

Exact Algorithms and Lower Bounds for Stable Instances of Euclidean k -MEANS*

Zachary Friggstad^{†1}, Kamyar Khodamoradi², and Mohammad R. Salavatipour^{‡1}

¹Department of Computing Science, University of Alberta

²Department of Computer Science, University of Regina

Abstract

We investigate the complexity of solving stable or perturbation-resilient instances of k -MEANS and k -MEDIAN clustering in fixed dimension Euclidean metrics (or more generally doubling metrics). The notion of stable or perturbation resilient instances was introduced by Bilu and Linial [2010] and Awasthi, Blum, and Sheffet [2012]. In our context, we say a k -MEANS instance is α -stable if there is a unique optimum solution which remains unchanged if distances are (non-uniformly) stretched by a factor of at most α . Stable clustering instances have been studied to explain why heuristics such as Lloyd’s algorithm perform well in practice. In this work we show that for any fixed $\epsilon > 0$, $(1 + \epsilon)$ -stable instances of k -MEANS in doubling metrics, which include fixed-dimensional Euclidean metrics, can be solved in polynomial time. More precisely, we show a natural multi-swap local-search algorithm in fact finds the optimum solution for $(1 + \epsilon)$ -stable instances of k -MEANS and k -MEDIAN in a polynomial number of iterations.

We complement this result by showing that this is essentially tight: that when the dimension d is part of the input, there is a fixed $\epsilon_0 > 0$ such there is not even a PTAS for $(1 + \epsilon_0)$ -stable k -MEANS in \mathbb{R}^d unless $\text{NP}=\text{RP}$. To do this, we consider a robust property of CSPs; call an instance stable if there is a unique optimum solution x^* and for any other solution x' , the number of unsatisfied clauses is proportional to the Hamming distance between x^* and x' . Dinur, Goldreich, and Gur have already shown stable QSAT is hard to approximation for some constant Q [19]. Recently, Paradise [31] extended this to the setting with bounded variable occurrence. More specifically, it implies that stable QSAT with bounded variable occurrence is APX-hard. Given this, we consider “stability-preserving” reductions to prove our hardness for stable k -MEANS. Such reductions seem to be more fragile and intricate than standard L -reductions and may be of further use to demonstrate other stable optimization problems are hard to solve.

*A preliminary version of this appeared in Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA) 2019.

[†]This research was undertaken, in part, thanks to funding from the Canada Research Chairs program and an NSERC Discovery Grant.

[‡]Supported by NSERC.

1 Introduction

The interest in explaining the difference between performance of many heuristic algorithms (in particular for clustering problems) in practice vs. worst-case performance bounds has recently attracted attention and led to new research directions. It has been long observed that for many optimization problems, such as clustering problems, the performance of some well known heuristics are much better than their worst case performance analysis.

There have been several approaches to study and explain these differences. Bilu and Linial [14] as well as Awasthi, Blum, and Sheffet [6] introduced the notion of stability and perturbation resilience. The idea is that for many problems, such as a clustering problem, a typical instance of the problem is stable in the sense that the optimum solution is unique and it does not change even if one modifies or perturbs input parameters by a small factor. Informally, instances of a problem are called α -stable or α -perturbation resilient if the structure of the optimum solution remains unchanged even if the input is perturbed by an α factor. For example, a clustering problem is α -stable if there is a unique optimum solution which remains the unique optimum after some distances are scaled up to a factor of α : different pairs of points may have their distances scaled differently.

Balcan et al. [9] argue that for clustering problems the goal is to find the “target” clustering and typically the objective function is just a proxy. Therefore, the distances of the input points and how they contribute to objective function are typically not very precise; thus small changes in these values usually does not change the target clustering. It has been shown that for α -stable instances of several problems such as centre-based clustering problems (e.g. k -CENTER, k -MEDIAN, k -MEANS), graph partitioning problems (e.g. Max-cut, Multiway cut), and other problems, one can find the optimum solution in polynomial time.

In this paper, we focus on α -stable instances of the classical clustering problems k -MEDIAN and k -MEANS in Euclidean metrics \mathbb{R}^d . Perhaps the most widely used clustering model is the k -MEANS clustering: Given a set \mathcal{X} of n data points in d -dimensional Euclidean space \mathbb{R}^d , and an integer k , find a set of k points $c_1, \dots, c_k \in \mathbb{R}^d$ to act as centres that minimize the sum of squared distances of each data point to its nearest centre. In other words, we would like to partition \mathcal{X} into k cluster sets, $\{C_1, \dots, C_k\}$ and find a centre c_i for each C_i to minimize

$$\sum_{i=1}^k \sum_{x \in C_i} \|x - c_i\|_2^2.$$

Here, $\|x - c_i\|_2$ is the standard Euclidean distance in \mathbb{R}^d between points x and c_i . This value is called the cost of the clustering. Typically, the centres c_i are selected to be the centroid (mean) of the cluster C_i . In other situations the centres must be from the data points themselves (i.e. $c_i \in C_i$) or from a given set \mathcal{C} . This latter version is referred to as discrete k -MEANS clustering. There are results that show that one can reduce k -MEANS to discrete version at a small loss (see [29]). In this paper we study discrete k -MEANS. The problem is known to be NP-hard even for $k = 2$ for \mathbb{R}^d when d is not fixed or for arbitrary k in \mathbb{R}^2 [3, 20, 27, 33]. Several approximation algorithms have been proposed for the problem; for a while the best ratio being a $(9 + \epsilon)$ via a local search algorithm [25]. This was recently improved to a PTAS independently by [22, 23] and [16] when the dimension d can be regarded as a constant and an 6.357-approximation for arbitrary dimensions [1]. Awasthi et al. [8] showed that k -MEANS is APX-hard in \mathbb{R}^d when $d = \Omega(\log n)$. Improvements and extensions of this lower bound in high-dimensional spaces have appeared in [26, 15] with the current-best lower bound for discrete k -MEANS in Euclidean spaces being 1.73 [17]. We briefly remark here that their techniques do not imply hardness of stable clustering problems, this will be discussed further in Section 1.2.

We now precisely define what it means for an instance of k -MEANS be stable in our paper. One can similarly define what it means for a k -MEDIAN instance to be α -stable. We present a k -MEANS instance as a triple $(\mathcal{X}, \mathcal{C}, \delta)$ where δ is a symmetric distance function between points in $\mathcal{X} \cup \mathcal{C}$ that satisfies the triangle inequality unless otherwise explicitly stated. A solution is viewed as a set $\mathcal{S} \subseteq \mathcal{C}$ with $|\mathcal{S}| = k$ and its cost is $\text{cost}(\mathcal{S}) := \sum_{j \in \mathcal{X}} \delta(j, \mathcal{S})^2$ where we let $\delta(j, \mathcal{S})$ to denote $\min_{i \in \mathcal{S}} \delta(j, i)$.

Definition 1 (α -stability). *For $\alpha \geq 1$, call an instance $\mathcal{I} = (\mathcal{X}, \mathcal{C}, \delta)$ of metric k -MEANS α -stable if it has a unique optimum solution \mathcal{O} which is also the unique optimum solution in every related (not necessarily metric) instance $\mathcal{I}' = (\mathcal{X}, \mathcal{C}, \delta')$ (that need not satisfy the triangle inequality but still satisfies symmetry) with $\delta(i, j) \leq \delta'(i, j) \leq \alpha \cdot \delta(i, j)$ for all $i, j \in \mathcal{X} \cup \mathcal{C}$.*

Several papers have studied complexity of α -stable instances of k -MEANS and k -MEDIAN. The main goal is to find algorithms that work for smaller values of α (i.e. weak requirement for stability). Awasthi, Blum, and

Sheffet [7] showed that 3-stable instances of k -MEANS and k -MEDIAN can be solved in polynomial time. Balcan and Liang [11] improved this by showing that for $\alpha = 1 + \sqrt{2}$, α -stable instances of k -MEANS and k -MEDIAN can be solved in polynomial time. This was further improved in the case of metric stability recently by Agelidakis, Makarychev, Makarychev [4] who showed that 2-metric stable (or 2-metric perturbation resilient) instances of centre-based clustering problems such as k -MEANS and k -MEDIAN can be solved in polynomial time.

In this work we focus on discrete k -MEANS and k -MEDIAN on Euclidean metrics \mathbb{R}^d (and more generally doubling metrics) and prove both upper and lower bounds. We prove that for any fixed $\epsilon > 0$, $(1 + \epsilon)$ -stable instances of these problems on fixed dimension Euclidean spaces (\mathbb{R}^d for fixed d) can be solved in polynomial time and this is tight. In fact, our result is slightly stronger in that we show $(1 + \epsilon)$ -stable instances do not even admit a PTAS unless NP=RP.

More specifically, starting with a new proximity PCP theorem explained below in Theorem 3 (which is a bounded-occurrence version of Theorem 3.1 of [19]) we can show that for some fixed $\epsilon > 0$, $(1 + \epsilon)$ -stable instances of k -MEANS and k -MEDIAN cannot be solved in polynomial time when restricted to \mathbb{R}^d (but unbounded d) unless NP=RP. We had presented Theorem 3 as a hypothesis in the conference version of our paper. Since then, Paradise [31] has proved a bounded-occurrence version of Theorem 3.1 of [19] which directly implies Theorem 3.

1.1 Previous work

Bilu and Linial [14] gave a polynomial exact algorithm for $O(n)$ -stable instances of Max-Cut. This was improved to $O(\sqrt{n})$ -stable instances by Bilu, Daniely, Linial, Saks [13] and further by Makarychev, Makarychev, Vijayaraghavan [28] who provided a polynomial exact algorithm based on semidefinite programming for $O(\sqrt{\log n} \log \log n)$ -stable instances of Max-Cut. This result may be nearly tight: [28] also shows that solving $\alpha(n)$ -stable instances in polynomial time would imply an $\alpha(2n)$ -approximation for the nonuniform sparsest cut problem.

Awasthi, Blum, and Sheffet [7] showed that for 3-stable instances of large class of clustering problems, called separable centre-based objective (s.c.b.o) clustering problems (such as k -MEDIAN over finite metrics (with no Steiner points)) and for $(2 + \sqrt{3})$ -stable instances of s.c.b.o clustering problems with Steiner points one can find the optimum clustering in polynomial time. Furthermore, they proved NP-hardness for instances of k -MEDIAN with Steiner points that satisfy *3-centre proximity* condition (α -centre proximity is the condition that for any point $x \in \mathcal{X}$ in cluster C_i with cluster centre c_i , $\alpha \cdot \delta(x, c_i) < \delta(x, c_j)$ if $i \neq j$). Ben-David and Reyzin [12] showed the NP-hardness of $(2 - \epsilon)$ -stable instances of k -MEDIAN with no Steiner points for general metrics. Balcan, Haghtalab, and White [10] prove that 2-stable instances of k -centre can be solved in polynomial time and any $(2 - \epsilon)$ -stable instances of the problem are NP-hard. Angelidakis, Makarychev, and Makarychev [4] show that for class of clustering problems called centre-based clustering (such as k -MEANS, k -MEDIAN, k -CENTER), 2-metric perturbation resilient instances can be solved in polynomial time, improving the bound of $1 + \sqrt{2}$ from Balcan and Liang [11]. Ostrovsky et al. [30] showed that for ϵ -separated (defined below) instances of continuous k -MEANS a variant of Lloyd's algorithm is an $O(1)$ -approximation.

For an instance of continuous k -MEANS with $\mathcal{X} \in \mathbb{R}^d$, let $\Delta_k^2(\mathcal{X})$ denote the optimal k -MEANS clustering cost. Say that the instance given by \mathcal{X} is ϵ -separated if $\Delta_k^2(\mathcal{X}) \leq \epsilon^2 \Delta_{k-1}^2(\mathcal{X})$. Ostrovsky et al. [30] showed that one can achieve a $(1 + f(\epsilon))$ -approximation to k -MEANS in polynomial time. This result was further improved by Awasthi, Blum, and Sheffet [6] that if $\Delta_k^2(\mathcal{X}) \leq \alpha \Delta_{k-1}^2(\mathcal{X})$ for some constant $\alpha > 1$ then one can obtain a PTAS for k -MEANS in time polynomial in n, k but exponential in α, ϵ . A solution S_0 to k -MEDIAN is $\frac{1}{\epsilon}$ -locally optimal if any solution S_1 such that $|S_0 - S_1| + |S_1 - S_0| \leq 2/\epsilon$ has cost at least as big as that of S_0 . Cohen-Addad and Schwiegelshohn [18] showed that for $\alpha > 3$, for any instance of k -MEDIAN that is α -stable, any $\frac{2}{\alpha-3}$ -locally optimal solution is optimum. Hence a local search algorithm that swaps up to $\frac{2}{\alpha-3}$ centres finds the optimum solution. However, they do not show how to find such a local optima in polynomial time.

Vijayaraghavan, Dutta, and Wang [35] studied additive perturbation stable (APS) instances of Euclidean k -MEANS for $k = 2$. An instance is δ -APS if the (unique) optimum clustering remains optimum even if each point is moved up to δ . They [35] showed that for any fixed $\epsilon > 0$, ϵ -additive instances of Euclidean k -MEANS for $k = 2$ can be solved in polynomial time. There are also several results on stable instances of graph partitioning problems such as Max-Cut.

Another interesting aspect of our result is that we prove the local search dynamics find the optimum solution in polynomial time in stable instances of k -MEDIAN and k -MEANS. This stands in stark contrast to the fact that the complexity of finding a local minimum for the standard local search algorithm is PLS-complete for k -MEDIAN [2]. We refer the interested reader to [2] and the references therein for more details of Polynomial

Local Search complexity. Essentially, our result shows that if stable instances of k -MEDIAN and k -MEDIAN too are PLS-hard then the fact that local search terminates in a polynomial number of iterations would allow us to solve all problems in PLS in polynomial time. The fact that local search terminates in polynomial time for stable instances is not surprising, but we finally provide the first proof of this fact.

1.2 Our Results

Our main results are Theorems 1 and 4 below. Recall a metric has doubling dimension d if any ball with radius R in the metric can be covered by at most 2^d balls of radius $R/2$. Thus, d -dimensional Euclidean metrics have doubling dimension $O(d)$.

Theorem 1. *For any fixed $d \geq 1$ and $\epsilon' > 0$, $(1 + \epsilon')$ -stable instances of k -MEANS and k -MEDIAN in metrics with doubling dimension d can be solved in polynomial time.*

This theorem is proved by showing that the simple ρ -swap local search algorithm for a suitable constant $\rho = \rho(\epsilon', d)$ finds the optimum k -MEANS clustering in polynomial time if the best improvement is taken in each iteration. We should note that, in all the previous studies of local search algorithms, in order to obtain polynomial run time, a swap is performed if it yields a “significant” improvement of the solution. Hence, the result of the algorithms is not a true local optimum, but, in some sense, an approximate local optimum. However, in order to find the actual optimum, one cannot rely on an algorithm that produces an approximate local optimum. For instance, the result of Cohen-Addad and Schwiegelshohn [18] shows that a *true* local optimum is also optimum in α -stable instance of k -MEDIAN (for $\alpha > 3$) but it does not show how to find a true local optimum in polynomial time. In order to prove Theorem 1 we must show that the local search algorithm that performs the best swap in each step in fact finds the true local (and hence global) optimum in stable instances. We focus only on our setting of doubling metrics, but the ideas can also be used to show how to find the global optimum in polynomial time for α -stable instances of k -MEANS and k -MEDIAN in general metrics studied in [18] for constant $\alpha > 3$.

As a side effect, we also show how to avoid the “ ϵ -loss” that so many local search procedures lose when being modified to run in polynomial time. For example, a local optimum solution to the single-swap heuristic for k -MEDIAN is known have cost at most 5 times the global optimum cost, yet a modification to the standard single-swap algorithm in [5] to ensure the algorithm runs in polynomial time is a $(5 + \epsilon)$ -approximation. We provide analysis of the local-search procedure that takes the best swap at each step and prove the solution is a true 5-approximate solution (no ϵ -loss) after a polynomial number of iterations, even if it has not yet stabilized at a local optimum. Our approach may be helpful for others to communicate more clean approximation ratios for their local search algorithms. The details of this analysis technique appear in Appendix A.

Our second major result is to show that Theorem 1 is essentially tight in that the assumption of d being constant is critical to allow instances with arbitrarily small (constant) stability to be solved optimally.

