



# Lossy Kernelization of Same-Size Clustering

Sayan Bandyapadhyay<sup>1</sup> · Fedor V. Fomin<sup>2</sup> · Petr A. Golovach<sup>2</sup> ·  
Nidhi Purohit<sup>2</sup> · Kirill Simonov<sup>3</sup>

Accepted: 23 May 2023 / Published online: 10 July 2023  
© The Author(s) 2023

## Abstract

In this work, we study the  $k$ -median clustering problem with an additional equal-size constraint on the clusters from the perspective of parameterized preprocessing. Our main result is the first lossy (2-approximate) polynomial kernel for this problem parameterized by the cost of clustering. We complement this result by establishing lower bounds for the problem that eliminate the existence of an (exact) kernel of polynomial size and a PTAS.

**Keywords**  $k$ -median clustering · Parameterized approximation · Kernelization · Lossy kernels

## 1 Introduction

*Lossy kernelization* stems from *parameterized complexity*, a branch in theoretical computer science that studies the complexity of problems as functions of multiple *parameters* of the input or output [1]. A central notion in parameterized complexity is *kernelization*, which is a generic technique for designing efficient algorithms availing

---

✉ Nidhi Purohit  
nidhi.purohit@uib.no

Sayan Bandyapadhyay  
sayanb@pdx.edu

Fedor V. Fomin  
fedor.fomin@uib.no

Petr A. Golovach  
petr.golovach@uib.no

Kirill Simonov  
kirillsimonov@gmail.com

<sup>1</sup> Department of Computer Science, Portland State University, Portland, US

<sup>2</sup> Department of Informatics, University of Bergen, Bergen, Norway

<sup>3</sup> Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

a polynomial-time preprocessing step that transforms a “large” instance of a problem into a smaller, equivalent instance. Naturally, the preprocessing step is called the *kernelization algorithm* and the smaller instance is called the *kernel*. One limitation of the classical kernelization technique is that kernels can only deal with “lossless” preprocessing in the sense that a kernel must be equivalent to the original instance. This is why most of the interesting problems arising from machine learning, e.g., clustering, are intractable from the perspective of kernelization. Lossy or approximate kernelization is a successful attempt of combining kernelization with approximation algorithms. Informally, in lossy kernelization, given an instance of the problem and a parameter, we would like the kernelization algorithm to output a reduced instance of size polynomial in the parameter; however, the notion of equivalence is relaxed in the following way. Given a  $c$ -approximate solution (i.e., one with the cost within  $c$ -factor of the optimal cost) to the reduced instance, it should be possible to produce in a polynomial time an  $\alpha c$ -approximate solution to the original instance. The factor  $\alpha$  is the loss incurred while going from the reduced instance to the original instance. The notion of lossy kernelization was introduced by Lokshantov et al. in [2]. However, most of the developments of lossy kernelization up to now are in graph algorithms [3–7], see also [8, Chapter 23] for an overview.

One of the actively developing areas of parameterized complexity concerns *fixed-parameter tractable*- or FPT-approximation. We refer to the survey [9] for an overview of the area. Several important advances in FPT-approximation concern clustering problems. It includes tight algorithmic and complexity results for  $k$ -means and  $k$ -median [10] and constant factor FPT-approximation for capacitated clustering [11]. The popular approach for data compression used for FPT-approximation of clustering is based on *coresets*. The notion of coresets originated from computational geometry. It was introduced by Har-Peled and Mazumdar [12] for  $k$ -means and  $k$ -median clustering. Informally, a coreset is a summary of the data that for every set of  $k$  centers, approximately (within  $(1 \pm \varepsilon)$  factor) preserves the optimal clustering cost.

Lossy kernels and coresets have a lot of similarities. Both compress the space compared to the original data, and any algorithm can be applied on a coreset or kernel to efficiently retrieve a solution with a guarantee almost the same as the one provided by the algorithm on the original input. The crucial difference is that coreset constructions result in a small set of weighted points. The weights could be up to the input size  $n$ . Thus, a coreset of size polynomial in  $k/\varepsilon$ , is not a polynomial-sized lossy kernel for parameters  $k, \varepsilon$  because of the  $\log n$  bits required to encode the weights. Moreover, usually coreset constructions do not bound the number of coordinates or the dimension of the points.

While the notion of lossy kernelization proved to be useful in the design of graph algorithms, we are not aware of its applicability in clustering. This brings us to the following question: *What can lossy kernelization offer to clustering?*

In this work, we make the first step towards the development of lossy kernels for clustering problems. Our main result is the design of a lossy kernel for a variant of the well-studied  $k$ -MEDIAN clustering with clusters of equal sizes. More precisely, consider a collection (multiset) of points from  $\mathbb{Z}^d$  under the  $\ell_p$ -norm. Thus, every point is a  $d$ -dimensional vector with integer coordinates. For a nonnegative integer  $p$ , we use

$\|\mathbf{x}\|_p$  to denote the  $\ell_p$ -norm of a  $d$ -dimensional vector  $\mathbf{x} = (x[1], \dots, x[d]) \in \mathbb{R}^d$ , that is, for  $p \geq 1$ ,

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^d |x[i]|^p \right)^{1/p}$$

and for  $p = 0$ ,  $\|\mathbf{x}\|_0$  is the number of nonzero elements of  $\mathbf{x}$ , i.e., the Hamming norm. For any subset of points  $T \subseteq \mathbb{Z}^d$ , we define

$$\text{cost}_p(T) = \min_{\mathbf{c} \in \mathbb{R}^d} \sum_{\mathbf{x} \in T} \|\mathbf{c} - \mathbf{x}\|_p.$$

Then  $k$ -MEDIAN<sup>1</sup> clustering (without constraints) is the task of finding a partition  $\{X_1, \dots, X_k\}$  of a given set  $\mathbf{X} \subseteq \mathbb{Z}^d$  of points minimizing the sum

$$\sum_{i=1}^k \text{cost}_p(X_i).$$

In many real-life scenarios, it is desirable to cluster data into clusters of equal sizes. For example, to tailor teaching methods to the specific needs of various students, one would be interested in allocating  $k$  fair class sizes by grouping students with homogeneous abilities and skills [13]. In scheduling, the standard task is to distribute  $n$  jobs to  $k$  machines while keeping identical workloads on each machine and simultaneously reducing the configuration time. In the setting of designing a conference program, one might be interested in allocating  $n$  scientific papers according to their similarities to  $k$  “balanced” sessions [14].

The following model is an attempt to capture such scenario:

#### EQUAL CLUSTERING

*Input:* A collection (multiset)  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $n$  points of  $\mathbb{Z}^d$  and a positive integer  $k$  such that  $n$  is divisible by  $k$ .

*Task:* Find a partition  $\{X_1, \dots, X_k\}$  ( $k$ -clustering) of  $\mathbf{X}$  with  $|X_1| = \dots = |X_k| = \frac{n}{k}$  minimizing

$$\sum_{i=1}^k \text{cost}_p(X_i).$$

First, note that EQUAL CLUSTERING is a restricted variant of the capacitated version [11] of  $k$ -MEDIAN where the size of each cluster is required to be bounded by a given number  $U$ . Also note, that some points in  $\mathbf{X}$  may be equal (in the above examples, several students, jobs, or scientific papers can have identical features but could be

<sup>1</sup> Traditionally this problem is studied with real input points, but because of the choice of the parameterization, it is natural for us to assume that points have integer coordinates. As the coordinates can be scaled, this does not lead to the loss of generality.

assigned to different clusters due to the size limitations). We refer to the multisets  $X_1, \dots, X_k$  as the *clusters*.

To describe the lossy-kernel result, we need to define the parameterized version of EQUAL CLUSTERING with the cost of clustering  $B$  (the budget) being the parameter. Following the framework of lossy kernelization [2], when the cost of an optimal clustering exceeds the budget, we assume it is equal to  $B + 1$ . More precisely, in PARAMETERIZED EQUAL CLUSTERING, we are given an additional integer  $B$  (budget parameter). The task is to find a  $k$ -clustering  $\{X_1, \dots, X_k\}$  with  $|X_1| = \dots = |X_k|$  and minimizing the value

$$\text{cost}_p^B(X_1, \dots, X_k) = \begin{cases} \sum_{i=1}^k \text{cost}_p(X_i) & \text{if } \sum_{i=1}^k \text{cost}_p(X_i) \leq B, \\ B + 1 & \text{otherwise.} \end{cases}$$

Before stating our results, let us first discuss some limitations and advantages of parameterization of DECISION EQUAL CLUSTERING by the budget  $B$ . First, parameterization by  $B$  is reasonable when the vectors are integer-valued, which is a common situation when the data is categorical, that is, can admit a fixed number of possible values. For example, it could be gender, blood type, or political orientation. A prominent example of categorical data is binary data, where the points are binary vectors. Binary data arise in several critical applications. For example, in electronic commerce, each transaction can be modeled as a binary vector (known as market basket data), each of whose coordinates denotes whether a particular item is purchased or not [15, 16]. In document clustering, each document can be modeled as a binary vector, each of whose coordinates denotes whether a specific word is present or not in the document [15, 16].

The most drastic effect of compression occurs when  $B$  is small. Intuitively, this means that many of the data points are the same. Such a condition is common in handling personal data that cannot be re-identified. For example, the  $k$ -anonymity property requires each person in the data set to be undistinguishable from at least  $k$  individuals whose information appears in the release [17].

Finally, comparing lossy kernelization from Theorem 1 and the coresets for  $k$ -MEDIAN and  $k$ -MEANS. The sizes of all known coreset constructions depend on the number of clusters  $k$  and thus guarantee compression only for small values of  $k$ . On the other hand, the size of the lossy kernel is independent of  $k$ . In particular, such type of results is interesting when we have to identify many clusters of small size.

Our first main result is the following theorem providing a polynomial 2-approximate kernel.

**Theorem 1** *For every nonnegative integer constant  $p$ , PARAMETERIZED EQUAL CLUSTERING admits a 2-approximate kernel when parameterized by  $B$ , where the output collection of points has  $\mathcal{O}(B^2)$  points of  $\mathbb{Z}^{d'}$  with  $d' = \mathcal{O}(B^{p+2})$ , where each coordinate of a point takes an absolute value of  $\mathcal{O}(B^3)$ .*

In other words, the theorem provides a polynomial-time algorithm that compresses the original instance  $\mathbf{X}$  to a new instance whose size is bounded by a polynomial of

$B$  and such that any  $c$ -approximate solution in the new instance can be turned in a polynomial time to a  $2c$ -approximate solution to the original instance.

A natural question is whether the approximation ratio of the lossy kernel in Theorem 1 is optimal. While we do not have a complete answer to this question, we provide lower bounds supporting our study of the problem from the perspective of approximate kernelization. Our next result rules out the existence of an “exact” kernel for the problem. To state the result, we need to define the decision version of EQUAL CLUSTERING. In this version, we call it DECISION EQUAL CLUSTERING, the question is whether for a given budget  $B$ , there is a  $k$ -clustering  $\{X_1, \dots, X_k\}$  with clusters of the same size such that  $\sum_{1 \leq i \leq k} \text{cost}_p(X_i) \leq B$ .

**Theorem 2** *For the  $\ell_0$  and  $\ell_1$ -norms, DECISION EQUAL CLUSTERING has no polynomial kernel when parameterized by  $B$  unless  $\text{NP} \subseteq \text{coNP} / \text{poly}$ , even if the input points are binary, that is, are from  $\{0, 1\}^d$ .*

On the other hand, we prove that DECISION EQUAL CLUSTERING admits a polynomial kernel when parameterized by  $k$  and  $B$ .

**Theorem 3** *For every nonnegative integer constant  $p$ , DECISION EQUAL CLUSTERING admits a polynomial kernel when parameterized by  $k$  and  $B$ , where the output collection of points has  $\mathcal{O}(kB)$  points of  $\mathbb{Z}^{d'}$  with  $d' = \mathcal{O}(kB^{p+1})$  and each coordinate of a point takes an absolute value of  $\mathcal{O}(kB^2)$ .*

When it comes to approximation in a polynomial time, we show (Theorem 5) that it is NP-hard to obtain a  $(1 + \varepsilon_c)$ -approximation for EQUAL CLUSTERING with  $\ell_0$  (or  $\ell_1$ ) distances for some  $\varepsilon_c > 0$ . However, parameterized by  $k$  and  $\varepsilon$ , standard techniques yield  $(1 + \varepsilon)$ -approximation in FPT time. For the  $\ell_2$  norm, there is a general framework for designing algorithms of this form for  $k$ -MEDIAN with additional constraints on cluster sizes, introduced by Ding and Xu [18]. The best-known improvements by Bhattacharya et al. [19] achieve a running time of  $2^{\tilde{\mathcal{O}}(k/\varepsilon^{\mathcal{O}(1)})} n^{\mathcal{O}(1)} d$  in the case of EQUAL CLUSTERING, where  $\tilde{\mathcal{O}}$  hides polylogarithmic factors. In another line of work, FPT-time approximation is achieved via constructing small-sized coresets of the input, and the work [20] guarantees an  $\varepsilon$ -coreset for EQUAL CLUSTERING (in the  $\ell_2$  norm) of size  $(kd \log n / \varepsilon)^{\mathcal{O}(1)}$ , and consequently a  $(1 + \varepsilon)$ -approximation algorithm with running time  $2^{\tilde{\mathcal{O}}(k/\varepsilon^{\mathcal{O}(1)})} (nd)^{\mathcal{O}(1)}$ .

Moreover, specifically for EQUAL CLUSTERING, simple  $(1 + \varepsilon)$ -approximations with similar running time can be designed directly via sampling. A seminal work of Kumar et al. [21] achieves a  $(1 + \varepsilon)$ -approximation for  $k$ -MEDIAN (in the  $\ell_2$  norm) with running time  $2^{\tilde{\mathcal{O}}(k/\varepsilon^{\mathcal{O}(1)})} nd$ . The algorithm proceeds as follows. First, take a small uniform sample of the input points, and by guessing ensure that the sample is taken only from the largest cluster. Second, estimate the optimal center of this cluster from the sample. In the case of  $k$ -MEDIAN, Theorem 5.4 of [21] guarantees that from a sample of size  $(1/\varepsilon)^{\mathcal{O}(1)}$  one can compute in time  $2^{(1/\varepsilon)^{\mathcal{O}(1)}} d$  a set of candidate centers such that at least one of them provides a  $(1 + \varepsilon)$ -approximation to the cost of the cluster. Finally, “prune” the set of points so that the next largest cluster is at least  $\Omega(1/k)$  fraction of the remaining points and continue the same process with one less cluster. One can observe that in the case of EQUAL CLUSTERING, a simplification of the above algorithm

suffices: one does not need to perform the “pruning” step, as we are only interested in clusterings where all the clusters have size exactly  $n/k$ . Thus,  $(1/\varepsilon)^{O(1)}$ -sized uniform samples from each of the clusters can be computed immediately in total time  $2^{\tilde{O}(k/\varepsilon^{O(1)})}nd$ . This achieves  $(1 + \varepsilon)$ -approximation for EQUAL CLUSTERING with the same running time as the algorithm of Kumar et al. In fact, the same procedure works for the  $\ell_0$  norm as well, where for estimating the cluster center it suffices to compute the optimal center of a sample of size  $O(1/\varepsilon^2)$ , as proven by Alon and Sudakov [22]. Thus, in terms of FPT approximation, EQUAL CLUSTERING is surprisingly “simpler” than its unconstrained variant  $k$ -MEDIAN, however, our hardness result of Theorem 5 shows that the problems are similarly hard in terms of polynomial-time approximation.

**Related work** Since the work of Har-Peled and Mazumdar [12] for  $k$ -means and  $k$ -median clustering, designing small coresets for clustering has become a flourishing research direction. For these problems, after a series of interesting works, the best-known upper bound on coreset size in general metric space is  $O((k \log n)/\varepsilon^2)$  [23] and the lower bound is known to be  $\Omega((k \log n)/\varepsilon)$  [24]. For the Euclidean space (i.e., the  $\ell_2$ -norm) of dimension  $d$ , it is possible to construct coresets of size  $(k/\varepsilon)^{O(1)}$  [25, 26]. Remarkably, the size of the coresets in this case does not depend on  $n$  and  $d$ . For EQUAL CLUSTERING, the best known coreset size of  $(kd \log n/\varepsilon)^{O(1)}$  (for  $p = 2$ ) follows from coresets for the more general capacitated clustering problem [11, 20].

Clustering is undoubtedly one of the most common procedures in unsupervised machine learning. We refer to the book [27] for an overview on clustering. Algorithms for  $k$ -median and  $k$ -means has been one of the most interesting problems in the area of approximation and led to a plethora of work [28–32]. For  $k$ -median, the best known polynomial-time approximation factor is 2.675 [31] and for  $k$ -means, it is 6.356 [32]. Moreover, if one is allowed to use FPT time parameterized by  $k$ , then these two factors can be improved to  $\approx (1 + 2/e) \leq 1.736$  and  $\approx (1 + 8/e) \leq 3.944$ , respectively [10]. Assuming the Gap-ETH [33], these factors are indeed tight [10]. PTASes are known for Euclidean version of  $k$ -median and  $k$ -means when the dimension is a constant [34–38]. Cohen-Addad et al. [39] obtained an  $n^{o(k)}$  lower bound for  $k$ -median when  $d \geq 4$ . When the dimension  $d$  is arbitrary, one can obtain  $(1 + \varepsilon)$ -approximation in  $\text{FPT}(k)$  time where the dependency on  $n$  and  $d$  are only linear [21]. Feng et al. in [40] gave a unified framework to design FPT approximation algorithms for clustering problems.

EQUAL CLUSTERING belongs to a wide class of clustering with constraints on the sizes of the clusters. In many applications of clustering, constraints come naturally [41]. In particular, there is a rich literature on approximation algorithms for various versions of capacitated clustering. While for the capacitated version of  $k$ -median and  $k$ -means in general metric space, no polynomial time  $O(1)$ -approximation is known, bicriteria constant-approximations violating either the capacity constraints or the constraint on the number of clusters, by an  $O(1)$  factor can be obtained [28, 42–47]. Cohen-Addad and Li [11] designed  $\text{FPT} \approx 3$ - and  $\approx 9$ -approximation with parameter  $k$  for the capacitated version of  $k$ -median and  $k$ -means, respectively. For these problems in the Euclidean plane, Cohen-Addad [48] obtained a true PTAS. Moreover, for higher dimensional spaces (i.e.,  $d \geq 3$ ), he designed a  $(1 + \varepsilon)$ -approximation that runs in

time  $n^{(\log n/\varepsilon)^{O(d)}}$  [48]. Being a restricted version of capacitated clustering, EQUAL CLUSTERING admits all the approximation results mentioned above.

**Our approach** We briefly sketch the main ideas behind the construction of our lossy kernel for PARAMETERIZED EQUAL CLUSTERING. The lossy kernel's main ingredients are a) a polynomial algorithm based on an algorithm for computing a minimum weight perfect matching in bipartite graphs, b) preprocessing rules reducing the size and dimension of the problem, and c) a greedy algorithm. Each of the steps is relatively simple and easily implementable. However, proving that these steps result in a lossy kernel with the required properties is not easy.

Recall that for a given budget  $B$ , we are looking for a  $k$ -clustering of a collection of points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  into  $k$  clusters of the same size minimizing the cost. We also assume that the cost is  $B + 1$  if the instance points do not admit a clustering of cost at most  $B$ . Informally, we are only interested in optimal clustering when its cost does not exceed the budget. First, if the cluster's size  $s = \frac{n}{k}$  is sufficiently large (with respect to the budget), we can construct an optimal clustering in a polynomial time. More precisely, we prove that if  $s \geq 4B + 1$ , then the clusters' medians could be selected from  $\mathbf{X}$ . Moreover, we show how to identify the (potential) medians in a polynomial time. In this case, constructing an optimal  $k$ -clustering could be reduced to the classical problem of computing a perfect matching of minimum weight in a bipartite graph.

