Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams*

Dániel Marx[1] and Michał Pilipczuk[2]

[1] Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI) dmarx@cs.bme.hu
[2] Institute of Informatics, University of Warsaw, Poland, michal.pilipczuk@mimuw.edu.pl

Abstract. We study a general family of facility location problems defined on planar graphs and on the 2-dimensional plane. In these problems, a subset of k objects has to be selected, satisfying certain packing (disjointness) and covering constraints. Our main result is showing that, for each of these problems, the $n^{\mathcal{O}(k)}$ time brute force algorithm of selecting k objects can be improved to $n^{\mathcal{O}(\sqrt{k})}$ time. The algorithm is based on focusing on the Voronoi diagram of a hypothetical solution of k objects; this idea was introduced recently in the design of geometric QPTASs, but was not yet used for exact algorithms and for planar graphs. As concrete consequences of our main result, we obtain $n^{\mathcal{O}(\sqrt{k})}$ time algorithms for the following problems: d-Scattered Set in planar graphs (find k vertices at pairwise distance d); d-Dominating Set/(k,d)-Center in planar graphs (find k vertices such that every vertex is at distance at most d from these vertices); select k pairwise disjoint connected vertex sets from a given collection; select k pairwise disjoint disks in the plane (of possibly different radii) from a given collection; cover a set of points in the plane by selecting k disks/axis-parallel squares from a given collection. We complement these positive results with lower bounds suggesting that some similar, but slightly more general problems (such as covering points with axis-parallel rectangles) do not admit $n^{\mathcal{O}(\sqrt{k})}$ time algorithms.

1 Introduction

Parameterized problems often become easier when restricted to planar graphs: usually significantly better running times can be achieved and sometimes problems that are W[1]-hard on general graphs become fixed-parameter tractable on planar graphs. In most cases, the improved running time involves a square root

* M. Pilipczuk was partially supported by ERC Grant no. 267959, by the Polish National Science Centre grant DEC-2013/11/D/ST6/03073 and by the Foundation for Polish Science via the START stipend programme. M. Pilipczuk also holds a postdoc position at Warsaw Center of Mathematics and Computer Science. D. Marx was supported by ERC Grant PARAMTIGHT (No. 280152) and OTKA grant NK105645.

dependence on the parameter: it is often of the form $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ or $n^{\mathcal{O}(\sqrt{k})}$. The appearance of the square root can be usually traced back to the fact that a planar graph with n vertices has treewidth $\mathcal{O}(\sqrt{n})$. Indeed, the theory of bidimensionality gives a quick explanation why problems such as Independent Set, Longest Path, Feedback Vertex Set, Dominating Set, or even distance-r versions of Independent Set and Dominating Set (for fixed r) have algorithms with running time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ (cf. survey [6]). In all these problems, there is a relation between the size of the largest grid minor and the size of the optimum solution, which allows us to bound the treewidth of the graph in terms of the parameter of the problem. More recently, subexponential parameterized algorithms have been explored also for problems where there is no such straightforward parameter-treewidth bound: for examples, see [5,9,10,14].

A similar "square root phenomenon" has been observed in the case of geometric problems: it is usual to see a square root in the exponent of the running time of algorithms for NP-hard problems defined in the 2-dimensional Euclidean plane. Most relevant to our paper is the fact that Independent Set for unit disks (given a set of n unit disks, select k of them that are pairwise disjoint) and the discrete k-center problem (given a set of n points and a set of n unit disks, select k disks whose union covers every point) can be solved in time $n^{\mathcal{O}(\sqrt{k})}$ by geometric separation theorems and shifting arguments [3,4,8,12], improving on the trivial $n^{\mathcal{O}(k)}$ time brute force algorithm. However, all of these algorithms are crucially based on a notion of area and rely on the property that all the disks have the same size (at least approximately). Therefore, it seems unlikely that these techniques can be generalized to the case when the disks can have very different radii or to planar-graph versions of the problem, where the notion of area is meaningless. Using similar techniques, one can obtain approximation schemes for these and related geometric problems, again with the limitation that the objects need to have (roughly) the same area. Very recently, a new and powerful technique emerged from a line of quasi-polynomial time approximation schemes (QPTAS) for geometric problems [1,2,7,13]. As described explicitly by Har-Peled [7], the main idea is to reason about the Voronoi diagram of the k objects in the solution. In particular, we are trying to guess a separator consisting of $\mathcal{O}(\sqrt{k})$ segments that corresponds to a balanced separator of the Voronoi diagram. In this paper, we show how this basic idea and its extensions can be implemented to obtain $n^{\mathcal{O}(\sqrt{k})}$ time exact algorithms for a wide family of geometric packing and covering problems in a uniform way. In fact, we show that the algorithms can be made to work in the much more general context of planar graph problems.

