Parameterized Geometric Graph Modification with Disk Scaling

Fedor V. Fomin ✉ 🄳
University of Bergen, Norway

Petr A. Golovach ✉ 🄳
University of Bergen, Norway

Tanmay Inamdar ✉ 🄳
Indian Institute of Technology Jodhpur,India

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

Meirav Zehavi ✉ 🄳
Ben-Gurion University of the Negev, Beer-Sheva, Israel

────── **Abstract** ──────

The parameterized analysis of graph modification problems represents the most extensively studied area within Parameterized Complexity. Given a graph G and an integer $k \in \mathbb{N}$ as input, the goal is to determine whether we can perform at most k operations on G to transform it into a graph belonging to a specified graph class $\mathcal{F}$. Typical operations are combinatorial and include vertex deletions and edge deletions, insertions, and contractions. However, in many real-world scenarios, when the input graph is constrained to be a geometric intersection graph, the modification of the graph is influenced by changes in the geometric properties of the underlying objects themselves, rather than by combinatorial modifications. It raises the question of whether vertex deletions or adjacency modifications are necessarily the *most appropriate* modification operations for studying modifications of geometric graphs.

We propose the study of the disk intersection graph modification through the scaling of disks. This operation is typical in the realm of topology control but has not yet been explored in the context of Parameterized Complexity. We design parameterized algorithms and kernels for modifying to the most basic graph classes: edgeless, connected, and acyclic. Our technical contributions encompass a novel combination of linear programming, branching, and kernelization techniques, along with a fresh application of bidimensionality theory to analyze the area covered by disks, which may have broader applicability.

1 Introduction

In a *graph modification problem*, the input consists of an n-vertex graph G and an integer k. The objective is to determine whether *k modification operations* – such as vertex deletions, or edge deletions, insertions or contractions, or combinations thereof – are sufficient to obtain a graph with prescribed structural properties such as being planar, bipartite, chordal, interval, acyclic or edgeless. Graph modification problems encompass some of the fundamental challenges in graph theory and graph algorithms. They represent the most intensively studied area of research within Parameterized Complexity [23, 20, 32, 47, 8, 13, 14, 9, 2, 36, 3, 27, 11, 38, 37] (this list is illustrative rather than comprehensive; see [19, 23, 20, 32] for more information). Here, the number of allowed modifications, k, is considered a *parameter*. With respect to k, we seek a *fixed parameter tractable* (FPT) algorithm, namely, an algorithm whose running time has the form $f(k)n^{\mathcal{O}(1)}$ for some computable function f. It would not be an exaggeration to state that the design of FPT algorithms for graph modification problems has played one of the most central roles in the development of Parameterized Complexity as a field.

In recent years, parametrized analysis of classic graph problems restricted to geometric intersection graphs – that is, where the input graph G belongs to a prespecified class of geometric intersection graphs – has gained a lot of interest. Naturally, particular attention has been given to graph modification problems restricted to geometric intersection graphs. For example, we refer to [4, 41, 42, 33, 35, 31, 21, 29, 45, 28, 22, 10, 30]. While most classic NP-hard graph problems become polynomial-time solvable on proper interval, interval and circular arc graphs, they remain NP-hard on most other geometric intersection graph classes of interest, which gives rise to their parameterized analysis (see the aforementioned papers). Geometric intersection graph classes studied in this context include, for example, unit disk and disk graphs, unit square and square graphs, convex polygon and polygon graphs, map graphs, string graphs, and the generalization of these classes to higher dimensions. However, one central question arises:

> Are vertex deletion and edge deletion/insertion/contraction the *right* modification operations to use when dealing with graph modification problems on geometric intersection graphs?

Geometric intersection graphs naturally give rise to modification operations that are *geometric* – specifically, we modify the graph by modifying its geometric structure! Arguably, the most fundamental operations in this regard are *scaling* and *shifting (i.e., moving)* the input geometric objects.

> In this paper, our primary objective is to launch a systematic investigation into parameterized graph modification problems concerning geometric intersection graphs, utilizing the operation of scaling the input geometric structure. We regard a significant aspect of our contribution as both conceptual and foundational, with the goal of paving the way for this emerging research direction.

1.1   Topology Control and Disk Scaling as a Modification Operation

Topology Control is a fundamental technique used in wireless ad hoc network and sensor networks to reduce energy consumption and radio interference [46]. This technique is used to control the topology of the graph representing the communication links between network nodes. The usual goal is to maintain or achieve some global graph property (like being connected), or reduce interference (i.e., obtain an edgeless graph) while reducing energy consumption. The geometric operation of scaling – i.e., shrinking or expanding radii – of disks has received significant attention within research of wireless network design due to its relevance to power consumption and interference in the networks. From a theoretical perspective, there is also a large body of literature on algorithmic methods for range assignment in wireless sensor and ad-hoc networks, see [39, 17, 12, 15, 48] for a (very incomplete) sample of various approaches, and it is also of interest for other purposes (see, e.g., [1, 24, 7]). Due to this significance, we consider problems that use disk scaling (i.e., shrinking or expanding a subset of disks) in order to achieve certain topological properties in graph. Our work in fact fits in the larger framework of geometric graph modification, which has been a growing trend in recent years.

Recent work on geometric graph modification.   Recently, Fomin et al. [26, 25] have studied some problems related to (re)packing of unit disks and point spreading from the perspective of parameterized complexity. Taking a broader perspective, these problems can also be seen as working in the model of geometric graph modification via *geometric operations*, also studied in [17, 5, 16]. Indeed, these papers consider *movement* of points or disks as an operation to achieve intersection graph that is edgeless. In this paper, we focus on *scaling* (i.e., shrinking or expanding) as the operation, and in addition to edgelessness, we also consider acyclicity and connectivity. Another common theme with the work of [25] is that they also combine enumeration of partial solutions from the (partial) kernel, and then reduce the problem to checking a system of polynomial inequalities. This high level scheme is similar to the one described for designing FPT algorithms for MIN k-SHRINKING TO INDEPENDENCE/ACYCLICITY, where we use an LP. But this similarity is only superficial, and seems unavoidable due to the inherently *continuous nature* of the problems. The specific context, and in particular, the kind of arguments used to arrive at a point where we can use LP, are quite involved, and different from that in [25], as we will explain this further in the following section. Nevertheless, we hope that using geometric operations for graph modification opens up a new research direction, and we return to this topic in the conclusion. Next, we formally introduce our model and problem definitions, and subsequently delve into the overview of our results.

