Journal of Computer and System Sciences 117 (2021) 50–74

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Parameterized k-Clustering: Tractability island[☆]

Fedor V. Fomin, Petr A. Golovach, Kirill Simonov[*]

Department of Informatics, University of Bergen, Thormøhlens Gate 55, 5008 Bergen, Norway

ARTICLE INFO

Article history:
Received 18 December 2019
Received in revised form 17 August 2020
Accepted 12 October 2020
Available online 19 November 2020

Keywords:
Clustering
Parameterized complexity
k-means
k-median

ABSTRACT

In k-Clustering we are given a multiset of n vectors $X \subset \mathbb{Z}^d$ and a nonnegative number D, and we need to decide whether X can be partitioned into k clusters $C_1, \dots, C_k$ such that the cost

$$\sum_{i=1}^{k} \min_{c_i \in \mathbb{R}^d} \sum_{x \in C_i} \|x - c_i\|_p^p \le D,$$

where $\|\cdot\|_p$ is the L_p-norm. For $p = 2$, k-Clustering is k-Means. We study k-Clustering from the perspective of parameterized complexity. The problem is known to be NP-hard for $k = 2$ and also for $d = 2$. It is a long-standing open question, whether the problem is fixed-parameter tractable (FPT) for the combined parameter $d + k$. In this paper, we focus on the parameterization by D. We complement the known negative results by showing that for $p = 0$ and $p = \infty$, k-Clustering is W[1]-hard when parameterized by D. Interestingly, we discover a tractability island of k-Clustering: for every $p \in (0, 1]$, k-Clustering is solvable in time $2^{\mathcal{O}(D \log D)}(nd)^{\mathcal{O}(1)}$.

1. Introduction

Recall that for $p > 0$, the *Minkowski* or L_p*-norm* of a vector $x = (x[1], \dots, x[d]) \in \mathbb{R}^d$ is defined as

$$\|x\|_p = \Big(\sum_{i=1}^{d} |x[i]|^p\Big)^{1/p}.$$

Respectively, we define the *(L_p-norm) distance* between two vectors $x = (x[1], \dots, x[d])$ and $y = (y[1], \dots, y[d])$ as

$$\mathrm{dist}_p(x, y) = \|x - y\|_p^p = \sum_{i=1}^{d} |x[i] - y[i]|^p.$$

We also consider dist_p for $p = 0$ and $p = \infty$. For $p = 0$, dist_p is L_0 (or the Hamming) distance, that is the number of different coordinates in x and y:

[☆] This work is supported by the Research Council of Norway via the project "MULTIVAL". The grant number is 263317.

[*] Corresponding author.
E-mail addresses: Fedor.Fomin@uib.no (F.V. Fomin), Petr.Golovach@uib.no (P.A. Golovach), Kirill.Simonov@uib.no (K. Simonov).

https://doi.org/10.1016/j.jcss.2020.10.005

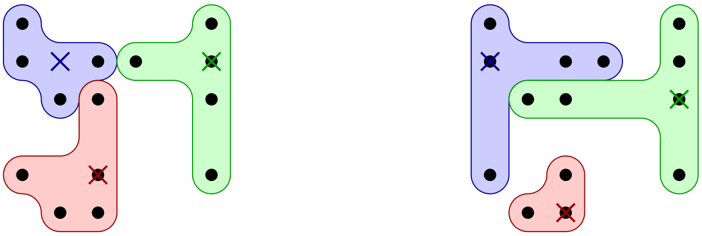


Fig. 1. Optimal clusterings of the same set of vectors with different distances: dist_1 in the left subfigure, $\mathrm{dist}_{1/4}$ in the right subfigure. Shapes denote clusters, crosses denote cluster centroids.

$$\mathrm{dist}_0(x,y) = |\{i \in \{1,\dots,d\} \mid x[i] \neq y[i]\}|.$$

For $p=\infty$, dist_p is L_∞-distance, which is defined as

$$\mathrm{dist}_\infty(x,y) = \max_{i\in\{1,\dots,d\}} |x[i]-y[i]|.$$

The k-Clustering problem is defined as follows. For a given (multi) dataset of n vectors (points) $X \subset \mathbb{Z}^d$, the task is to find a partition of X into k clusters $C_1,\dots,C_k$ minimizing the cost

$$\sum_{i=1}^{k} \min_{c_i\in\mathbb{R}^d} \sum_{x\in C_i} \mathrm{dist}_p(x,c_i),$$

intuitively, c_i is a centroid of the cluster C_i.

In particular, for $p=1$, dist_p is the L_1-distance and the corresponding clustering problem is known as k-Median. (Often in the literature, k-Median is also used for clustering minimizing the sums of the Euclidean distances.) For $p=2$, dist_p is the L_2 (Euclidean) distance, and then the clustering problem becomes k-Means.

Let us note that optimal clusterings for the same set of vectors can be drastically different for various values of p, as shown in Fig. 1. As we show in the paper, the complexity of k-Clustering also strongly depends on the choice of p.

k-Clustering, and especially k-Median and k-Means, are among the most prevalent problems occurring in virtually every subarea of data science. We refer to the survey of Jain [1] for an extensive overview. While in practice the most common approaches to clustering are based on different variations of Lloyd's heuristic [2], the problem is interesting from the theoretical perspective as well. In particular, there is a vast amount of literature on approximation algorithms for k-Clustering whose behavior can be analyzed rigorously, see e.g. [3–17].

When it comes to exact solutions, we observe the following phenomena. While heuristic algorithms for k-Clustering work surprisingly well in practice, from the perspective of parameterized complexity, k-Clustering is intractable for all previously studied parameterizations, see Table 1. The k-Clustering problem is naturally "multivariate": in addition to the number of points n, there are also parameters like space dimension d, number of clusters k or the cost of clustering D. The problem is known to be NP-complete for $k=2$ [18,19] and for $d=2$ [20,21]. By the classical work of Inaba et al. [22], in the case when both d and k are constants, k-Clustering is solvable in polynomial time $\mathcal{O}(n^{dk+1})$. It is a long-standing open problem whether k-Clustering is FPT parameterized by $d+k$. Under ETH, the lower bound of $n^{\Omega(k)}$, even when $d=4$, was shown by Cohen-Addad et al. in [23] for the settings where the set of potential candidate centers is explicitly given as input. However the lower bound of Cohen-Addad et al. does not generalize to the settings of this paper where any point in Euclidean space can serve as a center. For the special case, when the input consists of binary vectors and the distance is Hamming, the problem is solvable in time $2^{\mathcal{O}(D\log D)}(nd)^{\mathcal{O}(1)}$ [24].

Our results and approaches. In this paper we investigate the dependence of the complexity of k-Clustering on the cost of clustering D. It appears that adding this new "dimension" makes the complexity landscape of k-Clustering intricate and interesting. More precisely, we consider the following problem.

k-Clustering with distance dist

Input: A multiset X of n vectors in $\mathbb{Z}^d$, a positive integer k, and a nonnegative number D.

Task: Decide whether there is a partition of X into k clusters $\{C_i\}_{i=1}^k$ and k vectors $\{c_i\}_{i=1}^k$, called *centroids*, in $\mathbb{R}^d$ such that

$$\sum_{i=1}^{k}\sum_{x\in C_i} \mathrm{dist}(x,c_i) \le D.$$

Let us remark that vector set X (like the column set of a matrix) can contain many equal vectors. Also we consider the situation when vectors from X are integer vectors, while centroid vectors are not necessarily from X. Moreover, coordinates of centroids can be reals.

Our main algorithmic result is the following theorem.

Theorem 1. *k-Clustering with distance dist_p is solvable in time $2^{\mathcal{O}(D\log D)}(nd)^{\mathcal{O}(1)}$ for every $p \in (0, 1]$.*

Thus k-Clustering when parameterized by D is fixed-parameter tractable (FPT) for Minkowski distance dist_p of order $0 < p \leq 1$. In the first step of our algorithm we use color coding to reduce the problem to Cluster Selection, which we find interesting on its own. In Cluster Selection we have t groups of weighted vectors and the task is to select exactly one vector from each group such that the weighted cost of the composite cluster is at most D. More formally,

Cluster Selection with distance dist

Input: A set of m vectors X given together with a partition $X = X_1 \cup \cdots \cup X_t$ into t disjoint sets, a weight function $w : X \to \mathbb{Z}_+$, and a nonnegative number D.

Task: Decide whether it is possible to select exactly one vector x_i from each set X_i such that the total cost of the composite cluster formed by $x_1, \ldots, x_t$ is at most D:

$$\min_{c\in\mathbb{R}^d} \sum_{i=1}^{t} w(x_i) \cdot \mathrm{dist}(x_i, c) \leq D.$$

The Cluster Selection problem is closely related to variants of the well-known Consensus Pattern problem. Namely, for the Hamming distance, the definition of Cluster Selection nearly coincides with the Colored Consensus Strings with Outliers problem studied in [25], only in the latter the alphabet is assumed to be of constant size.

Informally (see Theorem 10 for the precise statement), our reduction shows that if the distance norm satisfies some specific properties (which dist_p satisfies for all p) and if Cluster Selection is FPT parameterized by D, then so is k-Clustering. Therefore, in order to prove Theorem 1, all we need is to show that Cluster Selection is FPT parameterized by D when $p \in (0, 1]$. This is the most difficult part of the proof. Here we invoke the theorem of Marx [26] on the number of subhypergraphs in hypergraphs of bounded fractional edge cover.

Superficially, the general idea of the proof of Theorem 1 is similar to the idea behind the algorithm for Binary r-Means for L_0 from [24]. In both cases, the classical color coding technique of Alon et al. [27] is used as a preprocessing step. However, the further steps in [24] strongly exploit the fact that the data is binary. As we will see in Theorem 2, the existence of an FPT algorithm for k-Clustering in L_0 is highly unlikely. Thus the reductions from [24] cannot be applied in our case, and we need a new approach.

More precisely, for clustering in L_0 we prove the following theorem.

Theorem 2. *With distance dist_0, k-Clustering parameterized by $d + D$ and Cluster Selection parameterized by $d + t + D$ are W[1]-hard.*

In particular, this means that up to a widely-believed assumption in complexity that FPT $\neq$ W[1], Theorem 2 rules out algorithms solving k-Clustering in time $f(d, D) \cdot n^{\mathcal{O}(1)}$ and algorithms solving Cluster Selection in L_0 in time $g(t, d, D) \cdot n^{\mathcal{O}(1)}$ for any functions $f(d, D)$ and $g(t, d, D)$. A similar hardness result holds for L_∞.

Theorem 3. *With distance dist_∞, k-Clustering parameterized by D and Cluster Selection parameterized by $t + D$ are W[1]-hard.*

This naturally brings us to the question: What happens with k-Clustering for $p \in (1, \infty)$, especially for the Euclidean distance, that is $p = 2$. Unfortunately, we are not able to answer this question when the parameter is D only. However, we can prove that

Theorem 4. *k-Clustering and Cluster Selection with distance dist_2 are FPT when parameterized by $d + D$.*

Thus in particular, Theorem 4 implies that k-Clustering with distance dist_2 is FPT parameterized by $d + D$. On the other hand, we prove that

Theorem 5. *Cluster Selection with distance dist_p is W[1]-hard for every $p \in (1, \infty)$ when parameterized by $t + D$.*

Table 1
Complexity of k-Clustering and Cluster Selection.

dist_p	k-Clustering	Cluster Selection
$p=0$	W[1]-hard param. $d+D$ [Theorem 2] NP-c for $k=2$ [19]	W[1]-hard param. $d+t+D$ [Theorem 2]
$0<p\le 1$	$2^{\mathcal{O}(D\log D)}(nd)^{\mathcal{O}(1)}$ [Theorem 1] NP-c for $k=2$ when $p=1$ [19] NP-c for $d=2$ when $p=1$ [20]	$2^{\mathcal{O}(D\log D)}(nd)^{\mathcal{O}(1)}$ [Theorem 15] W[1]-hard param. $t+d$ for $p=1$ [Theorem 20]
$1<p<+\infty$	FPT param. $d+D$ for $p=2$ [Theorem 4] NP-c for $k=2$ when $p=2$ [18] NP-c for $d=2$ when $p=2$ [21]	FPT param. $d+D$ for $p=2$ [Theorem 4] W[1]-hard param. $t+D$ [Theorem 5]
$p=\infty$	W[1]-hard param. D [Theorem 3] NP-c for $k=2$ [Theorem 30]	W[1]-hard param. $t+D$ [Theorem 3]

In particular, Theorem 5 yields that the approach we used to establish the tractability (with parameter D) of k-Clustering for $p=1$ will not work for $p>1$.

We summarize our and previously known algorithmic and hardness results for k-Clustering and Cluster Selection with different distances in Table 1. Observe that Theorem 10 works also in the setting where possible cluster centers are restricted to be from a set given in the input, and so do our algorithmic Theorems 1 and 4 since Cluster Selection is trivially solvable in polynomial time in this setting.

Now we discuss the choice of the parameter D. It might be noted that the regime where the cost of clustering D is small compared to the number of points n, is quite special. Indeed, if the cost of clustering is at most D, then there are but a few points that are not equal to the respective cluster centers. Thus, the problem we study has the spirit of an editing problem: check whether a given instance is close to a "structured" one, where in our case a "structured" instance has at most k distinct points, and closeness is measured via the sum of L_p-distances. Editing problems are extensively studied in the parameterized algorithms literature, ranging from the vast area of graph modification (see e.g. a recent survey by Crespelle et al. [28]) to studies very close to ours, like the Consensus Patterns algorithm by Marx [26], and the study of Binary r-Means by Fomin et al. [24] that is essentially a special case of our k-Clustering problem. And still, even in this highly structured regime, our results show a very intricate picture: for instance, for k-Clustering parameterized just by D, we provide a highly non-trivial FPT algorithm in the case $0<p\le 1$. While on the other hand, conditionally, the same scheme could not lead to an analogous algorithm in the case $p=2$, and there could not be any FPT algorithm at all in the cases $p=0$ and $p=\infty$. Finally we believe that studying k-Clustering with respect to the parameter D is an essential question provided the notorious hardness of the problem. Recall that for the combination of the two other natural parameters, the dimension d and the number of clusters k, only a $O(n^{dk+1})$ algorithm of Inaba et al. is known [22], and the hardness result by Cohen-Addad et al. in [23] serves as a strong indication that a better algorithm might not exist.

Observe that we always consider integer-valued instances. We believe this is the most natural model for studying complexity of k-Clustering with respect to the parameter D. Here it is important to note that considering D as a parameter only makes sense if the input values are suitably discretized. Imagine input vectors could have arbitrary real-valued (or rational-valued) entries, then for a given instance it is always possible to scale the values down by the same factor such that the cost of an optimal clustering is arbitrarily small, but the structure of the instance is completely preserved. Thus the restriction to integer values in our study is a natural discretization of the problem. It allows the parameter D to bear deep structural significance, as our results demonstrate.