Algorithmic results. We study a general family of facility location problems for planar graphs, where a set of k objects has to be selected, subject to certain independence and covering constraints. Two archetypal problems from this family are (1) selecting k vertices of an edge-weighted planar graph that are at distance at least d from each other (d-Scattered Set) and (2) selecting k vertices of an edge-weighted planar graph such that every vertex of the graph is at distance at most d from a selected vertex (d-Dominating Set); for both problems, d is a real value being part of the input. We show that, under very general conditions, the trivial

$n^{\mathcal{O}(k)}$ time brute force algorithm can be improved to $n^{\mathcal{O}(\sqrt{k})}$ time for problems in this family. Our result is not just a simple consequence of bidimensionality and bounding the treewidth of the input graph. Instead, we focus on the Voronoi diagram of a hypothetical solution, which can be considered as a planar graph with $\mathcal{O}(k)$ vertices. It is known that such a planar graph has a balanced separator cycle of length $\mathcal{O}(\sqrt{k})$, which can be translated into a separator that breaks the instance in way suitable for using recursion on the resulting subproblems. Of course, we do not know the Voronoi diagram of the solution and its balanced separator cycle, but we argue that only $n^{\mathcal{O}(\sqrt{k})}$ separator cycles can be potential candidates. Hence, by guessing one of these cycles, we define and solve $n^{\mathcal{O}(\sqrt{k})}$ subproblems. The running time of the algorithm is thus governed by a recurrence relation of the form $f(k) = n^{\mathcal{O}(\sqrt{k})} f(k/2)$, which resolves to $f(k) = n^{\mathcal{O}(\sqrt{k})}$.

In Section 3, we define a general facility location problem Disjoint Network Coverage that contains numerous concrete problems of interest as special cases. Now, we discuss specific algorithmic results following from the general result.

Informally, the input of Disjoint Network Coverage consists of an edge-weighted planar graph G, a set $\mathcal{D}$ of objects (which are connected sets of vertices in G) and a set $\mathcal{C}$ of clients (which are vertices of G). The task is to select a set of exactly k pairwise-disjoint[3] objects that maximizes the total number of the covered clients. We define covering as follows: the input contains a radius for each object in $\mathcal{D}$ and a sensitivity for each client in $\mathcal{C}$, and a client is considered covered by an object if the sum of the radius and the sensitivity is at least the distance between the object and the client. When both the radius and the sensitivity are 0, then this means that the client is inside the object; when the radius is r and the sensitivity is 0, then this means that the client is at distance at most r from the object. The objects and the clients may be equipped with costs and prizes, and we may want to maximize/minimize the total revenue of the solution.

The first special case of the problem is when there are no clients at all: then the task is to select k objects that are pairwise disjoint. Our algorithm solves this problem in complete generality: the only condition is that each object is a connected vertex set (i.e. it induces a connected subgraph of G).

Theorem 1.1 (packing connected sets). *Let G be a planar graph, $\mathcal{D}$ be a family of connected vertex sets of G, and k be an integer. In time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$, we can find a set of k pairwise disjoint objects in $\mathcal{D}$, if such a set exists.*

We can also solve the weighted version, where we want to select k members of $\mathcal{D}$ maximizing the total weight. As a special case, Theorem 1.1 gives us an $n^{\mathcal{O}(\sqrt{k})}$ time algorithm for d-Scattered Set, which asks for k vertices that are at distance at least d from each other (with d being part of the input).

If each object in $\mathcal{D}$ is a single vertex and $r(\cdot)$ assigns a radius to each object (potentially different radii for different objects), then we get a natural covering problem. Thus, the following theorem is also a corollary of our general result.

[3] More precisely, if objects have different radii, then instead of requiring disjointness, we set up a technical condition called "normality," which we define in Section 3.

Theorem 1.2 (covering vertices with centers of different radii). *Let G be a planar graph, let $D, C \subseteq V(G)$ be two subsets of vertices, let $r\colon D \to \mathbb{Z}^+$ be a function, and k be an integer. In time $|D|^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$, we can find a set $S \subseteq D$ of k vertices that maximizes the number of vertices covered in C, where a vertex $u \in C$ is covered by $v \in S$ if the distance between u and v is at most $r(v)$.*

If $D = C = V(G)$, $r(v) = d$ for every $v \in V(G)$, and we are looking for a solution fully covering C, then we obtain as a special case d-Dominating Set (also called (k, d)-Center). Theorem 1.2 gives an $n^{\mathcal{O}(\sqrt{k})}$ time algorithm for this problem (with d being part of the input). Theorem 1.2 can be also interpreted as covering the vertices in C by very specific objects: balls of radius $r(v)$ around a center v. If we require that the selected objects of the solution are pairwise disjoint, then we can generalize this problem to arbitrary objects.

Theorem 1.3 (covering vertices with independent objects). *Let G be a planar graph, let $\mathcal{D}$ be a set of connected vertex sets in G, let $C \subseteq V(G)$ be a set of vertices, and let k be an integer. In time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$, we can find a set S of at most k pairwise disjoint objects in $\mathcal{D}$ that maximizes the number of vertices of C in the union of the vertex sets in S.*

By simple reductions, geometric packing/covering problems can be reduced to problems on planar graphs. In particular, given a set of disks (of possibly different radii), the problem of selecting k disjoint disks can be reduced to selecting disjoint connected vertex sets in a planar graph, and Theorem 1.1 can be applied.

Theorem 1.4 (packing disks). *Given a set $\mathcal{D}$ of disks (of possibly different radii) in the plane, in time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})}$ we can find a set of k pairwise disjoint disks, if such a set exists.*

This is a strong generalization of the results of Alber and Fiala [4], which gives an $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})}$ time algorithm only if the ratio of the radii of the smallest and largest disks can bounded by a constant (in particular, if all the disks are unit disks). As Theorem 1.1 works for arbitrary connected sets of vertices, we can prove the analog of Theorem 1.4 for most reasonable sets of connected geometric objects.

