Perturbations, Optimization, and Statistics

Editors:

Tamir Hazan tamir.hazan@technion.ac.il
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

George Papandreou gpapan@google.com
Google Inc.
340 Main St., Los Angeles, CA 90291 USA

Daniel Tarlow dtarlow@microsoft.com
Microsoft Research
Cambridge, CB1 2FB, United Kingdom

This is a draft version of the author chapter.

The MIT Press
Cambridge, Massachusetts
London, England

1 Bilu–Linial Stability

Konstantin Makarychev komakary@microsoft.com
Microsoft Research
Redmond, WA, USA

Yury Makarychev yury@ttic.edu
Toyota Technological Institute at Chicago
Chicago, IL, USA

This chapter describes recent results on Bilu–Linial stability, also known as perturbation resilience. It offers an overview of the subject and presents algorithms for stable and weakly stable instances of graph partitioning and clustering problems, including Max Cut, Minimum Multiway Cut, k-center, and clustering problems with separable center-based objectives.

1.1 Introduction

In this chapter, we survey recent research on instance stability and perturbation resilience. Many discrete optimization problems in machine learning, operations research, and other areas are NP-hard. For many of them, not only the exact but even a good approximate solution cannot be found efficiently in the worst case. At the same time, instances appearing in real life can often be solved exactly or almost exactly. This raises the following question:

Why are real-life instances often significantly easier than worst-case instances?

To formally study this question, we must define a model for real-life instances. The two most popular approaches are either to assume that a real-

life instance has certain structural properties, or to assume that it is generated by a random or semi-random process. Both approaches are very natural and have led to the discovery of many interesting results. In this chapter, we study the former approach, focusing on stable instances of clustering and graph partitioning problems. We refer the reader to several papers describing the latter approach (Blum and Spencer, 1995; Feige and Kilian, 1998; Mathieu and Schudy, 2010; Makarychev et al., 2012, 2014a, 2013, 2015; Feige et al., 2015).

Instance stability, or perturbation resilience, was introduced by Bilu and Linial (2010). Informally, an instance is Bilu–Linial stable if the optimal solution does not change when we perturb the instance.

Definition 1.1. *Consider an instance of a graph partitioning problem, a graph $G = (V, E, w)$ with a set of edge weights w_e. An instance $G' = (V, E, w')$ is an α-perturbation ($\alpha \geq 1$) of G if $w(e) \leq w'(e) \leq \alpha w(e)$; that is, if we can obtain the perturbed instance from the original by multiplying the weight of each edge by a number from 1 to α (the number may be different for every edge).*

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 d.

Definition 1.2. *An instance $\mathcal{I}$ is α-stable if every α-perturbation of $\mathcal{I}$ has the same optimal solution as $\mathcal{I}$.*

Adhering to the literature, we will refer to α-stable instances of graph partitioning problems as "Bilu–Linial stable" and to α-stable instances of clustering problems as "α-perturbation resilient". Additionally, for clustering problems, we will consider a weaker, and perhaps somewhat more natural, notion of α-metric perturbation resilience.

Definition 1.3. *An instance (V, d) of a clustering problem is α-metric perturbation resilient if every α-metric perturbation of (V, d) has the same optimal solution as $\mathcal{I}$.*

Why is it reasonable to assume that many real-life instances are stable? As Bilu and Linial (2010); Balcan et al. (2009); Bilu et al. (2013) argue, the reason is that often the optimal solution "stands out" among all other solutions — it is significantly better than all other solutions, and, therefore, the optimal solution remains the same even if we slightly perturb the instance. Also, we are often interested not in optimizing the objective function per se, but rather in finding the "true" clustering or partitioning.

problem	main results	reference
Max Cut & 2-correlation clustering	$O(\sqrt{\log n}\log\log n)$ (incl. weakly stable instances) SDP gap and hardness result	Makarychev et al. (2014b)
Min Multiway Cut	4, (incl. weakly stable instances)	Makarychev et al. (2014b)
Max k-Cut	hardness for ∞-stable instances	Makarychev et al. (2014b)
sym./asym. k-center	2 hardness for $(2-\varepsilon)$-pert. resil.	Balcan et al. (2015)
s.c.b. objective	$1+\sqrt{2}$ $(2+\sqrt{3},\varepsilon)$ for k-median 2, assuming cluster verifiability	Balcan and Liang (2016) Balcan et al. (2015)
s.c.b., Steiner points	$2+\sqrt{3}$	Awasthi et al. (2012)
min-sum objective	$O(\rho)$ and $(O(\rho),\varepsilon)$, where ρ is the ratio between the sizes of the largest and smallest clusters	Balcan and Liang (2016)
TSP	1.8	Mihalák et al. (2011)

Table 1.1: The table summarizes some known results for Bilu–Linial stability. It shows a number α if there is an algorithm for α-stable/perturbation resilient instances; it shows (α,ε) if there is an algorithm for (α,ε)-perturbation resilient instances. "s.c.b." is a shortcut for a clustering problem with a separable center-based objective.

If the optimal solution changes drastically when we slightly perturb the weights, then by solving the problem exactly, we will likely not find the true clustering since we often know the values of edge weights or distances only approximately. Therefore, if the instance is not stable, we are not interested in solving it in the first place.

Nevertheless, the definition of Bilu–Linial stability is somewhat too strict. Perhaps, it is more natural to require that the optimal solution to a perturbed instance be "ε-close" but not necessarily equal to the optimal solution for the original instance. This notion is captured in the definitions of *α-weak Bilu–Linial stability* and *(α,ε)-perturbation resilience* (we present a formal definition of weak Bilu–Linial stability for Max Cut in Section 1.2.3).

Let us now briefly describe the research on Bilu–Linial stability. We refer the reader to Table 1.1 for the list of known results. The notion of instance stability was introduced by Bilu and Linial (2010). They offered the first evidence that stable instances are much easier than worst-case instances; specifically, they gave an exact algorithm for $O(n)$-stable instances of Max Cut. This result was improved by Bilu et al. (2013), who designed an algorithm for $O(\sqrt{n})$-stable instances. Makarychev et al. (2014b) developed

a general approach to analyzing stable instances of graph partitioning problems, showing that if there exist a convex relaxation and a rounding scheme for a problem satisfying certain properties, then

- the convex relaxation for stable instances of the problem is integral;
- there are polynomial-time algorithms for stable and weakly stable instances of the problem;
- the algorithm for stable instances is robust — it either solves the problem or certifies that the instance is not stable.

In particular, this result applies to $O(\sqrt{\log n}\log\log n)$-stable and weakly stable instances of Max Cut, and 4-stable and weakly stable instances of Minimum Multiway Cut. Moreover, the results for Max Cut are essentially tight; see (Makarychev et al., 2014b) for details.

Awasthi et al. (2012) initiated the study of perturbation resilience of clustering problems. They defined a wide class of clustering problems with separable center-based objectives, including such problems as k-center, k-means, and k-median, and presented an algorithm for solving 3-perturbation resilient instances of such problems. Additionally, in a more general setting, where Steiner points are allowed, they gave an algorithm for $(2+\sqrt{3})$-perturbation resilient instances, and showed that there is no polynomial-time algorithm for 3-perturbation resilient instances with Steiner points.

Later, Balcan and Liang (2016) improved the result of Awasthi et al. (2012) for clustering problems with separable center-based objectives (without Steiner points), by showing that $(1+\sqrt{2})$-perturbation resilient instances can be efficiently solved. In addition, they gave an approximation algorithm for $(2+\sqrt{3},\varepsilon)$-perturbation resilient (weakly stable) instances. They also presented an algorithm for clustering with the min-sum objective, as well as sub-linear algorithms for clustering problems.

Most recently, Balcan et al. (2015) designed algorithms for 2-perturbation resilient instances of symmetric and asymmetric k-center and obtained a matching hardness result. They also considered clustering instances with separable center-based objectives satisfying the cluster verifiability condition. This condition requires that there be a polynomial-time algorithm that, given a set S, determines which of the following statements holds true:

(1) $S = C_i$ for some i, (2) $S \subset C_i$ for some i, (3) $S \supset C_i$ for some i

(where $C_1, \dots, C_k$ is the optimal clustering); under the promise that one of these statements is true. Balcan et al. (2015) showed how to solve 2-stable instances satisfying this condition.

