On Small Pair Decompositions for Point Sets

Kevin Buchin
Department of Computer Science, TU Dortmund University, Germany

Jacobus Conradi
Institute of Computer Science, Universität Bonn, Germany
Basic Algorithms Research Copenhagen, University of Copenhagen, Denmark

Sariel Har-Peled
Department of Computer Science, University of Illinois, Urbana, IL, USA

Antonia Kalb
Department of Computer Science, TU Dortmund University, Germany

Abhiruk Lahiri
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
Heinrich Heine Universität Düsseldorf, Germany

Lukas Plätz
Institute of Computer Science, Ruhr University Bochum, Germany

Carolin Rehs
Department of Computer Science, TU Dortmund University, Germany
Department of Mathematics and Computing Science, TU Eindhoven, The Netherlands

Sampson Wong
Basic Algorithms Research Copenhagen, University of Copenhagen, Denmark

Abstract

We study the problem of computing a minimum-size Well-Separated Pair Decomposition (WSPD) of a given point set. We obtain the following results for the minimum-size WSPD: (1) a constant-factor approximation in doubling metrics, (2) a simple 3-approximation in $\mathbb{R}$, and (3) an NP-hardness proof in $\mathbb{R}^2$. We also provide an optimal output-sensitive runtime for the algorithm in doubling metrics and an implementation of the 3-approximation algorithm in $\mathbb{R}$. Furthermore, we introduce a new pair-decomposition for point sets in a metric space. It is defined using a relaxed requirement that, for all pairs $\{X, Y\}$ in the decomposition, all the distances of pairs of points in $X \times Y$ are equal up to a factor in $[1 \pm \varepsilon]$. Surprisingly, we show that in a general metric space, one can compute such a decomposition of size $O(\frac{n}{\varepsilon} \log n)$, which is dramatically smaller than the $\Theta(n^2)$ bound for WSPDs. For a point set in $\mathbb{R}^d$, the bound improves to $O(d\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Well-separated pair decomposition, Semi-separated pair decomposition, Approximation algorithms, Doubling metrics, Computational geometry

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.102

Related Version *Full Version*: `https://arxiv.org/abs/2601.22728`

Funding *Abhiruk Lahiri*: Partially supported by the Strategic Research Fund, Heinrich Heine Universität, Düsseldorf.
Carolin Rehs: Partly supported by the PRIME programme of the German Academic Exchange Service (DAAD) with funds from the German Federal Ministry of Research, Technology and Space (BMFTR).

Acknowledgements This paper results from the Rhine-Ruhr Computational Geometry Workshop 2025. We thank the organizers from FernUniversität Hagen.

1 Introduction

Given a set P of n points, the *Well-Separated Pair Decomposition* (WSPD), introduced by Callahan and Kosaraju [9], is an approximate compact representation of the metric induced by P. It has been used to solve numerous problems, such as the all-nearest-neighbors [9], approximate minimum spanning tree [9], spanners [25], Delaunay triangulations [8], distance oracles [15], and others [20]. See Smid [27] for a survey.

Given $\varepsilon \in (0, 1)$ and a set $P \subseteq \mathbb{R}^d$ of n points, the $\frac{1}{\varepsilon}$-WSPD represents the $\binom{n}{2}$ pairwise distances as a union of $O(n/\varepsilon^d)$ (edge-disjoint) bicliques, where all the distances in a single biclique are the same up to a multiplicative factor in $1 \pm O(\varepsilon)$. More precisely, for a pair $\{X, Y\}$ in a $\frac{1}{\varepsilon}$-WSPD, we require that $\max\bigl(\nabla(X), \nabla(Y)\bigr) \leq \varepsilon \cdot d(X, Y)$, where $\nabla(X)$ is the diameter of X, and $d(X, Y) = \min_{x \in X, y \in Y} d(x, y)$, where $d(x, y)$ denotes the distance between x and y. In other words, for a biclique $X \otimes Y$ in this cover, the diameters of the two sets are at most ε-fraction of their distance from each other. This biclique cover can be computed in $O(n \log n + n/\varepsilon^d)$ time, where each biclique is represented as a pair of nodes in a tree, see [9]. The *size* of the WSPD is the number of bicliques.

A natural question is: Using bicliques, how small of a representation of the pairwise distances in P can we construct? Specifically, what is the smallest number of bicliques required to cover the $\binom{n}{2}$ pairwise distances in $\binom{P}{2}$, so that distances in a biclique are roughly the same? Our approach to answering the above question is to first consider how to minimize the size of the WSPD, and then to introduce a relaxation of the WSPD and construct even smaller covers, where the diameter requirement on the biclique sides is removed.

Minimum-size WSPDs. While a bound of $O(n/\varepsilon^d)$ on the size of a $\frac{1}{\varepsilon}$-WSPD in $\mathbb{R}^d$ is given in [9], this bound can be far from optimal: In $\mathbb{R}^2$, the point set P with $p_i = \bigl(i, (4/\varepsilon)^i\bigr)$ for $i = 1, \ldots, n$ has a $\frac{1}{\varepsilon}$-WSPD formed by the pairs $X_i = \{p_1, \ldots, p_i\}$ and $Y_i = \{p_{i+1}\}$, for $i = 2, \ldots, n$. Its size is $(n-1)$[1], which is significantly smaller than the worst-case bound $\Theta(n/\varepsilon^2)$. Computing small WSPDs may have downstream effects on its applications, such as the sizes of WSPD-based spanners or distance oracles. This effect motivates us to study the minWSPD problem.

▶ **Problem 1** (minWSPD). *Given a set P of n points in a metric space $(\mathcal{X}, d)$, and a parameter ε, compute the minimum-size $\frac{1}{\varepsilon}$-WSPD $(\mathcal{W})$ of P. That is, minimize the number of pairs in $\mathcal{W}$, while guaranteeing that the resulting WSPD provides the desired separation.*

Semi-separated pair decomposition (SSPD). The requirements for a WSPD are, in some regards, quite stringent. Thus, for some applications, obtaining smaller pair decompositions by relaxing the requirements of the WSPD can be useful. A neat example of this is the semi-separated pair decomposition (SSPD), which requires only one of the sides in each pair to be small. The SSPD has found numerous applications [2, 3, 4, 1, 29]. Formally, for a pair $\{X, Y\}$ in $\frac{1}{\varepsilon}$-SSPD, we require that $\min\bigl(\nabla(X), \nabla(Y)\bigr) \leq \varepsilon \cdot d(X, Y)$ (note the min here instead of max used in WSPD). Thus, a semi-separated pair might have a "tiny" set (say) X on one side, while the other side diameter might be unbounded in terms of the distance to the other set, see Figure 1. For $\mathbb{R}^d$, there are constructions of $\frac{1}{\varepsilon}$-SSPD with $O(n/\varepsilon^d)$ pairs, with the striking property that their weight (i.e. the total size of the sets used in it) is $O(\varepsilon^{-d} n \log n)$. In contrast, the weight of the WSPD, in the worst case, is quadratic.

[1] We note that $n-1$ is in fact the smallest possible size of a PartitionWSPD (partition) of n points: by the Graham–Pollak theorem, the edges of K_n cannot be partitioned into fewer than $n-1$ complete bipartite graphs [5]. Hence this example is optimal among partition WSPDs.

Approximate Biclique Covers ($\frac{1}{\varepsilon}$-ABC). For some applications, the diameter of the two sides of the pairs are not important at all, as long as the distances are roughly the same. We define the Approximate Biclique Cover (ABCs) as a pair decomposition, such that for every pair $\{X, Y\}$, we require that $d(x, y) \le (1+\varepsilon) d(X, Y)$ for all $x \in X, y \in Y$. Such a pair $\{X, Y\}$ is *ε-stable*. This decomposition still provides sufficient information to efficiently approximate quantities such as the diameter, MST, or the closest pair. Specifically, small ABCs should be useful in speeding up approximate matching algorithms. While the three concepts of separation are inherently different (see Figure 1), in several applications that require only similar distances, the new ABC decompositions could be a useful replacement for WSPD or SSPD.

While covering graphs by bi/cliques is a well-known problem, we are unaware of any previous work on Approximate Biclique Covers (ABCs) for metric spaces. As the WSPD can have quadratic size and weight in general metrics, it is of particular interset whether one can obtain a small ABC in metrics that do not admit small WSPDs.

WSPDs in other metrics. Although WSPDs may have $\Omega(n^2)$ size in general metrics, they are known to have sub-quadratic size in special cases:

(I) Callahan and Kosaraju [9] construct a $\frac{1}{\varepsilon}$-WSPD of size $O(n/\varepsilon^d)$ in $\mathbb{R}^d$.

(II) Gao and Zhang [15] showed how to construct $\frac{1}{\varepsilon}$-WSPD of size $O(\varepsilon^{-4} n \log n)$ for the unit disk graph for a set P of n points in the plane (the bound is $O_\varepsilon(n^{2-2/d})$ for $d > 2$). The 2d bound was slightly improved to $O(\varepsilon^{-2} n \log n)$ by Har-Peled *et al.* [21].

(III) Har-Peled and Mendel [20] showed how to compute $\frac{1}{\varepsilon}$-WSPDs, of size $n/\varepsilon^{O(\dim)}$ for spaces with doubling dimension dim.

(IV) Gudmundsson and Wong [17] showed a $\frac{1}{\varepsilon}$-WSPD of size $O(\lambda^2 \varepsilon^{-4} n \log n)$ for a λ-low-density graph.

(V) Deryckere *et al.* [13] showed how to compute a linear size $\frac{1}{\varepsilon}$-WSPD for c-packed graphs.

WSPD as a min-covering problem. A WSPD represents $P \otimes P$ as a union of bicliques. While these bicliques are typically required to be disjoint, applications of the WSPD do not make use of the disjointness[2]. We therefore also investigate whether dropping this requirement results in significantly smaller WSPDs, referred to as CoverWSPD. In contrast, the disjoint version is PartitionWSPD (see Definition 9).

The problem of computing such a CoverWSPD can be seen as a SetCover instance, where the sets are all maximal well-separated pairs in $\binom{P}{2}$. When the number of such pairs is polynomial, this readily leads to $\Theta(\log n)$ approximation of the minimum-size CoverWSPD, see Section 2.5. In Section 6, we present algorithms with better approximation factors for $P \subset \mathbb{R}$, which in turn also leads to a constant-factor approximation of the minimum-size PartitionWSPD for such P.

Our Contributions

This paper presents new results for the minimum $\frac{1}{\varepsilon}$-WSPD and the $\frac{1}{\varepsilon}$-ABC.

(I) ABC and SSPD. In Section 3, we show, surprisingly, that *any* finite metric space admits a near-linear size ABC and SSPD. In particular, we construct a $\frac{1}{\varepsilon}$-ABC of $O(\frac{n}{\varepsilon} \log n)$ size that is also a $\frac{1}{\varepsilon}$-SSPD. Note that WSPDs only have sub-quadratic size in special cases.

[2] An exception is the k-closest-pairs problem, which can be solved using the WSPD, and where disjointness is used.

In $\mathbb{R}^d$, we improve the size bound to $O(d^3 \frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$. The new construction is again an $\frac{1}{\varepsilon}$-SSPD. The construction improves the number of pairs in SSPD from the old bound of $O(n/\varepsilon^d)$ to the stated bound, which has only polynomial dependency on the dimension d instead of exponential. While the new ABC/SSPD has a compact representation, it has two drawbacks: (i) it is a cover, not a disjoint cover, of the pairs, and (ii) the weight of the decomposition can be quadratic in the worst case.