The remaining part of this paper is organized as follows. Section 2 contains preliminaries. In Section 3 we prove Theorem 10 which provides us with FPT Turing reduction from k-Clustering to Cluster Selection. Theorem 10 appears to be a handy tool to establish tractability of k-Clustering. In Section 4 we collect the results on clustering with L_p-norm for $p\in(0,1]$. In particular, in Subsection 4.1, we prove Theorem 1, the main algorithmic result of this work, stating that when $p\in(0,1]$, k-Clustering and Cluster Selection admit FPT algorithms with parameter D. In Subsection 4.2 we complement the algorithmic upper bounds with lower bounds by proving that Cluster Selection is W[1]-hard when $p=1$ and parameter is $t+d$ (Theorem 20). In Section 5, we consider the case $p=0$ and prove Theorem 2 establishing W[1]-hardness of k-Clustering and Cluster Selection. Section 6 is devoted to the case $p=\infty$. Here we establish two hardness results about k-Clustering: W[1]-hardness when parameterized by D and NP-hardness in the case $k=2$. In Section 7, we look at the case $p\in(1,\infty)$, with the particular emphasis on the most commonly used case $p=2$. We show that when $d+D$ is the parameter, then Cluster Selection and k-Clustering in the L_2 distance are FPT. We also show that Cluster Selection is W[1]-hard when parameterized by $t+D$ for all $p\in(1,\infty)$. We conclude with open problems in Section 8.

2. Preliminaries and notation

Cluster notation. By a *cluster* we always mean a multiset of vectors in $\mathbb{Z}^d$. For distance dist, the *cost* of a given cluster C is the total distance from all vectors in the cluster to the optimally selected cluster centroid, $\min_{c\in\mathbb{R}^d}\sum_{x\in C}\mathrm{dist}(x,c)$. An *optimal* cluster centroid for a given cluster C is any $c\in\mathbb{R}^d$ minimizing $\sum_{x\in C}\mathrm{dist}(x,c)$. For most of the considered distances, we argue that an optimal cluster centroid could always be chosen among a specific family of vectors (e.g. integral). Whenever we show this, we only consider optimal cluster centroids of the stated form afterwards.

Complexity. A *parameterized problem* is a language $Q \subseteq \Sigma^* \times \mathbb{N}$ where Σ^* is the set of strings over a finite alphabet Σ. Respectively, an input of Q is a pair (I, k) where $I \subseteq \Sigma^*$ and $k \in \mathbb{N}$; k is the *parameter* of the problem. A parameterized problem Q is *fixed-parameter tractable* (FPT) if it can be decided whether $(I, k) \in Q$ in time $f(k) \cdot |I|^{\mathcal{O}(1)}$ for some function f that depends of the parameter k only. Respectively, the parameterized complexity class FPT is composed by fixed-parameter tractable problems. The W-hierarchy is a collection of computational complexity classes: we omit the technical definitions here. The following relation is known amongst the classes in the W-hierarchy: $\mathsf{FPT} = \mathsf{W}[0] \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \ldots \subseteq \mathsf{W}[P]$. It is widely believed that $\mathsf{FPT} \neq \mathsf{W}[1]$, and hence if a problem is hard for the class $\mathsf{W}[i]$ (for any $i \geq 1$) then it is considered to be fixed-parameter intractable. We refer to books [29,30] for the detailed introduction to parameterized complexity.

We also provide conditional lower bounds by making use of the following complexity hypothesis formulated by Impagliazzo, Paturi, and Zane [31].

> **Exponential Time Hypothesis (ETH)**: There is a positive real s such that 3-CNF-SAT with n variables and m clauses cannot be solved in time $2^{sn}(n+m)^{\mathcal{O}(1)}$.

Graphs. In our W[1]-hardness proofs, we heavily employ graph-theoretical notation. Whenever we work with a graph G, we always fix some ordering on the vertices $\pi_V : V(G) \to \{1, \ldots, |V(G)|\}$ and on the edges $\pi_E : E(G) \to \{1, \ldots, |E(G)|\}$. We drop π_V and π_E to simplify notation, so when we consider a vertex $v \in V(G)$ or an edge $e \in E(G)$, v and e also denote integers—numbers of v and e according to the orderings π_V and π_E correspondingly.

Real computations. Since we deal with the problem concerning real-valued matrices, we express the running time of algorithms in terms of number of operations over the reals. This is natural since to compute L_p-distances we have to deal with numbers of form x^p where x is an integer and p is any real number. However, in special cases the bounds hold even for more restrictive models, e.g. when $p = 1$ or $p = 2$ the algorithms operate only on integers of polynomially bounded length.

3. From k-Clustering to Cluster Selection

In this section we present a general scheme for obtaining an FPT algorithm parameterized by D, which is later applied to various distances.

First, we formalize the following intuition: there is no reason to assign equal vectors to different clusters.

Definition 6 *(Initial cluster and regular partition).* For a multiset of vectors X, an inclusion-wise maximal multiset $I \subset X$ such that all vectors in I are equal is called *an initial cluster*.

We say that a clustering $\{C_1, \ldots, C_k\}$ of X is *regular* if for every initial cluster I there is a $i \in \{1, \ldots, k\}$ such that $I \subset C_i$.

Now we prove that it suffices to look only for regular solutions.

Proposition 7. *Let (X, k, D) be a yes-instance to k-Clustering. Then there exists a solution of (X, k, D) which is a regular clustering.*

Proof. Let us assume that the instance (X, k, D) has a solution. There are k clusters $\{C_i\}_{i=1}^k$ and k vectors $\{c_i\}_{i=1}^k$ in $\mathbb{R}^d$ such that $\sum_{i=1}^k \sum_{x \in C_i} \mathrm{dist}(x, c_i) \leq D$. Note that for every $x \in C_j$, $\mathrm{dist}(x, c_j) \geq \min_{1 \leq i \leq k} \mathrm{dist}(x, c_i)$. So if we consider a new clustering $\{C'_1, \ldots, C'_k\}$ with the same centroids, where C'_j are all vectors from X for which c_j is the closest centroid, the total distance does not increase. If we also break ties in favor of the lower index, then for any initial cluster I the same centroid c_i will be the closest, and all vectors from I will end up in C'_i, so $\{C'_1, \ldots, C'_k\}$ is a regular clustering. □

From now on, we consider only regular solutions.

Definition 8 *(Simple and composite clusters).* We say that a cluster C is *simple* if it is an initial cluster. Otherwise, the cluster is *composite*.

Next we state a property of k-Clustering with a particular distance, which is required for the algorithm. Intuitively, each unique vector adds at least some constant to the cluster cost.

Definition 9 *(α-property).* We say that a distance has the *α-property* for some $\alpha > 0$ if for any s the cost of any composite cluster which consists of s initial clusters is at least $\alpha(s-1)$.

In the subsequent sections we show that the α-property holds for all the distance measures for which we present algorithms. Namely, the L_p-distance has the α-property with a certain constant α, for each $p \in [0, 1] \cup \{2, \infty\}$. Analogously

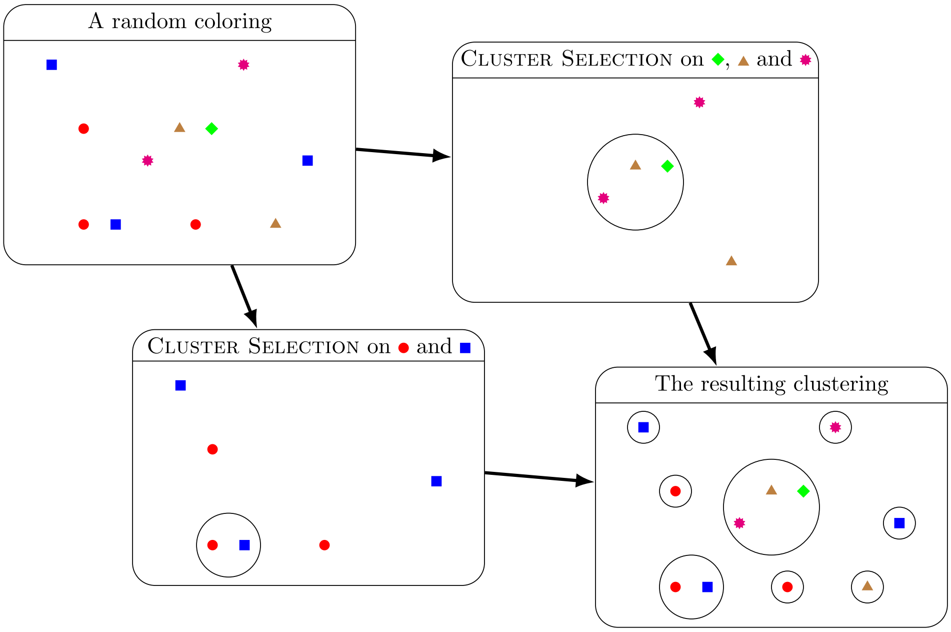


Fig. 2. An illustration of the algorithm in Theorem 10. We start with a particular random coloring and a particular partition of colors $\mathcal{P} = \{P_1, P_2\}$, where $P_1 = \{$●, ■$\}$ and $P_2 = \{$◆, ▲, ✸$\}$. We make two calls to Cluster Selection with respect to P_1 and P_2 and construct the resulting clustering. In the example, all input vectors are distinct.

to the case $p = 2$, one can show that it holds for all other values of p between 1 and ∞ as well, although we do not need this fact.

The Cluster Selection problem defined in the introduction is a key subroutine in our algorithm. In some cases the problem is solvable trivially, but it presents the main challenge for our main algorithmic result with the L_1 distance. The intuition to the weight function in the definition of Cluster Selection is that it represents sizes of initial clusters, that is, how many equal vectors are there.

We also need a procedure to enumerate all values of the cost of each possible cluster, with respect to an optimally selected cluster centroid, that are at most D. It may not be straightforward since not all distances in our consideration are integer. So for the purpose of stating Theorem 10 for general metrics, we assume that the set of all possible optimal cluster costs which are less than D is also given in the input. For the L_p-distances we consider, in the respective algorithmic theorems we show how to provide this set without raising any additional assumptions or increasing the running time. Now we are ready to state the result formally.

Theorem 10. *Assume that the α-property holds,* Cluster Selection *is solvable in time $\Phi(m, d, t, D)$, where Φ is a non-decreasing function of its arguments, and we are given the set $\mathcal{D}$ of all possible optimal cluster costs which are at most D. Then k-*Clustering *is solvable in time*

$$2^{\mathcal{O}(D \log D)}(nd)^{\mathcal{O}(1)}|\mathcal{D}|\Phi(n, d, 2D/\alpha, D).$$

Proof. By the α-property, in any solution there are at most D/α composite clusters, since each contains at least two initial clusters. Moreover, there are at most $2D/\alpha$ initial clusters in all composite clusters.

Thus by Proposition 7, solving k-Clustering is equivalent to selecting at most $T := \lceil 2D/\alpha \rceil$ initial clusters and grouping them into composite clusters such that the total cost of these clusters is at most D. We design an algorithm which, taking as a subroutine an algorithm for Cluster Selection, solves k-Clustering. The algorithm is sketched in Fig. 3, an example is shown in Fig. 2.

To perform the selection and grouping, our algorithm uses the color coding technique of Alon, Yuster, and Zwick from [27]. Consider the input as a family of initial clusters $\mathcal{I}$. We color initial clusters from $\mathcal{I}$ independently and uniformly at random by T colors 1, 2, ..., T. Consider any solution, and the particular set of at most T initial clusters which are included into composite clusters in this solution. These initial clusters are colored by distinct colors with probability at least $\frac{T!}{T^T} \geq e^{-T}$. Now we construct an algorithm for finding a colorful solution.

We consider all possible ways to split colors between clusters (some colors may be unused). Hence we consider all possible families $\mathcal{P} = \{P_1, \dots, P_h\}$ of pairwise disjoint non-empty subsets of $\{c \in \{1, \dots, T\} :$ there exists $J \in \mathcal{I}$ colored by $c\}$. Each family $\mathcal{P}$ corresponds to a partition of the set of colors $\{1, \dots, T\}$ if we add one fictitious subset for colors which are not used in the composite clusters. The total number of partitions does not exceed $T^T = 2^{\mathcal{O}(D \log D)}$.

```
k-Clustering ($X$, $k$, $D$, $\alpha$, $\mathcal{D}$)
  Input  : A multiset $X \subset \mathbb{Z}^d$, a positive integer $k$, real nonnegative values $D$ and $\alpha$, a set $\mathcal{D}$, an algorithm $\mathcal{A}$ for Cluster Selection
  Output: Yes or No
  $T \leftarrow \lceil 2D/\alpha \rceil$
  $\mathcal{I} \leftarrow$ initial clusters of $X$
  for $\lceil e^T \rceil$ iterations do
      Fix a random coloring $c$ of $\mathcal{I}$ with colors $\{1, \dots, T\}$
      for valid partitions $\mathcal{P}$ of $\{1, \dots, T\}$ do
          for $i = 1$ to $|\mathcal{P}|$ do
              $P_i = \{i_1, \dots, i_t\}$
              for $j = 1$ to $t$ do
                  $X_j \leftarrow \emptyset$
                  for $J \in \mathcal{I} : c(J) = i_j$ do
                      $x \leftarrow$ a point from $J$
                      $X_j \leftarrow X_j \cup \{x\}$
                      $w(x) \leftarrow |J|$
              $d_i \leftarrow D + 1$
              foreach $d \in \mathcal{D}$ do
                  if $\mathcal{A}(X_1, \dots, X_t, w, d)$ then
                      $d_i \leftarrow d$
                      BREAK
          if $\sum_{i=1}^{t} d_i \le D$ then
              Yes, STOP
  No, STOP
```

Fig. 3. k-Clustering algorithm from Theorem 10.

When partition $\mathcal{P}$ is fixed, we form clusters by solving instances of Cluster Selection: For each $i \in \{1, \dots, h\}$, we take initial clusters colored by elements of P_i, bundle together those with the same color, and pass the resulting family to Cluster Selection. First note that there cannot be $P \in \mathcal{P}$ of size at most one, since then Cluster Selection has to make a simple cluster while we assume that all clusters obtained from $\mathcal{P}$ are composite. Second, the total number of clusters has to be k, the number of clusters is $|\mathcal{I}| - \sum_{P \in \mathcal{P}} |P| + |\mathcal{P}|$. For each $\mathcal{P}$ we check that both conditions hold, and if not, we discard the choice of $\mathcal{P}$ and move to the next one, before calling the Cluster Selection subroutine.

Next, we formalize how we call the Cluster Selection subroutine. We fix the set of colors $P_i = \{c_1, \dots, c_t\}$, then take the sets $I_j = \{J \in \mathcal{I} : J$ is colored by $c_j\}$ for $j \in \{1, \dots, t\}$. We turn each set of initial clusters I_j into a set of weighted vectors X_j naturally: For each $J \in I_j$, we put one vector $x \in J$ into X_j, and $w(x) := |J|$. The family of sets of vectors $X_1, \dots, X_t$ and the weight function w are the input for Cluster Selection. Then we search for the minimum cluster cost bound $d_i \le D$ from $\mathcal{D}$, for which the instance $(X_1, \dots, X_t, d_i)$ of Cluster Selection is a yes-instance, running each time the algorithm for Cluster Selection.

If for some i setting d_i to D leads to a no-instance, or if $\sum_{i=1}^{h} d_i > D$, then we discard the choice of the partition $\mathcal{P}$ and move to the next one. Otherwise, we report that k-Clustering has a solution and stop. Next, we prove that in this case the solution indeed exists.