Theorem 1.5 (packing simple polygons). *Given a set $\mathcal{D}$ of simple polygons in the plane, in time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ we can find a set of k polygons in $\mathcal{D}$ with pairwise disjoint closed interiors, if such a set exists. Here n is the total number of vertices of the polygons in $\mathcal{D}$.*

Similarly, the problem of covering the maximum number of points by selecting k disks from a given set $\mathcal{D}$ of disks can be reduced to a problem on planar graphs and then Theorem 1.2 can be invoked.

Theorem 1.6 (covering with disks). *Given a set $\mathcal{C}$ of points and a set $\mathcal{D}$ of disks (of possibly different radii) in the plane, in time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot |\mathcal{C}|^{\mathcal{O}(1)}$ we can find a set of k disks in $\mathcal{D}$ maximizing the total number of points they cover in $\mathcal{C}$.*

Covering points with axis-parallel squares (of different sizes) can be handled similarly, by treating axis-parallel squares as balls in the *in the ℓ_∞ metric.*

Theorem 1.7 (covering with squares). *Given a set $\mathcal{C}$ of points and a set $\mathcal{D}$ of axis-parallel squares (of possibly different size) in the plane, in time $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot |\mathcal{C}|^{\mathcal{O}(1)}$ we can find a set of k squares in $\mathcal{D}$ maximizing the total number of points they cover in $\mathcal{C}$.*

Hardness results. Comparing packing results Theorems 1.1 and 1.5 with covering results Theorems 1.2, 1.6, and 1.7, one can observe that our algorithm solves packing problems in much wider generality than covering problems. It seems that we can handle arbitrary objects in packing problems, while it is essential for covering problems that each object is a ball in some metric. We present a set of hardness results suggesting that this apparent difference is not a shortcoming of our algorithm, but it is inherent to the problem: there are natural geometric covering problems where the square root phenomenon does not occur.

Using a result of Pătraşcu and Williams [15] and a simple reduction from Dominating Set, we show that if the task is to cover points with convex polygons, then improving upon a brute-force algorithm is unlikely.

Theorem 1.8 (covering with convex polygons, lower bound). *Let $\mathcal{D}$ be a set of convex polygons and let $\mathcal{P}$ be a set of points in the plane. Assuming SETH, there is no $f(k) \cdot (|\mathcal{D}| + |\mathcal{P}|)^{k-\epsilon}$ time algorithm for any computable function f and $\epsilon > 0$ that decides if there are k polygons in $\mathcal{D}$ that together cover $\mathcal{P}$.*

Theorem 1.8 gives a lower bound only if the covering problem allows arbitrary convex polygons. We present also two lower bounds in the much more restricted setting of covering with axis-parallel rectangles.

Theorem 1.9 (covering with rectangles, lower bound). *Consider the problem of covering a point set $\mathcal{P}$ by selecting k axis-parallel rectangles from a set $\mathcal{D}$.*

1. *Assuming ETH, there is no algorithm for this problem with running time $f(k) \cdot (|\mathcal{P}| + |\mathcal{D}|)^{o(k)}$ for any computable function f, even if each rectangle in $\mathcal{D}$ is of size $1 \times k$ or $k \times 1$.*
2. *Assuming ETH, for every $\epsilon_0 > 0$, there is no algorithm for this problem with running time $f(k) \cdot (|\mathcal{P}| + |\mathcal{D}|)^{o(k/\log k)}$ for any computable function f, even if each rectangle in $\mathcal{D}$ has both width and height in the range $[1 - \epsilon_0, 1 + \epsilon_0]$.*

This shows that even a minor deviation from the setting of Theorem 1.7 makes the existence of $n^{\mathcal{O}(\sqrt{k})}$ algorithms implausible. It seems that for covering problems, the square root phenomenon depends not on the objects being simple, or fat, or similar in size, but really on the fact that the objects are balls in a metric.

2 Geometric problems

Our main algorithmic result is a technique for solving a general facility location problem on planar graphs in time $n^{\mathcal{O}(\sqrt{k})}$. With simple reductions, we can use

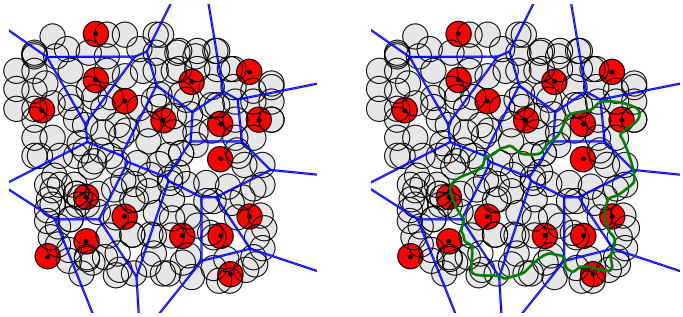
this algorithm to solve 2-dimensional geometric problems. However, our main algorithmic ideas can be implemented also directly in the geometric setting, giving self-contained geometric algorithms. These algorithms avoid some of the technical complications that arise in the planar graph counterparts, such as the Voronoi diagram having bridges or shortest paths sharing subpaths. The full algorithm appears in the full version [11], here we focus on these simpler cases.

