e, \end{aligned}$$

which implies that the LP is not integral for the instance $G_{E^*, \gamma}$. Setting $\delta = \varepsilon' = \varepsilon/2$ finishes the proof. \square

Claim 6.2 allows us to convert any integrality gap result for the CKR relaxation into a lower bound for non-integrality. Thus, by using the Freund-Karloff integrality gap construction [10], we can deduce that there are $\left(\frac{8}{7+\frac{1}{k-1}} - \varepsilon\right)$ -stable instances of Minimum Multiway Cut for which the CKR relaxation is not integral. An improved integrality gap construction by Angelidakis, Makarychev, and Manurangsi [1] also implies that there are $\left(\frac{6}{5+\frac{1}{k-1}} - \varepsilon\right)$ -stable instances of Minimum Multiway Cut for which the CKR relaxation is not integral. But, with a more careful analysis, we can obtain a stronger lower bound. More formally, we prove the following theorem.

THEOREM 6.3. *For every $\varepsilon > 0$ and $k \geq 3$, there exist $\left(\frac{4}{3+\frac{1}{k-1}} - \varepsilon\right)$ -stable instances of Minimum Multiway Cut with k terminals for which the CKR relaxation is not integral.*

PROOF. We use the Freund-Karloff construction [10], that is, for any k , we construct the graph $G = (V, E, w)$, where the set of vertices is $V = \{1, \dots, k\} \cup \{(i, j) : 1 \leq i < j \leq k\}$, and the set of edges is $E = E_1 \cup E_2$, $E_1 = \{(i, (i, j)), [(i, j), j]\} : 1 \leq i < j \leq k\}$ and $E_2 = \{(i, j), (i', j') : i < j, i' < j', \|(i, i'), (j, j')\| = 3\}$. Here, we use the notation $[u, v]$ to denote an edge, instead of the standard (u, v) , so as to avoid confusion with the tuples used to describe the vertices. The set of terminals is $T = \{1, \dots, k\} \subset V$. The weights are set in the same way as in the Freund and Karloff construction, i.e. the edges in E_1 all have weight 1 and the edges in E_2 all have weight $w = \frac{3}{2k}$. Freund and Karloff proved that by setting the weights in this way, the graph has an optimal solution that assigns every vertex $(i, j), i < j$, to terminal i . Let $E^* \subseteq E$ be the edges cut by this solution. We have $OPT = w(E^*) = \binom{k}{2} + \frac{3}{2k} \cdot 2 \binom{k}{3} = (k-1)^2$. They also proved that an optimal fractional solution assigns each vertex (i, j) to the vector $(e_i + e_j)/2$, and, thus, the (fractional) length of each edge $e \in E$ is $d_e = \frac{1}{2}$. This implies that $OPT_{LP} = \frac{1}{2} \sum_{e \in E} w_e = \frac{1}{2} \cdot \left(2 \binom{k}{2} + \frac{3}{2k} \cdot 3 \binom{k}{3}\right) = OPT / \frac{8}{7+\frac{1}{k-1}}$.

We now scale the weights of all edges in E^* down by a factor $\gamma > 1$, and, by Claim 6.1, obtain a $(\gamma - \varepsilon)$ -stable instance $G_{E^*, \gamma}$, whose unique optimal solution is E^* . The cost of this optimal solution is $OPT_{\gamma} = \frac{1}{\gamma} \cdot OPT$. We consider the same fractional solution that assigns every node (i, j) to the vector $(e_i + e_j)/2$. The fractional cost now is:

$$X_{E^*, \gamma} = \frac{1}{2} \left[\frac{1}{\gamma} \cdot \binom{k}{2} + \frac{3}{2\gamma k} \cdot 2 \binom{k}{3} \right] + \frac{1}{2} \left[\binom{k}{2} + \frac{3}{2k} \binom{k}{3} \right].$$

We want to maintain non-integrality, i.e. we want $OPT_\gamma > X_{E^*, \gamma}$. Thus, we must have

$$\frac{1}{2\gamma}(k-1)^2 > \frac{1}{8}(k-1)(3k-2), \quad \text{which gives } \gamma < \frac{4(k-1)}{3k-2}.$$

This implies that, for every $\varepsilon > 0$, there exist $\left(\frac{4}{3+\frac{1}{k-1}} - \varepsilon\right)$ -stable instances of Minimum Multiway Cut with k terminals that are not integral with respect to the CKR relaxation. \square

6.2 Minimum Vertex Cover

In this subsection, we prove that, under standard complexity assumptions, no robust algorithms (as defined in Definition 2.8) exist for γ -stable instances of Minimum Vertex Cover, even when γ is very large (we precisely quantify this later in this subsection). Before presenting our results, it is worth noting that robustness is a very desirable property of algorithms, since it guarantees that the output is always correct, even when the instance is not stable (and it is usually the case that we do not know whether the input is stable or not). Furthermore, proving that no robust algorithm exists for γ -stable instances of a given problem implies that no LP/SDP or other convex relaxation that is solvable in polynomial time can be integral for γ -stable instances of the problem, thus ruling out the possibility of having an algorithm that solves γ -stable instances by solving the corresponding relaxation. We now turn our attention to Minimum Vertex Cover.