The case of cluster's size  $s \leq 4B$  is different. We apply a set of reduction rules. These rules run in a polynomial time. After exhaustive applications of reduction rules, we either correctly conclude that the considered instance has no clustering of cost at most  $B$  or constructs an equivalent reduced instance. In the equivalent instance, the dimension is reduced to  $\mathcal{O}(kB^{p+1})$  while the absolute values of the coordinates of the points are in  $\mathcal{O}(kB^2)$ .

Finally, we apply the only approximate reduction on the reduced instance. The approximation procedure is greedy: whenever there are  $s$  equal points, we form a cluster out of them. For the points remaining after the exhaustive application of the greedy procedure, we conclude that either there is no clustering of cost at most  $B$  or the number of points is  $\mathcal{O}(B^2)$ . This construction leads us to the lossy kernel. However, the greedy selection of the clusters composed of equal points may not be optimal. In particular, the reductions used to obtain our algorithmic lower bounds given in Sections 4 and 5 exploit the property that it may be beneficial to split a block of  $s$  equal points between distinct clusters.

Nevertheless, the greedy clustering of equal points leads to a 2-approximation. The proof of this fact requires some work. We evaluate the clustering cost obtained from a given optimal clustering by swapping some points to form clusters composed of equal points. Further, we upper bound the obtained value by the cost of the optimum clustering. For the last step, we introduce an auxiliary clustering problem formulated as a min-cost flow problem. This reduction allows us to evaluate the cost and obtain the required upper bound.

**Organization of the paper** The remaining part of the paper is organized as follows. In Section 2, we introduce basic notation and show some properties of clusterings. In Section 3, we show our main result that PARAMETERIZED EQUAL CLUSTERING admits

a lossy kernel. In Section 4, we complement this result by proving that it is unlikely that DECISION EQUAL CLUSTERING admits an (exact) kernel of a polynomial size when parameterized by  $B$ . Still, the problem has a polynomial kernel when parameterized by  $B$  and  $k$ . In Section 5, we show that EQUAL CLUSTERING is APX-hard. We conclude in Section 6 by stating some open problems.

## 2 Preliminaries

In this section, we give basic definitions and introduce the notation used throughout the paper. We also state some useful auxiliary results.

**Parameterized complexity and kernelization** We refer to the recent books [8, 49] for a formal introduction to the area. Here we only define the notions used in our paper.

Formally, a *parameterized problem*  $\Pi$  is a subset of  $\Sigma^* \times \mathbb{N}$ , where  $\Sigma$  is a finite alphabet. Thus, an instance of  $\Pi$  is a pair  $(I, k)$ , where  $I \subseteq \Sigma^*$  and  $k$  is a nonnegative integer called a *parameter*. It is said that a parameterized problem  $\Pi$  is *fixed-parameter tractable* (FPT) if it can be solved in  $f(k) \cdot |I|^{\mathcal{O}(1)}$  time for some computable function  $f(\cdot)$ .

A *kernelization* algorithm (or *kernel*) for a parameterized problem  $\Pi$  is an algorithm that, given an instance  $(I, k)$  of  $\Pi$ , in a polynomial time produces an instance  $(I', k')$  of  $\Pi$  such that

- (i)  $(I, k) \in \Pi$  if and only if  $(I', k') \in \Pi$ , and
- (ii)  $|I'| + k' \leq g(k)$  for a computable function  $g(\cdot)$ .

The function  $g(\cdot)$  is called the *size* of a kernel; a kernel is *polynomial* if  $g(\cdot)$  is a polynomial. Every decidable FPT problem admits a kernel. However, it is unlikely that all FPT problems have polynomial kernels and the parameterized complexity theory provides tools for refuting the existence of polynomial kernels up to some reasonable complexity assumptions. The standard assumption here is that  $\text{NP} \not\subseteq \text{coNP} / \text{poly}$ .

We also consider the parameterized analog of optimization problems. Since we only deal with minimization problems where the minimized value is nonnegative, we state the definitions only for optimization problems of this type. A *parameterized minimization* problem  $P$  is a computable function

$$P: \Sigma^* \times \mathbb{N} \times \Sigma^* \rightarrow \mathbb{R}_{\geq 0} \cup \{+\infty\}.$$

The instances of a parameterized minimization problem  $P$  are pairs  $(I, k) \in \Sigma^* \times \mathbb{N}$ , and a solution to  $(I, k)$  is simply a string  $s \in \Sigma^*$ , such that  $|s| \leq |I| + k$ . Then the function  $P(\cdot, \cdot, \cdot)$  defines the *value*  $P(I, k, s)$  of a solution  $s$  to an instance  $(I, k)$ . The optimum value of an instance  $(I, k)$  is

$$\text{Opt}_P(I, k) = \min_{s \in \Sigma^* \text{ s.t. } |s| \leq |I| + k} P(I, k, s).$$



A solution  $s$  is *optimal* if  $\text{Opt}_P(I, k) = P(I, k, s)$ . A parameterized minimization problem  $P$  is said to be FPT if there is an algorithm that for each instance  $(I, k)$  of  $P$  computes an optimal solution  $s$  in  $f(k) \cdot |I|^{\mathcal{O}(1)}$  time, where  $f(\cdot)$  is a computable function. Let  $\alpha \geq 1$  be a real number. An FPT  $\alpha$ -approximation algorithm for  $P$  is an algorithm that in  $f(k) \cdot |I|^{\mathcal{O}(1)}$  time computes a solution  $s$  for  $(I, k)$  such that  $P(I, k, s) \leq \alpha \cdot \text{Opt}_P(I, k)$ , where  $f(\cdot)$  is a computable function.

It is useful for us to make some comments about defining  $P(\cdot, \cdot, \cdot)$  for the case when the considered problem is parameterized by the solution value. For simplicity, we do it informally and refer to [8] for details and explanations. If  $s$  is not a “feasible” solution to an instance  $(I, k)$ , then it is convenient to assume that  $P(I, k, s) = +\infty$ . Otherwise, if  $s$  is “feasible” but its value is at least  $k + 1$ , we set  $P(I, k, s) = k + 1$ .

**Lossy kernels** Finally, we define  $\alpha$ -approximate or *lossy* kernels for parameterized minimization problems. Informally, an  $\alpha$ -approximate kernel of size  $g(\cdot)$  is a polynomial-time algorithm, that given an instance  $(I, k)$ , outputs an instance  $(I', k')$  such that  $|I'| + k' \leq g(k)$  and any  $c$ -approximate solution  $s'$  to  $(I', k')$  can be turned in a polynomial time into a  $(c \cdot \alpha)$ -approximate solution  $s$  to the original instance  $(I, k)$ . More precisely, let  $P$  be a parameterized minimization problem and let  $\alpha \geq 1$ . An  $\alpha$ -approximate (or *lossy*) kernel for  $P$  is a pair of polynomial algorithms  $\mathcal{A}$  and  $\mathcal{A}'$  such that

- (i) given an instance  $(I, k)$ ,  $\mathcal{A}$  (called a *reduction algorithm*) computes an instance  $(I', k')$  with  $|I'| + k' \leq g(k)$ , where  $g(\cdot)$  is a computable function,
- (ii) the algorithm  $\mathcal{A}'$  (called a *solution-lifting algorithm*), given the initial instance  $(I, k)$ , the instance  $(I', k')$  produced by  $\mathcal{A}$ , and a solution  $s'$  to  $(I', k')$ , computes an solution  $s$  to  $(I, k)$  such that

$$\frac{P(I, k, s)}{\text{Opt}_P(I, k)} \leq \alpha \cdot \frac{P(I', k', s')}{\text{Opt}_P(I', k')}.$$

For simplicity, we assume here that  $\frac{P(I, k, s)}{\text{Opt}_P(I, k)} = 1$  if  $\text{Opt}_P(I, k) = P(I, k, s) = 0$  and  $\frac{P(I, k, s)}{\text{Opt}_P(I, k)} = +\infty$  if  $\text{Opt}_P(I, k) = 0$  and  $P(I, k, s) > 0$ ; the same assumption is used for  $\frac{P(I', k', s')}{\text{Opt}_P(I', k')}$ . As with classical kernels,  $g(\cdot)$  is called the *size* of an approximate kernel, and an approximate kernel is polynomial if  $g(\cdot)$  is a polynomial.

**Vectors and clusters** For a vector  $\mathbf{x} \in \mathbb{R}^d$ , we use  $\mathbf{x}[i]$  to denote the  $i$ -th element of the vector for  $i \in \{1, \dots, d\}$ . For a set of indices  $R \subseteq \{1, \dots, d\}$ ,  $\mathbf{x}[R]$  denotes the vector of  $\mathbb{R}^{|R|}$  composed by the elements of vector  $\mathbf{x}$  from set  $R$ , that is, if  $R = \{i_1, \dots, i_r\}$  with  $i_1 < \dots < i_r$  and  $\mathbf{y} = \mathbf{x}[R]$ , then  $\mathbf{y}[j] = \mathbf{x}[i_j]$  for  $j \in \{1, \dots, r\}$ . In our paper, we consider collections  $\mathbf{X}$  of points of  $\mathbb{Z}^d$ . We underline that some points of such a collection may be equal. However, to simplify notation, we assume throughout the paper that the equal points of  $\mathbf{X}$  are distinct elements of  $\mathbf{X}$  assuming that the points are supplied with unique identifiers. By this convention, we often refer to (sub)collections of points as (sub)sets and apply the standard set notation.

Let  $X$  be a collection of points of  $\mathbb{Z}^d$ . For a vector  $\mathbf{c} \in \mathbb{R}^d$ , we define the *cost of  $X$  with respect to  $\mathbf{c}$*  as

$$\text{cost}_p(X, \mathbf{c}) = \sum_{\mathbf{x} \in X} \|\mathbf{c} - \mathbf{x}\|_p.$$

Slightly abusing notation we often refer to  $\mathbf{c}$  as a (given) *median* of  $X$ . We say that  $\mathbf{c}^* \in \mathbb{R}^d$  is an *optimum median* of  $X$  if

$$\text{cost}_p(X) = \text{cost}_p(X, \mathbf{c}^*) = \min_{\mathbf{c} \in \mathbb{R}^d} \text{cost}_p(X, \mathbf{c}).$$

Notice that the considered collections of points have integer coordinates but the coordinates of medians are not constrained to integers and may be real.

Let  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  a collection of points of  $\mathbb{Z}^d$  and let  $k$  be a positive integer such that  $n$  is divisible by  $k$ . We say that a partition  $\{X_1, \dots, X_k\}$  of  $\mathbf{X}$  is an *equal  $k$ -clustering* of  $\mathbf{X}$  if  $|X_i| = \frac{n}{k}$  for all  $i \in \{1, \dots, k\}$ . For an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  and given vectors  $\mathbf{c}_1, \dots, \mathbf{c}_k$ , we define the *cost of clustering with respect to  $\mathbf{c}_1, \dots, \mathbf{c}_k$*  as

$$\text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k) = \sum_{i=1}^k \text{cost}_p(X_i, \mathbf{c}_i).$$

The *cost* of an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  is

$$\text{cost}_p(X_1, \dots, X_k) = \text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k),$$

where  $\mathbf{c}_1, \dots, \mathbf{c}_k$  are optimum medians of  $X_1, \dots, X_k$ , respectively. For an integer  $B \geq 0$ ,

$$\text{cost}_p^B(X_1, \dots, X_k) = \begin{cases} \text{cost}_p(X_1, \dots, X_k) & \text{if } \text{cost}_p(X_1, \dots, X_k) \leq B, \\ B + 1 & \text{otherwise.} \end{cases}$$

We define

$$\text{Opt}(\mathbf{X}, k) = \min\{\text{cost}_p(X_1, \dots, X_k) \mid \{X_1, \dots, X_k\} \text{ is an equal } k\text{-clustering of } \mathbf{X}\},$$

and given a nonnegative integer  $B$ ,

$$\text{Opt}(\mathbf{X}, k, B) = \min\{\text{cost}_p^B(X_1, \dots, X_k) \mid \{X_1, \dots, X_k\} \text{ is an equal } k\text{-clustering of } \mathbf{X}\}.$$

We conclude this section by the observation that, given vectors  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ , we can find an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  that minimizes  $\sum_{i=1}^k \text{cost}_p(X_i, \mathbf{c}_i)$

using a reduction to the classical MINIMUM WEIGHT PERFECT MATCHING problem on bipartite graphs that is well-known to be solvable in a polynomial time. Recall that a *matching*  $M$  of a graph  $G$  is a set of edges without common vertices. It is said that a matching  $M$  *saturates* a vertex  $v$  if  $M$  has an edge incident to  $v$ . A matching  $M$  is *perfect* if every vertex of  $G$  is saturated. The task of MINIMUM WEIGHT PERFECT MATCHING is, given a bipartite graph  $G$  and a weight function  $w : E(G) \rightarrow \mathbb{Z}_{\geq 0}$ , find a perfect matching  $M$  (if it exists) such that its weight  $w(M) = \sum_{e \in M} w(e)$  is minimum. The proof of the following lemma essentially repeats the proof of Lemma 1 of [50] but we provide it here for completeness.

**Lemma 1** *Let  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  be a collection of points of  $\mathbb{Z}^d$  and  $k$  be a positive integer such that  $n$  is divisible by  $k$ . Let also  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ . Then an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  of minimum  $\text{cost}(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k)$  can be found in a polynomial time.*

**Proof** Given  $\mathbf{X}$  and  $\mathbf{c}_1, \dots, \mathbf{c}_k$ , we construct the bipartite graph  $G$  as follows. Let  $s = \frac{n}{k}$ .

- For each  $i \in \{1, \dots, k\}$ , we construct a set of  $s$  vertices  $V_i = \{v_1^i, \dots, v_s^i\}$  corresponding to the median  $c_i$ . Denote  $V = \bigcup_{i=1}^k V_i$ .
- For each  $i \in \{1, \dots, n\}$ , construct a vertex  $u_i$  corresponding to the vector  $\mathbf{x}_i$  of  $\mathbf{X}$  and make  $u_i$  adjacent to the vertices of  $V$ . Denote  $U = \{u_1, \dots, u_n\}$ .

We define the edge weights as follows.

- For every  $i \in \{1, \dots, n\}$  and  $j \in \{1, \dots, k\}$ , set  $w(u_i v_h^j) = \|\mathbf{c}_j - \mathbf{x}_i\|_p$  for  $h \in \{1, \dots, s\}$ , that is, the weight of all edges joining  $u_i$  corresponding to  $\mathbf{x}_i$  with the vertices of  $V_j$  corresponding to the median  $\mathbf{c}_j$  are the same and coincide with the  $\ell_p$  distance between  $\mathbf{x}_i$  and  $\mathbf{c}_j$ .

Observe that  $G(U, V)$  is a complete bipartite graph, where  $U$  and  $V$  form the bipartition. Note also that  $|U| = |V| = n$ .

Notice that we have the following one-to-one correspondence between perfect matchings of  $G$  and  $k$ -clusterings of  $\mathbf{X}$ . In the forward direction, assume that  $M$  is a perfect matching of  $G$ . We construct the clustering  $\{X_1, \dots, X_k\}$  as follows. For every  $h \in \{1, \dots, n\}$ ,  $u_h$  is saturated by  $M$  and, therefore, there are  $i_h \in \{1, \dots, k\}$  and  $j_h \in \{1, \dots, s\}$  such that edge  $u_h v_{j_h}^{i_h} \in M$ . We cluster the vectors of  $\mathbf{X}$  according to  $M$ . Formally, we place  $x_h$  in  $X_{i_h}$  for each  $h \in \{1, \dots, n\}$ . Clearly,  $\{X_1, \dots, X_k\}$  is a partition of  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  and  $|X_i| = s$  for all  $i \in \{1, \dots, k\}$ . By the definition of weights of the edges of  $G$ ,  $\text{cost}_p(X_1, \dots, X_k, c_1, \dots, c_k) = w(M)$ . For the reverse direction, consider an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  of  $\mathbf{X}$ . Let  $i \in \{1, \dots, k\}$ . Consider the cluster  $X_i$  and assume that  $X_i = \{j_1, \dots, j_s\}$ . Denote by  $M_i = \{u_{j_1} v_1^i, \dots, u_{j_s} v_s^i\}$ . Clearly,  $M_i$  is a matching saturating the vertices of  $V_i$ . We construct  $M_i$  for every  $i \in \{1, \dots, k\}$  and set  $M = \bigcup_{i=1}^k M_i$ . Since  $\{X_1, \dots, X_k\}$  is a partition of  $\{1, \dots, n\}$ ,

$M$  is a matching saturating every vertex of  $U$ . By the definition of the weight of edges,  $w(M) = \text{cost}_p(X_1, \dots, X_k, c_1, \dots, c_k)$ . Thus, finding a  $k$ -clustering  $\{X_1, \dots, X_k\}$  that minimizes  $\text{cost}_p(X_1, \dots, X_k), c_1, \dots, c_k$  is equivalent to computing a perfect matching of minimum weight in  $G$ . Then, because a perfect matching of minimum weight in  $G$  can be found in a polynomial time [51, 52], a  $k$ -clustering of minimum cost can be found in a polynomial time. This completes the proof of the lemma.  $\square$

### 3 Lossy Kernel

In this section, we prove Theorem 1 by establishing a 2-approximate polynomial kernel for PARAMETERIZED EQUAL CLUSTERING. In Subsection 3.1, we provide some auxiliary results, and in Subsection 3.2, we prove the main results. Throughout this section, we assume that  $p \geq 0$  defining the  $\ell_p$ -norm is a fixed constant.

#### 3.1 Technical Lemmata

We start by proving the following results about the medians of clusters when their size is sufficiently big with respect to the budget.

**Lemma 2** *Let  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering of a collection of points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $\mathbb{Z}^d$  of cost at most  $B \in \mathbb{Z}_{\geq 0}$ , and let  $s = \frac{n}{k}$ . Then each cluster  $X_i$  for  $i \in \{1, \dots, k\}$  contains at least  $s - 2B$  equal points.*

**Proof** The claim is trivial if  $s \leq 2B + 1$ . Let  $s \geq 2B + 2$ . Assume to the contrary that a cluster  $X_i$  has at most  $s - 2B - 1$  equal points for some  $i \in \{1, \dots, k\}$ . Let  $\mathbf{c}_1, \dots, \mathbf{c}_k$  be optimum medians for the clusters  $X_1, \dots, X_k$ , respectively. Then we have that  $\text{cost}_p(X_1, \dots, X_k) = \text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k)$ .

Let  $\mathbf{x}_{i_0} \in X_i$  be a point at the minimum distance from  $\mathbf{c}_i$ . Since there are at most  $s - 2B - 1$  points in  $X_i$  which are equal to  $\mathbf{x}_{i_0}$ , there are  $t = 2B + 1$  points  $\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_t} \in X_i$  distinct from  $\mathbf{x}_{i_0}$ . Observe that

$$\sum_{\mathbf{x}_h \in X_i} \|\mathbf{c}_i - \mathbf{x}_h\|_p \geq \sum_{j=0}^t \|\mathbf{c}_i - \mathbf{x}_{i_j}\|_p \geq \sum_{j=1}^t \|\mathbf{c}_i - \mathbf{x}_{i_j}\|_p. \quad (1)$$

Because the points have integer coordinates and by the triangle inequality,

$$1 \leq \|\mathbf{x}_{i_0} - \mathbf{x}_{i_j}\|_p \leq \|\mathbf{x}_{i_0} - \mathbf{c}_i\|_p + \|\mathbf{x}_{i_j} - \mathbf{c}_i\|_p \quad (2)$$

for every  $j \in \{1, \dots, t\}$ . Since  $\mathbf{x}_{i_0}$  is a point of  $X_i$  at minimum distance from  $\mathbf{c}_i$ ,

$$\|\mathbf{x}_{i_0} - \mathbf{c}_i\|_p + \|\mathbf{x}_{i_j} - \mathbf{c}_i\|_p \leq 2\|\mathbf{x}_{i_j} - \mathbf{c}_i\|_p. \quad (3)$$

From (2) and (3), we get  $\|\mathbf{x}_{i_j} - \mathbf{c}_i\|_p \geq \frac{1}{2}$  for  $j \in \{1, \dots, t\}$ . Thus, from (1), we get

$$\sum_{\mathbf{x}_h \in X_i} \|\mathbf{c}_i - \mathbf{x}_h\|_p \geq \sum_{j=1}^t \|\mathbf{c}_i - \mathbf{x}_{i_j}\|_p \geq \frac{1}{2}t = \frac{1}{2}(2B + 1) > B,$$

which is a contradiction with  $\text{cost}_p(X_1, \dots, X_k) \leq B$ .  $\square$  This completes the proof.