Packing unit disks. We start with INDEPENDENT SET for unit disks: given a set $\mathcal{D}$ of closed disks of unit radius in the plane, the task is to select k disjoint disks. This problem is known to be solvable in time $n^{\mathcal{O}(\sqrt{k})}$ [4,12]. We present another $n^{\mathcal{O}(\sqrt{k})}$ algorithm for the problem, demonstrating how we can solve it recursively by focusing on the Voronoi diagram of a hypothetical solution.

The main combinatorial idea behind the algorithm is the following. Let $\mathcal{P}$ be a set of points in the plane. The *Voronoi region* of $p \in \mathcal{P}$ is the set of those points x in the plane that are "closest" to p in the sense that the distance of x and $\mathcal{P}$ is exactly the distance of x and p. Consider a hypothetical solution consisting of k independent disks and let us consider the Voronoi diagram of the centers of these k disks (see Figure 1(a)). To emphasize that we consider the Voronoi diagram of the centers of the k disks in the solution and *not* the centers of the n disks in the input, we call this diagram the *solution Voronoi diagram.* For simplicity, let us assume that the solution Voronoi diagram is a 2-connected 3-regular graph embedded on a sphere. We need a balanced separator theorem of the following form. A *noose* of a plane graph G is a closed curve δ on the sphere such that δ alternately travels through faces and vertices of G, and every vertex and face of G is visited at most once. It is possible to show that every 3-regular planar graph G with k faces has a noose δ of length $\mathcal{O}(\sqrt{k})$ (that is, going through $\mathcal{O}(\sqrt{k})$ faces and vertices) that is *face balanced*, in the sense that there are at most $\frac{2}{3}k$ faces of G strictly inside δ and at most $\frac{2}{3}k$ faces of G strictly outside δ.

Consider a face-balanced noose δ of length $\mathcal{O}(\sqrt{k})$ as above (see Figure 1(b)). Noose δ goes through $\mathcal{O}(\sqrt{k})$ faces of the solution Voronoi diagram, which correspond to a set Q of $\mathcal{O}(\sqrt{k})$ disks of the solution. The noose can be turned into a polygon Γ with $\mathcal{O}(\sqrt{k})$ vertices the following way (see Figure 1(c)). Consider a subcurve of δ that is contained in the face corresponding to disk $p \in Q$ and its endpoints are vertices x and y of the solution Voronoi diagram. Then we can "straighten" this subcurve by replacing it with straight line segments connecting the center of p with x and y. Thus, the vertices of polygon Γ are center points of disks in Q and vertices of the solution Voronoi diagram. Observe that Γ intersects the Voronoi regions of the points in Q only; this follows from the convexity of the Voronoi regions. In particular, among the disks in the solution, Γ does not intersect any disk outside Q.

The main idea is to use this polygon Γ to separate the problem into two subproblems. Of course, we do not know the solution Voronoi diagram and hence we have no way of computing from it the balanced noose δ and the polygon Γ. However, we can efficiently list $n^{\mathcal{O}(\sqrt{k})}$ candidate polygons. By definition, every vertex of the polygon Γ is either the center of a disk in $\mathcal{D}$ or a vertex of the solution Voronoi diagram. Every vertex of the solution Voronoi diagram is



(a) Voronoi diagram of the centers of disks in solution.

(b) A noose in the Voronoi diagram.

(c) The polygon corresponding to the noose.

(d) The problem breaks into two independent parts.

Fig. 1: Using a noose in the Voronoi diagram for divide and conquer.

equidistant from the centers of three disks in $\mathcal{D}$ and for any three such centers (in general position) there is a unique point in the plane equidistant from them. Thus every vertex of the polygon Γ is either a center of a disk in $\mathcal{D}$ or can be described by a triple of disks in $\mathcal{D}$, and hence Γ can be described by an $\mathcal{O}(\sqrt{k})$-tuple of disks from $\mathcal{D}$. By branching into $n^{\mathcal{O}(\sqrt{k})}$ directions, we may assume that we have correctly guessed the subset Q of the solution and the polygon Γ.

Provided Q is indeed part of the solution (which we assume), we may remove these disks from $\mathcal{D}$ and decrease the target number of disks by $|Q|$. We can also perform the following cleaning steps: (1) Remove any disk that intersects a disk in Q. (2) Remove any disk that intersects Γ. The correctness of the cleaning steps above follows directly from our observations on the properties of Γ.

After these cleaning steps, the instance falls apart into two independent parts: each remaining disk is either strictly inside Γ or strictly outside Γ (see Figure 1(d)). As δ was face balanced, there are at most $\frac{2}{3}k$ faces of the solution Voronoi diagram inside/outside δ, and hence the solution contains at most $\frac{2}{3}k$ disks inside/outside Γ. Therefore, for $k' := 1, \ldots, \lfloor \frac{2}{3}k \rfloor$, we recursively try to find exactly k' independent disks from the input restricted to the inside/outside Γ, resulting in $2 \cdot \frac{2}{3}k$ recursive calls. Taking into account the $n^{\mathcal{O}(\sqrt{k})}$ guesses for Q and

Γ, the number of subproblems we need to solve is $2 \cdot \frac{2}{3}k \cdot n^{\mathcal{O}(\sqrt{k})} = n^{\mathcal{O}(\sqrt{k})}$ and the parameter value is at most $\frac{2}{3}k$ in each subproblem. Therefore, the running time of the algorithm is governed by the recursion $T(n,k) = n^{\mathcal{O}(\sqrt{k})} \cdot T(n,(2/3)k)$, which solves to $T(n,k) = n^{\mathcal{O}(\sqrt{k})}$. This proves the first result: packing unit disks in the plane in time $n^{\mathcal{O}(\sqrt{k})}$. Let us repeat that this result was known before [4,12], but as we shall see, our algorithm based on Voronoi diagrams can be generalized to objects of different size, planar graphs, and covering problems.