A Minimum Vertex Cover (VC) instance $G = (V, E, w)$, $w : V \rightarrow \mathbb{R}_{\geq 0}$, is called γ -stable, for $\gamma \geq 1$, if it has a unique optimal solution $X^* \subseteq V$, and for every γ -perturbation (i.e. for every instance $G' = (V, E, w')$ where $w(u) \leq w'(u) \leq \gamma w(u)$, for every $u \in V$), the solution X^* remains the unique optimal solution. In order to prove our impossibility result for Vertex Cover, we need the following definition.

Definition 6.4 (GAP-IS). For any $0 < \alpha < \beta$, the (α, β) -GAP-IS problem is a promise problem that takes as input a (vertex-weighted) graph G whose independent set is either strictly larger than β or at most α and asks to distinguish between the two cases, i.e. decide whether G has an independent set of size

- strictly larger than β (i.e. $OPT > \beta$; YES instance)
- at most α (i.e. $OPT \leq \alpha$; NO instance)

We will prove that the existence of a robust algorithm for γ -stable instances of VC would allow us to solve $(\beta/\gamma - \delta, \beta)$ -GAP-IS, for every $\beta > 0$ and arbitrarily small $\delta > 0$.

LEMMA 6.5. *Given a robust algorithm for γ -stable instances of Minimum Vertex Cover, for some $\gamma > 1$, there exists an algorithm that can be used to efficiently solve $(\beta/\gamma - \delta, \beta)$ -GAP-IS, for every $\beta > 0$ and every $\delta \in (0, \beta/\gamma)$.*

PROOF. Given a $(\beta/\gamma - \delta, \beta)$ -GAP-IS instance $G = (V, E, w)$, $w : V \rightarrow \mathbb{R}_{\geq 0}$, we construct the graph $G' = (V', E', w')$, where $V' = V \cup \{s\}$, $E' = E \cup \{(v, s) : v \in V\}$, $w'(v) = w(v)$ for all $v \in V$ and $w'(s) = \beta$. Every vertex cover $X \subseteq V'$ of G' is of one of the following forms:

- $X = V$, with cost $w'(X) = w(V)$.

- $X = (V \setminus I) \cup \{s\}$, where I is an independent set of the original graph G . The cost of X in this case is $w'(X) = w(V) - w(I) + \beta$.

Let $I^* \subseteq V$ denote a maximum independent set of G and $OPT_{IS(G)} = w(I^*)$ denote its cost. Then, an optimal vertex cover is either V or $(V \setminus I^*) \cup \{s\}$. Observe that we can never have $w(V) = w((V \setminus I^*) \cup \{s\})$, since this would imply that $OPT_{IS(G)} = \beta$, and this is impossible, given that G is a $(\beta/\gamma - \delta, \beta)$ -GAP-IS instance.

We now run the robust algorithm for γ -stable instances of VC on G' , and depending on the output Y , we make the following decision:

- $Y = V$: V is the optimal VC of G' , and so $w(V) \leq w(V) - w(I) + \beta$ for all independent sets I of G . This implies that $w(I^*) \leq \beta$, and, since the instance is a $(\beta/\gamma - \delta, \beta)$ -GAP-IS instance, we must have $w(I^*) \leq \beta/\gamma - \delta$. We output **NO**.
- $Y = (V \setminus I^*) \cup \{s\}$ for some (maximum) independent set I^* : We have $w(V) \geq w(V) - w(I^*) + \beta$, and so $w(I^*) \geq \beta$. From the above discussion, this implies that $w(I^*) > \beta$, and so we output **YES**.
- $Y = \text{not stable}$: Since the instance is not γ -stable, it is not hard to see that there must exist an independent set I of G , such that $w(V \setminus I) + \gamma w(I) \geq w(V \setminus I) + \beta$ (since otherwise the instance would be γ -stable with V being the optimal VC), which implies that $w(I) \geq \beta/\gamma$. Thus, $w(I^*) > \beta$, and so we output **YES**.

We designed an algorithm that uses a robust algorithm for γ -stable instances of Minimum Vertex Cover as a black-box and solves the $(\beta/\gamma - \delta, \beta)$ -GAP-IS problem, for every $\beta > 0$ and arbitrarily small $\delta > 0$. \square

We now use the known inapproximability results for Independent Set in conjunction with Lemma 6.5. In particular, we need the following two theorems, the first proved by Zuckerman [20] (also proved earlier by Håstad in [11] under the complexity assumption that $NP \not\subseteq ZPP$), and the second by Khot and Ponnuswami [12].

THEOREM 6.6 (ZUCKERMAN [20]). *It is NP-hard to approximate the Maximum Independent Set to within $n^{1-\varepsilon}$, for every constant $\varepsilon > 0$. Equivalently, it is NP-hard to solve (α, β) -GAP-IS, for $\beta/\alpha = n^{1-\varepsilon}$, for every constant $\varepsilon > 0$.*

THEOREM 6.7 (KHOT AND PONNUSWAMI [12]). *For every constant $\varepsilon > 0$, there is no polynomial time algorithm that approximates the Maximum Independent Set to within $n/2^{(\log n)^{3/4+\varepsilon}}$, assuming that $NP \not\subseteq BPTIME\left(2^{(\log n)^{O(1)}}\right)$.*

Combining Lemma 6.5 with the above two theorems, we obtain the following theorem.

THEOREM 6.8.