**Lemma 3** *Let  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering of a collection of points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $\mathbb{Z}^d$  of cost at most  $B \in \mathbb{Z}_{\geq 0}$ , and let  $s = \frac{n}{k} \geq 4B + 1$ . Let also  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$  be optimum medians for  $X_1, \dots, X_k$ , respectively. Then for every  $i \in \{1, \dots, k\}$ ,  $\mathbf{c}_i = \mathbf{x}_j$  for  $\mathbf{x}_j \in X_i$  such that  $X_i$  contains at least  $s - 2B$  points that are equal to  $\mathbf{x}_j$  and the choice of  $\mathbf{c}_i$  is unique.*

**Proof** Consider a cluster  $X_i$  with the median  $\mathbf{c}_i$  for arbitrary  $i \in \{1, \dots, k\}$ . Since  $s \geq 4B + 1$ , then by Lemma 2, there is  $\mathbf{x}_j \in X_i$  such that  $X_i$  contains at least  $s - 2B$  points that are equal to  $\mathbf{x}_j$ . We show that  $\mathbf{c}_i = \mathbf{x}_j$ . Notice that the choice of the set of at least  $s - 2B$  equal points is unique because  $X_i$  can contain at most  $s - (s - 2B) = 2B$  distinct from  $\mathbf{x}_j$  points, and since  $s \geq 4B + 1$ ,  $s - 2B \geq 2B + 1 > 2B$ .

The proof is by contradiction. Assume that  $\mathbf{c}_i \neq \mathbf{x}_j$ . Let  $S \subseteq \{1, \dots, n\}$  be the set of indices of the points  $\mathbf{x}_h \in X_i$  that coincide with  $\mathbf{x}_j$ , and denote by  $T$  the set of indices of the remaining points in  $X_i$ . We know that  $|T| \leq 2B < |S|$  because  $s \geq 4B + 1$  and  $|S| \geq 2B + 1$ . Then

$$\begin{aligned} \text{cost}_p(X_i) &= \text{cost}_p(X_i, \mathbf{c}_i) = \sum_{h \in X_i} \|\mathbf{c}_i - \mathbf{x}_h\|_p \\ &= \sum_{h \in S} \|\mathbf{c}_i - \mathbf{x}_h\|_p + \sum_{h \in T} \|\mathbf{c}_i - \mathbf{x}_h\|_p \\ &= (|S| - |T|)\|\mathbf{c}_i - \mathbf{x}_j\|_p + \sum_{h \in T} (\|\mathbf{c}_i - \mathbf{x}_j\|_p + \|\mathbf{c}_i - \mathbf{x}_h\|_p). \end{aligned} \tag{4}$$

On using the triangle inequality, we get

$$\begin{aligned} (|S| - |T|)\|\mathbf{c}_i - \mathbf{x}_j\|_p + \sum_{h \in T} (\|\mathbf{c}_i - \mathbf{x}_j\|_p + \|\mathbf{c}_i - \mathbf{x}_h\|_p) \\ \geq (|S| - |T|)\|\mathbf{c}_i - \mathbf{x}_j\|_p + \sum_{h \in T} \|\mathbf{x}_j - \mathbf{x}_h\|_p. \end{aligned} \tag{5}$$

We know that  $(|S| - |T|)\|\mathbf{c}_i - \mathbf{x}_j\|_p > 0$  because  $|S| > |T|$  and  $\mathbf{c}_i \neq \mathbf{x}_j$ . Then by (5), we have

$$(|S| - |T|)\|\mathbf{c}_i - \mathbf{x}_j\|_p + \sum_{h \in T} \|\mathbf{x}_j - \mathbf{x}_h\|_p > \sum_{h \in T} \|\mathbf{x}_j - \mathbf{x}_h\|_p. \tag{6}$$

Combining (4)–(6), we conclude that  $\text{cost}_p(X_i) > \sum_{h \in T} \|\mathbf{x}_j - \mathbf{x}_h\|_p$ . Let  $\mathbf{c}'_i = \mathbf{x}_j$ .

Then

$$\begin{aligned} \text{cost}_p(X_i, \mathbf{c}'_i) &= \sum_{h \in X_i} \|\mathbf{c}'_i - \mathbf{x}_h\|_p = \sum_{h \in S} \|\mathbf{c}'_i - \mathbf{x}_h\|_p + \sum_{h \in T} \|\mathbf{c}'_i - \mathbf{x}_h\|_p \\ &= \sum_{h \in T} \|\mathbf{c}'_i - \mathbf{x}_h\|_p < \text{cost}_p(X_i) \end{aligned}$$

which contradicts that  $\mathbf{c}_i$  is an optimum median for  $X_i$ . This concludes the proof.  $\square$

We use the following lemma to identify medians.

**Lemma 4** *Let  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering of a collection of points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $\mathbb{Z}^d$  of cost at most  $B \in \mathbb{Z}_{\geq 0}$ , and let  $s = \frac{n}{k} \geq 4B + 1$ . Suppose that  $Y \subseteq \mathbf{X}$  is a collection of at least  $B + 1$  equal points of  $\mathbf{X}$ . Then there is  $i \in \{1, \dots, k\}$  such that an optimum median of  $X_i$  coincides with  $\mathbf{x}_j$  for  $\mathbf{x}_j \in Y$ .*

**Proof** Let  $\mathbf{c}_1, \dots, \mathbf{c}_k$  be optimum medians of  $X_1, \dots, X_k$ , respectively. Since  $s \geq 4B + 1$ , then by Lemma 3, for every  $i \in \{1, \dots, k\}$ ,  $\mathbf{c}_i$  coincides with some element  $\mathbf{x}_{h_i}$  of the cluster  $X_i$ . For the sake of contradiction, assume that  $\mathbf{c}_1, \dots, \mathbf{c}_k$  are distinct from  $\mathbf{x}_j \in Y$ . This means that  $\|\mathbf{x}_j - \mathbf{c}_i\|_p \geq 1$  because the coordinates of the points of  $\mathbf{X}$  are integer. Then

$$\begin{aligned} \text{cost}_p(X_1, \dots, X_k) &= \sum_{i=1}^k \text{cost}_p(X_i, \mathbf{c}_i) \geq \sum_{i=1}^k \sum_{\mathbf{x}_h \in Y \cap X_i} \|\mathbf{c}_i - \mathbf{x}_h\|_p \geq \sum_{i=1}^k |X_i \cap Y| \\ &= |Y| \geq B + 1 > B, \end{aligned}$$

contradicting that  $\text{cost}_p(X_1, \dots, X_k) \leq B$ . This proves the lemma.  $\square$

We use our next lemma to upper bound the clustering cost if we collect  $s = \frac{n}{k}$  equal points in the same cluster.

**Lemma 5** *Let  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering of a collection of points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $\mathbb{Z}^d$ , and let  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ . Suppose that  $S$  is a collection of  $s = \frac{n}{k}$  equal points of  $\mathbf{X}$  and  $\mathbf{x}_j \in S$ . Then there is an equal  $k$ -clustering  $\{X'_1, \dots, X'_k\}$  of  $\mathbf{X}$  with  $X'_1 = S$  such that*

$$\text{cost}_p(X'_1, \dots, X'_k, \mathbf{c}'_1, \dots, \mathbf{c}'_k) \leq \text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k) + s\|\mathbf{c}_1 - \mathbf{x}_j\|_p,$$

where  $\mathbf{c}'_1 = \mathbf{x}_j$  and  $\mathbf{c}'_h = \mathbf{c}_h$  for  $h \in \{2, \dots, k\}$ .

**Proof** The claim is trivial if  $S = X_1$  because we can set  $X'_i = X_i$  for  $i \in \{1, \dots, k\}$ . Assume that this is not the case and there are elements of  $S$  that are not in  $X_1$ ; denote by  $\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_t}$  these elements. We assume that  $\mathbf{x}_{i_h} \in X_{i'_h}$ , for  $h \in \{1, \dots, t\}$  for  $i'_h \geq 2$ . Because  $|S| = s$ , there are  $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_t} \in X_1$  such that  $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_t} \notin S$ . We construct  $X'_1, \dots, X'_k$  from  $X_1, \dots, X_k$  by exchanging the points  $\mathbf{x}_{j_h}$  and  $\mathbf{x}_{i_h}$  between  $X_1$  and  $X_{i'_h}$  for every  $h \in \{1, \dots, t\}$ . Notice that  $|X'_1| = \dots = |X'_k|$  because the exchanges

do not modify the sizes of the clusters. Thus,  $\{X'_1, \dots, X'_k\}$  is an equal  $k$ -clustering. We claim that  $\{X'_1, \dots, X'_k\}$  satisfies the required property.

We have that

$$\begin{aligned} & \text{cost}(X'_1, \dots, X'_k, \mathbf{c}'_1, \dots, \mathbf{c}'_k) - \text{cost}(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k) \\ &= \sum_{i=1}^k \sum_{\mathbf{x}_h \in X'_i} \|\mathbf{x}_h - \mathbf{c}'_i\|_p - \sum_{i=1}^k \sum_{\mathbf{x}_h \in X_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p \\ &= \sum_{\mathbf{x}_h \in X'_1} \|\mathbf{x}_h - \mathbf{c}'_1\|_p - \sum_{\mathbf{x}_h \in X_1} \|\mathbf{x}_h - \mathbf{c}_1\|_p \tag{7} \\ &+ \sum_{i=2}^k \left( \sum_{\mathbf{x}_h \in X'_i} \|\mathbf{x}_h - \mathbf{c}'_i\|_p - \sum_{\mathbf{x}_h \in X_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p \right). \end{aligned}$$

Note that  $\sum_{\mathbf{x}_h \in X'_1} \|\mathbf{x}_h - \mathbf{c}'_1\|_p = 0$  and  $\sum_{\mathbf{x}_h \in X_1} \|\mathbf{x}_h - \mathbf{c}_1\|_p \geq \sum_{h=1}^t \|\mathbf{x}_{j_h} - \mathbf{c}_1\|_p$ . Also by the construction of  $X'_1, \dots, X'_k$  and because  $\mathbf{c}_i = \mathbf{c}'_i$  for  $i \in \{2, \dots, k\}$ , we have that

$$\begin{aligned} \sum_{i=2}^k \left( \sum_{\mathbf{x}_h \in X'_i} \|\mathbf{x}_h - \mathbf{c}'_i\|_p - \sum_{\mathbf{x}_h \in X_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p \right) &= \sum_{h=1}^t \|\mathbf{x}_{j_h} - \mathbf{c}'_{i_h}\|_p - \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{i_h}\|_p \\ &= \sum_{h=1}^t \|\mathbf{x}_{j_h} - \mathbf{c}_{i_h}\|_p - \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{i_h}\|_p. \end{aligned}$$

Then extending (7) and applying the triangle inequality twice, we obtain that

$$\begin{aligned} & \text{cost}(X'_1, \dots, X'_k, \mathbf{c}'_1, \dots, \mathbf{c}'_k) - \text{cost}(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k) \\ &\leq - \sum_{h=1}^t \|\mathbf{x}_{j_h} - \mathbf{c}_1\|_p + \sum_{h=1}^t \|\mathbf{x}_{j_h} - \mathbf{c}_{i_h}\|_p - \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{i_h}\|_p \\ &= \sum_{h=1}^t \left( - \|\mathbf{x}_{j_h} - \mathbf{c}_1\|_p + \|\mathbf{x}_{j_h} - \mathbf{c}_{i_h}\|_p - \|\mathbf{x}_{i_h} - \mathbf{c}_{i_h}\|_p \right) \\ &\leq \sum_{h=1}^t \left( \|\mathbf{x}_{i_h} - \mathbf{c}_{i_h}\|_p - \|\mathbf{c}_1 - \mathbf{c}_{i_h}\|_p \right) \\ &\leq \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_1\|_p \leq t \|\mathbf{x}_j - \mathbf{c}_1\|_p \leq s \|\mathbf{x}_j - \mathbf{c}_1\|_p \end{aligned}$$

as required by the lemma. □

Our next lemma shows that we can solve PARAMETERIZED EQUAL CLUSTERING in a polynomial time if the cluster size is sufficiently big with respect to the budget.

**Lemma 6** *There is a polynomial-time algorithm that, given a collection  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $n$  points of  $\mathbb{Z}^d$ , a positive integer  $k$  such that  $n$  is divisible by  $k$ , and a nonnegative integer  $B$  such that  $\frac{n}{k} \geq 4B + 1$ , either computes  $\text{Opt}(\mathbf{X}, k) \leq B$  and produces an equal  $k$ -clustering of minimum cost or correctly concludes that  $\text{Opt}(\mathbf{X}, k) > B$ .*

**Proof** Let  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  be a collection of  $n$  points of  $\mathbb{Z}^d$  and let  $k$  be a positive integer such that  $n$  is divisible by  $k$ , and suppose that  $s = \frac{n}{k} \geq 4B + 1$  for a nonnegative integer  $B$ .

First, we exhaustively apply the following reduction rule.

**Reduction Rule 1** If  $\mathbf{X}$  contains a collection of  $s$  equal points  $S$ , then set  $\mathbf{X} := \mathbf{X} \setminus S$  and  $k := k - 1$ .

To argue that the rule is safe, let  $\mathbf{X}' = \mathbf{X} \setminus S$ , where  $S$  is a collection of  $s$  equal points of  $\mathbf{X}$ , and let  $k' = k$ . Clearly,  $\mathbf{X}'$  contains  $n' = n - s$  points and  $\frac{n'}{k'} = s$ . If  $\{X'_1, \dots, X'_{k'}\}$  is an equal  $k'$ -clustering of  $\mathbf{X}'$ , then  $\{S, X'_1, \dots, X'_{k'}\}$  is an equal  $k$ -clustering of  $\mathbf{X}$ . Note that  $\text{cost}_p(S) = 0$  because the elements of  $S$  are the same. Then  $\text{cost}_p(S, X'_1, \dots, X'_{k'}) = \text{cost}_p(X'_1, \dots, X'_{k'})$ . Therefore,  $\text{Opt}(\mathbf{X}, k) \leq \text{Opt}(\mathbf{X}', k')$ . We show that if  $\text{Opt}(\mathbf{X}, k) \leq B$ , then  $\text{Opt}(\mathbf{X}, k) \geq \text{Opt}(\mathbf{X}', k')$ .

Suppose that  $\{X_1, \dots, X_k\}$  is an equal  $k$ -clustering of  $\mathbf{X}$  with  $\text{cost}_p(X_1, \dots, X_k) = \text{Opt}(\mathbf{X}, k) \leq B$ . Denote by  $\mathbf{c}_1, \dots, \mathbf{c}_k$  optimum medians of  $X_1, \dots, X_k$ , respectively. Because  $|S| = s \geq 4B + 1 \geq B + 1$ , there is a cluster whose optimum median is  $\mathbf{x}_j$  for  $\mathbf{x}_j \in S$ . We assume without loss of generality that  $X_1$  is such a cluster and  $\mathbf{c}_1 = \mathbf{x}_j$ . By Lemma 5, there is a  $k$ -clustering  $\{S, X'_2, \dots, X'_k\}$  of  $\mathbf{X}$  such that  $\text{cost}_p(S, X'_2, \dots, X'_k, \mathbf{c}'_1, \dots, \mathbf{c}'_k) \leq \text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k) + s \|\mathbf{c}_1 - \mathbf{x}_j\|_p$ , where  $\mathbf{c}'_1 = \mathbf{x}_j$  and  $\mathbf{c}'_h = \mathbf{c}_h$  for  $h \in \{2, \dots, k\}$ . Because  $\mathbf{c}_1 = \mathbf{x}_j$ , we conclude that  $\text{cost}_p(X'_2, \dots, X'_k) = \text{cost}_p(S, X'_2, \dots, X'_k, \mathbf{c}'_1, \dots, \mathbf{c}'_k) \leq \text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k) = \text{Opt}(\mathbf{X}, k)$ . Since  $\{X'_2, \dots, X'_k\}$  is a  $k'$ -clustering of  $\mathbf{X}'$ , we have that  $\text{Opt}(\mathbf{X}', k') \leq \text{cost}_p(X'_2, \dots, X'_k) \leq \text{Opt}(\mathbf{X}, k)$  as required.

We obtain that either  $\text{Opt}(\mathbf{X}, k) = \text{Opt}(\mathbf{X}', k') \leq B$  or  $\text{Opt}(\mathbf{X}, k) > B$  and  $\text{Opt}(\mathbf{X}', k') > B$ . Notice also that, given an optimum equal  $k'$ -clustering of  $\mathbf{X}'$ , we can construct the optimum  $k$ -clustering of  $\mathbf{X}$ , by making  $S$  a cluster. Thus, it is sufficient to prove the lemma for the collection of points obtained by the exhaustive application of Reduction Rule 1. Note that if this collection is empty, then  $\text{Opt}(\mathbf{X}, k) = 0$  and the lemma holds. This allows us to assume from now that  $\mathbf{X}$  is nonempty and has no  $s$  equal points.

Suppose that  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering with  $\text{cost}_p(X_1, \dots, X_k) = \text{Opt}(\mathbf{X}, k) \leq B$ . By Lemma 3, we have that for every  $i \in \{1, \dots, k\}$ , the optimum median  $\mathbf{c}_i$  for  $X_i$  is unique and  $\mathbf{c}_i = \mathbf{x}_j$  for  $\mathbf{x}_j \in \mathbf{X}_i$  such that  $\mathbf{X}_i$  contains at least  $s - 2B$  points that are equal to  $\mathbf{x}_j$ . Notice that  $\mathbf{c}_1, \dots, \mathbf{c}_k$  are pairwise distinct because a collection of equal points cannot be split between distinct clusters in such a way that each of these clusters would contain at least  $s - 2B$  points. This holds because any collection of equal points of  $\mathbf{X}$  contains at most  $s - 1$  elements and  $2(s - 2B) > s$  as  $s \geq 4B + 1$ . By Lemma 4, we have that if  $\mathbf{X}$  contains a collection of equal points  $S$  of size  $B + 1 \leq s - 2B$ , then one of the optimum medians should be equal to a point from  $S$ .



These observations allow us to construct (potential) medians  $\mathbf{c}_1, \dots, \mathbf{c}_t$  as follows: we iteratively compute inclusion maximal collections  $S$  of equal points of  $\mathbf{X}$  and if  $|S| \geq B + 1$ , we set the next median  $\mathbf{c}_i$  be equal to a point of  $S$ . If the number of constructed potential medians  $t \neq k$ , we conclude that  $\mathbf{X}$  has no equal  $k$ -clustering of cost at most  $B$ . Otherwise, if  $t = k$ , we have that  $\mathbf{c}_1, \dots, \mathbf{c}_k$  should be optimum medians for an equal  $k$ -clustering of minimum cost if  $\text{Opt}(\mathbf{X}, k) \leq B$ .