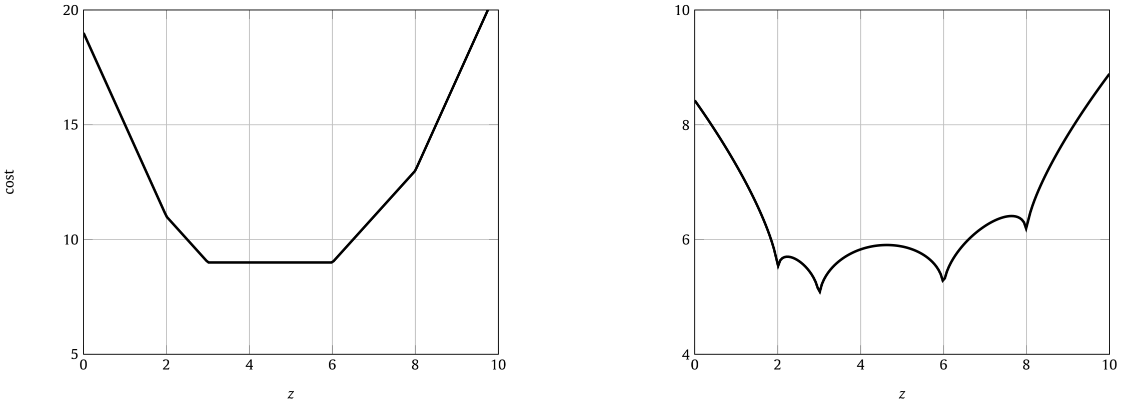
We reconstruct the solution to k-Clustering as follows: For each $i \in \{1, \dots, h\}$ the corresponding to $P_i = \{c_1, \dots, c_t\}$ instance of Cluster Selection has a solution $\{x_1, \dots, x_t\}$. For each $j \in \{1, \dots, t\}$, consider the corresponding initial cluster J_j consisting of $w(x_j)$ vectors equal to x_j. For each $i \in \{1, \dots, h\}$ we obtain a composite cluster $\cup_{j=1}^{t} J_j$, all other clusters are simple. So the total cost is $\sum_{i=1}^{h} d_i$, which is at most D. Thus, if the algorithm finds a solution, then (X, d, D) is a yes-instance.

In the opposite direction. If there is a solution to k-Clustering, then there is a regular solution, and with probability at least e^{-T} initial clusters which are parts of composite clusters in this solution are colored by distinct colors. Then, there is a partition $\mathcal{P} = \{P_1, \dots, P_h\}$ which corresponds to this solution. This partition is obtained as follows: put into P_1 colors from the first composite cluster, into P_2 from the second and so on. At some point our algorithm checks the partition $\mathcal{P}$, and as it finds the optimal cost value for each cluster, then it is at most the cost of the corresponding cluster of the solution from which we started.

To analyze the running time, we consider $2^{\mathcal{O}(D \log D)}$ partitions $\mathcal{P}$, for each $\mathcal{P}$ we $|\mathcal{P}| = \mathcal{O}(D)$ times search for optimal d_i. And for each of $|\mathcal{D}|$ possible values[1] of d_i we make one call to the Cluster Selection algorithm, which takes time at most $\Phi(n, d, T, D)$.

To amplify the error probability to be at most $1/e$, we do $N = \lceil e^T \rceil$ iterations of the algorithm, each time with a new random coloring. As each iteration succeeds with probability at least e^{-T}, the probability of not finding a colorful solution after N iterations is at most $(1 - e^{-T})^{e^T} \le e^{-1} < 1$. So the total running time is $2^{\mathcal{O}(D \log D)} \cdot (nd)^{\mathcal{O}(1)} |\mathcal{D}| \Phi(n, d, 2D/\alpha, D)$.

[1] We could also binary search for the optimal $d_i \in \mathcal{D}$ instead, thus replacing $|\mathcal{D}|$ by $\log |\mathcal{D}|$ in the running time. However, for all choices of $\mathcal{D}$ we consider this does not make a difference.



(a) $\mathrm{cost}(z) = |z-2| + |z-3| + |z-6| + |z-8|$

(b) $\mathrm{cost}(z) = |z-2|^{1/2} + |z-3|^{1/2} + |z-6|^{1/2} + |z-8|^{1/2}$

Fig. 4. Graphs of cluster cost over different values of z: dist_1 in the left plot, $\mathrm{dist}_{1/2}$ in the right plot. The set of coordinate values is given as $y_1 = 2$, $y_2 = 3$, $y_3 = 6$, $y_4 = 8$.

The algorithm could be derandomized by the standard derandomization technique using perfect hash families [27,32]. So k-Clustering is solvable in the same deterministic time. □

4. Algorithms and complexity for distances with $p \in (0, 1]$

The main motivation for the results in this section is the study of k-Clustering with the L_1 distance, the case widely known as k-Medians. However, our main algorithmic result also extends to distances of order $p \in (0, 1)$ since in some sense they behave similarly to the L_1 distance.

4.1. FPT algorithm when parameterized by D

In this subsection, we prove Theorem 1: when $p \in (0, 1]$, k-Clustering admits an FPT algorithm with parameter D. First we state basic geometrical observations for cases $p = 1$ and $p \in (0, 1)$. Then we propose a general algorithm for Cluster Selection which relies only on these properties. Finally, we show how Theorem 10 could be applied.

The next two claims deal with the structure of optimal cluster centroids. We state and prove them in the case of weighted vectors where each vector has a positive integer weight given by a weight function w. The unweighted case is just a special case when the weight of each vector is one.

First, we show that coordinates of cluster centroids could always be selected among the values present in the input, which helps greatly in enumerating cluster centroids that may be optimal.

Claim 11. *Assume $p \in (0, 1]$, let $C = \{x_1, \dots, x_t\}$ be a cluster and $w : \{x_1, \cdots, x_t\} \to \mathbb{Z}_+$ be a weight function. There is an optimal (subject to the weighted distance $w(x_i) \cdot \mathrm{dist}_p(x_i, c)$) centroid c of C such that for each $i \in \{1, \dots, d\}$, the i-th coordinate $c[i]$ of the centroid is from the values present in the input in this coordinate, that is $c[i] \in \{x_1[i], \dots, x_t[i]\}$. Moreover, for $p = 1$ we may assume that the optimal value is a weighted median of the values present in the i-th coordinate.*

Proof. For cluster C, consider the corresponding multiset of unweighted vectors $C' = \{x_1, \dots, x_t\}$, where each vector $x \in C$ is repeated $w(x)$ times. We define $y_j = x_j[i]$ for $j \in \{1, \dots, t\}$. Assume that $y_1 \le y_2 \le \cdots \le y_t$. Let us consider an optimal cluster centroid c for C and denote $z = c[i]$. Fig. 4 shows how the cluster cost behaves with respect to z on a concrete set of values $\{y_i\}$ for $p = 1$ and $p = 1/2$.

For the formal proof, we start with the case $p = 1$. The total cost of C contributed by the i-the coordinate is

$$|y_1 - z| + |y_2 - z| + \cdots + |y_t - z|.$$

If $z \in (y_i, y_{i+1})$ for $i \in \{1, \dots, t-1\}$, then the derivative with respect to z is

$$((z - y_1) + \cdots + (z - y_i) + (y_{i+1} - z) + \cdots + (y_t - z))' = i - (t - i).$$

Analogously, when $z = y_i$ for $i \in \{1, \dots, t\}$, the derivative is $i - 1 - (t - i)$. When $z < y_1$ the derivative is $-t$, and when $z > y_t$ the derivative is t. So if t is odd, then the derivative is zero at $y_{\lceil t/2 \rceil}$, strictly negative before and strictly positive after, so $y_{\lceil t/2 \rceil}$, which is the only median, is the optimal value for z. If t is even, then the derivative is zero on $[y_{t/2}, y_{t/2+1}]$, strictly negative before and strictly positive after. So any value from $[y_{t/2}, y_{t/2+1}]$ is optimal, and we may assume that it is one of the two medians $y_{t/2}$, $y_{t/2+1}$.

Now to the case $p \in (0,1)$, the contribution of the coordinate i is

$$|y_1 - z|^p + |y_2 - z|^p + \cdots + |y_t - z|^p.$$

When z is between y_i and y_{i+1}, then the derivative of the above with respect to z is equal to

$$p \cdot \left((z - y_1)^{p-1} + \cdots + (z - y_i)^{p-1} - (y_{i+1} - z)^{p-1} - \cdots - (y_t - z)^{p-1}\right).$$

It is monotone on (y_i, y_{i+1}): when z increases, the sum decreases, as terms of the form $(z - y_j)^{p-1}$ decrease and terms of the form $(y_j - z)^{p-1}$ increase, because $p - 1 < 0$. Thus, the optimal value on this interval is achieved at one of its ends. Doing the same for all intervals, we conclude that the optimal value for z must be in $\{y_1, \dots, y_t\}$. □

In particular, by Claim 11 we may assume that the coordinates of optimal cluster centroids are integers. Then, the α-property holds with $\alpha = 1$ since at most one of the initial clusters could have distance zero to the cluster centroid, and all others have distance at least one since the cluster centroid is integral. Namely, let x be a vector in the cluster, and c be the cluster centroid, if $x \neq c$, then there is a coordinate j where x and c differ, and since they are both integral, $|x[j] - c[j]| \geq 1$, and

$$\text{dist}_p(x, c) = \sum_{i=1}^{d} |x[i] - c[i]|^p \geq |x[j] - c[j]|^p \geq 1^p = 1.$$

In what follows, the expression *half of vectors by weight* means that the total weight of the corresponding set of vectors is at least half of the total weight of C.

Claim 12. *If at least half of the vectors by weight in the cluster C have the same value z in some coordinate i, then the optimal cluster centroid is also equal to z in this coordinate.*

Proof. Let S be the weight-respecting multiset of values which vectors from C have in the i-th coordinate: $S = \{x[i] : x \in C, w(x) \text{ times}\}$. Consider the difference between selecting z and some other value z' as the i-th coordinate of the centroid:

$$\sum_{y \in S} |y - z|^p - \sum_{y \in S} |y - z'|^p \leq \sum_{y \in S, y \neq z} (|y - z|^p - |y - z'|^p - |z - z'|^p).$$

The inequality holds since at least half of the elements of S are equal to z, and so for any value $y \neq z$ there is a term $|z - z'|^p$ in $\sum_{y \in S} |y - z'|^p$ corresponding to one of the values from S equal to z. The last sum is non-positive because in every term

$$|y - z|^p \leq |y - z'|^p + |z - z'|^p,$$

as $p \in (0, 1]$. This concludes the proof. □

In order to apply Theorem 10, we need an FPT algorithm for Cluster Selection. Before obtaining it, we state some properties of hypergraphs, which we need for the algorithm. Intuitively, our algorithm reduces selecting a centroid in a k-Clustering instance to finding a subhypergraph with certain properties.

A *hypergraph* G is a set of vertices $V(G)$ and a collection of hyperedges $E(G)$, each hyperedge is a subset of $V(G)$. If G and H are hypergraphs, we say that H *appears* at $V' \subset V(G)$ as a *subhypergraph* if there is a bijection $\pi : V(H) \to V'$ with a property that for any $E \in E(H)$ there is $E' \in E(G)$ such that $\pi(E) = E' \cap V'$. Here we consider that the action of π is extended to subsets of $V(H)$ in a natural way, $\pi(E) = \{\pi(v)\}_{v \in E}$ for $E \subset V(H)$.

A *fractional edge cover* of a hypergraph H is an assignment $\psi : E(H) \to [0, 1]$ such that for every $v \in V(H)$, $\sum_{E \in E(H): v \in E} \psi(E) \geq 1$. The *fractional cover number* $\rho^*(H)$ is the minimum of $\sum_{E \in E(H)} \psi(E)$ taken over all fractional edge covers ψ.

We need the following result of Marx [26] about finding occurrences of one hypergraph in another.

Lemma 13 *([26]). Let H be a hypergraph with fractional cover number $\rho^*(H)$, and let G be a hypergraph where each hyperedge has size at most ℓ. There is an algorithm that enumerates in time $|V(H)|^{\mathcal{O}(|V(H)|)} \cdot \ell^{|V(H)|\rho^*(H)+1} \cdot |E(G)|^{\rho^*(H)+1} \cdot |V(G)|^2$ every subset $V' \subset V(G)$ where H appears in G as a subhypergraph.*

Also, the following version of the Chernoff Bound will be of use.

Proposition 14 *([33]). Let $X_1, X_2, \dots, X_n$ be independent 0-1 random variables. Denote $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$. Then for $0 < \beta \leq 1$,*

$D = 2$

v	1	2	3	4	5
x_1	0	2	1	3	2
x_2	0	**1**	1	3	**1**
x_3	**1**	2	1	3	**1**
x_4	0	2	**2**	3	2
x_5	0	2	**2**	3	**1**
c	0	2	**2**	3	2

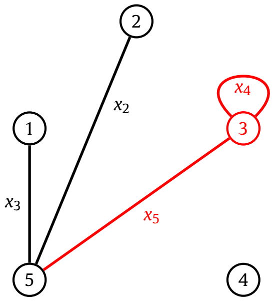


Fig. 5. An illustration of the hypergraph construction in Claim 16. On the left, the vector x_1 and other input vectors x_2, ..., x_5 are given. On the right, the corresponding hypergraph G. The solution is marked in red on both sides: on the left, the resulting cluster $\{x_1, x_4, x_5\}$ of cost 2; on the right, the corresponding to $\{x_1, x_4, x_5\}$ subhypergraph H. Note that in H the hyperedge x_5 is restricted to the only vertex 3, so its size is one. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$P[X \le (1-\beta)\mu] \le \exp(-\beta^2\mu/2),$$
$$P[X \ge (1+\beta)\mu] \le \exp(-\beta^2\mu/3).$$

We are ready to proceed with the proof that Cluster Selection with $p \in (0, 1]$ is FPT when parameterized by D.

Theorem 15. *For every $p \in (0, 1]$,* Cluster Selection *with distance* dist_p *is solvable in time* $2^{\mathcal{O}(D\log D)}(md)^{\mathcal{O}(1)}$.

Proof. First we check if any of the given vectors could be the centroid of the resulting composite cluster. When the centroid is fixed, we find the optimal solution in polynomial time by just selecting the cheapest vector with respect to this centroid from each set. If at some point we find a suitable centroid, then we return that the solution exists. If not, we may assume that the centroid is not equal to any of the given vectors. As a consequence, any vector x selected into the solution cluster contributes at least $w(x)$ to the total distance, since the centroid must be integral by Claim 11. So we may now consider only vectors of weight at most D and, moreover, the total weight of the resulting cluster is at most D.

Consider a resulting cluster C with the centroid c. There is some x_1 in C from X_1, and $\mathrm{dist}_p(x_1, c) \le D$. So if we try all possible x_1 from X_1 (there are at most m of them), any feasible centroid is at distance at most D from at least one of them. Since x_1 and c are integral, they could be different in at most D coordinates, as $\mathrm{dist}_p(x_1, c) = \sum_{i=1}^{d} |x_1[i] - c[i]|^p \le D$.

We try all possible $x_1 \in X_1$. After x_1 is fixed, we enumerate all subsets P of coordinates $\{1, \dots, d\}$ where x_1 and c could differ, we show how to do it efficiently afterwards. When the subset of coordinates P is fixed, we consider all possible centroids, which are integral, equal to x_1 in all coordinates except P, and differ from x_1 by at most $D^{1/p}$ in each of coordinates from P. If $|x_1[i^*] - c[i^*]| > D^{1/p}$ for some coordinate i^*, then $\mathrm{dist}_p(x_1, c) = \sum_{i=1}^{d} |x_1[i] - c[i]|^p \ge |x_1[i^*] - c[i^*]|^p > D$, so c can not be a centroid. With restrictions stated above, there are at most $2^{\mathcal{O}(D\log D)}$ possible centroids.