Part of our reduction is inspired by recent work on k -MEANS hardness in Euclidean spaces, notably on ideas in the reduction from [8]. It should be noted that their reduction has each point being at a distance 1 or some constant $c > 1$ from every possible centre and that in the “yes” case there is a solution where every point is within distance 1 of a centre. But this does not guarantee the instance is stable, even if one follows parsimonious reductions from UNAMBIGUOUS QSAT (definition below) to ensure there is a unique solution where each point pays 1 in its cluster. The simple reason is that stability is a much stronger requirement of a problem: slight perturbation of distances could change the optimal solution structure even in instances with distances 1 and $c > 1$. It is not difficult to come up with such examples.

In order to prove k -MEANS is hard even in stable instances, we prove Theorem 4 using a new PCP construction by Paradise [31], which is a slightly stronger version of Theorem 3.1 in [19]. We introduced this PCP construction as a hypothesis in the conference version of this paper. Since then, Paradise [31] has actually proved this hypothesis. We need the following definitions to state our result formally.

Definition 2 (UNAMBIGUOUS QSAT). *In an instance of promise problem U-QSAT, we are given a set of n variables x_1, x_2, \dots, x_n and m clauses C_1, C_2, \dots, C_m where each clause is a Q -CNF over Q (distinct) variables. The promise is that there is at most one satisfying assignment.*

U-QSAT was proven hard using a randomized reduction in [32] in the sense that an algorithm that can be used to determine satisfiability of a U-QSAT instance could then be used to solve any language in NP with a randomized reduction. That is, we would have $NP = RP$.

Given two binary vectors x, x' of the same length, let $HW(x, x') \in [0, 1]$ denote the Hamming weight of x, x' : the fraction of coordinates i with $x_i \neq x'_i$. The following is a corollary of Theorem 3.1 of [19], obtained by producing their PCPP for a given instance of U-QSAT.

Theorem 2. (Theorem 3.1 of [19]) *There are universal constants $q, s, \epsilon > 0$ such that the following holds. For every $L \in NP$ there is a polynomial time randomized reduction from L to an instance Φ of U-QSAT with the following properties:*

- Yes case:** *if L is a yes case then Φ has a unique satisfying assignment x^* with probability $\Omega(1/\text{poly}(n))$. Also, for any assignment x to Φ , the fractions of clauses not satisfied by x is at least $s \cdot HW(x, x^*)$.*
- No case:** *if L is a no case then no assignment satisfies more than $(1 - \epsilon)$ -fraction of clauses of Φ .*

An instance of U-QSAT-B is the same as U-QSAT with the additional condition that each variable appears in a bounded number of clauses. We typically use B to refer to this bound as well. The following bounded occurrence version of Theorem 2 is the basis for our hardness result. It simply repeats Theorem 2 with the condition that the SAT instance has bounded occurrence for each variable.

Theorem 3. [Hypothesis 1 in [21]] *There are universal constants $B, Q, s, \epsilon > 0$ such that: For every $L \in NP$ there is a polynomial time (randomized) reduction from L to an instance Φ of U-QSAT-B with the following properties:*

- Yes case:** *if L is a yes case then Φ has a unique satisfying assignment x^* with probability $\Omega(1/\text{poly}(n))$. Also, for any assignment x to Φ , the fractions of clauses not satisfied by x is at least $s \cdot HW(x, x^*)$.*
- No case:** *if L is a no case then no assignment satisfies more than $(1 - \epsilon)$ -fraction of clauses of Φ .*

As mentioned above, we stated the above as a hypothesis in the conference version of this paper [21]. After the conference appeared, Paradise [31] proved a stronger version of Theorem 3.1 of [19]. Rather than recalling all definitions in [31], we simply summarize the main features of their result. Theorem 1.6 in [31] shows every $L \in NP$ admits a PCP verifier that is:

- **Strong:** The reduction to the PCP is parsimonious in that there is a 1-to-1 correspondence between witnesses for yes instances of L and proofs that are accepted with probability exactly 1 by the verifier: the so-called **canonical proofs** \mathcal{C} . Further, for any proof π the probability the PCP rejects π is $\Omega(HW(\pi, \mathcal{C}))$: i.e. at least proportional to the minimum Hamming distance between π and a proof in \mathcal{C} .
- **Smooth:** All bits of the proof are queried by the verifier with the equal probability.

The proof of Theorem 3 is then immediate. First, use the reduction in [32] to reduce L to an instance of the promise problem U-QSAT. Then apply the PCP verifier from Theorem 1.6 in [31] followed by the reduction to QSAT-B from the proof of Corollary 1.8 in [31]. This directly establishes Theorem 3 with no further arguments being required. Note that the randomization in the reduction only comes from the reduction in [32] proving U-QSAT is hard, every step from [31] is deterministic.

We will frequently refer to SAT instances with the properties mentioned in the Yes case of Theorem 3, so it will be convenient to use two more definitions.

Definition 3 (Stable SAT Instances). *For $0 \leq s \leq 1$, an instance Φ of SAT is said to be s -stable if there is exactly one satisfying assignment x^* and for any assignment x the fraction of clauses of Φ that are not satisfied is at least $s \cdot HW(x^*, x)$.*

Definition 4 (S-QSAT-B). *In an instance of promise problem S-QSAT-B, we are given an instance Φ of QSAT-B with the following guarantee: Either Φ is s -stable or it is not satisfiable.*

So, Theorem 3 gives a randomized reduction from any $L \in NP$ to QSAT-B that always maps a no instance to a no instance and, with polynomially-large probability, maps a yes instance to an instance of S-QSAT-B which, by definition, has exactly one satisfying assignment.

Using Theorem 3 we prove our main result for hardness of approximating Stable k -Means in Euclidean metrics whose dimension is not fixed.

Theorem 4. *There exists universal constants $\epsilon, \gamma > 0$ such that there is no $(1 + \gamma)$ -approximation for $(1 + \epsilon)$ -stable instances of k -MEANS in \mathbb{R}^d unless $NP = RP$.*

Before our work, it was not known that it was even hard to solve stable k -MEANS in Euclidean spaces exactly much less hard to approximate.

Hardness Reduction Techniques: Our goal is to take a stable instance of QSAT-B and map it to a stable instance of k -MEANS. The definitions of stability in these two problems, of course, differ and our goal is to not only provide a hardness reduction for k -MEANS but also to translate the notion of stability from QSAT-B to k -MEANS. To this end, we informally call a reduction “stability-preserving” if it maps stable instances of one problem to stable instances of the other problem.

A strong caution to the reader is that standard L -reductions do not always preserve stability. In decision problems like QSAT-B or 3D-MATCHING (an auxiliary problem we encounter on the way to proving hardness for stable k -MEANS), we certainly need our reduction to be parsimonious to ensure uniqueness of the optimal solution. But even parsimonious L -reductions do not suffice. In Appendix B, we given an example of a stable QSAT and show the classic parsimonious L -reduction from QSAT to QSAT-B for some constant B that is based on expander graphs fails to preserve stability¹. Ultimately, this shows why we required the newer construction by Paradise [31] rather than the previous construction in [19] providing a strong PCP for every language in NP but does not guarantee smoothness.

Thus, stability-preserving reductions are more fragile than L -reductions. The arguments about why stability is preserved are also more challenging and in-depth than the standard “no-case” analysis in an L -reduction. We believe such reduction may be interesting in other contexts, especially in proving hardness for other problems under certain stability assumptions.

Outline of the paper: The algorithm for solving stable instances of k -MEANS in constant-dimension doubling metrics appears in Section 2. The presentation focuses only on k -MEANS, the algorithm for solving stable instances of k -MEDIAN is nearly identical.

Some details of the hardness reduction are given in Section 3. Then the reduction is broken into three steps. In Section 4, we begin by reducing S-QSAT-B to S-3SAT-B which also serves as our introduction to the concept of stability-preserving reductions. In Section 5, we provide a stability-preserving reduction from S-3SAT-B to the classic 3D-MATCHING problem with appropriate stability and bounded-degree guarantees maintained in the reduction. A further step in reduction is needed to reduce from stable instances of 3D-MATCHING to a covering problem. This simple step appears in Section 6. Finally, Section 7 finishes with the reduction to an α -stable Euclidean k -MEANS instance in \mathbb{R}^d with $d = \Omega(n)$ for some absolute constant $\alpha > 1$ that, ultimately, depends on the Q, B and s from Theorem 3. The ϵ from Theorem 3 also factors into showing hardness of even approximating α -stable k -MEANS instances in high-dimensional Euclidean metrics within some small constant factor.

2 Solving Low-Dimensional Stable Instances

Our main goal in this section is to prove Theorem 1. We prove it for k -MEANS only, the proof for k -MEDIAN is essentially identical. Suppose ϵ', d are fixed constants and we are given an instance $(\mathcal{X}, \mathcal{C}, \delta)$ of k -MEANS in a doubling metric with doubling dimension d that is $(1 + \epsilon')$ -stable; i.e. it has a unique optimum solution $\mathcal{O} \subseteq \mathcal{C}$ and it remains the unique optimum solution even if distances between points in $\mathcal{X} \cup \mathcal{C}$ are scaled (non-uniformly) by at most $(1 + \epsilon')$ factor. If the reader is not comfortable with doubling metrics, nothing will be lost by thinking of \mathbb{R}^d in particular whose doubling dimension is $\Theta(d)$.

Let ϵ be such that $1 + 6\epsilon = (1 + \epsilon')^2$, roughly speaking we have $\epsilon \approx \epsilon'/3$ for small ϵ' . Without loss of generality, we assume $\epsilon \leq 1/6$ since we can shrink ϵ' if necessary; an instance that is $(1 + \epsilon')$ -stable is also $(1 + \epsilon'')$ -stable for $\epsilon'' < \epsilon'$. Let $\rho := \rho(\epsilon, d)$ be the constant from the k -MEANS local search algorithm in [22, 23]. We briefly recall that the ρ -swap local search analysis for k -MEANS in Euclidean metrics of dimension d finds a solution whose cost is at most $1 + \epsilon$ times the optimum solution cost. For impressions, $\rho = d^{O(d)} \cdot \epsilon^{-O(d/\epsilon)}$.

Let $\mathfrak{F}_k = \{\mathcal{S} \subseteq \mathcal{C} : |\mathcal{S}| = k\}$ be set of *feasible* solutions. Consider Algorithm 1, which is the standard ρ -swap local search algorithm with slight modification that in each step it performs the swap that yields the best improvement.

Each iteration runs in polynomial time because ρ is a constant. Unlike standard polynomial-time local search algorithms that stop once no improvement by a factor of ϵ/k can be made, our algorithm simply terminates

¹Paradise [31] shows this reduction does preserve stability if the QSAT instance has each variable appearing in the same number of clauses, which would follow from the smooth property of their PCP.

Algorithm 1 ρ -Swap Local Search

```
let  $\mathcal{S}$  be any set in  $\mathfrak{F}_k$ 
while  $\exists$  sets  $\mathcal{S}' \in \mathfrak{F}_k$  with  $|\mathcal{S} - \mathcal{S}'| \leq \rho$  and  $\text{cost}(\mathcal{S}') < \text{cost}(\mathcal{S})$  do
     $\mathcal{S} \leftarrow \arg \min_{\substack{\mathcal{S}' \in \mathfrak{F}_k \\ |\mathcal{S} - \mathcal{S}'| \leq \rho}} \text{cost}(\mathcal{S}')$ 
return  $\mathcal{S}$ 
```

once no improvement is possible at all. We will argue that the algorithm terminates in a polynomial number of iterations in $(1 + \epsilon')$ -stable instances and give an explicit bound on the number of iterations.

An interesting observation on the quality of the returned solutions in non-stable instances is made at the end of this section: informally it says that if we truncate the main loop of Algorithm 1 to a polynomial number of iterations, then in $(1 + \epsilon')$ -stable instances it finds the optimum solution and, further, in arbitrary instances of k -MEANS in \mathbb{R}^d it finds a $(1 + O(\epsilon))$ -approximate solution.

Let $\mathcal{O} \in \mathfrak{F}_k$ be the unique optimum solution. For any set $\mathcal{S} \in \mathfrak{F}_k$, define the following:

- For $j \in \mathcal{X}$, let $\sigma(j, \mathcal{S})$ be the centre in \mathcal{S} nearest to j , breaking ties by some fixed ordering of \mathcal{C} .
- $\bar{\mathcal{X}}_{\mathcal{S}} = \{j \in \mathcal{X} : \sigma(j, \mathcal{S}) \in \mathcal{S} - \mathcal{O} \text{ and } \sigma(j, \mathcal{O}) \in \mathcal{O} - \mathcal{S}\}$.
- $\Psi(\mathcal{S}) = \sum_{j \in \bar{\mathcal{X}}_{\mathcal{S}}} \delta(j, \sigma(j, \mathcal{S}))^2 + \delta(j, \sigma(j, \mathcal{O}))^2$.

Here is why we define the function $\Psi(\cdot)$. In the analysis of local search algorithms such as in [22, 23], in order to show that a local optimum solution \mathcal{S} is $(1 + \epsilon)$ -approximate, one shows that $\text{cost}(\mathcal{S}) \leq \text{cost}(\mathcal{O}) + O(\epsilon)(\text{cost}(\mathcal{O}) + \text{cost}(\mathcal{S}))$. That bound is too crude for our purposes here. Instead, we require $\text{cost}(\mathcal{S}) \leq \text{cost}(\mathcal{O}) + \epsilon \cdot \Psi(\mathcal{S})$, i.e. the error term is not an ϵ factor of $\text{cost}(\mathcal{O}) + \text{cost}(\mathcal{S})$; instead it is only an ϵ factor of the cost of \mathcal{O} and \mathcal{S} for points in $\bar{\mathcal{X}}_{\mathcal{S}}$. The function $\Psi(\mathcal{S})$ is a bit challenging to track, it does not necessarily decrease as $\text{cost}(\mathcal{S})$ decreases. Still, it is an important quantity in our analysis.

Definition 5. *Say $\mathcal{S} \in \mathfrak{F}_k$ is a nearly-good solution if $\text{cost}(\mathcal{S}) \leq \text{cost}(\mathcal{O}) + 2\epsilon \cdot \Psi(\mathcal{S})$.*

2.1 A Structural Theorem

We fix some $\mathcal{S} \in \mathfrak{F}_k$ in this section, which may or may not be a local optimum solution. Let $\mathcal{S}' = \mathcal{S} - \mathcal{O}$ and let $\mathcal{O}' = \mathcal{O} - \mathcal{S}$. We will use analysis from [22, 23] to handle clients in $\bar{\mathcal{X}}_{\mathcal{S}}$.

For now, consider the k -MEANS instance with points $\bar{\mathcal{X}}_{\mathcal{S}}$, possible centres $\mathcal{C}' := \mathcal{S}' \cup \mathcal{O}'$, and $k' = |\mathcal{S}'| = |\mathcal{O}'|$. Note $\mathcal{S}', \mathcal{O}'$ are disjoint, which was a technical requirement for the analysis in [22, 23]. For a subset $P \subseteq \mathcal{S}' \cup \mathcal{O}'$ with $|P \cap \mathcal{S}'| = |P \cap \mathcal{O}'|$, define Δ_j^P for each $j \in \bar{\mathcal{X}}_{\mathcal{S}}$ to be $\delta(j, \sigma(\mathcal{S}' \Delta P))^2 - \delta(j, \sigma(\mathcal{S}'))^2$, i.e. the change in j 's assignment cost if we replaced solution \mathcal{S}' with $\mathcal{S}' \Delta P$.

Now, it is not necessarily clear that \mathcal{O}' is an optimal solution for this restricted instance of k -MEANS nor is it clear that \mathcal{S}' is a locally-optimal solution with respect to the ρ -swap heuristic. However, we do not need these facts. No inequality in [22, 23] requires that \mathcal{O}' be a global optimum solution (it is just used to conclude that a local optimum solution is a near-optimal solution). Furthermore, all upper bounds on Δ_j^P terms in [22, 23] do not require local optimality of \mathcal{S}' : local optimality was used there simply to show $\Delta_j^P \geq 0$ if $|P| \leq \rho$ but we do not use this bound.

Thus, the following holds in our setting which follows from results in [22, 23].

