Geometric Clustering: Fixed-Parameter Tractability and Lower Bounds with Respect to the Dimension

Sergio Cabello∗ Panos Giannopoulos† Christian Knauer‡ Günter Rote‡

Abstract
We present an algorithm for the 3-center problem in \((\mathbb{R}^d, L_\infty)\), i.e., for finding the smallest side length for 3 cubes that cover a given \(n\)-point set in \(\mathbb{R}^d\), that runs in \(O(n \log n)\) time for any fixed dimension \(d\). This shows that the problem is fixed-parameter tractable when parameterized with \(d\). On the other hand, using tools from parameterized complexity theory, we show that this is unlikely to be the case with the \(k\)-center problem in \((\mathbb{R}^d, L_2)\), for any \(k \geq 2\). In particular, we prove that deciding whether a given \(n\)-point set in \(\mathbb{R}^d\) can be covered by the union of 2 balls of given radius is \(W[1]\)-hard with respect to \(d\), and thus not fixed-parameter tractable unless \(FPT=W[1]\). Our reduction also shows that even an \(O(n^{o(d)})\)-time algorithm for the latter does not exist, unless \(SNP \subseteq DTIME(2^{o(n)})\).

Keywords: Clustering, Fixed-parameter tractability, Complexity, Lower bound, Dimension.

1 Introduction
A common type of facility location or clustering problem is the \(k\)-center problem, which is defined as follows: Given a set \(P\) of \(n\) points in a metric space and a positive integer \(k\), find a set of \(k\) supply points such that the maximum distance between a point in \(P\) and its nearest supply point is minimized. For the cases of the \((\mathbb{R}^d, L_2)\) and \((\mathbb{R}^d, L_\infty)\)-metric the problem is usually referred to as the Euclidean and rectilinear \(k\)-center respectively. Drezner [6] describes many variations of the facility location problem and their numerous applications. \(k\)-center problems as well as other clustering problems can be formulated as geometric optimization problems and, as such, they have been studied extensively in the field of computational geometry; see, for example, the survey by Agarwal and Sharir [1] and the references therein.

Our Results. We show that the rectilinear 3-center problem can be solved in \(O(6^d dn \log (dn))\) time, which is a considerable improvement over the fastest currently known algorithm of Assa and Katz [2], which runs in \(O(n^{d/3} \log n)\) time. Our algorithm is based on two ingredients. First, we solve the corresponding decision problem in \(O(6^d dn + dn \log n)\) time by an elegant and quite simple reduction to 2-satisfiability (2SAT). In the decision problem, one asks whether \(P\) can be covered by 3 axis-aligned cubes of given size. Second, we use the technique by Frederickson and Johnson [10] to efficiently search among the candidate values for the optimal side length of the cubes.

In terms of parameterized complexity theory (see below), our algorithm proves that the rectilinear 3-center problem is fixed-parameter tractable with respect to \(d\): the running time is of the form \(O(f(d) \cdot n^c)\), where \(c\) is independent of \(d\). Note that we cannot expect an algorithm that is polynomial in \(n\) and \(d\) because the problem is NP-hard [12].

On the negative side, we prove that the Euclidean \(k\)-center can probably not be solved with a running time of the form \(O(f(d) \cdot n^c)\), even when \(k = 2\). More precisely, we show that the corresponding decision problem is \(W[1]\)-hard with respect to \(d\). The decision problem amounts to deciding whether \(P\) can be covered by 2 balls of given radius. We prove this by an fpt-reduction from the \(k\)-independent set problem in general graphs, which is known to be \(W[1]\)-complete [8]. Moreover, our reduction implies that this problem and, consequently, the Euclidean \(k\)-center problem, for any \(k \geq 2\), cannot be solved in \(O(n^{o(d)})\) time unless \(SNP \subseteq DTIME(2^{o(n)})\). This considerably strengthens Megiddo’s [12] result that Euclidean 2-center is NP-hard.

Parameterized Complexity. We review some basic definitions of parameterized complexity theory; for an introduction to the field, the reader is referred to the textbooks by Downey and Fellows [5], and Flum and Grohe [8]. A problem with input size \(n\) and a positive integer parameter \(k\) is fixed-parameter tractable if it can be solved by an algorithm that runs in \(O(f(k) \cdot n^c)\) time, where \(f\) is a computable function depending only on \(k\), and \(c\) is a constant independent of \(k\); such an algorithm is (informally) said to run in fpt-time. The class of all fixed-parameter tractable problems is denoted by \(FPT\). An infinite hierarchy of classes, the \(W\)-hierarchy,
has been introduced for establishing fixed-parameter tractability. Its first level, \(W[1]\), can be thought of as the parameterized analog of NP: a parameterized problem that is hard for \(W[1]\) is not in FPT unless \(\text{FPT} = \text{W}[1]\), which is considered highly unlikely under standard complexity theoretic assumptions. Hardness is sought via fpt-reductions: an \textit{fpt-reduction} is an fpt-time Turing reduction from a problem \(\Pi\), parameterized with \(k\), to a problem \(\Pi'\), parameterized with \(k'\), such that \(k' \leq g(k)\) for some computable function \(g\).