It remains to show that we could enumerate all possible coordinate subsets efficiently. We reduce this task to the task of finding a specific subhypergraph and then apply Lemma 13.

Claim 16. *There are $2^{\mathcal{O}(D\log D)}$ coordinate subsets where x_1 and an optimal cluster centroid c could differ. There exists an algorithm which enumerates all of them in time $2^{\mathcal{O}(D\log D)}(md)^{\mathcal{O}(1)}$.*

Proof. Let G be a hypergraph with $V(G) = \{1, \dots, d\}$, one vertex for each coordinate, and for each vector x in $\cup_{j=1}^{t} X_j$ we take $w(x)$ multiple hyperedges E_x which contains exactly the coordinates where x and x_1 differ. We add an edge only if there are at most D such coordinates, otherwise x can not be in the same cluster as x_1. So hyperedges in G are of size at most D. Since we consider only vectors of weight at most D, $|E(G)| \le Dm$.

For a solution, let x_j be the vector selected from the corresponding X_j, for $j \in \{1, \dots, t\}$, $C = \{x_1, \dots, x_t\}$ be the solution cluster and c be the centroid. All vectors in C are identical in all coordinates except at most D, since if there are different values in at least $D+1$ coordinates, the cost is at least $D+1$. Denote this subset of coordinates as Q, c could also differ from x_1 only at Q. Denote the subset of coordinates where c differs from x_1 as P, $P \subset Q$ and so $|P| \le D$. The solution (C, c) induces a subhypergraph H of G in the following way. Leave only hyperedges corresponding to the vectors in C, and restrict them to vertices in P. There are at most D vertices and at most D hyperedges in H, since the total weight is at most D. An example of the correspondence between input vectors and hypergraphs is given in Fig. 5.

The next claim shows that the fractional cover number of H is bounded by a constant.

Claim 17. *Each vertex in H is covered by at least half of the hyperedges of H, and $\rho^*(H) \le 2$.*

```
Cluster Selection $(X_1, \ldots, X_t, w, D)$
    Input : Sets of vectors $X_1, \ldots, X_t$, a weight function $w$, a nonnegative integer $D$
    Output: Yes or No
for vector c in the input do
    if $\sum_{i=1}^{t} \min_{x_i \in X_i} w(x_i)\,\mathrm{dist}_p(x_i, c) \le D$ then
        Yes, STOP
for $x_1 \in X_1$ do
    $G \leftarrow$ hypergraph with $V(G) = \{1, \ldots, d\}$, $E(G) = \{$positions where $x_1$ and $x$ differ : $x \in \cup_{j=1}^{t} X_j$, $w(x)$ times$\}$
    for hypergraph H with at most D vertices and at most $160 \ln D$ hyperedges do
        if each vertex of H is covered by at least 1/4 of its hyperedges then
            for place P where H appears in G as subhypergraph do
                for integer vector c which differs from $x_1$ only at P by at most $D^{1/p}$ do
                    if $\sum_{i=1}^{t} \min_{x_i \in X_i} w(x_i)\,\mathrm{dist}_p(x_i, c) \le D$ then
                        Yes, STOP
No, STOP
```

Fig. 6. Cluster Selection algorithm from Theorem 15.

Proof. Consider a vertex $p \in P$, and assume that less than half of the hyperedges cover p. It means that in the p-th coordinate the centroid c differs from x_1, but less than half of the vectors in C by weight differ from x_1 in this coordinate. This contradicts Claim 12.

So each vertex is covered by at least half of the hyperedges, and setting $\psi \equiv \frac{2}{|E(H)|}$ leads to $\rho^*(H) \le 2$. □

In order to enumerate all possible subsets of coordinates P, we try all hypergraphs H with at most D vertices and at most D hyperedges, and if each vertex is covered by at least half of the hyperedges, we find all places where H appears in G by Lemma 13. The last step is done in $2^{\mathcal{O}(D \log D)} \cdot (md)^{\mathcal{O}(1)}$ time. However, the number of possible H could be up to $2^{\Omega(D^2)}$. The following claim, which is analogous to Proposition 6.3 in [26], shows that we could consider only hypergraphs with a logarithmic number of hyperedges.

Claim 18. *If $D \ge 2$, it is possible to delete all except at most $160 \ln D$ hyperedges from H so that in the resulting hypergraph H^* each vertex is covered by at least $1/4$ of the hyperedges, and $\rho^*(H^*) \le 4$.*

Proof. Denote $s = |E(H)|$, construct a new hypergraph H^* on the same vertex set $V(H)$ by independently selecting each hyperedge of H with probability $(120 \ln D)/s$. Applying Proposition 14 with $\beta = 1/3$, probability of selecting more than $160 \ln D$ hyperedges is at most $\exp((-120 \ln D)/27) < 1/D^2$. By Claim 17, each vertex v of H is covered by at least $s/2$ hyperedges, and the expected number of hyperedges covering v in H^* is at least $60 \ln D$. By Proposition 14 with $\beta = 1/3$, the probability that v is covered by less than $40 \ln D$ hyperedges in H^* is at most $\exp(-60 \ln D/18) \le 1/D^3$. By the union bound, with probability at least $1 - 1/D^2 - D \cdot 1/D^3 > 0$ we select at most $160 \ln D$ hyperedges and each vertex is covered by at least $40 \ln D$ hyperedges. So the claim holds, and $\rho^*(H^*) \le 4$ by setting $\psi \equiv \frac{4}{|E(H^*)|}$. □

Thus, if there is a subhypergraph H in G corresponding to a solution, then there is also a subhypergraph H^* in G appearing at the same subset of $V(G)$ with at most $160 \ln D$ hyperedges and where each vertex is covered by at least $1/4$ of the hyperedges. Since we only need to enumerate possible coordinate subsets, in our algorithm we try all hypergraphs of this form and apply Lemma 13 for each of them. Since there are at most $2^{\mathcal{O}(D \log D)}$ hypergraphs with at most $160 \ln D$ hyperedges and since the fractional cover number is still bounded by a constant, the total running time is $2^{\mathcal{O}(D \log D)} \cdot (md)^{\mathcal{O}(1)}$, as desired. □

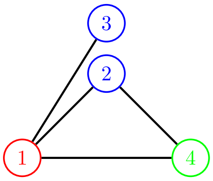
With Claim 16 proven, the proof of the theorem is complete. The pseudocode given in Fig. 6 summarizes the main steps of the algorithm. □

Combining Theorem 10 and Theorem 15, we obtain an FPT algorithm for k-Clustering. This proves Theorem 1, which we recall here.

Theorem 1. *k-Clustering with distance dist_p is solvable in time $2^{\mathcal{O}(D \log D)}(nd)^{\mathcal{O}(1)}$ for every $p \in (0, 1]$.*

Proof. We have an algorithm for Cluster Selection whose running time is specified by Theorem 15. By Claim 11, the α-property holds. The only missing part is to describe the way of producing the set $\mathcal{D}$ of all possible cluster costs which are at most D.

In the case $p = 1$ all distances are integral since optimal centroids have integral coordinates by Claim 11, and we can take $\mathcal{D} = \{0, \ldots, D\}$.



$X_{1,2}$	1	2	0
	1	3	0
$X_{2,3}$	0	2	4
$X_{1,3}$	1	0	4
$Y_{1,2}$	1	2	5
	1	3	5
$Y_{2,3}$	5	2	4
$Y_{1,3}$	1	5	4

x_{12}	1	2	0
x_{24}	0	2	4
x_{14}	1	0	4
y_{12}	1	2	5
y_{24}	5	2	4
y_{14}	1	5	4
c	1	2	4

Fig. 7. An example illustrating the reduction in Theorem 20: an input graph G with vertices colored in three colors, the sets of vectors produced by the reduction, and the resulting optimal cluster, corresponding to the clique on $\{1, 2, 4\}$.

For the general case, let $\mathcal{B} = \{a^p : a \in \{1, \dots, \lceil D^{1/p} \rceil\}\}$. Consider a cluster $C = \{x_1, \dots, x_t\}$ and the corresponding optimal cluster centroid c. For any $x_j \in C$, $\text{dist}_p(x_j, c) = \sum_{i=1}^d |x_j[i] - c[i]|^p$ is a combination of elements of $\mathcal{B}$ with nonnegative integer coefficients. This is because x_j and c are integral and the cluster cost is at most D, hence $|x_j[i] - c[i]| \le D^{1/p}$ for each $i \in \{1, \dots, d\}$. Since weights are also integral, the whole cluster cost is a combination of distances between cluster vectors and the centroid with nonnegative integer coefficients, and so also a combination of elements of $\mathcal{B}$ with nonnegative integer coefficients. This means that we can take

$$\mathcal{D} = \left\{ \sum_{b \in \mathcal{B}} a_b \cdot b : a_b \in \mathbb{Z}, a_b \ge 0, \sum_{b \in \mathcal{B}} a_b \le D \right\},$$

the sum of coefficients a_b is at most D since all elements of $\mathcal{B}$ are at least 1. The size of $\mathcal{D}$ is at most $|\mathcal{B}|^D = 2^{\mathcal{O}(D \log D)}$. □

Another widely studied version of k-Clustering is where centroids c_i could be selected only among the set of given vectors. Naturally, Theorem 1 also holds in this setting since Cluster Selection is then trivially solvable in polynomial time. As was observed in the proof of Theorem 15, if the cluster center is fixed, we can pick the cheapest vector from each of the sets given to a Cluster Selection algorithm, and there are now only polynomially many candidates for the cluster center.

Note that Claim 11 and Claim 12 do not hold in the case $1 < p < \infty$, and our algorithm relies heavily on the structure provided by them. Therefore, it does not seem that the algorithm could be extended to the case $1 < p < \infty$. Moreover, in Theorem 5 we formally prove that Cluster Selection parameterized by D is W[1]-hard for $1 < p < \infty$.

In the cases $p = 0$ and $p = \infty$ there are different obstacles for the algorithm above. In Cluster Selection for $p = 0$, even knowing the center that differs in at most k positions from an optimal one, is not enough, as any distinct value in the coordinate would incur the same cost of one. For $p = \infty$, it simply does not hold that the number of coordinates where points and the center can differ is small: any number of coordinates might differ as long as the absolute difference is at most D. To formalize this intuition we later prove Theorem 2 and Theorem 3, showing that k-Clustering parameterized by D is W[1]-hard for $p = 0$ and $p = \infty$, respectively.

4.2. W[1]-hardness of Cluster Selection *parameterized by $t + d$ for $p = 1$*

In this subsection, we restrict our attention to the $p = 1$ case. What happens when D is not bounded, but the dimension d and the number of clusters k are parameters? There is a trivial XP-algorithm in time $n^{\mathcal{O}(kd)}$, as by Claim 11 it suffices to try all possible combinations of the values present in coordinates as possible cluster centroids. There are at most n distinct values in each coordinate, so at most n^d candidates for a cluster centroid. After the cluster centroids are fixed, each vector goes to the cluster with the closest centroid. The next observation is the result of this discussion.

Observation 19. k-Clustering for the distance dist_1 is solvable in time $n^{\mathcal{O}(kd)}$.

We do not know of a lower bound for k-Clustering complementing Observation 19. However, we are able to show the hardness of Cluster Selection with respect to the dimension.

Theorem 20. Cluster Selection *with distance* dist_1 *is* W[1]*-hard when parameterized by* $t + d$.

Proof. We construct a reduction from Multicolored Clique with the input G and k. We set d to k, for each pair of colors $1 \le i < j \le k$ and each $e = \{u, v\}$ between a vertex u of color i and a vertex v of color j we add a vector x_e to the set $X_{i,j}$, such that $x_e[i] = u$, $x_e[j] = v$ and all other coordinates are set to zero, and a vector y_e to the set $Y_{i,j}$ which is the same as x_e, only coordinates other that i and j are set to $|V(G)| + 1$. We will refer to 0 and $|V(G)| + 1$ as boundary values. The sets $X_{i,j}$ and $Y_{i,j}$ are the input to Cluster Selection, so t is $2\binom{k}{2}$, and we set D to $k(|V(G)| + 1)\binom{k-1}{2}$. Intuitively, the set $X_{i,j}$ corresponds to the choice of the clique edge between i-th and j-th color, and $Y_{i,j}$ mirrors it. All vectors have weight one. An example is given in Fig. 7.

Note that in any feasible cluster, each coordinate i has exactly $2(k - 1)$ values in $[1, |V(G)|]$, one from each of the sets $X_{i,j}$ and $Y_{i,j}$ for $j \ne i$. Out of all $2(\binom{k}{2} - k + 1) = 2\binom{k-1}{2}$ other values, exactly half are zero and half are $|V(G)| + 1$. So the median is always in $[1, |V(G)|]$, and the boundary values in each column contribute exactly $(|V(G)|+1)\binom{k-1}{2}$ to the total distance.

Assume there is a colorful k-clique in G, with vertices $v_1, v_2, \dots, v_k$. We form the resulting cluster by choosing the vector corresponding to the clique's edge between its i-th and j-th vertices from $X_{i,j}$, and also from $Y_{i,j}$, for all $1 \le i < j \le k$. For this cluster, in the i-th coordinate we have all non-boundary values equal to v_i. So the median is also v_i, and the total distance is D, since non-boundary values do not contribute anything.

In the other direction, if we are able to select a cluster of cost exactly D, then all non-boundary values in each coordinate must be equal, denote this common value in the i-th coordinate as v_i. We claim that vertices $v_1, v_2, \dots, v_k$ form a colorful clique in G. Indeed, since we have $2(k-1)$ times v_i in the i-th column, then we have $(k-1)$ of them from the sets $X_{i,j}$, one from each, and in the j-th column the only non-boundary value is v_j. So v_i must have an edge to each v_j for $j \neq i$. By construction, vertices in the i-th coordinate are of color i. □

5. The L_0 distance

In this section, we consider the case $p = 0$. It is a natural measure of difference to consider since observation parameters are often incomparable, and we very well may be interested in counting only the number of different entries. From another point of view, the L_0 distance gives the k-Clustering problem a more combinatorial flavor, since the input vectors could be viewed as strings and we are interested about how close they are according to the Hamming distance. However, in comparison to a number of problems on strings, the size of the alphabet is unbounded.

First, note that there is a simple rule for finding the optimal cluster centroid for a given cluster.

Observation 21. For a given cluster C, the coordinates of the optimal cluster centroid c could be set as

$$c[i] = \text{the most frequent element of the multiset } \{x[i]\}_{x \in C},\ 1 \le i \le d,$$

breaking ties in favor of the lowest values.

By Observation 21, we may assume that optimal cluster centroids could never have values not present in the input, and in particular that they are integral.

We prove W[1]-hardness of k-Clustering with the L_0 distance by showing a reduction from Clique. The reduction also shows hardness of Cluster Selection.

Note that when d is fixed, we could apply Theorem 10 to obtain an FPT algorithm: Cluster Selection solves trivially by trying every present value in each coordinate as a value for the centroid, there are only n^d variants. The α-property holds for L_0 distance with $\alpha = 1$ since at most one initial cluster could coincide with the cluster centroid, and all others have distance at least one. We state this formally in the next observation.