(II) WSPD in doubling metrics. In Section 4, we show that the classical algorithm using net-trees [20] for constructing a WSPD in metric spaces with bounded doubling dimension is nearly optimal. More specifically, for a point set P in a metric space with constant doubling dimension and $\varepsilon \in (0,1)$ we show that the computed WSPD has size $O(\text{opt})$, where opt is the size of the minimum-size WSPD of P, and the runtime of the algorithm is in $O(\text{opt} + n \log n)$, making its runtime optimal output-sensitive.[3] This is in contrast to the best known runtime of the algorithm of $O(\varepsilon^{-d} n + n \log n)$.

(III) Computational hardness. In Section 5, we study the hardness of minWSPD. We show that for general metric spaces, computing the minimum size 3-WSPD is NP-Hard (Theorem 28). Furthermore, a simple padding of this construction shows that the problem is polynomially hard to approximate (Lemma 29). In the plane, we show that deciding if a $\frac{1}{\varepsilon}$-WSPD has a prespecified size is NP-Hard, see Theorem 33 (here ε is quite small, roughly $1/4n$).

(IV) Improvements in one dimension. In Section 6, we study the special case of $P \subset \mathbb{R}$, where well-separation becomes an interval condition. Our first algorithm is to construct a SetCover instance and to use known PTAS algorithms to obtain a $(1+\delta)$-approximation algorithm with runtime $n^{O(1/\delta^2)}$. Our second algorithm is an easy-to-implement sweepline algorithm that is a 3-approximation and runs in $O(n \log n)$ time. Finally, when requiring the separated pairs to be disjoint (PartitionWSPD), we give a constant approximation.

We implemented six WSPD algorithms for points in $\mathbb{R}$, and experimentally compared the sizes of the outputs. In particular, we compared solutions for CoverWSPD and PartitionWSPD to understand the impact of the disjointness requirement on the size. Our results show that the difference between the two variants is typically very small. In addition, we compared our 3-approximation with IP-based solutions. Although the 3-approximation generates larger decompositions, the algorithm is significantly faster and much easier to implement, making it an attractive practical alternative. The experimental results are given in the full version.

Comparison to previous work. Computing a WSPD with instance-specific guarantees has been considered by Har-Peled *et al.* [21]. Our result differs in that our $\frac{1}{\varepsilon}$-WSPD is a constant factor approximation compared to the minimum $\frac{1}{\varepsilon}$-WSPD, whereas Har-Peled *et al.* [21] construction size is proportional to the minimum $\frac{33}{\varepsilon}$-WSPD.

---

[3] It is known that computing the WSPD requires $\Omega(n \log n)$ time due to a reduction to Nearest Pair.

2 Preliminaries

2.1 Metrics, Distances, Balls and Spread

A ***finite metric space*** $\mathcal{X}$ is a pair $(\mathcal{X}, d)$, where $\mathcal{X}$ is the ground set and $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ is a ***metric*** satisfying the following conditions for all $x, y \in \mathcal{X}$: (i) $d(x, y) = 0$ if and only if $x = y$, (ii) $d(x, y) = d(y, x)$, and (iii) $d(x, y) + d(y, z) \geq d(x, z)$.

For a set $P \subseteq \mathcal{X}$, its ***diameter*** is $\nabla(P) = \max_{x,y \in P} d(x, y)$.

The ***distance*** between two sets $X, Y \subseteq \mathcal{X}$ is $d(X, Y) = d_{\min}(X, Y) = \min_{x \in X, y \in Y} d(x, y)$. The ***maximum distance*** between the two sets is $d_{\max}(X, Y) = \max_{x \in X, y \in Y} d(x, y)$. (As $\mathcal{X}$ is finite, all such minima and maxima exist.)

▶ **Definition 2.** *For a metric space $\mathcal{X}$, and a point $c \in \mathcal{X}$, let $b(c, r) = \{p \in \mathcal{X} \mid d(c, p) \leq r\}$ denote the* **ball** *of radius r centered at c.*

▶ **Definition 3.** *For a set $P \subseteq \mathcal{X}$, its* **closest pair distance** *is $\mathrm{cp}(P) = \min_{x,y \in P : x \neq y} d(x, y)$. The* **spread** *of P is $\Psi(P) = \nabla(P)/\mathrm{cp}(P)$.*

▶ **Definition 4.** *For two sets $X, Y \subseteq \mathcal{X}$, let $X \otimes Y = \{\{x, y\} \mid x \in X, y \in Y, x \neq y\}$. In particular, let $\binom{X}{2} = X \otimes X$ be the set of all (unordered) pairs of X.*[4]

2.2 Different Notions of Separation

These different notions of separation are illustrated in Figure 1.

▶ **Definition 5.** *A pair of sets $\{X, Y\}$ of points in a metric space $(\mathcal{X}, d)$ is*

$\frac{1}{\varepsilon}$**-separated**	*if*	$\max\bigl(\nabla(X), \nabla(Y)\bigr) \leq \varepsilon d(X, Y)$, *and*
$\frac{1}{\varepsilon}$**-semi-separated**	*if*	$\min\bigl(\nabla(X), \nabla(Y)\bigr) \leq \varepsilon d(X, Y)$.

A $\frac{1}{\varepsilon}$-well-separated pair is also $\frac{1}{\varepsilon}$-semi-separated (see Figure 1 for an example demonstrating these concepts).

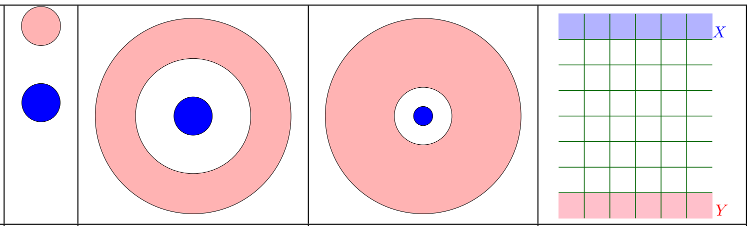
▶ **Definition 6.** *The* **stability** *of a pair $p = \{X, Y\}$ is*

$$\mathsf{S}(X, Y) = \frac{d_{\max}(X, Y) - d_{\min}(X, Y)}{2 d_{\min}(X, Y)}.$$

The pair p is ε**-stable** *if $\mathsf{S}(X, Y) \leq \varepsilon$.*

▶ Remark 7. If $p = \{X, Y\}$ is ε-stable, then $\frac{d_{\max}(X,Y)}{d_{\min}(X,Y)} = 1 + 2\mathsf{S}(X, Y) \leq 1 + 2\varepsilon$. Namely, all pairwise distances in such a pair are roughly the same up to a factor of $1 + 2\varepsilon$. On the other hand, if the pair p is $\frac{1}{\varepsilon}$-separated, then $d_{\max}(X, Y) \leq \nabla(X) + \nabla(Y) + d(X, Y) \leq (1 + 2\varepsilon) d(X, Y)$, and then $\mathsf{S}(X, Y) \leq \varepsilon$, implying that p is ε-stable. Note, that an $\frac{1}{\varepsilon}$-semi-separated pair is not necessarily ε-stable, see Figure 1.

[4] We use $\otimes$ instead of the Cartesian product $\times$, as the elements of $X \otimes Y$ are unordered pairs, whereas $\times$ would yield ordered pairs.



1-well-separated	✓	✗	✗	✗
1-semi-separated	✓	✓	✓	✗
1-stable	✓	✓	✗	✓

Figure 1 Rightmost example, if the sidelength of the grid is 1, then $m = d_{\min}(X, Y) = 6$, $M = d_{\max}(X, Y) = 10$, and $\nabla(X) = \nabla(Y) = \sqrt{37} \approx 6.08$. Thus, the pair $\{X, Y\}$ is $\frac{1}{3}$-stable, as $\frac{M-m}{2m} = \frac{1}{3}$.

2.3 Pair Decompositions

▶ **Definition 8.** *For a point set $P \subseteq \mathcal{X}$, a* **pair decomposition** *of P is a set of pairs*

$$\mathcal{W} = \left\{\{X_1, Y_1\}, \ldots, \{X_s, Y_s\}\right\}, \qquad \text{such that}$$

(I) $X_i, Y_i \subset P$ *for every i,*
(II) $X_i \cap Y_i = \emptyset$ *for every i, and*
(III) $\bigcup_{i=1}^{s} X_i \otimes Y_i = \binom{P}{2}$.

▶ **Definition 9.** *For a point set P, a* **well-separated pair decomposition** *of P with parameter $1/\varepsilon$, denoted by $\frac{1}{\varepsilon}$-WSPD, is a pair decomposition $\mathcal{W} = \left\{\{X_i, Y_i\}\right\}_{i=1}^{s}$ of P, such that, for all i, the sets X_i and Y_i are $\frac{1}{\varepsilon}$-separated. The* **size** *of the WSPD $\mathcal{W}$ is the number of pairs in the decomposition (i.e., s).*

The WSPD $\mathcal{W}$ is a PartitionWSPD if every pair of $\binom{P}{2}$ is covered exactly once by the pairs of $\mathcal{W}$. Otherwise, it is a CoverWSPD.

▶ **Definition 10.** *For a pair decomposition $\mathcal{W} = \left\{\{X_i, Y_i\}\right\}_{i=1}^{s}$ its* **weight** *is $\mathrm{w}(\mathcal{W}) = \sum_i (|X_i| + |Y_i|)$.*

It is known that the weight of any pair decomposition of a set of n points is $\Omega(n \log n)$ [18, Lemma 3.31]. In fact, Bollobás and Scott [7] determined this minimum exactly: writing $n = 2^k + \ell$ with $0 \le \ell < 2^k$, the minimum weight of a pair decomposition of n points equals $nk + 2\ell$. In the worst case, the weight of a WSPD of n points is $\Omega(n^2)$.

▶ **Definition 11.** *A pair decomposition $\mathcal{W}$ of a point set P is* **semi-separated pair decomposition ($\frac{1}{\varepsilon}$-SSPD)** *of P with parameter $1/\varepsilon$, if all pairs in $\mathcal{W}$ are $\frac{1}{\varepsilon}$-semi-separated.*

Surprisingly, one can construct $\frac{1}{\varepsilon}$-SSPD with total weight $O(\varepsilon^{-d} n \log n)$ with $O(\varepsilon^{-d} n)$ pairs in $\mathbb{R}^d$ [3]. Similar bounds are known for doubling metrics [4].

2.4 Doubling Metrics

The ***doubling constant*** of a metric space $\mathcal{X}$ is the maximum, over all balls b in the metric space $\mathcal{X}$, of the number of balls needed to cover b, using balls with half the radius of b. The logarithm of the doubling constant is the ***doubling dimension*** of the space. The doubling dimension is a generalization of the Euclidean dimension, as $\mathbb{R}^d$ has $\Theta(d)$ doubling dimension. Both the doubling constant and the doubling dimension are usually not known precisely.

2.5 Constructing WSPDs via SetCover

Given a set P of n points, one possible approach is to compute a set $\mathcal{F}$ of all maximal well-separated pairs in $\binom{P}{2}$ i.e., if $\{X, Y\} \in \mathcal{F}$, then (i) $X, Y \subseteq P$, (ii) $\{X, Y\}$ are well-separated, and (iii) there is no pair $\{X', Y'\} \in \mathcal{F}$, such that $X \subseteq X'$ and $Y \subseteq Y'$. A pair $\mathfrak{p} = \{X, Y\} \in \mathcal{F}$ covers all the pairs of points in $E_{\mathfrak{p}} = X \otimes Y$. Thus, the problem of finding a minWSPD is equivalent to solving the minimum SetCover problem for the instance $\left(\binom{P}{2}, \{E_{\mathfrak{p}} \mid \mathfrak{p} \in \mathcal{F}\}\right)$.