**Related Results.** Efficient polynomial-time algorithms have been found for the planar \(k\)-center problem when \(k\) is a small constant [3, 7, 14]. Also, the rectilinear 2-center problem can be solved in polynomial time, even when \(d\) is part of the input [12]. However only, \(O(n^{O(kd)})\)-time algorithms are known when both \(k\) and \(d\) are part of the input, in particular, for \(k \geq 2\) and \(d \geq 2\) for the Euclidean case [1], and \(k \geq 3\) and \(d \geq 6\) for the rectilinear case [2]. As for lower bounds, the only ones known come from classical complexity theory: Both the Euclidean and rectilinear (decision) problems are NP-hard, for \(d = 2\) when \(k\) is part of the input [9, 13], while, as mentioned above, the Euclidean 2-center and rectilinear 3-center are NP-hard when \(d\) is part of the input [12]. These results do not exclude the possibility of algorithms in which the exponent of \(n\) in the running time is independent of the parameter \(k\) or \(d\) or both.

In this paper, we study the parameterized \(k\)-center problem when the dimension \(d\) is the parameter. For the case where the number \(k\) of clusters is considered as the parameter, Marx [11] showed that the rectilinear \(k\)-center decision problem is \(W[1]\)-hard, for any \(d \geq 2\); according to him (personal communication), the reduction carries over to the Euclidean case as well.

**Note:** While this paper was under review, Dániel Marx has shown that the rectilinear 4-center decision problem is \(W[1]\)-hard when parameterized with the dimension \(d\).

### 2 The Rectilinear 3-Center Problem

The function \(x_j\) denotes the projection onto the \(j\)-coordinate axis. Therefore, \(x_j(p)\) is the \(j\)-th coordinate of a point \(p\) and \(x_j(A)\) is an interval for any cube \(A\).

**Theorem 2.1.** (a) Given \(n\) points in \(d\) dimensions, we can decide whether they can be covered by three axis-aligned cubes of given side length \(m\) in \(O(6^d \cdot dn + 6n \log n)\) time.

(b) The smallest side length \(m\) for which the given points can be covered can be determined in \(O(6^d \cdot dn \log(dn))\) time.

**Proof.** (a) We can assume w.l.o.g. that \(m = 1\). Let \(P = \{p_1, \ldots, p_n\}\) be the input point set. We denote the three cubes by \(A\), \(B\), and \(C\). Each cube is the Cartesian product of \(d\) unit intervals.

Projecting the \(n\) points and the cubes on the \(j\)-th coordinate axis, we get \(n\) real numbers \(x_j(p_u)\) and \(3\) unit intervals \(x_j(A), x_j(B),\) and \(x_j(C)\) (whose positions are to be determined). We sort the coordinates of the points in each of the coordinate directions in \(O(dn \log n)\) time.

We have a covering if we can assign every point \(p_u\) to one of the cubes \((A, B,\) or \(C)\) such that, in each coordinate, this point is covered by the interval corresponding to the assigned cube.

In the following, we will consider the dimensions separately. We will look at the projection on each coordinate \(j\) and try to see by which interval a point can be covered in this coordinate. Let the minimum and maximum coordinate values be \(l_j\) and \(r_j\).

If the diameter \(r_j - l_j\) is at most one, we can, for example, align the three left interval endpoints with the leftmost point \(l_j\). Then, in this coordinate, all points are covered by all intervals. This means that we can eliminate this coordinate from consideration. From now on, we will assume that all these irrelevant coordinates have been eliminated, and thus, the diameter in coordinate \(j\) is bigger than one. Then we can assume, w.l.o.g., that no interval sticks out to the left of \(l_j\) or to the right of \(r_j\). On the other hand, these points must be covered by \textit{some} interval. Thus we can make the following assumption:

In dimension \(j\), one of the intervals \(x_j(A), x_j(B),\) \(x_j(C)\) has its left endpoint aligned with the leftmost point \(l_j\). Another interval has its right endpoint aligned with the rightmost point \(r_j\). The third interval (the “middle” interval) lies between these two positions. Intuitively we can see the middle interval “floating” between \(l_j\) and \(r_j\) because its position is not yet determined. The boundary cases, where the middle interval coincides with the left or right interval, are permitted.

We can thus classify the solutions into \(6^d\) patterns, according to the intervals \((x_j(A), x_j(B),\) or \(x_j(C))\) which are the left, middle, and right intervals in each coordinate direction. Formally, a pattern is represented as a sequence \((L_1, M_1, R_1), \ldots, (L_d, M_d, R_d)\), where each triplet \((L_j, M_j, R_j)\) is a permutation of the three symbols \(A, B, C\).

Let us restrict our attention to one fixed pattern. We now describe how to model this restricted covering problem as a logical satisfiability problem in conjunctive normal form, and decide whether such restricted covering exists in \(O(dn)\) time.