Theorem 5. *There is a randomized algorithm that samples a partition π of $\mathcal{S}' \cup \mathcal{O}'$ with the following properties:*

- $|P \cap \mathcal{S}'| = |P \cap \mathcal{O}'| \leq \rho$ for each $P \in \pi$ and
- $\mathbf{E}_{\pi} \left[\sum_{P \in \pi} \sum_{j \in \bar{\mathcal{X}}_{\mathcal{S}}} \Delta_j^P \right] \leq \sum_{j \in \bar{\mathcal{X}}_{\mathcal{S}}} (1 + \epsilon) \cdot \delta(j, \mathcal{O}')^2 - (1 - \epsilon) \cdot \delta(j, \mathcal{S}')^2$.

The randomized partition is described by Theorem 3.2 in [23] and the upper bound on the expected cost change when performing this swap is given at the end of Section 3.2 in [23] (for the appropriate choice of ρ). A careful reader can verify Theorem 5 indeed holds simply by reading Section 3.2 in [23] plus appropriate definitions preceding that section. Instead of reiterating every argument from that paper to show this holds,

we simply point out that Theorem 5 does not require optimality of \mathcal{O}' nor local optimality of \mathcal{S}' . Again for emphasis, in [23] local optimality was only used to show $0 \leq \mathbf{E}_\pi [\sum_{P \in \pi} \Delta_j^P]$ which we do not require.

We use Theorem 5 to show the following main technical result that supports our analysis of the running time of Algorithm 1 and proves it returns \mathcal{O} in stable instances is the following. Intuitively, it says that if the solution is not nearly-good then the next step of the local search algorithm will make a significant step.

Theorem 6. *For any $\mathcal{S} \in \mathfrak{F}_k$, if $\text{cost}(\mathcal{S}) > \text{cost}(\mathcal{O}) + \epsilon \cdot \Psi(\mathcal{S})$ then there is some $\mathcal{S}' \in \mathfrak{F}_k$ with $|\mathcal{S} - \mathcal{S}'| \leq \rho$ where*

$$\text{cost}(\mathcal{S}') \leq \text{cost}(\mathcal{S}) + \frac{\text{cost}(\mathcal{O}) - \text{cost}(\mathcal{S}) + \epsilon \cdot \Psi(\mathcal{S})}{k}.$$

Proof. Sample a random partition π of $\mathcal{O}' \cup \mathcal{S}'$ as in Theorem 5 but now consider the effect of the swap $\mathcal{S} \rightarrow \mathcal{S} \Delta P$ for each part $P \in \pi$. We place an upper bound on $\mathbf{E}_\pi [\sum_{P \in \pi} \text{cost}(\mathcal{S} \Delta P) - \text{cost}(\mathcal{S})]$ by describing a valid way to redirect each $j \in \mathcal{X}$ in each swap on a case-by-case as follows. For brevity, let $c_j^* = \delta(j, \sigma(j, \mathcal{O}))^2$ be the cost of connecting j in the global optimum solution and, analogously, $c_j = \delta(j, \sigma(j, \mathcal{S}))^2$.

- We never move any j with both $\sigma(j, \mathcal{S}), \sigma(j, \mathcal{O}) \in \mathcal{S} \cap \mathcal{O}$. Note $\sigma(j, \mathcal{S})$ remains open after each swap so this is valid. Observe for such clients that $c_j^* = c_j$ so we, conveniently, say the total assignment cost change for j over all swaps $P \in \pi$ is bounded by $c_j^* - c_j$.
- For j with $\sigma(j, \mathcal{S}) \in \mathcal{S}'$ and $\sigma(j, \mathcal{O}) \in \mathcal{S} \cap \mathcal{O}$, move j to $\sigma(j, \mathcal{O})$ when swapping the part P with $\sigma(j, \mathcal{S}) \in P$. As $\sigma(j, \mathcal{S})$ remains open when swapping all other $P' \neq P$, we can leave j assigned to $\sigma(j, \mathcal{S})$ to upper bound its cost change for swaps $P' \neq P$ by 0. The total cost assignment for j is then bounded by $c_j^* - c_j$.
- For j with $\sigma(j, \mathcal{S}) \in \mathcal{S} \cap \mathcal{O}$ and $\sigma(j, \mathcal{O}) \in \mathcal{O}'$, move j to $\sigma(j, \mathcal{O})$ when swapping the part P with $\sigma(j, \mathcal{O}) \in P$ and do not move j when swapping any other part $P' \neq P$. This places an upper bound of $c_j^* - c_j$ on the total assignment cost change for j .
- Finally, consider j with $\sigma(j, \mathcal{S}) \in \mathcal{S}'$ and $\sigma(j, \mathcal{O}) \in \mathcal{O}'$. Note these are precisely the points $j \in \overline{\mathcal{X}}_{\mathcal{S}}$. From Theorem 6,

$$\mathbf{E}_\pi \left[\sum_{P \in \pi} \Delta_j^P \right] \leq (1 + \epsilon) \cdot c_j^* - (1 - \epsilon) \cdot c_j = c_j^* - c_j + \epsilon \cdot (c_j + c_j^*).$$

Aggregating this cost bound for all clients, we see

$$\mathbf{E}_\pi \left[\sum_{P \in \pi} \text{cost}(\mathcal{S} \Delta P) - \text{cost}(\mathcal{S}) \right] \leq \text{cost}(\mathcal{O}) - \text{cost}(\mathcal{S}) + \epsilon \cdot \Psi(\mathcal{S}).$$

Therefore there is some π and some $P \in \pi$ with

$$\text{cost}(\mathcal{S} \Delta P) - \text{cost}(\mathcal{S}) \leq \frac{\text{cost}(\mathcal{O}) - \text{cost}(\mathcal{S}) + \epsilon \cdot \Psi(\mathcal{S})}{|\pi|} \leq \frac{\text{cost}(\mathcal{O}) - \text{cost}(\mathcal{S}) + \epsilon \cdot \Psi(\mathcal{S})}{k},$$

the latter bound using $|\pi| \leq k$ and the fact the numerator is negative. \square

2.2 Polynomial-Time Convergence to a Nearly-Good Solution

In order to show that Algorithm 1 terminates in polynomial time on stable instances, we first show that a nearly-good solution will be encountered by Algorithm 1 within a polynomial number of iterations even if the instance is not stable. The next section shows that the only nearly-good solution is the optimal solution in $(1 + \epsilon')$ -stable instances, thereby completing the proof of Theorem 1.

From Theorem 6, we show solutions that are not nearly-good are improved significantly in a single step of the local search algorithm (i.e. in terms of their full cost, not just when considering clients in $\overline{\mathcal{X}}_{\mathcal{S}}$).

Lemma 1. *Suppose $\mathcal{S} \in \mathfrak{F}_k$ is a solution which is not nearly-good. Then there is some $\mathcal{S}' \in \mathfrak{F}_k$ with $|\mathcal{S} - \mathcal{S}'| \leq \rho$ satisfying*

$$\text{cost}(\mathcal{S}') - \text{cost}(\mathcal{O}) \leq \left(1 - \frac{1}{2k}\right) \cdot (\text{cost}(\mathcal{S}) - \text{cost}(\mathcal{O})).$$

Proof. Consider the set \mathcal{S}' guaranteed by Theorem 6. The fact that \mathcal{S} is not a nearly-good solution means

$$\begin{aligned} \text{cost}(\mathcal{S}') - \text{cost}(\mathcal{O}) &\leq \text{cost}(\mathcal{S}) - \text{cost}(\mathcal{O}) + \frac{\text{cost}(\mathcal{O}) - \text{cost}(\mathcal{S}) + \epsilon \cdot \Psi(\mathcal{S})}{k} \\ &< \text{cost}(\mathcal{S}) - \text{cost}(\mathcal{O}) + \frac{\text{cost}(\mathcal{O}) - \text{cost}(\mathcal{S})}{2k} = \left(1 - \frac{1}{2k}\right) \cdot (\text{cost}(\mathcal{S}) - \text{cost}(\mathcal{O})). \end{aligned}$$

□

To argue about the number of iterations of Algorithm 1, we make the assumption that all coordinates of all points in $\mathcal{X} \cup \mathcal{C}$ are integers. This is without loss of generality: scaling all points by the product of all denominators increases the bit complexity of the input by a polynomial factor and Algorithm 1 would behave exactly as it would before the scaling (i.e. would consider the same sequence of sets \mathcal{S}).

Let $\Delta = \max_{j \in \mathcal{X}, i \in \mathcal{C}} \delta^2(i, j)$. Observe that $\text{cost}(\mathcal{S}) - \text{cost}(\mathcal{O}) \leq n\Delta$, $\text{cost}(\mathcal{S})$ is an integer for any $\mathcal{S} \in \mathfrak{F}_K$, and $\ln \Delta$ is polynomial in the bit complexity of the input.

Corollary 1. *When Algorithm 1 terminates, the returned solution is a nearly-good solution. Also, within $2k \cdot \ln(n\Delta)$ iterations Algorithm 1 will have had some iteration with \mathcal{S} being a nearly-good solution.*

Proof. Lemma 1 shows that if \mathcal{S} is not a nearly-good solution then there is a better solution $\mathcal{S}' \in \mathfrak{F}_k$ with $|\mathcal{S} - \mathcal{S}'| \leq \rho$, so the local search algorithm can only terminate with a nearly-good solution.

For the sake of contradiction, suppose that after $K = \lfloor 2k \cdot \ln(n\Delta) \rfloor$ iterations Algorithm 1 has still not encountered a nearly-good solution. Say $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_K \in \mathfrak{F}_k$ is the sequence of sets held by the algorithm after the first K iterations, where \mathcal{S}_0 is the initial set.

For $0 \leq i < K$, Lemma 1 and the fact that Algorithm 1 always chooses the best improving swap shows $\text{cost}(\mathcal{S}_{i+1}) - \text{cost}(\mathcal{O}) \leq (1 - 1/(2k)) \cdot (\text{cost}(\mathcal{S}_i) - \text{cost}(\mathcal{O}))$. Thus,

$$\text{cost}(\mathcal{S}_K) - \text{cost}(\mathcal{O}) \leq \left(1 - \frac{1}{2k}\right)^K \cdot (\text{cost}(\mathcal{S}_0) - \text{cost}(\mathcal{O})) \leq \left(1 - \frac{1}{2k}\right)^K \cdot n\Delta < 1.$$

Because costs are integral, $\text{cost}(\mathcal{S}_K) - \text{cost}(\mathcal{O}) = 0$ which contradicts that \mathcal{S}_K is not a nearly-good solution. □

This does not yet show that Algorithm 1 terminates in a polynomial number of steps. This fact will be proven after the next subsection when we show the only nearly-good solution in stable instances is \mathcal{O} .

2.3 Nearly-Good Solutions are Optimal in Stable Instances

Our high-level approach is inspired by [18], but we must address larger technical challenges. Roughly speaking, the added difficulty is because the local search analysis from [22, 23] we are following has the bound on the cost change of the swaps depending mildly on $\text{cost}(\mathcal{S})$. We are also burdened with proving that the optimum is found in a polynomial number of iterations, something that was not addressed in [18].

Throughout this section, let \mathcal{S} be a fixed nearly-good solution. Define perturbed distances $\delta'(i, j)$ for $i \in \mathcal{C}, j \in \mathcal{X}$ as follows:

$$\delta'(i, j) = \begin{cases} (1 + \epsilon') \cdot \delta(i, j) & \text{if } i \neq \sigma(j, \mathcal{S}) \\ \delta(i, j) & \text{otherwise} \end{cases}$$

Due to this k -MEANS instance being $(1 + \epsilon')$ -perturbation stable, \mathcal{O} remains the unique optimum solution under these perturbed distances δ' . For any $\mathcal{S}' \in \mathfrak{F}_k$, let $\text{cost}'(\mathcal{S}') = \sum_{j \in \mathcal{X}} \min_{i \in \mathcal{S}'} \delta'(i, j)^2$ be the cost of \mathcal{S}' under distances δ' . Partition the points in \mathcal{X} into the following groups:

- $\mathcal{X}^1 = \{j \in \mathcal{X} : \sigma(j, \mathcal{S}) \in \mathcal{S} - \mathcal{O} \text{ and } \sigma(j, \mathcal{O}) \in \mathcal{S} \cap \mathcal{O}\}$,
- $\mathcal{X}^2 = \{j \in \mathcal{X} : \sigma(j, \mathcal{S}) \in \mathcal{S} \cap \mathcal{O} \text{ and } \sigma(j, \mathcal{O}) \in \mathcal{O} - \mathcal{S}\}$,
- $\mathcal{X}^3 = \{j \in \mathcal{X} : \sigma(j, \mathcal{S}), \sigma(j, \mathcal{O}) \in \mathcal{S} \cap \mathcal{O}\}$, and
- $\mathcal{X}^4 = \{j \in \mathcal{X} : \sigma(j, \mathcal{S}) \in \mathcal{S} - \mathcal{O} \text{ and } \sigma(j, \mathcal{O}) \in \mathcal{O} - \mathcal{S}\}$.

Observe $\mathcal{X}^4 = \overline{\mathcal{X}}_{\mathcal{S}}$ (notation from the previous section) and that $\sigma(j, \mathcal{O}) = \sigma(j, \mathcal{S})$ for $j \in \mathcal{X}^3$.

As in the proof of Theorem 6, let $c_j^* = \delta(j, \sigma(j, \mathcal{O}))^2$ be the clustering cost incurred by point j in the optimum solution and, analogously for \mathcal{S} , $c_j = \delta(j, \sigma(j, \mathcal{S}))^2$. By considering the connection cost of each point on a case-by-case basis, we easily see

$$\text{cost}'(\mathcal{O}) = \sum_{j \in \mathcal{X}^1} (1 + \epsilon')^2 \cdot c_j^* + \sum_{j \in \mathcal{X}^2} \min\{(1 + \epsilon')^2 \cdot c_j^*, c_j\} + \sum_{j \in \mathcal{X}^3} c_j^* + \sum_{j \in \mathcal{X}^4} (1 + \epsilon')^2 \cdot c_j^*. \quad (1)$$

Before putting all pieces together, we make one last observation. As \mathcal{S} is a nearly-good solution, $c_j^* \leq c_j$ for $j \in \mathcal{X}^2$, and $c_j^* = c_j$ for $j \in \mathcal{X}^3$, we have:

$$\begin{aligned} \sum_{j \in \mathcal{X}^4} c_j &\leq \sum_{j \in \mathcal{X}^1} c_j + \sum_{j \in \mathcal{X}^4} c_j = \text{cost}(\mathcal{S}) - \sum_{j \in \mathcal{X}^2} c_j - \sum_{j \in \mathcal{X}^3} c_j \\ &\leq \text{cost}(\mathcal{S}) - \sum_{j \in \mathcal{X}^2} c_j^* - \sum_{j \in \mathcal{X}^3} c_j^* \leq \text{cost}(\mathcal{O}) + 2\epsilon \cdot \Psi(\mathcal{S}) - \sum_{j \in \mathcal{X}^2} c_j^* - \sum_{j \in \mathcal{X}^3} c_j^* \\ &= \sum_{j \in \mathcal{X}^1} c_j^* + \sum_{j \in \mathcal{X}^4} c_j^* + 2\epsilon \left(\sum_{j \in \mathcal{X}^4} c_j^* + c_j \right). \end{aligned}$$

Rearranging,

$$\sum_{j \in \mathcal{X}^4} c_j \leq \frac{1}{1 - 2\epsilon} \left(\sum_{j \in \mathcal{X}^1} c_j^* + (1 + 2\epsilon) \cdot \sum_{j \in \mathcal{X}^4} c_j^* \right) \leq (1 + 6\epsilon) \cdot \left(\sum_{j \in \mathcal{X}^1} c_j^* + \sum_{j \in \mathcal{X}^4} c_j^* \right). \quad (2)$$

Lemma 2. *The nearly-good solution \mathcal{S} is the optimum solution.*

Proof. Using (2), we bound $\text{cost}(\mathcal{S})$ in the following way:

$$\begin{aligned} \text{cost}(\mathcal{S}) &\leq \text{cost}(\mathcal{O}) + 2\epsilon \cdot \Psi(\mathcal{S}) = \text{cost}(\mathcal{O}) + 2\epsilon \sum_{j \in \mathcal{X}^4} c_j^* + 2\epsilon \sum_{j \in \mathcal{X}^4} c_j \\ &\leq \text{cost}(\mathcal{O}) + 2\epsilon \sum_{j \in \mathcal{X}^4} c_j^* + 2\epsilon \cdot (1 + 6\epsilon) \cdot \left(\sum_{j \in \mathcal{X}^1} c_j^* + \sum_{j \in \mathcal{X}^4} c_j^* \right) \\ &\leq \sum_{j \in \mathcal{X}^1} (1 + 6\epsilon) \cdot c_j^* + \sum_{j \in \mathcal{X}^2} c_j^* + \sum_{j \in \mathcal{X}^3} c_j^* + \sum_{j \in \mathcal{X}^4} (1 + 6\epsilon) \cdot c_j^* \\ &\leq \sum_{j \in \mathcal{X}^1} (1 + 6\epsilon) \cdot c_j^* + \sum_{j \in \mathcal{X}^2} \min\{(1 + 6\epsilon) \cdot c_j^*, c_j\} + \sum_{j \in \mathcal{X}^3} c_j^* + \sum_{j \in \mathcal{X}^4} (1 + 6\epsilon) \cdot c_j^*. \quad (3) \end{aligned}$$

The last bound again uses $c_j^* \leq c_j$ for $j \in \mathcal{X}^2$. Recall we chose ϵ so $(1 + 6\epsilon) = (1 + \epsilon')^2$. Thus, combining (1), (3) and the simple observation that $\text{cost}'(\mathcal{S}) = \text{cost}(\mathcal{S})$ we see

$$\text{cost}'(\mathcal{S}) = \text{cost}(\mathcal{S}) \leq \text{cost}'(\mathcal{O}).$$

Finally, because the instance is $(1 + \epsilon')$ -stable with \mathcal{O} being the unique optimum, it remains the unique optimum solution under the perturbed distances δ' . This shows $\mathcal{S} = \mathcal{O}$. \square

We now conclude the proof of our main algorithmic result.