- (1) *For every constant $\varepsilon > 0$, there is no robust algorithm for γ -stable instances of Minimum Vertex Cover, for $\gamma = n^{1-\varepsilon}$, assuming that $P \neq NP$.*
- (2) *For every constant $\varepsilon > 0$, there is no robust algorithm for γ -stable instances of Minimum Vertex Cover, for $\gamma = \frac{n}{2^{(\log n)^{3/4+\varepsilon}}}$, assuming that $NP \not\subseteq BPTIME\left(2^{(\log n)^{O(1)}}\right)$.*

As an immediate corollary, we get the same lower bounds for stability for Set Cover, since Minimum Vertex Cover can be formulated as a Set Cover instance.

COROLLARY 6.9.

- (1) For every constant $\varepsilon > 0$, there is no robust algorithm for γ -stable instances of Set Cover, for $\gamma = n^{1-\varepsilon}$, assuming that $P \neq NP$.
- (2) For every constant $\varepsilon > 0$, there is no robust algorithm for γ -stable instances of Set Cover, for $\gamma = \frac{n}{2^{(\log n)^{3/4+\varepsilon}}}$, assuming that $NP \not\subseteq \text{BPTIME}\left(2^{(\log n)^{O(1)}}\right)$.

6.3 Min 2-Horn Deletion

In this subsection, we focus on Min 2-Horn Deletion, and prove that the lower bound for robust algorithms for VC can be extended to this problem as well, since VC can be formulated as a Min 2-Horn Deletion in a convenient way. We start with the definition of Min 2-Horn Deletion and then state and prove the main theorem of this section.

Definition 6.10 (Min 2-Horn Deletion). Let $\{x_i\}_{i \in [n]}$ be a set of boolean variables and let $\mathcal{F} = \{C_j\}_{j \in [m]}$ be a set of clauses on these variables, where each $C \in \mathcal{F}$ has one of the following forms: x_i , \bar{x}_i , $\bar{x}_i \vee x_j$, or $\bar{x}_i \vee \bar{x}_j$. In words, each clause has at most two literals and is allowed to have at most one positive literal. We are also given a weight function $w : \mathcal{F} \rightarrow \mathbb{R}_{\geq 0}$, and the goal is to find an assignment $f : \{x_1, \dots, x_n\} \rightarrow \{\text{true}, \text{false}\}$ such that the weight of the unsatisfied clauses is minimized.

It will be convenient to work with the dual Min 2-Horn Deletion, in which each clause contains at most one negated literal. Observe that the two problems are equivalent, since, given a Min 2-Horn Deletion instance with variables $\{x_i\}_{i \in [n]}$, we can define the variables $y_i = \bar{x}_i$, $i \in [n]$, and substitute them in \mathcal{F} , thus obtaining a dual Min 2-Horn Deletion with the exact same value. We now prove the following theorem.

THEOREM 6.11.

- (1) For every constant $\varepsilon > 0$, there is no robust algorithm for γ -stable instances of Min 2-Horn Deletion, for $\gamma = n^{1-\varepsilon}$, assuming that $P \neq NP$.
- (2) For every constant $\varepsilon > 0$, there is no robust algorithm for γ -stable instances of Min 2-Horn Deletion for $\gamma = \frac{n}{2^{(\log n)^{3/4+\varepsilon}}}$, assuming that $NP \not\subseteq \text{BPTIME}\left(2^{(\log n)^{O(1)}}\right)$.

PROOF. Let us assume that there exists a robust algorithm for γ -stable instances of Min 2-Horn Deletion, for some $\gamma > 1$. We will prove that this would give a robust algorithm for γ -stable instances of VC. For that, we consider any Minimum Vertex Cover instance $G = (V, E, w)$, $w : V \rightarrow \mathbb{R}_{\geq 0}$, and construct an instance $F(G)$ of Min 2-Horn Deletion as follows (for convenience, as explained above, we assume that each clause contains at most one negation, i.e. we construct a dual Min 2-Horn Deletion formula). We introduce variables $\{x_u\}_{u \in V}$ and $|V|+|E|$ clauses, with $C_u := \bar{x}_u$, for every $u \in V$, and $C_{(u,v)} := x_u \vee x_v$, for every $(u, v) \in E$. We also assign weights w' , with $w'(C_u) = w(u)$, $u \in V$, and $w'(C_{(u,v)}) = 1 + \gamma \cdot \sum_{q \in V} w(q)$, for every $(u, v) \in E$.

Observe that an immediate upper bound for the cost of the optimal assignment of $F(G)$ is $\sum_{u \in V} w(u)$, since we can always delete all the clauses C_u and set all variables to *true*. Thus, an optimal assignment never violates a clause $C_{(u,v)}$, $(u, v) \in E$. This

means that in an optimal assignment f^* , for every $(u, v) \in E$, either $f^*(x_u) = \text{true}$ or $f^*(x_v) = \text{true}$. This implies that the set $X(f^*) = \{u \in V : f^*(x_u) = \text{true}\}$ is a feasible vertex cover of G . It also means that the cost of an optimal assignment is $\sum_{u \in V: f^*(x_u) = \text{true}} w(u) = w(X(f^*))$. We will now show that $X(f^*)$ is in fact an optimal vertex cover of G . First, note that the cost of any assignment g (not necessarily optimal) that does not violate any of the clauses $C_{(u,v)}$, $(u, v) \in E$, is $\sum_{u \in V: g(x_u) = \text{true}} w(u)$. Suppose now that there exists a vertex cover $X' \neq X(f^*)$ with cost $w(X') < w(X(f^*))$. Let $g(x_u) = \text{true}$ if $u \in X'$, and $g(x_u) = \text{false}$ if $u \notin X'$. It is easy to see that g does not violate any of the clauses $C_{(u,v)}$, $(u, v) \in E$. Thus, the cost of the assignment g is equal to $\sum_{u \in V: g(x_u) = \text{true}} w(u) = w(X') < w(X(f^*))$, which contradicts the optimality of f^* , and so, we conclude that the set $X(f^*)$ is an optimal vertex cover of G .