We have \(3n\) Boolean variables \(y_{A,u}, y_{B,u}, y_{C,u}\). The variable \(y_{X,u}\) represents the fact that point \(p_u\) is covered by box \(X\), for \(X = A, B, C\).
We have the $n$ covering clauses

\[(2.1) \quad (y_{Au} \lor y_{Bu} \lor y_{Cu}),\]

for $u = 1, \ldots, n$, expressing the fact that every point is covered (by at least one box).

Let us now look at some dimension $j$, where $x_j(L_j)$, $x_j(M_j)$, $x_j(R_j)$ are the left, middle, and right interval in dimension $j$ according to the chosen pattern. $(L_j, M_j, R_j)$ is a permutation of $A, B, C$.

The positions of the intervals $x_j(L_j)$ and $x_j(R_j)$ are fixed, and we only have to decide the position of the middle interval $x_j(M_j)$, that floats between $l_j$ and $r_j$.

When $x_j(p_u) > l_j + 1$, the point $p_u$ cannot be covered by the box $L_j$, and we can put the following set of clauses with one literal:

\[(2.2) \quad (\neg y_{Lju}),\]

for all $u$ with $x_j(p_u) \geq l_j + 1$. A similar argument applies to the box $R_j$, and we can put the following set of clauses:

\[(2.3) \quad (\neg y_{Rju})\]

for all $u$ with $x_j(p_u) < r_j - 1$. We can cover two points $p_u$ and $p_v$ with the box $M_j$ only if the distance between $x_j(p_u)$ and $x_j(p_v)$ is at most one. Thus we add the following set of clauses:

\[(2.4) \quad (\neg y_{Mju} \lor \neg y_{Mjv}),\]

for all $u, v$ with $|x_j(p_u) - x_j(p_v)| > 1$.

**LEMMA 2.1.** There is a covering conforming to the chosen pattern if and only if the clauses (2.1–2.4) are satisfiable.

**Proof.** Suppose we have a covering conforming to the chosen pattern. Set $y_{Xu}$ to true if and only if point $p_u$ is covered by box $X$. Then it is easy to check that all clauses are satisfied.

Conversely, assume that we have a Boolean assignment that satisfies all clauses. In each dimension $j$, the intervals $x_j(L_j)$ and $x_j(R_j)$ are already fixed, and we place the interval $x_j(M_j)$ as follows: we align its left endpoint with the leftmost point $x_j(p_u)$ (in dimension $j$) for which $y_{Mju}$ is true. This defines the position of the boxes $A, B, C$.

For a point $p_u$ the clauses (2.1) imply that at least one of $y_{Au}$, $y_{Bu}$, $y_{Cu}$ is true. We have to show that, if $y_{Xu}$ is true, then these chosen unit intervals for box $X$ cover point $p_u$ in every dimension.

If $X = L_j$ or $X = R_j$ in dimension $j$, the clauses (2.2) or (2.3) ensure that point $p_u$ is covered in dimension $j$. Thus, suppose finally that $X = M_j$. The interval for $M_j$ was chosen such that $x_j(p_u)$ does not lie to the left of $x_j(M_j)$. If $x_j(p_u)$ lies to the right of $x_j(M_j)$ it means that some point $p_v$, whose distance $x_j(p_u) - x_j(p_v)$ from $p_u$ is bigger than 1, has also $y_{Mjv}$ true. This contradicts the clause (2.4). □

All clauses except the clauses (2.1) contain at most two literals. We will now show that the clauses (2.1) can be eliminated, turning the problem into a 2-satisfiability problem, which can be solved in linear time.

Any of the clauses (2.2) or (2.3) effectively sets a variable to false, and it can be immediately used to eliminate a literal from one of the clauses (2.1). If we perform this elimination for all literals, we end up with $n$ clauses, each of which contains a subset of $y_{Au}$, $y_{Bu}$, $y_{Cu}$. (If we obtain an empty clause, we know that the problem is not satisfiable.) We call the reduced covering clauses the resulting clauses which contain at most two literals, and we denote them by (2.1').

**LEMMA 2.2.** There is a covering conforming to the chosen pattern if and only if the clauses (2.1') and (2.2–2.4) are satisfiable.

**Proof.** The new set of clauses is weaker than the old one: it is derived by drawing logical conclusions (actually, some form of resolution), and omitting the clauses with three literals. Therefore, when the clauses (2.1–2.4) are satisfiable, also the new set of clauses is satisfiable.

Thus we only have to show that the clauses (2.1–2.4) are satisfiable whenever the reduced system of clauses is satisfiable.