Then we compute in a polynomial time an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  of  $\mathbf{X}$  that minimizes  $\text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k)$  using Lemma 1. If  $\text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k) > B$ , then we conclude that  $\text{Opt}(\mathbf{X}, k) > B$ . Otherwise, we have that  $\text{Opt}(\mathbf{X}, k) = \text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k)$  and  $\{X_1, \dots, X_k\}$  is an equal  $k$ -clustering of minimum cost.  $\square$

Our next aim is to show that we can reduce the dimension and the absolute values of the coordinates of the points if  $\text{Opt}(X, k) \leq B$ . To achieve this, we mimic some ideas of the kernelization algorithm of Fomin et al. in [53] for the related clustering problem. However, they considered only points from  $\{0, 1\}^d$  and the Hamming norm.

**Lemma 7** *There is a polynomial-time algorithm that, given a collection  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $n$  points of  $\mathbb{Z}^d$ , a positive integer  $k$  such that  $n$  is divisible by  $k$ , and a nonnegative integer  $B$ , either correctly concludes that  $\text{Opt}(\mathbf{X}, k) > B$  or computes a collection of  $n$  points  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$  of  $\mathbb{Z}^{d'}$  such that the following holds:*

- i For every partition  $\{I_1, \dots, I_k\}$  of  $\{1, \dots, n\}$  such that  $|I_1| = \dots = |I_k| = \frac{n}{k}$ , either  $\text{cost}_p(X_1, \dots, X_k) > B$  and  $\text{cost}_p(Y_1, \dots, Y_k) > B$  or  $\text{cost}_p(X_1, \dots, X_k) = \text{cost}_p(Y_1, \dots, Y_k)$ , where  $X_i = \{\mathbf{x}_h \mid h \in I_i\}$  and  $Y_i = \{\mathbf{y}_h \mid h \in I_i\}$  for every  $i \in \{1, \dots, k\}$ .*
- ii  $d' = \mathcal{O}(kB^{p+1})$ .*
- iii  $|y_i[h]| = \mathcal{O}(kB^2)$  for  $h \in \{1, \dots, d'\}$  and  $i \in \{1, \dots, n\}$ .*

**Proof** Let  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  be a collection of  $n$  points of  $\mathbb{Z}^d$  and let  $k$  be a positive integer such that  $n$  is divisible by  $k$ . Let also  $B$  be a nonnegative integer.

We iteratively construct the partition  $S = \{S_1, \dots, S_t\}$  of  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  using the following greedy algorithm. Let  $j \geq 1$  be an integer and suppose that the sets  $S_0, \dots, S_{j-1}$  are already constructed assuming that  $S_0 = \emptyset$ . Let  $\mathbf{Z} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \setminus \cup_{i=0}^{j-1} S_i$ . If  $\mathbf{Z} = \emptyset$ , then the construction of  $S$  is completed. If  $\mathbf{Z} \neq \emptyset$ , we construct  $S_j$  as follows:

- set  $S_j := \{\mathbf{x}_h\}$  for arbitrary  $\mathbf{x}_h \in \mathbf{Z}$  and set  $\mathbf{Z} := \mathbf{Z} \setminus \{\mathbf{x}_h\}$ ,
- while there is  $\mathbf{x}_r \in \mathbf{Z}$  such that  $\|\mathbf{x}_r - \mathbf{x}_{r'}\|_p \leq B$  for some  $\mathbf{x}_{r'} \in S_j$ , set  $S_j := S_j \cup \{\mathbf{x}_r\}$  and set  $\mathbf{Z} = \mathbf{Z} \setminus \{\mathbf{x}_r\}$ .

The crucial property of the partition  $S$  is that every cluster of an equal  $k$ -clustering of cost at most  $B$  is entirely in some part of the partition.

**Claim 3.1** Let  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering of  $\mathbf{X}$  of cost at most  $B$ . Then for every  $i \in \{1, \dots, k\}$  there is  $j \in \{1, \dots, t\}$  such that  $X_i \subseteq S_j$ .

**Proof of Claim 3.1** Denote  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$  the optimum medians for the clusters  $X_1, \dots, X_k$ , respectively. Assume to the contrary that there is a cluster  $X_i$  such that

$\mathbf{x}_u, \mathbf{x}_v \in X_i$  with  $\mathbf{x}_u$  and  $\mathbf{x}_v$  in distinct collections of the partition  $\{S_1, \dots, S_t\}$ . Then  $\|\mathbf{x}_u - \mathbf{x}_v\|_p > B$  by the construction of  $S_1, \dots, S_t$  and

$$\begin{aligned} \text{cost}_p(X_1, \dots, X_k) &\geq \text{cost}_p(X_i) = \text{cost}_p(X_i, \mathbf{c}_i) \geq \|\mathbf{c}_i - \mathbf{x}_u\|_p + \|\mathbf{c}_i - \mathbf{x}_v\|_p \\ &\geq \|\mathbf{x}_u - \mathbf{x}_v\|_p > B \end{aligned}$$

contradicting that  $\text{cost}_p(X_1, \dots, X_k) \leq B$ . □

From the above Claim 3.1, we have that if  $t > k$ , then  $\mathbf{X}$  has no equal  $k$ -clustering of cost at most  $B$ , that is,  $\text{Opt}(X, B) > B$ . In this case, we return this answer and stop. From now on, we assume that this is not the case and  $t \leq k$ .

By Lemma 2, at least  $\frac{n}{k} - 2B$  points in every cluster of an equal  $k$ -clustering of cost at most  $B$  are the same. Thus, if  $\{X_1, \dots, X_k\}$  is an equal  $k$ -clustering of cost at most  $B$ , then for each  $i \in \{1, \dots, k\}$ ,  $X_i$  contains at most  $2B + 1$  distinct points. By Claim 3.1, we obtain that for every  $i \in \{1, \dots, t\}$ ,  $S_i$  should contain at most  $k(2B + 1)$  distinct points if  $\mathbf{X}$  admits an equal  $k$ -clustering of cost at most  $B$ . Then for each  $i \in \{1, \dots, t\}$ , we compute the number of distinct points in  $S_i$  and if this number is bigger than  $k(2B + 1)$ , we conclude that  $\text{Opt}(\mathbf{X}, k) > B$ . In this case, we return this answer and stop. From now, we assume that this is not the case ‘  $S_i$  for  $i \in \{1, \dots, t\}$  contains at most  $k(2B + 1)$  distinct points.

For a collection of points  $Z \subseteq \mathbf{X}$ , we say that a coordinate  $h \in \{1, \dots, d\}$  is *uniform* for  $Z$  if  $\mathbf{x}_j[h]$  is the same for all  $\mathbf{x}_h \in Z$  and  $h$  is *nonuniform* otherwise.

Let  $\ell_i$  be the number of nonuniform coordinates for  $S_i$  for  $i \in \{1, \dots, t\}$ , and let  $\ell = \max_{1 \leq i \leq t} \ell_i$ . For each  $i \in \{1, \dots, t\}$ , we select a set of indices  $R_i \subseteq \{1, \dots, d\}$  of size  $\ell$  such that  $R_i$  contains all nonuniform coordinates for  $S_i$ . Note that  $R_i$  may be empty if  $\ell = 0$ . We also define a set of coordinates  $T_i = \{1, \dots, d\} \setminus R_i$ , for  $i \in \{1, \dots, t\}$ .

For every  $i \in \{1, \dots, n\}$  and  $j \in \{1, \dots, t\}$  such that  $x_i \in S_j$ , we define an  $(\ell + 1)$ -dimensional point  $\mathbf{x}'_i$ , where  $\mathbf{x}'_i[1, \dots, \ell] = \mathbf{x}_i[R_j]$  and  $\mathbf{x}'_i[\ell + 1] = (j - 1)(B + 1)$ . This way we obtain a collection of points  $\mathbf{X}' = \{\mathbf{x}'_1, \dots, \mathbf{x}'_n\}$ . For every  $j \in \{1, \dots, t\}$ , we define  $S'_j = \{\mathbf{x}'_h \mid \mathbf{x}_h \in S_j\}$ , that is, we construct the partition  $S' = \{S'_1, \dots, S'_t\}$  of  $\{\mathbf{x}'_1, \dots, \mathbf{x}'_n\}$  corresponding to  $S$ .

For each  $i \in \{1, \dots, t\}$ , we do the following:

- For each  $h \in \{1, \dots, \ell\}$ , we find  $M_h^{(i)} = \min\{\mathbf{x}'_j[h] \mid \mathbf{x}'_j \in S'_i\}$ .
- For every  $\mathbf{x}'_j \in S'_i$ , we define a new point  $\mathbf{y}_j$  by setting  $\mathbf{y}_j[h] = \mathbf{x}'_j[h] - M_h^{(i)}$  for  $h \in \{1, \dots, \ell\}$  and  $\mathbf{y}_j[\ell + 1] = \mathbf{x}'_j[\ell + 1] = (j - 1)(B + 1)$ .

This way, we construct the collection  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$  of points from  $\mathbb{Z}^{\ell+1}$ . Our algorithm returns this collection of points.

It is easy to see that the described algorithm runs in a polynomial time. We show that if the algorithm outputs  $\mathbf{Y}$ , then this collection of the points satisfies conditions (i)–(iii) of the lemma.

To show (i), let  $\{I_1, \dots, I_k\}$  be a partition of  $\{1, \dots, n\}$  such that  $|I_1| = \dots = |I_k| = \frac{n}{k}$ , and let  $X_i = \{\mathbf{x}_h \mid h \in I_i\}$  and  $Y_i = \{\mathbf{y}_h \mid h \in I_i\}$  for every  $i \in \{1, \dots, k\}$ . We show that either  $\text{cost}_p(X_1, \dots, X_k) > B$  and  $\text{cost}_p(Y_1, \dots, Y_k) > B$  or  $\text{cost}_p(X_1, \dots, X_k) = \text{cost}_p(Y_1, \dots, Y_k)$ .

Suppose that  $\text{cost}_p(X_1, \dots, X_k) \leq B$ . Consider  $i \in \{1, \dots, k\}$  and denote by  $\mathbf{c}_i$  the optimum median for  $X_i$ . By Claim 3.1, there is  $j \in \{1, \dots, t\}$  such that  $X_i \subseteq S_j$ . We define  $\mathbf{c}'_i \in \mathbb{R}^{\ell+1}$  by setting  $\mathbf{c}'_i[1, \dots, \ell] = \mathbf{c}_i[R_j]$  and  $\mathbf{c}'_i[\ell + 1] = (j - 1)(B + 1)$ . Further, we consider  $\mathbf{c}''_i \in \mathbb{R}^{\ell+1}$  such that  $\mathbf{c}''_i[h] = \mathbf{c}'_i[h] - M_h^{(j)}$  for  $h \in \{1, \dots, \ell\}$  and  $\mathbf{c}''_i[\ell + 1] = (j - 1)(B + 1)$ . Then by the definitions of  $\mathbf{X}'_i$  and  $\mathbf{Y}_i$ , we have that

$$\text{cost}_p(X_i) = \text{cost}_p(X_i, \mathbf{c}_i) = \text{cost}_p(X'_i, \mathbf{c}'_i) = \text{cost}_p(Y_i, \mathbf{c}''_i) \geq \text{cost}_p(Y_i).$$

This implies that  $\text{cost}_p(X_1, \dots, X_k) \geq \text{cost}_p(Y_1, \dots, Y_k)$ .

For the opposite direction, assume that  $\text{cost}_p(X_1, \dots, X_k) \leq B$ . Similarly to  $S'$ , for every  $j \in \{1, \dots, t\}$ , we define  $S''_j = \{\mathbf{y}_h \mid \mathbf{x}_h \in S_j\}$ , that is, we construct the partition  $S'' = \{S''_1, \dots, S''_t\}$  of  $\mathbf{Y}$  corresponding to  $S$ . We claim that for each  $i \in \{1, \dots, k\}$ , there is  $j \in \{1, \dots, t\}$  such that  $Y_i \subseteq S_j$ .

The proof is by contradiction and is similar to the proof of Claim 3.1. Assume that there is  $i \in \{1, \dots, k\}$  such that there are  $\mathbf{y}_u, \mathbf{y}_v \in Y_i$  belonging to distinct sets of  $S''$ . Then  $\|\mathbf{y}_u - \mathbf{y}_v\|_p \geq |\mathbf{y}_u[\ell + 1] - \mathbf{y}_v[\ell + 1]| > B$  by the construction of  $S''_1, \dots, S''_t$ . Then

$$\begin{aligned} \text{cost}_p(Y_1, \dots, Y_k) &\geq \text{cost}_p(Y_i) = \text{cost}_p(Y_i, \mathbf{c}_i) \geq \|\mathbf{c}_i - \mathbf{x}_u\|_p + \|\mathbf{c}_i - \mathbf{x}_v\|_p \\ &\geq \|\mathbf{x}_u - \mathbf{x}_v\|_p > B, \end{aligned}$$

where  $\mathbf{c}_i$  is an optimum median of  $Y_i$ . However, this contradicts that  $\text{cost}_p(Y_1, \dots, Y_k) \leq B$ .

Consider  $i \in \{1, \dots, k\}$  and let  $\mathbf{c}''_i \in \mathbb{R}^{\ell+1}$  an optimum median for  $Y_i$ . Let also  $j \in \{1, \dots, t\}$  be such that  $Y_i \subseteq S_j$ . Notice that  $\mathbf{c}''_i[\ell + 1] = (j - 1)(B + 1)$  by the definition of  $S_j$ . We define  $\mathbf{c}'_i \in \mathbb{R}^{\ell+1}$  by setting  $\mathbf{c}'_i[h] = \mathbf{c}''_i[h] + M_h^{(j)}$  for  $h \in \{1, \dots, \ell\}$  and  $\mathbf{c}'_i[\ell + 1] = \mathbf{c}''_i[\ell + 1] = (j - 1)(B + 1)$ . Then we define  $\mathbf{c}_i \in \mathbb{R}^d$ , we setting  $\mathbf{c}_i[R_j] = \mathbf{c}'_i[1, \dots, \ell]$  and  $\mathbf{c}_i[T_j] = \mathbf{x}_h[T_j]$  for arbitrary  $\mathbf{x}_h \in S_j$ . Because the coordinates in  $T_j$  are uniform for  $S_j$ , the values in each coordinate  $h \in T_j$  of the coordinates of the points of  $X_i$  are the same. This implies that

$$\text{cost}_p(X_i) \leq \text{cost}_p(X_i, \mathbf{c}_i) = \text{cost}_p(X'_i, \mathbf{c}'_i) = \text{cost}_p(Y_i, \mathbf{c}''_i) = \text{cost}_p(Y_i).$$

Hence,  $\text{cost}_p(X_1, \dots, X_k) \leq \text{cost}_p(Y_1, \dots, Y_k)$ . This completes the proof of (i).

To show (ii), we prove that  $\ell \leq kB^p(2B + 1)$ . For this, we show that  $\ell_i \leq kB^p(2B + 1)$  for every  $i \in \{1, \dots, t\}$ . Consider  $i \in \{1, \dots, t\}$ . Recall that  $S_i$  contains at most  $k(2B + 1)$  distinct points. Denote by  $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_p}$  the distinct points in  $X_i$  and assume that they are numbered in the order in which they are included in  $S_i$  by the greedy procedure constructing this set.

Let  $Z_q = \{\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_q}\}$  for  $q \in \{1, \dots, r\}$ . We claim that  $Z_q$  has at most  $(q - 1)B^p$  nonuniform coordinates for each  $q \in \{1, \dots, r\}$ . The proof is by induction. The claim is trivial if  $q = 1$ . Let  $q > 1$  and assume that the claim is fulfilled for  $Z_{q-1}$ . By the construction of  $S_i$ ,  $\mathbf{x}_{j_q}$  is at distance at most  $B$  from  $\mathbf{x}_{j_h}$  for some  $h \in \{1, \dots, q - 1\}$ . Then because  $\|\mathbf{x}_{j_q} - \mathbf{x}_{j_h}\|_p \leq B$ , we obtain that the points  $\mathbf{x}_{j_q}$  and  $\mathbf{x}_{j_h}$  differ in at most  $B^p$  coordinates by the definition of the  $\ell_p$ -norm. Then because  $Z_{q-1}$  has at most  $(q - 2)B^p$  nonuniform coordinates,  $Z_q$  has at most  $(q - 1)B^p$  nonuniform coordinates as required.

Because the number of nonuniform coordinates for  $S_i$  is the same as the number of nonuniform coordinates for  $Z_r$  and  $r \leq k(2B + 1)$ , we obtain that  $\ell_i \leq kB^p(2B + 1)$ . Then  $\ell = \max_{1 \leq i \leq t} \ell_i \leq kB^p(2B + 1)$ . Because the points of  $\mathbf{Y}$  are in  $\mathbb{Z}^{\ell+1}$ , we have the required upper bound for the dimension. This concludes the proof of (ii).

Finally, to show (iii), we again exploit the property that every  $S_i$  contains at most  $k(2B + 1)$  distinct points. Let  $i \in \{1, \dots, t\}$  and  $h \in \{1, \dots, d\}$  and denote by  $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_r}$  the distinct points in  $X_i$ . Let  $h \in \{1, \dots, d\}$ . We can assume without loss of generality that  $\mathbf{x}_{j_1}[h] \leq \dots \leq \mathbf{x}_{j_r}[h]$ . We claim that  $\mathbf{x}_{j_r}[h] - \mathbf{x}_{j_1}[h] \leq B(k(2B + 1) - 1)$ . This is trivial if  $r = 1$ . Assume that  $r > 1$ . Observe that  $\mathbf{x}_{j_q}[h] - \mathbf{x}_{q-1}[h] \leq B$  for  $q \in \{2, \dots, r\}$ . Otherwise, if there is  $q \in \{2, \dots, r\}$  such that  $\mathbf{x}_{j_q}[h] - \mathbf{x}_{q-1}[h] > B$ , then the distance from any point in  $\{\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_{q-1}}\}$  to any point in  $\{\mathbf{x}_{j_q}, \dots, \mathbf{x}_{j_r}\}$  is more than  $B$  but this contradicts that these points are the distinct points of  $S_i$ . Then because  $\mathbf{x}_{j_q}[h] - \mathbf{x}_{q-1}[h] \leq B$  for  $q \in \{2, \dots, r\}$  and  $r \leq k(2B + 1)$ , we obtain that  $\mathbf{x}_{j_r}[h] - \mathbf{x}_{j_1}[h] \leq B(k(2B + 1) - 1)$ .

Then, by the definition of  $\mathbf{x}'_1, \dots, \mathbf{x}'_n$ , we obtain that for every  $\mathbf{x}'_q, \mathbf{x}'_r \in S'_i$  for some  $i \in \{1, \dots, t\}$  and every  $h \in \{1, \dots, \ell\}$ ,  $|\mathbf{x}'_q[h] - \mathbf{x}'_r[h]| \leq B(k(2B + 1) - 1)$ . By the definition of  $M_h^{(i)}$  for  $i \in \{1, \dots, t\}$ , we obtain that  $|\mathbf{y}_j[h]| \leq B(k(2B + 1) - 1)$  for every  $j \in \{1, \dots, n\}$  and every  $h \in \{1, \dots, \ell\}$ . Because  $|\mathbf{y}_j[\ell + 1]| \leq (k - 1)(B + 1)$ , we have that  $|\mathbf{y}_i[h]| \leq B(k(2B + 1) - 1)$  for  $h \in \{1, \dots, d'\}$  and  $i \in \{1, \dots, n\}$ . This completes the proof of (iii) and the proof of the lemma.  $\square$

Finally in this subsection, we show the following lemma that is used to upper bound the additional cost incurred by the greedy clustering of blocks of equal points.