Proof of Theorem 1. By Corollary 1, within a polynomial number of iterations Algorithm 1 will have \mathcal{S} being a nearly-good solution. By Lemma 2, $\mathcal{S} = \mathcal{O}$. Certainly Algorithm 1 will then terminate because there can be no improving swap for an optimal solution. \square

We make the following interesting observation. It states that the local-search algorithm provided earlier, when truncated to a polynomial number of steps, provides a PTAS for arbitrary (not necessarily stable) instances of k -MEANS and will fully solve stable instances whose stability constant is related to the ϵ in the PTAS approximation guarantee.

Consider fixed $0 < \epsilon \leq 1/6$ and fixed dimension d . Let $\rho := \rho(\epsilon, d)$ be as before (the constant in [22, 23]).

Corollary 2. Consider an instance $\mathcal{X} = (\mathcal{X}, \mathcal{C}, \delta)$ of k -MEANS in a metric with doubling dimension d . If we stop the loop of the ρ -swap local search heuristic in Algorithm 1 after $2k \cdot \ln(n\Delta)$ iterations, then if \mathcal{X} is $\sqrt{(1+6\epsilon)}$ -stable the algorithm will find the optimum solution and, otherwise, the returned solution \mathcal{S} satisfies $\text{cost}(\mathcal{S}) \leq (1+6\epsilon) \cdot \text{cost}(\mathcal{O})$.

Proof. We already argued it finds the optimum solution in $(1+\epsilon')$ -stable instances where ϵ' satisfies $(1+\epsilon')^2 = 1+6\epsilon$. From Corollary 1, which did not require the assumption that the instance is stable, within $2k \cdot \ln(n\Delta)$ iterations some set \mathcal{S} considered in the algorithm satisfies $\text{cost}(\mathcal{S}) \leq \text{cost}(\mathcal{O}) + 2\epsilon \cdot \Psi(\mathcal{S})$.

Deriving the bound in (2) also did not rely on stability, so

$$\text{cost}(\mathcal{S}) \leq \text{cost}(\mathcal{O}) + 2\epsilon \cdot \Psi(\mathcal{S}) \leq (1+2\epsilon) \cdot \text{cost}(\mathcal{O}) + 2\epsilon \cdot \text{cost}(\mathcal{S}).$$

Rearranging, $\text{cost}(\mathcal{S}) \leq \frac{1+2\epsilon}{1-2\epsilon} \cdot \text{cost}(\mathcal{O}) \leq (1+6\epsilon) \cdot \text{cost}(\mathcal{O})$. Thus, the final set returned by Algorithm 1 is at most this expensive: at most $(1+6\epsilon) \cdot \text{cost}(\mathcal{O})$. \square

3 Roadmap of the Reduction for Theorem 4

Our overall goal in the remaining sections is to prove Theorem 4. We remark that all of our reductions are deterministic reductions and run in polynomial time. The only randomization in the reduction is in Theorem 3 itself. Indeed, this seems essential given the current understanding of U-QSAT as the only known hardness proofs are by randomized reductions. Ultimately, by composing Theorem 3 with our reductions, we obtain a randomized, polynomial time reduction from every language $L \in NP$ to k -MEANS that has the following property. For every instance I of L we will have computed a value c_I such that the resulting k -MEANS instance \mathcal{I}_I has the following properties depending on whether I is a yes case or a no case.

Yes case: With probability $\geq 1/\text{poly}(|I|)$, \mathcal{I}_I is s -stable for some universal constant s and the optimum solution to \mathcal{I}_I has cost c_I .

No case: Always, the optimum solution cost to \mathcal{I}_I is at least $\gamma \cdot c_I$ for some universal constant $\gamma > 1$.

Given this, if there was an efficient γ' -approximation (for some $\gamma' < \gamma$) for s -stable instances of discrete k -MEANS in \mathbb{R}^d where d is part of the input then we could decide languages in NP in the following way. By repeating the reduction polynomially many times and running the γ' -approximation on each of the resulting k -MEANS instances, with probability $\geq 1/2$ if I was a yes instance then we would find some solution with cost $< \gamma \cdot c_I$ and, always, if I was a no instance then every solution would have cost $\geq \gamma \cdot c_I$. That is, we would have decided L with a randomized, polynomial time algorithm with one-sided error (in the yes case) meaning $NP = RP$.

Starting with Theorem 3, we first reduce Q parameter in S-QSAT-B to 3: that is we provide a stability-preserving reduction from S-QSAT-B to S-3SAT-B². We should point out that the letter B in S-3SAT-B and S-QSAT-B (and other subsequent problems we consider) is to mean bounded occurrence but the exact bound may change as we reduce from one problem to another. We articulate these bounds in our reductions. For instance, in our reduction from an instance Φ of S-QSAT-B with bound B we generate an instance Ψ of S-3SAT-B with bound $B' = \max\{7B, 4BQ^2\}$ on the occurrence of each variable.

Our reduction from S-QSAT-B to S-3SAT-B is a relatively simple reduction, but it serves as a good introduction to the concept of preserving stability. Then we reduce S-3SAT-B to stable instances of 3D-MATCHING for some appropriate concept of stability for this problem. Problem definitions, precise details of what we mean by stability for 3D-MATCHING, and other finer-grained details we need to preserve will be discussed later. Finally, we reduce stable 3D-MATCHING instances to stable k -MEANS instances to complete our proof.

4 A Stability-Preserving Reduction From S-QSAT-B to S-3SAT-B

Our first step is to show hardness of S-3SAT-B. There are standard reductions from QSAT-B to 3SAT-B and, if Q is regarded as a constant, the most commonly-taught reduction is also an L -reduction. But more is

²Paradise [31] also presents a reduction from S-QSAT-B to the variant of S-3SAT-B where every clause involves at most 3 variables. But we need each clause to depend on exactly 3 distinct variables so we will present our approach here. We call it stable E3SAT-B

needed to preserve stability, likely the simple reduction the reader has in mind is not even parsimonious. While the reduction in this section is still quite simple, it serves as a warm-up to the concept of preserving stability in a reduction and it is a necessary technical step toward our final goal. This is the first step of proof of hardness of stable instances of k -MEANS.

Let B, Q, s, ϵ be constants from Theorem 3. Let Φ be an instance of QSAT-B; we construct an instance Ψ of E3SAT-B (3SAT-B with exactly 3 literals per clause) for some bound B' on the number of occurrences of each variable which will depend only on Q and B . Properties of this reduction, including how it preserves stability, will be proven below. Our reduction actually produces an instance of S-QSAT-B with *exactly* 3 literals per clause, hence it is actually a reduction to stable instances of E3SAT-B and we require addressing clauses of size < 3 of the S-QSAT-B instance in the reduction.

Say Φ has variables X and clauses \mathcal{C} where each clause $C \in \mathcal{C}$ is viewed as a set of literals over distinct variables of X . We may depict a clause as, say, $x \vee \bar{y} \vee z$. Before describing the reduction, consider the following gadget. For literals ℓ_1, ℓ_2, ℓ_3 let $F(\ell_1, \ell_2, \ell_3)$ be the following collection of seven 3CNF clauses, applying the reduction $\bar{\bar{x}} \rightarrow x$ for any doubly-negated variable:

$$\begin{aligned} & \bar{\ell}_1 \vee \ell_2 \vee \ell_3, & \bar{\ell}_1 \vee \ell_2 \vee \bar{\ell}_3, & \bar{\ell}_1 \vee \bar{\ell}_2 \vee \ell_3, & \bar{\ell}_1 \vee \bar{\ell}_2 \vee \bar{\ell}_3, \\ & \ell_1 \vee \ell_2 \vee \ell_3, & \ell_1 \vee \bar{\ell}_2 \vee \bar{\ell}_3, & \text{and } \ell_1 \vee \ell_2 \vee \bar{\ell}_3. \end{aligned}$$

One can easily check that the only way to satisfy all clauses in $F(\ell_1, \ell_2, \ell_3)$ is for all literals to be FALSE. So $F(\ell_1, \ell_2, \ell_3)$ enforces that all these three literals have to be false.

Our instance Ψ of 3SAT-B is constructed as follows. The variables of Ψ consist of X and a collection of new variables Y we introduce below as we describe the clauses. For each $C \in \mathcal{C}$, say $C = \{\ell_C^1, \dots, \ell_C^{|C|}\}$. We introduce some new variables Y_C and clauses \mathcal{C}'_C for Ψ .

- **Case** $|C| = 1$. Let $Y_C = \{y_C, z_C\}$ and $\mathcal{C}'_C = F(\bar{\ell}_C^1, y_C, z_C)$.
- **Case** $|C| = 2$. Let $Y_C = \{w_C, y_C, z_C\}$ and $\mathcal{C}'_C = \{\ell_C^1 \vee \ell_C^2 \vee w_C\} \cup F(w_C, y_C, z_C)$.
- **Case** $|C| = 3$. Let $Y_C = \emptyset$ and $\mathcal{C}'_C = \{C\}$.
- **Case** $|C| \geq 4$. Let $Y_C = \{y_C^0, \dots, y_C^{|C|}, z_C\}$ and \mathcal{C}'_C be comprised of the following constraints:
 1. $y_C^{i-1} \vee \ell_C^i \vee \bar{y}_C^i$ for each $1 \leq i \leq |C|$,
 2. $\bar{y}_C^{i-1} \vee y_C^i \vee z_C$ for each $1 \leq i \leq |C|$,
 3. $\bar{\ell}_C^i \vee y_C^{j-1} \vee z_C$ for each pair $1 \leq i < j \leq |C|$,
 4. and $F(y_C^0, \bar{y}_C^{|C|}, z_C)$.

Note the coarse upper bound of $|\mathcal{C}'_C| \leq 3Q^2$ and $|Y_C| \leq 2Q$ holds in each case.

Finally, the variables of Ψ are $X' := X \cup Y$ where $Y = \bigcup_{C \in \mathcal{C}} Y_C$ and the constraints of Ψ are $\mathcal{C}' := \bigcup_{C \in \mathcal{C}} \mathcal{C}'_C$. By construction, each clause in Ψ has exactly three literals over distinct variables. Also, each variable of Ψ appears in at most $B' = \max\{7B, 4BQ^2\}$ clauses of Ψ . The cases with $|C| = 1$ or 2 are simply padding gadgets to get exactly 3 literals per clause. For Case 4, the constraints are to ensure that $z_C = \text{FALSE}$, variables $y_C^0, \dots, y_C^{|C|}$ are monotonic: set (i.e FALSE, FALSE, FALSE, \dots , TRUE, TRUE) with one switch from FALSE to TRUE, and that the switch from FALSE to TRUE appears at the first index i where ℓ_i is TRUE. This property is proven during the proof of Claim 1.

Let $n = |X|, m = |\mathcal{C}|, n' = |X'|$ and $m' = |\mathcal{C}'|$. Note,

$$n' = n + \sum_{C \in \mathcal{C}} |Y_C| \leq n + (Q+1) \cdot m \leq n + (Q+1)B \cdot n \leq 2QB \cdot n \quad \text{and} \quad m' = \sum_{C \in \mathcal{C}} |\mathcal{C}'_C| \leq 3Q^2 \cdot m.$$

The following two claims are straightforward.

Claim 1. *Any satisfying assignment for Φ can be extended uniquely to a satisfying assignment for Ψ . Conversely, the restriction of any satisfying assignment for Ψ to variables in X is a satisfying assignment for Φ .*

Proof. First, consider a satisfying assignment $\phi : X \rightarrow \{\text{TRUE}, \text{FALSE}\}$ for Φ . For each clause $C \in \mathcal{C}$, we claim there is a unique way to assign values to Y_C to satisfy all clauses in \mathcal{C}'_C . This is simple to verify when $|C| \leq 3$, recalling the only way to satisfy all clauses in $F(u, v, w)$ is for the three literals to be false.

Suppose $|C| \geq 4$. We know one of the literals $\ell_C^1, \dots, \ell_C^{|C|}$ is true, let i be the least-indexed literal that is true. Set y_C^0, \dots, y_C^{i-1} to FALSE, $y_C^i, \dots, y_C^{|C|}$ to TRUE, and z_C to FALSE. This satisfies all clauses in \mathcal{C}'_C . We also claim this is the only way to satisfy all clauses.

To see this, note $F(y_C^0, \overline{y_C^{|C|}}, z_C)$ forces $y_C^0 = z_C = \text{FALSE}$ and $y_C^{|C|} = \text{TRUE}$. Then the second set of constraints ensure the y_C -variables are “monotone”: that there exists some $1 \leq i' \leq |C|$ with $y_C^j = \text{FALSE}$ for $j < i'$ and $y_C^j = \text{TRUE}$ for $j \geq i'$.

It cannot be that $i' < i$, otherwise $y_C^{i'-1} \vee \ell_C^{i'} \vee \overline{y_C^{i'}}$ is FALSE (recall $\ell_C^{i'}$ is FALSE for $i' < i$). It also cannot be that $i' > i$, otherwise $\overline{\ell_C^i} \vee y_C^{i'-1} \vee z_C$ is FALSE. Thus, $i' = i$ meaning the assignment described above is the only way to extend the satisfying assignment for Φ to one that satisfies \mathcal{C}'_C .

The argument essentially reverses. Consider a satisfying assignment $\psi : X' \rightarrow \{\text{TRUE}, \text{FALSE}\}$ for Ψ and consider a clause $C \in \mathcal{C}$ of Φ . If $|C| \leq 3$, it is easy to see the restriction of the assignment for Ψ to X satisfies C itself. So suppose $|C| \geq 4$. As argued before, we must have both y_C^0 and z_C being FALSE and $y_C^{|C|}$ being TRUE. Consider the least index $1 \leq i \leq |C|$ with y_C^{i-1} being FALSE and y_C^i being TRUE. Then ℓ_C^i is TRUE, so the original clause C of Φ is satisfied. \square

Claim 2. *Suppose at most $(1 - \gamma) \cdot m$ clauses of Φ can be satisfied by any assignment for some $\gamma \geq 0$. Then at most $\left(1 - \frac{\gamma}{3Q^2}\right) \cdot m'$ clauses of Ψ can be satisfied by any assignment.*

Proof. Consider a truth assignment ψ for Ψ and let ϕ be its restriction to X . By Claim 1, any clause $C \in \mathcal{C}$ of Φ is satisfied by ϕ if all corresponding clauses in \mathcal{C}'_C are satisfied by ψ . This can happen for at most $(1 - \gamma) \cdot m$ clauses $C \in \mathcal{C}$. So at least $\gamma \cdot m$ clauses $C \in \mathcal{C}$ have at least one corresponding clause in \mathcal{C}'_C not being satisfied. Overall, the number of unsatisfied clauses in Ψ is at least $\gamma \cdot m \geq \frac{\gamma}{3Q^2} \cdot m'$. \square

Finally, we show the reduction preserves stability. One might be tempted to think the analysis will be very similar to the **no** case analysis. However, as stated in the introduction, we will later give an example showing these concepts are fundamentally different by showing a well-known parsimonious L -reduction that does not preserve stability. This example is found in Appendix B.