1.2   Problem Definitions

We fix some notation. Let P be a finite family of points in the plane, and $r : P \to \mathbb{R}_{\geq 0}$ be an assignment of points to radii. Then, the disk graph associated with P and R is defined as follows. Let $\mathcal{D}(P, r)$ be a set that contains the disk centered at p with radius $r(p)$ for each $p \in P$, and let $G(P, r)$ denote the disk graph corresponding to $\mathcal{D}(P, r)$, i.e., a graph where we have a vertex for each point in P, and we have an edge between two points if and only if their corresponding disks intersect. Let $\mathbf{1} : P \to \{1\}$ denote the unit radius assignment, and let $G(P, \mathbf{1})$ denote the *unit disk graph* corresponding to a set of points P.
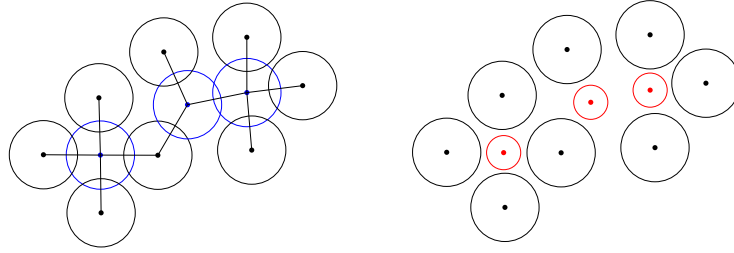
■ **Figure 1** Left: Original intersection graph $G = G(P, \mathbf{1})$ denoting an instance $(P, 1/2, 3)$ of k-SHRINKING TO INDEPENDENCE. Right: Edgeless graph $G(P, r)$ that shrinks three disks to radius $1/2$ (shown in red).

1.2.1   Shrinking to Achieve Edgelessness and Acyclicity

The first set of problems is related to shrinking a subset of disks in order to achieve two fundamental graph properties, namely, that of being edgeless and acyclic. The general setting is as follows. We are given a unit disk graph $G(P, \mathbf{1})$ corresponding to a set of n points $P \subset \mathbb{R}^2$. We want to *shrink* the disks corresponding to a subset $S \subseteq P$ (i.e., change the radii of disks of each point in S to a value smaller than 1), such that the resulting intersection graph induces an independent set (i.e., is edgeless). We consider two variants of this setting. First, we have a cardinality constrained problem, called k-SHRINKING TO INDEPENDENCE (see Figure 1).

---

k-SHRINKING TO INDEPENDENCE
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $0 \leq \alpha \leq 1$, and an integer $k \geq 0$.
Task: Decide whether there exists a solution (S, r) such that
   1. $S \subseteq P$ is a set of size **at most** k corresponding to shrunken disks,
   2. corresponding radii $r : P \to \mathbb{R}_{\geq 0}$ such that $r(p) = \begin{cases} 1 & \text{if } p \notin S \\ \alpha & \text{if } p \in S \end{cases}$
   3. $G(P, r)$ is edgeless

We also consider a cost-minimization version of the problem, called MIN k-SHRINKING TO INDEPENDENCE, where the cost of a solution is the total change in the radii, or the sum of *shrinking factors*.

MIN k-SHRINKING TO INDEPENDENCE
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $0 \leq \alpha \leq 1$, an integer $k \geq 0$, and budget $\mu \geq 0$.
Task: Decide whether there exists a solution (S, r) such that
   1. $S \subseteq P$ is a set of size **at most** k corresponding to shrunken disks,
   2. corresponding radii $r : P \to \mathbb{R}_{\geq 0}$ where $\begin{cases} r(p) = 1 & \text{if } p \notin S \\ \alpha \leq r(p) \leq 1 & \text{if } p \in S \end{cases}$
   3. $G(P, r)$ is edgeless, and
   4. $\text{cost}(S, r) := \sum_{p \in S} (1 - r(p)) \leq \mu$.

Analogous to k-SHRINKING TO INDEPENDENCE, we define k-SHRINKING TO ACYCLICITY, where the goal is to achieve $G(P, r)$ that is *acyclic* instead of edgeless (as in item 3 in the definition) (see Figure 2 for an example). MIN k-SHRINKING TO ACYCLICITY is the
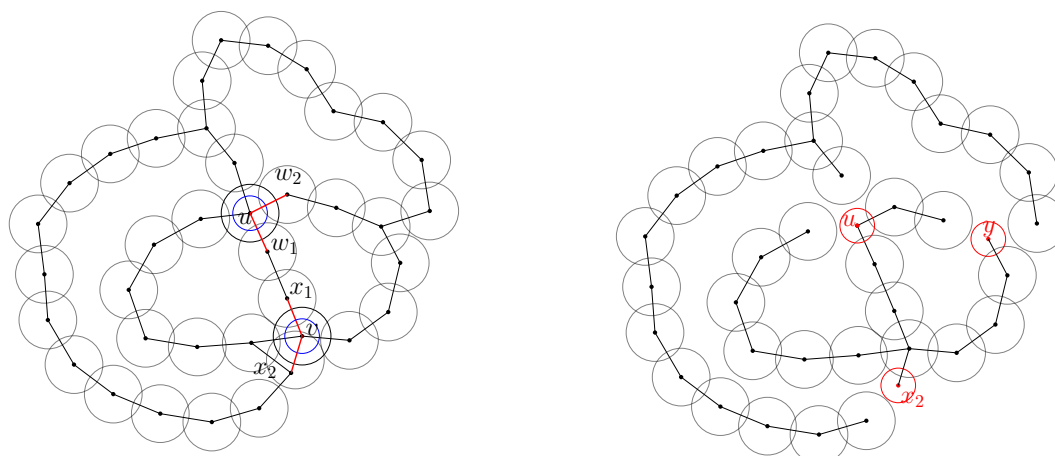
Figure 2 Left: Original intersection graph $G = G(P, \mathbf{1})$ corresponding to an instance $(P, 1/2, 3)$ of k-Shrinking to Acyclicity. Note that $\{u, v\}$ is a minimum feedback vertex set for G. However, even after shrinking the corresponding disks to $1/2$ (shown in blue), the edges $\{uw_1, uw_2, vx_1, v_2\}$ (shown in red) are present in the resulting intersection graph, showing that shrinking disks is not equivalent to vertex deletion. Right: A solution that shrinks three disks (shown in red) to radius $1/2$ corresponding to u, x_2, y, resulting in an acyclic graph $G(P, r)$.

"cost-minimization" version (similar to Min k-Shrinking to Independence), where we want to determine whether we can shrink at most k disks to radii in $[\alpha, 1]$ in order to achieve an acyclic subgraph, such that $\mathsf{cost}(S, r) \le \mu$.

Note that the operation of shrinking a disk to radius 0 is not necessarily equivalent to deleting the corresponding vertex. Thus, even when $\alpha = 0$, k-Shrinking to Acyclicity (resp. k-Shrinking to Independence) is not exactly equivalent to finding a feedback vertex set (resp. vertex cover) of size k in the original unit disk graph. Nevertheless, we will make use of this connection between the two problems. Now, we turn to the second class of problems considered in this paper.