Observation 22. For the distance dist_0, Cluster Selection is solvable in time $n^{\mathcal{O}(d)}$, and k-Clustering is solvable in time $2^{\mathcal{O}(D \log D)} n^{\mathcal{O}(d)}$.

Next we restate and prove Theorem 2. Note that Theorem 2 essentially complements the trivial algorithms given by Observation 22.

Theorem 2. *With distance* dist_0, *k-Clustering parameterized by $d + D$ and Cluster Selection parameterized by $d + t + D$ are W[1]-hard.*

Proof. First we show how to obtain an FPT reduction from Clique parameterized by the clique size to k-Clustering.

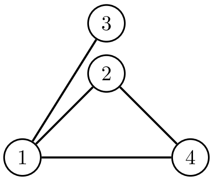
Given an instance (G, k) of Clique, for each pair of indices $\{i, j\}$, $1 \le i < j \le k$, we make $|E(G)|$ vectors in $\mathbb{Z}^k$, assume $k \ge 3$. For each $e = \{u, v\} \in E(G)$, we add a vector $x_{i,j,e}$: two coordinates are set to vertex values, $x_{i,j,e}[i] = u$, $x_{i,j,e}[j] = v$, and in all other coordinates $x_{i,j,e}$ is set to the special padding value $c_{i,j,e} = |V(G)| + (k \cdot i + j) \cdot |E(G)| + e$. In total, there are $n = \binom{k}{2}|E(G)|$ vectors and $|V(G)| + \binom{k}{2}|E(G)|$ different values, since there are $|V(G)|$ vertex values, all padding values are distinct from vertex values and from each other.

Finally, we set $k' = n - \binom{k}{2} + 1$ and $D = \binom{k}{2}(k-2)$. An example of the reduction is shown in Fig. 8.

Now we prove that the original instance has a k-clique iff the transformed instance has a k'-clustering of cost at most D.

If there is a k-clique, there is a clustering with cost D: we take one nontrivial cluster of size $\binom{k}{2}$ and all other clusters are of size 1. Let $v_1, \dots, v_k$ be the vertices of the clique, for each $\{i, j\}$, $1 \le i < j \le k$ we take $x_{i,j,\{v_i,v_j\}}$ into the cluster. The cluster centroid is $(v_1, \dots, v_k)$, each vector in the cluster has distance to the centroid of exactly $(k-2)$.

Now to the opposite direction. Assume that there is a clustering of cost at most D, and there are t composite clusters: $C_1, \dots, C_t$. In each cluster and each coordinate, by Observation 21 we may assume that we select the most frequent vertex there as the value of the centroid, since all padding values are distinct. If there are no vertex values in this cluster in this



$x_{1,2,\cdot}$	1	2	·
	1	3	·
	1	4	·
	2	4	·
$x_{1,3,\cdot}$	1	·	2
	1	·	3
	1	·	4
	2	·	4
$x_{2,3,\cdot}$	·	1	2
	·	1	3
	·	1	4
	·	2	4

$x_{1,2,12}$	1	2	·
$x_{1,3,14}$	1	·	4
$x_{2,3,24}$	·	2	4
c	1	2	4

Fig. 8. An example illustrating the reduction in Theorem 2: an input graph G, the vectors produced by the reduction (for clarity, they are separated over corresponding pairs $\{i, j\}$, and padding values are replaced by dots), and the only composite cluster in the resulting optimal clustering of cost 3, corresponding to the clique on $\{1, 2, 4\}$.

coordinate, we may assume that we select any of the occurring padding values. For a cluster C, denote the number of vertex-containing coordinates as $\beta(C)$, and the total number of vertex-valued entries which do not match with the centroid value in the corresponding coordinate as $\gamma(C)$. We could write the total cost of the clustering as

$$\sum_{i=1}^{t} (|C_i|(k-2) - (k-\beta(C_i)) + \gamma(C_i)).$$

That holds since in each cluster C_i each of the $|C_i|(k-2)$ padding values is not matched with the cluster centroid and increases the total distance by one, except for the $(k-\beta(C_i))$ vertex-free coordinates, where exactly one of the padding values is selected as the value of the centroid. Also each vertex-valued entry which is not matched with the centroid increases the total distance by one, there are $\gamma(C_i)$ of them.

There are $n - \binom{k}{2} + 1$ clusters in total, $n - \binom{k}{2} + 1 - t$ of them are simple. We may assume that in the optimal clustering there are no empty clusters, since we could always move a vector from a composite cluster to an empty one without increasing the cost. So there are $n - (n - \binom{k}{2} + 1 - t) = t + \binom{k}{2} - 1$ vectors in the composite clusters, which is equal to $\sum_{i=1}^{t} |C_i|$. We could rewrite the total cost as

$$(t + \binom{k}{2} - 1)(k-2) - tk + \sum_{i=1}^{t} (\beta(C_i) + \gamma(C_i)) = \binom{k}{2}(k-2) - (k-2) + \sum_{i=1}^{t} (\beta(C_i) - 2 + \gamma(C_i)).$$

Now we show that for any clustering the value $\sum_{i=1}^{t} (\beta(C_i) - 2 + \gamma(C_i))$ is at least $(k-2)$, and it is equal to $(k-2)$ only in the k-clique clustering. It suffices to prove the following lemma.

Lemma 23. *For any cluster C such that $2 \le |C| \le \binom{k}{2}$, $\frac{\beta(C)-2+\gamma(C)}{|C|-1} \ge \kappa$, where $\kappa = \frac{k-2}{\binom{k}{2}-1} = \frac{2}{k+1}$, and the equality holds only when C is a k-clique.*

The lemma implies

$$\sum_{i=1}^{t} (\beta(C_i) - 2 + \gamma(C_i)) = \sum_{i=1}^{t} \frac{\beta(C_i) - 2 + \gamma(C_i)}{|C_i| - 1} (|C_i| - 1) \ge \kappa \sum_{i=1}^{t} (|C_i| - 1) = \kappa \left(\binom{k}{2} - 1 \right) = k - 2,$$

and also that the equality holds only when each term is equal to κ, so each C_i is a k-clique, but then $t = 1$ since $\sum_{i=1}^{t} (|C_i| - 1) = \binom{k}{2} - 1$. So G must contain a k-clique if there is a clustering of cost at most D, and the reduction is correct. Note that none of the C_i could have size larger than $\binom{k}{2}$ since there are $n - \binom{k}{2} + 1$ clusters in total.

Proof of Lemma 23. First, we consider the case $\gamma(C) = 0$, so in each coordinate all vertex values are equal.

Claim 24. *If C is a cluster of vectors obtained by applying the reduction described in the proof of Theorem 2 to any graph H, $\gamma(C) = 0$, and $\binom{l}{2} < |C|$, then $\beta(C) \ge l + 1$.*

Proof. The proof is by induction on l. The base is $l = 1$, and each non-empty cluster contains at least one vector and so at least 2 coordinates with vertices, we assume $\binom{1}{2} = 0$.

For the general case, if there are at least l occurrences of a vertex v in a coordinate i, then there are at least $(l+1)$ coordinates with vertices. Each vector with v in the i-th coordinate has also some other vertex in some other coordinate. As in each coordinate all vertex values are equal, it could not be that two of the vectors with the value v in the i-th coordinate share the second vertex-valued coordinate, since then they would represent the same edge.

So each coordinate has at most $(l-1)$ vertex occurrences, otherwise the claim holds. Select a coordinate j which contains some vertex value u and remove the j-th coordinate and all vectors which have the value u in the j-th coordinate. That corresponds to the natural restriction C' of the cluster C to a subgraph $H-u$. The size of C' is at least $\binom{l}{2}+1-(l-1)=\binom{l-1}{2}+1$, and by induction there are at least l coordinates which contain vertex values, so the original cluster C has at least $l+1$ such coordinates, since there is also the j-th coordinate with the vertex value u. □

Now consider a cluster C with $\gamma(C)=0$. Let l be the largest value with $\binom{l}{2}+1\le|C|$, so $|C|\le\binom{l+1}{2}$. Since $|C|\le\binom{k}{2}$, $l+1\le k$. By Claim 24, $\beta(C)\ge l+1$, then

$$\frac{\beta(C)-2}{|C|-1}\ge\frac{l-1}{\binom{l+1}{2}-1}=\frac{2}{l+2}\ge\frac{2}{k+1}=\kappa,$$

and so if $l+1<k$, the inequality is strict. It is also strict if $l+1=k$ and $|C|<\binom{k}{2}$, as the denominator becomes larger in the first step. Thus the only possibility of getting exactly κ is when $|C|=\binom{k}{2}$.

But then we have exactly $k\cdot(k-1)$ vertex values across k coordinates, and each coordinate has at most $(k-1)$ vertex values by the argument in Claim 24, so each coordinate must have exactly $(k-1)$ vertex values. Since $\gamma(C)=0$, they must be all equal. Denote the common vertex value in the i-th coordinate as v_i. Since each occurrence of v_i in the i-th coordinate corresponds to an edge to a different v_j, vertices $v_1,\dots,v_k$ form a clique in G.

In the case $\gamma(C)>0$, consider a new cluster C' which is obtained from C by removing all vectors which have a vertex-valued entry not equal to the centroid value. Assume for now that $|C'|\ge2$. By the proof above, $\frac{\beta(C')-2}{|C'|-1}\ge\kappa$, since $\gamma(C')=0$. The value $\frac{\beta(C)-2+\gamma(C)}{|C|-1}$ could be obtained from $\frac{\beta(C')-2}{|C'|-1}$ by adding $\gamma(C)+(\beta(C)-\beta(C')$ to the numerator and $|C|-|C'|$ to the denominator. Removing vectors could not increase β, so $\beta(C)-\beta(C')\ge0$, and $\gamma(C)\ge|C|-|C'|$ since each of the removed vectors has at least one vertex value not equal to the centroid value. If $\frac{\beta(C')-2}{|C'|-1}\ge1$, then the new fraction is also at least 1 and so strictly greater than κ. If $|C'|\le1$, then $\frac{\beta(C)-2+\gamma(C)}{|C|-1}\ge1$ since $\beta(C)\ge2$ and $\gamma(C)\ge|C|-|C'|$. If $\frac{\beta(C')-2}{|C'|-1}<1$, then the new fraction became strictly larger, and so strictly larger than κ. In all cases, the inequality is strict when $\gamma(C)>0$. □

Now to Cluster Selection: the reduction is almost the same, only we start from Multicolored Clique, and for each pair of indices $\{i,j\}$, $1\le i<j\le k$ we obtain the set of vectors $X_{i,j}$ from edges in G starting in color i and ending in color j. The vectors are constructed in the same way as in the previous reduction. All weights are set to one. The value of D is the same, $D=\binom{k}{2}(k-2)$.

Since vectors are constructed in the same way, all statements about the cost of grouping them remain valid, in particular Lemma 23. Only now the statement of Cluster Selection already guarantees that we select exactly one cluster and exactly one vector from each $X_{i,j}$, so exactly one edge between each pair of colors. And by Lemma 23 only the proper k-clique has the optimal cost. □

Note that Cluster Selection with the L_0 distance is very similar to the known problem Consensus String With Outliers, studied e.g. in [34]. The only difference of Cluster Selection is that we have to select one point from each of the given sets, whereas in Consensus String With Outliers the goal is to select the arbitrary subset of size $(n-k)$. The construction from Theorem 2 also shows W[1]-hardness of Consensus String With Outliers with respect to $(d+D+n-k)$ in the case of unbounded alphabet.

6. The L_∞ distance

In this section, we consider the case $p=\infty$. We prove two hardness results of k-Clustering: W[1]-hardness when parameterized by D and NP-hardness in the case $k=2$.

First, we prove some useful facts about the structure of optimal cluster centroids. The one thing, in which the L_∞ distance is harder than all other distances in our consideration, is that even when the cluster is given, we can not just find the optimal cluster centroid by optimizing the value in each coordinate independently. So there seems to be no simple rule of finding the optimal cluster centroid of a given cluster. However, one could still do that in polynomial time by solving a linear program.

Claim 25. *Given a multiset C of vectors in $\mathbb{Z}^d$, there is a polynomial time algorithm to find $c\in\mathbb{R}^d$ minimizing*

$$\sum_{x\in C}\mathrm{dist}_\infty(x,c).$$

Proof. We reduce to solving a linear program, which we define next. Denote $C = \{x_1, \dots, x_n\}$, introduce variables c_1, …, c_d corresponding to coordinates of the cluster centroid and variables d_1, …, d_n, where d_i corresponds to the value $\mathrm{dist}_\infty(x_i, c)$. Consider the following linear program.

$$\sum_{i=1}^{n} d_i \to \min$$

$$x_i[j] - c_j \le d_i \quad \forall i, j : 1 \le i \le n, 1 \le j \le d$$

$$c_j - x_i[j] \le d_i \quad \forall i, j : 1 \le i \le n, 1 \le j \le d$$

Clearly, solving this linear program provides an optimal cluster centroid by the values c_1, …, c_d. □

The next claim shows that we could only consider half-integral cluster centroids.

Claim 26. *For any multiset C of vectors in $\mathbb{Z}^d$, the vector $c \in \mathbb{R}^d$ which minimizes*

$$\sum_{x \in C} \mathrm{dist}_\infty(x, c)$$

could always be chosen from $\frac{1}{2}\mathbb{Z}^d$ (coordinates are either integer or half-integer).

Proof. Assume that we have an optimal solution c which has at least one coordinate not of the form $z/2$, $z \in \mathbb{Z}$. For $a \in \mathbb{R}$ we denote $\mathrm{frac}(a) = a - \lfloor a \rfloor$, and

$$\mathrm{rem}(a) = \begin{cases} \mathrm{frac}(a), & \text{if } \mathrm{frac}(a) < 1/2 \\ 1 - \mathrm{frac}(a), & \text{otherwise} \end{cases},$$

calling this value the *remainder* of a.

We could partition all coordinates into equivalence classes by remainder of c. One could also define a partition of all vectors by the remainder of the distance to c. These two partitions are related in the following sense: if $\mathrm{dist}_\infty(x, c)$ has remainder ξ then each coordinate j where $|x[j] - c[j]| = \mathrm{dist}_\infty(x, c)$ also has remainder ξ, and vice versa. Now we take one particular remainder and show that we can shift it without losing optimality.

There are two kinds of vectors with the particular remainder ξ: call *bottom* those vectors x for which $\mathrm{frac}(\mathrm{dist}_\infty(x, c)) = \xi$, and call *top* those vectors x for which $\mathrm{frac}(\mathrm{dist}_\infty(x, c)) = 1 - \xi$. Similarly, there are also two kinds of coordinates of c, which we also call bottom and top depending of the value of $\mathrm{frac}(c[j])$.

Consider a bottom coordinate j. Increasing $c[j]$ increases $|x[j] - c[j]|$ for all bottom vectors x, and decreases $|x[j] - c[j]|$ for all top vectors x. Decreasing $c[j]$ does the opposite, as well as increasing a top coordinate. So if we take some sufficiently small value β and simultaneously increase all bottom coordinates and decrease all top coordinates by β then for all bottom vectors their distance will become larger by β, and for all top vectors — smaller by β. An if we do the opposite, the bottom vectors will cost less and the top vectors will cost more. Then, we could just take the group which has more vectors (bottom or top) and choose that action which decreases the distance for these vectors. The larger group has at least as many vectors as the smaller group, so the total distance does not increase.