There has also been research on algorithms for stable instances of other

problems. Mihalák et al. (2011) gave an algorithm for 1.8-stable instances of the Travelling Salesperson Problem (TSP). Balcan and Braverman (2010) studied the problem of finding the Nash equilibrium under stability assumptions. Also of much interest are the papers by Ostrovsky et al. (2006) and Balcan et al. (2009), which study notions of stability closely related to Bilu–Linial stability. Finally, let us mention that Leontev gave a similar definition of stability for combinatorial optimization problems in 1975. However, his motivation for studying instance stability was different from the motivation of Bilu and Linial; and the questions studied in his paper (Leontev, 1975) and a number of subsequent papers are not related to the questions addressed in this survey.

1.1.1 Organization

We describe several results for stable instances of graph partitioning and clustering problems. We begin with a general definition of graph partitioning problems in Section 1.2.1. Then, we prove that convex relaxations for γ-stable instances of graph partitioning problems, which satisfy certain assumptions, are integral (for the appropriate choice of γ), and, therefore, these instances can be solved in polynomial time. In Section 1.2.2, we apply this theorem to the Minimum Multiway Cut problem to show that 4-stable instances of the problem have an integral LP relaxation. In Section 1.2.1, we also state a general theorem for *weakly* stable instances of graph partitioning problems (Theorem 1.1, part II). However, we omit the proof in this survey. Instead, in Section 1.2.3, we prove a special case of the theorem, presenting an algorithm for γ-weakly stable instances of Max Cut (for $\gamma \geq c\sqrt{\log n}\log\log n$).

Then we proceed to clustering problems. In Section 1.3.1, we give an algorithm for 2-metric perturbation resilient instances of k-center (due to Balcan et al., 2015). Then, in Section 1.3.2, we give the definition of clustering problems with a center-based objective and present an algorithm for solving $(\sqrt{2}+1)$-metric perturbation resilient instances of such problems (due to Balcan and Liang, 2016).

1.2 Stable instances of graph partitioning problems

1.2.1 Relaxations for stable instances are integral

In this section, we study stable instances of graph partitioning problems. We show that under certain conditions convex relaxations (e.g., linear pro-

gramming and semidefinite programming relaxations) for stable instances of graph partitioning problems are integral. In particular, the result of this section implies that 4-stable instances of Minimum Multiway Cut and $c\sqrt{\log n \log\log n}$-stable instances of Max Cut have integral convex relaxations.

The result applies to a wide class of graph partitioning problems. Let us start with defining graph partitioning problems — our definition will include such problems as Min Cut, Max Cut, Minimum Multiway Cut, Minimum Balanced Cut, Minimum Multicut, and many others.

Definition 1.4. *In a graph partitioning problem, we are given a graph $G = (V, E, w)$ with positive edge weights $w(e)$. Our goal is to remove a subset of edges $E_{cut} \subset E$ that satisfies certain conditions, which depend on the specific problem at hand, so as to minimize or maximize the weight of cut edges. Specifically, in a minimization problem, we minimize $\sum_{e\in E_{cut}} w(e)$; in a maximization problem, we maximize $\sum_{e\in E_{cut}} w(e)$.*

Consider a few examples that show how our definition captures standard graph partitioning problems; for each problem, we will state the requirements on the set E_{cut}. The global Min Cut problem is a minimization problem, in which we require that the set of edges E_{cut} consist exactly of all the edges between some set A and its complement $\bar{A}$ (both sets A and $\bar{A}$ must not be empty). Max Cut is a maximization problem, in which we similarly require that E_{cut} consist of all the edges between sets A and $\bar{A}$. Minimum Multiway Cut is a minimization problem, in which we require that every two terminals s_i and s_j in a given set of terminals $\{s_1, \ldots, s_k\}$ be disconnected in $G - E_{cut}$.

We show an interesting connection between Bilu–Linial stability and *rounding algorithms or schemes* for convex relaxations of graph partitioning problems. First, let us briefly discuss how rounding schemes are used in solving graph partitioning problems. We write a linear programming (LP) or semidefinite programming (SDP) relaxation for the problem. The relaxation has two types of feasible solutions. First of all, the relaxation has feasible *integral* solutions, which are in one-to-one correspondence with feasible solutions to the graph partitioning problem (we will refer to solutions of the graph partitioning problem as combinatorial solutions). Secondly, the relaxation has solutions that do not correspond to any combinatorial solutions. We solve the relaxation and find an optimal *fractional* solution, which might not be integral. However, since there is an integral solution corresponding to the optimal combinatorial solution, the optimal fractional solution value must be at least the optimal combinatorial value for a maximization problem and at most the optimal combinatorial value for a minimization problem.

Now we use a (randomized) *rounding scheme* to transform a fractional solution to a combinatorial solution.[1] Most linear and semidefinite programming relaxations for graph partitioning problems are metric-based. Let us give a very general definition of a metric-based fractional solution.

Definition 1.5. *We say that x is a metric-based fractional solution of value* $\mathrm{val}(x)$ *for a graph partitioning problem if there is a polynomial-time algorithm that, given x, finds a distance function $d : E \to [0,1]$ such that*

$$\mathrm{val}(x) = \sum_{(u,v)\in E} w(u,v)\, d(u,v).$$

We say that distance d is defined by solution x.

Assume that there is a polynomial-time (optimization) algorithm $\mathcal{A}$ that, given an instance of the problem, finds a metric-based fractional solution x of value $\mathrm{val}(x)$*,*

$$\mathrm{val}(x) \geq \mathsf{OPT} \qquad \text{for a maximization problem,}$$
$$\mathrm{val}(x) \leq \mathsf{OPT} \qquad \text{for a minimization problem,}$$

where OPT *is the value of the optimal combinatorial solution. Then we say that x is an optimal fractional solution found by the optimization algorithm $\mathcal{A}$.*

A standard example of an algorithm $\mathcal{A}$ is an LP or SDP solver that finds an optimal solution to an LP or SDP relaxation of a graph partitioning problem. Then an optimal fractional solution x is just an optimal LP or SDP solution to the relaxation.

Definition 1.6. *Consider a graph partitioning problem and an optimization algorithm $\mathcal{A}$ as in Definition 1.5. We say that a randomized algorithm $\mathcal{R}$ is a rounding scheme (w.r.t. $\mathcal{A}$) if, given an optimal fractional solution x for an instance of the problem, it returns a feasible solution to the instance.*

Now note that, by combining an optimization procedure $\mathcal{A}$ and (polynomial-time) rounding scheme $\mathcal{R}$, we get a randomized approximation algorithm (see Algorithm 1.1). The mere existence of a rounding scheme, however, does not guarantee that the approximation algorithm based on it performs well. Let us say that we have a minimization problem. One of the most common ways to ensure that the approximation algorithm has an approximation factor of α is to use a rounding scheme $\mathcal{R}$ satisfying the following condition:

1. We note that "rounding algorithms" are often very non-trivial; they do not merely round real numbers to integers as their name might suggest.

Algorithm 1.1 Approximation algorithm based on optimization procedure $\mathcal{A}$ and rounding scheme $\mathcal{R}$

1: Run $\mathcal{A}$ on the input instance $\mathcal{I}$ and get an optimal fractional solution x.
2: Run $\mathcal{R}$ on x and get a feasible solution to $\mathcal{I}$.

given an optimal fractional solution x, $\mathcal{R}$ returns a random solution E'_{cut} such that

$$\Pr\left((u,v) \in E'_{cut}\right) \leq \alpha d(u,v), \tag{1.1}$$

where d is the distance defined by x. Observe that, then, the expected cost of the solution E'_{cut} is

$$\begin{aligned}\mathbb{E}\left[w(E'_{cut})\right] &= \sum_{(u,v)\in E} w(u,v)\Pr\left((u,v) \in E'_{cut}\right) \\ &\leq \alpha \sum_{(u,v)\in E} w(u,v)\, d(u,v) = \alpha\,\mathrm{val}(x) \leq \alpha\mathsf{OPT}.\end{aligned}$$

That is, in expectation, the algorithm finds a solution of cost at most $\alpha\mathsf{OPT}$, and thus has an approximation factor of α. Now consider the complementary optimization problem of *maximizing* the weight of uncut edges, $w(E \setminus E'_{cut})$. Note that an optimal solution to the original problem is also an optimal solution to the complementary problem, since the sum of their objectives, $w(E'_{cut}) + w(E \setminus E'_{cut}) = w(E)$, depends only on the instance and not on the solution E'_{cut}. However, the problems might be very different in terms of multiplicative approximability — a good approximation algorithm for one of them is not necessarily good for the other. It is not hard to see that in order to get a β approximation algorithm for the complementary problem, we can use a rounding procedure $\mathcal{R}$ satisfying the following condition,

$$\Pr\left((u,v) \notin E'_{cut}\right) \geq \beta^{-1}(1 - d(u,v)). \tag{1.2}$$

We stress that conditions (1.1) and (1.2) are completely independent, and a rounding procedure may satisfy one of them and not the other.