We can compute greedily an $\Theta(\log |\mathcal{F}|)$-approximation (in polynomial time) by adding at each step the pair in $\mathcal{F}$ that covers the maximum number of uncovered pairs in $\binom{P}{2}$ [26]. It is known that SetCover cannot be efficiently approximated within a factor of $(1 - o(1)) \ln n$, unless P = NP [23]. If one can limit the size of $\mathcal{F}$ to be polynomial in n, this readily leads to $O(\log n)$ approximation of minWSPD.

Since $|\mathcal{F}| = O(n^2)$ for $d = 1$ (see Section 6 for details), greedy SetCover yields an $O(\log n)$-approximation for computing a minimum $\frac{1}{\varepsilon}$-WSPD for a point set on the real line for any $\varepsilon > 0$. For higher dimensions, no polynomial bound on $|\mathcal{F}|$ is known.

Naturally, one can restrict the sets used in building the pairs to be of a certain (smaller) family of sets. For example, in two dimensions, consider all the subsets of P of the form $P \cap \square$ where $\square$ is some square. Then, one can solve/approximate the resulting SetCover problem. However, even in two dimensions and squares, the resulting discrete SetCover problem corresponds to a set of points in four dimensions, which one has to cover using a set of axis-aligned boxes. To the best of our knowledge, no better than $O(\log n)$ approximation is known in this case.

3 Approximate Biclique Cover

▶ **Definition 12.** *For a parameter $\varepsilon \in (0, 1)$, a pair decomposition of P of the form $\mathcal{W} = \left\{\{X_i, Y_i\}\right\}_{i=1}^{s}$ is an **ε-approximate biclique cover** (**$\frac{1}{\varepsilon}$-ABC**), if all its pairs are ε-stable, see Definition 6.*

In other words, a $\frac{1}{\varepsilon}$-ABC is a biclique cover of the metric graph of P, where for all i, the graph $(X_i \cup Y_i, X_i \otimes Y_i)$ is a biclique, where all the edges have roughly the same length, where an edge xy is assigned the length $d(x, y)$.

Specifically, we construct an $\frac{1}{\varepsilon}$-ABC with $O(\frac{n}{\varepsilon} \log n)$ size, that is also an $\frac{1}{\varepsilon}$-SSPD. Note that WSPDs only have sub-quadratic size in special cases.

3.1 Constructing ABC in the General Metric Case

▶ **Definition 13.** *A **ring** centered at p with radii $r < R$ is the set $\circledcirc(p, r, R) = b(p, R) \setminus b(p, r)$.*

▶ **Theorem 14.** *Given a set P of n points in a metric space $(\mathcal{X}, d)$, and a parameter $\varepsilon \in (0, 1/2)$, one can construct a $\frac{1}{\varepsilon}$-ABC of P with $O(\frac{n}{\varepsilon} \log n)$ pairs.*

Proof. The ABC is constructed inductively. If $|P| = O(1)$ then $\mathcal{W} = \binom{P}{2} = \left\{\{p, q\} \mid p, q \in P\right\}$. Otherwise, let p, q be the closest pair of points in P, and let $Q = P \setminus \{p\}$. By induction, the point set Q admits a $\frac{1}{\varepsilon}$-ABC cover, say $\mathcal{W}$. It remains to add the missing pairs $\{p\} \otimes P$ to $\mathcal{W}$. To this end, let

$$M = 4 + \lceil (32/\varepsilon) \ln n \rceil$$

and $\ell = d(p,q)$. Let $P_i = \circledcirc(p, r_i, r_{i+1}) \cap P$, where $r_i = \ell(1+\varepsilon/8)^{i-2}$, for $i = 1, \ldots, M$. We add the pair $p_i = \{\{p\}, P_i\}$, for $i = 1, \ldots, M$, to $\mathcal{W}$. These pairs cover all the pairs of points $x \in P$ such that $\ell \le d(p,x) \le n^4\ell$. Observe that the stability of a pair p_i is

$$\mathsf{S}(p_i) \le \frac{r_{i+1} - r_i}{2r_i} = \frac{(1+\varepsilon/8)^{i-1} - (1+\varepsilon/8)^{i-2}}{2(1+\varepsilon/8)^{i-2}} = \frac{\varepsilon}{16}.$$

For the remaining points $f \in P_{\mathrm{far}} = P \setminus b(p, \ell n^4)$, consider the pair qf (as a proxy for pf). It must be covered by an existing pair $\{X, Y\} \in \mathcal{W}$, where (say) $q \in X$ and $f \in Y$. Add p to the set X. This process is repeated for all the points in P_{far}. Let $\mathcal{W}'$ be the resulting pair decomposition of P, which we claim is the desired $\frac{1}{\varepsilon}$-ABC.

Consider a pair $p_0 = \{X_0, Y_0\}$ when it was first created. Initially, p_0 is $\varepsilon/16$-stable, by construction, and let S_0 denote its stability. Let $\Delta_0 = d_{\max}(X_0, Y_0)$. Assume this pair evolved by the construction inserting points into it on either side, creating a sequence of pairs $p_0 = \{X_0, Y_0\}, \ldots, p_t = \{X_t, Y_t\}$, where $t \le n$. Consider when the ith point p_i is added to the evolving pair $p_{i-1} = \{X_{i-1}, Y_{i-1}\}$ to create the new pair p_i, by (say) adding the point p_i to X_{i-1}. This was done because there is a point $q_i \in X_{i-1}$ and $f_i \in Y_{i-1}$, such that

$$d(p_i, q_i) n^4 \le d(q_i, f_i) \le \Delta_{i-1} = d_{\max}(X_{i-1}, Y_{i-1}).$$

Thus, for $i = 1, \ldots, t$, we have

$$\begin{aligned}\Delta_i &= d_{\max}(X_i, Y_i) \le \Delta_{i-1} + d(p_i, q_i) \le (1 + 1/n^4)\Delta_{i-1} \le (1+1/n^4)^n \Delta_0 \\ &\le (1 + 2/n^3)\Delta_0.\end{aligned}$$

A similar argument implies that $d(X_i, Y_i) \ge (1 - 4/n^3) d(X_0, Y_0)$. A straightforward argument now shows that $\mathsf{S}(X_t, Y_t) \le \varepsilon/8$. Thus, all the pairs computed are $\varepsilon/8$-stable, and the resulting decomposition is $\frac{1}{\varepsilon}$-ABC.

We claim that the number of pairs generated is at most $f(n) = \frac{c}{\varepsilon} n \ln n$ for some sufficiently large constant c. By induction, the number of pairs in $\mathcal{W}$ is at most $f(n-1)$, and the number of new pairs added to it to get $\mathcal{W}'$ is M. For the resulting decomposition, the number of pairs is bounded by

$$f(n-1) + M \le \frac{c}{\varepsilon}(n-1)\ln(n-1) + 4 + \left\lceil \tfrac{32}{\varepsilon} \ln n \right\rceil \le \frac{c}{\varepsilon} n \ln n = f(n),$$

for c sufficiently large. ◀

▶ Remark. We do not currently have a matching lower bound to Theorem 14, and closing the gap is an interesting open problem.

▶ Remark 15. Observe that all the pairs of Theorem 14 start as ring sets, with one side being a single point. The sets in the pair get bigger by points being added to either side of this pair. The above argumentation implies that, at the end of the process, the diameter of the set that started as a singleton in the pair would still be at most ε-fraction of the distance between the two pairs. Namely, the ABC constructed in this case is also a $\frac{1}{\varepsilon}$-SSPD.

Sadly, the resulting ABC/SSPD can have quadratic weight in the worst case. Indeed, consider a 1-ABC $\mathcal{W}$ for the point set $P = \{6^i \mid i = 1, \ldots, n\} \subseteq \mathbb{R}$. If there is a pair $\{X, Y\} \in \mathcal{W}$, such that $|X| > 1$ and $|Y| > 1$, then assume Y contain the largest number 6^y that appear in both sets, by disjointness of X, Y, we have $6^y \notin X$. Observe that the closest number to 6^y, that is smaller than it, in P is 6^{y-1}. Thus, we have $d_{\max}(X, Y) \ge 6^y - 6^{y-1} = 5 \cdot 6^{y-1}$. Observe, that Y contains at least one other number 6^t, for $t < y$. Thus, we have $d_{\min}(X, Y) \le |\max(X) - 6^t| \le 6^{y-1}$. We thus have that

$$\mathsf{S}(X_i, Y_i) = \frac{d_{\max}(X, Y) - d_{\min}(X, Y)}{2 d_{\min}(X, Y)} \ge \frac{5 \cdot 6^{y-1} - 6^{y-1}}{2 \cdot 6^{y-1}} = 2.$$

Namely, such a pair can not be 1-stable. Thus, all pairs in $\mathcal{W}$ must have at least one of their sets as a singleton. But the weight of such a pair is the number of original point-pairs it covers (plus one). Since overall there are $\binom{n}{2}$ pairs, the weight of $\mathcal{W}$ is at least $\binom{n}{2}$.

▶ **Corollary 16.** *Given a set P of n points in a metric space $(\mathcal{X}, d)$, and a parameter $\varepsilon \in (0, 1/2)$, one can construct a pair decomposition of P with $O(\frac{n}{\varepsilon} \log n)$ pairs, that is both $\frac{1}{\varepsilon}$-ABC and $\frac{1}{\varepsilon}$-SSPD.*

▶ **Observation 17.** *An easy inductive argument shows that the resulting ABCs/SSPDs of Theorem 14 are disjoint – every pair is covered exactly once by the cover computed.*

▶ Remark 18. The construction of Theorem 14 can be implemented in $O(n^2 \log n)$ time, as follows. Compute all pairwise distances and sort them, in $O(n^2 \log n)$ time. Since deleting points can only increase the closest-pair distance, the closest pairs of all n levels of the induction can be extracted by a single scan of this sorted list. At each level, assigning a point to its ring takes constant time. Finally, maintain a table storing, for every pair of points, the pair of the decomposition covering it. Handling a far point is then an $O(1)$ time lookup, and since the decomposition is disjoint, every entry of this table is written at most once, for a total of $O(n^2)$ time. As the input is a distance matrix of size $\Theta(n^2)$, the running time is near-linear in the input size.

High-dimensional data often has a low spread (i.e., the spread is a small constant). This is only possible in high dimensions, as the spread of n points in $\mathbb{R}^d$ is $\Omega(n^{1/d})$. By using only the rings in the construction of Theorem 14, we readily get the following.

▶ **Corollary 19.** *Given a set P of n points in a metric space $(\mathcal{X}, d)$, with spread $\Psi = \Psi(P)$, one can construct a pair decomposition of P of size $O\big(\frac{n}{\varepsilon} \log \Psi\big)$, that is both $\frac{1}{\varepsilon}$-ABC and $\frac{1}{\varepsilon}$-SSPD.*

3.2 Construction ABC in the Euclidean Case

In $\mathbb{R}^d$, we improve the size bound to $O(d^3 \frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$. The new construction is again an $\frac{1}{\varepsilon}$-SSPD. The construction improves the number of pairs in SSPD from the old bound of $O(n/\varepsilon^d)$ to the stated bound, which has only polynomial dependency on the dimension d instead of exponential.

▶ **Theorem 20.** *For a set P of n points in $\mathbb{R}^d$, and $\varepsilon \in (0, 1)$, one can compute a pair decomposition $\mathcal{W}$ of size $O(\frac{d^3 n}{\varepsilon} \log \frac{d}{\varepsilon})$. The decomposition $\mathcal{W}$ is $\frac{1}{\varepsilon}$-ABC and $\frac{1}{\varepsilon}$-SSPD.*

The construction realizing Theorem 20, and its proof, are given in Appendix A. The construction computes $O(d \log \frac{d}{\varepsilon})$ compressed quadtrees, and then extracts the pairs, in their implicit representation, in time proportional to their number. The overall running time is $O\big(d^2 n \log \frac{d}{\varepsilon} (\log n + \frac{d}{\varepsilon})\big)$.