We have not attempted to optimize constants in our analysis below, but dependence on B and Q seems essential. The following shows that the stability drops only by at most a constant factor, assuming Φ has bounded clause size and bounded occurrence for each variable.

Theorem 7. *Suppose Φ is s -stable. Then Ψ is s' -stable where $s' = \frac{s}{8BB'Q^2}$.*

Proof. Because Φ has a unique satisfying assignment x^* , then by Claim 1 there is some y^* (assignment of values to variables in Y) such that (x^*, y^*) is the unique satisfying assignment for Ψ . Consider any truth assignment (x', y') for Ψ , we show the fraction of unsatisfied clauses in Ψ is at least $s' \cdot HW((x^*, y^*), (x', y'))$.

Let $h = HW((x^*, y^*), (x', y'))$, $h_x = HW(x^*, x')$, and $h_y = HW(y^*, y')$. Observe $|X| \cdot h_x + |Y| \cdot h_y = |X'| \cdot h$. Consider the following two cases.

Case 1: $|X| \cdot h_x \geq |X'| \cdot h/2$.

Because Φ is s -stable, then x' leaves at least $m \cdot s \cdot h_x$ clauses of Φ unsatisfied. For each unsatisfied clause C of Φ , at least one clause of Ψ in \mathcal{C}'_C is not satisfied by (x', y') . So at least $m \cdot s \cdot h_x$ clauses of Ψ are also not satisfied. Note the following:

- $|X| \leq Q \cdot m$, because each clause in Φ has at most Q literals.
- $B' \cdot |X'| \geq m'$, because each variable of Ψ appears in at most B' clauses.

The number of unsatisfied clauses of Ψ is then at least

$$m \cdot s \cdot h_x \geq \frac{s|X|}{Q} \cdot h_x \geq \frac{s|X'|}{2Q} \cdot h \geq \frac{s}{2B'Q} \cdot h \cdot m'.$$

Case 2: $|X| \cdot h_x < |X'| \cdot h/2$, equivalently $|Y| \cdot h_y > |X'| \cdot h/2$.

Let $\mathcal{C}_{bad} \subseteq \mathcal{C}$ be the clauses C of Φ such that at least one variable in Y_C is different between y^* and y' . The number of y -variables that differ between y^* and y' is $|Y| \cdot h_y$ and $|Y_C| \leq 2Q$ for each $C \in \mathcal{C}$, so

$$|\mathcal{C}_{bad}| \geq \frac{|Y|}{2Q} \cdot h_y. \quad (4)$$

Partition \mathcal{C}_{bad} into two groups:

- \mathcal{C}_{bad}^1 : Clauses $C_i \in \mathcal{C}_{bad}$ such that at least one clause of Ψ in \mathcal{C}'_i is not satisfied by (x', y') .
- \mathcal{C}_{bad}^2 : Clauses $C_i \in \mathcal{C}_{bad}$ such that all clauses in \mathcal{C}'_i are satisfied by (x', y') .

Subcase 2.1: $|\mathcal{C}_{bad}^1| \geq |\mathcal{C}_{bad}|/2$.

Then the number of clauses not satisfied by (x', y') can be bounded from below as follows:

$$|\mathcal{C}_{bad}^1| \geq \frac{|\mathcal{C}_{bad}|}{2} \geq \frac{|Y|}{4Q} \cdot h_y \geq \frac{|X'|}{8Q} \cdot h \geq \frac{h}{8B'Q} \cdot m',$$

where the 2nd inequality follows from (4).

Subcase 2.2: $|\mathcal{C}_{bad}^2| > |\mathcal{C}_{bad}|/2$.

Consider some $C \in \mathcal{C}_{bad}^2$. All constraints in \mathcal{C}'_C are satisfied yet y^* disagrees with y' on Y_C , so it must be that $|C| \geq 4$ because, by construction of \mathcal{C}'_C , the only way to satisfy all clauses in \mathcal{C}'_C for $|C| \leq 3$ has all variables in Y_C being FALSE.

By construction of \mathcal{C}'_C in the case $|C| \geq 4$, the fact that all clauses are satisfied means there is a unique index $1 \leq i \leq |C|$ with $y'^i_C = \text{TRUE}$ and $y'^{i-1}_C = \text{FALSE}$. There is also a unique index j with $y^{*j}_C = \text{TRUE}$ and $y^{*j-1}_C = \text{FALSE}$. But because y^* and y' disagree on Y_C , it must be that $i \neq j$. Observe, then that $\ell_C^{\min\{i,j\}}$ has different values under x^* and x' . That is, there is some variable x_ℓ appearing in C where $x_\ell^* \neq x'_\ell$.

The fact that each $C \in \mathcal{C}_{bad}^2$ witnesses at least one such variable x_ℓ with $x_\ell^* \neq x'_\ell$ and the fact that each variable of Φ appears in at most B clauses means there are at least $|\mathcal{C}_{bad}^2|/B$ variables $x_\ell \in X$ with $x_\ell^* \neq x'_\ell$. That is:

$$|X| \cdot h_x \geq \frac{|\mathcal{C}_{bad}^2|}{B} \geq \frac{|\mathcal{C}_{bad}|}{2B} \geq \frac{|Y|}{4BQ} \cdot h_y \geq \frac{|X'|}{8BQ} \cdot h \geq \frac{h}{8BB'Q} \cdot m'.$$

The fact that Φ is s -stable means at least $s \cdot h_x \cdot m$ clauses are not satisfied by x' . As before, for each clause $C \in \mathcal{C}$ not satisfied by x' there is at least one clause in the group of clauses \mathcal{C}'_C of Ψ that is not satisfied by (x', y') . Thus, the number of clauses of Ψ that are not satisfied by (x', y') is at least

$$s \cdot h_x \cdot m \geq s \cdot \frac{|X|}{Q} \cdot h_x \geq \frac{s}{8BB'Q^2} \cdot h \cdot m',$$

as required. □

5 Reduction From STABLE 3SAT-B to STABLE 3DM-B

In this section we show how to reduce stable instances of bounded 3SAT (S-3SAT-B) to stable instances of bounded occurrence 3D matching. We formally define the problems we consider in our reduction below.

Definition 6 (UNAMBIGUOUS 3DM-B or U-3DM-B). *An instance of U-3DM-B problem is given via a hypergraph $G = (V_1 \cup V_2 \cup V_3, \mathcal{T})$ where, for $i = 1, 2, 3$, $|V_i| = n$, and each triple $t \in \mathcal{T}$ is of the form (v_1, v_2, v_3) with $v_i \in V_i$. In the decision version of the problem, the task is to decide whether a perfect matching, i.e., a subset $T^* \subseteq \mathcal{T}$ of n disjoint triples, exists given the guarantees that:*

1. Each vertex v appears in at most B triples, where B is a constant.
2. There is at most one perfect matching.

Throughout this section, for sets of triples T^* and T , the Hamming weight function $HW(T^*, T)$ is defined as $\frac{|T^* \Delta T|}{2|T^*|}$ for a nonempty set T^* .

Definition 7 (STABLE 3DM-B or S-3DM-B). *An instance of the S-3DM-B problem is an instance of the U-3DM-B problem that is (s, γ) -stable for $0 < s, \gamma < 1$, in the sense that it has the following guarantees.*

1. *If a perfect matching T^* exists, any subset of disjoint triples T has a size at most $(1 - s \cdot HW(T^*, T))n$, where $HW(T^*, T)$ is the fraction of the triples on which T^* and T disagree.*
2. *If no perfect matchings exist, then any subset $T \subseteq \mathcal{T}$ of disjoint triples has $|T| \leq (1 - \gamma) \cdot n$.*

Definition 8 (UNAMBIGUOUS COVERING BY TRIPLES-B). *An instance of the U-CBT-B problem is given via the same hypergraph as in the U-3DM-B problem. In the decision version of the problem, the task is to decide whether a subset $T^* \subseteq \mathcal{T}$ of size n that covers all the $3n$ vertices exists given the guarantees that:*

1. *Each vertex v appears in at most B triples, where B is a constant.*
2. *There is at most one set of triples solution covering all the vertices.*

Furthermore, if an instance \mathcal{I} of U-CBT-B has a solution that covers all the nodes, we call it a covering instance.

Definition 9 (STABLE COVERING BY TRIPLES-B). *An instance of the S-CBT-B problem is an instance of the U-CBT-B problem that is (s, γ) -stable for $0 < s, \gamma < 1$, in the sense that it has the following guarantees.*

1. *If a unique covering solution T^* exists, any subset of n triples T fails to cover at least an $s \cdot HW(T^*, T)$ fraction of the $3n$ vertices.*
2. *If no covering solutions exist, then any subset $T \subseteq \mathcal{T}$ of triples, $|T| = n$, covers at most $(1 - \gamma)$ fraction of the points.*

Here we show that the hardness of the S-3SAT-B problem implies hardness for S-3DM-B.

Theorem 8. *There exists a polynomial reduction transforming any instance Ψ of S-3SAT-B with n variables and m clauses where each variable appears in at most B clauses to an instance \mathcal{I} of S-3DM-B with $36 \cdot m$ points and $34 \cdot m$ triples where each point appears in at most 7 triples with the following properties*

1. **Yes Case:** *if Ψ is a satisfiable instance of S-3SAT-B, then there exists a unique set of $12 \cdot m$ disjoint triples T^* called the perfect matching. Furthermore, for any disjoint set of triples T , it is true that $|T| \leq (1 - s_1 \cdot HW(T^*, T)) \cdot |T^*|$.*
2. **No Case:** *if Ψ is not satisfiable, then every set of disjoint triples has a size at most $(1 - \gamma_1) \cdot 12 \cdot m$,*

where s_1 and γ_1 are universal constants satisfying $0 < s_1, \gamma_1 < 1$ depending only on B and the stability constant s' from Theorem 7.

We note that $\frac{n}{3} \leq m \leq \frac{B}{3} \cdot n$ holds for any 3SAT instance where each clause contains exactly three different variables, this will be used frequently in the analysis of our reduction.

5.1 The Reduction

The reduction is inspired by known reductions (eg. [24]), but we change the clause gadget to make the reduction parsimonious. Throughout, we call a subset of triples T a *packing* if the triples in T are pairwise-disjoint. The reduction uses two types of gadgets that are described below.

Variable Gadgets

This is a standard construction. For each variable x_i of Ψ , we create a “gear”, depicted in Figure 1. That is, we introduce *inner points* $u_i[b], w_i[b]$ and *tip points* $v_i[b], \bar{v}_i[b]$ for each $1 \leq b \leq \beta_i$ where $1 \leq \beta_i \leq B$ is the number of clauses of Ψ that include variable x_i . The triples in this gadget are:

$$G_i := \{ \{v_i[b], u_i[b], w_i[b]\} : 1 \leq b \leq \beta_i \} \cup \{ \{\bar{v}_i[b], w_i[b], u_i[b+1]\} : 1 \leq b \leq \beta_i \}$$

where we have used wraparound indexing (i.e. $u_i[\beta_i + 1]$ is $u_i[1]$). Note $|G_i| = 2 \cdot \beta_j$.

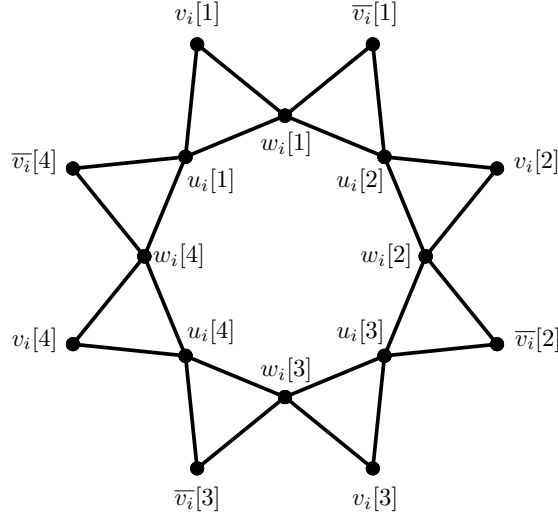


Figure 1: The variable gadget for a variable x_i with $\beta_i = 4$.

Intuitively, setting x_i to **True** corresponds to selecting all triples that include the points $\bar{v}_i[b]$, $1 \leq b \leq \beta_i$ and setting x_i to **False** corresponds to selecting all triples that include the points $v_i[b]$, $1 \leq b \leq \beta_i$. That is, the points that are not covered by such a packing of triples corresponds to the truth assignment being modelled.

Claim 3. *Let $T \subseteq G_i$ be any packing of triples. Then $|T| \leq \beta_i$ and $|T| = \beta_i$ if and only if $T = \{\{v_i[b], u_i[b], w_i[b]\} : 1 \leq b \leq \beta_i\}$ or $T = \{\{\bar{v}_i[b], w_i[b], u_i[b + 1]\} : 1 \leq b \leq \beta_i\}$.*

Proof. First, note that since each $t \in G_i$ contains two inner points and there are $2 \cdot \beta_i$ inner points, then no packing $T \subset G_i$ has size more than β_i . It is also clear that any packing of size exactly β_i consists of alternating triples around the gear, i.e. it is one of the two packings from the statement of the claim. \square

Clause Gadgets

Consider each clause $C = \ell_i \vee \ell_j \vee \ell_k$ involving variables x_i, x_j, x_k . Also let b_i be such that C is the b_i 'th clause containing x_i according to an arbitrary but fixed ordering of the occurrences of each variable. Similarly define b_j and b_k for x_j and x_k , respectively. Let S_C be the set of ways to assign values to the variables x_i, x_j, x_k to satisfy C , so $|S_C| = 7$.

We create 24 new points for the clause gadget. Three of them are *literal points* we call $y_i[b_i], y_j[b_j], y_k[b_k]$. Then, for $\alpha \in S_C$ we create three new vertices that we call *control points*: call these $z_i[\alpha], z_j[\alpha]$ and $z_k[\alpha]$.

There are 28 triples in this clause gadget that involve the literal points and control points for this gadget as well as the 6 tip points $v_i[b_i], \bar{v}_i[b_i], v_j[b_j], \bar{v}_j[b_j], v_k[b_k], \bar{v}_k[b_k]$ from the variable gadgets corresponding to this particular occurrence of each variable in C .

For each $\alpha \in S_C$ let $v(\alpha, i)$ be the tip point $v_i[b_i]$ or $\bar{v}_i[b_i]$ corresponding to the truth assignment of x_i under α . That is, if α assigns **True** to x_i then let $v(\alpha, i) = v_i[b_i]$ otherwise let $v(\alpha, i) = \bar{v}_i[b_i]$.

$$G_C := \{\{z_i[\alpha], z_j[\alpha], z_k[\alpha]\} : \alpha \in S_C\} \cup \{\{v(\alpha, \ell), z_\ell[\alpha], y_\ell[b_\ell]\} : \alpha \in S_C, \ell \in \{i, j, k\}\}.$$

A portion of this construction is depicted in Figure 2. The top layer of points are the control points for this particular α and do not appear in any other triples in the clause gadget (i.e. triples for different $\alpha' \neq \alpha$). The middle layer of points are the literal points: these appear in the other vertically-drawn triples for other portions of the clause gadget (i.e. for other $\alpha' \in S_C$). The bottom points are tip points from variable gadgets that

correspond to this satisfying assignment: by leaving all three tips uncovered when we choose the triples from the corresponding variable gadgets, we are indicating to this clause that the variables have truth assignment α so we may pick the three vertically-drawn triples and cover all points in the figure.

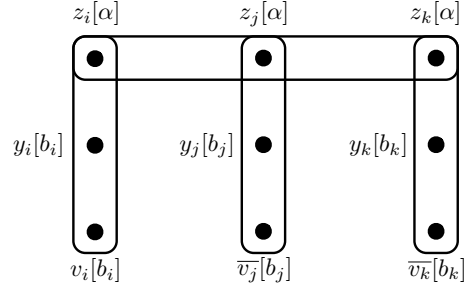


Figure 2: Example of part of a clause gadget for a clause $C = \overline{x_i} \vee x_j \vee \overline{x_k}$ where b_i, b_j and b_k denote which occurrence of the corresponding variable appears in C . Here, α is the assignment $x_i = \text{True}, x_j = \text{False}, x_k = \text{False}$ which satisfies C . In particular, the tips of the gears that appear in this figure correspond to α , not necessarily to the sign of the original variable in C . The rounded rectangles indicate the four triples in this gadget associated with α .