Makarychev et al. (2014b) showed that if there is a rounding scheme $\mathcal{R}$ satisfying both conditions (1.1) and (1.2), then the relaxation for $(\alpha\beta)$-stable instances is integral, and, consequently, there is a robust exact algorithm for $(\alpha\beta)$-stable instances.

Theorem 1.1 (Makarychev et al. (2014b))**.** *I. Consider a graph partitioning problem. Suppose that there is a rounding scheme that, given a graph $G = (V, E, w)$ and an optimal fractional solution x, returns a feasible solution E'_{cut} such that for some $\alpha \geq 1$ and $\beta \geq 1$ (α and β may depend on n),*

For a cut minimization problem,

1. $\Pr((u,v) \in E'_{cut}) \le \alpha d(u,v)$,
2. $\Pr(u \notin E'_{cut}) \ge \beta^{-1}(1 - d(u,v))$.

For a cut maximization problem,

1′. $\Pr((u,v) \in E'_{cut}) \ge \alpha^{-1} d(u,v)$

2′. $\Pr((u,v) \notin E'_{cut}) \le \beta(1 - d(u,v))$

where distance d is defined by the fractional solution x.

Then distance d is integral for $(\alpha\beta)$-stable instances of the problem; specifically, for every edge $(u,v) \in E$

$$d(u,v) = \begin{cases} 0, & \text{if } (u,v) \notin E^*_{cut}, \\ 1, & \text{if } (u,v) \in E^*_{cut}, \end{cases}$$

where E^*_{cut} is the optimal combinatorial solution.[2] *Consequently, there is a robust polynomial-time algorithm for $(\alpha\beta)$-stable instances.*

II. Furthermore, there is an algorithm for $(\alpha\beta + \varepsilon, N)$-weakly stable instances of the problem that finds a feasible solution $E'_{cut} \in N$ (for every $\varepsilon > 0$).

The theorem also holds for graph partitioning problems with positive and negative weights if we require that all four properties 1, 1′, 2 and 2′ hold.

In this survey, we are going to prove only part I of Theorem 1.1. Since the proofs of Theorem 1.1 for minimization and maximization problems are completely analogous, let us only consider a minimization problem. Before we proceed with the proof itself, we prove the following auxiliary lemmas.

Lemma 1.2 (Bilu and Linial (2010))**.** *Consider a γ-stable instance of a minimization graph partitioning problem. Suppose E^*_{cut} is the optimal combinatorial solution. Then, for any combinatorial solution E'_{cut}, we have*

$$\gamma\, w(E^*_{cut} \setminus E'_{cut}) < w(E'_{cut} \setminus E^*_{cut}).$$

Proof. Consider the following γ-perturbation of w: $w'(u,v) = \gamma w(u,v)$ for $(u,v) \in E^*_{cut} \setminus E'_{cut}$; and $w'(u,v) = w(u,v)$ otherwise. Since the instance is is γ-stable, we have $w'(E^*_{cut}) < w'(E'_{cut})$. Write,

$$\underbrace{w'(E^*_{cut} \setminus E'_{cut}) + w'(E^*_{cut} \cap E'_{cut})}_{w'(E^*_{cut})} < \underbrace{w'(E'_{cut} \setminus E^*_{cut}) + w'(E^*_{cut} \cap E'_{cut})}_{w'(E'_{cut})}.$$

Thus, $w'(E^*_{cut} \setminus E'_{cut}) < w'(E'_{cut} \setminus E^*_{cut})$. Using the definition of w', we get

2. In particular, given d, we can find E^*_{cut}: $E^*_{cut} = \{(u,v) : d(u,v) = 1\}$.

the desired inequality: $\gamma\, w(E^*_{cut} \setminus E'_{cut}) < w(E'_{cut} \setminus E^*_{cut})$. □

Lemma 1.3. *If the distance d defined by a fractional solution x is not integral, then the rounding algorithm returns a solution E'_{cut} different from the optimal combinatorial solution E^*_{cut} with non-zero probability.*

Proof. Note that if $d(u,v) < 1$ for some edge $(u,v) \in E^*_{cut}$, then $(u,v) \notin E'_{cut}$ with probability at least $\beta^{-1}(1 - d(u,v)) > 0$, and hence $E^*_{cut} \neq E'_{cut}$ with non-zero probability. So let us assume that $d(u,v) = 1$ for every $(u,v) \in E^*_{cut}$. Since the cost of the optimal combinatorial solution is at least the cost of the optimal fractional solution x, we have

$$\begin{aligned}\sum_{(u,v)\in E^*_{cut}} w(u,v) \geq \operatorname{val}(x) &= \sum_{(u,v)\in E} w(u,v)\, d(u,v) \\ &= \sum_{(u,v)\in E^*_{cut}} w(u,v) + \sum_{(u,v)\in E\setminus E^*_{cut}} w(u,v)\, d(u,v).\end{aligned}$$

Therefore,

$$\sum_{(u,v)\in E\setminus E^*_{cut}} w(u,v)\, d(u,v) \leq 0,$$

and $d(u,v) = 0$ for every $(u,v) \in E \setminus E^*_{cut}$. □

Proof of Theorem 1.1. Consider an $(\alpha\beta)$-stable instance of the problem. Let d be the distance defined by an optimal solution. We are going to prove that d is integral. Assume to the contrary that it is not. Let E'_{cut} be a random combinatorial solution obtained by rounding d, and let E^*_{cut} be the optimal combinatorial solution. Since d is not integral, $E'_{cut} \neq E^*_{cut}$ with non-zero probability.

From $(\alpha\beta)$-stability of the instance (see Lemma 1.2), we get that

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

and therefore (here we use that $\Pr(E^*_{cut} \neq E'_{cut}) > 0$),

$$(\alpha\beta)\mathbb{E}\left[w(E^*_{cut} \setminus E'_{cut})\right] < \mathbb{E}\left[w(E'_{cut} \setminus E^*_{cut})\right]. \tag{1.3}$$

Let

$$\begin{aligned}\mathsf{LP}_+ &= \sum_{(u,v)\in E^*_{cut}} w(u,v)(1 - d(u,v)), \\ \mathsf{LP}_- &= \sum_{(u,v)\in E\setminus E^*_{cut}} w(u,v)\, d(u,v).\end{aligned}$$

From conditions 1 and 2 in the statement of the theorem, we get

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

Using inequality (1.3), we conclude that $\mathsf{LP}_+ < \mathsf{LP}_-$. On the other hand, from the formulas for LP_+ and LP_-, we get

$$\mathsf{LP}_+ - \mathsf{LP}_- = w(E^*_{cut}) - \sum_{(u,v)\in E} w(u,v)\, d(u,v) \geq 0,$$

since the value of the fractional solution is at most the value of the integral solution. We get a contradiction, which concludes the proof. □

1.2.2 An LP relaxation and rounding scheme for Minimum Multiway Cut

In this section, we show that the linear programming relaxation for 4-stable instances of Minimum Multiway Cut is integral. To this end, we present an LP relaxation for Minimum Multiway Cut and a rounding scheme satisfying the conditions of Theorem 1.1. Recall the definition of the Multiway Cut problem.

Definition 1.7. *An instance of Minimum Multiway Cut consists of a graph $G = (V, E, w)$ with positive edge weights w_e and a set of terminals $T = \{s_1, \ldots, s_k\} \subset V$. The goal is to partition the graph into k pieces $S_1, \ldots, S_k$ with $s_i \in S_i$ so as to minimize the total weight of cut edges*

$$E_{cut} = \{(u,v) \in E : u \in S_i, v \in S_j \text{ for } i \neq j\}.$$

The problem has been actively studied since it was introduced by Dahlhaus et al. (1994). There has been a series of approximation algorithms for it (Călinescu et al., 1998; Karger et al., 2004; Buchbinder et al., 2013); the current state-of-the-art approximation algorithm by Sharma and Vondrák (2014) gives a 1.30217 approximation.