We will now show that if $F(G)$ is not γ -stable, then G cannot be γ -stable. First, observe that any γ -perturbation of $F(G)$ has an optimal solution of cost at most $\gamma \cdot \sum_{u \in V} w(u)$, implying that in every γ -perturbation of $F(G)$, an optimal solution only deletes clauses of the form $C_u = x_u$, for $u \in V$. In other words, in every γ -perturbation of $F(G)$, an optimal assignment g defines a feasible vertex cover $X = \{u \in V : g(u) = \text{true}\}$. This also implies that the perturbation of the weights $w(C_{(u,v)})$ cannot change the optimal assignment, and so, the weights of the clauses C_u , $u \in V$, completely specify the optimal value. Moreover, if \tilde{w} is the weight function for a γ -perturbation of $F(G)$ (whose optimal assignment defines the set X as before), we can use the observation of the previous paragraph to conclude that the vertex cover X is optimal for the instance $G' = (V, E, w')$, in which $w'(u) = \tilde{w}(C_u)$ for all $u \in V$. Note that G' is a γ -perturbation of G . Suppose now that $F(G)$ is not γ -stable. Thus, there exists a subset $X \subseteq V$ such that an optimal assignment for $F(G)$ deletes the clauses $\{C_u : u \in X\}$ (i.e. $f(x_u) = \text{true}$ iff $u \in X$) while there exists a γ -perturbation $F'(G)$ of $F(G)$ such that an optimal assignment for $F'(G)$ deletes the clauses $\{C_u : u \in X'\}$ for some $X' \neq X$. As argued, X is an optimal vertex cover for G and X' is an optimal vertex cover for some γ -perturbation of G . Since $X \neq X'$, the instance G is not γ -stable.

We are ready to present our robust algorithm for γ -stable instances of VC. We use the robust algorithm for γ -stable instances of Min 2-Horn Deletion on $F(G)$. Let Y be the output of the algorithm, when ran on the instance $F(G)$:

- $Y = f$, where $f : \{x_u\}_{u \in V} \rightarrow \{\text{true}, \text{false}\}$: As discussed previously, the set $X = \{u \in V : f(x_u) = \text{true}\}$ is an optimal vertex cover for G , and so we output X .
- $Y = \text{not stable}$: We output “not stable”, since, by the previous discussion, the VC instance cannot be γ -stable.

Plugging in the bounds of Theorem 6.8, we obtain our bounds. \square

REFERENCES

- [1] Haris Angelidakis, Yury Makarychev, and Pasin Manurangsi. An Improved Integrality Gap for the Călinescu-Karloff-Rabani Relaxation for Multiway Cut. *CoRR*, abs/1611.05530, 2016.
- [2] Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based Clustering under Perturbation Stability. *Inf. Process. Lett.*, 112(1):49–54, 2012.
- [3] Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-Center Clustering under Perturbation Resilience. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 68:1–68:14, 2016.
- [4] Maria-Florina Balcan and Yingyu Liang. Clustering under Perturbation Resilience. *SIAM J. Comput.*, 45(1):102–155, 2016.

- [5] Shalev Ben-David and Lev Reyzin. Data stability in clustering: A closer look. *Theor. Comput. Sci.*, 558:51–61, 2014.
- [6] Yonatan Bilu, Amit Daniely, Nati Linial, and Michael Saks. On the practically interesting instances of MAXCUT. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, pages 526–537, 2013.
- [7] Yonatan Bilu and Nathan Linial. Are stable instances easy? In *Proceedings of the Symposium on Innovations in Computer Science (ICS)*, pages 332–341, 2010.
- [8] Niv Buchbinder, Roy Schwartz, and Baruch Weizman. Simplex Transformations and the Multiway Cut Problem. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*, pages 2400–2410, 2017.
- [9] Grigori Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for Multiway Cut. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 48–52, 1998.
- [10] Ari Freund and Howard J. Karloff. A lower bound of $8/(7 + (1/k) - 1)$ on the integrality ratio of the Călinescu-Karloff-Rabani relaxation for Multiway Cut. *Inf. Process. Lett.*, 75(1-2):43–50, 2000.
- [11] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 627–636, 1996.
- [12] Subhash Khot and Ashok Kumar Ponnuswami. Better inapproximability results for MaxClique, Chromatic Number and Min-3Lin-Deletion. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 226–237, 2006.
- [13] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Bilu-Linial stable instances of Max Cut and Minimum Multiway Cut. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*, pages 890–906, 2014.
- [14] Matúš Mihalák, Marcel Schöngens, Rastislav Šrámek, and Peter Widmayer. On the complexity of the metric TSP under stability considerations. In *SOFSM 2011: Theory and Practice of Computer Science*, pages 382–393. Springer, 2011.
- [15] Vijay Raghavan and Jeremy Spinrad. Robust algorithms for restricted domains. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*, pages 460–467, 2001.
- [16] Tim Roughgarden. BWCA lecture 8 (recovering graph cuts). <https://youtu.be/QQFakNAhUE>, 2014.
- [17] Tim Roughgarden. S264: Beyond worst-case analysis. lecture #8: Exact recovery in stable cut instances. theory.stanford.edu/~tim/f14/l18.pdf, 2014.
- [18] Tim Roughgarden. Beyond worst-case analysis. <https://youtu.be/thHt1lhLqJA>, 2016.
- [19] Ankit Sharma and Jan Vondrák. Multiway Cut, Pairwise Realizable Distributions, and Descending Thresholds. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 724–733, 2014.
- [20] David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 681–690, 2006.