Covering points by unit disks. Let us now consider the following problem: given a set $\mathcal{D}$ of unit disks and a set $\mathcal{C}$ of client points, we need to select k disks from $\mathcal{D}$ that together cover every point in $\mathcal{C}$. We show that this problem can be solved in time $n^{\mathcal{O}(\sqrt{k})}$ using a similar approach. Note that this time the disks in the solution are not necessarily disjoint, but this does not change the fact that their centers (which can be assumed to be distinct) define a Voronoi diagram. Therefore, it will be convenient to switch to an equivalent formulation of the problem described in terms of the centers of the disks: $\mathcal{D}$ is a set of points and we say that a selected point in $\mathcal{D}$ covers a point in $\mathcal{C}$ if their distance is at most 1.

As before, we can try $n^{\mathcal{O}(\sqrt{k})}$ possibilities to guess a set $Q \subseteq \mathcal{D}$ of center points and a polygon Γ corresponding to a face-balanced noose. The question is how to use Γ to split the problem into two independent subproblems. The cleaning steps (1) and (2) for the packing problem are no longer applicable: the solution may contain disks intersecting the disks with centers in Q as well as further disks intersecting Γ. Instead we do as follows. First, if we assume that Q is part of the solution, then any point in $\mathcal{C}$ covered by some point in Q can be removed. Second, we know that in the solution Voronoi diagram every point of Γ belongs to the Voronoi region of some point in Q. Hence we can remove any point from $\mathcal{D}$ that contradicts this assumption. That is, if there is $p \in \mathcal{D}$ and $v \in \Gamma$ such that v is closer to p than to every point in Q, then we can safely remove p from $\mathcal{D}$. Thus we have the following cleaning steps: (1) Remove every point of $\mathcal{C}$ covered by Q. (2) Remove every point of $\mathcal{D}$ that is closer to a point of Γ than every point in Q. Let $\mathcal{D}_{\text{in}}, \mathcal{D}_{\text{out}}$ ($\mathcal{C}_{\text{in}}, \mathcal{C}_{\text{out}}$) be the remaining points in $\mathcal{D}$ ($\mathcal{C}$) strictly inside and outside Γ, respectively. We know that the solution contains at most $\frac{2}{3}k$ center points inside/outside Γ. Hence, for $1 \leq k' \leq \lfloor \frac{2}{3}k \rfloor$, we solve two subproblems, with point sets $(\mathcal{D}_{\text{in}}, \mathcal{C}_{\text{in}})$ and $(\mathcal{D}_{\text{out}}, \mathcal{C}_{\text{out}})$.

If there is a set of k_{in} center points in $\mathcal{D}_{\text{in}}$ covering $\mathcal{C}_{\text{in}}$ and there is a set of k_{out} center points in $\mathcal{D}_{\text{out}}$ covering $\mathcal{C}_{\text{out}}$, then, together with Q, they form a solution of $|Q| + k_{\text{in}} + k_{\text{out}}$ center points. By solving the defined subproblems optimally, we know the minimum value of k_{in} and k_{out} required to cover $\mathcal{C}_{\text{in}}$ and $\mathcal{C}_{\text{out}}$, and hence we can determine the smallest solution that can be put together this way. But is it true that we can always put together an *optimum* solution this way? The problem is that, in principle, the solution may contain a center point $p \in \mathcal{D}_{\text{out}}$ that covers some point $q \in \mathcal{C}_{\text{in}}$ that is not covered by any center point in $\mathcal{D}_{\text{in}}$. In this case, in the optimum solution the number of center points selected from $\mathcal{D}_{\text{in}}$ can be strictly less than what is needed to cover $\mathcal{C}_{\text{in}}$.

Fortunately, we can show that this problem never arises. Suppose that there is such a $p \in \mathcal{D}_{\text{out}}$ and $q \in \mathcal{C}_{\text{in}}$. As p is outside Γ and q is inside Γ, the

segment connecting p and q intersects Γ at some point $v \in \Gamma$, which means $\mathrm{dist}(p,q) = \mathrm{dist}(p,v) + \mathrm{dist}(v,q)$. By cleaning step (2), there has to be a $p' \in Q$ such that $\mathrm{dist}(p',v) \leq \mathrm{dist}(p,v)$, otherwise p would be removed from $\mathcal{D}$. This means that $\mathrm{dist}(p,q) = \mathrm{dist}(p,v) + \mathrm{dist}(v,q) \geq \mathrm{dist}(p',v) + \mathrm{dist}(v,q) \geq \mathrm{dist}(p',q)$. Therefore, if p covers q, then so does $p' \in Q$. But in this case we would have removed q from $\mathcal{C}$ in the first cleaning step. Thus we can indeed obtain an optimum solution the way we proposed, by solving optimally the defined subproblems. Again we have $n^{\mathcal{O}(\sqrt{k})}$ subproblems, with parameter value at most $\frac{2}{3}k$. Hence, the same recursion applies to the running time, resulting in an $n^{\mathcal{O}(\sqrt{k})}$ time algorithm.