We use the LP relaxation of Călinescu et al. (1998). In this relaxation, we have a variable $\bar{u} = (\bar{u}_1, \ldots, \bar{u}_k) \in \mathbb{R}^k$ for every vertex $u \in V$. Let $e_1, \ldots, e_k$

be the standard basis in $\mathbb{R}^k$ and $\Delta = \{x : \|x\|_1 = 1, x_1 \geq 0, \ldots, x_k \geq 0\}$ be the simplex with vertices $e_1, \ldots, e_k$.

$$\text{minimize } \frac{1}{2} \sum_{(u,v)\in E} w(u,v)\, \|\bar{u} - \bar{v}\|_1 \tag{1.4}$$

subject to:

$$\begin{aligned} &\bar{s}_i = e_i && \text{for every } i, \\ &\bar{u} \in \Delta && \text{for every } u \in V. \end{aligned}$$

Every feasible LP solution defines a metric on V: $d(u,v) = \|\bar{u} - \bar{v}\|_1/2$. Note that the objective function equals $\sum_{e\in E} w(u,v)\, d(u,v)$. Let us now present a randomized rounding scheme for this LP relaxation.

Theorem 1.4 (Makarychev et al. (2014b))**.** *Consider a feasible LP solution $\{\bar{u} : u \in V\}$ and metric $d(u,v) = \|\bar{u} - \bar{v}\|_1/2$. There is a randomized algorithm that finds a partition $S_1, \ldots, S_k$ of V and a set E_{cut} such that*

- *$s_i \in S_i$ for every $i \in \{1, \ldots, k\}$ (always),*
- *$\Pr((u,v) \in E_{cut}) \leq \frac{2d(u,v)}{1+d(u,v)}$ for every $(u,v) \in E$. In particular,*

$$\Pr((u,v) \in E_{cut}) \leq 2d(u,v) \quad \text{and} \quad \Pr((u,v) \notin E_{cut}) \geq \frac{1 - d(u,v)}{2}.$$

The rounding procedure satisfies the conditions of Theorem 1.1 with parameters $\alpha = \beta = 2$, and, therefore, the LP relaxation for 4-stable instances of Multiway Cut is integral.

Proof. We use the rounding algorithm by Kleinberg and Tardos (2002). The algorithm starts with empty sets $S_1, \ldots, S_k$ and then iteratively adds vertices to sets $S_1, \ldots, S_k$. It stops when each vertex is assigned to some set S_i. In each iteration, the algorithm chooses independently and uniformly at random $r \in (0,1)$ and $i \in \{1, \ldots, k\}$. It adds each vertex u to S_i if $r \leq \bar{u}_i$ and u has not yet been added to any set S_j.

Algorithm 1.2 Rounding Algorithm for Minimum Multiway Cut

$S_1 = \varnothing, \ldots, S_k = \varnothing$
$R = V$ ▷ R is the set of unpartitioned vertices
while $R \neq \varnothing$ **do**
  $r \in_U (0,1)$; $i \in_U \{1, \ldots, k\}$
  $S_i = S_i \cup \{u \in R : \bar{u}_i \geq r\}$
  $R = R \setminus \{u \in R : \bar{u}_i \geq r\}$
end while
return $S_1, \ldots, S_k$ and $E_{cut} = \{(u,v) \in E : u \in S_i, v \in S_j \text{ for } i \neq j\}$.

First, note that we add every vertex u to some S_i with probability $\sum_{i=1}^{k} \bar{u}_i/k = 1/k$ in each iteration (unless u already lies in some S_j). So eventually we will add every vertex to some set S_i. Also note that we cannot add s_i to S_j if $j \neq i$. Therefore, $s_i \in S_i$.

Now consider an edge (u, v). Consider one iteration of the algorithm. Suppose that neither u nor v is assigned to any set S_j in the beginning of the iteration. The probability that at least one of them is assigned to some S_i in this iteration is

$$\frac{1}{k}\sum_{i=1}^{k} \Pr(\bar{u}_i \geq r \text{ or } \bar{v}_i \geq r) = \frac{1}{k}\sum_{i=1}^{k} \max(\bar{u}_i, \bar{v}_i)$$
$$= \frac{1}{k}\sum_{i=1}^{k}\left(\frac{\bar{u}_i + \bar{v}_i}{2} + \frac{|\bar{u}_i - \bar{v}_i|}{2}\right) = \frac{1}{k}\left(1 + \frac{\|\bar{u} - \bar{v}\|_1}{2}\right) = \frac{1 + d(u, v)}{k}.$$

The probability that exactly one of them is assigned to some S_i is

$$\frac{1}{k}\sum_{i=1}^{k} \Pr(\bar{u}_i < r \leq \bar{v}_i \text{ or } \bar{v}_i < r \leq \bar{u}_i) = \frac{1}{k}\sum_{i=1}^{k} |\bar{u}_i - \bar{v}_i| = \frac{\|\bar{u} - \bar{v}\|_1}{k} = \frac{2d(u, v)}{k}.$$

We get that in one iteration, the conditional probability that u and v are separated given that at least one of them is assigned to some set is $2d(u, v)/(1 + d(u, v))$. Therefore, the probability that u and v are separated in some iteration is $2d(u, v)/(1 + d(u, v))$. Thus the probability that (u, v) is cut is at most $2d(u, v)/(1 + d(u, v))$. □

1.2.3 Weakly Stable Instances of Max Cut

Bilu–Linial stability imposes rather strong constraints on an instance of a graph partitioning problem. Can these constraints be relaxed? In this section, we give a definition of a more robust notion — a notion of weak stability. Then we present an algorithm for weakly stable instances of the Max Cut problem. Note that using Theorem 1.1 from the previous section, one can show that a certain SDP relaxation for Max Cut is integral for γ-stable instances of Max Cut with $\gamma \geq c\sqrt{\log n \log\log n}$. However, the SDP does not have to be integral for weakly stable instances of Max Cut. Let us now recall the definition of Max Cut.

Definition 1.8 (Max Cut)**.** *In the Max Cut Problem, we are given a weighted graph $G = (V, E, w)$. Our goal is to partition the set of vertices into two sets S and $\bar{S}$ so as to maximize $w(E(S, \bar{S}))$.*

Max Cut is an NP-hard problem (Karp, 1972). The approximation factor

of the best known algorithm due to Goemans and Williamson (1995) is 0.878. It cannot be improved if the Unique Games Conjecture holds true (Khot et al., 2007). We now give the definition of weak stability for Max Cut.

Definition 1.9. *Consider a weighted graph $G = (V, E, w)$. Let $(S, \bar{S})$ be a maximum cut in G, N be a set of cuts that contains $(S, \bar{S})$, and $\gamma \geq 1$. We say that G is a (γ, N)-weakly stable instance of Max Cut if for every γ-perturbation $G' = (V, E, w')$ of G, and every cut $(T, \bar{T}) \notin N$, we have*

$$w'(E(S, \bar{S})) > w'(E(T, \bar{T})).$$

The notion of weak stability generalizes the notion of stability: an instance is γ-stable if and only if it is $(\gamma, \{(S, \bar{S})\})$-weakly stable. We think of the set N in the definition of weak stability as a neighborhood of the maximum cut $(S, \bar{S})$; it contains cuts that are "close enough" to $(S, \bar{S})$. Intuitively, the definition requires that every cut that is sufficiently different from $(S, \bar{S})$ be much smaller than $(S, \bar{S})$, but does not impose any restrictions on cuts that are close to $(S, \bar{S})$. One natural way to define the neighborhood of $(S, \bar{S})$ is captured in the following definition.

Definition 1.10. *Consider a weighted graph G. Let $(S, \bar{S})$ be a maximum cut in G, $\delta \geq 0$, and $\gamma \geq 1$. We say that G is a (γ, δ)-weakly stable instance of Max Cut if G is $(\gamma, \{(S', \bar{S}') : |S \Delta S'| \leq \delta n\})$-weakly stable. In other words, G is (γ, δ)-weakly stable if for every cut $(T, \bar{T})$ such that $|S \Delta T| > \delta n$ and $|S \Delta \bar{T}| > \delta n$, we have $w'(E(S, \bar{S})) > w'(E(T, \bar{T}))$.*

We prove the following analog of Lemma 1.2.