A reduced clause (2.1') implies that the corresponding original clause is also satisfied. Consider now a clause (2.1) for a point $p_u$ which remains intact during the reduction process. None of $y_{Au}$, $y_{Bu}$, and $y_{Cu}$ ever appears in a clause (2.2) or (2.3). In other words, in each dimension $j$, point $p_u$ lies within distance 1 both of the leftmost point $l_j$ and of the rightmost point $r_j$; see Figure 1. This means that point $p_u$ is covered by all three intervals, no matter where the interval $x_j(M_j)$ is.
On the logical level, none of \( y_{Au}, y_{Bu}, \) and \( y_{Cu} \) appears in the clauses (2.4), and thus they do not appear in negated form at all. We can thus satisfy the clause 
\((y_{Au} \lor y_{Bu} \lor y_{Cu})\) simply by setting all three variables to true. \( \square \)

Thus we have reduced the covering problem for a fixed pattern to an equivalent 2-SAT instance. There are \( O(n) \) clauses of type (2.1'), \( O(dn) \) clauses of types (2.2) and (2.3), but \( O(dn^2) \) clauses of type (2.4).

The clauses of the last type can be replaced by \( O(dn) \) clauses by introducing auxiliary variables, as follows: Let us look at a fixed dimension \( j \). The \( O(n^2) \) clauses of the form (2.4) involve the \( n \) variables \( y_{M_j u} \), which we abbreviate by \( w_u \), and we assume for simplicity of notation that the points are ordered by the \( j \)-th coordinate: \( x_j(p_1) \leq x_j(p_2) \leq \cdots \leq x_j(p_n) \).

The \( O(n^2) \) clauses of the form (2.4) can be equivalently written as implications:

\[(2.5) \quad w_u \Rightarrow \neg w_v, \]

whenever \( x_j(p_u) - x_j(p_v) > 1 \).

We introduce auxiliary variables \( z_u \) that are intended to represent the fact that the interval for \( M_j \) starts left of \( x_j(p_u) \) or at \( x_j(p_u) \). Then we have the implications

\[(2.6) \quad w_u \Rightarrow z_u, \]

for \( u = 1, \ldots, n \), and

\[(2.7) \quad z_u \Rightarrow z_{u+1}, \]

for \( u = 1, \ldots, n - 1 \).

Finally, for a given point \( p_v \) with \( x_j(p_v) > l_j + 1 \), let \( \bar{u}(v) \) denote the largest index \( u \) such that \( x_j(p_u) < x_j(p_v) - 1 \), (i.e., \( p_{\bar{u}(v)} \) is the right-most point with this property). Then we add the \( O(n) \) clauses

\[(2.8) \quad z_{\bar{u}(v)} \Rightarrow \neg w_v, \]

for all \( v = 1, \ldots, n \) with \( x_j(p_v) > l_j + 1 \). We have omitted the reference to \( j \) for the variables \( w \) and \( z \), but it should be kept in mind that this procedure has to be carried out for each dimension \( j \) separately.

**Lemma 2.3.** For any given values of the variables \( w_1, \ldots, w_n \), the clauses (2.5) are satisfied if and only if there is a truth assignment for the variables \( z_1, \ldots, z_n \), that satisfies (2.6–2.8).

**Proof.** If we have a truth assignment \( w_1, \ldots, w_n \) satisfying the clauses (2.5), we set \( z_u := w_1 \lor w_2 \lor \cdots \lor w_n \). Then (2.6) and (2.7) are satisfied by construction. To prove (2.8), assume for contradiction that \( w_v \) and \( z_{\bar{u}(v)} \)

are true, for some \( v \). By the definition of \( z_{\bar{u}(v)} \), there is some true \( w_u \) with \( u \leq \bar{u}(v) \). Since we have \( x_j(p_u) \leq x_j(p_{\bar{u}(v)}) \leq x_j(p_v) - 1 \) and \( w_u, w_v \) are true, then \( w_u, w_v \) violate (2.5).

Conversely, assume that (2.6–2.8) is fulfilled, and let us prove (2.5) for each pair \( u, v \) with \( x_j(p_u) < x_j(p_v) - 1 \). The clauses (2.8) include the clause \( z_{\bar{u}(v)} \Rightarrow \neg w_v \), and from the definition of \( \bar{u}(v) \) we have \( u \leq \bar{u}(v) < v \). Thus, the chain of implications \( w_u \Rightarrow z_u \Rightarrow z_{u+1} \Rightarrow \cdots \Rightarrow z_{\bar{u}(v)} \Rightarrow \neg w_v \) proves (2.5). \( \square \)

We have reduced the number of clauses to \( O(dn) \), and each clause has at most two literals. The clauses can be generated in \( O(dn) \) time if the input coordinates are sorted in each dimension, and the satisfiability of these clauses can be tested in \( O(dn) \) time as well. This procedure has to be repeated for each of the \( d^2 \) patterns. This concludes the proof of part (a) of Theorem 2.1.

(b) The minimum side length \( m \) for which the given points are covered is one of the \( O(dn^2) \) pairwise distances \( |x_j(p_u) - x_j(p_v)| \). We initially sort in \( O(dn \log n) \) time the input coordinates in each dimension. For each dimension \( j \), assuming for simplicity of notation that the points are indexed such that \( x_j(p_1) \leq x_j(p_2) \leq \cdots \leq x_j(p_n) \), we define an \( n \times n \) matrix \( \Delta^j = \{ \delta^j_{uv} \} \) with entries \( \delta^j_{uv} = x_j(p_u) - x_j(p_v) \). Each matrix \( \Delta^j \) is a sorted matrix: each column has non-decreasing values and each row has non-increasing values. The matrices \( \Delta^1, \ldots, \Delta^d \) are not constructed explicitly, but only some of their entries will be evaluated. Let \( \Delta \) denote the multiset of \( dn^2 \) entries in \( \Delta^1, \ldots, \Delta^d \). Clearly, the sought value \( m \) is one of the values in \( \Delta \).