A UNIVERSAL SEPARABLE CENTER-BASED CLUSTERING OBJECTIVES

In this section, we show that every separable center-based objective, satisfying some additional properties, is a natural center-based objective. To this end, we define a *universal* center-based objective and show that every universal center-based objective is a natural center-based objective.

Loosely speaking, a universal center-based objective is a center-based objective that satisfies two properties, which we discuss now:

- An arbitrary center-based objective is defined on a specific set of points X and can be used to compute the cost of clustering only of the set X (given a metric d on X). **In contrast, a universal objective can be used to compute the cost of clustering of any ground set X .**
- Recall that in every optimal clustering with a center-based objective each point $u \in X$ is closer to the center of its own cluster than to the center of any other cluster. If a partition is not optimal, some points might be closer to the centers of clusters that do not contain them than to the centers of their own clusters. Then, we can move such points to other clusters so as to minimize their distance to the cluster centers. In fact, one of the two steps of Lloyd’s algorithm does exactly this; hence, we call such a transformation a *Lloyd’s improvement*. We

slightly strengthen the definition of a center-based objective by requiring that a **universal objective not increase when we make a Lloyd’s improvement of any – not necessarily optimal – clustering**.

Now we give a few auxiliary definitions and then formally define universal center-based objectives. Since data sets used in applications are usually labeled, we will consider “labeled metric spaces”. We will assume that the cost of clustering of X may depend on the distances between the points in X and point labels (but not on the identities of points).

Definition A.1 (Labeled Metric Space). A metric space labeled with a set of labels L is a pair $((X, d), l)$, where (X, d) is a metric space, and $l : X \rightarrow L$ is a function that assigns a label to each point in X .

Definition A.2 (Isomorphic Labeled Metric Spaces). We say that two metric spaces $((X', d'), l')$ and $((X'', d''), l'')$ labeled with the same set L are isomorphic if there exists an isometry $\varphi : X' \rightarrow X''$ (i.e., φ is a bijection preserving distances: $d'(u, v) = d''(\varphi(u), \varphi(v))$ for all $u, v \in X'$) that preserves labels; i.e. $l'(u) = l''(\varphi(u))$ for all $u \in X'$.

We note that in the definition above the set L may be infinite. We denote the restriction of $((X, d), l)$ to a non-empty subset $C \subset X$ by $((X, d), l)|_C : ((X, d), l)|_C = ((C, d|_C), l|_C)$. Note that the restriction of a metric set labeled with L to a cluster C is also a metric set labeled with L .

Definition A.3. Consider a clustering problem $((X, d), \mathcal{H}, k)$ with a center-based objective. We say that a clustering C'_1, \dots, C'_k is a Lloyd’s improvement of a clustering C_1, \dots, C_k if there exists a set of centers c_1, \dots, c_k of C_1, \dots, C_k (i.e., each $c_i \in \text{center}(C_i, d)$) such that

- $c_i \in C'_i$ (a Lloyd’s improvement does not move the centers to other clusters)
- for every $x \in X$: if $x \in C_i$ and $x \in C'_j$, then $d(x, c_j) \leq d(x, c_i)$ (a Lloyd’s improvement may move point x from C_i to C'_j only if $d(x, c_j) \leq d(x, c_i)$).

Definition A.4 (Universal Objective). We say that \mathcal{H} is a universal center-based clustering objective for a label set L , if for every metric space $((X, d), l)$ labeled with L the problem $((X, d), \mathcal{H}, k)$ is a clustering problem with a separable center-based objective and the following two conditions hold.

- (1) Cluster scores H_l are universal (“can be used on any metric space”): Given any finite metric space $((C, d), l)$ labeled with L , the function $H_l(C, d)$ returns a real number – the cost of C ; and $H_{l'}(C', d') = H_{l''}(C'', d'')$ for any two isomorphic labeled metric spaces $((C', d'), l')$ and $((C'', d''), l'')$.
- (2) If C'_1, \dots, C'_k is a Lloyd’s improvement of C_1, \dots, C_k , then $\mathcal{H}(C'_1, \dots, C'_k; d) \leq \mathcal{H}(C_1, \dots, C_k; d)$.

