Kernelization for Spreading Points

Fedor V. Fomin ✉ 📷
University of Bergen, Norway

Petr A. Golovach ✉ 📷
University of Bergen, Norway

Tanmay Inamdar ✉ 📷
University of Bergen, Norway

Saket Saurabh ✉
Institute of Mathematical Sciences, Chennai, India
University of Bergen, Norway

Meirav Zehavi ✉ 📷
Ben-Gurion University, Beer-Sheva, Israel

─── **Abstract** ───

We consider the following problem about dispersing points. Given a set of points in the plane, the task is to identify whether by moving a small number of points by small distance, we can obtain an arrangement of points such that no pair of points is "close" to each other. More precisely, for a family of n points, an integer k, and a real number $d > 0$, we ask whether at most k points could be relocated, each point at distance at most d from its original location, such that the distance between each pair of points is at least a fixed constant, say 1. A number of approximation algorithms for variants of this problem, under different names like distant representatives, disk dispersing, or point spreading, are known in the literature. However, to the best of our knowledge, the parameterized complexity of this problem remains widely unexplored. We make the first step in this direction by providing a kernelization algorithm that, in polynomial time, produces an equivalent instance with $\mathcal{O}(d^2 k^3)$ points. As a byproduct of this result, we also design a non-trivial fixed-parameter tractable (FPT) algorithm for the problem, parameterized by k and d. Finally, we complement the result about polynomial kernelization by showing a lower bound that rules out the existence of a kernel whose size is polynomial in k alone, unless NP $\subseteq$ coNP /poly.

1 Introduction

The problem of dispersing a family of objects is a common theme in many situations in computational geometry. It appears naturally in the wide range of settings that require assigning elements to locations. In many scenarios, dispersing has two often contradicting objectives. On the one hand, it is desirable not to place the objects too close to each other. This can be due to a variety of reasons, e.g., placing customers in a restaurant in socially distant manner, to placing wireless sensors far from each other in order to avoid interference. On the other hand, we may already have an existing placement of the objects, and wish to optimize the resources spent on moving the objects.

With this motivation, we consider the following mathematical model of the dispersing problems. In this model, our aim is to modify a given arrangement of points in the plane, by moving some of the points into new positions within a given distance, such that the Euclidean distance between each pair of points in the final arrangement is at least a fixed constant, say 2. Equivalently, the problem can be reformulated in terms of finding a non-overlapping arrangement of unit disks, formulated below as the problem DISK DISPERSAL.
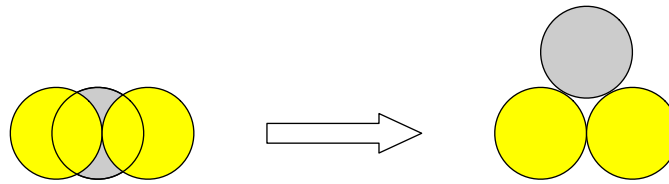

DISK DISPERSAL

Input: A family $\mathcal{S}$ of n unit disks, an integer $k \geq 0$, and a real $d \geq 0$.

Task: Decide whether it is possible to obtain from $\mathcal{S}$ a family of non-overlapping unit disks $\mathcal{P}$ by moving at most k disks into new positions in such a way that each unit disk is moved a distance at most d. [1]

DISK DISPERSAL– and therefore, the problem of spreading points – is closely related to the problem of finding a system of q-*distant representatives*. This problem was introduced by Fiala, Kratochvíl, and Proskurowski [14] as a geometric extension of the classic combinatorial notion of the "systems of distinct representatives". For a set of geometric objects in a metric space and a number $q > 0$, the task is to choose one representative point from each object such that the selected points are at a distance at least q from each other. For $k = n$, an instance $(\mathcal{S}, d, k)$ of DISK DISPERSAL can be viewed as an instance of the problem of finding a system of q-distance representatives by setting $q = 2$ and defining the set of geometric objects as follows: for each disk $D \in \mathcal{S}$, create a disk with the same center but with radius d (instead of 1). This yields that DISK DISPERSAL is also NP-hard for $d = 2$ from the result of [14].

The problem of computing the distant representatives has applications in map labeling and data vizualization, where the goal is to place labels as close as possible to the specified features of the map but avoiding overlapping (thus the centers of labels are the centers of non-intersecting disks, ensuring that they are sufficiently separated) [9, 20, 21]. The problem is also related to problems of "imprecise points" [22, 23], the settings where locations of points are given with some precision. Approximation algorithms for this and related point spreading problems – where the goal is to place the specified number of points within a certain region so as to maximize the smallest pairwise distance between the points – were developed in [3, 4, 6, 10, 11, 12, 13, 19, 2, 18].

---

[1] All (unit) disks considered in the paper are open unless specified otherwise. In particular, two unit disks touching each other are not considered to be overlapping. Due to this simplifying assumption, we avoid the discussion about placing disks such that the distance between their boundaries is infinitesimally small.

Figure 1 An example of DISK DISPERSAL with $k = 1$ and $d = \sqrt{3}$. A non-overlapping arrangement of disks obtained from a family of three disks by moving the central disk at distance $\sqrt{3}$.

To the best of our knowledge, the parameterized complexity of dispersal problems are widely unexplored. The notable exception is the work of Demaine, Hajiaghayi, and Marx [7] on dispersion in graphs. In this problem, we are given an underlying edge-weighted graph, called the *connectivity graph G*, and a set of k "agents" or "pebbles", located at a subset of vertices G. The task is to move the pebbles to distinct vertices and such that no two pebbles are adjacent. The movement problem is W[1]-hard parameterized by the number of pebbles, even in the case when each pebble is allowed to move at most one step.

1.1 Our Results

Our first result concerns kernelization (polynomial compression) of DISK DISPERSAL. Informally speaking, in parameterized complexity, the polynomial kernel is a polynomial-time algorithm that compresses the instance of a parameterized problem to the instance whose size is bounded by a polynomial of the parameter. Theorem 1 gives an algorithm that runs in polynomial time, and reduces the number of disks to some polynomial of d and k.

▶ **Theorem 1.** *There is a polynomial-time algorithm that, given an instance $(\mathcal{S}, k, d)$ of DISK DISPERSAL, outputs an equivalent instance $(\mathcal{S}', k, d)$ of the same problem, where the number of unit disks is $|\mathcal{S}'| = \mathcal{O}((d + 1)^2 k^3)$, and $\mathcal{S}' \subseteq \mathcal{S}$.*

Strictly speaking, the algorithm in Theorem 1 is not a polynomial kernel according to the standard definition of this notion – we do not guarantee that the coordinates of disks, and thus the overall size of the compressed instance, is bounded by a polynomial in k and d. We call such a compression algorithm a *partial kernel*. Further, we observe in Theorem 12 that the partial kernel from Theorem 1 can be modified to be a polynomial kernel if the centers of input disks are constrained to be rationals and we parameterize the problem by k, d, and the maximum denominator of coordinates of centers.

For a parameterized problem, given the existence of a (partial) kernel, it is usually straightforward to design a fixed-paramter tractable (FPT) algorithm by an exhaustive enumeration of all candidate solutions. For DISK DISPERSAL, however, this is not entirely obvious. After computing an equivalent reduced instance by applying Theorem 1, one can enumerate all possible subsets of at most k unit disks that are to be moved. Now, for each such subset, we want to decide whether each unit disk in the subset can be moved by a distance of at most d that results in a non-overlapping configuration. Since there are infinitely many possible target locations for each unit disk, this step requires some additional work. We show that this decision subroutine can be reduced to checking whether a system of polynomial inequalities has a solution over real numbers, which can then be determined in FPT time by using classical results from computational real algebra. Thus, we obtain the following non-trivial corollary.