Frederickson and Johnson [10] showed how to select for any \( 1 \leq k \leq dn^2 \) the \( k \)-th largest entry in the collection of sorted matrices \( \Delta^1, \ldots, \Delta^d \) evaluating \( O(dn) \) entries. In our scenario, any desired entry \( \delta^j_{uv} \) can be obtained in \( O(1) \) time, after the initial sorting of the coordinates. Thus, we can find the \( k \)-th largest value of \( \Delta \) in \( O(dn) \) time.

We can now perform a binary search for \( m \) on the entries of \( \Delta \). Since \( \Delta \) has \( dn^2 \) values, the binary search requires \( O(d \log(dn)) \) calls to the selection procedure and applications of the decision algorithm from part (a). Therefore, each of the \( O(d \log(dn)) \) steps of the binary search requires \( O(d^4 \cdot dn) \) time, after the initial sorting of the coordinates. \( \square \)

### 3 The Euclidean 2-Center Problem

In this section we give an fpt-reduction from the parameterized \( k \)-independent set problem in general graphs, which is known to be \( W[1] \)-complete [8], to the Euclidean 2-center decision problem, parameterized with the dimension \( d \). Let \( [n] = \{1, \ldots, n\} \). Let \( k \) be the
size of the independent set being looked for in a graph $G([n], E)$. We assume that $n \geq 4$ and $n$ is even, by adding an additional vertex to $G$ if necessary and connecting it to all other vertices. Using $G$, we will construct a point set $P$ in $\mathbb{R}^{2k+1}$ with the property that $P$ can be covered by 2 unit balls if and only if $G$ has an independent set of size $k$.

We first give a high-level overview of our reduction at the logical level. We start with a scaffolding point set $P^0$ of $nk + 2$ points. For an appropriate radius $\rho$, the set $P^0$ has the property that there are $n^k$ ways to cover it with two balls of radius $\rho$, in one-to-one correspondence with all $k$-tuples $(u_1, \ldots, u_k)$ with $1 \leq u_i \leq n$. These coverings allow us to represent the potential independent sets of vertices in the graph. More precisely, they represent ordered selections of $k$ not necessarily distinct vertices of the graph.

The structure of the input graph is represented using additional constraint points: for each pair of distinct indices $i \neq j$ (1 $\leq i, j \leq k$) and for each pair of (possibly equal) vertices $u, v \in [n]$, we define a constraint point $q^k_{ij}$ which is covered by all solutions $(u_1, \ldots, u_k)$ with the exception of those with $u_i = u$ and $u_j = v$. In particular, we add to $P^0$ the $nk$ constraint points $Q_V = \{ q^k_{uv} \mid 1 \leq u \leq n, 1 \leq i < j \leq k \}$ to ensure that all components $u_i$ in a solution must be distinct. Also, for each edge $uv \in E$ we add all $(k-1)$ points $q^k_{ij}$ with $i \neq j$. In this way, we ensure that the remaining coverings $(u_1, \ldots, u_k)$ represent independent sets of size $k$. In total the edges are represented by $k(k-1)|E|$ points $Q_E = \{ q^k_{ij} \mid uv \in E, 1 \leq i, j \leq k, i \neq j \}$. The resulting set $P = P^0 \cup Q_V \cup Q_E$ will have in total $nk + 2 + \binom{k}{2}(n+2|E|) + k$ points. A covering of $P$ exists if and only if the graph has an independent set of size $k$. (Each independent set of size $k$ is represented by $k!$ coverings.)

We will first describe the geometry of the point sets exactly, as if exact square roots and expressions of the form $\frac{\sin}{n}$ were available. We will later show that the essential features of our construction are preserved when the data are perturbed within some tolerance. This allows us to work with fixed-precision roundings of the exact construction, making the reduction suitable for the Turing machine model.

**Notation.** For our construction it is convenient to view $\mathbb{R}^{2k+1}$ as the product of $k$ orthogonal planes $E_1, \ldots, E_k$, where each $E_i$ has coordinate axes $X_i, Y_i$, plus an extra axis denoted by $Z$. For giving coordinates, the axes are considered in the order $X_1, Y_1, \ldots, X_k, Y_k, Z$. The coordinate on $X_i$, $Y_i$, and $Z$ of a point $p$ is denoted by $x_i(p)$, $y_i(p)$, and $z(p)$, respectively.