4 Constant Approximation in Doubling Metrics

4.1 Preliminaries

Let P be a set of n points in a metric space $(\mathcal{X}, d)$, with bounded doubling dimension dim. A standard construction of WSPDs in doubling metrics is by Har-Peled and Mendel [20]. It is an adaptation of the construction in Euclidean spaces by Callahan and Kosaraju [9]. The first step is constructing a hierarchical tree over the points, similar in spirit to quadtrees.

▶ **Definition 21.** *A* **net-tree** *of P is a tree T storing the points of P in its leaves. For a node $v \in T$, $P_v \subset P$ is the set of leaves in the subtree of v. Each node v has an associated representative point $\rho(v) \in P_v$. Internal nodes have at least two children. Each vertex v has a* **level** *$L(v) \in \mathbb{Z} \cup \{-\infty\}$. The levels satisfy $L(v) < L(\overline{\mathrm{p}}(v))$, where $\overline{\mathrm{p}}(v)$ is the parent of v in T. The levels of the leaves are $-\infty$.*

We require the following properties from T:

(I) Covering. *For all $v \in T$: $\mathcal{b}\left(\rho(v), \frac{2\tau}{\tau-1} \cdot \tau^{L(v)}\right) \supset P_v$. where $\tau = 11$.*

(II) Packing. *For all non-root $v \in T$: $\mathcal{b}\left(\rho(v), \frac{\tau-5}{2(\tau-1)} \cdot \tau^{L(\overline{\mathrm{p}}(v))-1}\right) \bigcap P \subset P_v$.*

(III) Inheritance. *For all nonleaf $u \in T$, there exists a child $v \in T$ of u such that $\rho(u) = \rho(v)$.*

Net-trees are an abstraction of compressed quadtrees, as the levels of a node and its child can be dramatically different. Uncompressed quadtree corresponds to net-tree where the levels of consecutive nodes along a path differ by exactly one (except the leaves). A net-tree where all the levels are consecutive is uncompressed.

▶ **Theorem 22** [20]**.** *Given a set P of n points in $\mathcal{X}$, one can construct a net-tree for P in $2^{O(\dim)} n \log n$ expected time, where $\dim$ is the doubling dimension of $\mathcal{X}$.*

Once such a net-tree is available, computing the WSPD is relatively straightforward.

▶ **Lemma 23** [20]**.** *Let P be a set of n points in a metric space $\mathcal{X}$. For $\varepsilon \in (0,1)$, one can construct a $\frac{1}{\varepsilon}$-WSPD of size $n\varepsilon^{-O(\dim)}$, and expected construction time is $2^{O(\dim)} n \log n + n\varepsilon^{-O(\dim)}$. Furthermore, the pairs of the WSPD are of the form $\{P_u, P_v\}$, where u, v are vertices of a net-tree of P, such that $\nabla(P_u), \nabla(P_v) \leq \varepsilon \mathcal{d}(\rho(u), \rho(v))$.*

Proof Sketch. The algorithm computes a net-tree T using Theorem 22. Then, the algorithm maintains a FIFO queue of pairs of nodes $\{u, v\}$ that it has to separate. Initially, the queue consists of a single pair, that is, $\{\mathrm{root}(T), \mathrm{root}(T)\}$. The algorithm repeatedly considers the pair $\mathcal{p} = \{u, v\}$ at the head of the queue, and checks if the pair is separated. For a pair $\mathcal{p}$, define its ***level*** to be $L(\mathcal{p}) = \max(L(u), L(v))$. The pair $\mathcal{p}$ is defined to be separated if $8\frac{2\tau}{\tau-1} \cdot \tau^{L(\mathcal{p})} \leq \varepsilon \cdot \mathcal{d}(\rho(u), \rho(v))$. In other words, the distance between the representatives $\rho(u)$ and $\rho(v)$ is large enough to separate (the balls covering) P_u and P_v. Now, if the pair $\mathcal{p}$ is separated, then the pair $\{P_u, P_v\}$ is in the output. Otherwise, let $L(u) > L(v)$. The algorithm splits the bigger pair u by replacing $\mathcal{p}$ in the queue with the pairs $C_u \otimes \{v\}$, where C_u is the set of children of v.

The output pairs have the form $\{P_u, P_v\}$. One can verify that every pair of points is covered by a well-separated pair $\{P_u, P_v\}$ in the output. Moreover, these pairs satisfy $\nabla(P_u), \nabla(P_v) \leq \varepsilon \mathcal{d}(\rho(u), \rho(v))$, since by the covering property (I) of the net-tree, $\nabla(P_u), \nabla(P_v) \leq \frac{\varepsilon}{4} \mathcal{d}(P_u, P_v)$, as $P_u \subseteq \mathcal{b}(\rho(u), r)$ and $P_v \subseteq \mathcal{b}(\rho(v), r)$, where $r = \frac{2\tau}{\tau-1} \cdot \tau^L$.

The standard charging argument for WSPDs yields a size bound of $n/\varepsilon^{O(\dim)}$ pairs in the WSPD. In particular, if $L(u) > L(v)$, we charge the pair $\{P_u, P_v\}$ to the parent of v in the net-tree, and show using a packing argument in the doubling metric that each node $v \in T$ is charged at most $\varepsilon^{-O(\dim)}$ times. For details, see [20]. ◀

4.2 Output-Sensitive Runtime

For the runtime analysis, we improve on [20] by providing an output-sensitive analysis. Let $\mathcal{W}$ be the $\frac{1}{\varepsilon}$-WSPD computed by the algorithm. Consider a history tree H of the computation of the WSPD, having the pairs being created by the algorithm as nodes. Every internal node in this tree has two or more children, and every leaf corresponds to a pair being output. Further, at most $O(n)$ superfluous leaves correspond to self-pairs $\{u, u\}$, with $u \in T$, and u being a leaf.

Overall, the size of H is $N = O(n+|\mathcal{W}|)$. A careful inspection of the algorithm (see Lemma 23) reveals that its overall runtime is $2^{O(\dim)} n \log n + 2^{O(\dim)} |H| = 2^{O(\dim)}(n \log n + |\mathcal{W}|)$. We state the output-sensitive runtime as a corollary.

▶ **Corollary 24.** *The WSPD algorithm in Lemma 23 runs in* $2^{O(\dim)}(n \log n + |\mathcal{W}|)$ *expected time, where* $\mathcal{W}$ *is the computed* $\frac{1}{\varepsilon}$*-WSPD.*

4.3 Approximation Factor for minWSPD

▶ **Lemma 25.** *The WSPD algorithm of Lemma 23 outputs* $\frac{1}{\varepsilon}$*-WSPD of size at most* $2^{O(\dim)} |\mathrm{opt}|$*, where* opt *is the optimal* $\frac{1}{\varepsilon}$*-WSPD for* P*.*

Proof. Let $\mathcal{W}$ be the WSPD output by the algorithm. It would be easier to argue here on the uncompressed net-tree. In this scenario, a pair $\{u, v\} \in \mathcal{W}$ has the property that $|L(u) - L(v)| \leq 1$. Note that running the algorithm on the uncompressed net-tree, or the original net-tree, would output *exactly* the same pairs. The difference is that in the compressed version, the algorithm can "slide" down a compressed edge in constant time, whereas the uncompressed version must laboriously go down the levels one by one.

Consider an "unbalanced" pair $\{u, v\}$ with $L(u) = L(v) + 1$ (i.e., u is larger than v). We replace this pair in $\mathcal{W}$ by the pairs $C_u \otimes \{v\}$, where C_u are the children u in T. We repeatedly do this replacement till all pairs are balanced. This replacement process increases, by a factor of at most $2^{O(\dim)}$, the size of $\mathcal{W}$, as this bounds the maximum degree in T. Let $\mathcal{W}'$ be the resulting $\frac{1}{\varepsilon}$-WSPD.

Write $\mathcal{W}' = \{\mathscr{p}_1, \ldots, \mathscr{p}_m\}$, where each $\mathscr{p}_i = \{u_i, v_i\}$ is a pair of net-tree nodes, and let the optimal $\frac{1}{\varepsilon}$-WSPD be $\{\mathscr{o}_1, \ldots, \mathscr{o}_k\}$, with $\mathscr{o}_j = \{X_j, Y_j\}$. We ***register*** each pair $\mathscr{p} = \{u, v\} \in \mathcal{W}'$ with the (assumed unique) optimal pair $\mathscr{o}_j = \{X_j, Y_j\}$ for which $\rho(u) \in X_j$ and $\rho(v) \in Y_j$. Since every pair of $\mathcal{W}'$ is registered with exactly one optimal pair, it suffices to show that each optimal pair has at most $2^{O(\dim)}$ pairs of $\mathcal{W}'$ registered with it. This yields $|\mathcal{W}'| \leq 2^{O(\dim)} k = 2^{O(\dim)} |\mathrm{opt}|$, and hence $|\mathcal{W}| \leq |\mathcal{W}'| \leq 2^{O(\dim)} |\mathrm{opt}|$.

So fix an optimal pair $\mathscr{o} = \{X, Y\}$, and let $\mathscr{p}_1, \ldots, \mathscr{p}_t$ be the pairs of $\mathcal{W}'$ registered with it. These occupy only two consecutive levels, $\mu - 1$ and μ, for some integer μ. Indeed, otherwise let $\mathscr{p}, \mathscr{p}'$ be the registered pairs of maximum and minimum level, so that $L(\mathscr{p}) > L(\mathscr{p}') + 1$. Writing $\mathscr{p}' = \{u', v'\}$, a tedious but straightforward argument then shows that the pair $\{\overline{\mathrm{p}}(\rho(u')), \overline{\mathrm{p}}(\rho(v'))\}$ would already be separated by the algorithm, so $\mathscr{p}'$ would never be output, a contradiction.

It remains to bound the number of registered pairs on these two levels. Let Q be the set of representatives $\rho(u)$ over all registered pairs $\mathscr{p} = \{u, v\}$ of level $\mu - 1$. As each such pair is registered with $\mathscr{o}$, we have $\rho(u) \in X$, and so $Q \subseteq X$. Put $\Delta = \nabla(X \cup Y)$. Then $\nabla(Q) \leq \nabla(X) \leq \varepsilon\Delta$, while the packing property of the net-tree forces the points of Q to be at pairwise distance $\Omega(\varepsilon\Delta)$. By the doubling dimension, $|Q| \leq 2^{O(\dim)}$. The same bound holds for the representatives on the Y side and for the pairs of level μ. As a registered pair is determined by its two representatives, at most $2 \cdot 2^{O(\dim)} \cdot 2^{O(\dim)} = 2^{O(\dim)}$ pairs of $\mathcal{W}'$ are registered with $\mathscr{o}$, as claimed. ◀

4.4 The Result

▶ **Theorem 26.** *Let* P *be a set of* n *points in a metric space* $\mathcal{X}$ *with doubling dimension* $\dim$*. For* $\varepsilon \in (0, 1)$*, the algorithm of Lemma 23 constructs a* $\frac{1}{\varepsilon}$*-WSPD of size* $2^{O(\dim)} |\mathrm{opt}|$*, and in* $2^{O(\dim)}(|\mathrm{opt}| + n \log n)$ *expected time, where* opt *is the minimal sized* $\frac{1}{\varepsilon}$*-WSPD of* P*.*

5 Hardness of minWSPD

We prove two hardness results. For general metric spaces, we show that the problem is NP-Hard already for a *fixed* separation parameter (i.e., a 3-WSPD), and that it is even polynomially hard to approximate. In the Euclidean plane, we show NP-Hardness when the separation parameter is part of the input. Note that the second result does not subsume the first one. The Euclidean reduction requires a polynomially small separation parameter $\varepsilon \approx 1/4n$, whereas the result for general metric spaces holds for a constant separation.