**Lemma 8** *Let  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  be a collection of  $n$  points of  $\mathbb{Z}^d$  and set  $k$  be a positive integer such that  $n$  is divisible by  $k$ . Suppose that  $S_1, \dots, S_t$  are disjoint collections of equal points of  $\mathbf{X}$  such that  $|S_1| = \dots = |S_t| = \frac{n}{k}$  and  $\mathbf{Y} = \mathbf{X} \setminus (S_1 \cup \dots \cup S_t)$ . Then  $\text{Opt}(\mathbf{Y}, k - t) \leq 2 \cdot \text{Opt}(\mathbf{X}, k)$ .*

**Proof** Let  $\{X_1, \dots, X_k\}$  be an optimum equal  $k$ -clustering of  $\mathbf{X}$  with optimum medians  $\mathbf{c}_1, \dots, \mathbf{c}_k$  of  $X_1, \dots, X_k$ , respectively, that is,  $\text{Opt}(\mathbf{X}, k) = \text{cost}_p(X_1, \dots, X_k) = \text{cost}_p(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k)$ . Let  $\mathbf{x}_{i_h} \in S_h$  for  $h \in \{1, \dots, t\}$ . Consider a  $t$ -tuple of  $(j_1, \dots, j_t)$  of distinct indices from  $\{1, \dots, k\}$  such that

$$\|\mathbf{x}_{i_1} - \mathbf{c}_{j_1}\|_p + \dots + \|\mathbf{x}_{i_t} - \mathbf{c}_{j_t}\|_p = \min_{(q_1, \dots, q_t)} (\|\mathbf{x}_{i_1} - \mathbf{c}_{q_1}\|_p + \dots + \|\mathbf{x}_{i_t} - \mathbf{c}_{q_t}\|_p), \tag{8}$$

where the minimum in the right part is taken over all  $t$ -tuples  $(q_1, \dots, q_t)$  of distinct indices from  $\{1, \dots, k\}$ . Denote  $\ell = k - t$ . Iteratively applying Lemma 5 for  $S_1, \dots, S_t$  and the medians  $\mathbf{c}_{j_1}, \dots, \mathbf{c}_{j_t}$ , we obtain that there is an equal  $\ell$ -clustering  $\{Y_1, \dots, Y_\ell\}$  of  $\mathbf{Y}$  such that

$$\text{cost}_p(S_1, \dots, S_t, Y_1, \dots, Y_\ell) \leq \text{cost}_p(X_1, \dots, X_k) + s \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{j_h}\|_p. \tag{9}$$

Because the points in each  $S_i$  are the same,  $\text{cost}_p(S_i) = 0$  and, therefore,  $\text{cost}_p(S_1, \dots, S_t, Y_1, \dots, Y_\ell) = \text{cost}_p(Y_1, \dots, Y_\ell)$ . Then by (9),

$$\text{Opt}(\mathbf{Y}, \ell) \leq \text{cost}_p(Y_1, \dots, Y_\ell) \leq \text{Opt}(\mathbf{X}, k) + s \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{j_h}\|_p. \tag{10}$$

This implies that to prove the lemma, it is sufficient to show that

$$s \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{j_h}\|_p \leq \text{Opt}(\mathbf{X}, k). \tag{11}$$

To prove (11), we consider the following auxiliary clustering problem. Let  $\mathbf{Z} = S_1 \cup \dots \cup S_t$  and  $s = \frac{n}{k}$ . The task of the problem is to find a partition  $\{Z_1, \dots, Z_k\}$  of  $\mathbf{Z}$ , where some sets may be empty and  $|Z_i| \leq s$  for every  $i \in \{1, \dots, k\}$ , such that

$$\sum_{i=1}^t \sum_{\mathbf{x}_h \in Z_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p \tag{12}$$

is minimum. In words, we cluster the elements of  $\mathbf{Z}$  in the optimum way into clusters of size at most  $s$  using the optimum medians  $\mathbf{c}_1, \dots, \mathbf{c}_k$  for the clustering  $\{X_1, \dots, X_k\}$ . Denote by  $\text{Opt}^*(\mathbf{Z}, k)$  the minimum value of (12). Because in this problem the task is to cluster a subcollection of points of  $\mathbf{X}$  and we relax the cluster size constraints, we have that  $\text{Opt}^*(\mathbf{Z}, k) \leq \text{Opt}(\mathbf{X}, k)$ . We show the following claim.

**Claim 3.2**

$$\text{Opt}^*(\mathbf{Z}, k) \geq s \cdot \min_{(q_1, \dots, q_t)} \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{q_h}\|_p,$$

where the minimum is taken over all  $t$ -tuples  $(q_1, \dots, q_t)$  of distinct indices from  $\{1, \dots, k\}$ .

**Proof of Claim 3.2** We show that the considered auxiliary clustering problem can be reduced to the MIN COST FLOW problem (see, e.g., the textbook of Kleinberg and Tardos [54] for the introduction)<sup>2</sup>. We construct the directed graph  $G$  and define the cost and capacity functions  $c(\cdot)$  and  $\omega(\cdot)$  on the set of arcs  $A(G)$  as follows.

<sup>2</sup> Equivalently one may use the ILP statement.

- Construct two vertices  $a$  and  $b$  that are the *source* and *target* vertices, respectively.
- For every  $i \in \{1, \dots, t\}$ , construct a vertex  $u_i$  (corresponding to  $S_i$ ) and an arc  $(a, u_i)$  with  $\omega(a, u_i) = 0$ .
- For every  $j \in \{1, \dots, k\}$ , construct a vertex  $v_j$  (corresponding to  $Z_j$ ) and an arc  $(v_j, b)$  with  $\omega(v_j, b) = 0$ .
- For every  $h \in \{1, \dots, t\}$  and every  $j \in \{1, \dots, k\}$ , construct an arc  $(u_h, v_j)$  and set  $\omega(u_h, v_j) = \|\mathbf{x}_{i_h} - \mathbf{c}_j\|_p$  (recall that  $\mathbf{x}_{i_h} \in S_h$ ).
- For every arc  $e$  of  $G$ , set  $c(e) = s$ , where  $s = \frac{n}{k}$ .

Then the volume of a flow  $f: A(G) \rightarrow \mathbb{R}_{\geq 0}$  is  $v(f) = \sum_{i=1}^t f(a, u_i)$  and its cost is  $\omega(f) = \sum_{a \in A(G)} \omega(a) \cdot f(a)$ . Let  $f^*(\cdot)$  be a flow of volume  $st$  with minimum cost. We claim that  $\omega(f^*) = \text{Opt}^*(\mathbf{Z}, k)$ .

Assume that  $\{Z_1, \dots, Z_k\}$  is a partition of  $\mathbf{Z}$  such that  $|Z_i| \leq s$  for every  $i \in \{1, \dots, k\}$  and  $\text{Opt}^*(\mathbf{Z}, k) = \sum_{i=1}^t \sum_{\mathbf{x}_h \in Z_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p$ . We define the flow  $f(\cdot)$  as follows:

- for every  $i \in \{1, \dots, t\}$ , set  $f(a, u_i) = s$ ,
- for every  $i \in \{1, \dots, t\}$  and  $j \in \{1, \dots, k\}$ , set  $f(u_i, v_j) = |S_i \cap Z_j|$ , and
- for every  $j \in \{1, \dots, k\}$ , set  $f(v_j, b) = |Z_j|$ .

It is easy to verify that  $f$  is a feasible flow of volume  $st$  and  $\omega(f) = \sum_{i=1}^t \sum_{\mathbf{x}_h \in Z_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p$ . Thus,  $\omega(f^*) \leq \omega(f) = \text{Opt}^*(\mathbf{Z}, k)$ .

For the opposite inequality, consider  $f^*(\cdot)$ . By the well-known property of flows (see [54]), we can assume that  $f^*(\cdot)$  is an integer flow, that is,  $f^*(e)$  is a nonnegative integer for every  $e \in A(G)$ . Since  $v(f^*) = st$ , we have that  $f^*(a, u_i) = s$  for every  $i \in \{1, \dots, t\}$ . Then we construct the clustering  $\{Z_1, \dots, Z_k\}$  as follows: for every  $i \in \{1, \dots, t\}$  and  $j \in \{1, \dots, k\}$ , we put exactly  $f^*(u_i, v_j)$  points of  $S_i$  into  $Z_j$ . Because  $f^*(a, u_i) = s$  for every  $i \in \{1, \dots, t\}$  and  $c(v_j, b) = s$  for every  $j \in \{1, \dots, k\}$ , we obtain that  $\{Z_1, \dots, Z_k\}$  is a partition of  $\mathbf{Z}$  such that  $|Z_i| \leq s$  for every  $i \in \{1, \dots, k\}$  and  $\sum_{i=1}^t \sum_{\mathbf{x}_h \in Z_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p = \omega(f^*)$ . This implies that  $\text{Opt}^*(\mathbf{Z}, k) \leq \sum_{i=1}^t \sum_{\mathbf{x}_h \in Z_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p = \omega(f^*)$ .

This proves that  $\omega(f^*) = \text{Opt}^*(\mathbf{Z}, k)$ . Moreover, we can observe that, given an integer flow  $f(\cdot)$  with  $v(f) = st$ , we can construct a feasible clustering  $\{Z_1, \dots, Z_k\}$  of cost  $\omega(f)$  such that for every  $i \in \{1, \dots, t\}$  and every  $j \in \{1, \dots, k\}$ ,  $|S_i \cap Z_j| = f(u_i, v_j)$ . Recall that the capacities of the arcs of  $G$  are the same and are equal to  $s$ . Then again exploiting the properties of flows (see [54]), we observe that there is a flow  $f^*(\cdot)$  with  $v(f^*) = st$  of minimum cost such that *saturated* arcs (that is, arcs  $e$  with  $f^*(e) = c(e) = s$ ) compose internally vertex disjoint  $(a, b)$ -paths, and the flow on other arcs is zero. This implies, that for the clustering  $\{Z_1, \dots, Z_k\}$  constructed for  $f^*(\cdot)$ , for every  $j \in \{1, \dots, k\}$ , either  $Z_j = \emptyset$  or there is  $i \in \{1, \dots, t\}$  such that  $Z_j = S_i$ . Assume that  $j_1, \dots, j_t$  are distinct indices from  $\{1, \dots, k\}$  such that  $Z_{j_h} = S_h$  for  $h \in \{1, \dots, t\}$ . Then  $\omega(f^*) = \sum_{i=1}^t \sum_{\mathbf{x}_h \in Z_i} \|\mathbf{x}_h - \mathbf{c}_i\|_p = s \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{j_h}\|_p$  and

$$\text{Opt}^*(\mathbf{Z}, k) = \omega(f^*) = s \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{j_h}\|_p \geq s \cdot \min_{(q_1, \dots, q_t)} \sum_{h=1}^t \|\mathbf{x}_{i_h} - \mathbf{c}_{q_h}\|_p,$$

where the minimum is taken over all  $t$ -tuples  $(q_1, \dots, q_t)$  of distinct indices from  $\{1, \dots, k\}$ . This proves the claim.  $\square$

Recall that  $\text{Opt}^*(\mathbf{Z}, k) \leq \text{Opt}(\mathbf{X}, k)$ . By the choice of  $j_1, \dots, j_t$  in (8) and Claim 3.2, we obtain that inequality (11) holds. Then by (11), we have that  $\text{Opt}(\mathbf{Y}, k - t) \leq 2 \cdot \text{Opt}(\mathbf{X}, k)$  as required by the lemma.  $\square$

### 3.2 Proof of Theorem 1

Now we are ready to show the result about the approximate kernel that we restate.

**Theorem 1** *For every nonnegative integer constant  $p$ , PARAMETERIZED EQUAL CLUSTERING admits a 2-approximate kernel when parameterized by  $B$ , where the output collection of points has  $\mathcal{O}(B^2)$  points of  $\mathbb{Z}^{d'}$  with  $d' = \mathcal{O}(B^{p+2})$ , where each coordinate of a point takes an absolute value of  $\mathcal{O}(B^3)$ .*

**Proof** Let  $(\mathbf{X}, k, B)$  be an instance of PARAMETERIZED EQUAL CLUSTERING with  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , where the points are from  $\mathbb{Z}^d$  and  $n$  is divisible by  $k$ . Recall that a lossy kernel consists of two algorithms. The first algorithm is a polynomial time reduction producing an instance  $(\mathbf{X}', k', B')$  of bounded size. The second algorithm is a solution-lifting and for every equal  $k'$ -clustering  $\{X'_1, \dots, X'_{k'}\}$  of  $\mathbf{X}'$ , this algorithm produces in a polynomial time an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  of  $\mathbf{X}$  such that

$$\frac{\text{cost}_p^B(X_1, \dots, X_k)}{\text{Opt}(\mathbf{X}, k, B)} \leq 2 \cdot \frac{\text{cost}_p^{B'}(X'_1, \dots, X'_{k'})}{\text{Opt}(\mathbf{X}', k', B')}. \tag{13}$$

We separately consider the cases when  $\frac{n}{k} \geq 4B + 1$  and  $\frac{n}{k} \leq 4B$ .

Suppose that  $\frac{n}{k} \geq 4B + 1$ . Then we apply the algorithm from Lemma 6. If the algorithm returns the answer that  $\mathbf{X}$  does not admit an equal  $k$ -clustering of cost at most  $B$ , then the reduction algorithm returns a trivial no-instance  $(\mathbf{X}', k', B')$  of constant size, that is, an instance such that  $\mathbf{X}'$  has no clustering of cost at most  $B'$ . For example, we set  $\mathbf{X}' = \{(0), (1)\}$ ,  $k' = 1$ , and  $B' = 0$ . Here and in the further cases when the reduction algorithm returns a trivial no-instance, the solution-lifting algorithm returns an arbitrary equal  $k$ -clustering of  $\mathbf{X}$ . Since  $\text{cost}_p^B(X_1, \dots, X_k) = \text{Opt}(\mathbf{X}, k, B) = B + 1$ , (13)<sup>3</sup> holds. Assume that the algorithm from Lemma 6 produced an equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  of minimum cost. Then the reduction returns an arbitrary instance of PARAMETERIZED EQUAL CLUSTERING of constant size. For example, we can use  $\mathbf{X}' = \{(0)\}$ ,  $k' = 1$ , and  $B' = 0$ . The solution-lifting algorithms always returns  $\{X_1, \dots, X_k\}$ . Clearly,  $\text{cost}_p^B(X_1, \dots, X_k) = \text{Opt}(\mathbf{X}, k, B)$  and (13) is fulfilled.

From now on, we assume that  $\frac{n}{k} \leq 4B$ , that is,  $n \leq 4Bk$ . We apply the algorithm from Lemma 6. If this algorithm reports that there is no equal  $k$ -clustering of cost at

<sup>3</sup> Note that by our simplifying assumption,  $\frac{\text{cost}_p^B(X_1, \dots, X_k)}{\text{Opt}(\mathbf{X}, k, B)} = 1$  if  $\text{Opt}(\mathbf{X}, k, B) = \text{cost}_p^B(X_1, \dots, X_k) = 0$  and  $\frac{\text{cost}_p^{B'}(X'_1, \dots, X'_{k'})}{\text{Opt}(\mathbf{X}', k', B')} = +\infty$  if  $\text{Opt}(\mathbf{X}, k, B) = 0$  and  $\text{cost}_p^B(X_1, \dots, X_k) > 0$ , and the same assumption is used for  $\frac{\text{cost}_p^{B'}(X'_1, \dots, X'_{k'})}{\text{Opt}(\mathbf{X}', k', B')}$ .

most  $B$ , then the reduction algorithm returns a trivial no-instance and the solution-lifting algorithm outputs an arbitrary equal  $k$ -clustering of  $\mathbf{X}$ . Clearly, (13) is satisfied. Assume that this is not the case. Then we obtain a collection of  $n \leq 4Bk$  points  $\mathbf{Y} = \{y_1, \dots, y_n\}$  of  $\mathbb{Z}^{d'}$  satisfying conditions (i)–(iii) of Lemma 7. That is,

- (i) for every partition  $\{I_1, \dots, I_k\}$  of  $\{1, \dots, n\}$  such that  $|I_1| = \dots = |I_k| = \frac{n}{k}$ , either  $\text{cost}_p(X_1, \dots, X_k) > B$  and  $\text{cost}_p(Y_1, \dots, Y_k) > B$  or  $\text{cost}_p(X_1, \dots, X_k) = \text{cost}_p(Y_1, \dots, Y_k)$ , where  $X_i = \{\mathbf{x}_h \mid h \in I_i\}$  and  $Y_i = \{y_h \mid h \in I_i\}$  for every  $i \in \{1, \dots, k\}$ ,
- (ii)  $d' = \mathcal{O}(kB^{p+1})$ , and
- (iii)  $|y_i[h]| = \mathcal{O}(kB^2)$  for  $h \in \{1, \dots, d'\}$  and  $i \in \{1, \dots, n\}$ .

By (i), for given an equal  $k$ -clustering  $\{Y_1, \dots, Y_k\}$  of  $\mathbf{Y}$ , we can compute the corresponding clustering  $\{X_1, \dots, X_k\}$  by setting  $X_i = \{\mathbf{x}_h \mid y_h \in Y_i\}$  for  $i \in \{1, \dots, k\}$ . Then  $\text{Opt}(\mathbf{X}, k, B) = \text{Opt}(\mathbf{Y}, k, B)$  and

$$\frac{\text{cost}_p^B(X_1, \dots, X_k)}{\text{Opt}(\mathbf{X}, k, B)} = \frac{\text{cost}_p^B(Y_1, \dots, Y_k)}{\text{Opt}(\mathbf{Y}, k, B)}. \tag{14}$$

Hence the instances  $(\mathbf{X}, k, B)$  and  $(\mathbf{Y}, k, B)$  are equivalent. We continue with the compressed instance  $(\mathbf{Y}, k, B)$ .

Now we apply the greedy procedure that constructs clusters  $S_1, \dots, S_t$  composed by equal points. Formally, we initially set  $\mathbf{X}' := Y, k' := k$ , and  $i := 0$ . Then we do the following:

- while  $\mathbf{X}'$  contains a collections  $S$  of  $s$  identical points, set  $i := i + 1, S_i := S, \mathbf{X}' := \mathbf{X}' \setminus S$ , and  $k' := k' - 1$ .

Denote by  $\mathbf{X}'$  the set of points obtained by the application of the procedure and let  $S_1, \dots, S_t$  be the collections of equal points constructed by the procedure. Note that  $k' = k - t$ . We also define  $B' = 2B$ . Notice that it may happen that  $\mathbf{X}' = \mathbf{Y}$  or  $\mathbf{X}' = \emptyset$ . The crucial property exploited by the kernelization is that by Lemma 8,  $\text{Opt}(\mathbf{X}', k') \leq 2 \cdot \text{Opt}(\mathbf{Y}, k)$ .

We argue that if  $k' > B$ , then we have no  $k$ -clustering of cost at most  $B$ . Suppose that  $k' > B'$ . Consider an arbitrary equal  $k'$ -clustering  $\{X'_1, \dots, X'_{k'}\}$  of  $\mathbf{X}'$ . Because the construction of  $S_1, \dots, S_t$  stops when there is no collection of  $s$  equal points, each cluster  $X'_i$  contains at least two distinct points. Since all points have integer coordinates, we have that  $\text{cost}_p(X'_i) \geq 1$  for every  $i \in \{1, \dots, k'\}$ . Therefore,  $\text{cost}_p(X'_1, \dots, X'_{k'}) = \sum_{i=1}^{k'} \text{cost}_p(X'_i) \geq k' > B' = 2B$ . This means that  $2 \cdot \text{Opt}(\mathbf{Y}, k) \geq \text{Opt}(\mathbf{X}', k') > 2B$  and  $\text{Opt}(\mathbf{Y}, k) > B$ . Using this, our reduction algorithm returns a trivial no-instance. Then the solution-lifting algorithm outputs an arbitrary equal  $k$ -clustering of  $\mathbf{X}$  and this satisfies (13).