Packing in planar graphs. How can we translate the geometric ideas explained above to the context of planar graphs? Let G be an edge-weighted planar graph and let $\mathcal{F}$ be a set of disjoint "objects" — connected sets of vertices in G. Then we can define the analog of the Voronoi regions: for every $p \in \mathcal{F}$, let M_p contain every vertex v to which p is the closest object in $\mathcal{F}$, that is, $\mathrm{dist}(v,\mathcal{F}) = \mathrm{dist}(v,p)$. It is easy to verify that region M_p has the following convexity property: if $v \in M_p$ and P is a shortest path between v and p, then every vertex of P is in M_p.

While Voronoi regions are easy to define in graphs, the proper definition of Voronoi diagrams and the construction of polygon Γ are far from obvious. We omit the discussion of these technical details, and we only state in Lemma 2.1 below (a simplified version of) the main technical tool that is at the core of the algorithm. Note that the statement of Lemma 2.1 involves only the notion of Voronoi regions, hence there are no technical issues in interpreting and using it. However, in the proof we have to define the analog of the Voronoi diagram for planar graphs and address issues such that this diagram is not 2-connected etc.

Let us consider first the packing problem: given an edge-weighted graph G, a set $\mathcal{D}$ of d objects (connected subsets of vertices), and an integer k, find a subset $\mathcal{F} \subseteq \mathcal{D}$ of k disjoint objects. Looking at the algorithm for packing unit disks, what we need is a suitable *guarded separator*: a pair (Q,Γ) consisting of a set $Q \subseteq \mathcal{D}$ of $\mathcal{O}(\sqrt{k})$ objects and a subset $\Gamma \subseteq V(G)$ of vertices. If there is a hypothetical solution $\mathcal{F} \subseteq \mathcal{D}$ consisting of k disjoint objects, then a suitable guarded separator (Q,Γ) should satisfy the following three properties: (1) $Q \subseteq \mathcal{F}$, (2) Γ is fully contained in the Voronoi regions of the objects in Q, and (3) Γ separates the objects in $\mathcal{F}$ in a balanced way. Our main technical result is that it is possible to enumerate a set of $d^{\mathcal{O}(\sqrt{k})}$ guarded separators such that for every solution $\mathcal{F}$, one of the enumerated guarded separators satisfies these three properties. We state here a simplified version that is suitable for packing problems.

Lemma 2.1. *Let G be an n-vertex edge-weighted planar graph, $\mathcal{D}$ a set of d connected subsets of $V(G)$, and k an integer. We can enumerate (in time polynomial in the size of the output and n) a set $\mathcal{N}$ of $d^{\mathcal{O}(\sqrt{k})}$ pairs (Q,Γ) with $Q \subseteq \mathcal{D}$, $|Q| = \mathcal{O}(\sqrt{k})$, $\Gamma \subseteq V(G)$ such that the following holds. If $\mathcal{F} \subseteq \mathcal{D}$ is a set of k pairwise disjoint objects, then there is a pair $(Q,\Gamma) \in \mathcal{N}$ such that*

1. *$Q \subseteq \mathcal{F}$,*
2. *if $(M_p)_{p\in\mathcal{F}}$ are the Voronoi regions of $\mathcal{F}$, then $\Gamma \subseteq \bigcup_{p\in Q} M_p$,*

3. *for every connected component C of $G - \Gamma$, there are at most $\frac{2}{3}k$ objects of $\mathcal{F}$ that are fully contained in C.*

The proof goes along the same lines as the argument for the geometric setting. After carefully defining the analog of the Voronoi diagram, we can use the planar separator result to obtain a noose δ. As before, we "straighten" the noose into a closed walk in the graph using shortest paths connecting $\mathcal{O}(\sqrt{k})$ objects and $\mathcal{O}(\sqrt{k})$ vertices of the Voronoi diagram. The vertices of this walk separate the objects that are inside/outside the noose, hence it has the required properties. Thus by trying all sets of $\mathcal{O}(\sqrt{k})$ objects and $\mathcal{O}(\sqrt{k})$ vertices of the Voronoi diagram, we can enumerate a suitable set $\mathcal{N}$. A technical difficulty in the proof is that the definition of the vertices of the Voronoi diagram is nontrivial. Moreover, to achieve the bound $d^{\mathcal{O}(\sqrt{k})}$ instead of $n^{\mathcal{O}(\sqrt{k})}$, we need a more involved way of finding a set of $d^{\mathcal{O}(1)}$ candidate vertices; unlike in the geometric setting, enumerating vertices equidistant from three objects is not sufficient.