▶ **Corollary 2.** *DISK DISPERSAL is FPT when parameterized by $d + k$. Specifically, it is solvable in time $(dk)^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$.*

Our next result is a companion lower bound to the partial kernelization of Theorem 1, which shows that one cannot remove the dependence on d from the kernel size.

▶ **Theorem 3.** DISK DISPERSAL *parameterized by* k *does not admit a polynomial kernel unless* coNP $\subseteq$ NP/poly. *This result holds even if the distance* d *is an integer, and the centers of the given disks have rational coordinates.*

As we already mentioned, by the result of Fiala, Kratochvíl, and Proskurowski about q-distant representatives, DISK DISPERSAL is NP-hard for $d = 2$. Thus the problem is in the class para-NP for parameter d. However, the complexity of parameterization by k is more interesting, which remains open. However, in the appendix, we show that a *rectilinear* version of DISK DISPERSAL is indeed W[1]-hard parameterized by k.

Organization

In Section 2 we introduce basic notions. In Section 3, we consider kernelization for DISK DISPERSAL. Further, we give complexity lower bounds. In Section 4, we show that it is unlikely that DISK DISPERSAL admits a polynomial kernel when parameterized by k only. Finally, in Section 5, we provide some concluding remarks and future directions.

2   Preliminaries

As it is common in computational geometry, we assume the *real RAM* computational model, that is, we are working with real numbers and assume that basic operations can be executed in unit time.

Disks and Segments

For two points A and B in the plane, we use AB to denote the line segment with endpoints at A and B. The *distance* between $A = (x_1, y_1)$ and $B = (x_2, y_2)$ or the *length* of AB, is $|AB| = \|A - B\|_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. The *(open unit) disk* with a *center* $C = (c_1, c_2)$ in the plane is the set of points (x, y) satisfying the inequality $(x - c_1)^2 + (y - c_2)^2 < 1$. Whenever we write "disk" we mean an *open unit disk*, unless radius or closed-ness is specified explicitly. Clearly, two disks with centers A and B are disjoint if and only if the distance between A and B is at least two. We say that the disks *touch* if $|AB| = 2$. For real numbers $a \leq b$, we use $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$ to denote a *closed interval*. For $a_1 \leq b_1$ and $a_2 \leq b_2$, $[a_1, b_1] \times [a_2, b_2] = \{(x, y) \in \mathbb{R}^2 \mid a_1 \leq x \leq b_1 \text{ and } a_2 \leq y \leq b_2\}$. A point X is *properly inside* of a polygon P if it is inside P but X is not on the boundary; if we say that X is inside P, we allow it to be on the boundary. A disk is *(properly) iniside* of a polygon P if every point of the disk is (properly) inside of P.

Graphs

We use standard graph-theoretic terminology and refer to the textbook of Diestel [8] for definitions of standard notions. Let $\mathcal{S}$ be a set of geometric objects in the plane (i.e., non-empty subsets of $\mathbb{R}^2$). Then, it is possible to define an intersection graph $G(\mathcal{S})$ as follows: $G(\mathcal{S})$ contains a unique vertex corresponding to every object in $\mathcal{S}$, and there is an edge between the two vertices iff the corresponding two objects in $\mathcal{S}$ have a non-empty intersection. *Unit disk graphs* are the intersection graphs of unit disks in the plane. Note that, given a family $\mathcal{S}$ of unit disks, we can construct the corresponding unit disk graph $G(\mathcal{S})$ in quadratic time.

Parameterized Complexity

We refer to the standard textbooks ([5, 17]) for introduction to the area and formal definitions. Here, we only give a brief overview. Let $(\mathcal{I}, k)$ be an instance of a decision problem Π, where k is a non-negative integer. We say that Π is *fixed-parameter tractable* by k, if there exists an algorithm that can decide whether $\mathcal{I}$ is a yes-instance of Π in time $f(k) \cdot |\mathcal{I}|^{\mathcal{O}(1)}$ for some computable function f, where $|\mathcal{I}|$ denotes the size of the instance $\mathcal{I}$. A common way to show that it is unlikely that a parameterized problem is in FPT, one can prove that it is W[1]-hard by demonstrating a *parameterized reduction* from a known W[1]-hard problem; we refer to [5] for the formal definitions of the class W[1] and parameterized reductions.

A *kernelization* (or *kernel*) for Π is a polynomial time algorithm that, given an instance $(\mathcal{I}, k)$ of Π, outputs an equivalent instance $(\mathcal{I}', k')$ of Π such that $|\mathcal{I}'| + k' \leq g(k)$ for a computable function g. A kernel is *polynomial* if g is a polynomial. It can be shown that every decidable FPT problem admits a kernel. However, it is unlikely that all FPT problems have polynomial kernels. In particular, there is the now standard *cross-composition* technique to show that a parameterized problem does not admit a polynomial kernel unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.

Systems of Polynomial Inequalities

In our FPT algorithm, we will need to find suitable locations for new disks that need to be added such that the locations are "compatible" with an existing arrangement of disks. We will achieve this by solving systems of polynomial inequalities. We use the following result.

▶ **Proposition 4** (Theorem 13.13 in [1]). *Let R be a real closed field, and let $\mathcal{P} \subseteq R[X_1, \ldots, X_k]$ be a finite set of s polynomials, each of degree at most c, and let*

$$(\exists X_1)(\exists X_2) \ldots (\exists X_k) F(X_1, X_2, \ldots, X_k)$$

be a sentence, where $F(X_1, \ldots, X_k)$ is a quantifier-free boolean formula involving $\mathcal{P}$-atoms of type $P \odot 0$, where $\odot \in \{=, \neq, >, <\}$, and P is a polynomial in $\mathcal{P}$. Then, there exists an algorithm to decide the truth of the sentence with complexity $s^{k+1} c^{\mathcal{O}(k)}$ in D,[2] where D is the ring generated by the coefficients of the polynomials in $\mathcal{P}$.

Furthermore, a point $(X_1^*, \ldots, X_k^*)$ satisfying $F(X_1, \ldots, X_k)$ can be computed in the same time by Algorithm 13.2 (sampling algorithm) of [1] (see Theorem 13.11 of [1]).

3 Kernelization and FPT Algorithms for Disk Dispersal

In this section, we first prove Theorem 1 on partial kernel for DISK DISPERSAL parameterized by $k + d$. Specifically, the output instance of the partial kernel is guaranteed to consist of only $\mathcal{O}(d^2 k^3)$ unit disks. In case the coordinates of the disks in the input instance are rationals of the form $a + \frac{b}{c}$ where b, c are bounded by a fixed constant (or a polynomial in $k + d$), our partial kernel in fact yields a (normal) kernel. Finally, using our partial kernel, we prove in Corollary 2 that DISK DISPERSAL is FPT parameterized by $k + d$.

The proofs of our partial kernels begin with the simple observation that if we are given a yes-instance, then the unit disk graph corresponding to the input set of unit disks admits a vertex cover of size at most k. So, in polynomial time we obtain a vertex cover U of size at

---

[2] That is, the algorithm performs $s^{k+1} c^{\mathcal{O}(k)}$ operations in D.
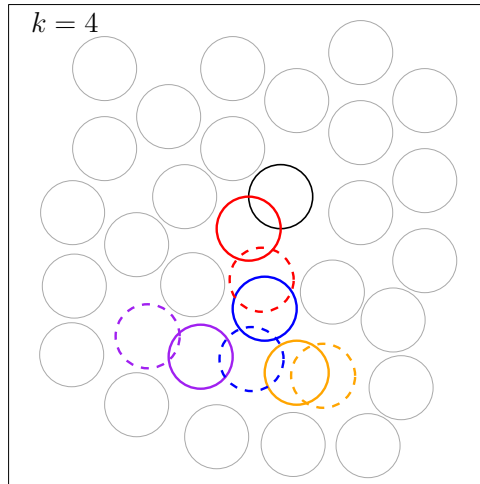
Figure 2 Example of the propagation effect. The dotted objects correspond to a solution where an object of a certain color is replaced by the dashed object of the same color.

most $2k$. At first glance, one may think to remove all input unit disks that do not intersect any unit disk in U. However, we might be forced to perform movement operations that make some neighborhood sets larger (e.g., see Figure 2), which, in turn, can have a propagating effect that forces us to move unit disks that are "quite far" from all unit disks in U. Still, we can prove by induction on k that if the input instance is a yes-instance, then it admits a solution where all the unit disks that are moved are at distance at most $\mathcal{O}(d^2k^2)$ from at least one unit disk in U. This gives rise to a reduction rule where we only keep the unit disks within this distance from at least one unit disk in U as well as additional unit disks at some (almost negligible) distance from them.