From now on we assume that  $k' \leq B' = 2B$  and construct the reduction and solution lifting algorithms for this case.

If  $k' = 0$ , then  $\mathbf{X}' = \emptyset$  and the reduction algorithm simply returns an arbitrary instance of constant size. Otherwise, our reduction algorithms returns  $(\mathbf{X}', k', B')$ .



Observe that since  $k' \leq B' = 2B$ ,  $|\mathbf{X}'| \leq n \leq 4B^2$ . Recall that  $d' = \mathcal{O}(B^{p+2})$  and  $|\mathbf{x}'_i[h]| = \mathcal{O}(B^3)$  for  $h \in \{1, \dots, d'\}$  for every point  $\mathbf{x}'_i \in \mathbf{X}'$ . We conclude that the instance  $(\mathbf{X}', k', B')$  of PARAMETERIZED EQUAL CLUSTERING satisfies the size conditions of the theorem.

Now we describe the solution-lifting algorithm and argue that inequality (13) holds.

If  $k' = 0$ , then the solution-lifting algorithm ignores the output of the reduction algorithm which was arbitrary. It takes the equal  $k$ -clustering  $\{S_1, \dots, S_k\}$  of  $\mathbf{Y}$  and outputs the equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  of  $\mathbf{X}$  by setting  $X_i = \{\mathbf{x}_h \mid \mathbf{y}_h \in S_i\}$  for  $i \in \{1, \dots, k\}$ . Clearly,  $\text{cost}_p(S_1, \dots, S_k) = \text{cost}_p(X_1, \dots, X_k) = 0$ . Therefore, (13) holds.

If  $k' > 0$ , we consider an equal  $k'$ -clustering  $\{X'_1, \dots, X'_{k'}\}$  of  $\mathbf{X}'$ . The solution-lifting algorithm constructs an equal  $k$ -clustering  $\{S_1, \dots, S_t, X'_1, \dots, X'_{k'}\}$ , that is, we just add the clusters constructed by our greedy procedure. Since the points in each set  $S_i$  are the same,  $\text{cost}_p(S_i) = 0$  for every  $i \in \{1, \dots, t\}$ . Therefore,

$$\text{cost}_p(S_1, \dots, S_t, X'_1, \dots, X'_{k'}) = \text{cost}_p(X'_1, \dots, X'_{k'}).$$

Notice that since  $\text{Opt}(\mathbf{X}', k') \leq 2 \cdot \text{Opt}(\mathbf{Y}, k)$ , we have that  $\text{Opt}(\mathbf{X}', k', B') \leq 2 \cdot \text{Opt}(\mathbf{Y}, k, B)$ . Indeed, if  $\text{Opt}(\mathbf{Y}, k) \leq B$ , then  $\text{Opt}(\mathbf{X}', k') \leq 2B = B'$ . Hence,  $\text{Opt}(\mathbf{Y}, k, B) = \text{Opt}(\mathbf{Y}, k)$ ,  $\text{Opt}(\mathbf{X}', k', B') = \text{Opt}(\mathbf{X}', k')$ , and  $\text{Opt}(\mathbf{X}', k', B') \leq 2 \cdot \text{Opt}(\mathbf{Y}, k, B)$ . If  $\text{Opt}(\mathbf{Y}, k) > B$ , then  $\text{Opt}(\mathbf{Y}, k, B) = B + 1$ . In this case,  $2 \cdot \text{Opt}(\mathbf{Y}, k, B) = 2B + 2 > \text{Opt}(\mathbf{X}', k', B')$  because  $\text{Opt}(\mathbf{X}', k', B') \leq B' + 1 = 2B + 1$ . Finally, since  $\text{cost}_p(S_1, \dots, S_t, X'_1, \dots, X'_{k'}) = \text{cost}_p(X'_1, \dots, X'_{k'})$  and  $\text{Opt}(\mathbf{X}', k', B') \leq 2 \cdot \text{Opt}(\mathbf{Y}, k, B)$ , we conclude that

$$\frac{\text{cost}_p^B(S_1, \dots, S_t, X'_1, \dots, X'_{k'})}{\text{Opt}(\mathbf{Y}, k, B)} \leq 2 \cdot \frac{\text{cost}_p^B(X'_1, \dots, X'_{k'})}{\text{Opt}(\mathbf{X}', k', B')}. \tag{15}$$

Then the solution-lifting algorithm computes the equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  for the equal  $k$ -clustering  $\{Y_1, \dots, Y_k\} = \{S_1, \dots, S_t, X'_1, \dots, X'_{k'}\}$  of  $\mathbf{Y}$  by setting  $X_i = \{\mathbf{x}_h \mid \mathbf{y}_h \in Y_i\}$  for  $i \in \{1, \dots, k\}$ . Combining (14) and (15), we obtain (13).

This concludes the description of the reduction and solution-lifting algorithms, as well as the proof of their correctness. To argue that the reduction algorithm is a polynomial-time algorithm, we observe that the algorithms from Lemmata 6 and 7 run in a polynomial time. Trivially, the greedy construction of  $S_1, \dots, S_t, \mathbf{X}$ , and  $k'$  can be done in a polynomial time. Therefore, the reduction algorithm runs in a polynomial time. The solution-lifting algorithm is also easily implementable to run in a polynomial time. This concludes the proof.  $\square$

### 4 Kernelization

In this section, we study (exact) kernelization of clustering with equal sizes. In Subsection 4.1 we prove Theorem 2 claiming that decision version of the problem, DECISION EQUAL CLUSTERING, does not admit a polynomial kernel being parameterized by  $B$

only. We also show in Subsection 4.2 that the technical lemmata developed in the previous section for approximate kernel, can be used to prove that DECISION EQUAL CLUSTERING parameterized by  $k$  and  $B$  admits a polynomial kernel.

#### 4.1 Kernelization Lower Bound

In this subsection, we show that it is unlikely that DECISION EQUAL CLUSTERING admits a polynomial kernel when parameterized by  $B$  only. We prove this for the  $\ell_0$  and  $\ell_1$ -norms. Our lower bound holds even for points with binary coordinates, that is, for points from  $\{0, 1\}^d$ . For this, we use the result of Dell and Marx [55] about kernelization lower bounds for the PERFECT  $r$ -SET MATCHING problem.

A hypergraph  $\mathcal{H}$  is said to be  $r$ -uniform for a positive integer  $r$ , if every hyperedge of  $\mathcal{H}$  has size  $r$ . Similarly to graphs, a set of hyperedges  $M$  is a *matching* if the hyperedges in  $M$  are pairwise disjoint, and  $M$  is *perfect* if every vertex of  $\mathcal{H}$  is *saturated* in  $M$ , that is, included in one of the hyperedges of  $M$ . PERFECT  $r$ -SET MATCHING asks, given a  $r$ -uniform hypergraph  $\mathcal{H}$ , whether  $\mathcal{H}$  has a perfect matching. Dell and Marx [55] proved the following kernelization lower bound.

**Proposition 1** ([55]) *Let  $r \geq 3$  be an integer and let  $\varepsilon$  be a positive real. If  $\text{NP} \subseteq \text{coNP} / \text{poly}$ , then PERFECT  $r$ -SET MATCHING does not have kernels with  $\mathcal{O}\left(\left(\frac{|V(\mathcal{H})|}{r}\right)^{r-\varepsilon}\right)$  hyperedges.*

We need a weaker claim.

**Corollary 1** PERFECT  $r$ -SET MATCHING *admits no polynomial kernel when parameterized by the number of vertices of the input hypergraph unless  $\text{NP} \subseteq \text{coNP} / \text{poly}$ .*

**Proof** To see the claim, it is sufficient to observe that the existence of a polynomial kernel for PERFECT  $r$ -SET MATCHING parameterized by  $|V(\mathcal{H})|$  implies that the problem has a kernel such that the number of hyperedges is polynomial in  $|V(\mathcal{H})|$  with the degree of the polynomial that does not depend on  $d$  contradicting Proposition 1.  $\square$

We show the kernelization lower bound for  $\ell_0$  and  $\ell_1$  using the fact that optimum medians can be computed by the *majority rule* for a collection of binary points. Let  $X$  be a collection of points of  $\{0, 1\}^d$ . We construct  $\mathbf{c} \in \{0, 1\}^d$  as follows: for  $i \in \{1, \dots, d\}$ , consider the multiset  $S_i = \{\mathbf{x}[i] \mid \mathbf{x} \in X\}$  and set  $\mathbf{c}[i] = 0$  if at least half of the elements of  $S_i$  are zeros, and set  $\mathbf{c}[i] = 1$  otherwise. It is straightforward to observe the following.

**Observation 1** Let  $X$  be a collection of points of  $\{0, 1\}^d$  and let  $\mathbf{c} \in \{0, 1\}^d$  be a vector constructed by the majority rule. Then for the  $\ell_0$  and  $\ell_1$ -norms,  $\mathbf{c}$  is

We also use the following lemma which is a special case of Lemma 9 of [50].

**Lemma 9** ([50]) *Let  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering of a collection of points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  from  $\{0, 1\}^d$ , and let  $\mathbf{c}_1, \dots, \mathbf{c}_k$  be optimum medians for  $X_1, \dots, X_k$ , respectively. Let also  $\mathbf{C} \subseteq \{\mathbf{c}_1, \dots, \mathbf{c}_k\}$  be the set of medians coinciding with some points of  $\mathbf{X}$ . Suppose that every collection of the same points of*

$\mathbf{X}$  has size at most  $\frac{n}{k}$ . Then there is an equal  $k$ -clustering  $\{X'_1, \dots, X'_k\}$  of  $\mathbf{X}$  such that  $\text{cost}_0(X'_1, \dots, X'_k, \mathbf{c}_1, \dots, \mathbf{c}_k) \leq \text{cost}_0(X_1, \dots, X_k, \mathbf{c}_1, \dots, \mathbf{c}_k)$  and for every  $i \in \{1, \dots, k\}$ , the following is fulfilled: if  $\mathbf{c}_i \in \mathbf{C}$ , then each  $\mathbf{x}_h \in \mathbf{X}$  coinciding with  $\mathbf{c}_i$  is in  $X'_i$ .

Now we are ready to prove Theorem 2, we restate it here.

**Theorem 2** For the  $\ell_0$  and  $\ell_1$ -norms, DECISION EQUAL CLUSTERING has no polynomial kernel when parameterized by  $B$  unless  $\text{NP} \subseteq \text{coNP} / \text{poly}$ , even if the input points are binary, that is, are from  $\{0, 1\}^d$ .

**Proof** Notice that for any binary vector  $\mathbf{x} \in \{0, 1\}^d$ ,  $\|\mathbf{x}\|_0 = \|\mathbf{x}\|_1$ . Since we consider only instances where the input points are binary, we can assume that the medians of clusters are binary as well by Observation 1. Then it is sufficient to prove the theorem for one norm, say  $\ell_0$ . We reduce from PERFECT  $r$ -SET MATCHING. Let  $\mathcal{H}$  be an  $r$ -uniform hypergraph. Denote by  $v_1, \dots, v_n$  the vertices and by  $E_1, \dots, E_m$  the hyperedges of  $\mathcal{H}$ , respectively. We assume that  $n$  is divisible by  $r$ , as otherwise  $\mathcal{H}$  has no perfect matching. We also assume that  $r \geq 3$  because for  $r \leq 2$ , PERFECT  $r$ -SET MATCHING can be solved in a polynomial time [52].

We construct the instance  $(\mathbf{X}, k, B)$  of DECISION EQUAL CLUSTERING, where  $\mathbf{X}$  is a collection of  $(r - 1)n + rm$  points of  $\{0, 1\}^d$ , where  $d = 2rn$ .

To describe the construction of  $\mathbf{X}$ , we partition the set  $\{1, \dots, 2rn\}$  of coordinate indices into  $n$  blocks  $R_1, \dots, R_n$  of size  $2r$  each. For every  $i \in \{1, \dots, n\}$ , we select an index  $p_i \in R_i$  and set  $R'_i = R_i \setminus \{p_i\}$ . Formally,

- $R_i = \{2r(i - 1) + 1, \dots, 2ri\}$  for  $i \in \{1, \dots, n\}$ ,
- $p_i = 2r(i - 1) + 1$  for  $i \in \{1, \dots, n\}$ , and
- $R'_i = \{2r(i - 1) + 2, \dots, 2ri\}$  for  $i \in \{1, \dots, n\}$ .

The set of points  $\mathbf{X}$  consists of  $n + m$  blocks of equal points  $V_1, \dots, V_n$  and  $F_1, \dots, F_m$ , where  $|V_i| = r - 1$  for each  $i \in \{1, \dots, n\}$  and  $|F_i| = r$  for  $i \in \{1, \dots, m\}$ . Each block  $V_i$  is used to encode the vertex  $v_i$ , and each block  $F_i$  is used to encode the corresponding hyperedge  $E_i$ . An example is shown in Fig. 1.

For each  $i \in \{1, \dots, n\}$ , we define the vector  $\mathbf{v}_i \in \{0, 1\}^{2rn}$  corresponding to the vertex  $v_i$  of  $\mathcal{H}$ :

$$\mathbf{v}_i[j] = \begin{cases} 1 & \text{if } j \in R_i, \\ 0 & \text{otherwise.} \end{cases}$$

Then  $V_i$  consists of  $r - 1$  copies of  $\mathbf{v}_i$  that we denote  $\mathbf{v}_i^{(1)}, \dots, \mathbf{v}_i^{(r-1)}$ .

For every  $j \in \{1, \dots, m\}$ , we define the vector  $\mathbf{f}_j \in \{0, 1\}^{2rn}$  corresponding to the hyperedge  $E_j = \{v_{i_1^{(j)}}, \dots, v_{i_r^{(j)}}\}$ :

$$\mathbf{f}_j[h] = \begin{cases} 1 & \text{if } h = p_s \text{ for some } s \in \{i_1^{(j)}, \dots, i_r^{(j)}\}, \\ 0 & \text{otherwise.} \end{cases}$$

Then  $F_j$  includes  $r$  copies of  $\mathbf{f}_j$  denoted by  $\mathbf{f}_j^{(1)}, \dots, \mathbf{f}_j^{(r)}$ .



To complete the construction of the instance of DECISION EQUAL CLUSTERING, we define

- $k = n + m - \frac{n}{r}$ ,
- $B = (3r - 2)n$ .

Recall that  $n$  is divisible by  $r$  and note that  $\frac{(r-1)n+rm}{k} = r$ .

It is straightforward to verify that the construction of  $(\mathbf{X}, k, B)$  is polynomial. We claim that the hypergraph  $\mathcal{H}$  has a perfect matching if and only if  $(\mathbf{X}, k, B)$  is a yes-instance of DECISION EQUAL CLUSTERING. The proof uses the following property of the points of  $\mathbf{X}$ : for every  $i \in \{1, \dots, n\}$  and every  $j \in \{1, \dots, m\}$ ,

$$\|v_i - f_j\|_0 = \begin{cases} 3r - 2 & \text{if } v_i \in E_j, \\ 3r & \text{if } v_i \notin E_j. \end{cases} \tag{16}$$

For the forward direction, assume that  $\mathcal{H}$  has a perfect matching  $M$ . Assume without loss of generality that  $M = \{E_1, \dots, E_s\}$  for  $s = \frac{n}{r}$ . Since  $M$  is a perfect matching, for every  $i \in \{1, \dots, n\}$ , there is a unique  $h_i \in \{1, \dots, s\}$  such that  $v_i \in E_{h_i}$ . We construct the equal  $k$ -clustering  $\{X_1, \dots, X_k\}$  as follows.

For every  $i \in \{1, \dots, n\}$ , we define  $X_i = V_i \cup \{f_{h_i}^{(t)}\}$ , where  $t$  is chosen from the set  $\{1, \dots, r\}$  in such a way that  $X_1, \dots, X_n$  are disjoint. In words, we initiate each cluster  $X_i$  by setting  $X_i := V_i$  for  $i \in \{1, \dots, n\}$ . This way, we obtain  $n$  clusters of size  $r - 1$  each. Then we consider the blocks of points  $F_1, \dots, F_s$  corresponding to the hyperedges of  $M$  and split them between the clusters  $X_1, \dots, X_n$  by including a single element into each cluster. It is crucial that each  $X_i = V_i$  is complemented by an element of  $F_{h_i}$ , that is, by an element of the initial cluster corresponding to the hyperedge saturating the vertex  $v_i$ . Since  $M$  is a perfect matching, this splitting is feasible.

Notice that the first  $s$  blocks of points  $F_1, \dots, F_s$  are split between  $X_1, \dots, X_n$ . The remaining  $m - s$  blocks  $F_{s+1}, \dots, F_m$  have size  $r$  each and form clusters  $X_{n+1}, \dots, X_k$ . This completes the construction of  $\{X_1, \dots, X_k\}$ .

To evaluate  $\text{cost}_0(X_1, \dots, X_k)$ , notice that the optimal median  $c_i = v_i$  for  $i \in \{1, \dots, n\}$  by the majority rule. Then, by (16),  $\text{cost}_0(X_i) = \|v_i - f_{h_i}\|_0 = 3r - 2$ . Since the clusters  $X_{n+1}, \dots, X_r$  consist of equal points, we have that  $\text{cost}_0(X_i) = 0$  for  $i \in \{1, \dots, m-s\}$ . Then  $\text{cost}(X_1, \dots, X_k) = (3r - 2)n = B$ . Therefore,  $(\mathbf{X}, k, B)$  is a yes-instance of DECISION EQUAL CLUSTERING.

For the opposite direction, let  $\{X_1, \dots, X_k\}$  be an equal  $k$ -clustering of  $\mathbf{X}$  of cost at most  $B$ . Denote by  $c_1, \dots, c_r$  the optimal medians constructed by the majority rule. Observe that the choice of a median by the majority rule described above is not symmetric because if  $i$ -th coordinates of the points in a cluster have the same number of zeros and ones, the rule selects the zero value for the  $i$ -coordinate of the median. We show the following claim.

**Claim 4.1** For every  $i \in \{1, \dots, k\}$ , either  $c_i \in \{v_1, \dots, v_n\}$  or  $c_i[j] = 0$  for all  $j \in R'_1 \cup \dots \cup R'_n$ . Moreover, the medians of the first type, that is, coinciding with one of  $v_1, \dots, v_n$ , are distinct.

**Proof of Claim 4.1** Suppose that  $\mathbf{c}_i[h] \neq 0$  for some  $h \in R'_j$ , where  $j \in \{1, \dots, n\}$ . Observe that, by the construction of  $\mathbf{X}$ , for every point  $\mathbf{x} \in \mathbf{X}$ ,  $\mathbf{x}[h] = 1$  only if  $\mathbf{x} \in V_j$ . Since  $\mathbf{c}_i$  is constructed by the majority rule, we obtain that more than half of elements of  $X_i$  are from  $V_j$  and  $\mathbf{c}_i = \mathbf{v}_j$ . To see the second part of the claim, notice that  $|V_j| = r - 1$  and, therefore, at most one cluster  $X_i$  of size  $r$  can have at least half of its elements from  $V_j$ .  $\square$

By Claim 4.1, we assume without loss of generality that  $\mathbf{c}_i = \mathbf{v}_i$  for  $i \in \{1, \dots, \ell\}$  for some  $\ell \in \{0, \dots, r\}$  ( $\ell = 0$  if there is no cluster with the median from  $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ ) and  $\mathbf{c}_i[j] = 0$  for  $j \in R'_1 \cup \dots \cup R'_n$  whenever  $i \in \{\ell + 1, \dots, k\}$ . Because the medians  $\mathbf{c}_1, \dots, \mathbf{c}_\ell$  are equal to points of  $\mathbf{X}$ , by Lemma 9, we can assume that  $V_i \subset X_i$  for  $i \in \{1, \dots, \ell\}$ .