1.2.2 Shrinking/Expanding to Connectivity

Connectedness is another fundamental graph property in topology control. We consider two variants: shrinking *at least k* disks while *retaining connectivity*, and expanding *at most k* disks to *achieve connectivity*.[1] Next, we define the two problems.

In the first problem, called k-Shrinking to Connectivity[2], we are given a connected unit disk graph $G(P, \mathbf{1})$ corresponding to a set of n points $P \subset \mathbb{R}^2$. We want to *shrink* the disks corresponding to a subset $S \subseteq P$ of size *at least k* (i.e., change the radii of disks corresponding to S to a value smaller than 1), such that the resulting intersection graph *remains* connected.

---

[1] Observe that expansion versions do not make sense for the problems considered in the earlier section, where we want to obtain an edgeless/acyclic graph – if the graph is not already edgeless/acyclic, one cannot achieve the desired property by expanding a disk.

[2] Ideally, the problem should be named k-Shrinking *While Retaining* Connectivity. However, for the sake of brevity, we stick with k-Shrinking to Connectivity.
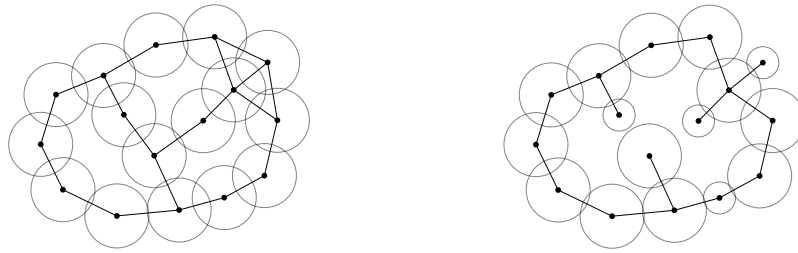
k-SHRINKING TO CONNECTIVITY
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $0 \le \alpha \le 1$, and a parameter $k \ge 0$.
Task: Decide whether there exists a solution (S, r) such that

- $S \subseteq P$ is a set of size **at least** k corresponding to shrunken disks,

- corresponding radii $r : P \to \mathbb{R}_{\ge 0}$ such that $r(p) = \begin{cases} 1 & \text{if } p \notin S \\ \alpha & \text{if } p \in S \end{cases}$

- $G(P, r)$ is connected.

Figure 3 Left: Original intersection graph $G = G(P, \mathbf{1})$ corresponding to an instance $(P, 1/2, 4)$ of k-SHRINKING TO CONNECTIVITY. Right: A solution that shrinks 4 disks to radius $1/2$ while maintaining connectivity in the resulting intersection graph $G(P, r)$.

The next problem, which we call k-EXPANDING TO CONNECTIVITY, is in a sense, the "complement version" of k-SHRINKING TO INDEPENDENCE. Here, we are given a unit disk graph $G(P, \mathbf{1})$ corresponding to a set of n points $P \subseteq \mathbb{R}^2$. We want to *expand* the disks corresponding to a subset $S \subseteq P$ of size at most k (i.e., change the radii of the disks corresponding to S to a value larger than 1), such that the resulting intersection graph *becomes* connected. Formally, the problem is defined as follows.

k-EXPANDING TO CONNECTIVITY
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $\alpha \ge 1$, and a parameter $k \ge 0$.
Task: Decide whether there exists a solution (S, r) such that

- $S \subseteq P$ is a set of size **at most** k corresponding to expanded disks,

- corresponding radii $r : P \to \mathbb{R}_{\ge 0}$ such that $r(p) = \begin{cases} 1 & \text{if } p \notin S \\ \alpha & \text{if } p \in S \end{cases}$

- $G(P, r)$ is connected.

2 Overview of the Technical Results

Our algorithmic technical contribution is, mainly, twofold:

- The first contribution is a novel combination of linear programming, kernelization and branching/exhaustive search: we observe that linear programming can handle continuous domains, kernelization can reduce the search space within these domains, and, when combined with exhaustive search/branching, these can yield the desired solution. While several of our results are based on this approach, we would like to highlight, in particular, the FPT algorithm for k-SHRINKING TO ACYCLICITY (see Theorem 8), which exhibits a more intricate combination.

Table 1 A summary of the results in the paper. Additionally, we also show that all of the problems to be NP-hard, which is not mentioned in the table.

Objective	Type of Result	Cardinality version	Cardinality and cost-minimization version
k-SHRINKING TO INDEPENDENCE	Kernelization	$\mathcal{O}(k)$ partial kernel (Thm. 4), true polynomial kernel (Thm. 5)	$\mathcal{O}(k)$ partial kernel (Thm. 4)
	Parameterized complexity	Subexponential FPT (Thm. 1)	FPT (Thm. 6)
	Approximation	EPTAS (Thm. 10)	Bicriteria EPTAS (Thm. 10)
k-SHRINKING TO ACYCLICITY	Kernelization & Compression	True polynomial kernel (Thm. 9)	Partial polynomial compression (Thm. 7)
	Parameterized complexity	Subexponential FPT (Thm. 2)	FPT (Thm. 8)
k-SHRINKING TO CONNECTIVITY	Parameterized complexity	Subexponential FPT (Thm. 3)	–
A generalization of k-EXPANDING TO CONNECTIVITY	Parameterized complexity	W[1]-hard (Full version)	–

- The second contribution is, in a sense, the first "geometric" use of bidimensionality theory. Here, the application is tied up with the analysis of the different areas covered by disks. We employ this approach to design a subexponential FPT algorithm for k-SHRINKING TO CONNECTIVITY (see Theorem 3).

We believe that our two approaches are quite general in nature, and will be applicable to other parameterized problems that involve placement of geometric objects in the Euclidean space, particularly when dealing with continuous domains. We summarize our results in Table 1. In this extended abstract, we give an in-depth overview of our algorithmic results, and the technical details are deferred to the full version. One remark is that, all of these problems are shown to be NP-hard, and these NP-hardness results are shown via reduction from well-known planar problems, namely, INDEPENDENT SET, FEEDBACK VERTEX SET in planar graphs, and PLANAR 3-SAT, respectively. Now we turn to the technical overview of the rest of the results.

2.1 Subexponential FPT Algorithms

In this section, we discuss the first technical contribution of our paper, namely, to design subexponential FPT algorithms for k-SHRINKING TO INDEPENDENCE, k-SHRINKING TO ACYCLICITY and k-SHRINKING TO CONNECTIVITY (specifically the last problem).