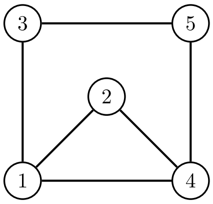
It remains to see which value of β we could take. We could safely shift until we either reach a value in $\frac{1}{2}\mathbb{Z}$ or another remainder. In any case, we reduce the number of distinct remainders by one, and so we conclude the proof by doing this inductively over the number of distinct remainders. □

By Claim 26, the α-property holds with $\alpha = 1/2$, since at most one vector could be equal to the cluster centroid, and all others have distance at least $1/2$ due to half-integrality. We can also see that when the problem is parameterized by $d + D$, it is FPT.

Claim 27. *k-Clustering with the L_∞ distance is FPT when parameterized by $d + D$.*

Proof. We use Theorem 10. We have the α-property, and for the set $\mathcal{D}$ of all possible cluster costs not exceeding D we could take all half-integral values not exceeding D by Claim 26. All that remains is to solve Cluster Selection in FPT time.

For that, we try all possible $x_1 \in X_1$, and then try each possible resulting cluster centroid c. Since $\mathrm{dist}_\infty(x_1, c) \le D$ and c is half-integral by Claim 26, we can try only vectors c of this form, and that is done in time $(2D + 1)^d$. □



	1	2	3	4	5	23	34	15	25
x_1	2	0	0	0	0	0	0	2	0
x_2	0	2	0	0	0	2	0	0	2
x_3	0	0	2	0	0	−2	2	0	0
x_4	0	0	0	2	0	0	−2	0	0
x_5	0	0	0	0	2	0	0	−2	−2

x_1	2	0	0	0	0	0	0	2	0
x_2	0	2	0	0	0	2	0	0	2
x_4	0	0	0	2	0	0	−2	0	0
c	1	1	0	1	0	1	−1	1	1

Fig. 9. An example illustrating the reduction in Theorem 3: an input graph G, the vectors produced by the reduction (for clarity, the coordinates corresponding to vertices and to non-edges are separated), and the only composite cluster in the resulting optimal clustering of cost 3, corresponding to the clique on $\{1, 2, 4\}$. Note that $\text{dist}_\infty(x_1, c) = \text{dist}_\infty(x_2, c) = \text{dist}_\infty(x_4, c) = 1$.

6.1. W[1]-*hardness when parameterized by D*

Knowing that k-Clustering with the L_∞ distance is FPT when parameterized by $d + D$, the next natural question is, is the problem FPT or W[1]-hard when parameterized only by D? We show that W[1]-hardness is the case, proving Theorem 3, which we recall here for convenience.

Theorem 3. *With distance* dist_∞*, k-*Clustering *parameterized by D and* Cluster Selection *parameterized by $t + D$ are* W[1]*-hard.*

Proof. First, we show a reduction from Clique to k-Clustering. Given a graph G and a clique size k, we construct the following instance of the clustering problem.

We set the dimension to $|V(G)| + \binom{|V(G)|}{2} - |E(G)|$. We take $|V(G)|$ vectors $\{x_i\}_{i=1}^{|V(G)|}$ corresponding to vertices. For the vertex v, first $|V(G)|$ coordinates are set to zero, except v-th coordinate, which is set to 2.

The last $\binom{|V(G)|}{2} - |E(G)|$ coordinates correspond to non-edges, vertex pairs which are not connected by an edge. For each vertex pair $\{u, v\} \notin E(G)$ in the coordinate $\{u, v\}$ we set x_u to 2, x_v to -2, the order on u, v is chosen arbitrarily, and all other vectors to zero.

Finally, we set the number of clusters to $|V(G)| - k + 1$ and the total distance to k. We show an example on how the reduction works in Fig. 9.

If there is a clique of size k in G, then we have a solution of cost k: take k vectors corresponding to the clique vertices in one cluster, and make all other clusters trivial. For the only nontrivial cluster C, we can always choose c so that $|x[j] - c[j]| \leq 1$ for any $x \in C$ and for any coordinate j. Each vertex coordinate has only 0 and 2, so setting c to 1 there suffices. As in C we have an edge between any two vertices, in any non-edge coordinate j there are either all zeros, or zeros and 2, or zeros and -2. In each of the cases there is a suitable value for c_j: 0, 1 or -1 correspondingly.

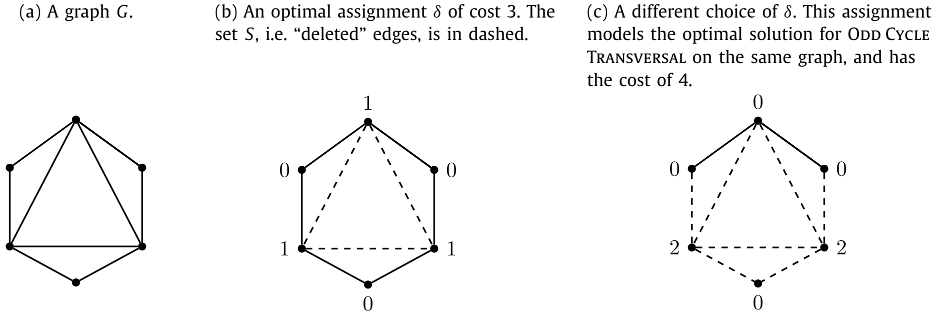
Next, we prove that any solution has cost at least k, and any solution which is not a k-clique has strictly larger cost. For that, we prove the following claim.

Claim 28. *In the instance above, the cost of any cluster C containing at least two vectors is at least $|C|$. If there is at least one non-edge in C, then the cost is at least $|C| + 1$.*

Proof. Denote the cluster centroid as c. If each vector x in C has $\text{dist}_\infty(x, c) \geq 1$, the first statement is trivial. So assume that there is a vector x^* in C such that $\text{dist}_\infty(x^*, c) = \xi < 1$. Consider the coordinate j^* which corresponds to the same vertex as the vector x^*, $x^*[j^*] = 2$, and all other vectors are zero in the coordinate j^*. As $\text{dist}_\infty(x^*, c) = \xi$, $c[j^*] \geq 2 - \xi$. Then, for any other $x \in C$, $\text{dist}_\infty(x, c) \geq 2 - \xi > 1$. The total cost of the cluster is at least $\xi + (|C| - 1)(2 - \xi) = 2 + (|C| - 2)(2 - \xi) \geq |C|$, as $2 - \xi > 1$.

Now to the second part of the claim. Assume there are only two vectors in C and they do not have an edge, there is a coordinate j^* where one is 2 and the other is -2. No matter what we choose for $c[j^*]$, the cost is at least $|2 - c[j^*]| + |-2 - c[j^*]| \geq 4$, and the statement follows. So assume that $|C| \geq 3$ and there is a coordinate j^* corresponding to a non-edge in C. One vector from C has 2 in the coordinate j^*, another -2, and all others have 0. Then there is a vector in C with distance to c of at least 2, as either $c[j^*] \geq 0$ and $|-2 - c[j^*]| \geq 2$ or $c[j^*] < 0$ and $|2 - c[j^*]| > 2$. Let us just forget about this vector and consider all other vectors in C. There are $|C| - 1 \geq 2$ of them, and by the reasoning in the proof of the first statement, their cost is at least $|C| - 1$. In this proof we considered only vertex coordinates, so the vector we forgot and the j^*-th coordinate (which is a non-edge coordinate) does not affect it. So, the total cost is at least $|C| - 1 + 2 = |C| + 1$. □

Assume that we have $l \geq 1$ nontrivial clusters of sizes $\{t_i\}_{i=1}^{l}$, nontrivial means that the size is at least two, $t_i \geq 2$ for $i \in \{1, \ldots, l\}$. By Claim 28, the total cost is at least


(a) A graph G.

(b) An optimal assignment δ of cost 3. The set S, i.e. "deleted" edges, is in dashed.

(c) A different choice of δ. This assignment models the optimal solution for Odd Cycle Transversal on the same graph, and has the cost of 4.

Fig. 10. An illustration of Half-Integral Odd Cycle Transversal.

$$\sum_{i=1}^{l} t_i = k + l - 1 \geq k,$$

as there are $|V(G)| - k + 1$ clusters in total, $|V(G)| - k + 1 - l$ trivial clusters, and the total number of vectors is $|V(G)| = \sum_{i=1}^{l} t_i + |V(G)| - k + 1 - l$, from which it follows that $\sum_{i=1}^{l} t_i = k + l - 1$. So no solution has cost less than k.

Also, if there are at least two nontrivial clusters, then $k + l - 1 \geq k + 1$. So if a solution has cost k, it must have only one nontrivial cluster, and its size must be k.

Finally, assume that the solution indeed has only one nontrivial cluster, but there is a non-edge in it. Then, as the size is k, by Claim 28 its cost is at least $k + 1$. So only a k-clique has cost k, which proves the correctness of the reduction.

Now, to Cluster Selection. We consider essentially the same reduction, only we start from Multicolored Clique. We obtain sets of vectors $X_1, \ldots, X_k$ in the same way as X in the reduction above, only vectors obtained from vertices of color j are put into X_j. The total distance parameter is also set to k. So parameters t and D of the obtained instance have the same value as the starting parameter k.

Since vectors are constructed in the same way, Claim 28 still works. And now the statement of Cluster Selection enforces that exactly one cluster of k vectors is selected. By Claim 28 it could be done with the cost k if and only if there is a colorful k-clique in the original graph. □

6.2. NP-*hardness when* $k = 2$

In this subsection we prove NP-hardness of k-Clustering with the L_∞ distance when $k = 2$. Intuitively, if we consider the previous reduction, partitioning the vectors optimally into two clusters loosely corresponds to partitioning the vertices into two sets such that there are as many as possible vertices having no edges inside their set. Which, in turn, is Odd Cycle Transversal: the problem of removing the smallest number of vertices so that the remaining graph is bipartite. However, to make everything really work, we need to consider a modified version of Odd Cycle Transversal which we call Half-Integral Odd Cycle Transversal.

Half-Integral Odd Cycle Transversal

Input: An undirected graph G, an integer t.

Task: Is there an assignment $\delta : V(G) \to \{0, 1, 2\}$, such that $\sum_{v \in V(G)} \delta(v) \leq t$ and $G - S$ is bipartite, where $S = \{\{u, v\} \in E(G) : \delta(u) + \delta(v) \geq 2\}$?

The definition of Half-Integral Odd Cycle Transversal is illustrated in Fig. 10.

First we show that Half-Integral Odd Cycle Transversal is also NP-hard by constructing a reduction from 3-SAT.

Lemma 29. *There is a polynomial-time reduction from* 3-SAT *to* Half-Integral Odd Cycle Transversal.

Proof. Given an instance of 3-SAT with n variables and m clauses, make a graph G as follows. The example of the reduction is given in Fig. 11. For each variable x_i, introduce two vertices x_i and x_i', connect them with an edge. Also introduce $2n + 1$ vertices $y_{i,j}$ connect them to both x_i and x_i'.

For each clause C_j introduce four vertices $C_{j,1}$,…,$C_{j,4}$. Consider following seven vertices: $C_{j,1}, \ldots, C_{j,4}$, and three variable vertices which are present in C_j: if $x_i \in C_j$ then we consider the vertex x_i, and if $\neg x_i \in C_j$ then we consider the vertex x_i'. Connect all these seven vertices in a cycle such that each variable vertex is adjacent to two clause vertices. Finally, set t to $2n$.

First, assume there is a satisfying assignment. Consider the following $\delta : V(G) \to \{0, 1, 2\}$: if x_i is true, $\delta(x_i) = 2$, otherwise $\delta(x_i') = 2$, on all other vertices $\delta \equiv 0$. Clearly, $\sum_{v \in V(G)} \delta(v) = 2n$.

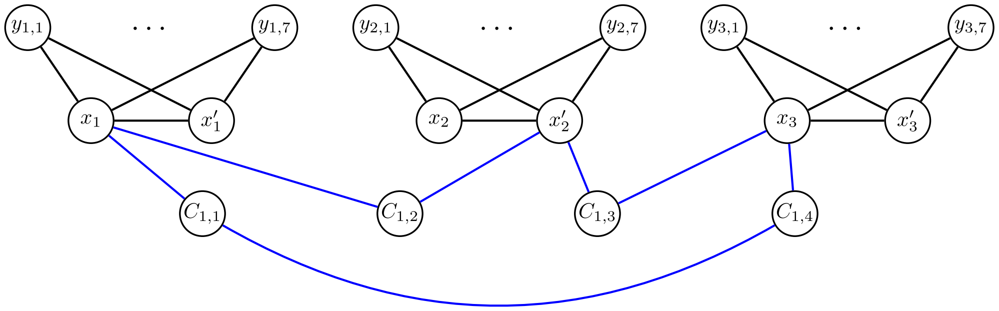


Fig. 11. A graph obtained from the 3CNF-formula $(x_1 \vee \neg x_2 \vee x_3)$ by the reduction from Lemma 29. A 7-cycle corresponding to the only clause of the formula is highlighted in blue.

Since δ does not take value 1, deleting edges $\{u, v\}$ with $\delta(u) + \delta(v) \geq 2$ is equivalent to deleting vertices on which δ is 2. From each vertex gadget we deleted either x_i or x'_i, so the remaining part is a star with leaves $y_{i,j}$ and center x_i or x'_i. Since the assignment we started from is satisfying, from each clause cycle we deleted at least one vertex. So each cycle present in G lost at least one vertex, and what remains is bipartite.

Now assume there is a solution δ to the Half-Integral Odd Cycle Transversal instance. We claim that $\delta(x_i) + \delta(x'_i) \geq 2$ for each variable x_i. Consider a 2-coloring of $G - S$: either x_i and x'_i have the same color or not. In the former case, $\delta(x_i) + \delta(x'_i) \geq 2$ since the edge $\{x_i, x'_i\}$ must be removed.

If x_i and x'_i have different colors, assume that $\delta(x_i) \leq 1$ and $\delta(x'_i) \leq 1$. Then, each of the $2n + 1$ vertices $y_{i,j}$ takes one of the two colors, and so has an incident edge to x_i or x'_i which needs to be deleted. But then, $\delta(y_{i,j}) \geq 1$ for each j, and the total cost on these vertices is already $2n + 1$. Then either $\delta(x_i) = 2$ or $\delta(x'_i) = 2$.

So we have n variables and δ is at least 2 on each pair of variable vertices, and in total δ is at most $2n$. Then δ has to be exactly 2 on each variable pair, and zero on all other vertices. Now we claim that on each clause cycle there is a variable vertex v with $\delta(v) = 2$. If not, then none of the cycle edges gets deleted, as δ is equal to zero on clause vertices. But then the remaining graph could not be bipartite, since it contains an odd cycle.