5.1 Hardness of Arbitrary Metric Spaces

The reduction is from ECliqueCover.

▶ **Problem 27** (ECliqueCover)**.** *An instance is a pair $(\mathcal{G}, k)$, where $\mathcal{G}$ is a graph, and one has to decide if one can cover the edges of $\mathcal{G}$ by k cliques that are contained in $\mathcal{G}$.*

ECliqueCover is NP-Hard, and it is NP-Hard to approximate within a factor of $n^{1-\delta}$, for some $\delta \in (0, 1)$, see [22].

▶ **Theorem 28.** *Computing the (exact) minimum size 3-WSPD is NP-Hard on an arbitrary metric space.*

Proof. Consider an instance $(\mathcal{G}, k)$ of the ECliqueCover problem, with n vertices. Construct a metric space $(\mathcal{X}, d)$ with $\mathsf{V} = \mathsf{V}(\mathcal{G})$ as the initial set of elements. Let $d(u, v) = 1$ if $uv \in \mathsf{E}(\mathcal{G})$, and 2 otherwise. The metric space $(\mathcal{X}, d)$ is a 1/2 metric space. Add a new point p to $\mathcal{X}$, such that $d(p, v) = 3$ for all $v \in \mathcal{X}$. The finite metric space $\mathcal{X}$ has a 3-WSPD of size $\binom{n}{2} + k$ if and only if CliqueCover on G has a clique cover of size k.

Indeed, if $\mathcal{G}$ has a clique cover $C_1, \ldots, C_k$, then a corresponding 3-WSPD, of the state size, is $\mathcal{W} = \{\{p\}, \mathsf{V}(C_i)\}_{i=1}^{k} \cup \binom{\mathsf{V}}{2}$. The set $\mathcal{W}$ is a valid 3-WSPD for $\mathcal{X}$, as it covers all the pairwise points in $\mathcal{X}$, and the diameter of each C_i is one by construction, and thus, all the pairs constructed are 3-separated.

Conversely, let $\mathcal{W}$ be a 3-WSPD of size N. Any two elements $u, v \in \mathsf{V}$ have distance either 1 or 2. Thus, as $\varepsilon = 1/3$, all the pairs involving only points of V must be singletons, as otherwise they would not be 3-separated. Similarly, the "special" point p must be a singleton in any pair of $\mathcal{W}$ including it, as any set containing p and some other point of V has diameter 3, which is not 3-separable from any set of points in $\mathcal{X}$. Thus, the pairs in $\mathcal{W}$ can be broken into two types of pairs:

(I) Singleton pairs $\{\{x, y\}\}$, with $x, y \in \mathsf{V}$. There are exactly $\binom{n}{2}$ such pairs.

(II) Pairs $\mathcal{p}_i = \{\{p\}, \mathsf{V}_i\}$, for $i = 1, \ldots, t$, where $N = \binom{n}{2} + t$ and $\mathsf{V}_i \subseteq \mathsf{V}(\mathcal{G})$.

If the induced subgraph $\mathcal{G}_{\mathsf{V}_i}$ form a clique, then $\nabla(\mathsf{V}_i) = 1$, and the pair $\mathcal{p}_i$ is 3-separated. If V_i is not a clique then $\nabla(\mathsf{V}_i) = 2$, and then $\mathcal{p}_i$ is not 3-separated. We conclude that $\mathcal{W}$ encodes a $N - \binom{n}{2}$ clique cover for $\mathcal{G}$.

Thus, $\mathcal{X}$ has a 3-WSPD of size $\binom{n}{2} + k \iff \mathcal{G}$ has a clique cover of size k. ◀

▶ **Lemma 29.** *For a finite metric space $\mathcal{X}$ defined over a set of n points, for $\delta \in (0, 1)$ sufficiently small,* no *n^{δ}-approximation to the size of the optimal 3-WSPD of $\mathcal{X}$ is possible, unless* $\mathrm{P} = \mathrm{NP}$.

Proof. Let us repeat the above proof, but use $\beta \gg n^2$ copies of the original graph. Given a graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$, with $n = |\mathsf{V}|$ and $m = |\mathsf{E}|$, consider the set $U = \mathsf{V} \times [\![\beta]\!]$, where β is an integer and $[\![\beta]\!] = \{1, \ldots, \beta\}$. Let $\mathcal{X} = (U, d)$ be a metric space, where for two points

$(x,i),(y,j) \in U$, we define their distance to be (i) 3 if $i \neq j$, (i) 1 if $xy \in \mathsf{E}$, and (iii) 2 if $i = j$ and $xy \notin \mathsf{E}$. Intuitively, we have β copies of the 1/2 metric from the previous proof, with distance 3 between the copies.

If $\mathcal{G}$ has an (edge) clique cover of size k, with cliques in the cover having the vertex sets $C_1, \ldots, C_k$, then setting $C_{\alpha,i} = C_\alpha \times \{i\}$, we have the 3-WSPD:

$$\mathcal{W} = \Big\{ \{C_{\alpha,i}, C_{\beta,j}\} \;\Big|\; i,j \in [\![\beta]\!], \alpha,\beta \in [\![k]\!], i < j \Big\} \cup \Big\{ \{\{(u,i)\},\{(v,i)\}\} \;\Big|\; i \in [\![\beta]\!], u,v \in [\![n]\!], u \neq v \Big\}.$$

The first part matches all the cliques between different layers, and the second part covers each layer's vertices with $\binom{n}{2}$ singleton pairs. Overall, this WSPD has size $\xi = \binom{\beta}{2}k^2 + \beta\binom{n}{2}$.

Assume that we had somehow computed a 3-WSPD $\mathcal{W}$ of $\mathcal{X}$ with $\alpha\xi$ pairs, for some $\alpha < k$ (here, the clique cover is not provided). A pair $\{X,Y\} \in \mathcal{W}$, must have the property that there are indices i,j, such that $X \subseteq \mathsf{V} \times \{i\}$ and $Y \subseteq \mathsf{V} \times \{j\}$, as otherwise $\nabla(X) \geq 3$ or $\nabla(Y) \geq 3$, which in either case would imply the pair can not be 3-separated as $\nabla(\mathcal{X}) = 3$.

Let $\mathcal{W}_{i,j}$ be all the pairs in $\mathcal{W}$ between subsets of $\mathsf{V} \times \{i\}$ and $\mathsf{V} \times \{j\}$. By averaging, there are indices i,j such that $|\mathcal{W}_{i,j}| \leq |\mathcal{W}| / \binom{\beta}{2} \leq \alpha k^2 + 1 \leq k^3$. Interpreting all sets of $\mathcal{W}_{i,j}$ covering $\mathsf{V} \times \{i\}$ as an edge clique cover of $\mathcal{G}$, implies that we had computed a k^3 cover of $\mathcal{G}$. To see why this is impossible, assume $k = O(n^{\delta/8})$, for some sufficiently small constant $\delta \in (0,1)$ (this can be ensured by adding a sufficiently large clique to the original graph $\mathcal{G}$). We thus had computed an k^2-approximate clique cover for $\mathcal{G}$, where $k^3/k = O(n^{\delta/4}) \lll O(n^{1-\delta})$. However, unless P = NP, no approximation of quality $n^{1-\delta}$ exists for edge clique cover (see [22]). ◀

5.2 NP-Hardness in the Euclidean Plane

5.2.1 Background

Given a metric space $(\mathcal{X}, d)$, a set of points $P \subseteq \mathcal{X}$, and a set of centers $C \subseteq \mathcal{X}$, the center clustering *price* of P by C is $d_{\max}(P,C) = \max_{p \in P} d(p,C) = \max_{p \in P} \min_{c \in C} d(p,c)$. Every point in P is within distance at most $d_{\max}(P,C)$ from its nearest center in C.

▶ **Definition 30.** *The* **k-center problem** *is to find a set $C \subseteq \mathcal{X}$ of k points, such that $d_{\max}(C,P)$ is minimized:* $\mathrm{opt}_k(P) = \min_{C \subseteq P, |C| = k} d_{\max}(P,C)$.

▶ **Definition 31** (PairwiseClustering)**.** *The* **k-pairwise clustering** *problem is to find a partition of P into k disjoint sets $\Pi = \{P_1, \ldots, P_k\}$, such that the diameter $\nabla(\Pi) = \max_{X \in \Pi} \nabla(X)$ is minimized. The optimal such diameter is denoted by*

$$\mathrm{opt}_{\nabla,k}(P) = \min_{\Pi \in [\![k]\!]^P} \nabla(\Pi),$$

where $[\![k]\!]^P$ denotes the set of all k disjoint partitions of P.

Gonzalez [16] presented a simple, $O(kn)$ time, 2-approximation algorithm to the optimal k-center clustering radius, and it is not hard to show that no better approximation exists in general metric spaces. The same algorithm provides a 2-approximation for the PairwiseClustering problem. In [14] is shown that in two dimensions, it is ≈ 1.822 hard to approximate the optimal k-center radius. For PairwiseClustering, they showed a slightly stronger hardness.

▶ **Theorem 32** [14, Theorem 2.1]**.** *PairwiseClustering, in the plane (under the L_2-norm) is,* NP-Hard *to approximate within a factor of $\alpha = 2\cos 10° \approx 1.969$.*

The proof of Feder and Greene [14] works via a reduction from VertexCover for planar graphs of degree at most 3. Specifically, given such a graph, they carefully replace its edges with long chains of points, and the vertices are carefully designed to merge such chains. In the resulting point set P (which looks "like" an embedding of the original graph), there is a parameter k, such that either P can be covered by k clusters of diameter 1, in which case the original graph has a vertex-cover of size k. Alternatively, any cover of the resulting point set by k clusters has PairwiseClustering diameter α (k here is quite large).

5.2.2 Reduction

▶ **Theorem 33.** *Computing the minimum size WSPD is* NP-HARD *for point sets in the Euclidean plane. Specifically, given a point set Q in the plane, and parameters $t \in \mathbb{N}$ and $\varepsilon \in (0,1)$, such that deciding if Q has a $\frac{1}{\varepsilon}$-WSPD of size at most t is* NP-HARD.

Proof. Let (P, k) be an instance of PairwiseClustering problem in the plane as generated by the reduction of Theorem 32. The task at hand is to decide if P can be partitioned into k clusters of diameter 1, or P has the property that any such partition has PairwiseClustering diameter α. Let $n = |P|$, and observe that by the construction of P, we have $\nabla(P) \leq 2n$. Note, that for $\varepsilon < 1/4n$, any $\frac{1}{\varepsilon}$-WSPD of P is an exhaustive list of all $\binom{n}{2}$ pairwise singletons.

The idea is to add a point p to P that is far away. Let p be a point in distance (say) $\ell = n/\delta$ from its closest point $p' \in P$, where $\delta \in (0,1)$ is a sufficiently small constant to be specified shortly. For any $X \subseteq P$, we have that its separation quality in the pair $\{\{p\}, X\}$ is $\frac{\nabla(X)}{d(p,X)}$, which is bounded in the range

$$(1-2\delta)\frac{\delta}{n}\nabla(X) \leq \frac{\nabla(X)}{(2+1/\delta)n} \leq \frac{\nabla(X)}{\ell+2n} \leq \frac{\nabla(X)}{d(p,X)} \leq \frac{\nabla(X)}{\ell} = \frac{\delta}{n}\nabla(X),$$

since $\frac{1}{2+1/\delta} = \frac{\delta}{1+2\delta} \geq \delta(1-2\delta)$.