Two antipodal points of $P_i$ and the top anchor form an isosceles triangle whose circumradius is $5/4$. Therefore, if $\rho < 5/4$, the top ball (or the bottom ball) cannot contain two antipodal points. (With a radius of $5/4$, the top ball could be centered on the $Z$-axis at
height 3/4 and cover all points \( P^0 \) except the bottom anchor.) By choosing \( \rho < 5/4 \), we ensure that each ball can cover at most half of the points from every \( P_i \). We define the radius \( \rho \) as the smallest radius such that the top ball can cover precisely \( n/2 \) consecutive points of each subset \( P_i \), besides the anchor \( p_2 \). From the discussion, it is clear that \( 1 < \rho < 5/4 \). The precise value of \( \rho \) will be given below.

Let \( A(u_1, \ldots, u_k) \) be the set of \( 2k \) points
\[
A(u_1, \ldots, u_k) = a_{1u_1} \cup a_{2u_2} \cup \cdots \cup a_{ku_k}.
\]
We denote by \( B(u_1, \ldots, u_k) \) the smallest enclosing ball of \( A(u_1, \ldots, u_k) \cup \{ p_2 \} \). The intersection of \( B(u_1, \ldots, u_k) \) with \( E_i \) is a disk of radius smaller than 1 that contains the pair \( a_{iu_i} \). It follows that \( B(u_1, \ldots, u_k) \) also contains the \( n/2 \) consecutive points of \( P_i \) between the points of the pair \( a_{iu_i} \). Since the planes \( E_1, \ldots, E_k \) are orthogonal, each \( u_i \) independently defines which of the \( n/2 \) consecutive points of the sets \( P_i \) is covered by \( B(u_1, \ldots, u_k) \). The complementary halves can then be covered by the bottom ball. In total, we have \( n^k \) possible partitions of \( P^0 \) into two groups covered by the two balls, which correspond to the \( n^k \) possible tuples \((u_1, \ldots, u_k) \in [n]^k \).

**Lemma 3.1.** All balls \( B(u_1, \ldots, u_k) \) have the same radius
\[
\rho = \frac{5}{4} \cdot \frac{k}{\sqrt{k + \sigma^2/4}}.
\]
The center \( c \) of \( B(u_1, \ldots, u_k) \) has coordinates
\[
\begin{align*}
x_i(c) &= -w \cos \left( \frac{u_i - 1}{n} \pi \right), \\
y_i(c) &= -w \sin \left( \frac{u_i - 1}{n} \pi \right), \\
z(c) &= h,
\end{align*}
\]
with
\[
w = \frac{5 \sigma}{2(4k + \sigma^2)}; \quad h = \frac{3k + 2 \sigma^2}{4k + \sigma^2}.
\]
**Proof.** By symmetry, it is sufficient to show this for the ball \( B(1, \ldots, 1) \), whose center we claim to be \( c = (0, -w, 0, -w, \ldots, 0, -w, h) \). We use the following well-known characterization of the smallest enclosing ball:

**Proposition 3.1.** A ball \( B \) containing a finite set of points \( A \) is the smallest enclosing ball for \( A \) if and only if its center lies in the convex hull of the points of \( A \) that lie on the boundary of \( B \). □

It is straightforward to check that all points of \( A(1, \ldots, 1) \) have the same distance \( \rho \) to \( c \). Moreover, the center \( c \) lies on the line segment between \( p_2 \) and the center of gravity of the remaining \( 2k \) points of \( A(1, \ldots, 1) \), which is the point \((0, -\sigma/k, 0, -\sigma/k, \ldots, 0, -\sigma/k, 0)\). Thus it lies in the convex hull of \( A(1, \ldots, 1) \). □

We will later need the fact that \( 3/4 < h < 1 \), which follows from \( \sigma^2 < k \) and \( h > 1 - k/(4k + \sigma^2) > 1 - k/(4k + \sigma^2) = 3/4 \). Note that by symmetry, any bottom ball will have its center \( c' \) with \( z(c') = -h \). We conclude with the following characterization of the possible coverings of \( P^0 \) with two balls of radius \( \rho \).

**Lemma 3.2.** Assume that two balls \( B, B' \) of radius \( \rho \) cover \( P^0 \), and that \( p_{2z} \in B \). Then there is a tuple \((u_1, \ldots, u_k) \in [n]^k \) such that \( B = B(u_1, \ldots, u_k) \) and \( B' = -B(u_1, \ldots, u_k) \).

**Proof.** As discussed before, \( B \) or \( B' \) can contain at most \( n/2 \) consecutive points of \( P_i \), and therefore, \( B \) and \( B' \) must cover complementary halves of each \( P_i \). If \( B \) covers the halves between the pairs \( a_{iu_i}, a_{ku_k} \), it follows from the uniqueness of the minimum enclosing balls that \( B = B(u_1, \ldots, u_k) \). Since \( B' \) covers the complementary halves of each \( P_i \) and \( -p_2 \), it follows that \( B' = -B(u_1, \ldots, u_k) \).

From this characterization, the bijection between the possible coverings of \( P^0 \) and \( [n]^k \) is clear. Every covering consists of a symmetric pair of balls \( B = B(u_1, \ldots, u_k) \) and \( B' = -B(u_1, \ldots, u_k) \).