After having reduced the number of unit disks, we can shift the unit disks that we keep so that the coordinates of their centers will be polynomial in $k + d$, under the assumption that the coordinates of the unit disks in the input instance are rationals of the form $a + \frac{b}{c}$ where b, c are bounded by a fixed constant (or a polynomial in $k + d$). To obtain FPT algorithms, we first apply our partial kernels. Afterwards, we guess which disks to move. Then, we determine how to move them by solving a corresponding system of polynomial inequalities.

For the sake of formality, we will use the notion of a solution in this section as follows.

▶ **Definition 5.** *Let $(\mathcal{S}, k, d)$ be an instance of* DISK DISPERSAL. *A solution is a bijective function* move$: \mathcal{S} \to \mathcal{P}$ *such that:*
1. $\mathcal{P}$ *is a* packing, *i.e., a non-overlapping set of unit disks.*
2. $|\{D \in \mathcal{S} : \text{move}(D) \neq D\}| \leq k$.
3. *For every $D \in \mathcal{S}$: The distance between the centers of D and* move(D) *is at most d.*
We define the set of unit disks moved by move *as* $\{D \in \mathcal{S} : \text{move}(D) \neq D\}$, *and the* size of move *as the size of this set.*

Notice that any set of unit disks that is moved by a solution to DISK DISPERSAL is in particular a vertex cover (though not necessarily a minimal one) for the intersection graph of the input set of unit disks. As previously discussed, since the VERTEX COVER problem admits a 2-approximation algorithm in polynomial time, this yields the following observation.

▶ **Observation 6.** *There exists a polynomial-time algorithm that, given an instance $(\mathcal{S}, k, d)$ of* DISK DISPERSAL, *either correctly concludes that $(\mathcal{S}, k, d)$ is a no-instance, or outputs a vertex cover of size at most $2k$ for the unit disk graph corresponding to $\mathcal{S}$.*

We will also need the following observation, which is directly implied by the fact that the area of a disk of radius r is πr^2, while the area of a unit disk (whose radius is 1) is π.

▶ **Observation 7.** *The number of pairwise non-intersecting unit disks in a disk of radius r is at most r^2.*

Towards the presentation of our partial kernel, we need to prove one lemma. Informally speaking, this lemma shows that the set of disks that may be potentially moved in a yes-instance is contained in a bounded area around a small number of disks, in particular the disks that form a vertex cover in the intersection graph. Furthermore, since all such disks, except that forming the vertex cover, are non-intersecting, this lemma eventually helps us bound the number of such disks by a polynomial in k and d.

▶ **Lemma 8.** *Let $(\mathcal{S}, d, k)$ be a yes-instance of DISK DISPERSAL. Let U be a vertex cover for the intersection graph of $\mathcal{S}$. Then, any minimum-sized solution to $(\mathcal{S}, k, d)$ only moves unit disks whose center is at distance at most $(d+2) \cdot k$ from the center of at least one unit disk in U.*

Proof. We prove the lemma by induction on k. When $k = 0$, the only minimum-sized solution to $(\mathcal{S}, k, d)$ is the one that moves no unit disk, and hence the claim trivially follows. Now, suppose that the claim holds for $k - 1 \geq 0$, and let us prove it for k. If the intersection graph of $\mathcal{S}$ is edgeless, then the only minimum-sized solution to $(\mathcal{S}, k, d)$ is the one that moves no unit disk, and hence the claim trivially follows as in the base case. So, we can next suppose that there exist two different unit disks $D, D' \in \mathcal{S}$ that intersect each other. See Figure 3 for an illustration.

Since U is a vertex cover, it must contain at least one unit disk among D and D', denoted by X. Moreover, any solution to $(\mathcal{S}, k, d)$ must move at least one unit disk among D and D'. Let $\mathsf{move} : \mathcal{S} \to \mathcal{P}$ be an arbitrary minimum-sized solution to $\mathcal{I} = (\mathcal{S}, k, d)$, and let Y be a unit disk among D and D' that move moves to attain $\mathcal{P}$. Let $Y' = \mathsf{move}(Y)$, and let $\mathcal{S}' = (\mathcal{S} \setminus \{Y\}) \cup \{Y'\}$. We attain solution $\mathsf{move}' : \mathcal{S}' \to \mathcal{P}'$ to a new instance $\mathcal{I}' = (\mathcal{S}', k-1, d)$ as follows: for every $\tilde{D} \in \mathcal{S} \setminus \{Y\}, \mathsf{move}'(\tilde{D}) = \mathsf{move}(\tilde{D}); \mathsf{move}'(Y') = Y'$. Note that move' must be a minimum-sized solution to $(\mathcal{S}', k-1, d)$, otherwise we can obtain a solution for the original instance $(\mathcal{S}, k, d)$ that is smaller than move, contradicting its optimality. Further, note that $(U \setminus \{Y\}) \cup \{Y'\}$ is a (not necessarily minimal) vertex cover for the intersection graph of $\mathcal{S}'$. By the inductive hypothesis, this means that move' only moves unit disks whose center is at distance at most $(d+2) \cdot (k-1)$ from the center of at least one unit disk in $(U \setminus \{Y\}) \cup \{Y'\}$. Moreover, the distance between the centers of Y and X is at most 2 (since they intersect) and the distance between the centers of Y' and Y is at most d, so the distance between the centers Y' and X is at most $d + 2$. In turn, this means that move only moves unit disks at distance at most $(d+2) \cdot k$ from at least one unit disk in U, which concludes the proof. ◀

We are now ready to present the partial kernel for DISK DISPERSAL. For the reader's convenience, we restate Theorem 1 here.

▶ **Theorem 1.** *There is a polynomial-time algorithm that, given an instance $(\mathcal{S}, k, d)$ of DISK DISPERSAL, outputs an equivalent instance $(\mathcal{S}', k, d)$ of the same problem, where the number of unit disks is $|\mathcal{S}'| = \mathcal{O}((d+1)^2 k^3)$, and $\mathcal{S}' \subseteq \mathcal{S}$.*