Important properties of this gadget are summarized in Claim 4.

Claim 4. *Let $T \subseteq G_C$ be a packing. Then $|T| \leq 9$. Furthermore, if $|T| = 9$ then for some $\alpha^* \in S_C$ we have*

$$T = \{\{z_i[\alpha], z_j[\alpha], z_k[\alpha]\} : \alpha \in S_C - \{\alpha^*\}\} \cup \{\{v(\alpha^*, \ell), z_\ell[\alpha^*], y_\ell[b_\ell]\} : \ell \in \{i, j, k\}\}.$$

Intuitively, if $|T| = 9$ then the triples that cover the literal points all correspond to the same satisfying assignment α^* .

Proof. If T contains all seven triples that cover only control points (i.e. all triples in $\{\{z_i[\alpha], z_j[\alpha], z_k[\alpha]\} : \alpha \in S_C\}$) then it cannot contain any other triple in G_C since they all include a control point. If not, notice still that T can only contain at most three triples that cover a literal point since there are only three literal points. In either case, $|T| \leq 9$.

Now suppose $|T| = 9$. Then T contains exactly three triples that cover a literal point and exactly six triples that only cover control points. Let $\alpha^* \in S_C$ be the satisfying assignment such that $\{z_i[\alpha^*], z_j[\alpha^*], z_k[\alpha^*]\} \notin T$. Then all triples covering a literal point must be of the form $\{v(\alpha^*, \ell), z_\ell[\alpha^*], y_\ell[b_\ell]\}$ for the three choices $\ell \in \{i, j, k\}$, as required, since all other triples covering a literal point share a control point in common with the triples covering only control points. \square

This completes the construction. Notice the number of points in the resulting instance is:

$$\sum_{i=1}^n 4 \cdot \beta_i + \sum_C 24 = 12 \cdot m + 24 \cdot m = 36 \cdot m$$

and the number of triples is:

$$\sum_{i=1}^n 2 \cdot \beta_i + \sum_C 28 = 6 \cdot m + 28 \cdot m = 34 \cdot m$$

where we have used the fact that $\sum_{i=1}^n \beta_i = 3 \cdot m$ as each clause involves precisely three distinct variables. Finally, it is easy to tell by inspection that every point lies in at most 7 triples: each inner point is in exactly 2 triples, tip points $v_i[b]$ and $\overline{v}_i[b]$ each lie in 1 triple from the variable gadget and at most 4 triples from the clause gadget corresponding to b 'th occurrence of x_i , each control point is in exactly 2 triples, and each literal point is in exactly 7 triples. So this is an instance of S-3DM-B where each point appears in at most 7 triples.

5.2 Completeness Analysis

The Canonical Packing T^*

Here we suppose Ψ has a unique satisfying assignment x^* and every truth assignment x' does not satisfy at least $s' \cdot HW(x^*, x') \cdot m$ clauses of Ψ where $0 < s' < 1$ is an absolute constant. We will show there is a single packing T^* of size $12 \cdot m$ (i.e. it covers all points) and any other packing T has $|T| \leq (1 - s_1 \cdot HW(T^*, T)) \cdot |T^*|$ for some absolute constant $0 < s_1 < 1$.

The construction of T^* is straightforward. For each i , if $x_i^* = \text{True}$ then include all triples $\{\bar{v}_i[b], w_i[b], u_i[b+1]\}, 1 \leq b \leq \beta_i$ in T^* and if $x_i^* = \text{False}$ then include all triples $\{v_i[b], u_i[b], w_i[b], 1 \leq b \leq \beta_i\}$ in T^* . The only points in the variable gadgets that are not yet covered are those that correspond to the truth values of their corresponding variables.

Next consider a clause C . As in the construction of the clause gadget, say x_i, x_j, x_k are the variables involved in C and that, respectively, b_i, b_j, b_k denotes which occurrence of the variable lies in C . Let α^* be the truth assignment to x_i, x_j, x_k given by x^* . Add the triples covering control points for $\alpha \neq \alpha^*$ to T , i.e. each of $\{z_i[\alpha], z_j[\alpha], z_k[\alpha]\}$ for $\alpha \in S_C - \{\alpha^*\}$. Finally, add the three triples covering literal points corresponding to α^* , namely $\{v(\alpha^*, \ell), z_\ell[\alpha^*], y_\ell[b_\ell]\}$.

By construction, every point in this S-3DM-B instance is covered by exactly one triple in T^* .

Stability Analysis

Next, consider some packing T of triples. We show $|T| \leq (1 - s_1 \cdot HW(T^*, T)) \cdot |T^*|$ for some absolute constant s_1 to complete the stability analysis. We classify variables and clauses as follows.

Consider a variable x_i .

- Call x_i **good** if $|T \cap G_i| = \beta_i$. From Claim 3, this means $T \cap G_i$ naturally corresponds to a truth assignment for x_i (i.e. the uncovered tips correspond to the value of x_i).
- Call x_i **bad** otherwise, so $|T \cap G_i| < \beta_i$.

Let x' be the following truth assignment. Let x'_i correspond to the truth assignment given by T (i.e. corresponding to the uncovered tips) if x_i is a good variable, otherwise let $x'_i = x_i^*$. Let ζ_B the fraction of bad variables, i.e. the number of bad variables is $\zeta_B \cdot n$.

Now consider a clause C .

- Call C **bad** if at least one of the three variables involved in C is bad.
- Call C **good-unsatisfied** if all three variables of C are good, yet x' does not satisfy C .
- Call C **good-satisfied** if all three variables of C are good and x' satisfies C .

Next, we claim it suffices to assume that if C is good-satisfied then we can assume that $|T \cap G_C| = 9$, i.e. the packing corresponds to a particular truth assignment in the canonical way described in Claim 4.

Lemma 3. *Suppose for some constant $0 < s_1 \leq \frac{2}{17}$ we have the following: For every packing T' with $|T' \cap G_C| = 9$ for each good-satisfied clause C (here good-satisfied is with respect to T') we have $|T'| \leq (1 - s_1 \cdot HW(T^*, T')) \cdot |T^*|$. Then $|T| \leq (1 - s_1 \cdot HW(T^*, T)) \cdot |T^*|$ for every packing T .*

Proof. By induction on the number of clauses C such that $|T' \cap G_C| < 9$ for each good-satisfied clauses. The base case where there is no such clause C is in fact our assumption.

Inductively, suppose C is a good-satisfied clause yet $|T' \cap G_C| < 9$. Since all variables in C are good and their corresponding values satisfy C , we can replace the triples in $T' \cap G_C$ with the 9 triples corresponding to this satisfying truth assignment α for C . Call this new packing T'' . Notice $|T^* \Delta T'| \leq |T^* \Delta T''| + 17$ since we removed at most 8 triples and then added exactly 9 triples to form T'' . By induction and recalling

$HW(T^*, T) = |T^* \Delta T| / (2|T^*|)$, we see

$$\begin{aligned}
|T'| &\leq |T''| - 1 \\
&\leq \left(1 - s_1 \cdot \frac{|T^* \Delta T''|}{2 \cdot |T^*|}\right) \cdot |T^*| - 1 \\
&\leq \left(1 - s_1 \cdot \frac{|T^* \Delta T'| - 17}{2 \cdot |T^*|}\right) \cdot |T^*| - 1 \\
&= (1 - s_1 \cdot HW(T^*, T')) \cdot |T^*| + s_1 \cdot \frac{17}{2} - 1 \\
&\leq (1 - s_1 \cdot HW(T^*, T')) \cdot |T^*|
\end{aligned}$$

where the last bound follows by our assumption that $s_1 \leq 2/17$. \square

Our goal is to prove the “base case” of Lemma 3 and articulate the constant s_1 . We will now assume $|T \cap G_C| = 9$ for every good-satisfied clause C . Continuing, we consider two cases. We have not attempted to “balance” the analysis between the two cases to optimize the resulting stability constant s_1 .

In either case we use the observation that:

$$|T| = \sum_{i=1}^n |T \cap G_i| + \sum_C |T \cap G_C| \leq \sum_{i=1}^n \beta_i + \sum_C 9 = 3m + 9m = 12m = |T^*|.$$

We will observe that throughout these cases, either T is deficient in enough variable gadgets (i.e. $|T \cap G_i| \leq \beta_i - 1$ for enough $1 \leq i \leq n$) or deficient enough in clause gadgets (i.e. $|T \cap G_C| \leq 8$ for enough clauses C), where “enough” means the total deficiency is proportional $HW(T^*, T) \cdot |T^*|$.

- **Case:** $\zeta_B > \frac{s'}{30 \cdot B^2} \cdot HW(T^*, T)$ where, recall, $0 < s' < 1$ is the stability constant for Ψ .

For each bad variable x_i , $|T \cap G_i| \leq \beta_i - 1$. So the number of variable gadgets for which T has strictly fewer than the maximum possible number of triples is at least

$$\zeta_B \cdot n > \frac{s'}{30 \cdot B^2} \cdot HW(T^*, T) \cdot n \geq \frac{s'}{10 \cdot B^3} \cdot HW(T^*, T) \cdot m = \frac{s'}{120 \cdot B^3} \cdot HW(T^*, T) \cdot |T^*|.$$

where we have used $B \cdot n \geq 3 \cdot m$ and $|T^*| = 12 \cdot m$.

That is,

$$|T| \leq \left(1 - \frac{s'}{120 \cdot B^3} \cdot HW(T^*, T)\right) \cdot |T^*|.$$

- **Case:** $\zeta_B \leq \frac{s'}{30 \cdot B^2} \cdot HW(T^*, T)$.

We begin by showing $HW(x^*, x')$ is at least proportional to $HW(T^*, T)$ in this case.

Claim 5. *If $\zeta_B \leq \frac{s'}{30 \cdot B^2} \cdot HW(T^*, T)$, then $HW(x^*, x') \geq \frac{1}{5 \cdot B} \cdot HW(T^*, T)$.*

Proof. Suppose otherwise. We count $|T^* \Delta T|$ by considering different gadgets.

- Triples from bad variable gadgets. Each of T and T^* can have at most B triples from any variable gadget, so at most $2B \cdot \zeta_B \cdot n$ triples come from bad variable gadgets.
- Triples from bad clause gadgets. Each of T and T^* can have at most 9 triples from any clause gadget and there are at most $B \cdot \zeta_B \cdot n$ bad clauses (as each variable participates in at most B clauses), so at most $18B \cdot \zeta_B \cdot n$ triples from from bad clause gadgets.
- Triples from good variable gadgets for a variable x_i with $x'_i \neq x_i^*$. The number of such variables is exactly $HW(x^*, x') \cdot n$ by how we constructed x' and each of T and T^* can have at most B triples from variable gadgets, so at most $2B \cdot HW(x^*, x') \cdot n$ triples come from variable gadgets for good variables with $x'_i \neq x_i^*$.

- Triples from clause gadgets that are not bad but contain a variable x_i with $x'_i \neq x_i^*$. The number of such clauses is at most $B \cdot HW(x^*, x') \cdot n$ since there are $HW(x^*, x') \cdot n$ good variables that disagree on their assignment between x^* and x' , and each variable appears in at most B clauses. Each of T and T^* has at most 9 triples from any clause gadget, so at most $18 \cdot B \cdot HW(x^*, x') \cdot n$ triples come from clause gadgets for clauses that are not bad but contain a variable x_i with $x'_i \neq x_i^*$.

Recalling our assumption that $|T \cap T'_C| = 9$ for every good-satisfied clause C and that x^* satisfies every clause, we see that T^* and T agree on every other gadget. So we have accounted for all triples in $|T^* \Delta T|$. From these four cases and using $n \leq 3 \cdot m$, $s' < 1$ and $|T^*| = 12 \cdot m$, we see

$$\begin{aligned}
|T^* \Delta T| &\leq 20B \cdot \zeta_B \cdot n + 20B \cdot HW(x^*, x') \cdot n \\
&\leq 60B \cdot \zeta_B \cdot m + 60B \cdot HW(x^*, x') \cdot m \\
&= 5B \cdot \zeta_B \cdot |T^*| + 5B \cdot HW(x^*, x') \cdot |T^*| \\
&\leq \left(\frac{1}{6B} + 1 \right) \cdot HW(T^*, T) \cdot |T^*|
\end{aligned}$$

This contradicts $HW(T^*, T) = |T^* \Delta T| / (2|T^*|)$ which is true by definition. \square

Thus, we have $HW(x^*, x') \geq \frac{1}{5 \cdot B} \cdot HW(T^*, T)$. By stability of Ψ , the number of clauses of Ψ not satisfied by x' is at least $s' \cdot HW(x^*, x') \cdot m \geq \frac{s'}{5B} \cdot HW(T^*, T) \cdot m$. The number of bad clauses is at most $B \cdot \zeta_B \cdot n$ since each bad variable appears in at most B bad clauses. So the number of bad clauses is bounded by

$$B \cdot \zeta_B \cdot n \leq 3B \cdot \zeta_B \cdot m \leq 3B \cdot \frac{s'}{30 \cdot B^2} \cdot HW(T^*, T) \cdot m = \frac{s'}{10 \cdot B} \cdot HW(T^*, T) \cdot m.$$

The number of good-unsatisfied clauses is at least the number of clauses not satisfied by x' minus the number of bad clauses, thus is at least

$$\frac{s'}{5 \cdot B} \cdot HW(T^*, T) \cdot m - \frac{s'}{10 \cdot B} \cdot HW(T^*, T) \cdot m = \frac{s'}{10 \cdot B} \cdot HW(T^*, T) \cdot m.$$

So in at least $\frac{s'}{10 \cdot B} \cdot HW(T^*, T) \cdot m = \frac{s'}{120 \cdot B} \cdot HW(T^*, T) \cdot |T^*|$ clause gadgets, T has one fewer triple than T^* . That is,

$$|T| \leq \left(1 - \frac{s'}{120 \cdot B} \cdot HW(T^*, T) \right) \cdot |T^*|.$$

In either case, we have $|T| \leq (1 - s_1 \cdot HW(T^*, T)) \cdot |T^*|$ where $s_1 = \frac{s'}{120 \cdot B^3}$. Clearly $s_1 \leq \frac{2}{17}$ also holds, which is required for Lemma 3. This completes the stability analysis.

5.3 Soundness Analysis

Suppose any truth assignment to Ψ satisfies at most $(1 - \rho) \cdot m$ clauses where $\rho > 0$ is an absolute constant. Our analysis breaks into two cases similar to the stability analysis, but it is much shorter.

Let T be any packing of triples. We define good and bad variables as in the stability analysis and let ζ_B denote the fraction of bad variables. Let x' be the following truth assignment. We set x'_i to be the truth assignment corresponding to the packing T for every good variable x_i . We set x'_i arbitrarily for every bad variable. Also let $\Lambda = 12 \cdot m$, which is the number of triples required to cover every point exactly once (i.e. the optimal value in the completeness case).

- **Case:** $\zeta_B > \frac{\rho}{6B}$.

As in the analogous case from the stability analysis and using $B \cdot n \geq 3m$.

$$|T| \leq \Lambda - \zeta_B \cdot n \leq \Lambda - \frac{\rho}{6B} \cdot n \leq \Lambda - \frac{\rho}{24B^2} \cdot m = \left(1 - \frac{\rho}{24B^2} \right) \cdot \Lambda.$$

Again, this is because in the $\zeta_B \cdot n$ variable gadgets for bad variables x_i , we have $|T \cap G_i| \leq \beta_i - 1$.

- **Case:** $\zeta_B \leq \frac{\rho}{6B}$.

The number of clauses that involve a bad variable is at most $B \cdot \zeta_B \cdot n$. Also, the number of clauses not satisfied by x' is at least $\rho \cdot m$. Of these, at least $\rho \cdot m - B \cdot \zeta_B \cdot n$ clauses include only good variables. Since these clauses are not satisfied, by Claim 4, T includes at most 8 triples from the corresponding clause gadget. That is, using our bound on ζ_B and $n \leq 3m$ we have

$$|T| \leq \Lambda - (\rho \cdot m - B \cdot \zeta_B \cdot n) \leq \Lambda - \frac{\rho}{2} \cdot m = \left(1 - \frac{\rho}{24}\right) \cdot \Lambda.$$

Considering either case, we see $|T| \leq (1 - \rho/(24B^2)) \cdot \Lambda$, as required.