Independence and Acyclicity. Recall that in these problems, the goal is to shrink at most k disks to radius α such that the resulting intersection graph is edgeless (resp. acyclic). For the sake of ease of exposition, let us suppose that α, the radius of the shrunken disks, is a constant. We will start by eliminating *large* cliques, i.e., we show using packing arguments that, if the largest clique in the original intersection graph $G = G(P, \mathbf{1})$ has size $\Omega_\alpha(1)$ [3], then no solution can shrink any subset of disks to radius α and remove all the edges. This bounds

---

[3] For the sake of simplicity, we use α in the subscript of $\mathcal{O}(\cdot)$ and $\Omega(\cdot)$ notation to suppress the dependence on α in the current overview.

the maximum clique-size. Now, we use *bidimensionality* arguments to show that, either the graph has $\Omega(\sqrt{k}) \times \Omega(\sqrt{k})$ grid as a minor, or the treewidth of the graph is bounded by $\mathcal{O}_\alpha(\sqrt{k})$. In the former case, we can conclude that it is a no-instance, whereas in the latter case we can perform dynamic programming on the tree decomposition of width $\mathcal{O}_\alpha(\sqrt{k})$ to solve the problem. This leads to the following results.

▶ **Theorem 1.** *There exists an algorithm that solves an instance* $\mathcal{I} = (P, k, \alpha)$ *of* k-SHRINKING TO INDEPENDENCE *in time* $2^{\mathcal{O}((\frac{1}{\alpha})^2 \sqrt{k})} \cdot n^{\mathcal{O}(1)}$.

▶ **Theorem 2.** *There exists an algorithm that solves an instance* $\mathcal{I} = (P, k, \alpha)$ *of* k-SHRINKING TO ACYCLICITY *in time* $(\frac{1}{\alpha}k)^{\mathcal{O}((\frac{1}{\alpha})^2 \sqrt{k})} \cdot n^{\mathcal{O}(1)}$.

Connectivity.   Now we turn to the subexponential FPT algorithm for k-SHRINKING TO CONNECTIVITY, which is one of the main results of the paper. This algorithm also begins with the elimination of large cliques. However, already for this step, our arguments are substantially more involved compared to all previously known subexponential-time algorithms for problems on geometric intersection graphs that also eliminate large cliques (see, e.g., [31, 29, 28, 30, 40]). Specifically, for all of these problems, the existence of a large clique trivially implies that we have a yes-instance (or a no-instance) at hand. However, for k-SHRINKING TO CONNECTIVITY, the proof is more complicated. Further, and perhaps more importantly, even once we have proved that every optimal solution contains "most" vertices of a clique, and we have "guessed" (by branching) which these vertices are, we can neither shrink their corresponding disks (since then, we might be left with a clique just as large as it was before) nor delete them. In particular, deletion is incorrect since it might turn a no-instance into a yes-instance, and vice versa (see Figure 4a). Fortunately, we are able to prove that it is possible to delete all of the guessed set except for a carefully chosen subset of a few representatives from it.
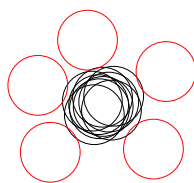
(a) There is a solution that shrinks some of the disks in the clique; however, shrinking *all* the disks in the clique results in the disconnected graph (due to the red disks).
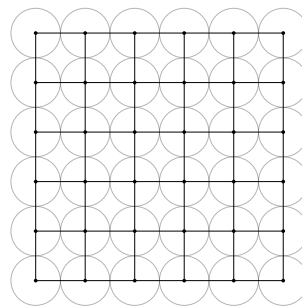
(b) Each disk corresponding to the vertices of the grid minor is "touching" its neighbors, and shrinking any of the disks results in a disconnected graph.

Figure 4 Illustrative figures for k-SHRINKING TO CONNECTIVITY.

The next step of our proof handles cycles (or, more precisely, closed walks) that we term *non-empty*. Specifically, we denote some disks as unshrinkable; at the beginning, these are simply the disks that cannot be shrunk without violating connectivity. Then, for a cycle, we consider the "area that it encloses", and identify two sets of disks of interest: the disks intersecting this area (the *relevant set*), and the disks strictly in the interior of this area (the *interior set*). If all the disks in the relevant set are unshrinkable, then the cycle is termed

unshrinkable, and if the interior set is non-empty, then the cycle is termed non-empty. Then, we prove that, given an unshrinkable cycle, it is safe to delete its entire interior set (which, in case of a non-empty cycle, yields progress). Additionally, if the interior set of a cycle is non-empty and contains a shrinkable disk, then the cycle is called *reducible*; we note that a shrinkable non-empty cycle might *not* be reducible. We prove that if we have a large enough collection of reducible cycles whose relevant sets are disjoint, then we have a yes-instance at hand.

Thirdly, we turn to deal with the case where we have a large grid as minor – here, it will also become clear why we had to analyze non-empty cycles. Again, let us remark that all previously known subexponential-time algorithms for problems on geometric intersection graphs, when encountering the case of a large grid as a minor, can simply say Yes or No. For us, having a large grid minor does not imply that we have a yes-instance (or a no-instance), although it may seem so at first glance – indeed, we might even be dealing with an "extreme" case where all of the disks that compose the grid minor are unshrinkable (see Figure 4b). To overcome this difficulty, first of all, we do not consider just any grid minor, but a grid minor such that the closed walks corresponding (according to the minor model) to its 4-cycles have some particular embedding on the plane. Then, we are able to show that within such a grid minor, one can identify either an unshrinkable non-empty closed walk (and, then, apply a reduction rule) or a large enough collection of reducible cycles whose relevant sets are disjoint (and, then, output Yes and terminate). Putting everything together, our algorithm works as follows. As long as we have a large clique, we branch to eliminate most of it. Then, based on a known result, we know that either we have a large grid minor of a particular form (and, then, proceed as described in the previous paragraph) or that the treewidth of the graph is small. In this latter case, we can simply use dynamic programming to directly solve the problem. Thus, we obtain the following result.

▶ **Theorem 3.** *There exists an algorithm that solves an instance* $\mathcal{I} = (P, k, \alpha)$ *of* k-SHRINKING TO CONNECTIVITY *in time* $(\frac{k}{\alpha})^{\mathcal{O}((\frac{1}{\alpha})^2 \cdot k^{3/4})} \cdot n^{\mathcal{O}(1)}$.