Armed with set $\mathcal{N}$ from Lemma 2.1, the packing problem can be solved in a way analogous to the case of unit disks. We guess a pair $(Q, \Gamma) \in \mathcal{N}$ that satisfies the properties of Lemma 2.1. Then objects of Q as well as those intersecting Γ can be removed from $\mathcal{D}$. In other words, we have to solve the problem on graph $G - \Gamma$, so we can focus on each connected component separately. However, we know that each such component contains at most $\frac{2}{3}k$ objects of the solution. Hence, for each component C of $G - \Gamma$ containing at least one object of $\mathcal{D}$ and for $k' = 1, \ldots, \lfloor \frac{2}{3}k \rfloor$, we recursively solve the problem on C with parameter k'. A similar reasoning as before shows that we can put together an optimum solution for the original problem from optimum solutions for the subproblems. As at most d components of $G - \Gamma$ contain objects from $\mathcal{D}$, we recursively solve at most $d \cdot \frac{2}{3}k$ subproblems for a given (Q, Γ). Hence, the total number of subproblems we solve is at most $d \cdot \frac{2}{3}k \cdot |\mathcal{N}| = d \cdot \frac{2}{3}k \cdot d^{\mathcal{O}(\sqrt{k})} = d^{\mathcal{O}(\sqrt{k})}$. The same analysis of the recurrence shows that the running time of the algorithm is $d^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$.

Covering in planar graphs. Let us consider now the following analog of covering points by unit disks: given an edge-weighted planar graph G, two sets of vertices $\mathcal{D}$ and $\mathcal{C}$, and integers k and r, the task is to find a set $\mathcal{F} \subseteq \mathcal{D}$ of k vertices that covers every vertex in $\mathcal{C}$. Here $p \in \mathcal{D}$ covers $q \in \mathcal{C}$ if $\mathrm{dist}(p, q) \leq r$, i.e., we can imagine that p represents a ball of radius r in the graph with center at p. Unlike in the case of packing, $\mathcal{D}$ is a set of vertices, not a set of connected sets.

Let $\mathcal{F}$ be a hypothetical solution. We can construct the set $\mathcal{N}$ given by Lemma 2.1 and guess a guarded separator (Q, Γ) satisfying the three properties. As we assume that Q is part of the solution, we remove from $\mathcal{C}$ every vertex that is covered by some vertex in Q; let $\mathcal{C}'$ be the remaining vertices. By the third property of Lemma 2.1, we can assume that in the solution $\mathcal{F}$, the set Γ is fully contained in the Voronoi regions of the vertices in Q. This means that if there is a $p \in \mathcal{D} \setminus Q$ and $v \in \Gamma$ such that $\mathrm{dist}(p, v) < \mathrm{dist}(p, Q)$, then p can be removed from $\mathcal{D}$. Let $\mathcal{D}'$ be the remaining set of vertices. For every component C of $G - S$ and $k' = 1, \ldots, \lfloor \frac{2}{3}k \rfloor$, we recursively solve the problem restricted to C, that is, with the restrictions $\mathcal{D}'[C]$ and $\mathcal{C}'[C]$ of the object and client sets. It is very important to point out that now (unlike how we did the packing problem)

we *do not* change the graph G in each call: we use the same graph G, only with the restricted sets $\mathcal{D}'[C]$ and $\mathcal{C}'[C]$. The reason is that restricting to the graph $G[C]$ can change the distances between vertices in C.

If k_C is the minimum number of vertices in $\mathcal{D}'[C]$ that can cover $\mathcal{C}'[C]$, then we know that there are $|Q| + \sum k_C$ vertices in $\mathcal{D}$ that cover every vertex in $\mathcal{C}$. As in the case of covering with disks, we argue that if there is a solution, then we can obtain a solution this way. The reasoning is basically the same: we just replace the Euclidean metric with the graph metric, and use the fact that the shortest path connecting any two points of $\mathcal{D}' \cup \mathcal{C}'$ lying in different components of $G - \Gamma$ must intersect Γ. As in the case of packing, we have at most $d \cdot \frac{2}{3}k \cdot d^{\mathcal{O}(\sqrt{k})}$ subproblems and the running time $d^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ follows the same way.

Nonuniform radius. A natural generalization of the covering problem is when every vertex $p \in \mathcal{D}$ is given a radius $r(p) \geq 0$ and p covers a $q \in \mathcal{C}$ if $\mathrm{dist}(p, q) \leq r(p)$. That is, now the vertices in $\mathcal{D}$ represent balls with possibly different radii.

There are two ways in which we can handle this more general problem. The first is a simple graph-theoretic trick. For every $p \in \mathcal{D}$, attach a path of length $R - r(p)$ to p, and replace p in $\mathcal{D}$ with the other end p' of this path, where $R = \max_{p' \in \mathcal{D}} r(p')$. Now a vertex $q \in \mathcal{C}$ is at distance at most $r(p)$ from p iff it is at distance at most R from p', so we can solve the problem by applying the algorithm for uniform radius R. The second way is somewhat more complicated, but it seems to be the robust solution of the issue. We can namely work with the *additively weighted* Voronoi diagram, that is, instead of defining the Voronoi regions of $\mathcal{F}$ by comparing distances $\mathrm{dist}(p, v)$ for $p \in \mathcal{F}$, we compare the weighted distances $\mathrm{dist}(p, v) - r(p)$. It can be verified that the main arguments of the algorithm, like convexity of the Voronoi regions or the separability of subproblems in the covering setting, all go through after redoing the same calculations.