6 From STABLE 3DM-B to STABLE COVERING BY TRIPLES-B

Here, we show a reduction from any instance of the S-3DM-B problem to an instance of the S-CBT-B problem. In this section, for two sets of triples T and T' , we define the Hamming distance function the same as before, $HW(T, T') = \frac{|T \Delta T'|}{2n}$, which is the size of the symmetric distance of the two set of triples divided by two times the size of the maximum set of disjoint triples.

Theorem 9. *There exists a polynomial reduction transforming any instance \mathcal{I} of S-3DM-B with $3n$ vertices and m triples to an instance \mathcal{I}' of S-CBT-B with the same number of vertices and triples, such that*

1. **Yes Case:** *if \mathcal{I} admits a perfect matching, then there exists a unique set of n triples T^* in \mathcal{I}' that cover the entire set of nodes. Furthermore, for any set of triples T of size n (different from T^*), T covers at most $(1 - s_2 \cdot HW(T^*, T)) \cdot 3n$ vertices.*
2. **No Case:** *if \mathcal{I} does not admit a perfect matching, then every set of triples T with $|T| = n$, covers at most $(1 - \gamma_2) \cdot 3n$ vertices,*

where $\gamma_2 (< \gamma_1)$ and s_2 are universal positive constants in the $(0, 1)$ interval.

Proof. The transformation function on the instances is an identity function, that is, we consider the same graph with the same set of triples.

Completeness: Let T^* be the perfect matching for instance \mathcal{I} . Obviously, the same set T^* covers the entire set of vertices in \mathcal{I}' as well. Any set of triples that covers all the vertices and has a size of n must be disjoint (since there are $3n$ nodes to cover), hence a perfect matching. By the uniqueness of perfect matching in \mathcal{I} , we conclude that the covering set of triples for \mathcal{I}' is also unique. Now, consider a set of triples T with $|T| = n$, and let T' be a maximal subset of disjoint triples of T . Let $cov(T)$ denote the number of vertices covered by T . Note that any of the triple in $T - T'$ intersect with at least one other triple in T' by maximality of T' , so they can cover at most 2 extra vertices compared to the vertices already covered by T' . Then

$$\begin{aligned} cov(T) &\leq 3|T'| + 2|T - T'| \\ &\leq 3(1 - s_1 \cdot HW(T^*, T'))n + 2|T - T'| \\ &= 3n - \frac{3s_1 \cdot |T^* \Delta T'|}{2n}n + 2|T - T'| \\ &= 3n - \frac{3s_1}{2}|T^* \Delta T'| + 2|T - T'| \\ &\leq 3n - \frac{3s_1}{2}|T^* \Delta T| + \frac{3s_1}{2}|T - T'| + 2|T - T'| \\ &= 3n - \frac{3s_1}{2}|T^* \Delta T| + \frac{3s_1 + 4}{2}|T - T'| \end{aligned} \tag{5}$$

where in the last inequality, we have used the fact that $|T^* \Delta T'| \geq |T^* \Delta T| - |T - T'|$. Now, we consider two cases:

Case 1: If $\frac{3s_1}{2}|T^*\Delta T| - \frac{3s_1+4}{2}|T-T'| > \frac{3s_1}{4}|T^*\Delta T|$. In this case, using (5) we simply bound $cov(T)$ as

$$\begin{aligned} cov(T) &\leq 3n - \frac{3s_1}{2}|T^*\Delta T| + \frac{3s_1+4}{2}|T-T'| \\ &< 3n - \frac{3s_1}{4}|T^*\Delta T| \\ &= 3n - \frac{3s_1 \cdot n}{4n}|T^*\Delta T| \\ &= 3n \left(1 - \frac{s_1 \cdot HW(T^*, T)}{2}\right). \end{aligned}$$

Case 2: If $\frac{3s_1}{2}|T^*\Delta T| - \frac{3s_1+4}{2}|T-T'| \leq \frac{3s_1}{4}|T^*\Delta T|$, with re-arranging the terms we get $|T-T'| \geq \frac{3s_1}{6s_1+8}|T^*\Delta T|$. Since every triple in $T-T'$ can cover at most 2 extra nodes compared to T' , each triple in this set can be charged with one deficiency in the coverage of T from a maximum of $3n$. Therefore, it must be the case that

$$\begin{aligned} cov(T) &\leq 3n - |T-T'| \\ &\leq 3n - \frac{3s_1}{6s_1+8}|T^*\Delta T| \\ &= 3n - \frac{3s_1 \cdot n}{(3s_1+4) \cdot 2n}|T^*\Delta T| \\ &= 3n \left(1 - \frac{s_1}{3s_1+4}\right) HW(T^*, T). \end{aligned}$$

Choosing $s_2 = \frac{s_1}{3s_1+4}$ is sufficient in both cases to obtain the required stability condition.

Soundness: Let T be a set of n triples in \mathcal{I}' , and assume T' is a maximal disjoint subset of T . From the No case of the S-3DM-B problem, we know that $|T'| \leq (1-\gamma_1)n$, thus $|T-T'| \geq \gamma_1 \cdot n$. As argued before, the size of $T-T'$ is a lower bound on the number of nodes that are not covered. Therefore

$$\begin{aligned} cov(T) &\leq 3n - |T-T'| \\ &\leq 3n - \gamma_1 \cdot n \\ &= 3n \left(1 - \frac{\gamma_1}{3}\right). \end{aligned}$$

Choosing $\gamma_2 = \frac{\gamma_1}{3}$ yields the required result. □

7 Reduction From STABLE COVERING BY TRIPLES-B to STABLE k -MEANS

We now show a reduction from any instance of the S-CBT-B to $(1+\epsilon_0)$ -stable instances of the discrete k -MEANS problem, hence completing proof of Theorem 4. In the following, by the Hamming weight function of two sets of centers C and C' we mean $\frac{|C\Delta C'|}{2k}$. When $|C| = |C'|$, we have $|C\Delta C'| = |C-C'| + |C'-C| = 2|C-C'|$, so the Hamming weight function effectively becomes $HW(C, C') = \frac{|C-C'|}{k}$.

Theorem 10. *There is a polynomial-time reduction transforming any instance \mathcal{I} of S-CBT-B to an instance $(\mathcal{X}, \mathcal{C}, \delta)$ of the (discrete) k -MEANS problem in Euclidean space with the following properties.*

1. **Yes Case:** *If \mathcal{I} is an instance from the **Yes** case in Theorem 9, then there is a unique optimum set of k centres $C^* \subset \mathcal{C}$ whose cost is $6k$ which remains the unique optimum solution after any $(1+\epsilon_0)$ perturbations. Furthermore, for any set of centres C of size k , the k -MEANS cost (without perturbations) of C is at least $(6+s_2 \cdot HW(C^*, C)) \cdot k$.*

2. **No Case:** If \mathcal{I} is an instance from the **No** case in Theorem 9, then every set of k centres $C \subset \mathcal{C}$, has a k -MEANS cost of at least $(1 + \gamma_2) \cdot 6k$.

where $\epsilon_o > 0$ is some universal constant and $\gamma_2, s_2 > 0$ is the constant from Theorem 9.

The last statement of the **Yes** case is not required to conclude the proof Theorem 4, but it may prove useful for future stability-preserving reductions that build off of our reduction.

Proof. For every point $v_j \in V$, $j = 1, 2, \dots, 3n$, create a point of \mathcal{X} , $x_j = \mathbf{e}_j \in \mathbb{R}^{3n}$, where \mathbf{e}_j has a 1 in its j^{th} coordinate, and zeros everywhere else. For every triple $t_\ell = (v_i, v_j, v_k) \in \mathcal{T}$, $\ell = 1, 2, \dots, m$, create a new point $y_\ell = \mathbf{e}_i + \mathbf{e}_j + \mathbf{e}_k$ in \mathcal{C} . Finally, use $k = n$ as the number of centres for the resulting k -MEANS instance.

We say that y_ℓ covers a point x_i if y_ℓ has a 1 in its i^{th} coordinate. Note for two points x_i, y_ℓ we have $\delta(x_i, y_\ell)^2 = 2$ if y_ℓ covers x_i and $\delta(x_i, y_\ell)^2 = 4$ otherwise.

Completeness: We use $\epsilon_o := \min\{0.4, \frac{2s_2}{3s_2+3}\}$.

Assume that, in a given instance \mathcal{I} of the S-CBT-B, there exists a unique covering set of triples T^* , and all other subsets $T \subseteq \mathcal{T}$, $|T| = n$ cover a fraction less than $1 - s_2 \cdot HW(T^*, T)$ of the points where s_2 is the constant from Theorem 9. Let $C^* \subset \mathcal{C}$ be the set of centres corresponding to the n triples in T^* , i.e. $C^* = \{\mathbf{e}_i + \mathbf{e}_j + \mathbf{e}_k : (v_i, v_j, v_k) \in T^*\}$. Since T^* covers all the points in instance \mathcal{I} , each $x_j \in \mathcal{X}$ is covered by some centre in C^* so $\delta(x_j, C^*)^2 = 2$. Thus the cost of C^* is $2 \cdot |\mathcal{X}| = 6n$.

We next show the cost of any set of centres $C \subset \mathcal{C}$, $|C| = k$ is at least $(6 + s_2 \cdot HW(C^*, C)) \cdot k$. To that end, let $T \subset \mathcal{T}$ be the triples corresponding to centres in C . Observe T leaves at least $s_2 \cdot HW(T^*, T) \cdot 3n$ points not covered in \mathcal{I} . The corresponding points in \mathcal{X} are not covered by C and $\delta(v_j, C)^2 = 4$ for any v_j not covered by C . Thus, if we let η_C denote the fraction of points in \mathcal{X} not covered by C (noting $\eta_C \geq s_2 \cdot HW(T^*, T)$) we have:

$$\sum_{j \in \mathcal{X}} \delta(j, C)^2 = (1 - \eta_C) \cdot 2 \cdot 3n + \eta_C \cdot 4 \cdot 3n = (1 + \eta_C) \cdot 6n \geq (1 + s_2 \cdot HW(C^*, C)) \cdot 6n,$$

as required (noting $HW(C^*, C) = HW(T^*, T)$).

Now let $\delta'()$ be any $(1 + \epsilon_o)$ -perturbation of δ , so $\delta(j, i) \leq \delta'(j, i) \leq (1 + \epsilon_o) \cdot \delta(j, i)$ for all $j \in \mathcal{X}, i \in \mathcal{C}$. We claim C^* remains the unique optimum solution to this k -MEANS instance. To that end, consider any $C \subset \mathcal{C}$, $|C| = k$ with $C \neq C^*$. Let $\sigma : \mathcal{X} \rightarrow C$ map each point to its nearest centre according to distances $\delta'()$. Also let $\sigma^* : \mathcal{X} \rightarrow C^*$ map each point to its nearest centre in C^* , again under distances $\delta'()$. Since each x_j is covered by exactly one centre in C^* and since the perturbation ϵ_o is small (i.e. $2 \cdot (1 + \epsilon_o)^2 < 4$), $\sigma^*(x_j)$ is the unique centre in C^* covering x_j .

Note as ϵ_o is small (i.e. $2 \cdot (1 + \epsilon_o)^2 < 4$ which holds because $\epsilon_o < 0.4$), if a point $x_j \in \mathcal{X}$ is covered by some centre in C , then σ will map x_j to a point in C that covers x_j . We partition \mathcal{X} into three groups:

- **A:** The points x_j not covered by any centre in C . Note $C \neq C^*$ means $\mathbf{A} \neq \emptyset$.
- **B:** The points x_j covered by a centre in C and have $\sigma(x_j) \in C - C^*$.
- **R:** The points x_j covered by a centre in C and have $\sigma(x_j) \in C \cap C^*$.

For each $x_j \in \mathcal{X}$, let ϵ_j be such that $\delta'(x_j, \sigma^*(x_j)) = \delta(x_j, \sigma(x_j)) \cdot (1 + \epsilon_j)$, so $0 \leq \epsilon_j \leq \epsilon_o$. Note the following bounds on $\delta(x_j, \sigma(x_j))$ for each point $x_j \in \mathcal{X}$:

- For $x_j \in \mathbf{A}$, $\delta'(x_j, \sigma(x_j))^2 \geq 4$ since C does not cover x_j .
- For $x_j \in \mathbf{B}$, $\delta'(x_j, \sigma(x_j))^2 \geq 2$.
- For $x_j \in \mathbf{R}$, $\delta'(x_j, \sigma(x_j))^2 = 2 \cdot (1 + \epsilon_j)^2$. This is because $\sigma(x_j) = \sigma^*(x_j)$ for such j as there is a unique centre (namely $\sigma^*(j)$) in $C \cap C^*$ covering x_j .

Therefore:

$$\text{cost}'(C) := \sum_{x_j \in \mathcal{X}} \delta'(x_j, C)^2 \geq |\mathbf{A}| \cdot 4 + |\mathbf{B}| \cdot 2 + \sum_{x_j \in \mathbf{R}} 2 \cdot (1 + \epsilon_j)^2.$$

On the other hand, using $(1 + \epsilon_o)^2 < 1 + 3\epsilon_o$ (as $\epsilon_o < 1$) we see:

$$\text{cost}'(C^*) := \sum_{x_j \in \mathcal{X}} \delta'(x_j, C^*)^2 < (|\mathbf{A}| + |\mathbf{B}|) \cdot 2 \cdot (1 + 3 \cdot \epsilon_o) + \sum_{x_j \in \mathbf{R}} 2 \cdot (1 + \epsilon_j)^2.$$

The bound is strict since $\mathbf{A} \neq \emptyset$. From what we showed earlier, $|\mathbf{A}| \geq 3n \cdot s_2 \cdot \text{HW}(C, C^*)$ since \mathbf{A} are the points not covered by C . Also, $|\mathbf{B}| \leq 3 \cdot |C - C^*| = 3k \cdot \text{HW}(C, C^*)$. We now see

$$\begin{aligned} \text{cost}'(C) - \text{cost}'(C^*) &> |\mathbf{A}| \cdot (2 - 3 \cdot \epsilon_o) - |\mathbf{B}| \cdot 3 \cdot \epsilon_o \\ &\geq 3n \cdot s_2 \cdot \text{HW}(C, C^*) \cdot (2 - 3 \cdot \epsilon_o) - 3n \cdot \text{HW}(C, C^*) \cdot \epsilon_o \\ &= 3n \cdot \text{HW}(C, C^*) \cdot (2s_2 - (3s_2 + 3) \cdot \epsilon_o) = 0. \end{aligned}$$

The equality with 0 is by our choice of ϵ_o . Therefore, $\text{cost}'(C) > \text{cost}'(C^*)$ as required. Since this holds for any $C \neq C^*$ with $|C| = n$, we see C^* remains the unique optimum set of n centres in the instance with perturbed distances δ' .

Soundness: Assume \mathcal{I} is a non-covering instance of S-CBT-B, so we have that any $T \subseteq \mathcal{T}, |T| = n$ can cover at most $(1 - \gamma_2) \cdot 3n$ points. In the k -MEANS instance, consider any $C \subseteq \mathcal{C}, |C| = k$. Let T be the triples corresponding to C . Since T fails to cover $\eta'_T \geq \gamma_2 \cdot 3n$ points, then the cost of C is:

$$2 \cdot (3n - \eta'_T) + 4 \cdot \eta'_T = 6n + 2\eta'_T \geq (1 + \gamma_2) \cdot 6n.$$

□

Proof of Theorem 4. Now we combine the results of Sections 4 to 7. Assuming Theorem 3, it follows that S-QSAT-B is hard. The reduction in Section 4 implies the hardness of S-3SAT-B. Then reduction from Section 5 implies hardness of S-3DM-B. Reduction of Section 6 implies hardness of S-CBT-B. Finally, Theorem 10 implies hardness of stable k -MEANS. □

8 Conclusion

We showed stable instances of k -MEDIAN and k -MEANS in metrics with constant doubling dimension, including constant-dimensional Euclidean metrics, can be solved in polynomial time by using a standard local search algorithm that always takes the best improvement. We also showed stable instances are hard to solve for some stability constant in arbitrary dimension Euclidean metrics. A natural problem is to find faster algorithms for solving stable k -MEANS. A related direction to consider is what notions of stability cause other **PLS**-complete problems to become polynomial-time solvable.

We also used the concept of stability-preserving reductions to show hardness for stable k -MEANS by leveraging a new PCP construction by Paradise [31]. What other stable optimization problems can be proven hard with this approach?