Proof. Given an instance $(\mathcal{S}, k, d)$ of DISK DISPERSAL, the (partial kernel) kernelization algorithm works as follows. Based on Observation 6, it computes a vertex cover U of size at most $2k$ for the intersection graph of $\mathcal{S}$. Then, it obtains $\mathcal{S}'$ from $\mathcal{S}$ by removing from $\mathcal{S}$ all
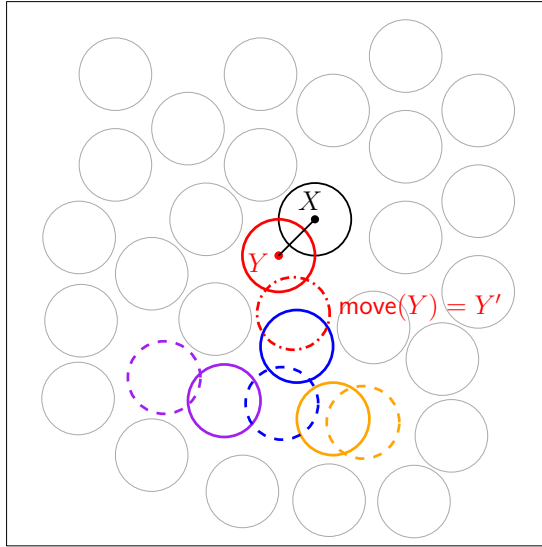
Figure 3 Illustration for Proof of Lemma 8. A vertex cover U contains disk X and a solution $\mathcal{S}$ moves a disk Y to its new location, $Y' = \mathsf{move}(Y)$, denoted in dash-dotted disk in red color. A new instance $\mathcal{I}'$ is obtained by replacing Y with Y' and reducing the budget by 1, and $U' = X \cup Y'$ is a vertex cover for the resulting intersection graph. A solution to $\mathcal{I}'$ moves the solid blue, purple, and orange disks to their new locations, shown in dashed disks of corresponding color. By inductive hypothesis, the new locations are at distance at most $(d+2) \cdot (k-1)$ from U', and the distance between X and Y' is at most $d+2$.

the unit disks at distance more than $(d+2) \cdot (k+1)$ from all unit disks in U. The output instance is $(\mathcal{S}', k, d)$. Clearly, the kernelization algorithm works in polynomial time. So, it suffices to prove that $(\mathcal{S}, k, d)$ and $(\mathcal{S}', k, d)$ are equivalent and that $|\mathcal{S}'| = \mathcal{O}(d^2 k^3)$.

We first prove the equivalence. In one direction, suppose that $(\mathcal{S}, k, d)$ is a yes-instance, and let $\mathsf{move} : \mathcal{S} \to \mathcal{P}$ be a solution to it. In particular, the restriction of move to $\mathcal{S}'$ clearly yields a packing (being a subset of $\mathcal{P}$) and moves at most as many disks as move does. So, the restriction of move to $\mathcal{S}'$ is a solution to $(\mathcal{S}', k, d)$.

In the other direction, suppose that $(\mathcal{S}', k, d)$ is a yes-instance. By Lemma 8, $(\mathcal{S}', k, d)$ admits a solution $\mathsf{move}' : \mathcal{S}' \to \mathcal{P}'$ that only moves unit disks whose centers are at distance at most $(d+2) \cdot k$ from the center of at least one unit disk in U.[3] Define $\mathsf{move} : \mathcal{S} \to \mathcal{P}$ for $\mathcal{P} = \mathcal{P}' \cup (\mathcal{S} \setminus \mathcal{S}')$ as follows: for every $D \in \mathcal{S}'$, $\mathsf{move}(D) = \mathsf{move}'(D)$, and for every $D \in \mathcal{S} \setminus \mathcal{S}'$, $\mathsf{move}(D) = D$. We claim that move is a solution to $(\mathcal{S}, k, d)$. To this end, first note that none of the unit disks in $\mathcal{P}'$ intersect each other (since move' is a solution to $(\mathcal{S}', k, d)$). In particular, the unit disks in $\{D \in U : \mathsf{move}(D) = D\}$ do not intersect any other unit disk in $\mathcal{P}'$. However, all unit disks in $\mathcal{S}$ that do not belong to U do not intersect each other (since U is a vertex cover for the intersection graph of $\mathcal{S}$). So, in $\mathcal{P}$, the only pairs of unit disks that can potentially intersect each other are pairs where one is a unit disk that was moved by move and the other belongs to $\mathcal{S} \setminus \mathcal{S}'$. However, the center of any unit disk D that is moved by move is at distance at most $(d+2) \cdot k$ from the center of at least one unit disk D' in U, and hence the center of $\mathsf{move}(D)$ is at distance at most $d + (d+2) \cdot k$ from the center of D', while the center of any unit disk in $\mathcal{S} \setminus \mathcal{S}'$ is at distance more than

---

[3] Note that $\mathcal{S}'$ may contains unit disks whose centers are at distance larger than $(d+2) \cdot k$ (but at most $(d+2) \cdot (k+1)$) from the centers of all unit disks in U.

$(d+2) \cdot (k+1)$ from the centers of all unit disks in U. Thus, $\mathcal{P}$ cannot have a pair of unit disks that intersect each other, such that one is a unit disk that was moved by move and the other belongs to $\mathcal{S} \setminus \mathcal{S}'$. So, move is indeed a solution to $(\mathcal{S}, k, d)$.

Now, note that for every $D \in U$, the unit disks whose center is at distance at most $(d+2) \cdot (k+1)$ from D are contained in a disk D' of radius $(d+2) \cdot (k+1)+1$ and whose center is the same as the center of D. So, by Observation 7 and since U is a vertex cover for the intersection graph of $\mathcal{S}$, this means that there exist at most $((d+2) \cdot (k+1) + 2)^2 = \mathcal{O}(d^2 k^2)$ unit disks in $\mathcal{S} \setminus U$ that intersect D'. As $|U| \leq 2k$, we conclude that $|\mathcal{S}'| \leq |U| + |U| \cdot ((d+2) \cdot (k+1) + 2)^2 = \mathcal{O}(d^2 k^3)$.   ◀

To reduce the bitsize of encoding the coordinates of the unit disks in the output instance, we make use of the following lemma.

▶ **Lemma 9.** *There exists a polynomial-time algorithm that, given a set $\mathcal{D}$ of unit disks whose centers have rational coordinates, a partition $(\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_\ell)$ of $\mathcal{D}$, and $r \in \mathbb{N}$, outputs a set $\mathcal{D}'$ of unit disks whose centers have rational coordinates and a bijective function $f : \mathcal{D} \to \mathcal{D}'$ with the following properties.*

- *For all $i \in \{1, 2, \ldots, \ell\}$, $\mathcal{D}_i$ and $\{f(D) : D \in \mathcal{D}_i\}$ are isometric, that is, for all $D, D' \in \mathcal{D}_i$, we have $\mathsf{distance}(D, D') = \mathsf{distance}(f(D), f(D'))$.*
- *For all distinct $i, j \in \{1, 2, \ldots, \ell\}, D \in \mathcal{D}_i$ and $D' \in \mathcal{D}_j$, we have $\mathsf{distance}(D, D') > r$.*
- *Encoding the coordinates (in unary) of all the unit disks in $\{f(D) : D \in \mathcal{D}_i\}$ requires space polynomial in $r, |\mathcal{D}|, m = \max_{i=1}^{\ell} \max_{D,D' \in \mathcal{D}_i} \mathsf{distance}(D, D')$ and $N = \max_{b,c}(b+c)$ over every $b, c \in \mathbb{N}, b < c$, and b, c are coprime, such that $a + \frac{b}{c}$ is a coordinate of a center of a unit disk in $\mathcal{D}$.*

Proof. For every $i \in \{1, 2, \ldots, \ell\}$, let L_i be a leftmost unit disk in $\mathcal{D}_i$ (i.e., with a smallest x-coordinate of its center), and let D_i be a bottommost unit disk in $\mathcal{D}_i$ (i.e., with a smallest y-coordinate of its center), and denote their centers by $(x_i^{\mathsf{left}}, y_i^{\mathsf{left}})$ and $(x_i^{\mathsf{bottom}}, y_i^{\mathsf{bottom}})$, respectively. Now, for every $i \in \{1, 2, \ldots, \ell\}$ and every $D \in \mathcal{D}_i$ with center (x, y), define $f(D)$ as the unit disk whose center is $(x - x_i^{\mathsf{left}} + (i-1) \cdot (m+r), y - y_i^{\mathsf{bottom}} + (i-1) \cdot (m+r))$. We define $\mathcal{D}'$ as the set of unit disks assigned by f. Clearly, $f : \mathcal{D} \to \mathcal{D}'$ is bijective and the third property in the lemma holds.