**Claim 4.2**  $\ell = n$ .

**Proof of Claim 4.2** The proof is by contradiction. Assume that  $\ell < n$ . Consider the elements of  $n - \ell$  blocks  $V_{\ell+1}, \dots, V_n$ . Let  $p$  be the number of elements of  $V_{\ell+1} \cup \dots \cup V_n$  included in  $X_1, \dots, X_\ell$  and the remaining  $q = (r - 1)(n - \ell) - p$  elements are in  $X_{\ell+1}, \dots, X_k$ . By the definition of  $\mathbf{v}_1, \dots, \mathbf{v}_n$ , if a point  $\mathbf{v}_h^{(t)} \in V_h$  for some  $h \in \{\ell + 1, \dots, n\}$  is in  $X_i$  for some  $i \in \{1, \dots, \ell\}$ , then  $\|\mathbf{v}_h^{(t)} - \mathbf{c}_i\|_0 = \|\mathbf{v}_h - \mathbf{v}_i\|_0 = 4r$ . Also we have that if  $\mathbf{v}_h^{(t)} \in V_h$  for some  $h \in \{\ell + 1, \dots, n\}$  is in  $X_i$  for some  $i \in \{\ell + 1, \dots, r\}$ , then  $\|\mathbf{v}_h^{(t)} - \mathbf{c}_i\|_0 = \|\mathbf{v}_h - \mathbf{c}_i\|_0 \geq |R'_h| = 2r - 1$ . By (16), if the unique point  $X_i \setminus V_i$  is  $\mathbf{f}_h^{(t)} \in F_h$  for some  $h \in \{1, \dots, m\}$ , then  $\|\mathbf{f}_h^{(t)} - \mathbf{c}_i\|_0 = \|\mathbf{f}_h - \mathbf{v}_i\|_0 \geq 3r - 2$ . Then  $\sum_{i=1}^\ell \text{cost}_0(X_i) \geq 4rp + (3r - 2)(\ell - p)$  and  $\sum_{i=\ell+1}^k \text{cost}_0(X_i) \geq (2r - 1)q$ . Recall also that  $r \geq 3$  and, therefore,  $r + 2 \leq 2r - 1$  and  $(r + 2)(r - 1) > 3r - 2$ . Summarizing, we obtain that

$$\begin{aligned} \text{cost}_0(X_1, \dots, X_k) &= \sum_{i=1}^\ell \text{cost}_0(X_i) + \sum_{i=\ell+1}^k \text{cost}_0(X_i) \\ &\geq (4rp + (3r - 2)(\ell - p)) + ((2r - 1)q) \\ &= (3r - 2)\ell + (r + 2)p + (2r - 1)q \\ &\geq (3r - 2)\ell + (r + 2)(p + q) = (3r - 2)\ell \\ &\quad + (r + 2)(r - 1)(n - \ell) \\ &> (3r - 2)n = B, \end{aligned}$$

but this contradicts that  $\text{cost}_0(X_1, \dots, X_k) \leq B$ . This proves the claim.  $\square$

By Claim 4.2, we obtain that  $\mathbf{c}_i = \mathbf{v}_i$  and  $X_i \subset I_i$  for  $i \in \{1, \dots, n\}$ . For every  $i \in \{1, \dots, n\}$ ,  $X_i \setminus V_i$  contains a unique point. Clearly, this is a point from  $F_1 \cup \dots \cup F_m$ . Denote by  $\mathbf{f}_{h_i}^{(t)}$  the point of  $X_i \subset I_i$  for  $i \in \{1, \dots, n\}$ . By (16),  $\|\mathbf{c}_i - \mathbf{f}_{h_i}^{(t)}\|_0 =$

$\|\mathbf{c}_i - \mathbf{f}_{h_i}\|_0 \geq 3r - 2$  for every  $i \in \{1, \dots, n\}$ . This means that

$$\begin{aligned} B &\geq \text{cost}_0(X_1, \dots, X_k) = \sum_{i=1}^n \text{cost}_0(X_i) + \sum_{i=n+1}^k \text{cost}_0(X_i) \geq \sum_{i=1}^n \text{cost}_0(X_i) \\ &\geq (3d - 2)n = B. \end{aligned}$$

Therefore,  $\sum_{i=n+1}^k \text{cost}_0(X_i) = 0$ . Hence,  $k - n = m - s$  clusters  $X_{n+1}, \dots, X_k \subseteq F_1 \cup \dots \cup F_m$ , where  $s = \frac{n}{r}$ , consists of equal points. Without loss of generality, we assume that  $F_{s+1}, \dots, F_m$  form these clusters. Then the elements of  $F_1, \dots, F_s$  are split to complement  $V_1, \dots, V_n$  to form  $X_1, \dots, X_n$ . In particular, for every  $i \in \{1, \dots, n\}$ , there is  $\mathbf{f}_{h_i}^{(t_i)} \in X_i$  for some  $h_i \in \{1, \dots, m\}$  and  $t_i \in \{1, \dots, r\}$ .

We claim that  $M = \{E_1, \dots, E_s\}$  is a perfect matching of  $\mathcal{H}$ . To show this, consider a vertex  $v_i \in V(\mathcal{H})$ . We prove that  $v_i \in E_{h_i}$ . For sake of contradiction, assume that  $v_i \notin E_{h_i}$ . Then  $\|\mathbf{f}_{h_i}^{(t_i)} - \mathbf{c}_i\|_0 = \|\mathbf{f}_{h_i} - \mathbf{v}_i\|_0 = 3r$  by (16) and

$$\begin{aligned} \text{cost}_0(X_1, \dots, X_k) &= \sum_{j=1}^n \text{cost}_0(X_j) \geq \sum_{j=1}^n \|\mathbf{f}_{h_j}^{(t_j)} - \mathbf{c}_j\|_0 = \sum_{j=1}^n \|\mathbf{f}_{h_j} - \mathbf{v}_j\|_0 \\ &\geq (3r - 2)n + 2 > B; \end{aligned}$$

a contradiction with  $\text{cost}_0(X_1, \dots, X_k) \leq B$ . Hence, every vertex of  $V(\mathcal{H})$  is saturated by some hyperedge of  $M$ . Since  $|M| = s = \frac{n}{r}$ , we have that the hyperedges of  $M$  are pairwise disjoint, that is,  $M$  is a matching. Since every vertex is saturated and  $M$  is a matching,  $M$  is a perfect matching.

This concludes the proof of our claim that  $\mathcal{H}$  has a perfect matching if and only if  $(\mathbf{X}, k, B)$  is a yes-instance of DECISION EQUAL CLUSTERING.

Observe that  $B = (3r - 2)n$  in the reduction meaning that  $B = \mathcal{O}(n^2)$ . Since DECISION EQUAL CLUSTERING is in NP, there is a polynomial reduction from DECISION EQUAL CLUSTERING to PERFECT  $r$ -SET MATCHING. Thus, if DECISION EQUAL CLUSTERING has a polynomial kernel when parameterized by  $B$ , then PERFECT  $r$ -SET MATCHING has a polynomial kernel when parameterized by the number of vertices of the input hypergraph. This leads to a contradiction with Corollary 1 and completes the proof of the theorem. □

### 4.2 Polynomial Kernel for $k + B$ Parameterization

In this subsection, we prove Theorem 3 that we restate here.

**Theorem 3** *For every nonnegative integer constant  $p$ , DECISION EQUAL CLUSTERING admits a polynomial kernel when parameterized by  $k$  and  $B$ , where the output collection of points has  $\mathcal{O}(kB)$  points of  $\mathbb{Z}^{d'}$  with  $d' = \mathcal{O}(kB^{p+1})$  and each coordinate of a point takes an absolute value of  $\mathcal{O}(kB^2)$ .*

**Proof** Let  $(\mathbf{X}, k, B)$  be an instance of DECISION EQUAL CLUSTERING with  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , where the points are from  $\mathbb{Z}^d$ . Recall that  $n$  is divisible by  $k$ .

Suppose  $\frac{n}{k} \geq 4B + 1$ . Then we can apply the algorithm from Lemma 6. If the algorithm returns that there is no equal  $k$ -clustering of cost at most  $B$ , then the kernelization algorithm returns a trivial no-instance of DECISION EQUAL CLUSTERING. Otherwise, if  $\text{Opt}(X, k) \leq B$ , then the algorithm returns a trivial yes-instance.

Assume from now that  $\frac{n}{k} \leq 4B$ , that is,  $n \leq 4Bk$ . Then we apply the algorithm from Lemma 7. If this algorithm reports that there is no equal  $k$ -clustering of cost at most  $B$ , then the kernelization algorithm returns a trivial no-instance of DECISION EQUAL CLUSTERING. Otherwise, the algorithm from Lemma 7 returns a collection of  $n \leq 4Bk$  points  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$  of  $\mathbb{Z}^{d'}$  satisfying conditions (i)–(iii) of the lemma. By (i), we obtain that the instances  $(\mathbf{X}, k, B)$  and  $(\mathbf{Y}, k, B)$  of DECISION EQUAL CLUSTERING are equivalent. By (ii), we have that the dimension  $d' = \mathcal{O}(k(B^{p+1}))$ , and by (iii), each coordinate of a point takes an absolute value of  $\mathcal{O}(kB^2)$ . Thus,  $(\mathbf{Y}, k, B)$  is a required kernel.  $\square$

## 5 APX-hardness of EQUAL CLUSTERING

In this section, we prove APX-hardness of DECISION EQUAL CLUSTERING w.r.t. Hamming ( $\ell_0$ ) and  $\ell_1$  distances. The constructed hard instances consist of high-dimensional binary (0/1) points. As the  $\ell_0$  and  $\ell_1$  distances between any two binary points are the same, we focus on the case of  $\ell_0$  distances. Our reduction is from 3-DIMENSIONAL MATCHING (3DM), where we are given three disjoint sets of elements  $X, Y$  and  $Z$  such that  $|X| = |Y| = |Z| = n$  and a set of  $m$  triples  $T \subseteq X \times Y \times Z$ . In addition, each element of  $W := X \cup Y \cup Z$  appears in at most 3 triples. A set  $M \subseteq T$  is called a matching if no element of  $W$  is contained in more than one triple of  $M$ . The goal is to find a maximum cardinality matching. We need the following theorem due to Petrank [56].

**Theorem 4** (Restatement of Theorem 4.4 from [56]) *There exists a constant  $0 < \gamma < 1$ , such that it is NP-hard to distinguish the instances of the 3DM problem in which a perfect matching exists, from the instances in which there is a matching of size at most  $(1 - \gamma)n$ .*

Here  $\gamma$  should be seen as a very small constant close to 0. We use the construction described in Section 4.1, with a small modification.

We are given an instance of 3DM. Let  $N = 3n$ , the total number of elements. We construct a binary matrix  $A$  of dimension  $6N \times (2N + 3m)$ . For each element, we take 2 columns and for each triple 3 columns. The  $6N$  row indexes are partitioned into  $N$  parts each of size 6. In particular, let  $R_1 = \{1, \dots, 6\}$ ,  $R_2 = \{7, \dots, 12\}$  and so on. For the  $i$ -th element, we construct the column  $a_i$  of length  $6N$  which has 1 corresponding to the indexes in  $R_i$  and 0 elsewhere.

Recall that each element can appear in at most 3 triples. For each element  $x$ , consider any arbitrary ranking of the triples that contain it. The occurrence of  $x$  in a triple with rank  $j$  is called its  $j$ -th occurrence for  $1 \leq j \leq 3$ . For example, suppose  $x$  appears in three triples  $t_w, t_y$  and  $t_z$ . One can consider the ranking  $1.t_w, 2.t_y, 3.t_z$ . Then, the



occurrence of  $x$  in  $t_y$  is called 2-nd occurrence. Let  $v_i^j$  be the  $j$ -th index of  $R_i$  for  $1 \leq i \leq N$ ,  $1 \leq j \leq 3$ . For each triple  $t$  with  $j_1$ -,  $j_2$ - and  $j_3$ -th occurrences of the elements  $p$ ,  $q$  and  $r$  in  $t$ , respectively, we construct the column  $b_t$  of length  $6N$  which has 1 corresponding to the indexes  $v_p^{j_1}$ ,  $v_q^{j_2}$  and  $v_r^{j_3}$ , and 0 elsewhere.

The triple columns are defined in a different way in our reduction in Section 4.1 where for each triple and each element, a fixed index is set to 1. But, we set different indexes based on the occurrences of the element. This ensures that for two different triple columns  $b_s$  and  $b_t$ , their Hamming distance  $d_H(b_s, b_t) = 6$ . Note that  $d_H(a_i, b_t) = 7$  if the element  $i$  is in triple  $t$ , otherwise  $d_H(a_i, b_t) = 9$ . Set cluster size to be 3 and the number of clusters  $k$  to be  $(2N/3) + m$ . We will prove the following lemma.

**Lemma 10** *If there is a perfect matching, there is a feasible clustering of cost  $7N$ . If all matchings have size at most  $(1 - \gamma)n$ , any feasible clustering has cost at least  $7(1 - \gamma)N + (23/3)\gamma N$ .*

Note that it is sufficient to prove the above lemma for showing the APX-hardness of the problem. The proof of the first part of the lemma is exactly the same as in the previous construction. We will prove the second part. To give some intuition of the cost suppose there is a matching of the maximum size  $(1 - \gamma)n$ . Then we can cluster the matched elements and triples in the same way as in the perfect matching case by paying a cost of  $7(1 - \gamma)N$ . Now for each unmatched element, we put its two columns in a cluster. Now we have  $\gamma N$  clusters with one free slot in each. One can fill in these slots by columns corresponding to  $\gamma N/3$  unmatched triples. All the remaining unmatched triples form their own cluster. Now, consider an unmatched triple  $s$  whose 3 columns are used to fill in slots of unmatched elements  $p$ ,  $q$ , and  $r$ . As this triple was not matched, it cannot contain all these three elements, i.e. it can contain at most 2 of these elements. Thus, for at least one element, the cost of the cluster must be 9. Therefore, the total cost of the three clusters corresponding to  $p$ ,  $q$ , and  $r$  is at least  $7 + 7 + 9 = 23$ . The total cost corresponding to all  $\gamma N/3$  unmatched triples is then  $(23/3)\gamma N$ . We will show that one cannot find a feasible clustering of lesser cost.

For our convenience, we will prove the contrapositive of the second part of the above lemma: if there is a feasible clustering of cost less than  $7(1 - \gamma)N + (23/3)\gamma N$ , then there is a matching of size greater than  $(1 - \gamma)n$ . So, assume that there is such a clustering. Let  $c_1, c_2, \dots, c_k$  be the cluster centers.

By Lemma 9, we can assume that if a column  $f$  of  $A$  is a center of a cluster  $C$ , all the columns equal to  $f$  are in  $C$ . We will use this in the following. A center  $c_i$  is called an element center if  $c_i$  is an element column. Suppose the given clustering contains  $\ell$  clusters with element centers for some  $\ell$ . WLOG, we assume that these are the first  $\ell$  clusters.

**Lemma 11** *If the cost of the given clustering is less than  $7(1 - \gamma)N + (23/3)\gamma N$ ,  $\ell > (1 - 2\gamma/9)N$ .*

**Proof** Note that if a cluster center is an element column, then by Lemma 9 we can assume that both element columns are present in the cluster. Thus, in our case, each of the first  $\ell$  clusters contains two element columns and some other column. Now,

each of these  $\ell$  other columns can be either a column of some other element or a triple column. Let  $\ell_1$  of these be element columns and  $\ell_2$  of these be triple columns, where  $\ell = \ell_1 + \ell_2$ . For each cluster corresponding to these  $\ell_1$  element columns, the cost is 12, as  $d_H(a_i, a_j) = 12$  for all  $i, j$ . Similarly, for each cluster corresponding to the  $\ell_2$  triple columns, the cost is at least 7, as  $d_H(a_i, b_t) \geq 7$  for all  $i, t$ .

Note that out of  $2N$  element columns,  $2\ell + \ell_1$  are in the first  $\ell$  clusters. The rest of the element columns are in the other clusters. Now there can be two cases: such a column is in a cluster that contains (i) at least 2 element columns and (ii) exactly one element column.

**Claim 5.1** The cost of each element column which is not in the first  $\ell$  clusters is at least 5 in the first case.

**Proof** Consider such a column  $a_i$  and let  $c_j$  be the center of the cluster that contains  $a_i$ . Note that the only 1 entries in  $a_i$  are corresponding to the indexes in  $R_i$ . We claim that at most one entry of  $c_j$  corresponding to the indexes in  $R_i$  can be 1. This proves the original claim, as  $|R_i| = 6$ . Consider an index  $z \in R_i$  such that  $c_j[z] = 1$ . As  $c_j$  is not an element column and the centers are defined based on the majority rule, there is a column  $e$  in the cluster with  $e[z] = 1$ . This must be a column of a triple that contains the element  $i$ . By construction,  $e$  does not contain 1 corresponding to the indexes in  $R_i \setminus \{z\}$ . As the third column in the cluster is another element column (as we are in the first case), its entries corresponding to the indexes in  $R_i$  are again 0. Hence, by majority rule, at most one entry of  $c_j$  corresponding to the indexes in  $R_i$  can be 1.  $\square$

Next, we consider case (ii).

**Claim 5.2** Consider a cluster that is not one of the first  $\ell$  clusters and contains exactly one element column. Then, its cost is at least 5. Moreover, the cost of the element column is at least 4.

**Proof** Consider the element column  $a_i$  of the cluster and let  $c_j$  be the center of the cluster. Note that the only 1 entries in  $a_i$  are corresponding to the indexes in  $R_i$ . Now, if the other two (triple) columns in the cluster are the same, there must be at most one entry of them corresponding to the indexes in  $R_i$  that is 1. This is true by the construction of triple columns. Hence, in this case, at most one entry of  $c_j$  corresponding to the indexes in  $R_i$  can be 1 and the cost is at least 5. Otherwise, there can be two distinct triple columns  $b_s$  and  $b_t$  in the cluster and at most two indexes  $z_1, z_2 \in R_i$  such that  $z_1 \neq z_2$  and  $b_s[z_1] = b_t[z_2] = 1$ . By construction of the triple columns, there are no other indices  $z \in R_i \setminus \{z_1, z_2\}$  such that  $b_s[z] = 1$  or  $b_t[z] = 1$ . Thus, by the majority rule, at most two entries of  $c_j$  corresponding to the indices in  $R_i$  can be 1. Hence, the cost of  $a_i$  is at least 4. Now, as  $b_s$  and  $b_t$  are distinct, the cost of either one of them must be at least 1. It follows that the cost of this cluster is at least 5.  $\square$

Now, again consider the  $2N - 2\ell - \ell_1$  element columns that are not in the first  $\ell$  clusters. Let  $\kappa$  be the number of clusters that are not the first  $\ell$  clusters and contain

exactly 1 element column. This implies that  $2N - 2\ell - \ell_1 - \kappa$  element columns are contained in the clusters which are not the first  $\ell$  clusters and contain at least 2 element columns. By Claim 5.1, the cost of each such column is at least 5. By Claim 5.2, the cost of each of the  $\kappa$  clusters defined above is at least 5.

It follows that the total cost of the clustering is  $12\ell_1 + 7\ell_2 + (2N - 2\ell - \ell_1 - \kappa)5 + 5\kappa = 10N - 3\ell$ , as  $\ell = \ell_1 + \ell_2$ . Now, given that the cost is less than  $7(1 - \gamma)N + (23/3)\gamma N$ .

$$\begin{aligned} 10N - 3\ell &< 7(1 - \gamma)N + (23/3)\gamma N = 7N + 2\gamma N/3 \\ 3N - 3\ell &< 2\gamma N/3 \\ \ell &> (1 - 2\gamma/9)N \end{aligned}$$

□

Like before, let  $\ell_2$  be the number of clusters out of the first  $\ell$  clusters such that  $\ell_2$  contains a triple column.