Lemma 1.5. *Consider a (γ, N)-weakly stable instance of Max Cut $G = (V, E, w)$. Let $(S, \bar{S})$ be a maximum cut in G. Then, for every cut $(T, \bar{T}) \notin N$:*

$$w(E(S, \bar{S}) \setminus E(T, \bar{T})) > \gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S})). \tag{1.5}$$

Proof. Fix a cut $(T, \bar{T}) \notin N$. Consider the following γ-perturbation of w: $w'(u, v) = \gamma w(u, v)$ for $(u, v) \in E(T, \bar{T}) \setminus E(S, \bar{S})$; and $w'(u, v) = w(u, v)$ otherwise. Since G is a γ-weakly stable instance, and $(T, \bar{T}) \notin N$, we have

$$w'(E(S, \bar{S})) > w'(E(T, \bar{T})).$$

Write,

$$\begin{aligned} w'(E(S, \bar{S})) &= w'(E(S, \bar{S}) \setminus E(T, \bar{T})) + w'(E(S, \bar{S}) \cap E(T, \bar{T})); \\ w'(E(T, \bar{T})) &= w'(E(T, \bar{T}) \setminus E(S, \bar{S})) + w'(E(S, \bar{S}) \cap E(T, \bar{T})). \end{aligned}$$

Thus, $w'(E(S, \bar{S}) \setminus E(T, \bar{T})) > w'(E(T, \bar{T}) \setminus E(S, \bar{S}))$. Using the definition of w', we get inequality (1.5). □

We are now ready to state the main result.

Theorem 1.6 (Makarychev et al. (2014b)). *There is a polynomial-time algorithm that, given a (γ, N)-stable instance of Max Cut, returns a cut from N if $\gamma \geq c\sqrt{\log n}\log\log n$ (for some absolute constant c). The set N is not part of the input and is not known to the algorithm.*

Overview of the algorithm. The algorithm starts with an arbitrary cut $(S_0, \bar{S}_0)$ and then iteratively improves it: first, it finds a cut $(S_1, \bar{S}_1)$ that is better than $(S_0, \bar{S}_0)$, then a cut $(S_2, \bar{S}_2)$ that is better than $(S_1, \bar{S}_1)$, etc.

$$(S_0, \bar{S}_0) \to (S_1, \bar{S}_1) \to (S_2, \bar{S}_2) \to \cdots \to (S_t, \bar{S}_t);$$

finally, it gets a cut $(S_t, \bar{S}_t)$ that it cannot improve. This cut necessarily belongs to the set N, and the algorithm outputs it. The key component of the algorithm is a procedure Improve that, given a cut $(S_i, \bar{S}_i) \notin N$, finds a better cut $(S_{i+1}, \bar{S}_{i+1})$ (if $(S_i, \bar{S}_i) \in N$, the procedure may either find an improved cut or output that $(S_i, \bar{S}_i) \in N$).

Now, we are going to present Improve. We note that we also must show that the improvement process finishes in polynomially many steps, and, thus, the running time is polynomial. In this survey, we assume for simplicity that all edge weights are polynomially bounded integers. Then the weight of every cut is a polynomially bounded integer; therefore, the weight of the cut increases by at least 1 in each iteration, and the algorithm terminates after polynomially many iterations. In the paper (Makarychev et al., 2014b), the theorem is proved without this simplifying assumption.

Before we describe the procedure Improve, we recall the definition of Sparsest Cut with non-uniform demands.

Definition 1.11 (Sparsest Cut with non-uniform demands). *We are given a graph $H = (V, E_c, \mathrm{cap})$ with non-negative edge capacities $\mathrm{cap}(u, v)$, a set of demand pairs E_d, and non-negative demands $\mathrm{dem} : E_d \to \mathbb{R}_{\geq 0}$. Our goal is to find a cut $(A, \bar{A})$ so as to minimize the ratio between the capacity of the cut edges and the amount of separated demands*

$$\text{minimize} \quad \sum_{\substack{(u,v)\in E_c \\ u\in A,\, v\in \bar{A}}} \mathrm{cap}(u,v) \Bigg/ \sum_{\substack{(u,v)\in E_d \\ u\in A,\, v\in \bar{A}}} \mathrm{dem}(u,v).$$

We call this ratio the sparsity of the cut $(A, \bar{A})$.

We use the approximation algorithm for Sparsest Cut by Arora et al. (2008) that gives a $(C_{\mathrm{sc}}\sqrt{\log n}\log\log n)$-approximation (where C_{sc} is an absolute constant).

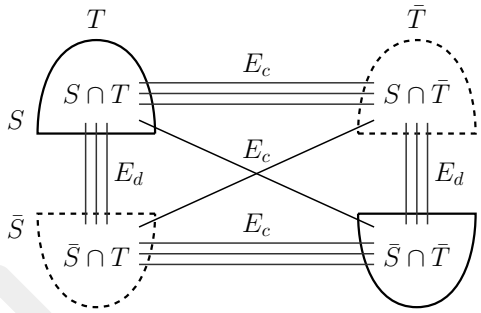

Figure 1.1: The figure shows sets S, $\bar{S}$, T, $\bar{T}$, and their pairwise intersections. Set E_c consists of horizontal and diagonal edges; set E_d consists of vertical edges, as well as edges within $S \cap T$, $S \cap \bar{T}$, $\bar{S} \cap T$, $\bar{S} \cap \bar{T}$; set $E(A^*, \bar{A}^*)$ consists of horizontal and vertical edges.

Theorem 1.7. *Let $\gamma = C_{\mathrm{sc}}\sqrt{\log n}\log\log n$. There is a polynomial-time algorithm* Improve *that, given a (γ, N)-weakly stable instance of Max Cut and a cut $(T, \bar{T}) \notin N$, finds a cut $(T', \bar{T}')$ of greater value,*

$$w(E(T', \bar{T}')) > w(E(T, \bar{T})).$$

Proof. Define an auxiliary Sparsest Cut instance $G_{aux} = (V, E_c, \mathrm{cap})$ on V:

$$E_c = E(T, \bar{T}) \qquad \mathrm{cap}(u, v) = w(u, v)$$
$$E_d = E \setminus E(T, \bar{T}) \qquad \mathrm{dem}(u, v) = w(u, v).$$

Now run the approximation algorithm for Sparsest Cut by Arora et al. (2008) and find an approximate cut $(A, \bar{A})$. Let $T' = (T \cap A) \cup (\bar{T} \cap \bar{A})$. If $w(T', \bar{T}') > w(T, \bar{T})$, return the cut $(T', \bar{T}')$; otherwise, output that $(T, \bar{T}) \in N$.

We need to show that if $(T, \bar{T}) \notin N$ then $w(T', \bar{T}') > w(T, \bar{T})$. Let $(S, \bar{S})$ be the maximum cut. First, we prove that there is a sparsest cut with sparsity at most $1/\gamma$ in the auxiliary graph. Let $A^* = (S \cap T) \cup (\bar{S} \cap \bar{T})$. Since $(T, \bar{T}) \notin N$, we have by Lemma 1.5:

$$w(E(S, \bar{S}) \setminus E(T, \bar{T})) > \gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S})).$$

Note that $E(A^*, \bar{A}^*) = E(S \cap T, S \cap \bar{T}) \cup E(S \cap T, \bar{S} \cap T) \cup E(\bar{S} \cap T, \bar{S} \cap \bar{T}) \cup E(S \cap \bar{T}, \bar{S} \cap \bar{T})$ (see Figure 1.1), and

$$E(S, \bar{S}) \setminus E(T, \bar{T}) = E_d \cap E(A^*, \bar{A}^*)$$
$$E(T, \bar{T}) \setminus E(S, \bar{S}) = E_c \cap E(A^*, \bar{A}^*).$$

The sparsity of the cut $(A^*, \bar{A}^*)$ is therefore at most

$$\frac{\mathrm{cap}(E_c \cap E(A^*, \bar{A}^*))}{\mathrm{dem}(E_d \cap E(A^*, \bar{A}^*))} = \frac{w(E(T, \bar{T}) \setminus E(S, \bar{S}))}{w(E(S, \bar{S}) \setminus E(T, \bar{T}))} < \frac{1}{\gamma}.$$