Expanding to achieve Connectivity. Surprisingly, it turns out that, unlike its shrinking counterpart, a generalization of k-EXPANDING TO CONNECTIVITY is W[1]-hard. In this generalization, we may only expand the disks centered at a specified subset of points; whereas the disks centered at the rest of the points are fixed to be unit disks. We show this using a reduction from COVERING POINTS BY UNIT DISKS, which is known to be W[1]-hard [43]. This result shows that not all natural problems of interest in this geometric graph modification model are fixed-parameter tractable. [4]

2.2 Polynomial Kernels and Compressions and their Consequences

Now we discuss our results regarding polynomial kernels and compressions for k-SHRINKING TO INDEPENDENCE/ACYCLICITY and their minimization variants. We also discuss how these results also enable us to design FPT algorithms for the minimization variants, using a combination of partial enumeration of the solution space of the kernels, and linear

---

[4] We remark that our algorithmic results (FPT algorithms, kernels, etc.) for the other problems – namely, k-SHRINKING TO INDEPENDENCE/ACYCLICITY/CONNECTIVITY and their variants – also hold for such a generalization, where a subset of disks cannot be shrunk. In fact, our algorithm for k-SHRINKING TO INDEPENDENCE supposes the presence of unshrinkable disks, as there, specifically, it simplifies the write-up. Thus, the preceding remark remains valid. However, for the sake of simplicity of presentation of the other results, we decided to stick with the special cases where any disks can be shrunk.

programming. Note that the subexponential FPT algorithms for k-Shrinking to Independence/Acyclicity discussed in the previous section work directly on the original instances and do not rely on the kernels.

2.2.1   (Min) k-Shrinking to Independence

We show the following result.

▶ **Theorem 4.** [Informal] *There exists a partial kernel for k-Shrinking to Independence (resp. for Min k-Shrinking to Independence), where the number of points in the resulting instance is $\mathcal{O}(k)$.*

Note that in this result, we are only able to bound the number of points in the resulting instance by $\mathcal{O}(k)$. However, the number of bits required to encode the coordinates of each point may still be unbounded. Therefore, we only obtain a *partial kernel* for these two problems.

Since k-Shrinking to Independence is a "discrete" problem, we can in fact extend this argument to obtain a (fully) polynomial kernel. Here, we need to additionally show that the geometric information that is relevant for solving k-Shrinking to Independence can be encoded using polynomially many bits in k. Thus, we obtain the following result.

▶ **Theorem 5.** *There exists a (true) polynomial kernel for k-Shrinking to Independence, paramterized by k.*

FPT Algorithm for Min k-Shrinking to Independence.   Note that usually the existence of (partial) kernels is sufficient to immediately conclude that the problem is FPT by enumerating all possible solutions. However, for the case of Min k-Shrinking to Independence, even if one fixes a subset of disks of size k to be shrunk (i.e., a *partial solution*), the new radius of each disk is a real number in the range $[\alpha, 1]$. Thus, the number of possible solutions is infinite. Here, we enumerate all partial solutions, and use *linear programming* [5] to check whether there exists a radius assignment of cost at most μ, resulting in an edgeless graph. Thus, we obtain the following result.

▶ **Theorem 6.** *There exists an algorithm that decides an instance $\mathcal{I} = (P, k, \alpha, \mu)$ of Min k-Shrinking to Independence in time $k^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. Additionally, if $\mathcal{I}$ is a yes-instance, the algorithm can return an* optimal solution *for $\mathcal{I}$.*

2.2.2   (Min) k-Shrinking to Acyclicity

We first give an overview of our result regarding the (partial) polynomial compressibility of Min k-Shrinking to Acyclicity, which, along with the FPT algorithm for the problem (as stated subsequently) is the second main result of the paper.

▶ **Theorem 7** (Informal). *There exists a partial polynomial compression for Min k-Shrinking to Acyclicity into an annotated graph problem, where the size of the resulting instance is bounded by a polynomial in k, assuming each real number in the instance can be stored using $\mathcal{O}(1)$ bits.*

---

[5] Since our LP involves real numbers, we cannot use standard LP solvers that run in (weakly) polynomial time. However, since we work in Real RAM model, we can use an algebraic LP solver such as [18, 44] to find an optimum solution, resulting in the algorithm with the claimed running time bound. We thank a reviewer for pointing this out. More details on this can be found in the full version.
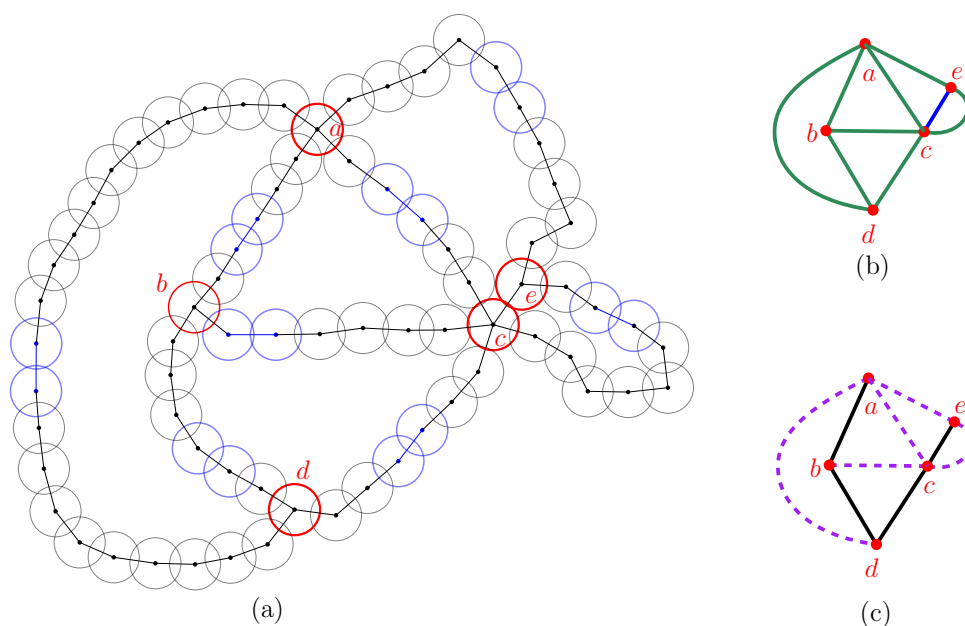
Figure 5 (a) Given intersection graph after removing the vertices of degree at most 1. Vertices of degree at least 3 are shown in red. (b) Resulting multigraph H on vertices of degree at least 3, obtained after replacing degree-2 paths with reducible edges (shown in green) and one of the two edges ce is an original edge (blue). Annotations of green edges include, for example, distance between farthest pairs of disks in the corresponding path (shown in blue in figure (a)). (c) A particular guess for the edges of R (dashed purple), such that $H - R$ (graph induced on black edges) is acyclic. For each edge of R, we guess the number of disks that will be shrunk to remove this edge. Then, an optimal values of the shrunk radii that is compatible with this guess is found using linear programming. For example, such a solution may shrink the disk centered at c (removing bc, ac), e (removing ae and together with c, the edge ce), and the two blue disks in the path corresponding to ad.