3 The general problem

Suppose we are given an edge-weighted undirected graph G, a family of *objects* $\mathcal{D}$, and a family of *clients* $\mathcal{C}$. Every object $p \in \mathcal{D}$ has three attributes. It has its *location* $\mathbf{loc}(p)$, which is a nonempty subset of vertices of G such that $G[\mathbf{loc}(p)]$ is connected. It has its *cost* $\lambda(p)$, which is a real number (possibly negative). Finally, it has its *radius* $r(p)$, which is a nonnegative real value denoting the strength of domination imposed by p. Every client $q \in \mathcal{C}$ has three attributes. It has its *placement* $\mathbf{pla}(q)$, which is a vertex of G where the client resides. It has also its *sensitivity* $s(q)$, which is a real value denoting how sensitive the client is to domination from objects. Finally, it has *prize* $\pi(q)$, which is a real value denoting the prize for dominating the client. Note that there can be multiple clients placed in the same vertex and the prizes may be negative.

We say that a subfamily $\mathcal{F} \subseteq \mathcal{D}$ is *normal* if locations of objects from $\mathcal{F}$ are disjoint, and moreover $\mathrm{dist}(\mathbf{loc}(p_1), \mathbf{loc}(p_2)) > |r(p_1) - r(p_2)|$ for all pairs (p_1, p_2) of different objects in $\mathcal{F}$. In particular, normality implies disjointness of locations of objects from $\mathcal{F}$, but if all the radii are equal, then the two notions coincide. We say that a client q is *covered* by an object p if $\mathrm{dist}(\mathbf{pla}(q), \mathbf{loc}(p)) \leq s(q) + r(p)$.

We are finally ready to define Disjoint Network Coverage. As input we get an edge-weighted graph G embedded on a sphere, families of objects $\mathcal{D}$ and clients $\mathcal{C}$ (described using their attributes), and an integer k. For a subfamily $\mathcal{Z} \subseteq \mathcal{D}$, we define its *revenue* $\Pi(\mathcal{Z})$ as the total sum of prizes of clients covered by at least one object from $\mathcal{Z}$ minus the total sum of costs of objects from $\mathcal{Z}$. In the Disjoint Network Coverage problem, the task is to find a subfamily $\mathcal{Z} \subseteq \mathcal{D}$ such that the following holds: (1) family $\mathcal{Z}$ is normal and has cardinality *exactly* k and (2) subject to the previous constraint, family $\mathcal{Z}$ maximizes the revenue $\Pi(\mathcal{Z})$. The main result of this paper is the following theorem.

Theorem 3.1 (Main result). Disjoint Network Coverage *can be solved in time* $|\mathcal{D}|^{\mathcal{O}(\sqrt{k})} \cdot (|\mathcal{C}| \cdot |V(G)|)^{\mathcal{O}(1)}$.

References

1. A. Adamaszek and A. Wiese. Approximation schemes for maximum weight independent set of rectangles. In *FOCS*, pages 400–409, 2013.
2. A. Adamaszek and A. Wiese. A QPTAS for maximum weight independent set of polygons with polylogarithmically many vertices. In *SODA*, pages 645–656, 2014.
3. P. K. Agarwal, M. H. Overmars, and M. Sharir. Computing maximally separated sets in the plane. *SIAM J. Comput.*, 36(3):815–834, 2006.
4. J. Alber and J. Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *J. Algorithms*, 52(2):134–151, 2004.
5. R. H. Chitnis, M. Hajiaghayi, and D. Marx. Tight bounds for Planar Strongly Connected Steiner Subgraph with fixed number of terminals (and extensions). In *SODA*, pages 1782–1801, 2014.
6. E. D. Demaine and M. Hajiaghayi. The bidimensionality theory and its algorithmic applications. *Comput. J.*, 51(3):292–302, 2008.
7. S. Har-Peled. Quasi-polynomial time approximation scheme for sparse subsets of polygons. In *SoCG*, page 120, 2014.
8. R. Z. Hwang, R. C. T. Lee, and R. C. Chang. The slab dividing approach to solve the Euclidean p-center problem. *Algorithmica*, 9(1):1–22, 1993.
9. P. N. Klein and D. Marx. Solving Planar k-Terminal Cut in $O(n^{c\sqrt{k}})$ time. In *ICALP (1)*, pages 569–580, 2012.
10. P. N. Klein and D. Marx. A subexponential parameterized algorithm for Subset TSP on planar graphs. In *SODA*, pages 1812–1830, 2014.
11. D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. *CoRR*, abs/1504.05476, 2015.
12. D. Marx and A. Sidiropoulos. The limited blessing of low dimensionality: when $1 - 1/d$ is the best possible exponent for d-dimensional geometric problems. In *SoCG*, page 67, 2014.
13. N. H. Mustafa, R. Raman, and S. Ray. QPTAS for geometric set-cover problems via optimal separators. *CoRR*, abs/1403.0835, 2014.
14. M. Pilipczuk, M. Pilipczuk, P. Sankowski, and E. J. van Leeuwen. Network sparsification for Steiner problems on planar and bounded-genus graphs. In *FOCS*, pages 276–285. IEEE Computer Society, 2014.
15. M. Pătraşcu and R. Williams. On the possibility of faster SAT algorithms. In *SODA*, pages 1065–1075, 2010.