Now, the gap in the PairwiseClustering diameter of P is between 1 and $\alpha \approx 1.969$. For $\delta = \min(1/4n, 10^{-5})$, a set $X \subseteq P$ of diameter 1, the corresponding pair $\{\{p\}, X\}$ has separation in the range $0.999\frac{\delta}{n}$ to $\tau = \frac{\delta}{n}$. On the other hand, if X has diameter α, then this pair has separation at least $0.999\frac{\delta}{n}\alpha > 1.96\tau$.

So, consider a $\frac{1}{\varepsilon}$-WSPD $\mathcal{W}$ of $Q = P \cup \{p\}$, for $\varepsilon = \tau = \min(1/4n, 10^{-5})/n$. This WSPD has $\binom{n}{2}$ singleton pairs to handle the points in P. All the pairs involving p must have p as a singleton on one side of the pair. The other side can have a diameter of at most 1 if we demand that it be $1/\tau$-separated. If X has diameter α or bigger, than the pair would not be $1/\tau$-separated. It follows, that if P has a k PairwiseClustering with k-clusters of diameter 1, then the corresponding $\frac{1}{\varepsilon}$-WSPD would be of size $\binom{n}{2} + k$. The implication also works in the other direction. If P has a minimum k-PairwiseClustering diameter α, then any ψ-WSPD of Q of size $\binom{n}{2} + k$, must have separation at least (say) $(0.99/\alpha)/\tau$. ◀

6 The One-Dimensional Problem

We further study the special case of $P \subset \mathbb{R}$. Our first algorithm constructs a SetCover instance and uses known PTAS algorithms to obtain a $(1+\delta)$-approximation algorithm with runtime $n^{O(1/\delta^2)}$.

Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in the real line $\mathbb{R}$, where $p_i < p_{i+1}$ for all i. As described in the introduction and in Section 2.5, we can reduce CoverWSPD to SetCover. We aim to cover the set $Q = \{(p_i, p_j) \mid i < j\}$ of all $\binom{n}{2}$ grid points strictly above the diagonal in the non-uniform grid $P \times P$.

A (maximal) well-separated pair for P, is a pair $\mathcal{p} = \{I, J\}$, where I, J are intervals on the line, and I and J are $\frac{1}{\varepsilon}$-well-separated. Formally, we require that $\ell(I), \ell(J) \leq \varepsilon d(I, J)$, where $\ell(I)$ denotes the length of I. We might as well pick I and J to be maximal while preserving the separation property. That is $\ell(I) = \ell(J) = \varepsilon d(I, J)$. In addition, if the two endpoints of I and J closest to each other are not points of P, we can move the two intervals apart, increasing their separation. Thus, for a fixed $\varepsilon > 0$, all the (maximal) separated pairs $\{I, J\} \in \mathcal{F}$ are uniquely defined by their two inner endpoints, which belong to P. Therefore, the size of $\mathcal{F}$ is $\binom{n}{2}$.

When we order the pair (I, J), denoting that $I < J$ (i.e., for all $x \in I$ and $y \in J$, we have $x < y$), the two inner points p_i, p_j form a point $u = (p_i, p_j)$ in Q. This gives a unique relation between $I \times J$ and the square $\mathsf{S}_u := [p_i - \tau, p_i] \times [p_j, p_j + \tau]$, where $\tau = \varepsilon(p_j - p_i)$, by the bottom-right point u. The square S_u is the ***shadow*** of u, and u is the ***anchor*** of S_u.

Thus, the minWSPD of P corresponds to a minimal set $\mathcal{C} \subseteq \{\mathsf{S}_u \mid u \in Q\}$ that covers the points of Q, which can be $(1+\delta)$-approximated in $n^{O(1/\delta^2)}$ time using local search [12, 24], as this is an instance of discrete set cover by pseudodisks. Thus, we get the following.

▶ **Lemma 34.** *Given a set P of n points in $\mathbb{R}$, and parameters $\varepsilon, \delta \in (0,1)$, one can compute, in $n^{O(1/\delta^2)}$ time, a $\frac{1}{\varepsilon}$-CoverWSPD of P with k pairs, where $k \leq (1+\delta)\mathrm{opt}$, where opt is the minimum number of pairs in any $\frac{1}{\varepsilon}$-CoverWSPD of P.*

Our second algorithm uses a sweep-line to obtain an easy-to-implement 3-approximation to the minimal CoverWSPD with runtime $O(n \log n)$.

As before, we interpret Q as the set of all points above the diagonal in the non-uniform grid $P \times P$, and search for a set of squares that covers Q. The algorithm uses the "One for the Price of Two" approach of Bar-Yehuda [6]. We compute a set of three squares that provides at least as much coverage as a single square in the optimal solution.

▶ **Definition 35.** *The **vicinity** $V(p)$ of a point $p \in Q$ is the set of all points $p' \in Q$, such that $p, p' \in \mathsf{S}_r$ for some $r \in \mathbb{R}^2$. That is, the vicinity of p is the region of all the points in the plane that might be covered by a single shadow square that also covers p.*

See Figure 2 for an example of a vicinity. We may reduce to $r \in Q$, because, if there exists a S_r with $p, p' \in \mathsf{S}_r$, then $r' := (\min(p_i, p'_i), \min(p_j, p'_j)) \in Q$ is an anchor of a shadow with $p, p' \in \mathsf{S}_{r'}$.

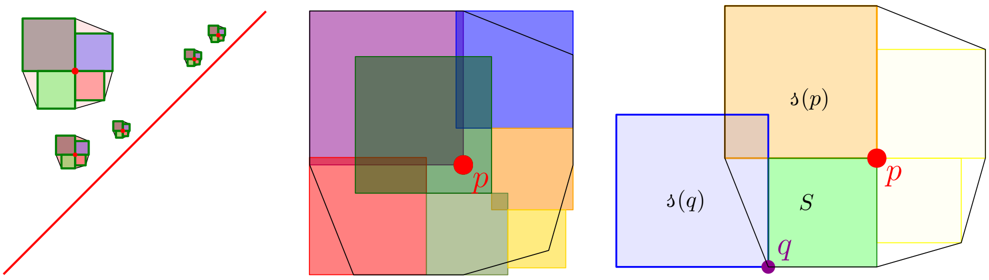


Figure 2 Left: A few points and their vicinities, for $\varepsilon = 0.4$. Middle: A covering of the vicinity with 7 squares. Right: The left half of the vicinity of a point p can be covered by three squares.

Our main idea is to sweep the points Q from right to left and cover the left half of the vicinity of a point with three squares (compare to Figure 2). Now, consider the optimal solution $\mathcal{O} \subseteq \mathcal{F}$, and consider a point $p \in Q$ not yet covered. The point p is covered by $\mathcal{O}$, and for all squares $\square \in \mathcal{O}$ that contain p, we have $\square \subset V(p)$ by definition. See Appendix B.1 for a detailed description of the algorithm and the proof of the following lemma.

▶ **Lemma 36.** *Given a set P of n points in $\mathbb{R}$, and parameter $\varepsilon > 0$, one can compute, in $O(n \log n)$ time, a cover $\frac{1}{\varepsilon}$-WSPD of P with k pairs, where $k \le 3\mathrm{opt}$, where opt is the minimum number of pairs in any cover $\frac{1}{\varepsilon}$-WSPD of P.*

Finally, when requiring the separated pairs to be disjoint (PartitionWSPD), we give a constant approximation (see Section B.2 for details).

▶ **Lemma 37.** *Given a set P of n points in $\mathbb{R}$, and a parameter $\varepsilon \in (0, 1)$, one can compute, in $n^{O(1)}$ time, a partition $\frac{1}{\varepsilon}$-WSPD of P with k pairs, where $k \le 10\mathrm{opt}$, where opt is the minimum number of pairs in any cover (or partition) $\frac{1}{\varepsilon}$-WSPD of P.*

7 Conclusions

We provided a number of results for approximating the minimum WSPD. Some open problems include: Is the minimum WSPD problem NP-Hard in $\mathbb{R}$? Is there a $(1 + \varepsilon)$-approximation algorithm, or a simple $O(1)$-approximation algorithm, in $\mathbb{R}^d$, for $d \ge 2$? What are the worst-case inputs for minWSPD in $\mathbb{R}^d$?

In Section 3, we introduced a new pair decomposition, an ABC, and showed that every finite metric space admits a near-linear-size ABC, which is also an SSPD. Euclidean spaces even admit a linear-size ABC and SSPD in high dimensions (the constant there is $O(d^3)$), and in low-dimensional Euclidean spaces, these decompositions have small implicit representations, similar to WSPD. Some open problems include: Are our bounds in Theorem 14 and Theorem 20 tight? Which metric spaces, other than $\mathbb{R}^d$, admit a linear-size ABC? We consider the existence of the ABC a surprising property that could lead to further structural and algorithmic insights.

References

1 Mohammad Ali Abam, Paz Carmi, Mohammad Farshi, and Michiel H. M. Smid. On the power of the semi-separated pair decomposition. *Comput. Geom.*, 46(6):631–639, 2013. doi:10.1016/j.comgeo.2013.02.003.

2 Mohammad Ali Abam, Mark de Berg, Mohammad Farshi, and Joachim Gudmundsson. Region-fault tolerant geometric spanners. *Discret. Comput. Geom.*, 41(4):556–582, 2009. doi:10.1007/s00454-009-9137-7.

3 Mohammad Ali Abam, Mark de Berg, Mohammad Farshi, Joachim Gudmundsson, and Michiel H. M. Smid. Geometric spanners for weighted point sets. *Algorithmica*, 61(1):207–225, 2011. doi:10.1007/s00453-010-9465-2.

4 Mohammad Ali Abam and Sariel Har-Peled. New constructions of sspds and their applications. *Comput. Geom.*, 45(5-6):200–214, 2012. doi:10.1016/j.comgeo.2011.12.003.

5 Martin Aigner and Günter M. Ziegler. *Proofs from THE BOOK*. Springer, 6th edition, 2018. Graham–Pollak theorem on biclique partitions of K_n.

6 Reuven Bar-Yehuda. One for the price of two: a unified approach for approximating covering problems. *Algorithmica*, 27(2):131–144, 2000. doi:10.1007/S004530010009.

7 Béla Bollobás and Alex D. Scott. On separating systems. *Eur. J. Comb.*, 28(4):1068–1071, 2007. doi:10.1016/j.ejc.2006.04.003.

8 Kevin Buchin and Wolfgang Mulzer. Delaunay triangulations in $O(\mathrm{sort}(n))$ time and more. *J. ACM*, 58(2):6:1–6:27, 2011. doi:10.1145/1944345.1944347.

9 Paul B. Callahan and S. Rao Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. ACM*, 42(1):67–90, 1995. doi:10.1145/200836.200853.

10 Timothy M. Chan. Approximate nearest neighbor queries revisited. *Discret. Comput. Geom.*, 20(3):359–373, 1998. doi:10.1007/PL00009390.