We first state relatively straightforward reduction rules. First, we argue that, if any vertex in the initial intersection graph has a "large" degree (i.e., larger than some ck), then it implies the existence of a clique of size at least $k + 3$ – note that no solution that is allowed to shrink at most k disks can remove all the edges of such a clique, which implies that we have a no-instance. Thus, either the maximum degree in the given intersection graph is bounded by $\mathcal{O}(k)$, or we can say No. Further, we can safely delete all vertices of degree at most one as well as acyclic components from the graph. So far, these arguments bear resemblance to those used to obtain a polynomial kernel for FEEDBACK VERTEX SET. From this point onward, however, we deviate significantly from the standard arguments, and need to rely on the inherent geometric nature of the problem.

We now handle each connected component of the graph separately. If a connected component is a simple cycle, then observe that an optimal solution shrinks either one or two disks, depending on the distances between the consecutive vertices in the cycle. We can compute the optimal choice and get rid of all such isolated cycles. Next, we aim to handle "long" degree-2 paths in a component. Note that an optimal solution shrinks $\ell \leq 2$ disks from such a path. We replace such a path in the intersection graph with a single edge, called a *reducible edge*, that is annotated with an optimal choice for shrinking $0 \leq \ell \leq 2$ disks from the original path. The remaining edges in the graph are *original* edges, and we annotate each edge with the Euclidean distance between its two endpoints.

Thus, we obtain a graph H (in fact, in the actual construction it will be a multigraph). Further, the minimum degree of H is 3, and the maximum degree of H is $\mathcal{O}(k)$. At this point, we can argue that, if the given instance is a yes-instance, then $|V(H)| = \mathcal{O}(k^2)$ and $|E(H)| = \mathcal{O}(k^3)$. The task of determining whether the original instance is a yes-instance of MIN k-SHRINKING TO ACYCLICITY is equivalent to determining whether H is a yes-instance of an appropriately defined graph problem, say Π. Note that although the size of H is bounded by $\mathcal{O}(k^3)$, the edges of H are annotated with relevant distances denoting optimal choices, and thus can be real numbers in general. Thus, we can only infer a (partial) polynomial compression from MIN k-SHRINKING TO ACYCLICITY into Π.

We can use the compressed instance of Π to design an FPT algorithm for MIN k-SHRINKING TO ACYCLICITY, as follows. Recall that $|V(H)| = \mathcal{O}(k^2)$ and $|E(H)| = \mathcal{O}(k^3)$. Thus, any acyclic subgraph of H with vertex set $V(H)$ must contain at most $|V(H)| - 1 = \mathcal{O}(k^2)$ edges. We iterate over all such edge sets $F \subseteq E(H)$ inducing an acyclic subgraph – note that there are at most $k^{\mathcal{O}(k^2)}$ guesses. This implies that each edge in $R = E(H) \setminus F$ must be removed by the solution; note that $|R| = \mathcal{O}(k^3)$. Further, an original edge of R can be removed by shrinking either one or both of its endpoints (3 choices), whereas a reducible edge can be removed by shrinking 1 or 2 disks in the corresponding path, and these choices are encoded in the annotation. Thus, there are a constant number of choices for removal of each edge of R, which results in $2^{\mathcal{O}(k^3)}$ further guesses. For each set of guesses, we can determine an optimal choice for each such guess using the annotated information, and formulate a linear program (LP) that captures the best choices. By solving the LP, in time $2^{\mathcal{O}(k^3 \log k)}$, we can find a minimum-cost solution that shrinks at most k disks that is compatible with all the shrinking choices, if one exists. Finally, we return the best solution found over all $2^{\mathcal{O}(k^3)}$ guesses, which leads to the following result.

▶ **Theorem 8.** *There exists an algorithm that takes as an input of* MIN k-SHRINKING TO ACYCLICITY*, and in time* $2^{\mathcal{O}(k^3 \log k)} \cdot n^{\mathcal{O}(1)}$ *time, either correctly concludes that it is a no-instance; or finds a solution of the smallest cost that shrinks at most k disks.*

The aforementioned approach also yields a polynomial compression for k-SHRINKING TO ACYCLICITY into an intermediate multigraph problem. However, since k-SHRINKING TO ACYCLICITY is a "discrete" problem, one can, in fact, encode all the relevant geometric information using polynomially bits in k, resulting in a polynomial kernel for the problem.

▶ **Theorem 9.** *There exists a polynomial kernel for k-SHRINKING TO ACYCLICITY parameterized by the number of disks k that one is allowed to shrink.*

2.3   Approximation Schemes for (MIN) k-SHRINKING TO INDEPENDENCE

Finally, we design EPTASes, i.e., efficient polynomial-time approximation schemes for (MIN) k-SHRINKING TO INDEPENDENCE. For k-SHRINKING TO INDEPENDENCE, for a fixed $\varepsilon > 0$, our algorithm either correctly concludes that the given instance is a no-instance, or returns a solution that shrinks at most $(1 + \varepsilon)k$ disks to radius α. For MIN k-SHRINKING TO INDEPENDENCE, our algorithm returns a *bicriteria* EPTAS. That is, it either correctly concludes that the given instance is a no-instance, or returns a solution of cost at most $(1 + \varepsilon)\mu$ that shrinks at most $(1 + \varepsilon)k$ disks to a radius in $[\alpha, 1]$. Thus, we obtain the following theorem.

▶ **Theorem 10.** *There exists a bicriteria EPTAS (resp. EPTAS) for* MIN k-SHRINKING TO INDEPENDENCE *(resp. k-SHRINKING TO INDEPENDENCE) running in time* $2^{\mathcal{O}(\frac{1}{\varepsilon} \log(\frac{1}{\varepsilon}))} \cdot n^{\mathcal{O}(1)}$.

These results are obtained using the well-known *shifting technique* [6, 34] to reduce the problem to bounded-size instances, at the expense of a small loss in the approximation factor. Each bounded-size instance is then solved optimally. We give an overview for our bicriteria EPTAS for MIN k-SHRINKING TO INDEPENDENCE where the arguments are more involved; those for k-SHRINKING TO INDEPENDENCE are relatively simpler. First, we subdivide the instance into sub-instances induced by $\mathcal{O}(1/\varepsilon) \times \mathcal{O}(1/\varepsilon)$ squares, such that the number of optimal disks intersecting the boundaries of the squares is at most εk, and the total sum of changes in the radii of such disks is at most $\varepsilon \mu$. In order to solve each subproblem, we further partition the square into smaller grid cells, such that each cell induces a clique in the given UDG. If such a clique contains at most c/ε disks, then we guess the exact subset of disks that are shrunk by an optimal solution. Otherwise, if the clique contains more than c/ε disks, then we mark all disks as *potentially shrinkable*. Note that all except at most one disk from each clique must be shrunk by any solution. Thus, we can argue that the total number of disks that are shrunk in the latter case is at most εk. Now, for each guess of the *potentially shrinkable disks*, we use linear programming to find an optimal solution that is only allowed to shrink the potentially shrinkable disks, in order to result in an edgeless graph. Finally, we need to combine the solutions of subproblems in a careful manner – since the disks near the boundaries of the cell appear in multiple subproblems. We achieve this using dynamic programming over the optimal solutions computed for the bounded-size instances.