For the first property, consider two unit disks $D, D' \in \mathcal{D}_i$ for some $i \in \{1, 2, \ldots, \ell\}$ with centers (x, y) and (x', y'), respectively. Then, $\mathsf{distance}(f(D), f(D'))$ is equal to the square root of $((x - x_i^{\mathsf{left}} + (i-1) \cdot (m+r)) - (x' - x_i^{\mathsf{left}} + (i-1) \cdot (m+r)))^2 + ((y - y_i^{\mathsf{bottom}} + (i-1) \cdot (m+r)) - (y' - y_i^{\mathsf{bottom}} + (i-1) \cdot (m+r)))^2$, which is precisely $\sqrt{(x - x')^2 + (y - y')^2} = \mathsf{distance}(D, D')$. So, the first property in the lemma holds.

For the second property, consider two unit disks $D \in \mathcal{D}_i, D' \in \mathcal{D}_j$ for some $i, j \in \{1, 2, \ldots, \ell\}, i < j$, with centers (x, y) and (x', y'), respectively. Then, $\mathsf{distance}(f(D), f(D'))$ is equal to the square root of $((x' - x_j^{\mathsf{left}} + (j-1) \cdot (m+r)) - (x - x_i^{\mathsf{left}} + (i-1) \cdot (m+r)))^2 + ((y' - y_j^{\mathsf{bottom}} + (j-1) \cdot (m+r)) - (y - y_i^{\mathsf{bottom}} + (i-1) \cdot (m+r)))^2$. Observe that $x' \geq x_j^{\mathsf{left}}, y' \geq y_j^{\mathsf{bottom}}, x \leq x_i^{\mathsf{left}} + m, y \leq y_i^{\mathsf{bottom}} + m$. So, the above expression is lower bounded by

$$\sqrt{2 \left((j-1) \cdot (m+r) - (m + (i-1) \cdot (m+r)) \right)^2} = \sqrt{2} \cdot ((j-i)(m+r) - m) \geq \sqrt{2} r.$$

In particular, $\mathsf{distance}(f(D), f(D')) > r$. So, the second property in the lemma holds.   ◀

We will also need the following simple observation.

▶ **Observation 10.** *Let $\mathcal{S}$ be a set of unit disks in the Euclidean plane. Let $D \in \mathcal{S}$. Then, by moving D by a distance of at most some $d \in \mathbb{N}$, D cannot intersect unit disks whose centers are at distance at least $d + 2$ from the original position of the center of D.*

Based on Lemma 9 and Observation 10, we prove the following.

▶ **Lemma 11.** *There exists a polynomial-time algorithm that, given an instance $(\mathcal{S}, k, d)$ of DISK DISPERSAL where the centers of all disks have rational coordinates, and a partition $(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell)$ of $\mathcal{S}$ such that for all $i, j \in \{1, 2, \ldots, \ell\}, D \in \mathcal{S}_i$ and $D' \in \mathcal{S}_j$, we have $\mathsf{distance}(D, D') \geq 2d + 2$, outputs an equivalent instance of DISK DISPERSAL, respectively, with the same parameters k, d and number of unit disks, where encoding the coordinates of all the unit disks (in unary) requires space polynomial in $d, |\mathcal{S}|, m = \max\limits_{i=1}^{\ell} \max\limits_{D,D' \in \mathcal{S}_i} \mathsf{distance}(D, D')$ and $N = \max_{b,c}(b + c)$ over every $b, c \in \mathbb{N}, b < c$, such that $a + \frac{b}{c}$ is a coordinate of a center of a unit disk in $\mathcal{D}$.*

Proof. The algorithm simply applies the algorithm in Lemma 9 with $r = 2d + 2$, and obtains $f : \mathcal{S} \to \mathcal{S}'$. Then, it returns $\mathcal{D}'$. From Lemma 9, it directly follows that encoding the coordinates of all the unit disks requires space polynomial in $d, |\mathcal{D}|, m$ and N. Recall that for all $i, j \in \{1, 2, \ldots, \ell\}, D \in \mathcal{S}_i$ and $D' \in \mathcal{S}_j$, we have $\mathsf{distance}(D, D') \geq 2d + 2$, and this property is preserved under the mapping f (by our choice of r). So, Observation 10 implies that the sub-instances induced by the different sets $\mathcal{S}_i$ are "independent" from each other: we cannot move unit disks in one set $\mathcal{S}_i$ so that they intersect unit disks in another set $\mathcal{S}_j$. Also, the same holds for the sub-instances they are mapped to by f. As every sub-instance induced by some set $\mathcal{S}_i$ is equivalent to the sub-instance it is mapped to by f since $\mathcal{S}_i$ and $\{f(D) : D \in \mathcal{S}_i\}$ are isometric, we conclude that $(\mathcal{S}, k, d)$ and $(\mathcal{S}', k, d)$ are equivalent. ◀

We our now ready to present our (non-partial) kernel for DISK DISPERSAL. In particular, if N is a constant (or polynomial in $k + d$), the parameterization can be assumed to be only by $k + d$.

▶ **Theorem 12.** *DISK DISPERSAL, restricted to instances where the centers of all disks have rational coordinates, admits a polynomial kernel with respect to $k + d + N$, where $N = \max_{b,c}(b + c)$ over every $b, c \in \mathbb{N}, b < c$, such that $a + \frac{b}{c}$ is a coordinate of a center of a unit disk in $\mathcal{S}$.*

Proof. Given an instance $(\mathcal{S}, k, d)$ of DISK DISPERSAL, restricted to instances where the centers of all disks have rational coordinates, the kernelization algorithm works as follows. First, we call the algorithm in Theorem 1 to obtain an equivalent instance $(\mathcal{S}', k, d)$ of DISK DISPERSAL. Here, k, d remain unchanged, and $\mathcal{S}'$ is a subset of $\mathcal{S}$. Let $\mathcal{W} = \{W_D : D \in \mathcal{S}'\}$ where W_D is a disk whose center is the same as the center of D and whose radius is $d + 1$. Let $\mathcal{C}$ be the set of connected components of the intersection graph of $\mathcal{W}$. Let $\mathcal{P}$ be the partition of $\mathcal{S}'$ such that two unit disks in $\mathcal{S}'$ belong to the same part if and only if there exists a connected component in $\mathcal{C}$ such that both are intersected by (possibly different) disks that belong to that component. It should be clear, from the definitions of $\mathcal{S}'$ and $\mathcal{W}$, that this is indeed a partition, and that if two unit disks in $\mathcal{S}'$ belong to different parts in this partition, then the distance between their centers is larger than $2d + 2$. So, the kernelization algorithm then calls the algorithm in Lemma 11 on $(\mathcal{S}', k, d)$ and $\mathcal{P}$ as the partition of $\mathcal{D}'$, and returns its output. ◀

Lastly, based on Theorem 1 and Proposition 4, we prove Corollary 2 stating that DISK DISPERSAL is FPT when parameterized by $d + k$. We restate the theorem here.

▶ **Corollary 2.** DISK DISPERSAL *is* FPT *when parameterized by* $d + k$. *Specifically, it is solvable in time* $(dk)^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$.