**Claim 5.3**  $\ell_2 > (1 - 2\gamma/3)N$ .

**Proof** Again consider the cost of the given clustering. The cost of the  $\ell_2$  clusters is at least 7. The cost of the remaining  $\ell - \ell_2$  clusters is exactly 12 as before. Now, as  $\ell > (1 - 2\gamma/9)N$  by Lemma 11,

$$\begin{aligned} 7\ell_2 + 12((1 - 2\gamma/9)N - \ell_2) &< 7(1 - \gamma)N + (23/3)\gamma N = 7N + 2\gamma N/3 \\ 7\ell_2 + 12N - 24\gamma N/9 - 12\ell_2 &< 7N + 2\gamma N/3 \\ 5\ell_2 &> 5N - 30\gamma N/9 \\ \ell_2 &> (1 - 2\gamma/3)N \end{aligned}$$

□

We show that out of the  $\ell_2$  elements corresponding to these  $\ell_2$  clusters, more than  $(1 - \gamma)N$  elements must be matched.

**Lemma 12** *There is a matching that matches more than  $(1 - \gamma)N$  elements.*

**Proof** Consider the set of elements corresponding to the  $\ell_2$  clusters, each of which contains a triple column. Let  $M$  be a maximum matching involving these elements and triples that matches  $\mu$  elements. We will show that  $\mu > (1 - \gamma)N$ . The total cost of the clusters corresponding to these matched elements is  $7\mu$ . Let  $\ell_1$  be the number of clusters out of the first  $\ell$  clusters that contain all element columns (see Fig. 2). The total cost of these clusters is  $12\ell_1$ . Note that  $3\ell_1$  columns are involved in these clusters. For the remaining at least  $2(N - \mu) - 3\ell_1$  element columns and correspondingly at least  $N - \mu - 3\ell_1/2$  elements, the corresponding columns can either be in one cluster along with a triple column or split into two clusters. Let  $\ell_3$  be the number of

such elements whose columns are in one cluster along with a triple column. Also, let  $\ell_4$  be the remaining elements whose columns are split into two clusters (see Fig. 2). By Claims 5.2 and 5.1, the cost of each split column is at least 4. Thus, the total cost corresponding to these  $\ell_4$  elements is at least  $8\ell_4$ . Now, we compute the cost corresponding to the  $\ell_3$  elements whose columns are in one cluster along with a triple column. Consider the set of triples involved in these clusters. Also, let  $T_1$  be the set of triples whose three columns appear in these  $\ell_3$  clusters. The cost of such triple columns is at least  $7 + 7 + 9 = 23$ , as they are not a part of the maximum matching. Let  $\ell_5$  be the number of clusters among the  $\ell_3$  clusters where the triples in  $T_1$  do not appear and  $T_2$  be the set of associated triples. Each triple in  $T_2$  thus appears in at most 2 clusters among the  $\ell_3$  clusters (see Fig. 2). Let  $T_3 \subseteq T_2$  be the set of triples each of which is only associated with the clusters of cost 7 and  $\ell_6$  be the number of these clusters. As these triples are not part of the maximum matching, each of them can cover at most two unmatched elements. Thus, the size of  $T_3$  is at least  $\ell_6/2$ . Note that, by definition, at least one column of each such triple does not belong to the first  $\ell$  clusters. We compute the cost of these triple columns. If such a triple column appears in all triple column clusters, the cost of the column is at least 3, by the construction of the triple columns and noting that two copies of the column cannot appear in the cluster. If such a triple is in a cluster with only one element column, its cost must be at least 2, as the element columns' at most one 1 entry can coincide with the 1 entries of the column. Now, if such a triple column appears in a cluster with two element columns, then the cost of the column is at least 1. However, the cost of the element columns must be at least 10. We charged each such element column a cost of 4 while charging the split columns corresponding to the  $\ell_4$  elements. So, we can charge  $10 - 8 = 2$  additional cost to those element columns. Instead, we charge this to the triple column. Thus, its charged cost is  $1 + 2 = 3$ . Thus, the total cost corresponding to the triples in  $T_3$  is at least  $(\ell_6/2) \cdot 2$ .

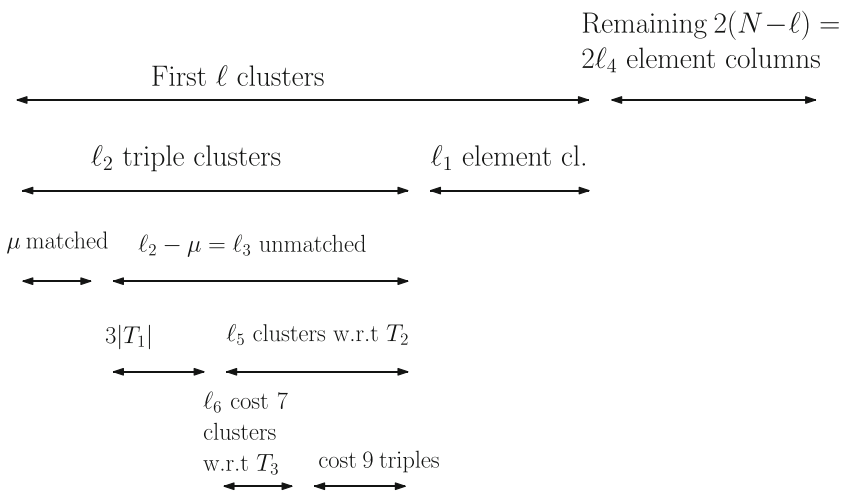


Fig. 2 Hierarchy of the clusters. Illustration of the proof of Lemma 12

The total cost of the clustering is at least,

$$\begin{aligned}
 & 7\mu + 12\ell_1 + 8\ell_4 + (23/3)|T_1| + (\ell_5 - \ell_6)((7 + 9)/2) + 7\ell_6 + (\ell_6/2) \cdot 2 \\
 = & 7\mu + 12\ell_1 + 8\ell_4 + (23/3)(\ell_3 - \ell_5) + 8\ell_5 && (\text{as } 3|T_1| = \ell_3 - \ell_5) \\
 \geq & 7\mu + 12\ell_1 + 8\ell_4 + (23/3)\ell_3 \\
 \geq & 7\mu + 12\ell_1 + (23/3)(\ell_3 + \ell_4) \\
 \geq & 7\mu + 12\ell_1 + (23/3)(N - \mu - 3\ell_1/2) && (\text{as } \ell_3 + \ell_4 \geq N - \mu - 3\ell_1/2) \\
 = & 7\mu + (23/3)(N - \mu) + \ell_1/2 \\
 \geq & (23/3)N - (2/3)\mu && (\text{as } \ell_1 \geq 0)
 \end{aligned}$$

Now, we know a strict upper bound on this cost. Thus,

$$\begin{aligned}
 (23/3)N - (2/3)\mu &< 7N + (2/3)\gamma N \\
 (23/3 - 7)N - (2/3)\gamma N &< (2/3)\mu \\
 (2/3)N(1 - \gamma) &< (2/3)\mu \\
 \mu &> (1 - \gamma)N
 \end{aligned}$$

□

We summarize the results of this section in the following theorem.

**Theorem 5** *There exists a constant  $\varepsilon_c > 0$ , such that it is NP-hard to obtain a  $(1 + \varepsilon_c)$ -approximation for EQUAL CLUSTERING with  $\ell_0$  (or  $\ell_1$ ) distances, even if the input points are binary, that is, are from  $\{0, 1\}^d$ .*

## 6 Conclusion

We initiated the study of lossy kernelization for clustering problems and proved that PARAMETERIZED EQUAL CLUSTERING admits a 2-approximation kernel. It is natural to ask whether the approximation factor may be improved. In particular, does the problem admit a *polynomial size approximate kernelization scheme* (PSAKS) that is a lossy kernelization analog of PTAS (we refer to [8] for the definition)? Note that we proved that EQUAL CLUSTERING is APX-hard and this refutes the existence of PTAS and makes it natural to ask the question about PSAKS. We also believe that it is interesting to consider the variants of the considered problems for means instead of medians. Here, the cost of a collection of points  $\mathbf{X} \subseteq \mathbb{Z}^d$  is defined as  $\min_{\mathbf{c} \in \mathbb{R}^d} \sum_{\mathbf{x} \in \mathbf{X}} \|\mathbf{c} - \mathbf{x}\|_p^p$  for  $p \geq 1$ . Clearly, if  $p = 1$ , that is, in the case of the Manhattan norm, our results hold. However, for  $p \geq 2$ , we cannot translate our results directly because our arguments rely on the triangle inequality. We would like to conclude the paper by underlining our belief that lossy kernelization may be a natural tool for the lucrative area of approximation algorithms for clustering problems.

**Acknowledgements** A preliminary version of these results appeared as an extended abstract in the proceedings of CSR 2022. The research leading to these results have been supported by the Research Council of Norway via the project BWCA (grant no. 314528), European Research Council (ERC) via grant LOPPRE, reference 819416, and DFG Research Group ADYN via grant DFG 411362735.

**Funding** Open access funding provided by University of Bergen (incl Haukeland University Hospital)

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer (2013)
- Lokshtanov, D., Panolan, F., Ramanujan, M.S., Saurabh, S.: Lossy kernelization. In: Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC), pp. 224–237. ACM (2017)
- Eiben, E., Kumar, M., Mouawad, A.E., Panolan, F., Siebertz, S.: Lossy kernels for connected dominating set on sparse graphs. *SIAM J. Discret. Math.* **33**(3), 1743–1771 (2019)
- Siebertz, S.: Lossy kernels for connected distance- $r$  domination on nowhere dense graph classes. *CoRR abs/1707.09819* (2017)
- Krithika, R., Misra, P., Rai, A., Tale, P.: Lossy kernels for graph contraction problems. In: Proceedings of the 36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 65, pp. 23:1–23:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
- Agrawal, A., Saurabh, S., Tale, P.: On the parameterized complexity of contraction to generalization of trees. *Theory Comput. Syst.* **63**(3), 587–614 (2019)
- Dvorák, P., Feldmann, A.E., Knop, D., Masarik, T., Masarik, T., Veselý, P.: Parameterized approximation schemes for steiner trees with small number of steiner vertices. *SIAM J. Discret. Math.* **35**(1), 546–574 (2021)
- Fomin, F.V., Lokshtanov, D., Saurabh, S., Zehavi, M.: Kernelization. Cambridge University Press, Cambridge (2019). Theory of parameterized preprocessing
- Feldmann, A.E., Karthik, C.S., Lee, E., Manurangsi, P.: A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms* **13**(6), 146 (2020)
- Cohen-Addad, V., Gupta, A., Kumar, A., Lee, E., Li, J.: Tight FPT approximations for  $k$ -median and  $k$ -means. In: Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP), *LIPIcs*, vol. 132, pp. 42:1–42:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
- Cohen-Addad, V., Li, J.: On the fixed-parameter tractability of capacitated clustering. In: 46th International Colloquium on Automata, Languages, and Programming, (ICALP), *LIPIcs*, vol. 132, pp. 41:1–41:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
- Har-Peled, S., Mazumdar, S.: On coresets for  $k$ -means and  $k$ -median clustering. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC), pp. 291–300. ACM (2004)
- Höppner, F., Klawonn, F.: Clustering with size constraints. In: L.C. Jain, M. Sato-Ilic, M. Virvou, G.A. Tsihrintzis, V.E. Balas, C. Abeynayake (eds.) *Computational Intelligence Paradigms, Innovative Applications*, vol. 137, pp. 167–180. Springer (2008)
- Vallejo-Huanga, D., Morillo, P., Ferri, C.: Semi-supervised clustering algorithms for grouping scientific articles. In: International Conference on Computational Science (ICCS), *Procedia Computer Science*, vol. 108, pp. 325–334. Elsevier (2017)

15. Zhang, Z., Li, T., Ding, C., Zhang, X.: Binary matrix factorization with applications. In: ICDM'07, pp. 391–400. IEEE (2007)
16. Li, T.: A general model for clustering binary data. In: KDD'05, pp. 188–197 (2005)
17. Sweeney, L.:  $k$ -anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **10**(05), 557–570 (2002)
18. Ding, H., Xu, J.: A unified framework for clustering constrained data without locality property. *Algorithmica* **82**(4), 808–852 (2020)
19. Bhattacharya, A., Jaiswal, R., Kumar, A.: Faster Algorithms for the Constrained  $k$ -means Problem. *Theory of Computing Systems* **62**(1), 93–115 (2018)
20. Bandyapadhyay, S., Fomin, F.V., Simonov, K.: On coresets for fair clustering in metric and euclidean spaces and their applications. In: 48th International Colloquium on Automata, Languages, and Programming (ICALP), *LIPICs*, vol. 198, pp. 23:1–23:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)
21. Kumar, A., Sabharwal, Y., Sen, S.: Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM* **57**(2), 5:1–5:32 (2010)
22. Alon, N., Sudakov, B.: On two segmentation problems. *Journal of Algorithms* **33**(1), 173–184 (1999)
23. Feldman, D., Langberg, M.: A unified framework for approximating and clustering data. In: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC), pp. 569–578. ACM (2011)
24. Baker, D., Braverman, V., Huang, L., Jiang, S.H.C., Krauthgamer, R., Wu, X.: Coresets for clustering in graphs of bounded treewidth. In: International Conference on Machine Learning, pp. 569–579. PMLR (2020)
25. Feldman, D., Schmidt, M., Sohler, C.: Turning big data into tiny data: Constant-size coresets for  $k$ -means, pca, and projective clustering. *SIAM Journal on Computing* **49**(3), 601–657 (2020)
26. Sohler, C., Woodruff, D.P.: Strong coresets for  $k$ -median and subspace approximation: Goodbye dimension. In: Proceedings of the 59th Annual Symposium on Foundations of Computer Science (FOCS), pp. 802–813. IEEE (2018)
27. Aggarwal, C.C., Reddy, C.K. (eds.): *Data Clustering: Algorithms and Applications*. CRC Press (2013)
28. Charikar, M., Guha, S., Tardos, É., Shmoys, D.B.: A constant-factor approximation algorithm for the  $k$ -median problem. *J. Comput. Syst. Sci.* **65**(1), 129–149 (2002)
29. Jain, K., Vazirani, V.V.: Approximation algorithms for metric facility location and  $k$ -median problems using the primal-dual schema and lagrangian relaxation. *J. ACM* **48**(2), 274–296 (2001)
30. Li, S., Svensson, O.: Approximating  $k$ -median via pseudo-approximation. *SIAM J. Comput.* **45**(2), 530–547 (2016)
31. Byrka, J., Pensyl, T.W., Rybicki, B., Srinivasan, A., Trinh, K.: An improved approximation for  $k$ -median and positive correlation in budgeted optimization. *ACM Trans. Algorithms* **13**(2), 23:1–23:31 (2017)
32. Ahmadian, S., Norouzi-Fard, A., Svensson, O., Ward, J.: Better guarantees for  $k$ -means and euclidean  $k$ -median by primal-dual algorithms. *SIAM J. Comput.* **49**(4) (2020)
33. Manurangsi, P., Raghavendra, P.: A birthday repetition theorem and complexity of approximating dense csps. In: 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
34. Arora, S., Raghavan, P., Rao, S.: Approximation schemes for euclidean  $k$ -medians and related problems. In: J.S. Vitter (ed.) Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23–26, 1998, pp. 106–113. ACM (1998)
35. Cohen-Addad, V., Mathieu, C.: Effectiveness of local search for geometric optimization. In: L. Arge, J. Pach (eds.) 31st International Symposium on Computational Geometry, SoCG 2015, June 22–25, 2015, Eindhoven, The Netherlands, *LIPICs*, vol. 34, pp. 329–343. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015)
36. Bandyapadhyay, S., Varadarajan, K.R.: On variants of  $k$ -means clustering. In: S.P. Fekete, A. Lubiw (eds.) 32nd International Symposium on Computational Geometry, SoCG 2016, June 14–18, 2016, Boston, MA, USA, *LIPICs*, vol. 51, pp. 14:1–14:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
37. Cohen-Addad, V., Klein, P.N., Mathieu, C.: Local search yields approximation schemes for  $k$ -means and  $k$ -median in euclidean and minor-free metrics. *SIAM J. Comput.* **48**(2), 644–667 (2019)
38. Frigstad, Z., Rezapour, M., Salavatipour, M.R.: Local search yields a PTAS for  $k$ -means in doubling metrics. *SIAM J. Comput.* **48**(2), 452–480 (2019)

39. Cohen-Addad, V., de Mesmay, A., Rotenberg, E., Roytman, A.: The bane of low-dimensionality clustering. In: A. Czumaj (ed.) Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7–10, 2018, pp. 441–456. SIAM (2018)
40. Feng, Q., Zhang, Z., Huang, Z., Xu, J., Wang, J.: A unified framework of FPT approximation algorithms for clustering problems. In: Y. Cao, S. Cheng, M. Li (eds.) 31st International Symposium on Algorithms and Computation, ISAAC 2020, December 14–18, 2020, Hong Kong, China (Virtual Conference), *LIPICs*, vol. 181, pp. 5:1–5:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
41. Basu, S., Davidson, I., Wagstaff, K.: Constrained clustering: Advances in algorithms, theory, and applications. CRC Press (2008)
42. Byrka, J., Rybicki, B., Uniyal, S.: An approximation algorithm for uniform capacitated  $k$ -median problem with  $1 + \varepsilon$  capacity violation. In: Q. Louveaux, M. Skutella (eds.) Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liège, Belgium, June 1–3, 2016, Proceedings, *Lecture Notes in Computer Science*, vol. 9682, pp. 262–274. Springer (2016)
43. Byrka, J., Fleszar, K., Rybicki, B., Spoerhase, J.: Bi-factor approximation algorithms for hard capacitated  $k$ -median problems. In: P. Indyk (ed.) Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4–6, 2015, pp. 722–736. SIAM (2015)
44. Chuzhoy, J., Rabani, Y.: Approximating  $k$ -median with non-uniform capacities. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23–25, 2005, pp. 952–958. SIAM (2005)
45. Demirci, H.G., Li, S.: Constant approximation for capacitated  $k$ -median with  $(1 + \varepsilon)$ -capacity violation. In: 43rd International Colloquium on Automata, Languages, and Programming (ICALP), pp. 73:1–73:14 (2016)
46. Li, S.: On uniform capacitated  $k$ -median beyond the natural LP relaxation. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 696–707 (2015)
47. Li, S.: On uniform capacitated  $k$ -median beyond the natural LP relaxation. *ACM Trans. Algorithms* **13**(2), 22:1–22:18 (2017)
48. Cohen-Addad, V.: Approximation schemes for capacitated clustering in doubling metrics. In: S. Chawla (ed.) Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5–8, 2020, pp. 2241–2259. SIAM (2020)
49. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015)
50. Fomin, F.V., Golovach, P.A., Purohit, N.: Parameterized complexity of categorical clustering with size constraints. *CoRR* [abs/2104.07974](https://arxiv.org/abs/2104.07974) (2021)
51. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.* **2**, 83–97 (1955)
52. Lovász, L., Plummer, M.D.: Matching theory. AMS Chelsea Publishing, Providence, RI (2009)
53. Fomin, F.V., Golovach, P.A., Panolan, F.: Parameterized low-rank binary matrix approximation. *Data Min. Knowl. Discov.* **34**(2), 478–532 (2020)
54. Kleinberg, J.M., Tardos, É.: Algorithm design. Addison-Wesley (2006)
55. Dell, H., Marx, D.: Kernelization of packing problems. *CoRR* [abs/1812.03155](https://arxiv.org/abs/1812.03155) (2018)
56. Petrank, E.: The hardness of approximation: Gap location. *Comput. Complex.* **4**, 133–157 (1994)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.