### 3.2 Constraint Points

We continue now the construction of point set \( P \), by showing how we encode the structure of \( G \). For each pair of distinct indices \( i \neq j \) (\( 1 \leq i, j \leq k \)) and for each pair of (possibly equal) vertices \( u, v \in [n] \), we define a constraint point \( q_{ij}^{uv} \). All constraint points lie on the hyperplane \( H = \{ p \in \mathbb{R}^{2k+1} \mid z(p) = h \} \), on which also the center of the top ball lies. This breaks the symmetry between top and bottom balls that the construction had until now. Since the center of the bottom ball lies on the hyperplane \( -H \) and \( \rho < 5/4 < 2h \), none of the constraint points can be covered by a bottom ball. Therefore, our discussion will only consider top balls. The constraint point \( q_{ij}^{uv} \) will lie in all top balls \( B(u_1, \ldots, u_k) \) except in those with \( u_i = u \) and \( u_j = v \).

We choose \( q_{ij}^{uv} \) in the four-dimensional affine subspace (4-flat)
\[
F_{ij} = \{ p \in \mathbb{R}^{2k+1} \mid x_i(p) = y_r(p) = 0, \text{ for } r \neq i, j, \text{ and } z(p) = h \}
\]
where \( o' = E_i \times E_j \),
\[
D = \{ B(u_1, \ldots, u_k) \cap F_{ij} \mid (u_1, \ldots, u_k) \in [n]^k \}.
\]
The intersection of any ball \( B = B(u_1, \ldots, u_k) \) with \( F_{ij} \) is a 4-dimensional ball \( D \), whose center \( c \) is the
orthogonal projection of the center of $B$ on $F_{ij}$. From Lemma 3.1, we have

$$x_i(c) = -w \cos \theta_i, \quad y_i(c) = -w \sin \theta_i,$$

$$x_j(c) = -w \cos \theta_j, \quad y_j(c) = -w \sin \theta_j,$$

where $\theta_i = (u_i - 1) \frac{2\pi}{n}$ and $\theta_j = (u_j - 1) \frac{2\pi}{n}$. The location of the center $c$ thus depends only on $u_i$ and $u_j$. We denote this center by $c_{ij}^{uv}$.

Looking at the distance between the centers of $B, D$, and $o'$, we get the following properties:

a) Every ball $D$ has radius $\rho_* = \sqrt{\rho^2 - (k-2)w^2}$;

b) A point $q \in F_{ij}$ lies in the ball $B(u_1, \ldots, u_k)$ if and only if $|q - c_{ij}^{uv}| \leq \rho_*$;

c) The center $c$ of $D$ lies on the three-dimensional sphere $C = \{ p \in F_{ij} \mid |p - o'| = w\sqrt{2} \}$;

d) The sphere $C$ is contained in the interior of $D$.

Let $D_{ij}^{uv}$ denote the ball in $F_{ij}$ with center $c_{ij}^{uv}$ and radius $\rho_*$. For each $u, v$, we want to find a point $q_{ij}^{uv} \in F_{ij}$ that lies outside the ball $D_{ij}^{uv}$ but in all other balls of $\mathcal{D}$.

Since the centers $c_{ij}^{uv} \in F_{ij}$ form a completely symmetric set and all balls have the same radius, we can find this point as follows. (See Fig. 3a for a two-dimensional analog of this situation.) Start at $c_{ij}^{uv}$ and move along the ray $L^+ = \{ c_{ij}^{uv} + \lambda(o' - c_{ij}^{uv}) \mid \lambda \geq 0 \}$ through $o'$. By properties (c) and (d), we are initially inside all balls $D \in \mathcal{D}$. At some point $l_1$, we hit the boundary of some ball. We prove below that this ball is $D_{ij}^{uv}$. Thus, after passing $l_1$, we are outside $D_{ij}^{uv}$ but still inside any ball $D \neq D_{ij}^{uv}$. We place $q_{ij}^{uv}$ at the point $l_2$ where $L^+$ intersects the boundary of the next ball.

**Lemma 3.3.** The ray $L^+$, after having visited $o'$, hits the boundary of the ball $D_{ij}^{uv}$ before the boundary of any other ball $D \in \mathcal{D}$.

**Proof.** Let $l_1$ be the point where $L^+$ intersects the boundary of $D_{ij}^{uv}$, and let $c$ be the center of any other ball $D \in \mathcal{D}$ (see Fig. 3b). By triangle inequality, we have

$$|c - l_1| = |c - o'| + |o' - l_1| < |c_{ij}^{uv} - o'| + |o' - l_1| = |c_{ij}^{uv} - l_1| = \rho_*.$$

This implies that the boundary of $D_{ij}^{uv}$ is the first boundary intersected by $L^+$. □

### 3.3 The Reduction

As mentioned in the beginning of this section, we add $\binom{n}{2}(n + 2|E|)$ constraint points $Q_V$ and $Q_E$ to the scaffolding set $P^0$ to represent the structure of the input graph $G([n], E)$.