Proof. Given an instance $(\mathcal{S}, k, d)$ of DISK DISPERSAL, the algorithm first calls the algorithm in Theorem 1 to obtain (in polynomial time) an equivalent instance $(\mathcal{S}', k, d)$ of DISK DISPERSAL, where $\mathcal{S}' \subseteq \mathcal{S}$ is of size $\mathcal{O}(d^2 k^3)$. Then, for every $\mathcal{A} \subseteq \mathcal{S}'$ of size at most k such that $\mathcal{S}' \setminus \mathcal{A}$ is a packing, the algorithm tests whether it is possible to move each unit disk in $\mathcal{A}$ by a distance of at most d so that, afterwards, $\mathcal{S}'$ becomes a packing. This can be done by using the algorithm in Proposition 4 to solve the following system of polynomial inequalities, which has variables x_A, y_A for every $A \in \mathcal{A}$:

- For every $S \in \mathcal{S}' \setminus \mathcal{A}$ and $A \in \mathcal{A}$: $(x_A - a)^2 + (y_A - b)^2 \geq 4$, where (a, b) denotes the center of S.
- For every distinct $A_1, A_2 \in \mathcal{A}$: $(x_{A_1} - x_{A_2})^2 + (y_{A_1} - y_{A_2})^2 \geq 4$.
- For every $A \in \mathcal{A}$, where (a, b) denotes the center of A in $\mathcal{S}'$: $(x_A - a)^2 + (y_A - b)^2 \leq d^2$.

The correctness of the algorithm is immediate. For its running time analysis, notice that there are only $\sum_{i=0}^{k} \binom{|\mathcal{S}'|}{i} \leq (dk)^{\mathcal{O}(k)}$ choices for $\mathcal{A}$. Further, each of the systems of polynomial equations that are solved has at most $2k$ variables, degree 2, and $\mathcal{O}(|\mathcal{A}| \cdot |\mathcal{S}'|) \leq (dk)^{\mathcal{O}(1)}$ equations. So, by Proposition 4, it is solvable in time $(dk)^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$. In turn, we conclude that the algorithm runs in time $(dk)^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$. ◀

4 Kernelization lower bound for Disk Dispersal

In this section, we prove Theorem 3. To this end, we show that from several instances of DISK APPENDING (defined below), we can construct a single instance $\mathcal{I}'$ of DISK DISPERSAL such that there is a solution to $\mathcal{I}'$ if and only if there is a solution to at least one of the instances of DISK APPENDING. The result then follows from the cross-composition technique (see [17], Chapter 17 for more details). DISK APPENDING is defined as follows.

DISK APPENDING

Input: A packing $\mathcal{P}$ of n unit disks inside a rectangle R and an integer $\kappa \geq 0$.

Task: Decide whether there is a packing $\mathcal{P}^*$ of $n + \kappa$ unit disks inside R obtained from $\mathcal{P}$ by adding κ new disks.

A recent result of Fomin et al. [15, 16] shows that the problem is NP-hard. In particular, they show the following result.

▶ **Proposition 13** (Corollary 2 in [15]). DISK APPENDING *is* NP-*hard. Furthermore, it remains* NP-*hard, even when restricted to instances* $(R, \mathcal{P}, \kappa)$ *of the following form.*
- *Rectangle* R *is* $[0, 2a] \times [0, 2b]$ *for integers* $a, b > 0$. *It can also be assumed that* $a = b$.
- *A packing* $\mathcal{P}$ *of disks with their centers inside* R *such that (i) for every* $i \in \{0, \ldots, a\}$, *the disks with centers* $(2i, 0)$ *and* $(2i, 2b)$ *are in* $\mathcal{P}$ *and (ii) for every* $j \in \{0, \ldots, b\}$, *the disks with centers* $(0, 2j)$ *and* $(2a, 2j)$ *are in* $\mathcal{P}$.

Proof of Theorem 3. The reader may wish to refer to Figure 4, which explains the schematics of the reduction. We consider instances $(R, \mathcal{P}, n, \kappa)$ of DISK APPENDING, where R is an $[0, a] \times [0, a]$ square, where a is an even positive integer, $\mathcal{P}$ is a packing of n disks with their centers inside R, such that the centers of the disks are rational, and κ is the number of disks that need to be added inside R, which is compatible with $\mathcal{P}$, to obtain a packing of $n + k$ disks. We also assume that for every $i \in \{1, \ldots, a/2\}$, the disks with centers $(2i - 1, 1)$, $(2i - 1, a - 1)$, $(1, 2i - 1)$ and $(a - 1, 2i - 1)$ are in $\mathcal{P}$.

For the cross-composition, we first show the polynomial equivalence relation $\mathcal{R}$, over instances $(R_i, \mathcal{P}_i, n_i, \kappa_i)$ of DISK APPENDING. The instances $(R_i, \mathcal{P}_i, n_i, \kappa_i)$ and $(R_j, \mathcal{P}_j, n_j, \kappa_j)$ go to the same equivalence classes if (1) the squares R_i and R_j have the same dimension, (2) $\mathcal{P}_i$ and $\mathcal{P}_j$ is a packing of $n_i = n_j$ disks inside R_i and R_j respectively with centers having rational coordinates, and (3) $\kappa_i = \kappa_j$. All the other malformed instances go into another equivalence class (see [17] for the formal requirements of the equivalence relation). Note that $\mathcal{R}$ satisfies the properties of polynomial equivalence relation, since the equivalence can be checked in polynomial time, and (2) $\mathcal{R}$ partitions the elements of S into at most $(\max_{x \in S} |x|)^{\mathcal{O}(1)}$ classes in a well-formed instance, since $\kappa_i \leq n_i$ (this can be assumed w.l.o.g. by padding the instance as required, as per Proposition 13).

Now we give a cross-composition algorithm for instances belonging to the same equivalence class. For the last equivalence class of malformed instances, we output a trivial no-instance. Thus, from now on, we focus on an equivalence class $(R_1, \mathcal{P}_1, n, \kappa), \dots, (R_t, \mathcal{P}_t, n, \kappa)$, such that a is the sidelength of every square $R_1, \dots, R_t$. We assume w.l.o.g. that t is odd, and a is an even integer that is at least 10κ.

For every $1 \leq i \leq t$, we construct a gadget G_i as follows; see Figure 4. Let R be a rectangle of height 6 and width $2\kappa + 6$. Suppose the cartesian coordinates of the bottom-left corner of G_i are $(0, 0)$ (note that this coordinate system is defined only for explaining the gadget structure, and should not be confused with the coordinate system in the next paragraph). Then, we place $2(k + 3)$ disks centered at points $(1, 1), (3, 1), \dots, (2k + 5, 1)$, as well as $(1, 5), (3, 5), \dots, (2\kappa + 5, 5)$, and 2 additional disks centered at $(1, 3)$, and $(2\kappa + 5, 3)$. These disks lie along the perimeter of the rectangle, with centers at distance 1 from the perimeter. We call these disks *surrounding disks* (shown in green). Additionally, we place κ disks with centers at $(4, 3), (6, 3), \dots, (2\kappa + 2, 3)$, which are termed as *interesting disks* (shown in blue). Note that this leaves a horizontal gap of 1 between the leftmost (resp. rightmost) interesting disk and the surrounding disks with center $(1, 3)$ (resp. $(2\kappa + 5, 3)$). Now, we pad the gadget horizontally by adding columns of 3 disks on both sides of the surrounding disks in a symmetric manner, such that the width of the gadget becomes exactly a.