Hence, the sparsity of the cut $(A, \bar{A})$ returned by the approximation algorithm is less than $(C_{\mathrm{sc}}\sqrt{\log n}\log\log n) \times (1/\gamma) \leq 1$. That is, $\mathrm{dem}(E_d \cap E(A, \bar{A})) > \mathrm{cap}(E_c \cap E(A, \bar{A}))$. We get

$$\begin{aligned} w(E(T', \bar{T}') \setminus E(T, \bar{T})) &= \mathrm{dem}(E_d \cap E(A, \bar{A})) > \\ &> \mathrm{cap}(E_c \cap E(A, \bar{A})) = w(E(T, \bar{T}) \setminus E(T', \bar{T}')), \end{aligned}$$

and, consequently,

$$\begin{aligned} w(T', \bar{T}') &= w(E(T', \bar{T}') \setminus E(T, \bar{T})) + w(E(T', \bar{T}') \cap E(T, \bar{T})) > \\ &> w(E(T, \bar{T}) \setminus E(T', \bar{T}')) + w(E(T', \bar{T}') \cap E(T, \bar{T})) = w(T, \bar{T}). \end{aligned}$$

Thus, the weight of the cut $(T', \bar{T}')$ obtained by the improvement algorithm Improve is greater than the weight of the cut $(T, \bar{T})$. This finishes the proof. □

1.3 Stable Instances of Clustering Problems

1.3.1 Metric perturbation resilient instances of k-Center

In this section, we present an algorithm by Balcan et al. (2015) that solves 2-metric perturbation resilient instances of k-center. In fact, we prove that any α-approximation algorithm for k-center finds the optimal solution of an α-metric perturbation resilient instance of k-center. Therefore, we can use known 2-approximation algorithms for k-center to solve 2-metric perturbation resilient instances of the problem (see Hochbaum and Shmoys (1985), and Dyer and Frieze (1985)). Recall the definition of the k-center problem.

Definition 1.12. *Consider a set of vertices V, a metric d on V, and a parameter k. Given a set of points ("centers") $c_1, \ldots, c_k$ in V, define a clustering $C_1, \ldots, C_k$ by assigning each vertex u to the closest center among $c_1, \ldots, c_k$:*

$$C_i = \{u : d(u, c_i) \leq d(u, c_j) \text{ for every } i \neq j\}$$

(we break the ties arbitrarily). We say that c_i is the center of cluster C_i. The cost of the clustering is the maximum distance between a point and the

center of the cluster it belongs to.

$$\text{cost} = \max_{i\in\{1,\ldots,k\}} \max_{u\in C_i} d(u, c_i).$$

In the k-center problem, our goal is to find a clustering of minimum cost given V, d, and k.

Note that given a set of centers we can efficiently find the corresponding clustering, and given a clustering we can efficiently find an optimal set of centers for it. In this section, however, it will be more convenient for us to view a solution for k-center as a clustering rather than a set of centers. The reason for that is that in the definition of the perturbation resilience, we do not want to require that the set of centers not change when we perturb the distances — that would be a very strong requirement (indeed, it might not be even satisfied by instances with $k = 1$; furthermore, there would be no 2-perturbation resilient instances). Instead, we require that the optimal clustering $C_1, \ldots, C_k$ not change when we perturb the distances.

Remark 1.1. *In this section, we consider perturbations d' of the metric d satisfying $d(u, v)/\gamma \le d'(u, v) \le d(u, v)$ for all u,v instead of perturbations satisfying $d(u, v) \le d'(u, v) \le \gamma\, d(u, v)$ as in Definition 1.3. We can do so as long as the clustering problem is invariant under rescaling of all distances by the same positive factor, i.e. the clustering for d is the same as the clustering for αd for every $\alpha > 0$. All clustering problems we consider in this section satisfy this property.*

Balcan et al. (2015) obtained their result for 2-perturbation resilient instances of k-center. Most recently, Makarychev and Makarychev (2016) strengthened this result, by showing that it also holds for *α-metric* perturbation resilient instances.

Theorem 1.8 (Balcan et al. (2015); see also Makarychev and Makarychev (2016))**.** *An α-approximation algorithm for k-center finds the optimal clustering of an α-metric perturbation resilient instance of k-center.*[3]

Proof. Consider the optimal clustering $C_1, \ldots, C_k$ and the clustering $C'_1, \ldots, C'_k$ found by the approximation algorithm. We are going to show that they are identical. Let r^* be the value of the clustering $C_1, \ldots, C_k$. Let $\{c'_1, \ldots, c'_k\}$ be an optimal set of centers for the clustering $C'_1, \ldots, C'_k$. Since the algorithm gives an α-approximation, $d(u, c'_i) \le \alpha r^*$ for every $u \in C'_i$.

3. Note that the algorithm finds the optimal clustering $C_1, \ldots, C_k$ but not necessarily an optimal set of centers $\{c_1, \ldots, c_k\}$; however, an optimal set of centers can be easily deduced from $C_1, \ldots, C_k$.

Define a new distance d' as follows

$$d'(u,v) = \begin{cases} d(u,v)/\alpha, & \text{if } d(u,v) \geq \alpha r^*, \\ r^*, & \text{if } d(u,v) \in [r^*, \alpha r^*], \\ d(u,v), & \text{if } d(u,v) \leq r^*. \end{cases}$$

We first prove that d' satisfies the triangle inequality. Define a function $f(x)$ as follows: $f(x) = 1/\alpha$ for $x \geq \alpha r^*$; $f(x) = r^*/x$ for $x \in [r^*, \alpha r^*]$, and $f(x) = 1$ for $x \leq r^*$. Observe, that $d'(u,v) = f(d(u,v))\, d(u,v)$; $f(x)$ is a nonincreasing function; $xf(x)$ is a nondecreasing function. Consider three points u, v, w and assume without loss of generality that $d'(u,w) \geq \max(d'(u,v), d'(v,w))$. We need to prove that $d'(u,w) \leq d'(u,v) + d'(v,w)$. Note that since $xf(x)$ is a nondecreasing function, $d(u,w) \geq \max(d(u,v), d(v,w))$ and $f(d(u,w)) \leq \min(f(d(u,v)), f(d(v,w)))$. Thus,

$$\begin{aligned} d'(u,v) + d'(v,w) &= f(d(u,v))d(u,v) + f(d(v,w))d(v,w) \geq \\ &\geq f(d(u,w))\big(d(u,v) + d(v,w)\big) \geq f(d(u,w))d(u,w) = d'(u,w). \end{aligned}$$

The last inequality follows from the triangle inequality $d(u,v) + d(v,w) \geq d(u,w)$ for the metric d.

Next, we check that $d'(u,v)$ is an α-perturbation, i.e. $d(u,v)/\alpha \leq d'(u,v) \leq d(u,v)$ (see Remark 1.1). We have, $f(x) \in [1/\alpha, 1]$, and, thus, $d'(u,v)/d(u,v) = f(d(u,v)) \in [1/\alpha, 1]$.

By the definition of α-metric perturbation resilience, $C_1, \ldots, C_k$ is the unique optimal clustering for d'. However, the optimal set of centers for d' may be different from $c_1, \ldots, c_k$. Denote it by $c_1'', \ldots, c_k''$. We prove that the cost of the clustering $C_1, \ldots, C_k$ is the same for metrics d and d'. Let

$$r(C_i) = \min_{c \in C_i} \max_{u \in C_i} d(u,c).$$

Since the cost of the clustering $C_1, \ldots, C_k$ equals r^* w.r.t. d, we have $r(C_i) = r^*$ for some i. Fix this i. By the definition of $r(C_i)$, for every $c \in C_i$ there exists $u \in C_i$ such that $d(u,c) \geq r(C_i) = r^*$. Particularly, for $c = c_i''$, there exists u such that $d(u, c_i'') \geq r^*$. Then $d'(u, c_i'') \geq r^*$ as well. Hence, the cost of the clustering $C_1, \ldots, C_k$ for the metric d' is at least r^*. (It cannot be larger than r^*, since $d'(u,v) \leq d(u,v)$ for all u and v.)

To conclude the proof, we observe that the cost of the clustering $C_1', \ldots, C_k'$ with centers $c_1', \ldots, c_k'$ also equals r^* w.r.t. the metric d'. Indeed, for $u \in C_i'$, we have $d(u, c_i') \leq \alpha r^*$, and, therefore, $d'(u, c_i') \leq r^*$. Thus, $C_1', \ldots, C_k'$ is an optimal clustering for d'. Therefore, it must be equal to the clustering $C_1, \ldots, C_k$. □

1.3.2 Clustering problems with separable center-based objectives

In this section, we present an algorithm by Balcan and Liang (2016) that solves $(\sqrt{2}+1)$-metric perturbation resilient instances of clustering problems with separable center-based objectives.[4]

Definition 1.13. *In a clustering problem, we are given a set of vertices (points) V and a distance function d on V. Our goal is to partition the vertices into clusters so as to minimize a cost function, which depends on the clustering problem.*

Following Awasthi et al. (2012), we define the notion of a clustering problem with *a center-based objective*. (We note that the definition in Awasthi et al. (2012) makes several implicit assumptions that we make explicit here.)