Note that every natural center-based objective is a universal objective. The label set is the set of pairs (f, g) , where $f \in \mathbb{R}$ is a real number; g is a nondecreasing function from $\mathbb{R}_{\geq 0}$ to \mathbb{R} . Every point $x \in X$ is assigned the label $l(x) = (f_x, h_x)$. The score of a

cluster C equals

$$H_1(C, d) = \min_{c \in C} \left(f_c + \sum_{x \in C} g_x(d(x, c)) \right); \quad (8)$$

$$H_1(C, d) = \min_{c \in C} \left(\max \left(f_c, \max_{x \in C} g_x(d(x, c)) \right) \right). \quad (9)$$

It is easy to see that Lloyd's improvements may only decrease the cost of a clustering, since the functions g_x are non-decreasing. We now show that every clustering problem with universal separable center-based objectives is a problem with natural center-based objectives.

THEOREM A.5. *I. Let $((X, d), \mathcal{H}, k)$ be a clustering problem with a universal center-based separable sum-objective. Then, the scoring function H_1 can be represented as (8) for some nondecreasing functions f and g such that the minimum is attained when c is a center of C ; and thus $\mathcal{H}(C_1, \dots, C_k; d) = \sum_{i=1}^k H_1(C_i, d|_{C_i})$ is a natural center-based objective.*

II. Let $((X, d), \mathcal{H}, k)$ be a clustering problem with a universal center-based separable max-objective. If the cost of any singleton cluster $\{x\}$ equals 0, then the scoring function H_1 can be represented as (9) for some nondecreasing functions f and g such that the minimum is attained when c is a center of C ; and thus $\mathcal{H}(C_1, \dots, C_k; d) = \max_{i \in \{1, \dots, k\}} H_1(C_i, d|_{C_i})$ is a natural center-based objective.

We defer the proof of this theorem to the full version of the paper (available on arXiv).

B PROOFS OF THEOREM 5.1 AND THEOREM 5.2

In this section, we prove Theorems 5.1 and Theorem 5.2. The proofs are very similar to the proofs from [13]; but we make a key observation that it is sufficient to only require that the rounding schemes are ε -local (see Section 5 for the definitions).

We use the following lemma by Bilu and Linial [7] (the lemma holds for any graph partitioning problem; here, we state it specifically for Minimum Multiway Cut).

LEMMA B.1 (BILU AND LINIAL [7]). *Consider a γ -stable instance of Minimum Multiway Cut. Let P be the optimal multiway cut and E_{cut} be the set of edges cut by P . Let $P' \neq P$ be any other multiway cut and E'_{cut} be the set of edges cut by P' . Then,*

$$\gamma w(E_{cut} \setminus E'_{cut}) < w(E'_{cut} \setminus E_{cut}).$$

PROOF OF THEOREM 5.1. Consider an $(\alpha\beta)$ -stable instance of Minimum Multiway Cut. Let P be the optimal multiway cut and E_{cut} be the set of edges cut by P .

Assume that there is an optimal LP solution, which is not integral. Denote it by $\{\bar{u}^{LP}\}$. Let $\{\bar{u}^{INT}\}$ be the LP solution corresponding to the optimal combinatorial solution. Consider a convex combination $\{\bar{u}\}$ of solutions $\{\bar{u}^{LP}\}$ and $\{\bar{u}^{INT}\}$, defined by $\bar{u} = \varepsilon \bar{u}^{LP} + (1 - \varepsilon) \bar{u}^{INT}$. Note that the cost of $\{\bar{u}^{LP}\}$ is at most the cost of $\{\bar{u}^{INT}\}$, and thus, by the convexity of the LP objective, the cost of $\{\bar{u}\}$ is at most the cost of $\{\bar{u}^{INT}\}$. Further, note that $\{\bar{u}\}$ is not integral, but it is ε -close to an integral solution (as defined in Section 5). Hence, we can apply our ε -local (α, β) -rounding scheme to it.

Let $d(u, v) = \frac{\|\bar{u} - \bar{v}\|_1}{2}$. Let P' be a random multiway cut obtained by rounding $\{\bar{u}\}$, and let E'_{cut} be the set of edges cut by P' . Since

the solution \bar{u} is not integral, $P' \neq P$ with non-zero probability. From $(\alpha\beta)$ -stability of the instance and Lemma B.1, we get that

$$(\alpha\beta)w(E_{cut} \setminus E'_{cut}) < w(E'_{cut} \setminus E_{cut}) \text{ unless } P' = P,$$

and therefore (here we use that $\Pr(P \neq P') > 0$),

$$(\alpha\beta)\mathbb{E}[w(E_{cut} \setminus E'_{cut})] < \mathbb{E}[w(E'_{cut} \setminus E_{cut})]. \quad (10)$$

Let

$$LP_+ = \sum_{(u,v) \in E_{cut}} w(u, v)(1 - d(u, v)) \quad \text{and}$$

$$LP_- = \sum_{(u,v) \in E \setminus E_{cut}} w(u, v)d(u, v).$$

From the approximation and co-approximation conditions that the rounding scheme satisfies, we get

$$\begin{aligned} \mathbb{E}[w(E_{cut} \setminus E'_{cut})] &= \sum_{(u,v) \in E_{cut}} w(u, v) \Pr((u, v) \notin E'_{cut}) \\ &\geq \sum_{(u,v) \in E_{cut}} w(u, v) \beta^{-1} (1 - d(u, v)) = \beta^{-1} LP_+, \\ \mathbb{E}[w(E'_{cut} \setminus E_{cut})] &= \sum_{(u,v) \in E \setminus E_{cut}} w(u, v) \Pr((u, v) \in E'_{cut}) \\ &\leq \sum_{(u,v) \in E \setminus E_{cut}} w(u, v) \alpha d(u, v) = \alpha LP_-. \end{aligned}$$

Using inequality (10), we conclude that $LP_+ < LP_-$. On the other hand, from the formulas for LP_+ and LP_- , we get

$$LP_+ - LP_- = w(E_{cut}) - \sum_{(u,v) \in E} w(u, v)d(u, v) \geq 0,$$

since the cost of the LP solution $\{\bar{u}\}$ is at most the cost of $\{\bar{u}^{INT}\}$. We get a contradiction, which concludes the proof. \square

Now we proceed with the proof of Theorem 5.2. We use the following lemma from [13].