To get a satisfying assignment, set x_i to true if $\delta(x_i) = 2$, or to false otherwise. In particular, if $\delta(x'_i) = 2$, x_i is set to false, since $\delta(x_1) + \delta(x'_1) = 2$. Each clause is satisfied since each clause cycle contains a variable vertex on which δ is equal to 2. □

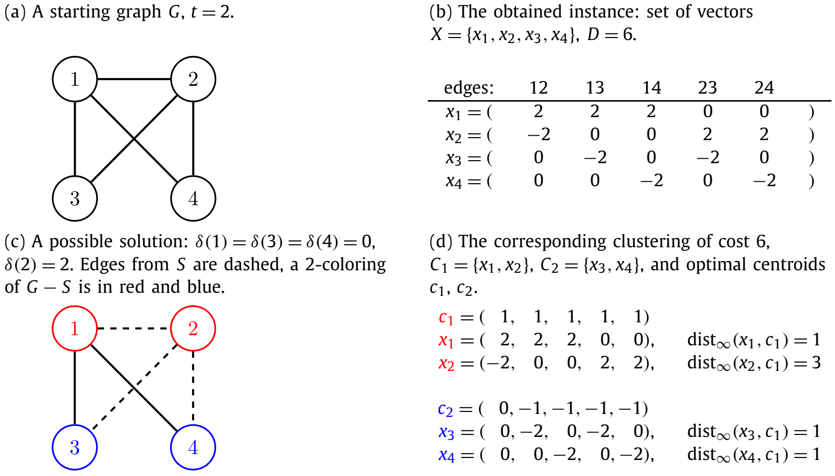
Now we prove NP-hardness of k-Clustering with $p = \infty$ and $k = 2$ by constructing a reduction from Half-Integral Odd Cycle Transversal.

Theorem 30. *k-Clustering with distance* dist_∞ *is* NP*–hard when* $k = 2$.

Proof. Consider an instance (G, t) of Half-Integral Odd Cycle Transversal, if $t \geq |V(G)|$, we have a yes-instance since $\delta \equiv 1$ deletes all edges from the graph, so we may assume $t < |V(G)|$. Remove all isolated vertices in G and add $t + 5$ isolated edges to G, it clearly does not change the type of the instance. The number of clusters k is 2, set the dimension d to $|E(G)|$, each coordinate corresponds to an edge. For each vertex $v \in V(G)$ add a vector x_v to X with all coordinates set to zero. Then, for each edge $\{u, v\} \in E(G)$ set $x_u[u, v]$ to 2 and $x_v[u, v]$ to -2, the order on u, v is chosen arbitrarily. Finally, set D to $|V(G)| + t$. An example is given in Fig. 12, additional isolated edges are dropped out for clarity.

If (G, t) is a yes-instance of Half-Integral Odd Cycle Transversal, consider the solution δ. Split vectors into clusters according to any proper 2-coloring of $G - S$. Now we show the way to select cluster centroids so that each vertex v has distance at most $1 + \delta(v)$ to the corresponding centroid. We consider separately each of two clusters and each coordinate, indexed by an edge $\{u, v\} \in E(G)$. For a cluster C, there are three cases on how u and v are present in the cluster, for each of them we assign a particular value to the cluster centroid c in the coordinate $\{u, v\}$.

- If u and v are both not in C, for vectors in C all entries in the coordinate $\{u, v\}$ are zero, and we set $c[u, v]$ also to zero. Each vector is at distance zero to the centroid in this coordinate.
- If only one of u and v are in C, for vectors in C all entries in the corresponding coordinate are zero, except one entry corresponding to the edge's endpoint belonging to C, which is either 2 or -2. Set $c[u, v]$ to 1 or -1, correspondingly, then each vector is at distance 1 in this coordinate.
- If both u and v are in C, w.l.o.g $x_u[u, v]$ is 2 and $x_v[u, v]$ is -2, and all other points are zero. It must hold that $\delta(u) + \delta(v) \geq 2$, either $\delta(u) = \delta(v) = 1$ or w.l.o.g $\delta(u) = 2$ and $\delta(v) = 0$. In the former case, set $c[u, v]$ to zero, then all vectors have distance zero, x_u and x_v have distance 2 in this coordinate. In the latter case, set $c[u, v]$ to -1, then u is at distance 3, and all other vectors, including v, are at distance 1.



(a) A starting graph G, $t = 2$.

(b) The obtained instance: set of vectors $X = \{x_1, x_2, x_3, x_4\}$, $D = 6$.

edges:	12	13	14	23	24	
$x_1 = ($	2	2	2	0	0	)
$x_2 = ($	−2	0	0	2	2	)
$x_3 = ($	0	−2	0	−2	0	)
$x_4 = ($	0	0	−2	0	−2	)

(c) A possible solution: $\delta(1) = \delta(3) = \delta(4) = 0$, $\delta(2) = 2$. Edges from S are dashed, a 2-coloring of $G - S$ is in red and blue.

(d) The corresponding clustering of cost 6, $C_1 = \{x_1, x_2\}$, $C_2 = \{x_3, x_4\}$, and optimal centroids c_1, c_2.

$$\begin{aligned} c_1 &= (\ 1,\ 1,\ 1,\ 1,\ 1) \\ x_1 &= (\ 2,\ 2,\ 2,\ 0,\ 0), && \text{dist}_\infty(x_1, c_1) = 1 \\ x_2 &= (-2,\ 0,\ 0,\ 2,\ 2), && \text{dist}_\infty(x_2, c_1) = 3 \end{aligned}$$

$$\begin{aligned} c_2 &= (\ 0, -1, -1, -1, -1) \\ x_3 &= (\ 0, -2,\ 0, -2,\ 0), && \text{dist}_\infty(x_3, c_1) = 1 \\ x_4 &= (\ 0,\ 0, -2,\ 0, -2), && \text{dist}_\infty(x_4, c_1) = 1 \end{aligned}$$

Fig. 12. An illustration of the reduction from Theorem 30.

For any $v \in V(G)$, since it holds for all coordinates that distance from x_v to the corresponding cluster centroid is at most $1 + \delta(v)$, then the L_∞ distance is also at most $1 + \delta(v)$, and the total cost of the clustering defined above is at most

$$\sum_{v \in V(G)} 1 + \delta(v) = |V(G)| + t.$$

In the other direction, assume there is a clustering C_1, C_2 with centroids c_1, c_2 such that the total cost is at most D. By Claim 26 we may assume that centroids are integral, and for any vector the distance to the nearest centroid is also an integer. We also may assume that centroids are between -2 and 2 in each coordinate since all the input vectors have entries in this range, and so we could move the centroids to the same range without increasing distances.

So, each vector has distance in $\{0, 1, 2, 3, 4\}$ to the closest centroid. We claim that it could not be that a vector x_v has distance zero: in this case w.l.o.g $x_v = c_1$, and so c_1 is equal to 2 or -2 in some coordinate, since each vertex has at least one incident edge. But then each vector in C_1 has distance at least 2 to c_1. And since at most two vectors could be equal to the centroids, each of the remaining $|V(G)| - 2$ vectors has distance at least 1. Consider $t + 5$ isolated edges, at least $t + 3$ of them do not have any endpoint equal to one of c_1 and c_2. For these edges, the total distance of their endpoints is at least 3: either their endpoints are in different clusters, and so the endpoint in C_1 costs at least 2, or both endpoints are in the same cluster, and in total they cost 4 since there are simultaneously values 2 and -2 in the coordinate corresponding to this edge. So each of the $t + 3$ edges increases the cost by additional 1, and the total cost is at least $|V(G)| - 2 + t + 3 > |V(G)| + t$.

Since each vector has distance at least 1, we may assume that the centroids are in $\{-1, 0, 1\}^d$. If we have 2 (or -2) we could change it to 1 (or -1), all vectors which could become farther from the centroid have 2 in this coordinate. But then the distance for these vectors is still at most 1. We also may assume that distances are in $\{1, 2, 3\}$, since distance 4 could be only from 2 to -2.

We claim that if we set $\delta(v) := \min_{i=1}^{2} \text{dist}_\infty(x_v, c_i)$, δ is a solution to Half-Integral Odd Cycle Transversal. Remove all edges $\{u, v\}$ with $\delta(u) + \delta(v) \geq 2$, and consider 2-coloring of G induced by the partition $\{C_1, C_2\}$. Assume that we have an edge $\{u, v\}$ such that $\delta(u) + \delta(v) \leq 1$ and u and v are in the same cluster (w.l.o.g C_1). Then we have a coordinate $\{u, v\}$ such that w.l.o.g $x_u[u, v] = 2$ and $x_v[u, v] = -2$, but $\text{dist}_\infty(x_u, c_1) + \text{dist}_\infty(x_v, c_1) \leq 3$ due to $\delta(u) + \delta(v) \leq 1$ and so $|x_u[u, v] - c_1[u, v]| + |x_v[u, v] - c_1[u, v]| \leq 3$, which is a contradiction. So (G, t) is also a yes-instance. □

Note that the reduction from the proof of Theorem 30 also implements k-Coloring, if we set k to the number of colors and D to $|V(G)|$, since with such a small budget we can not allow any same-colored neighbors in the optimal clustering. However, k-Coloring is only known to be NP-hard for $k \geq 3$ colors. Thus, in Theorem 30 we show a reduction from Half-Integral Odd Cycle Transversal to show the hardness of k-Clustering even for two clusters.

7. The case $p \in (1, \infty)$

In this section we consider the case $p \in (1, \infty)$, with the particular emphasis on the most commonly used case $p = 2$. With the L_2 distance, the k-Clustering problem is widely studied under the name k-Means.

7.1. FPT *when parameterized by* $d + D$ *for* $p = 2$

When we consider both d and D as the parameters, Cluster Selection in the L_2 distance becomes FPT, and so k-Clustering is also FPT by Theorem 10.

Note that in any composite cluster, each vector except at most one is at distance at least $1/4$ from the centroid, so the α-property holds with $\alpha = 1/4$. Consider two different vectors, they have different values in some coordinate, and in this coordinate at least one of them is at distance at least $(1/2)^2 = 1/4$ from the centroid.

Now we prove Theorem 4, which we restate here.

Theorem 4. *k-Clustering and Cluster Selection with distance* dist_2 *are* FPT *when parameterized by $d + D$.*

Proof. We start with the proof that Cluster Selection is FPT. Distance dist_2 satisfies the α-property. Hence if $t > 4D + 1$ then any composite cluster costs more than D and the instance is clearly a no-instance. So we may assume that $t \leq 4D + 1$.

We claim that there are at most $4mtD$ possible total weights of the resulting composite cluster. First, in the resulting cluster there could be at most one vector with weight strictly larger than $4D$. Otherwise, let us consider two such vectors and the coordinate in which they differ. No matter which value the centroid has there, it is at distance of at least $1/2$ from at least one of the vectors, so the total cost is larger than $4D(1/2)^2 \geq D$. So there are at most m possibilities for the largest weight, and all of the other $(t-1)$ weights are at most $4D$.

We fix the total resulting cluster weight W, the vector in the resulting cluster with the largest weight $x_{j^*} \in X_{j^*}$, and the coordinate i. Since the centroid c is the mean of the vectors in the resulting cluster, $c[i]$ is of form $\frac{y}{W}$, where $y \in \mathbb{Z}$. We claim that the distance from y to $W \cdot x_{j^*}[i]$ is bounded by a function of D, and so each possible y could be enumerated in FPT time. Moreover, all possible centroids could also be enumerated in FPT time since d is a parameter.

Let $\{x_1, \dots, x_t\}$ be the resulting cluster, $x_j \in X_j$ for all $j \in \{1, \dots, t\}$. The difference between $c[i]$ and $x_{j^*}[i]$ could be written as

$$x_{j^*}[i] - c[i] = x_{j^*}[i] - \sum_{j=1}^{t} \frac{w(x_j)x_j[i]}{W} = \frac{\sum_{j=1}^{t} w(x_j)(x_{j^*}[i] - x_j[i])}{W}.$$

The absolute value of the numerator is $\mathcal{O}(D^3)$ since $t = \mathcal{O}(D)$, $w(x_{j^*})$ gets multiplied by zero, and all other weights are at most $4D$. Also, for any $j \in \{1, \dots, t\}$, $|x_{j^*}[i] - x_j[i]| \leq 4D$, since

$$4D \geq 4\left((x_{j^*}[i] - c[i])^2 + (x_j[i] - c[i])^2\right) \geq (x_{j^*}[i] - x_j[i])^2 \geq |x_{j^*}[i] - x_j[i]|.$$

The total running time is at most

$$4mtd \cdot m \cdot \mathcal{O}(D^3)^d \cdot m,$$

since we try all possible cluster weights, all possible x_{j^*} out of the input vectors, then all possible centroids which differ from x_{j^*} by $\mathcal{O}(D^3)$ in each coordinate. And then for each centroid we check whether the optimal cluster for it has cost at most D by selecting the best $x_j \in X_j$ for each $j \in \{1, \dots, t\}$. This concludes the proof that Cluster Selection is FPT when parameterized by $d + D$.

Now we proceed with the proof that k-Clustering is FPT parameterized by $d + D$. For that we employ Theorem 10. We already have the α-property and FPT algorithm for Cluster Selection. Hence the only thing left is to enumerate the set $\mathcal{D}$ of all possible optimal cluster costs not exceeding D.

Since there are n vectors in total, each cluster contains from 1 to n vectors. For each possible cluster size s the centroid is of the form $\frac{y}{s}$, where $y \in \mathbb{Z}$. Since input vectors have integer coordinates, the cost of any cluster of size s is of form $\frac{z}{s^2}$, where $z \in \mathbb{Z}$. And since the cost is at most D, $z \in \{0, \dots, Ds^2\}$. We enumerate all possible cluster sizes in $\{1, \dots, n\}$, and for each cluster size s all possible cluster costs in $\{0/s^2, \dots, Ds^2/s^2\}$. In this way we obtain $\mathcal{D}$, and $|\mathcal{D}| = \mathcal{O}(Dn^3)$. □

7.2. W[1]*-hardness when parameterized by $t + D$*

In our setting, k-Clustering for $p = 2$ seems to be harder than for $p = 1$, since we do not have the nice property that if many vectors have the same value in some coordinate then the centroid must also have this value. On the contrary, even if only one vector diverges from the rest, the optimal centroid also diverges. So the approach with enumerating nontrivial coordinate sets, which we successfully used in the $p \in (0, 1]$ case, is not likely to work.

We are able to prove that Cluster Selection for $p \in (1, \infty)$ is W[1]-hard parameterized by $t + D$. It remains open whether k-Clustering for $p \in (1, \infty)$ or specifically for $p = 2$ is W[1]-hard or not, but our result shows that at least the approach we used to obtain an FPT algorithm in the $p \in (0, 1]$ case would not yield an FPT algorithm for $p \in (1, \infty)$.

First we state and prove two technical claims about the geometrical properties of clustering zero-one valued vectors in the $p \in (1, \infty)$ case.

Claim 31. *If we have a cluster of size $a + b$ where a vectors have zero and b vectors have one in the coordinate i, then the optimal centroid value in this coordinate is equal to*

$$\frac{b^{\frac{1}{p-1}}}{a^{\frac{1}{p-1}}+b^{\frac{1}{p-1}}},$$

and the coordinate i contributes

$$\frac{ab}{\left(a^{\frac{1}{p-1}}+b^{\frac{1}{p-1}}\right)^{p-1}},$$

to the total cost.