Definition 1.14. *Consider a clustering problem. We say that it has a center-based objective if the following three properties hold.*

1. *Given a subset $S \subset V$ and distance d_S on S, we can find the optimal center $c \in S$ for S, or, if there is more than one choice of an optimal center, a set of optimal centers* $\mathrm{center}(S, d_S)$. *(In the former case,* $\mathrm{center}(S, d_S) = \{c\}$*).*

2. *The set of centers does not change if we multiply all distances between points in S by α. That is,*

$$\mathrm{center}(S, \alpha d_S) = \mathrm{center}(S, d_S).$$

Also, the optimal clustering does not change if we multiply all distances between points in V by α.

3. *Let $C_1, \ldots, C_k$ be an optimal clustering of V (the clustering of minimum cost). For every i, let $c_i \in \mathrm{center}(C_i, d|_{C_i})$ be an optimal center for C_i (here, $d|_{C_i}$ is the restriction of d to C_i). Then each point $p \in C_i$ is closer to c_i than to any other center c_j, $d(p, c_i) < d(p, c_j)$.*

A clustering-objective is separable if we can define individual cluster scores so that the following holds.

1. *The cost of the clustering is either the maximum or sum of the cluster scores.*

2. *The score* $\mathrm{score}(S, d|_S)$ *of each cluster S depends only on S and $d|_S$, and*

4. The original result by Balcan and Liang (2016) applies to $(\sqrt{2}+1)$-perturbation resilient instances; recently, Makarychev and Makarychev (2016) showed that their algorithm also works for $(\sqrt{2}+1)$-*metric* perturbation resilient instances.

can be computed in polynomial time.

Many standard clustering problems, including k-center, k-means, and k-median, have separable center-based objectives.

We will assume below that the instance is α-metric perturbation resilient with $\alpha = 1+\sqrt{2}$. Denote the optimal clustering by $C_1, \ldots, C_k$. Fix an optimal set of centers $c_1, \ldots, c_k$ for the clustering ($c_i \in \text{center}(S, d_S)$). Define the radius of cluster C_i as $r_i = \max_{u \in C_i} d(c_i, u)$. For every point u, denote the ball of radius r around u by $B(u, r)$: $B(u, r) = \{v : d(u, v) \leq r\}$.

We start with proving some basic structural properties of the optimal clustering $C_1, \ldots, C_k$.

Lemma 1.9 (Awasthi et al. (2012); Makarychev and Makarychev (2016))**.** *Clusters satisfy the following α-center proximity property: for all $i \neq j$ and $p \in C_i$,*

$$d(p, c_j) > \alpha d(p, c_i).$$

Proof. Suppose that $d(p, c_j) \leq \alpha d(p, c_i)$. Let $r^* = d(p, c_i)$. Define a new metric d' as follows: for all u and v,

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

The metric $d'(u, v)$ is the shortest path metric on the complete graph on V with edge lengths $len(u, v) = d(u, v)$ for all edges (u, v) but the edge (p, c_j). The length of the edge (p, c_j) equals $len(p, c_j) = r^*$. 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' is an α-metric perturbation of d (see Remark 1.1).

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

Lemma 1.10. *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 $d(v, c_i) < d(v, c_j)$, and thus

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

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 $d(u, c_j) < d(u, c_i)$. Thus,

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

□

By the definition of α-metric perturbation stability, the optimal clusterings for metrics d and d' are the same. By Lemma 1.10, 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 points c_i and c_j, respectively (see Definition 1.14, item 1). Thus, $d'(c_i, p) < d'(c_j, p)$, and, consequently,

$$d(c_i, p) = d'(c_i, p) < d'(c_j, p) = r^* = d(c_i, p).$$

We get a contradiction, which finishes the proof. □

Lemma 1.11 (Awasthi et al. (2012); Balcan and Liang (2016))**.**

1. *All points outside of C_i lie at distance greater than r_i from c_i. Thus, $C_i = B(c_i, r_i)$.*

2. *Each point p in C_i is closer to c_i than to any point q outside of C_i. Furthermore, for every $p \in C_i$ and $q \notin C_i$, we have $\sqrt{2}\, d(p, c_i) < d(p, q)$.*

3. *For every two distinct clusters C_i and C_j,*

$$d(c_i, c_j) > \sqrt{2} \max(r_i, r_j).$$

Proof. We will prove items in the following order: 3, 1, and finally 2.

3. Let p be the farthest from c_i point in C_i. Then $r_i = d(c_i, p)$. By Lemma 1.9, $d(p, c_j) > \alpha d(p, c_i) = \alpha r_i$. By the triangle inequality,

$$d(c_i, c_j) \geq d(p, c_j) - d(p, c_i) > \alpha r_i - r_i = \sqrt{2} r_i.$$

Similarly, $d(c_i, c_j) > \sqrt{2} r_j$.

1. Consider a point $q \notin C_i$. Assume that $q \in C_j$. Then

$$d(c_i, c_j) \leq d(c_i, q) + d(q, c_j) \overset{\text{by Lemma 1.9}}{\leq} d(c_i, q) + d(c_i, q)/\alpha = \sqrt{2} d(c_i, q).$$

Combining this inequality with the inequality $d(c_i, c_j) > \sqrt{2} r_i$ from item 3,

we get that $d(c_i, q) > r_i$.

2. Assume that $q \in C_j$. If $d(c_j, q) \geq d(c_i, p)$, we have

$$d(p, q) \geq d(c_i, q) - d(c_i, p) \overset{\text{by Lemma 1.9}}{>} \alpha d(c_j, q) - d(c_i, p) \geq \sqrt{2} d(c_i, p).$$

If $d(c_j, q) < d(c_i, p)$, we similarly have

$$d(p, q) \geq d(c_j, p) - d(c_j, q) \overset{\text{by Lemma 1.9}}{>} \alpha d(c_i, p) - d(c_j, q) \geq \sqrt{2} d(c_i, p).$$

□

Now we sketch the algorithm of Balcan and Liang (2016). The algorithm consists of two stages. During the first stage, the algorithm employs a greedy approach: it starts with a trivial clustering of V, in which each vertex belongs to its own cluster. Then it repeatedly finds and links two "closest" clusters. The algorithm runs until it gets one cluster that contains all of the vertices. (Importantly, the algorithm does not stop when it gets k clusters — these k clusters are not necessarily optimal!) The result of the first stage of the algorithm is a binary decomposition tree $\mathcal{T}$ of V: the leaves of the tree are singleton clusters; internal nodes of $\mathcal{T}$ are intermediate clusters, obtained during the execution of the first stage; the root of $\mathcal{T}$ is V. We will show that each cluster C_i in the optimal clustering appears in the decomposition tree $\mathcal{T}$. During the second stage, the algorithm uses a simple bottom-up dynamic program to identify all clusters C_i in $\mathcal{T}$.

For the algorithm to succeed, it is important to use the right distance between clusters. We shall now define *the closure distance* to be used.

Definition 1.15. *We say that a point $x \in A$ is an r-central point for a set $A \subset V$ if it satisfies*

- *Coverage condition: $A \subset B(x, r)$.*
- *Padding condition: Every point p in $B(x, r)$ is closer to x than to any point outside of $B(x, r)$; that is, if $d(p, q) \leq d(p, x) \leq r$, then $d(q, x) \leq r$.*

Definition 1.16. *The closure distance $D_S(A_1, A_2)$ between two sets $A_1 \subset V$ and $A_2 \subset V$ is equal to the minimal r such that $A_1 \cup A_2$ has an r-central point.*

Note that the closure distance is well-defined since every point in $A_1 \cup A_2$ is r-central for $r = \operatorname{diam}(V) = \max_{u,v \in V} d(u, v)$.

Now we formally present Algorithm 1.3 (see the figure). It is clear that the algorithm runs in polynomial time. To prove the correctness of the algorithm, we need to show that every cluster C_i from the optimal clustering appears in the decomposition tree.