As mentioned earlier, we defer the formal descriptions of algorithmic and hardness results to the full version. We proceed directly to the conclusion and directions for future research.

3   Conclusion and Future Research

In this paper, we initiated the study of graph modification by *scaling of disks* from the perspective of Parameterized Complexity. Specifically, we focused on modifying the connectivity properties of the graph while shrinking or expanding a subset of disks. The (MIN) k-SHRINKING TO INDEPENDENCE/ACYCLICITY problems correspond to shrinking at most k disks to achieve a completely disconnected (i.e., edgeless), and minimally connected (i.e., acyclic) intersection graph, respectively. Whereas k-SHRINKING TO CONNECTIVITY (resp. k-EXPANDING TO CONNECTIVITY) corresponds to shrinking at least (resp. at most) k disks while retaining (resp. achieving) connectivity in the intersection graph.

Besides our NP/W[1]-hardness results for the above problems, we also gave several algorithmic results. For (MIN) k-SHRINKING TO INDEPENDENCE, we designed (partial) polynomial kernels, FPT algorithms, and (bicriteria) approximation schemes. For MIN k-SHRINKING TO INDEPENDENCE, we use a novel combination of linear programming with ideas from parameterized complexity and geometric approximation respectively. The FPT algorithm is obtained by first using kernelization to derive a bounded-size instance, and then using LP to solve this instance; whereas the approximation algorithm uses a combination of the classical shifting technique along with LP. We also obtained similar results for (MIN)k-SHRINKING TO ACYCLICITY; however our arguments and techniques are more sophisticated due to "non-local" nature of the problem. Finally, for k-SHRINKING TO CONNECTIVITY, we introduced a novel decomposition theorem that returns a grid minor of the disk graph along with a particular kind of embedding on the plane, and used this theorem to design a subexponential FPT algorithm for the problem. Similar ideas also gave subexponential FPT algorithms for k-SHRINKING TO INDEPENDENCE/ACYCLICITY.

Analogous to the results obtained for k-SHRINKING TO INDEPENDENCE, it is natural to ask whether k-SHRINKING TO CONNECTIVITY admits (partial) polynomial kernel, and approximation schemes. Furthermore, it is also possible to define cost-optimization version of k-SHRINKING TO CONNECTIVITY and study it from the perspective of fixed-parameter and approximation algorithms.

At a broader level, we believe that a significant portion of our contribution lies in the conceptual realm, with the goal of pioneering a new domain within the field of geometric graph modification problems. Furthermore, it opens up a multitude of potential directions for future research. In this context, we propose a few specific avenues to explore.

- **Modification to other graph Classes:** In addition to graph modification to achieve edgelessness or connectivity, it is intriguing to consider modifying the given geometric graph to belong to other families of graphs, such as acyclic graphs (for minimal connectivity assurance), cluster graphs (for clustering purposes), cliques (for direct communication facilitation), and bounded-degree graphs (which generalize edgeless graphs).

- **Various geometric intersection graphs as input:** Our study primarily focused on the simplest geometric intersection graphs, specifically disk graphs, where graph modification problems often become tractable within the realm of Parameterized Complexity despite being NP-hard. Beyond disk graphs, various other geometric intersection graphs, such as string graphs and map graphs, present substantial differences. Additionally, we can extend our investigations to three-dimensional spaces (e.g., unit ball graphs and more general ball graphs). Therefore, a natural research direction is to explore geometric modifications for graph classes within these broader and distinct geometric intersection graph categories.

- **New geometric modification operations:** While we utilized scaling as a quintessential example of a geometric modification operation, other geometric operations, like movement (recently studied in [25, 26]) or rotation, may be more suitable depending on the context and the nature of the geometric objects under consideration. Furthermore, one might contemplate simultaneously shifting, scaling, and rotating objects (or applying any combination of two out of these three operations). This parallels research on graph modification problems where significant efforts have been made to concurrently address edge deletions and insertions. Beyond these combinations, defining and researching other geometric operations can be highly relevant based on the objectives and the class of geometric intersection graphs at hand. This includes operations like smoothing and dimension reduction.

- **Meta-theorems.** While the study of research questions on a problem-to-problem basis can be sometimes challenging, yet interesting in its own right, the ultimate goal is often to prove meta-theorems that simultaneously assert or refute, say, membership in FPT or the existence of a polynomial kernel, for a host of problems. For example, we may claim, for a whole class of problems, that if a specific constraint X in the definition of any problem in the class can be expressed in some logic (e.g., Monadic Second Order Logic), then the problem is FPT, or that if and only if the input graph (or one of the input graphs) has bounded treewidth, then the problem is FPT, or that if the problem admits an OR-cross-composition from an NP-hard problem, then it has no polynomial kernel unless the polynomial hierarchy collapses. We refer to the books [23, 20, 32] for examples of such results. Clearly, after gaining a better understanding of the problem class introduced in this paper, the pursuit of such meta-theorems represents an intriguing avenue for further research.

References

1 Ankush Acharyya, Minati De, Subhas C Nandy, and Bodhayan Roy. Range assignment of base-stations maximizing coverage area without interference. *Theoretical Computer Science*, 804:81–97, 2020. `doi:10.1016/J.TCS.2019.10.044`.

2 Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. *ACM Trans. Algorithms*, 15(1):11:1–11:28, 2019. `doi:10.1145/3284356`.

3 Akanksha Agrawal, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Interval vertex deletion admits a polynomial kernel. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1711–1730, 2019. `doi:10.1137/1.9781611975482.103`.

4 Jochen Alber and Jiří Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *Journal of Algorithms*, 52(2):134–151, 2004. `doi:10.1016/J.JALGOR.2003.10.001`.

5 Greg Aloupis, Mirela Damian, Robin Y. Flatland, Matias Korman, Özgür Özkan, David Rappaport, and Stefanie Wuhrer. Establishing strong connectivity using optimal radius half-disk antennas. *Comput. Geom.*, 46(3):328–339, 2013. `doi:10.1016/J.COMGEO.2012.09.008`.

6 Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. `doi:10.1145/174644.174650`.

7 Ahmad Biniaz, Prosenjit Bose, Paz Carmi, Anil Maheshwari, J Ian Munro, and Michiel Smid. Faster algorithms for some optimization problems on collinear points. *arXiv preprint*, 2018. `arXiv:1802.09505`.

8 Ivan Bliznets, Marek Cygan, Pawel Komosa, Lukáš Mach, and Michał Pilipczuk. Lower bounds for the parameterized complexity of minimum fill-in and other completion problems. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1132–1151. SIAM, 2016.