Now we describe the construction of the instance of DISK DISPERSAL. It might be useful to refer to a schematic description shown in Figure 4. Let d, the distance by which a disk can be moved, be equal to $\frac{9}{4} t^2 a^2$. We place the first square R_1 and the corresponding packing of disks $\mathcal{P}_1$ from the first instance by placing the bottom-left of R_1 corner at the origin $(0, 0)$. Next, we place the instances $(R_2, \mathcal{P}_2), (R_3, \mathcal{P}_3), \dots, (R_t, \mathcal{P}_t)$ by aligning their bottom edge along the x-axis, and leaving a horizontal gap of $s := \sqrt{2ad}$ between the adjacent squares. Then, we place the gadgets G_i directly above the rectangle R_i such that the vertical distance between the top edge of R_i and the top edge of G_i is equal to $h := \sqrt{d^2 - a^2}$. Since the width of every gadget G_i is equal to a after padding, the vertical boundaries of R_i and the corresponding G_i are aligned. Next, we place a set C of $\kappa + 2$ co-located disks such that (1) the vertical distance between the bottom edge of $G_{(t+1)/2}$ and the centers of disks in C is equal to $d/2$, and (2) the centers of the disks in C are aligned with the horizontal center of the gadget G_i. We place a rectangle tightly enclosing the instance constructed thus far, and pack all the empty spaces outside the gadgets using disks with integral coordinates on the centers (not shown in the figure). Finally, we set the budget k, the number of disks that can be moved, to be $(\kappa + 1) + \kappa = 2\kappa + 1$. This finishes the construction of the instance of DISK DISPERSAL.

The proof of the following claim follows from the careful choice of d, h and s in terms of a and t.
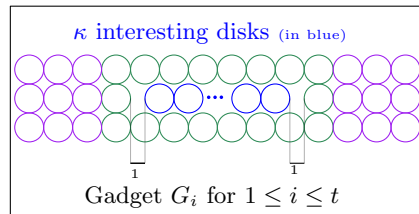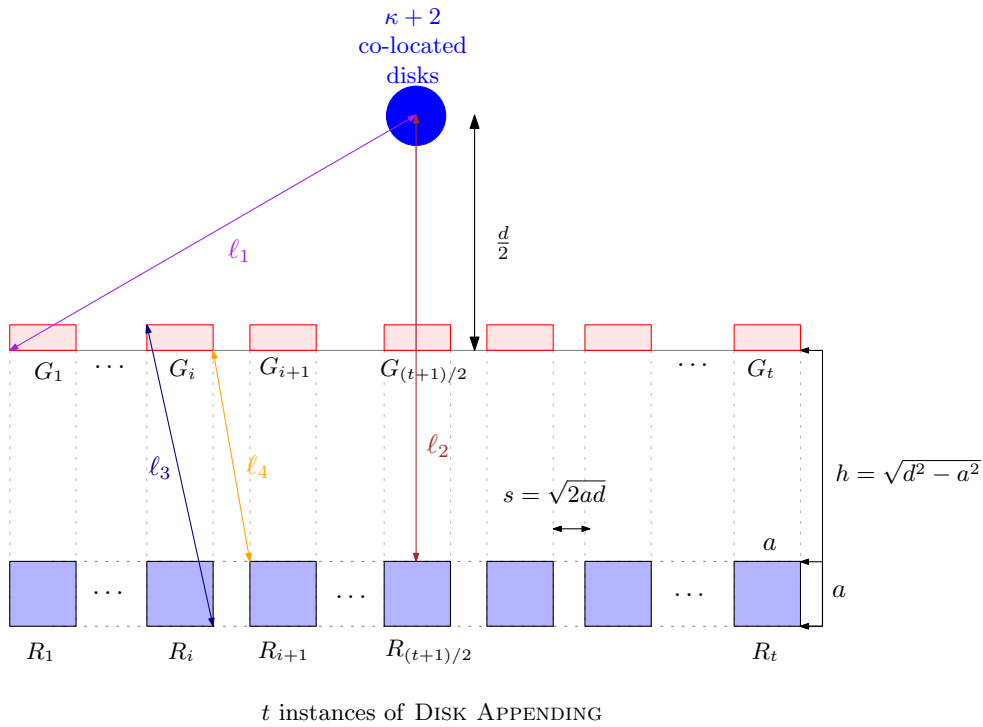
Figure 4 Schematic depiction of an instance of DISK DISPERSAL obtained by OR-composition of instances $(R_i, \mathcal{P}_i, n, \kappa)$ of DISK APPENDING. Each instance $(R_i, \mathcal{P}_i, n, \kappa)$ is shown in a blue square of sidelength a. Red rectangles are gadgets G_i, and an example gadget is shown below. Lengths $\ell_1, \ell_2, \ell_3, \ell_4$ are defined in Claim 14, and the values of s, h, and d are carefully chosen functions of t and a, in order to ensure that $\ell_1, \ell_3 \leq d < \ell_2, \ell_4$ (note that the figure is not to scale). All the empty spaces are filled with padding disks with integral coordinates of centers. This ensures that an interesting disk from G_i cannot be moved into an adjacent $R_{i\pm1}$, and thus different instances remain "isolated". In the gadget G_i, the *surrounding disks* are shown in green and *interesting disks* are shown in blue. Finally, purple disks are added on either side of the gadget in order to make the total width of the gadget exactly a. Then, the gadget G_i and the corresponding square R_i can be horizontally aligned as shown in the figure.

▷ Claim 14.
1. The maximum distance between the centers of disks in C and any point in any G_i is at most d (shown as ℓ_1 in Figure 4).
2. The minimum distance between the centers of disks in C and any point in any R_i is more than d (ℓ_2).
3. The maximum distance between a point in G_i and a point in the corresponding R_i is at most d (ℓ_3).
4. The minimum distance between a point in G_i and a point in another R_j is more than d (ℓ_4).

Proof. The values of d, h, and s are chosen carefully in terms of a and t in order to ensure these properties. First we observe that $s = \sqrt{2ad} \geq a$, since $d = \frac{9}{4} t^2 a^2$.

1. Note that the horizontal distance between the midpoint of $G_{(t+1)/2}$ and the leftmost point in G_1 can be upper bounded by $(t/2)(a+s) \leq (t/2) \cdot (2s) = t \cdot \sqrt{2ad}$. The vertical distance between the bottom edge of $G_{(t+1)/2}$ and the centers of disks in C is $d/2$. Therefore, it suffices to show that $\left(\frac{d}{2}\right)^2 + \left(t\sqrt{2ad}\right)^2 \leq d^2$, i.e., $t^2 \cdot 2ad \leq \frac{3d^2}{4}$, i.e., $2at^2 \leq \frac{3}{4} \cdot \frac{9}{4} \cdot t^2 a^2$. This holds assuming $a \geq 216$.

2. It suffices to consider the vertical distance between the centers of C and the top edge of $R_{(t+1)/2}$. This vertical distance is $\frac{d}{2} + h - a$, which we want to show is greater than d. Note that it suffices to show that $h = \sqrt{d^2 - a^2} > \frac{d}{2}$, i.e., $a^2 < \frac{3d^2}{4}$, i.e., $\frac{243}{64} t^4 a^2 > 1$. However, since $t, a \geq 1$, this is true.

3. $\ell_3^2 = h^2 + a^2 = d^2 - a^2 + a^2 = d^2$, since $h = \sqrt{d^2 - a^2}$.