LEMMA B.2. *Consider a (γ, N) -stable instance of Minimum Multiway Cut. Let P be the minimum multiway cut. Then for every multiway cut $P' \notin N$, we have*

$$\gamma w(E_{cut} \setminus E'_{cut}) < w(E'_{cut} \setminus E_{cut}),$$

where E_{cut} is the set of edges cut by P and E'_{cut} is the set of edges cut by P' .

LEMMA B.3. *Suppose that there is a polynomial-time ε -local (α, β) -rounding, where $\varepsilon \geq 1/\text{poly}(n) > 0$. Let $\delta \geq 1/\text{poly}(n) > 0$. Then there is a polynomial-time algorithm that, given an $(\alpha\beta + \delta, N)$ -stable instance of Minimum Multiway Cut and a feasible multiway cut P° , it does the following*

- if $P^\circ \notin N$, it finds a multiway cut P' such that

$$\text{cost}(P') - \text{cost}(P) \leq (1 - \tau) (\text{cost}(P^\circ) - \text{cost}(P)),$$

where P is the minimum multiway cut, $\text{cost}(P)$, $\text{cost}(P^\circ)$, $\text{cost}(P')$ are the costs of P , P° , and P' , respectively; $\tau = \frac{\varepsilon\delta\beta^{-1}}{\alpha\beta + \delta} \geq \frac{1}{\text{poly}(n)} > 0$.

- if $P^\circ \in N$, it either returns a multiway cut P' better than P° or certifies that $P^\circ \in N$.

PROOF. Let E_{cut}° be the set of edges cut by P° . Define edge weights $w'(u, v)$ by

$$w'(u, v) = \begin{cases} w(u, v), & \text{if } (u, v) \in E_{cut}^\circ, \\ (\alpha\beta) w(u, v), & \text{otherwise.} \end{cases}$$

We solve the LP relaxation for Minimum Multiway Cut with weights $w'(u, v)$. Let $\{\bar{u}^{LP}\}$ be the optimal LP solution. Let $\{\bar{u}^{INT}\}$ be the LP solution corresponding to P° (here, the definition of $\{\bar{u}^{INT}\}$ differs from that in Theorem 5.1). As in the proof of Theorem 5.1, we define a convex combination of solutions $\{\bar{u}^{LP}\}$ and $\{\bar{u}^{INT}\}$: $\bar{u} = \varepsilon \bar{u}^{LP} + (1 - \varepsilon) \bar{u}^{INT}$. Note that $\{\bar{u}\}$ is ε -close to an integral solution. Hence we can apply our ε -local (α, β) -rounding scheme to it.

Let $d(u, v) = \|\bar{u} - \bar{v}\|_1/2$, $d^{INT}(u, v) = \|\bar{u}^{INT} - \bar{v}^{INT}\|_1/2$, and $d^{LP}(u, v) = \|\bar{u}^{LP} - \bar{v}^{LP}\|_1/2$; let d^{OPT} be the distance defined by the optimal multiway cut P . From the subadditivity of the ℓ_1 -norm, we get

$$d(u, v) \leq (1 - \varepsilon)d^{INT}(u, v) + \varepsilon d^{LP}(u, v). \quad (11)$$

We apply the ε -local (α, β) -rounding procedure to the solution $\{\bar{u}\}$ and get a distribution of random multiway cuts $P' = (P'_1, \dots, P'_k)$. Let E'_{cut} be the set of edges cut by P' . Define

$$\begin{aligned} LP_+ &= \sum_{(u, v) \in E_{cut}^\circ} w'(u, v)(1 - d(u, v)) \\ &= \sum_{(u, v) \in E_{cut}^\circ} w(u, v)(1 - d(u, v)), \\ LP_- &= \sum_{(u, v) \in E \setminus E_{cut}^\circ} w'(u, v) d(u, v) \\ &= (\alpha\beta) \sum_{(u, v) \in E \setminus E_{cut}^\circ} w(u, v) d(u, v). \end{aligned}$$

We have,

$$\begin{aligned} \mathbb{E}[w(E_{cut}^\circ \setminus E'_{cut})] &= \sum_{(u, v) \in E_{cut}^\circ} w(u, v) \Pr((u, v) \notin E'_{cut}) \\ &\geq \beta^{-1} \sum_{(u, v) \in E_{cut}^\circ} w(u, v) (1 - d(u, v)) \\ &= \beta^{-1} LP_+, \end{aligned}$$

$$\begin{aligned} \mathbb{E}[w(E'_{cut} \setminus E_{cut}^\circ)] &= \sum_{(u, v) \in E \setminus E_{cut}^\circ} w(u, v) \Pr((u, v) \in E'_{cut}) \\ &\leq \alpha \sum_{(u, v) \in E \setminus E_{cut}^\circ} w(u, v) d(u, v) = \beta^{-1} LP_-. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}[w(E_{cut}^\circ) - w(E'_{cut})] &= \mathbb{E}[w(E_{cut}^\circ \setminus E'_{cut})] - \mathbb{E}[w(E'_{cut} \setminus E_{cut}^\circ)] \\ &\geq \beta^{-1}(LP_+ - LP_-). \end{aligned}$$