9 Ivan Bliznets, Fedor V Fomin, Marcin Pilipczuk, and Michał Pilipczuk. Subexponential parameterized algorithm for interval completion. *ACM Transactions on Algorithms (TALG)*, 14(3):35, 2018.

10 Édouard Bonnet and Pawel Rzazewski. Optimality program in segment and string graphs. *Algorithmica*, 81(7):3047–3073, 2019. `doi:10.1007/s00453-019-00568-7`.

11 Marin Bougeret, Bart M. P. Jansen, and Ignasi Sau. Bridge-depth characterizes which structural parameterizations of vertex cover admit a polynomial kernel. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, pages 16:1–16:19, 2020. `doi:10.4230/LIPICS.ICALP.2020.16`.

12 Gruia Calinescu and Peng-Jun Wan. Range assignment for high connectivity in wireless ad hoc networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 235–246. Springer, 2003. `doi:10.1007/978-3-540-39611-6_21`.

13 Yixin Cao. Linear recognition of almost interval graphs. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1096–1115. SIAM, 2016. `doi:10.1137/1.9781611974331.CH77`.

14 Yixin Cao and RB Sandeep. Minimum fill-in: Inapproximability and almost tight lower bounds. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 875–880. SIAM, 2017.

15 Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Energy-efficient wireless network design. *Theory of Computing Systems*, 39(5):593–617, 2006. `doi:10.1007/S00224-005-1204-8`.

16 Paz Carmi and Matthew J. Katz. Power assignment in radio networks with two power levels. *Algorithmica*, 47(2):183–201, 2007. `doi:10.1007/S00453-006-1230-1`.

17 Erin W. Chambers, Sándor P. Fekete, Hella-Franziska Hoffmann, Dimitri Marinakis, Joseph S. B. Mitchell, Srinivasan Venkatesh, Ulrike Stege, and Sue Whitesides. Connecting a set of circles with minimum sum of radii. *Comput. Geom.*, 68:62–76, 2018. `doi:10.1016/j.comgeo.2017.06.002`.

18 Timothy M. Chan. Improved deterministic algorithms for linear programming in low dimensions. *ACM Trans. Algorithms*, 14(3):30:1–30:10, 2018. `doi:10.1145/3155312`.

19 Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Comput. Sci. Rev.*, 48:100556, 2023. `doi:10.1016/j.cosrev.2023.100556`.

20 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`.

21 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for eth-tight algorithms and lower bounds in geometric intersection graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 574–586, 2018. `doi:10.1145/3188745.3188854`.

22 Mark de Berg, Sándor Kisfaludi-Bak, and Gerhard J. Woeginger. The complexity of dominating set in geometric intersection graphs. *Theor. Comput. Sci.*, 769:18–31, 2019. `doi:10.1016/J.TCS.2018.10.007`.

23 Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013. `doi:10.1007/978-1-4471-5559-1`.

24 David Eppstein. Maximizing the sum of radii of disjoint balls or disks. *arXiv preprint*, 2016. `arXiv:1607.02184`.

25 Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, Saket Saurabh, and Meirav Zehavi. Kernelization for spreading points. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 48:1–48:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPICS.ESA.2023.48`.

26 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`.

27 Fedor V. Fomin, Tien-Nam Le, Daniel Lokshtanov, Saket Saurabh, Stéphan Thomassé, and Meirav Zehavi. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems. *ACM Trans. Algorithms*, 15(1):13:1–13:44, 2019. `doi:10.1145/3293466`.

28 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Decomposition of map graphs with applications. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 60:1–60:15, 2019. `doi:10.4230/LIPICS.ICALP.2019.60`.

29 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Finding, hitting and packing cycles in subexponential time on unit disk graphs. *Discret. Comput. Geom.*, 62(4):879–911, 2019. `doi:10.1007/S00454-018-00054-X`.

30 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Eth-tight algorithms for long path and cycle on unit disk graphs. In *36th International Symposium on Computational Geometry, SoCG 2020, June 23-26, 2020, Zürich, Switzerland*, pages 44:1–44:18, 2020. `doi:10.4230/LIPICS.SOCG.2020.44`.

31 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1563–1575, 2012. `doi:10.1137/1.9781611973099.124`.

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

33 Panos Giannopoulos, Christian Knauer, and Sue Whitesides. Parameterized complexity of geometric problems. *The Computer Journal*, 51(3):372–384, 2008. `doi:10.1093/COMJNL/BXM053`.

34 Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985. `doi:10.1145/2455.214106`.

35 Bart Jansen. Polynomial kernels for hard problems on disk graphs. In *Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 310–321. Springer, 2010. `doi:10.1007/978-3-642-13731-0_30`.

36 Bart MP Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. *SIAM Journal on Discrete Mathematics*, 32(3):2258–2301, 2018. `doi:10.1137/17M112035X`.

37 Ariel Kulik and Hadas Shachnai. Analysis of two-variable recurrence relations with application to parameterized approximations. *CoRR*, abs/1911.02653, Accepted to Symposium on Foundations of Computer Sciences (FOCS 2020), 2019. `arXiv:1911.02653,`.

38 Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 971–989, 2020. `doi:10.1137/1.9781611975994.58`.

39 Errol L Lloyd, Rui Liu, Madhav V Marathe, Ram Ramanathan, and Sekharipuram S Ravi. Algorithmic aspects of topology control problems for ad hoc networks. *Mobile Networks and applications*, 10(1):19–34, 2005. `doi:10.1023/B:MONE.0000048543.95178.F5`.

40 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. Subexponential parameterized algorithms on disk graphs (extended abstract). In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2005–2031, 2022. `doi:10.1137/1.9781611977073.80`.

41 Dániel Marx. Efficient approximation schemes for geometric problems? In *Algorithms - ESA 2005, 13th Annual European Symposium, Palma de Mallorca, Spain, October 3-6, 2005, Proceedings*, pages 448–459, 2005. `doi:10.1007/11561071_41`.

42 Dániel Marx. Parameterized complexity of independence and domination on geometric graphs. In *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, pages 154–165, 2006. `doi:10.1007/11847250_14`.

43 Dániel Marx. On the optimality of planar and geometric approximation schemes. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 338–348. IEEE Computer Society, 2007. `doi:10.1109/FOCS.2007.50`.

44 Jirí Matousek, Micha Sharir, and Emo Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4/5):498–516, 1996. `doi:10.1007/BF01940877`.

45 Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1035–1054, 2019. `doi:10.1137/1.9781611975482.64`.

46 Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM computing surveys (CSUR)*, 37(2):164–194, 2005. `doi:10.1145/1089733.1089736`.

47 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010. `doi:10.1145/1721837.1721848`.

48 P-J Wan, Gruia Călinescu, X-Y Li, and Ophir Frieder. Minimum-energy broadcasting in static ad hoc wireless networks. *Wireless Networks*, 8(6):607–617, 2002. `doi:10.1023/A:1020381720601`.