Algorithm 1.3 Clustering Algorithm

Create n singleton clusters — one for each vertex in V. Add them to $\mathcal{C}$. ▷ Stage 1
Initialize a tree $\mathcal{T}$. Add all singletons from $\mathcal{C}$ to $\mathcal{T}$.
while $|\mathcal{C}| \neq 1$ **do**
    Find two closest clusters A and B in $\mathcal{C}$ w.r.t. the closure distance.
    Merge A and B:
        Replace A and B with $A \cup B$ in $\mathcal{C}$.
        Add node $A \cup B$ to $\mathcal{T}$ and make it the parent of A and B.
end while
▷ Stage 2
Using bottom-up dynamic programming, find among all clusterings $(C'_1, \ldots, C'_k)$ of V, in which all C'_i appear in the decomposition tree $\mathcal{T}$, the clustering of minimum cost.
return clustering $(C'_1, \ldots, C'_k)$.

Lemma 1.12. *Consider two subsets A_1 and A_2 of C_i. Assume that $c_i \in A_1 \cup A_2$. Then $d_S(A_1, A_2) \leq r_i$.*

Proof. We show that c_i is an r_i-central point for $A_1 \cup A_2$. Indeed, by Lemma 1.11, item 1, $C_i = B(c_i, r_i)$. Thus $A_1 \cup A_2 \subset C_i = B(c_i, r_i)$. Now consider $p \in B(x_i, r_i)$ and $q \notin B(x_i, r_i)$. We have $p \in C_i$ and $q \notin C_i$, and from Lemma 1.11, item 2, we get that $d(p, q) < d(c_i, p)$. □

Lemma 1.13. *Assume that a set A contains points from both C_i and the complement of C_i. If a point x is Δ-central for A then $\Delta > r_i$.*

In particular, the closure distance between non-empty sets $A_1 \subset C_i$ and $A_2 \subset V \setminus C_i$ is at least r_i.

Proof. Consider two cases. First, assume that $x \in C_i$. Consider an arbitrary point $q \in A \setminus C_i$. Let C_j be the cluster q lies in (then, $j \neq i$). Since x is Δ-central for A and $q \in A$, we have $d(x, q) \leq \Delta$. By Lemma 1.11, item 2, $d(q, c_j) < d(q, x)$. From the definition of a central point, we get that $d(c_j, x) \leq \Delta$. By Lemma 1.9, $d(c_i, x) \leq \Delta/\alpha$. Therefore,

$$d(c_i, c_j) \leq d(c_i, x) + d(x, c_j) \leq \Delta/\alpha + \Delta = \sqrt{2}\Delta.$$

On the other hand, $d(c_i, c_j) > \sqrt{2}\, r_i$ by Lemma 1.11, item 3. We conclude that $\Delta > r_i$.

Now assume that $x \notin C_i$. Consider a point $p \in A \cap C_i$. Since x is a Δ-central point for A, we have $d(x, p) \leq \Delta$. By Lemma 1.11, item 2, point p is closer to c_i than to x. Thus by the definition of a central point, $c_i \in B(x, \Delta)$. On the other hand, by our assumption, $x \notin C_i = B(c_i, r_i)$. We get that $r_i < d(c_i, x) \leq \Delta$. This concludes the proof.

Now consider $A_1 \subset C_i$ and $A_2 \subset V \setminus C_i$. Applying the lemma to the set $A_1 \cup A_2$, we get that $D_S(A_1, A_2) \geq r_i$. □

Lemma 1.14. *Consider a cluster C_i in the optimal clustering.*
1. Let C be a cluster/node in the decomposition tree $\mathcal{T}$. Then

$$C \subset C_i, \quad C_i \subset C, \quad \text{or} \quad C \cap C_i = \varnothing. \tag{1.6}$$

2. C_i appears in the decomposition tree $\mathcal{T}$.

Proof. 1. We prove that the statement holds for all sets C in $\mathcal{C}$ by induction. Initially, all clusters C in $\mathcal{C}$ are singletons, and therefore, satisfy condition (1.6). Now suppose that we proved that condition (1.6) holds until some iteration, in which we merge clusters A and B, and obtain a cluster $C = A \cup B$. We need to prove that C also satisfies the condition. Note that C satisfies condition (1.6) in the following 3 cases:

- Neither A nor B intersects C_i. Then $C \cap C_i = \varnothing$.
- Both sets A and B are subsets of C_i. Then $C \subset C_i$.
- One of the sets A and B contains C_i. Then $C_i \subset C$.

The only remaining case is that one of the sets is a proper subset of C_i and the other does not intersect C_i; let us say $A \subset C_i$ and $B \subset \bar{C}_i$. We will show now that this case actually cannot happen.

Since A is a proper subset of C_i, there is another cluster $A' \subset C_i$ in $\mathcal{C}$. Furthermore, if $c_i \notin A$, then there is A' in $\mathcal{C}$ that contains c_i. By Lemma 1.12, point c_i is r_i-central for $A \cup A'$, and therefore $d_S(A, A') \leq r_i$. On the other hand, by Lemma 1.13, $d_S(A, B) > r_i \geq d_S(A, A')$. Therefore, A and B are not two closest clusters in $\mathcal{C}$ w.r.t. the closure distance. We get a contradiction.

2. Consider the smallest cluster C in $\mathcal{T}$ that contains C_i. If C is a singleton, then $C = C_i$. Otherwise, C is the union of its child clusters A and B. By item 1, both A and B are subsets of C_i, and so $C \subset C_i$. Therefore, $C = C_i$. □

1.4 References

S. Arora, J. Lee, and A. Naor. Euclidean distortion and the sparsest cut. *Journal of the American Mathematical Society*, 21(1):1–21, 2008.

P. Awasthi, A. Blum, and O. Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 112(1):49–54, 2012.

M.-F. Balcan and M. Braverman. Approximate Nash equilibria under stability conditions. Technical report, 2010.

M.-F. Balcan and Y. Liang. Clustering under perturbation resilience. 2016. To appear.

M.-F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the Symposium on Discrete Algorithms*, pages 1068–1077. Society for Industrial and Applied Mathematics, 2009.

M.-F. Balcan, N. Haghtalab, and C. White. Symmetric and asymmetric k-center clustering under stability. *arXiv preprint arXiv:1505.03924*, 2015.

Y. Bilu and N. Linial. Are stable instances easy? In *Innovations in Computer Science*, pages 332–341, 2010.

Y. Bilu, A. Daniely, N. Linial, and M. Saks. On the practically interesting instances of maxcut. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, pages 526–537, 2013.

A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.

N. Buchbinder, J. S. Naor, and R. Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the Symposium on Theory of Computing*, pages 535–544, 2013.

G. Călinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. In *Proceedings of the Symposium on Theory of Computing*, pages 48–52, 1998.

E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

M. E. Dyer and A. M. Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.

U. Feige and J. Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 674–683, 1998.

U. Feige, Y. Mansour, and R. Schapire. Learning and inference in the presence of corrupted inputs. In *Proceedings of the Conference on Learning Theory*, pages 637–657, 2015.

M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. volume 42, pages 1115–1145, 1995.

D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Mathematics of Operations Research*, 29(3):436–461, 2004.

R. M. Karp. Reducibility among combinatorial problems. Springer, 1972.

S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? *SIAM Journal on Computing*, 37 (1):319–357, 2007.

J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002.

V. Leontev. Stability of the traveling salesman problem (in Russian). volume 15, pages 1293–1309, 1975.

K. Makarychev and Y. Makarychev. Metric perturbation resilience. 2016.

K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proceedings of the ACM symposium on Theory of computing*, pages 367–384, 2012.

K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Sorting noisy data with partial information. In *Proceedings of the Conference on Innovations in Theoretical Computer Science*, pages 515–528, 2013.

K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Constant factor approximation for balanced cut in the pie model. In *Proceedings of the Symposium on Theory of Computing*, pages 41–49, 2014a.

K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Bilu—Linial stable instances of Max Cut and Minimum Multiway Cut. In *Proceedings of the Symposium on Discrete Algorithms*, pages 890–906, 2014b.

K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Correlation clustering with noisy partial information. In *Proceedings of the Conference on Learning Theory*, pages 1321–1342, 2015.

C. Mathieu and W. Schudy. Correlation clustering with noisy input. In *Proceedings of the Symposium on Discrete Algorithms*, pages 712–728, 2010.

M. Mihalák, M. Schöngens, R. Šrámek, and P. Widmayer. On the complexity of the metric tsp under stability considerations. In *SOFSEM 2011: Theory and Practice of Computer Science*, pages 382–393. Springer, 2011.

R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *Proceeding of the Symposium on Foundations of Computer Science*, pages 165–176, 2006.

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