11 Timothy M. Chan, Sariel Har-Peled, and Mitchell Jones. On locality-sensitive orderings and their applications. *SIAM J. Comput.*, 49(3):583–600, 2020. doi:10.1137/19M1246493.
12 Minati De and Abhiruk Lahiri. Geometric dominating-set and set-cover via local-search. *Comput. Geom.*, 113:102007, 2023. doi:10.1016/J.COMGEO.2023.102007.
13 Lindsey Deryckere, Joachim Gudmundsson, André van Renssen, Yuan Sha, and Sampson Wong. A wspd, separator and small tree cover for c-packed graphs. In Pat Morin and Eunjin Oh, editors, *19th International Symposium on Algorithms and Data Structures, WADS 2025, York University, Toronto, Canada, August 11-15, 2025*, volume 349 of *LIPIcs*, pages 21:1–21:15, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.WADS.2025.21.
14 Tomás Feder and Daniel H. Greene. Optimal algorithms for approximate clustering. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, STOC '88, pages 434–444, New York, NY, USA, 1988. ACM. doi:10.1145/62212.62255.
15 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM J. Comput.*, 35(1):151–169, 2005. doi:10.1137/S0097539703436357.
16 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.
17 Joachim Gudmundsson and Sampson Wong. A well-separated pair decomposition for low density graphs. In Kasper Green Larsen and Barna Saha, editors, *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2026, Vancouver, BC, Canada, January 11-14, 2026*, pages 3688–3723. SIAM, 2026. doi:10.1137/1.9781611978971.135.
18 S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Math. Surveys & Monographs*. Amer. Math. Soc., Boston, MA, USA, 2011. doi:10.1090/surv/173.
19 S. Har-Peled, H. Kaplan, and M. Sharir. Approximating the k-level in three-dimensional plane arrangements. In M. Loebl, J. Nešetřil, and R. Thomas, editors, *A Journey Through Discrete Mathematics: A Tribute to Jiří Matoušek*, pages 467–503. Springer, 2017. doi:10.1007/978-3-319-44479-6.
20 Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006. doi:10.1137/S0097539704446281.
21 Sariel Har-Peled, Benjamin Raichel, and Eliot W. Robson. Well-separated pairs decomposition revisited. *CoRR*, abs/2509.05997, 2025. doi:10.48550/arXiv.2509.05997.
22 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994. doi:10.1145/185675.306789.
23 Dana Moshkovitz. The projection games conjecture and the np-hardness of ln n-approximating set-cover. *Theory Comput.*, 11(7):221–235, 2015. doi:10.4086/toc.2015.v011a007.
24 Nabil H. Mustafa and Saurabh Ray. PTAS for geometric hitting set problems via local search. In John Hershberger and Efi Fogel, editors, *Proceedings of the 25th ACM Symposium on Computational Geometry, Aarhus, Denmark, June 8-10, 2009*, pages 17–22. ACM, 2009. doi:10.1145/1542362.1542367.
25 Giri Narasimhan and Michiel H. M. Smid. *Geometric spanner networks*. Cambridge University Press, 2007. doi:10.1017/CBO9780511546884.
26 Peter Slavík. A tight analysis of the greedy algorithm for set cover. *J. Algorithms*, 25(2):237–254, 1997. doi:10.1006/jagm.1997.0887.
27 Michiel H. M. Smid. The well-separated pair decomposition and its applications. In Teofilo F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 2: Contemporary and Emerging Applications*. Chapman and Hall/CRC, 2018.
28 Jan van Leeuwen and Derick Wood. The measure problem for rectangular ranges in d-space. *J. Algorithms*, 2(3):282–300, 1981. doi:10.1016/0196-6774(81)90027-4.
29 Kasturi R. Varadarajan. A divide-and-conquer algorithm for min-cost perfect matching in the plane. In *39th Annual Symposium on Foundations of Computer Science, FOCS 1998, Palo Alto, California, USA, November 8-11, 1998*, pages 320–331. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743466.

A Construction of the ABC in the Euclidean Case

We start with some standard definitions, which are taken from Chan *et al.* [11]. In the following, we fix a parameter $\xi = 1/2^\lambda$, for some integer $\lambda > 0$, and use

$$\tfrac{1}{\xi} = 2^\lambda \qquad \text{and} \qquad \lambda = \log_2 \tfrac{1}{\xi}. \tag{1}$$

▶ **Definition 38.** *Let $\mathcal{C} \subseteq \mathbb{R}^d$ be an axis-parallel cube with side length ζ. For an integer $t > 1$, partitioning $\mathcal{C}$ uniformly into a $t \times t \times \cdots \times t$ subcubes, forms a **t-grid** $\mathcal{G}(\mathcal{C}, t)$. The diameter of a cube $\square$ is $\nabla(\square) = \mathrm{sidelength}(\square)\sqrt{d}$, and let $\overline{\mathrm{c}}(\square)$ denote its **center**.*

▶ **Definition 39.** *An **ξ-quadtree** $\mathcal{T}_\varepsilon$ is a quadtree-like structure, built on a cube with side length ζ, where each cell is partitioned into an $\frac{1}{\xi}$-grid. The construction then continues recursively into each grid cell of interest. As such, a node in this tree has up to $1/\xi^d$ children, and a node at level $i \geq 0$ has an associated cube of side length $\zeta\xi^i$. When $\frac{1}{\xi} = 2$, this is a regular quadtree.*

Informally, an ξ-quadtree connects a node directly to its descendants that are λ levels below it in the regular quadtree. Thus, to cover the quadtree, we need λ different ξ-quadtrees to cover all the levels of the quadtree. This cover can be realized by setting the level of the root of the ξ-quadtree. Specifically, we fix the root of the ξ-quadtree $\mathcal{T}_\varepsilon^i$ to be $[0, 2^i)^d$, for $i = 1, \ldots, \lambda$. We thus get the following.

▶ **Lemma 40.** *Let $\mathcal{T}$ be a regular (infinite) quadtree over $[0, 2)^d$. There are λ ξ-quadtrees $\mathcal{T}_\xi^1, \ldots, \mathcal{T}_\xi^\lambda$, such that the collection of cells at each level in $\mathcal{T}$ is contained in exactly one of these λ ξ-quadtrees.*

We need the following result on shifted quadtrees (see also [11, Appendix A.2]).

▶ **Lemma 41** ([10], Lemma 3.3)**.** *Consider any two points p, q in the unit cube $[0, 1)^d$, and let $\mathcal{T}$ be the infinite quadtree of $[0, 2)^d$. For $D = 2\lceil d/2 \rceil$ and $i = 0, \ldots, D$, let $\nu_i = (i/(D+1), \ldots, i/(D+1))$. Then there exists an $i \in \{0, \ldots, D\}$, such that $p + \nu_i$ and $q + \nu_i$ are contained in a cell of $\mathcal{T}$ with side length $\leq 2(D+1)\left\|p - q\right\|$.*

▶ **Observation 42.** *Consider a ball $\mathscr{b}(c, \ell) \subseteq [0, 1]^d$, and its axis aligned bounding cube C. Let p and q be the bottom and top corners of C. Applying Lemma 41 to p and q, we have that there is a node v in a shifted quadtree that its sidelength is at most $L \leq 2(D+1)\left\|p - q\right\| \leq (2d+4)\sqrt{d}2\ell \leq 8d^{3/2}\ell$, as $\left\|p - q\right\| = \sqrt{d} \cdot 2\ell$.*

▶ **Theorem 20.** *For a set P of n points in $\mathbb{R}^d$, and $\varepsilon \in (0, 1)$, one can compute a pair decomposition $\mathcal{W}$ of size $O(\frac{d^3 n}{\varepsilon} \log \frac{d}{\varepsilon})$. The decomposition $\mathcal{W}$ is $\frac{1}{\varepsilon}$-ABC and $\frac{1}{\varepsilon}$-SSPD.*

Proof. Let ξ be the maximum number that is a power of 2 and smaller than $< \varepsilon/(128d^2)$. We construct the $D = 2\lceil d/2 \rceil \leq d+1$ shifted quadtrees of Lemma 41 over the points of P (we can assume $P \subseteq [0, 1]^d$ by shifting and scaling). Next, for each such shifted quadtree, we compute the λ ξ-quadtrees each one of them induces. We compress the resulting $N = O(D\lambda) = O(d \log \frac{d}{\varepsilon})$ ε-quadtrees – they each store the n points of P in their leaves, and each one of them has at most $n - 1$ internal nodes.

Now, given such a compressed ξ-quadtree, for each of its nodes, v, we consider its $\frac{1}{\xi}$-grid $\mathcal{G}_v$, partitioning its cell $\square_v$, and assume the cell side length is ζ. For a cell $\square \in \mathcal{G}_v$ that contains at least one point of P, we generate several pairs associated with $\square$ to the computed ABC, as follows. Let

$$\rho = \frac{8\sqrt{d}}{\varepsilon}\zeta.$$

For $i = 0, \ldots, 1/\xi$, consider the spheres

$$S_i = \left\{ x \in \mathbb{R}^d \;\middle|\; \|\bar{\mathsf{c}} - x\| = \rho + i\zeta \right\}.$$

Let $\mathcal{G}_{v,i}$ be the set of all the grid cells of $\mathcal{G}_v$ intersecting S_i, and let $Y_i = \cup_{C \in \mathcal{G}_{v,i}} (C \cap P)$, for $i = 0, \ldots, 1/\xi$. We add the pairs $\mathcal{p}_i = \{P \cap \square, Y_i\}$ to the computed ABC, for $i = 0, \ldots, 1/\xi$, see Figure 3. It is straightforward to verify that all these pairs are ε-stable.

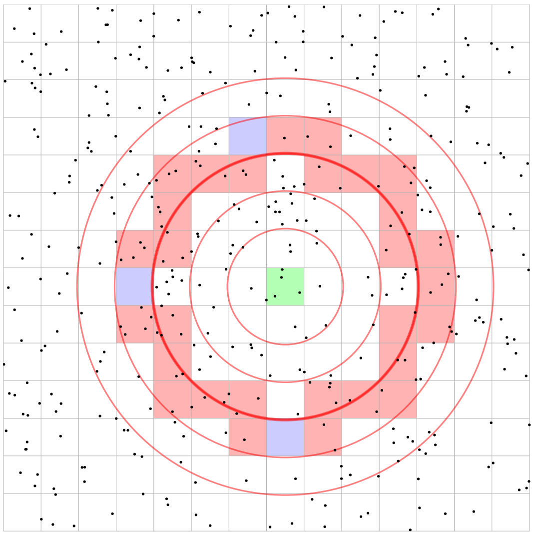


Figure 3 The set of points in the green cell is paired with the set of points in the red cells (the blue cells are empty).

Consider a specific compressed ξ-quadtree $\mathcal{T}$. A node v in it gives rise to at most $O(n_v(1 + 1/\xi))$ pairs, where n_v is the number of children v has (i.e., $n_v = |\{C \in \mathcal{G}_v \mid C \cap P \neq \emptyset\}|$). Clearly, $\sum_{v \in \mathcal{T}} n_v = n - 1$. Thus, each tree gives rise to $O(n/\xi)$ pairs, and overall there are $O(n/\xi) = O(d^2 n/\varepsilon)$ pairs generated for this tree. Since there are N ξ-quadtrees, it follows that the total number of pairs is $O(d^3 \frac{n}{\varepsilon} \log \frac{d}{\varepsilon})$.