**Lemma 3.4.** The set $P = P^0 \cup Q_V \cup Q_E$ can be covered with two balls of radius $\rho$ if and only if $G$ has an independent set of size $k$.

**Proof.** Any covering of $P$ with two balls $B, B'$ of radius $\rho$ must consist of the two balls $B = B(u_1, \ldots, u_k)$ and $B' = -B(u_1, \ldots, u_k)$ for some tuple $(u_1, \ldots, u_k)$, by Lemma 3.2. Since the constraint points exclude the tuples with two equal indices $u_i = u_j$, or with indices $u_i$ and $u_j$ when $(u_i, u_j)$ is an edge of $G$, the coverings represent precisely the independent sets of $G$. □

**Rounding coordinates.** To make the reduction suitable for a Turing machine, we round all data to multiples of a small “unit” $U$. Scaling by $U$ will then convert the input to integral data. We will show that...
choosing $U = \Theta(1/(n^6 k^2))$ will preserve all important characteristics of our point set. Since it is easy to evaluate $\sin^{-1}$ or $\sqrt{\cdot}$ to this precision of $O(\log(nk))$ bits, the reduction can be carried out in polynomial time. More precisely, we replace each input coordinate $x$ by a multiple $\hat{x}$ of $U$ in the range $x - U < \hat{x} < x + U$. This ensures that each input point is moved at most $\sqrt{U}$. $U$ from its original position. (Recall that most coordinates of our input points are 0.) We replace $\rho$ by a multiple $\hat{\rho}$ of $U$ in the range $\rho + \sqrt{U} \leq \hat{\rho} \leq \rho + \sqrt{U} + 2U$. In this way we ensure that for each ball that covers some set of input points, there exists an enlarged ball with radius $\hat{\rho}$ that covers the same input points. We want to exclude the possibility that the enlarged ball covers additional points (possibly after moving its center). In the following we will give only asymptotic estimates. The detailed calculations will be given in the full paper.

**Proposition 3.2.** Every input point that is not in one of the balls $B(u_1, \ldots, u_k)$ is at least $\varepsilon_1 = \Omega(1/(n^3 k))$ away from this ball. \hfill \Box

(This is the bound for the constraint points: the distance between $l_1$ and $l_2$ is at least $w/n^2 = \Omega(1/(n^3 k))$ (see Fig. 3a). The scaffolding points are at least $\Omega(1/(nk))$ away from the ball.) Since we have introduced some slack by enlarging the radius and moving the points, the center of the ball may move away from the original center. The following lemma bounds this movement.

**Lemma 3.5.** If the center of the ball $B(u_1, \ldots, u_k)$ is moved by more than $\varepsilon_2$, the distance to some point on its boundary increases by at least $\varepsilon_3 = \Omega(\min\{\varepsilon_2^2, \varepsilon_2/(nk)\})$. \hfill \Box

(The first bound comes from moving the center perpendicular to the hyperplane through the boundary points of the ball; the second one comes from any movement in this hyperplane.) It follows that the center $c$ of a ball with the modified data can move only $\varepsilon_2 = O(1/(n^3 k))$ from its original position, since otherwise, its distance from some boundary point would increase by $\varepsilon_3 = \Omega(1/(n^6 k^2))$. If $\varepsilon_3 > \sqrt{U} + (\sqrt{U} \cdot U + 2U)$, this means that the sphere can no longer contain this boundary point. Now, since $\varepsilon_2 + (\sqrt{U} \cdot U + 2U) < \varepsilon_1 - \sqrt{U}$, the sphere can swallow no additional points:

**Theorem 3.1.** If we chose $U = \text{const}/(n^6 k^2)$, for a sufficiently small constant, all possible coverings by two balls of radius $\hat{\rho}$ partition the rounded point set in the same way as two balls $B(u_1, \ldots, u_k)$ and $-B(u_1, \ldots, u_k)$ for the original data. \hfill \Box

Using the rounded coordinates for the points of $P$, and since $|P| = nk + 2 + \binom{k}{2} (n + 2 |E|)$, we see that this is an fpt-reduction. Thus, we have the following:

**Theorem 3.2.** The Euclidean 2-center decision problem is $W[1]$-hard with respect to the dimension $d$. \hfill \Box

For a parameterized complexity upper bound, the (integral) Euclidean 2-center decision problem, parameterized with $d$, is trivially in $W[P]$; see [5].

Since $d = 2k + 1$ in the above fpt-reduction, an $O(n^{o(d)})$-time algorithm for the Euclidean 2-center decision problem implies an $O(n^{o(k)})$-time algorithm for the parameterized $k$-independent set problem, which in turn implies that $\text{SNP} \subseteq \text{DTIME}(2^{o(n)})$ [4]. Thus:

**Corollary 3.1.** The Euclidean $k$-center problem, for any $k \geq 2$, cannot be solved in $O(n^{o(d)})$ time, unless $\text{SNP} \subseteq \text{DTIME}(2^{o(n)})$. \hfill \Box

### References