References

- [1] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k -means and euclidean k -median by primal-dual algorithms. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 61–72, 2017.
- [2] Ekaterina Alekseeva, Yuri Kochetov, and Alexander Plyasunov. Complexity of local search for the p -median problem. In *European Journal of Operational Research*, volume 191(3), pages 736–752, 2008.
- [3] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Mach. Learn.*, 75(2):245–248, 2009.
- [4] Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 438–451, New York, NY, USA, 2017. ACM.

- [5] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, , and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput*, 33(3):544–562, 2004.
- [6] Pranjal Awasthi, Avrim Blum, and Or Sheffet. Stability yields a ptas for k-median and k-means clustering. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS '10*, pages 309–318, Washington, DC, USA, 2010. IEEE Computer Society.
- [7] Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. In *Inf. Process. Lett.*, volume 112(1-2), pages 49–54, 2012.
- [8] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The Hardness of Approximation of Euclidean K-Means. In *Proceedings of 31st International Symposium on Computational Geometry (SoCG '15)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 754–767, 2015.
- [9] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '09*, pages 1068–1077, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.
- [10] Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-center clustering under perturbation resilience. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 68:1–68:14, 2016.
- [11] Maria-Florina Balcan and Yingyu Liang. Clustering under perturbation resilience. *SIAM J. Comput.*, 45(1):102–155, 2016.
- [12] Shalev Ben-David and Lev Reyzin. Data stability in clustering: A closer look. *Theor. Comput. Sci.*, 558:51–61, 2014.
- [13] Yonatan Bilu, Amit Daniely, Nati Linial, and Michael E. Saks. On the practically interesting instances of MAXCUT. In *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany*, pages 526–537, 2013.
- [14] Yonatan Bilu and Nathan Linial. Are stable instances easy? *Comb. Probab. Comput.*, 21(5):643–660, September 2012.
- [15] Vincent Cohen-Addad and Karthik C.S. Inapproximability of clustering in lp metrics. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 519–539, 2019.
- [16] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 353–364, 2016.
- [17] Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. *Johnson Coverage Hypothesis: Inapproximability of k-means and k-median in ℓ_p -metrics*, pages 1493–1530.
- [18] Vincent Cohen-Addad and Chris Schwiegelshohn. On the local structure of stable clustering instances. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 49–60, 2017.
- [19] Irit Dinur, Oded Goldreich, and Tom Gur. Every set in p is strongly testable under a suitable encoding. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:50, 2018.
- [20] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Mach. Learn.*, 56(1-3):9–33, 2004.
- [21] Zachary Friggstad, Kamyar Khodamoradi, and Mohammad R. Salavatipour. Exact algorithms and lower bounds for stable instances of euclidean k-means. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2958–2972. SIAM, 2019.

- [22] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 365–374, 2016.
- [23] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *SIAM J. Comput.*, 48(2):452–480, 2019.
- [24] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, first edition edition, 1979.
- [25] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for K-means clustering. *Comput. Geom. Theory Appl.*, 28(2-3):89–112, 2004.
- [26] Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Information Processing Letters*, 120:40–43, 2017.
- [27] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar K-means problem is NP-hard. In *Proceedings of the 3rd International Workshop on Algorithms and Computation (WALCOM '09)*, pages 274–285. Springer-Verlag, 2009.
- [28] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Bilu-linial stable instances of max cut and minimum multiway cut. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '14*, pages 890–906, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.
- [29] Jiri Matoušek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000.
- [30] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *J. ACM*, 59(6):28:1–28:22, 2012.
- [31] Orr Paradise. Smooth and strong pcps. *computational complexity*, 30(1), 2021.
- [32] L. Valiant and V. Vazirani. Np is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.
- [33] Andrea Vattani. The hardness of K-means clustering in the plane. *Manuscript*, 2009.
- [34] Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2001.
- [35] Aravindan Vijayaraghavan, Abhratanu Dutta, and Alex Wang. Clustering stable instances of euclidean k-means. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6503–6512, 2017.

A Alternative Local Search Convergence Analysis

Consider some problem where \mathfrak{F} is the set of feasible solutions and each $S \in \mathfrak{F}$ is endowed with an *integer* value $\text{cost}(S)$ that can be evaluated in polynomial time. The goal is to find some $S \in \mathfrak{F}$ with minimum $\text{cost}(S)$. We describe a setting encountered in many approximation algorithms based on local search and provide alternative analysis on the convergence of the local search heuristics that show the locality guarantee is obtained after a polynomial number of iterations. That is, we avoid the “ ϵ ” that is typically lost in the guarantee from many local search algorithm that only take noticeable improvements (*e.g.* only if the cost improves by a $(1 - \epsilon/k)$ -factor for some value k). When we say “polynomial time” in this context, we mean the running time is polynomial in the input size of the underlying problem.

For each $S \in \mathfrak{F}$, let $\mathcal{N}(S) \subseteq \mathfrak{F}$ be a set of *neighbouring solutions* with the property $S \in \mathcal{N}(S)$. We assume $\mathcal{N}(S)$ can be enumerated in polynomial time (implying $|\mathcal{N}(S)|$ is polynomially-bounded). Suppose we also know some Δ such that $\text{cost}(S) \leq \Delta$ for each $S \in \mathfrak{F}$ where $\log \Delta$ is polynomially bounded in the input size. If the reader wants to consider a specific setting, consider the k -MEDIAN problem where \mathfrak{F} is all subsets of

k centres and $\mathcal{N}(\mathcal{S})$ is the set of feasible solutions \mathcal{S}' with $|\mathcal{S} - \mathcal{S}'| = 1$ (i.e. the single-swap setting), and $\Delta = |\mathcal{X}| \cdot \max_{i,j} \delta(i, j)$.

Next, suppose we have the following “locality” analysis: for each $\mathcal{S} \in \mathfrak{F}$ there is a set of neighbouring solutions $\mathfrak{G}(\mathcal{S}) \subseteq \mathcal{N}(\mathcal{S})$ (sometimes called *test swaps*) where

$$\sum_{\mathcal{S}' \in \mathfrak{G}(\mathcal{S})} (\text{cost}(\mathcal{S}') - \text{cost}(\mathcal{S})) \leq \alpha \cdot \text{cost}(\mathcal{O}) - \beta \cdot \text{cost}(\mathcal{S}). \quad (6)$$

for some fixed rational values $\alpha \geq 1$ and $0 < \beta \leq 1$ that are both integer multiples of M (i.e. the least-common multiple of the denominators of α and β). Finally, suppose we have a bound $|\mathfrak{G}(\mathcal{S})| \leq \kappa$ for each $\mathcal{S} \in \mathfrak{F}$ on the number of “test swaps” in the above bound. We know κ is polynomially-bounded because $|\mathcal{N}(\mathcal{S})|$ is polynomially-bounded, but perhaps κ is even smaller. This is the case in many applications. In the single-swap k -MEDIAN setting, Arya et al. [5] find such a set of test swaps with $\alpha = 5, \beta = 1$ and $\kappa = k$ and, after scaling distances to clear denominators, we could pick $\Delta = n \cdot \max_{i,j} \delta(i, j)$.

Consider the following local search algorithm for this generic setting. Note we do not really need to track

Algorithm 2 A Generic Local Search Algorithm

```

let  $\mathcal{S}^0$  be any set in  $\mathfrak{F}$ 
 $i \leftarrow 0$ 
for  $K := \lceil \kappa \cdot \ln(\Delta) \cdot M / \beta \rceil$  iterations do
    let  $\mathcal{S}^{i+1}$  be the cheapest set in  $\mathcal{N}(\mathcal{S}^i)$  ▷ could be  $\mathcal{S}^{i+1} = \mathcal{S}^i$ 
     $i \leftarrow i + 1$ 
return  $\mathcal{S}$ 

```

the index i , we could just update the current set with the best one in its neighbourhood. The indices will be helpful in the proof.

Clearly Algorithm 2 runs in polynomial time under our assumptions. We show the approximation guarantee is what is guaranteed by local optimum solutions, even though the returned solution itself might not be a true local optimum (i.e. it might still be that $\text{cost}(\mathcal{S}) \neq \min_{\mathcal{S}' \in \mathfrak{G}(\mathcal{S})} \text{cost}(\mathcal{S}')$).

Theorem 11. *The returned solution \mathcal{S}^K satisfies $\text{cost}(\mathcal{S}) \leq \frac{\alpha}{\beta} \cdot \text{cost}(\mathcal{O})$.*

Proof. We show for some $0 \leq i \leq K$ that $\beta \cdot \text{cost}(\mathcal{S}^i) - \alpha \cdot \text{cost}(\mathcal{O}) \leq 0$. As $\text{cost}(\mathcal{S}^{j+1}) \leq \text{cost}(\mathcal{S}^j)$ at each step $0 \leq j < K$, this would show $\text{cost}(\mathcal{S}^K) \leq \text{cost}(\mathcal{S}^i) \leq \frac{\alpha}{\beta} \cdot \text{cost}(\mathcal{O})$.

To that end, suppose $\beta \cdot \text{cost}(\mathcal{S}^i) - \alpha \cdot \text{cost}(\mathcal{O}) > 0$ for all $0 \leq i < K$; otherwise we are done. For each such i , (6) and the fact that the local search algorithm takes the best improvement at each step shows

$$\begin{aligned} \text{cost}(\mathcal{S}^{i+1}) &\leq \text{cost}(\mathcal{S}^i) + \min_{\mathcal{S}' \in \mathfrak{G}(\mathcal{S}^i)} (\text{cost}(\mathcal{S}') - \text{cost}(\mathcal{S}^i)) \\ &\leq \text{cost}(\mathcal{S}^i) + \frac{1}{|\mathfrak{G}(\mathcal{S}^i)|} \sum_{\mathcal{S}' \in \mathfrak{G}(\mathcal{S}^i)} (\text{cost}(\mathcal{S}') - \text{cost}(\mathcal{S}^i)) \\ &\leq \text{cost}(\mathcal{S}^i) + \frac{\alpha \cdot \text{cost}(\mathcal{O}) - \beta \cdot \text{cost}(\mathcal{S}^i)}{|\mathfrak{G}(\mathcal{S}^i)|} \\ &\leq \text{cost}(\mathcal{S}^i) + \frac{\alpha \cdot \text{cost}(\mathcal{O}) - \beta \cdot \text{cost}(\mathcal{S}^i)}{\kappa} \end{aligned}$$

Thus,

$$\beta \cdot \text{cost}(\mathcal{S}^{i+1}) - \alpha \cdot \text{cost}(\mathcal{O}) \leq \left(1 - \frac{\beta}{\kappa}\right) \cdot (\beta \cdot \text{cost}(\mathcal{S}^i) - \alpha \cdot \text{cost}(\mathcal{O})).$$

which, inductively, shows

$$\begin{aligned} \beta \cdot \text{cost}(\mathcal{S}^K) - \alpha \cdot \text{cost}(\mathcal{O}) &\leq \left(1 - \frac{\beta}{\kappa}\right)^K \cdot (\beta \cdot \text{cost}(\mathcal{S}^0) - \alpha \cdot \text{cost}(\mathcal{O})) \\ &< e^{-\ln(\Delta M)} \cdot \Delta = 1/M. \end{aligned}$$

As $\beta \cdot \text{cost}(\mathcal{S}^K) - \alpha \cdot \text{cost}(\mathcal{O})$ is an integer multiple of $1/M$, then $\beta \cdot \text{cost}(\mathcal{S}^K) - \alpha \cdot \text{cost}(\mathcal{O}) \leq 0$ as required. \square

This analysis trivially extends to the weighted swap setting, where for each $\mathcal{S}' \in \mathfrak{G}(\mathcal{S})$ we have a value $\lambda_{\mathcal{S}'} \geq 0$ and

$$\sum_{\mathcal{S}' \in \mathfrak{G}(\mathcal{S})} \lambda_{\mathcal{S}'} \cdot (\text{cost}(\mathcal{S}') - \text{cost}(\mathcal{S})) \leq \alpha \cdot \text{cost}(\mathcal{O}) - \beta \cdot \text{cost}(\mathcal{S}).$$

Taking $\kappa = \sum_{\mathcal{S}' \in \mathfrak{G}(\mathcal{S})} \lambda_{\mathcal{S}'}$ yields the same conclusion: Algorithm 2 will return a solution \mathcal{S} with $\text{cost}(\mathcal{S}) \leq \frac{\alpha}{\beta} \cdot \text{cost}(\mathcal{O})$.

B A Parsimonious L -Reduction That Does Not Preserve Stability

In this section, we demonstrate that a classic L -reduction that reduces an instance of QSAT to one with bounded occurrence for each variable does not necessarily preserve stability within any constant. Apart from the obvious point that the simple reduction does not work, we wish to impart the lesson that reductions that preserve stability are not immediately obtained by classic (parsimonious) L -reductions: *stability-preserving reductions* are a distinct concept.

In particular, we show that the classic technique of replacing each occurrence of a variable with an expander gadget fails to preserve stability. Our presentation mirrors that in [34].

First, [34] points out that for any $k \geq 1$ that one can efficiently construct a 14-regular multigraph $G_k = (V_k, E_k)$ with $|V_k| = k$ so that $|\delta_{E_k}(S; V - S)| \geq \min\{|S|, |V - S|\}$ for any $S \subseteq V_k$. For each variable $x \in X$ of a QSAT instance Φ , let k_x denote the number of clauses of Φ that depend on x . Replace x with k_x variables in a one-to-one fashion in these clauses, call these new variables x^1, \dots, x^{k_x} . Finally, for each edge $(i, j) \in G_{k_x}$ (viewing V_{k_x} as integers from 1 to k_x) add constraints $x^i \vee x^j$ and $\overline{x^i} \vee \overline{x^j}$. These two constraints ensure x^i and x^j have the same truth value.

As shown in [34], this is an L -reduction and each variable appears in at most 29 clauses in the resulting QSAT instance. It is also easy to verify it is a parsimonious reduction, noting any satisfying assignment requires all copies of a variable of Φ to have the same truth value.

We demonstrate stable instances of QSAT where each variable does not appear in a bounded number of clauses such that applying this reduction does not result in a stable instance of QSAT-B. For an integer $n \geq 1$, let Φ_n be the SAT instance with variables $X_n = \{z, x_1, x_2, \dots, x_n\}$ and the following clauses:

- $z \vee \overline{x_i}$ for each $1 \leq i \leq n$,
- $\overline{x_i}$ for each $1 \leq i \leq n$, and
- \overline{z} .

We note this could be “padded” to a 3SAT instance by adding gadgets like $F(z, w, y)$ instead of the clause \overline{z} where $F(z, w, y)$ is the collection of 7 clauses that enforce all literals to be FALSE (and similarly for the other clauses), but we stick with this smaller instance for ease of discussion.

Claim 6. *For any $n \geq 1$, Φ_n is $\frac{1}{2}$ -stable.*

Proof. Setting all variables in X_n to FALSE satisfies all clauses, call this assignment (z^*, x^*) . Consider some assignment (z', x') and let $h = HW((z^*, x^*), (z', x'))$.

If b is the number of variables that are set to **True**, then at least b clauses are not satisfied, namely the singleton clauses. The fraction of unsatisfied clauses is at least $b/(2n+1)$ and $h = b/(n+1)$. So the number of unsatisfied clauses is at least $\frac{b}{2n+1} \geq \frac{b}{2 \cdot (n+1)} = \frac{h}{2}$. \square

Now, for $n \geq 1$ let Ψ_n be the QSAT-B instance for $B = 29$ that results by applying the above reduction to Φ_n . Let s_n denote the stability of Ψ_n .

Claim 7. $s_n \rightarrow 0$ as $n \rightarrow \infty$

Proof. Note that the unique satisfying assignment for Ψ_n is to set all variables to **FALSE**. Consider the truth assignment that assigns all copies of z the value **True** and all copies of each x_i the value **False**. The only clause that is not satisfied is the single clause $\overline{z^{i'}}$, for whatever copy i' of z was used in the singleton clause \overline{z} of Φ_n . So the fraction of unsatisfied clauses is $O(1/n)$.

On the other hand, number of variables in Ψ_n is $3n + 1$ (the total size of all clauses of Φ_n) and the given assignment sets $n + 1$ of them to **True**. Thus, the hamming distance between this assignment and the all-**FALSE** assignment is at least $1/3$. So $s_n = O(1/n)$. \square