As for the covering property, consider two points $p, q \in P$, with $\ell = \|p - q\|$. Let $\mathcal{b} = \mathcal{b}(p, \ell)$. By Observation 42, there is a shifted quadtree that contains this ball in a cell with sidelength L, where

$$2\ell \leq L \leq 8d^{3/2}\ell.$$

This cell $\square$ corresponds to a node v in one of the computed ξ-quadtrees. Let $\square_p$ and $\square_q$ be the two children cells of $\square$ that contains p and q. The sidelength of $\square_p$ is $\zeta = \xi L$. Observe that $\varepsilon/(256d^2) \leq \xi \leq \varepsilon/(128d^2)$, and as such

$$\frac{\ell}{\zeta} = \frac{\ell}{\xi L} \geq \frac{256 d^2 \ell}{\varepsilon \cdot 8 d^{3/2} \ell} = \frac{32\sqrt{d}}{\varepsilon} > \frac{8\sqrt{d}}{\varepsilon} = \frac{\rho}{\zeta}.$$

This implies that $\ell > \rho$ for this cell. Similarly, we have

$$\frac{\ell}{\zeta} = \frac{\ell}{\xi L} \leq \frac{128 d^2 \ell}{\varepsilon \cdot 2\ell} = \frac{64 d^2}{\varepsilon} \leq \frac{1}{\xi}.$$

Namely, the cell of $\square_p$ would be in the center of one of the generated sets in the pair decomposition, and it would contain the points of $P \cap \square_q$ on the other side. We conclude that the pair pq would be covered in the generated decomposition. ◀

▶ Remark 43. Every pair in the decomposition computed by Theorem 20 can be represented implicitly (in a similar fashion to WSPD). Indeed, each pair computed is a quadtree node coupled with a set of quadtree nodes, all children of the same ξ-quadtree node, that intersect a certain sphere. The number of such grid cells in a set is $O(1/\xi^{d-1}) = O(1/\varepsilon^{d-1})$. Namely, a single set can be represented as a list of $O(1/\varepsilon^{d-1})$ nodes, where each node contributes all the points stored in its subtree to the set.

▶ Remark 44. Theorem 20 improves the number of pairs in the $\frac{1}{\varepsilon}$-SSPD compared to previous work, from $O(n/\varepsilon^d)$, to $O(n \cdot \frac{d^3}{\varepsilon} \log \frac{1}{\varepsilon})$. This result should improve some results using SSPDs. However, the current construction of SSPDs does not have the property that the weight of the SSPD is near linear, and it can be quadratic in the worst case. It is quite conceivable that previous constructions can be modified to use the new ideas, and get a construction with an improved number of pairs and a small weight. Note that the resulting decomposition is not disjoint, and a single pair would get covered multiple times.

B The One-Dimensional Problem

B.1 A 3-Approximation for CoverWSPD

Algorithm. We sort the input P to sweep our points Q from right to left. As our points have a grid structure, we can process them column-wise. We only process uncovered points and add squares to cover the left half of the vicinity of the current point.

Horizontal and vertical intervals define a square. The horizontal interval I determines when an interval is inserted or deleted during the sweep, i.e. for which columns the square is relevant. The vertical intervals J cover points in the current column. We use a segment tree as status for the latter, see Bentley's algorithm [28]. A leaf of this segment tree is a point $p_i \in P$ representing the point (p_i, p_j), where p_j is fixed by the current column. We can then query the segment tree for the next uncovered point, ordered from bottom to top, in this column. To ensure that no grid points below the diagonal are considered, for each column, we add the interval below the diagonal into the segment tree.

Starting a new column and processing an uncovered point are our two events. For such a point p, we add three squares to cover the left half of its vicinity (see Figure 2):

- S_p (i.e., the shadow of p),
- S, which is the unique shadow square that has p as its top right corner, and
- S_q, where q is the bottom left corner of S.

Note that S and S_q might not be in $\mathcal{F}$. However, as described above, we can move their anchor up resp. up and left to a point in $Q \cap V(p)$. We only add S and S_q if they contain a point from $V(p)$ besides p.

We insert the vertical segments of S_p and S into the segment tree. Since the anchor of S_q is left of the current column, we insert S_q into a priority queue, prioritized by the column. In another priority queue, we add the delete events for the three squares, each with a priority indicating when to be deleted from the segment tree (defined by the horizontal segments).

When a new column is started, we check the priority queues for squares whose intervals need to be added to or deleted from the segment tree. Furthermore, at the start of a new column, we add an interval to the segment tree to cover everything in the current column below the diagonal, and delete the corresponding interval of the previous column.

We proceed with the following column if the segment tree contains no uncovered points. A visualization of the sweep and event status is shown in Figure 4.

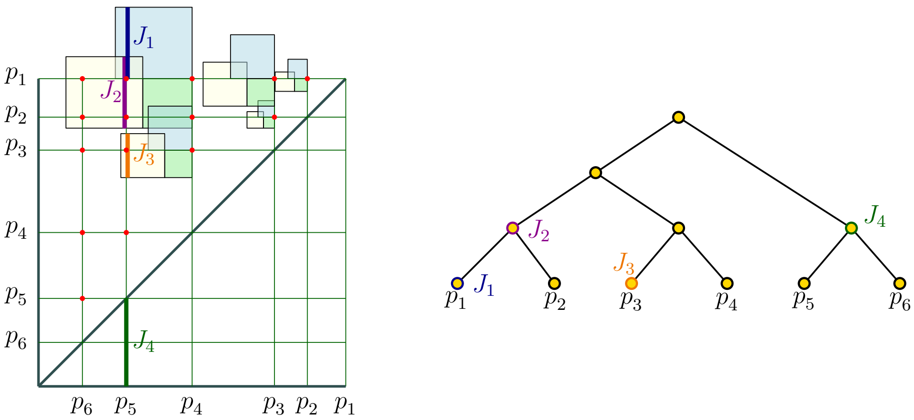


Figure 4 Visualization of the sweep and event status at the beginning of column p_5. A search would find p_4. When we place an interval over it on the backtrack, we check if the children of a node are covered. If they are, we mark the node as covered. So, covering p_4 propagates to the root node, and the next search would terminate at the root.

Approximation factor. When we process an uncovered point p, the three squares we add cover the intersection of its vicinity $V(p)$ with the swept halfplane, and hence every optimal square covering p. Such squares are never charged again. Conversely, an uncovered point has no charged optimal square covering it. Thus we spend at most three squares per optimal one, giving an approximation factor of 3. By the following lemma, three squares are the minimum needed to cover the swept part of a vicinity in any sweep direction, so the analysis of the approximation factor is tight.

▶ **Lemma 45.** *For any $\varepsilon > 0$, there exists a halfplane such that only three squares are needed to cover the intersection of the vicinity of a point p and the halfplane.*

Proof. The halfplane of our sweep direction works: as shown in Figure 2 (right), three squares cover the left half of the vicinity of p. It remains to show that no halfplane through p admits a cover by fewer squares. Consider the seven features highlighted in Figure 5: the upper-left point r, the neighborhood of p, the diagonals of the upper-right and lower-left quadrants, and three points on the two diagonals of the lower-right quadrant. By the structure of the cover in Figure 2, no shadow square can contain two of these features. Hence, a halfplane containing k features requires at least k squares. Every halfplane through p contains at least four features, except for two directions, the sweep direction and its mirror image, which contain exactly three. Therefore, every halfplane needs at least three squares, and three suffice precisely for these two directions. ◀

Finally, we analyze the runtime.

▶ **Lemma 36.** *Given a set P of n points in $\mathbb{R}$, and parameter $\varepsilon > 0$, one can compute, in $O(n \log n)$ time, a cover $\frac{1}{\varepsilon}$-WSPD of P with k pairs, where $k \leq 3\mathrm{opt}$, where opt is the minimum number of pairs in any cover $\frac{1}{\varepsilon}$-WSPD of P.*

Proof. The optimal solution has $O(n)$ squares, and charging each uncovered point to an optimal interval bounds the number of processed points by $O(n)$ as well. Each square is computed in constant time and triggers a single interval insertion, deletion, and point lookup in the segment tree, each in $O(\log n)$ time. To find the next uncovered point, every node stores a flag marking whether its subtree is fully covered, maintained during insertions and deletions by checking both children on backtracking. Together with the $O(\log n)$ priority-queue operations per event, the total runtime is $O(n \log n)$. ◀

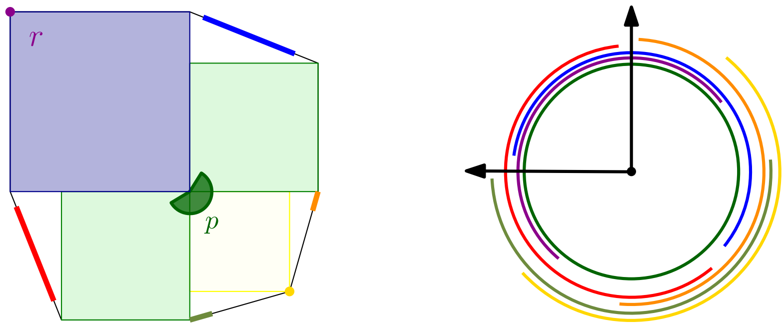


Figure 5 The seven features on the left in different colors, and an angle plot on the right with intervals shows which direction includes which features. The two arrows highlight directions of the normal vector, which are the only ones with fewer than four features.

B.2 From a CoverWSPD to a PartitionWSPD

If one requires the separated pairs to be disjoint, we give a constant approximation.

▶ **Lemma 46.** *Let $\mathcal{S}$ be a set of n axis-aligned squares in the plane. Then one can compute, in polynomial time, a set $\mathcal{R}$ of at most $9n$ interior-disjoint rectangles, such that $\cup_{\square \in \mathcal{S}} = \cup \mathcal{S} = \cup \mathcal{R}$. Furthermore, for each $\mathpzc{r} \in \mathcal{R}$, there exists a square $\square \in \mathcal{S}$, such that $\mathpzc{r} \subseteq \square$.*

Proof. Number the squares $\mathcal{S} = \{\mathpzc{s}_1, \ldots, \mathpzc{s}_n\}$ by increasing size, iteratively insert them into the union, and collect all vertices that ever appear on the boundary of the union into a set U.

Each iteration adds at most 8 vertices to U. Indeed, traversing an edge e of $\mathpzc{s}_i$, once e enters a larger square $\mathpzc{s}_j$ $(j < i)$ it cannot reappear on the union, as its endpoint lies in the interior of $\mathpzc{s}_j$. Thus the portion of e on the boundary of $\cup_{t=1}^{i} \mathpzc{s}_t$ is empty or a single segment, so $\mathpzc{s}_i$ contributes at most four edges, and hence at most 8 vertices, to U.

For $i = 1, \ldots, n$, let $\mathcal{U}_i = \mathbb{R}^2 \setminus \cup_{t=1}^{i} \mathpzc{s}_t$, and let $R_i = \mathcal{U}_i \cap \mathpzc{s}_i$ be the region newly covered in iteration i. The algorithm outputs the rectangles of the vertical decomposition of R_i. Let $V_i = \mathsf{V}(\partial \mathcal{U}_i)$ be the vertices of the union, let $B_i = V_{i-1} \cap \mathpzc{s}_i$ be those "buried" in iteration i, and let $b_i = |B_i|$.

The vertical decomposition of R_i needs at most $b_i + 1$ rectangles: sweeping R_i left to right, every new trapezoid past the first is caused by a vertex of B_i (and if $b_i = 0$, $\mathpzc{s}_i$ is an "island" contributing one rectangle). Since every buried vertex was created in an earlier iteration, $\sum_{i=1}^{n} b_i \leq 8n - 4$, so the algorithm outputs at most $9n$ rectangles. ◀

▶ Remark 47. A similar result to Lemma 46 holds for any (convex) pseudodisks, but the constant is somewhat worse. A direct proof follows by randomized incremental construction drawing the objects from "back to front", see [19, Section 2].

Combining Lemma 34 and Lemma 46 implies the following result.

▶ **Lemma 37.** *Given a set P of n points in $\mathbb{R}$, and a parameter $\varepsilon \in (0, 1)$, one can compute, in $n^{O(1)}$ time, a partition $\frac{1}{\varepsilon}$-WSPD of P with k pairs, where $k \leq 10\mathrm{opt}$, where opt is the minimum number of pairs in any cover (or partition) $\frac{1}{\varepsilon}$-WSPD of P.*