4. It suffices to consider adjacent G_i, R_{i+1} pairs (argument for R_{i-1} is identical). Then, $\ell_4^2 = (h-a)^2 + s^2 = (\sqrt{d^2 - a^2} - a)^2 + 2ad = d^2 - 2a\sqrt{d^2 - a^2} + 2ad$, which we want to show is at least d^2. This holds since $d > \sqrt{d^2 - a^2}$. $\triangleleft$

Now we explain the implications of Claim 14. In any yes-instance, at least $\kappa + 1$ disks from C must be moved by a distance at most d. Let C' be this set of disks from the set of $\kappa + 2$ co-located disks, that are moved. Note that in any gadget G_i, if all the κ interesting disks are moved, then this creates an available space for placing $\kappa + 1$ disks of C'. On the other hand, if any set of fewer than κ disks inducing a connected component in the *contact graph* (i.e., a special kind of intersection graph wherein there is an edge between the vertices corresponding to two disks iff their boundaries touch each other) of the disks is moved, then this creates space for at most κ disks from C' (note that the distance between C' and an R_i is more than d by item 2 of Claim 14). However, since the budget is $2\kappa + 1$, this cannot correspond to a feasible solution. Thus, in a solution to a yes-instance, C' can only be moved in the place of κ interesting disks corresponding to a gadget G_i. Next, an interesting disk can be moved anywhere in the corresponding square R_i (item 3), but cannot be moved to a different square R_j (item 4). Then, using an argument used for the disks in C', we conclude that the k interesting disks can only be moved in the empty spaces in the corresponding R_i. Thus, the created instance of DISK DISPERSAL is a yes-instance iff there exists some yes-instance $(R_i, \mathcal{P}_i, n, \kappa)$ of DISK APPENDING. Finally, we note that Proposition 13 implies that the coordinates of the centers of the disks in each instance of DISK DISPERSAL can be assumed to be rational. Furthermore, by letting $s \approx \sqrt{2ad}$, and $h \approx \sqrt{d^2 - a^2}$ as rational approximations of their original values with small enough error, we can ensure that the coordinates of all the centers of the disks in the constructed instance become rational, and furthermore, the inequalities from Claim 14 continue to hold. This concludes the proof of Theorem 3.   ◀

5   Conclusion and Open Problem

In this paper, we initiate the study of the problem of spreading points from the perspective of parameterized complexity and kernelization. We reformulate the problem in terms of moving at most k unit disks by a distance of at most d, which we call DISK DISPERSAL. We design a (partial) polynomial kernel for DISK DISPERSAL parameterized by k and d. Furthermore, we show that this can be transformed into a (true) kernel, assuming the coordinates of the centers of the unit disks are rational numbers with bounded denominators. We complement this result by showing that DISK DISPERSAL does not admit a polynomial kernel parameterized by k alone, assuming coNP $\subseteq$ NP$/$poly. These results provide a complete picture of the kernelization complexity of DISK DISPERSAL.

We show that DISK DISPERSAL is FPT parameterized by $k+d$, by combining the (partial) kernel with a non-trivial subroutine that involves solving a system of polynomial inequalities. It is natural to ask whether the problem is fixed-parameter tractable by the individual parameters d and k. Fiala et al. [14] have shown that DISK DISPERSAL is NP-hard even when $d = 2$. However, the parameterized complexity of DISK DISPERSAL parameterized by k alone remains open. We make some preliminary progress in this direction. In the full version of the paper, we prove that the *rectilinear* version of DISK DISPERSAL is W[1]-hard when parameterized by k. This is a constrained version of DISK DISPERSAL, called RECTILINEAR DISK DISPERSAL, which is defined as follows.

RECTILINEAR DISK DISPERSAL

Input: A family $\mathcal{S}$ of n unit disks, an integer $k \geq 0$, and a real $d \geq 0$.

Task: Decide whether it is possible to obtain from $\mathcal{S}$ a family of non-overlapping disks $\mathcal{P}$ by moving at most k disks into new positions parallel to the axes in such a way that each disk is moved at distance at most d.

By examining our algorithmic results, namely, the (partial) kernels and the FPT algorithm parameterized by $k + d$ also hold for RECTILINEAR DISK DISPERSAL. Thus we have a complete picture of the complexity of RECTILINEAR DISK DISPERSAL, with parameters k and d. Given this state of affairs, we conjecture that DISK DISPERSAL is also W[1]-hard when parameterized by k.

References

1 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*. Springer, Berlin, Heidelberg, 2009.

2 Christoph Baur and S'andor P Fekete. Approximation of geometric dispersion problems. *Algorithmica*, 30(3):451–470, 2001.

3 Therese Biedl, Anna Lubiw, Anurag Murty Naredla, Peter Dominik Ralbovsky, and Graeme Stroud. Distant Representatives for Rectangles in the Plane. In *29th Annual European Symposium on Algorithms (ESA)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ESA.2021.17`.

4 Sergio Cabello. Approximation algorithms for spreading points. *J. Algorithms*, 62(2):49–73, 2007. `doi:10.1016/j.jalgor.2004.06.009`.

5 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. `doi:10.1007/978-3-319-21275-3`.

6 Erik D. Demaine, Mohammad Taghi Hajiaghayi, Hamid Mahini, Amin S. Sayedi-Roshkhar, Shayan Oveis Gharan, and Morteza Zadimoghaddam. Minimizing movement. *ACM Trans. Algorithms*, 5(3):30:1–30:30, 2009. `doi:10.1145/1541885.1541891`.

7 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Dániel Marx. Minimizing movement: Fixed-parameter tractability. *ACM Trans. Algorithms*, 11(2):14:1–14:29, 2014. `doi:10.1145/2650247`.

8 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

9 Srinivas Doddi, Madhav V. Marathe, Andy Mirzaian, Bernard M. E. Moret, and Binhai Zhu. Map labeling and its generalizations. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 148–157. ACM/SIAM, 1997. URL: `http://dl.acm.org/citation.cfm?id=314161.314250`.

10   Adrian Dumitrescu and Minghui Jiang. Dispersion in unit disks. In *27th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 5 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 299–310, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.STACS.2010.2464`.

11   Adrian Dumitrescu and Minghui Jiang. Constrained k-center and movement to independence. *Discret. Appl. Math.*, 159(8):859–865, 2011. `doi:10.1016/j.dam.2011.01.008`.

12   Adrian Dumitrescu and Minghui Jiang. Dispersion in disks. *Theory Comput. Syst.*, 51(2):125–142, 2012. `doi:10.1007/s00224-011-9331-x`.

13   S'andor P Fekete and Henk Meijer. Maximum dispersion and geometric maximum weight cliques. *Algorithmica*, 38(3):501–511, 2004.

14   Jirí Fiala, Jan Kratochvíl, and Andrzej Proskurowski. Systems of distant representatives. *Discret. Appl. Math.*, 145(2):306–316, 2005. `doi:10.1016/j.dam.2004.02.018`.

15   Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, Saket Saurabh, and Meirav Zehavi. (re)packing equal disks into rectangle. *CoRR*, abs/2211.09603, 2022. `doi:10.48550/ARXIV.2211.09603`.

16   Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, and Meirav Zehavi. (re)packing equal disks into rectangle. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 60:1–60:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.ICALP.2022.60`.

17   Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of parameterized preprocessing*. Cambridge University Press, Cambridge, 2019.

18   Tien-Ruey Hsiang, Esther M Arkin, Michael A Bender, Sandor Fekete, and Joseph SB Mitchell. Online dispersion algorithms for swarms of robots. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 382–383, 2003.

19   Tien-Ruey Hsiang, Esther M Arkin, Michael A Bender, S'andor P Fekete, and Joseph SB Mitchell. Algorithms for rapidly dispersing robot swarms in unknown environments. In *Algorithmic Foundations of Robotics V*, pages 77–93. Springer, 2004.

20   Minghui Jiang. A new approximation algorithm for labeling points with circle pairs. *Inf. Process. Lett.*, 99(4):125–129, 2006. `doi:10.1016/j.ipl.2006.04.006`.

21   Minghui Jiang, Jianbo Qian, Zhongping Qin, Binhai Zhu, and Robert J. Cimikowski. A simple factor-3 approximation for labeling points with circles. *Inf. Process. Lett.*, 87(2):101–105, 2003. `doi:10.1016/S0020-0190(03)00256-4`.

22   Maarten Löffler and Marc J. van Kreveld. Largest bounding box, smallest diameter, and related problems on imprecise points. *Comput. Geom.*, 43(4):419–433, 2010. `doi:10.1016/j.comgeo.2009.03.007`.

23   Farnaz Sheikhi, Ali Mohades, Mark de Berg, and Ali D. Mehrabi. Separability of imprecise points. *Comput. Geom.*, 61:24–37, 2017. `doi:10.1016/j.comgeo.2016.10.001`.