Proof. Assume that the centroid value in the coordinate i is equal to c, then the cost is

$$ac^p+b(1-c)^p.$$

It is easy to see that $c<0$ is worse than $c=0$, and similarly $c>1$ is worse than $c=1$, so we could restrict c to $[0,1]$. The derivative with respect to c is

$$p(ac^{p-1}-b(1-c)^{p-1}),$$

as $p>1$, the derivative is zero if and only if

$$\begin{aligned}&ac^{p-1}=b(1-c)^{p-1}\\&\left(\frac{c}{1-c}\right)^{p-1}=\frac{b}{a}\\&\frac{c}{1-c}=\left(\frac{b}{a}\right)^{\frac{1}{p-1}}\\&c=\frac{1}{1+\left(\frac{a}{b}\right)^{\frac{1}{p-1}}}=\frac{b^{\frac{1}{p-1}}}{a^{\frac{1}{p-1}}+b^{\frac{1}{p-1}}}.\end{aligned}$$

The derivative increases monotonically: when we increase c, c^{p-1} increases and $(1-c)^{p-1}$ decreases as $p-1>0$. So the optimal value must be at its unique root defined by the expression above. Thus, the optimal cost is equal to

$$a\frac{b^{\frac{p}{p-1}}}{\left(a^{\frac{1}{p-1}}+b^{\frac{1}{p-1}}\right)^p}+b\frac{a^{\frac{p}{p-1}}}{\left(a^{\frac{1}{p-1}}+b^{\frac{1}{p-1}}\right)^p}=\frac{ab}{\left(a^{\frac{1}{p-1}}+b^{\frac{1}{p-1}}\right)^{p-1}},$$

finishing the proof. □

Now we prove that it is optimal to have as many ones in the same coordinate as possible. For that, we calculate how much each one adds to the total cost depending on how many ones are there in a coordinate.

Claim 32. *Consider a cluster of s zero-one valued vectors, denote as $f(b)$ the contribution of a coordinate in which there are b ones and $s-b$ zeros. The function $f(b)/b$ is strictly decreasing for $0<b<s$.*

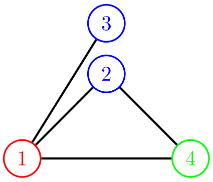
Proof. Denote the number of zeros in the coordinate as $a:=s-b$. By Claim 31, the contribution of the coordinate per each one is

$$\frac{f(b)}{b}=\frac{ab}{\left(a^{\frac{1}{p-1}}+b^{\frac{1}{p-1}}\right)^{p-1}}\cdot\frac{1}{b}=\frac{a/s}{\left((a/s)^{\frac{1}{p-1}}+(1-a/s)^{\frac{1}{p-1}}\right)^{p-1}}.$$

Let us denote $x=a/s$, $0<x<1$, the derivative of the above with respect to x is equal to

$$\frac{d}{dx}\left(\frac{x}{\left(x^{\frac{1}{p-1}}+(1-x)^{\frac{1}{p-1}}\right)^{p-1}}\right)=\left(x^{\frac{1}{p-1}}+(1-x)^{\frac{1}{p-1}}\right)^{-(p-2)}\cdot\left((1-x)^{\frac{1}{p-1}}+x(1-x)^{\frac{1}{p-1}-1}\right),$$

which is strictly positive for $0<x<1$, hence proving the claim. □



	1	2	3	4
$X_{1,2}$	1	1	0	0
	1	0	1	0
$X_{2,3}$	0	1	0	1
$X_{1,3}$	1	0	0	1

$\{1,2\}$	1	1	0	0
$\{2,4\}$	0	1	0	1
$\{1,4\}$	1	0	0	1
c	$\frac{2}{3}$	$\frac{2}{3}$	0	$\frac{2}{3}$

Fig. 13. An example illustrating the reduction in Theorem 5: an input graph G colored in three colors, the vector sets produced by the reduction, and the resulting optimal cluster of cost 2, corresponding to the clique on $\{1, 2, 4\}$. Note that in the resulting cluster, each non-zero coordinate has the maximal number of ones, $(k-1)$.

Now we are ready to prove the hardness result, which was stated in the introduction as Theorem 5. We recall the statement here.

Theorem 5. Cluster Selection *with distance* dist_p *is* W[1]*-hard for every* $p \in (1,\infty)$ *when parameterized by* $t + D$.

Proof. We construct a reduction from Multicolored Clique. Given a graph G and a clique size k, we construct the following instance of Cluster Selection.

We set t to $\binom{k}{2}$, each input set of vectors represents a choice of an edge of the clique between two particular colors, so we number them by unordered pairs of indices from 1 to k. We set the dimension d to $|V(G)|$, coordinates are numbered by vertices.

The set $X_{i,j}$ consists of the following vectors: for each edge $\{u, v\} \in E(G)$ between a vertex u of color i and vertex v of color j, we add a vector with 1 in the coordinate u and 1 in the coordinate v, all other coordinates are set to zero. All vectors have weight one. Finally, we set

$$D = k \cdot \frac{(k-1)\binom{k-1}{2}}{\left((k-1)^{\frac{1}{p-1}} + \binom{k-1}{2}^{\frac{1}{p-1}}\right)^{p-1}}.$$

In Fig. 13, we show the intuition behind the reduction by considering a simple example.

If there is a colorful k-clique in G then we construct a solution to our instance of Cluster Selection. Assume the clique is formed by vertices $v_1, v_2, \ldots, v_k$, for each $i \in \{1, \cdots, l\}$ vertex v_i is of color i. From each $X_{i,j}$ choose the vector corresponding to the edge $\{v_i, v_j\} \in E(G)$. Among the chosen vectors, in every coordinate of the form v_i there are $(k-1)$ ones from edges to v_i and $\binom{k}{2} - (k-1) = \binom{k-1}{2}$ zeros. All other coordinates are zeros in the chosen vectors, so they do not contribute anything to the total distance. By Claim 31, the total distance is

$$k \cdot \frac{(k-1)\binom{k-1}{2}}{\left((k-1)^{\frac{1}{p-1}} + \binom{k-1}{2}^{\frac{1}{p-1}}\right)^{p-1}} = D.$$

In the other direction, we prove that only the solution described above could have the cost D, all others have strictly larger cost. First notice that in any resulting cluster there are at most $(k-1)$ ones in each coordinate, since for any vertex $v \in V(G)$, if we denote its color by i, only vectors from $(k-1)$ sets of the form $X_{i,j}$ ($j \in \{1, \ldots, k\} \setminus \{i\}$) have ones in the coordinate v, and we take one vector from each set by the definition of Cluster Selection.

Each vector has exactly two ones, so in any resulting cluster there are $2 \cdot \binom{k}{2}$ ones in total. By Claim 32, any resulting cluster which does not have $(k-1)$ ones in k coordinates has strictly larger cost, since only coordinates with exactly $(k-1)$ ones have the optimal cost per each one.

So, if the resulting cluster has the cost D, then there are k coordinates such that in each of them exactly $(k-1)$ of the chosen vectors have one. We show that in this case the original instance of Clique has a k-clique. For any color $i \in \{1, \ldots, k\}$ there are at most $(k-1)$ ones in all coordinates indexed by vertices of color i in the resulting cluster. So all of these ones are in the same coordinate v_i for some v_i. We claim that the vertices $v_1, \ldots, v_k$ form a clique. Consider vertices v_i and v_j, we have taken some vector from $X_{i,j}$, and this vector must have added a one to the coordinates v_i and v_j, then by construction the edge $\{v_i, v_j\}$ is in $E(G)$. □

8. Conclusion and open problems

In this paper, we presented an FPT algorithm for k-Clustering with $p \in (0, 1]$ parameterized by D. However, for the case $p \in (1, \infty)$ we were able only to show the W[1]-hardness of Cluster Selection. While intractability of Cluster Selection does not exclude that k-Clustering could be FPT with $p \in (1, \infty)$, it indicates that the proof of this (if it is true at all) would require an approach completely different from ours. Thus an interesting and very concrete open question concerns the parameterized complexity of k-Clustering with $p \in (1, \infty)$ and parameter D.

Another open question is about the fine-grained complexity of k-Clustering when parameterized by $k+d$. For several distances, we know XP-algorithms: an $\mathcal{O}(n^{dk+1})$ algorithm by Inaba et al. [22] for $p=2$, as well as trivial algorithms for $p \in [0,1]$. For the case when the possible cluster centroids are given in the input, the matching lower bound is shown in [23]. However, we are not aware of a lower bound complementing the algorithmic results in the case when any point in Euclidean space can serve as a centroid.

Finally, let us note that our W[1]-hardness reductions could be easily adapted to obtain ETH-hardness results. Our reductions are from Clique and, assuming ETH, there is no $n^{o(k)}$ algorithm for Clique. In most of our results, the ETH lower bounds derived from our reductions, can be complemented by matching upper bounds through a trivial algorithm for Cluster Selection in time $n^{\mathcal{O}(d)}$ or $n^{\mathcal{O}(t)}$ and, consequently, an algorithm for k-Clustering obtained by Theorem 10. However, the reduction in Theorem 5 excludes only a $(nd)^{o(t^{1/2}+D^{1/2})}$ algorithm for Cluster Selection with $p \in (1,\infty)$ under ETH. Both the trivial algorithm in time $n^{\mathcal{O}(t)}$ and the algorithm from Theorem 4 in time $D^{\mathcal{O}(d)}$ (which could also be turned into a $d^{\mathcal{O}(D)}$-time algorithm) fail to match this lower bound. So, another open question is, whether there exists a better reduction or a subexponential algorithm could be obtained in this case.

CRediT authorship contribution statement

Fedor V. Fomin: Investigation, Supervision. **Petr A. Golovach:** Investigation. **Kirill Simonov:** Investigation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.

[2] S.P. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–136, https://doi.org/10.1109/TIT.1982.1056489.

[3] M.R. Ackermann, J. Blömer, C. Sohler, Clustering for metric and nonmetric distance measures, ACM Trans. Algorithms 6 (4) (2010) 59:1–59:26, https://doi.org/10.1145/1824777.1824779.

[4] P.K. Agarwal, S. Har-Peled, K.R. Varadarajan, Approximating extent measures of points, J. ACM 51 (4) (2004) 606–635, https://doi.org/10.1145/1008731.1008736.

[5] M. Badoiu, S. Har-Peled, P. Indyk, Approximate clustering via core-sets, in: Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC), ACM, 2002, pp. 250–257.

[6] C. Boutsidis, A. Zouzias, M.W. Mahoney, P. Drineas, Randomized dimensionality reduction for k-means clustering, IEEE Trans. Inf. Theory 61 (2) (2015) 1045–1062, https://doi.org/10.1109/TIT.2014.2375327.

[7] M.B. Cohen, S. Elder, C. Musco, C. Musco, M. Persu, Dimensionality reduction for k-means clustering and low rank approximation, in: Proceedings of the 47tg Annual ACM Symposium on Theory of Computing (STOC), ACM, 2015, pp. 163–172.

[8] D. Feldman, M. Langberg, A unified framework for approximating and clustering data, in: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC), ACM, 2011, pp. 569–578.

[9] D. Feldman, M. Schmidt, C. Sohler, Turning big data into tiny data: constant-size coresets for k-means, PCA and projective clustering, in: Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2013, pp. 1434–1453.

[10] S. Har-Peled, S. Mazumdar, On coresets for k-means and k-median clustering, in: Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC), ACM, 2004, pp. 291–300.

[11] A. Kumar, Y. Sabharwal, S. Sen, Linear-time approximation schemes for clustering problems in any dimensions, J. ACM 57 (2) (2010) 5:1–5:32, https://doi.org/10.1145/1667053.1667054.

[12] W.F. de la Vega, M. Karpinski, C. Kenyon, Y. Rabani, Approximation schemes for clustering problems, in: Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC), ACM, 2003, pp. 50–58.

[13] S.G. Kolliopoulos, S. Rao, A nearly linear-time approximation scheme for the Euclidean k-median problem, SIAM J. Comput. 37 (3) (2007) 757–782, https://doi.org/10.1137/S0097539702404055.

[14] V. Cohen-Addad, A fast approximation scheme for low-dimensional k-means, in: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2018, pp. 430–440, http://dl.acm.org/citation.cfm?id=3174304.3175298.

[15] C. Sohler, D.P. Woodruff, Strong coresets for k-median and subspace approximation: goodbye dimension, in: Proceedings of the 59th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2018, pp. 802–813.

[16] A. Anagnostopoulos, A. Dasgupta, R. Kumar, A constant-factor approximation algorithm for co-clustering, Theory Comput. 8 (26) (2012) 597–622, https://doi.org/10.4086/toc.2012.v008a026, http://www.theoryofcomputing.org/articles/v008a026.

[17] L. Bulteau, V. Froese, S. Hartung, R. Niedermeier, Co-clustering under the maximum norm, Algorithms 9 (1) (2016), https://doi.org/10.3390/a9010017, https://www.mdpi.com/1999-4893/9/1/17.

[18] D. Aloise, A. Deshpande, P. Hansen, P. Popat, NP-hardness of Euclidean sum-of-squares clustering, Mach. Learn. 75 (2) (2009) 245–248, https://doi.org/10.1007/s10994-009-5103-0.

[19] U. Feige, NP-hardness of hypercube 2-segmentation, CoRR, arXiv:1411.0821 [abs], 2014.

[20] N. Megiddo, K. Supowit, On the complexity of some common geometric location problems, SIAM J. Comput. 13 (1) (1984) 182–196, https://doi.org/10.1137/0213014.

[21] M. Mahajan, P. Nimbhorkar, K. Varadarajan, The planar k-means problem is NP-hard, in: Proceedings of the 3rd International Workshop on Algorithms and Computation (WALCOM), in: Lecture Notes in Comput. Sci., Springer, 2009, pp. 274–285.

[22] M. Inaba, N. Katoh, H. Imai, Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering, in: Proceedings of the 10th Annual Symposium on Computational Geometry (SoCG), ACM, 1994, pp. 332–339.

[23] V. Cohen-Addad, A. de Mesmay, E. Rotenberg, A. Roytman, The bane of low-dimensionality clustering, in: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2018, pp. 441–456, http://dl.acm.org/citation.cfm?id=3174304.3175300.

[24] F.V. Fomin, P.A. Golovach, F. Panolan, Parameterized low-rank binary matrix approximation, in: Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP), in: LIPIcs, vol. 107, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 53:1–53:16.

[25] C. Boucher, C. Lo, D. Lokshantov, Consensus patterns (probably) has no eptas, in: N. Bansal, I. Finocchi (Eds.), Algorithms - ESA 2015, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 239–250.

[26] D. Marx, Closest substring problems with small distances, SIAM J. Comput. 38 (4) (2008) 1382–1410, https://doi.org/10.1137/060673898.

[27] N. Alon, R. Yuster, U. Zwick, Color-coding, J. ACM 42 (4) (1995) 844–856, https://doi.org/10.1145/210332.210337.

[28] C. Crespelle, P.G. Drange, F.V. Fomin, P.A. Golovach, A survey of parameterized algorithms and the complexity of edge modification, arXiv:2001.06867, 2020.

[29] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.

[30] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013.

[31] R. Impagliazzo, R. Paturi, F. Zane, Which problems have strongly exponential complexity, J. Comput. Syst. Sci. 63 (4) (2001) 512–530.

[32] M. Naor, L.J. Schulman, A. Srinivasan, Splitters and near-optimal derandomization, in: Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 1995, pp. 182–191.

[33] D. Angluin, L. Valiant, Fast probabilistic algorithms for Hamiltonian circuits and matchings, J. Comput. Syst. Sci. 18 (2) (1979) 155–193, https://doi.org/10.1016/0022-0000(79)90045-X, http://www.sciencedirect.com/science/article/pii/002200007990045X.

[34] C. Boucher, C. Lo, D. Lokshtanov, Outlier detection for DNA fragment assembly, CoRR, arXiv:1111.0376 [abs], 2011.