Observe that (here, we use (11))

$$\begin{aligned} LP_+ - LP_- &= \sum_{(u, v) \in E} w'(u, v)(d^{INT}(u, v) - d(u, v)) \\ &\geq \varepsilon \sum_{(u, v) \in E} w'(u, v)(d^{INT}(u, v) - d^{LP}(u, v)). \end{aligned}$$

Since $\{\bar{u}^{LP}\}$ is an optimal LP solution for the instance with weights w' , we have

$$\sum_{(u, v) \in E} w'(u, v) d^{LP}(u, v) \leq \sum_{(u, v) \in E} w'(u, v) d^{OPT}(u, v).$$

Therefore, if $P^\circ \notin N$, using Lemma B.2, we get

$$\begin{aligned} LP_+ - LP_- &\geq \varepsilon \sum_{(u, v) \in E} w'(u, v)(d^{INT}(u, v) - d^{OPT}(u, v)) \\ &= \varepsilon (w'(E_{cut}^\circ \setminus E_{cut}) - w'(E_{cut} \setminus E_{cut}^\circ)) \\ &= \varepsilon (w(E_{cut}^\circ \setminus E_{cut}) - (\alpha\beta)w(E_{cut} \setminus E_{cut}^\circ)) \\ &\geq \frac{\varepsilon\delta}{\alpha\beta + \delta} w(E_{cut}^\circ \setminus E_{cut}) \\ &\geq \frac{\varepsilon\delta}{\alpha\beta + \delta} (w(E_{cut}^\circ) - w(E_{cut})). \end{aligned}$$

Thus, $\mathbb{E}[w(E_{cut}^\circ) - w(E'_{cut})] \geq \frac{\varepsilon\delta\beta^{-1}}{\alpha\beta + \delta} (w(E_{cut}^\circ) - w(E_{cut}))$, or equivalently

$$\mathbb{E}[w(E'_{cut}) - w(E_{cut})] \leq \left(1 - \frac{\varepsilon\delta\beta^{-1}}{\alpha\beta + \delta}\right) (w(E_{cut}^\circ) - w(E_{cut})).$$

Hence, if $P^\circ \in N$, for some multiway cut P' in the distribution, we have

$$w(E'_{cut}) - w(E_{cut}) \leq \left(1 - \frac{\varepsilon\delta\beta^{-1}}{\alpha\beta + \delta}\right) (w(E_{cut}^\circ) - w(E_{cut})).$$

We can efficiently find this multiway cut, since the distribution of P' has a support of polynomial size.

Note that the algorithm does not know whether $P^\circ \in N$ or not; it tries all multiway cuts P' and finds the best one P'' . If P'' is better than P° , the algorithm returns P'' ; otherwise, it certifies that $P^\circ \in N$. \square

PROOF OF THEOREM 5.2. We assume that all edge costs are integers between 1 and some W . Let C^* be the cost of the optimal solution. We start with an arbitrary feasible multiway cut $P^{(0)}$. Denote its cost by $C^{(0)}$. Let $T = \lceil \log_{1/(1-\tau)} C^{(0)} \rceil + 2 = O(n^2 \tau \log W)$ (note that T is polynomial in the size of the input). We iteratively apply the algorithm from Lemma B.3 T times: first we get a multiway cut $P^{(1)}$ from $P^{(0)}$, then $P^{(2)}$ from $P^{(1)}$, and so on. Finally, we get a multiway cut $P^{(T)}$. If at some point the algorithm does not return a multiway cut, but certifies that the current multiway cut $P^{(i)}$ is in N , we output $P^{(i)}$ and terminate the algorithm. We assume below that that does not happen, and we get multiway cuts $P^{(0)}, \dots, P^{(T)}$. Denote the cost of $P^{(i)}$ by $C^{(i)}$. Note that $C^{(0)} > C^{(1)} > \dots > C^{(T)} \geq C^*$. Further, if $C_i \notin N$ then $C^{(i+1)} - C^* \leq (1 - \tau)(C^{(i)} - C^*)$ and thus $C^{(i+1)} - C^{(T)} \leq (1 - \tau)(C^{(i)} - C^{(T)})$. Observe that we cannot have $C^{(i+1)} - C^{(T)} \leq (1 - \tau)(C^{(i)} - C^{(T)})$ for every i , because then we would have

$$C^{(T-1)} - C^{(T)} \leq (1 - \tau)^{T-1} (C^{(0)} - C^{(T)}) \leq (1 - \tau)^{T-1} C^{(0)} < 1,$$

which contradicts to our assumption that all edge weights are integral and, consequently, $C^{(T-1)} - C^{(T)}$ is a positive integer number. We find i such that $C^{(i+1)} - C^{(T)} > (1 - \tau)(C^{(i)} - C^{(T)})$ and output $P^{(i)}$. We are guaranteed that $P^{(i)} \in N$. \square