Linear-Time Approximation Algorithms for Geometric Intersection Graphs

Guilherme D. da Fonseca* Vinícius G. Pereira de Sá† Celina M. H. de Figueiredo‡

December 2, 2014

Abstract

Numerous approximation algorithms for problems on geometric intersection graphs have been proposed in the literature, exhibiting a sharp trade-off between running times and approximation ratios. We introduce a method to obtain linear-time constant-factor approximation algorithms for such problems. To illustrate its applicability, we obtain results for three well-known optimization problems on two classes of geometric intersection graphs. Among such results, our method yields linear-time $(4+\varepsilon)$-approximations for the maximum-weight independent set and the minimum dominating set of unit disk graphs, thus bringing dramatic performance improvements when compared to previous algorithms that achieve the same approximation ratios.

1 Introduction

Linear- and near-linear-time approximation algorithms constitute an active topic of research, even for problems that can be solved exactly in polynomial time, such as maximum flow and maximum matching [10, 14, 19, 24]. In this paper, we present a method to obtain linear-time constant-approximation algorithms for problems on geometric intersection graphs. We illustrate our method by presenting four algorithms, which cover three different optimization problems and two classes of geometric intersection graphs.

Geometric intersection graphs are graphs whose n vertices correspond to geometric objects and whose m edges correspond to pairs of intersecting objects. Several classes of geometric intersection graphs are defined by restricting the shape of the objects: disks, unit disks, squares, rectangles, etc. Approximation algorithms for such classes are either *graph-based*, when they receive as input solely the adjacency representation of the graph, or *geometric*, when the input consists of a geometric description of the objects. We note that when the goal is to design $O(n)$-time algorithms, the geometric representation is required, since the number m of edges in a geometric intersection graph can be as high as $\Theta(n^2)$.

Polynomial-time approximation schemes (PTASs) have been developed for several optimization problems on multiple classes of intersection graphs. Among such problems we shall focus on the following three in the present paper:

- *Maximum-weight independent set* (WIS): Given a graph with weighted vertices, find the maximum-weight subset of mutually non-adjacent vertices.

*gfonsecabr@gmail.com, Université Montpellier 2, France
†vigusmao@dcc.ufrj.br, Universidade Federal do Rio de Janeiro, Brazil
‡celina@cos.ufrj.br, Universidade Federal do Rio de Janeiro, Brazil

- *Minimum dominating set* (DS): Given a graph, find the minimum-cardinality vertex subset D such that every vertex not in D is adjacent to some vertex in D.
- *Minimum vertex cover* (VC): Given a graph, find the minimum-cardinality vertex subset C such that every edge is incident with some vertex in C.

The *shifting strategy* introduced in [16] gave rise to geometric PTASs for several problems on geometric intersection graphs [3, 9, 16, 17]. A set of n geometric objects has *constant diameter* if the Euclidean distance between any two points contained inside the objects is upper-bounded by a constant. Essentially, the shifting strategy reduces the original problem to a set of subproblems of constant diameter. Such a reduction takes $O(n)$ time and yields a $(1+\varepsilon)$-approximation to the original problem once the solutions to the subproblems are known. Each subproblem is then solved exactly by exploiting the fact that the point set has constant diameter. For example, we can often show by packing arguments that an input instance whose diameter is d admits a solution with $c = O(d^2)$ vertices, so that exhaustive enumeration can find the optimal solution in roughly $O(n^c)$ time. As a consequence, the running times of such PTASs are high-degree polynomials. Other PTASs that are not based on the shifting strategy also exist, but their complexities are usually even higher [23].

Among countless classes of geometric intersection graphs, unit disk graphs are arguably the most studied one, owing largely to their applicability in wireless networks [20, 23]. A *unit disk graph* (UDG) is the intersection graph of unit disks in the plane and it is often represented by the coordinates of the disk centers. With respect to this representation, two vertices are adjacent if the corresponding points (the disk centers) are within Euclidean distance at most 2 from one another. Next, we review the state of the art of the three aforementioned optimization problems on the class of UDGs.

The minimum dominating set problem (DS) on UDGs admits some PTASs [17, 23], the fastest of which is geometric and provides a $(1+\varepsilon)$-approximation in $n^{O(1/\varepsilon^2)}$ time. For the sake of comparison with the linear-time $(4+\varepsilon)$-approximation algorithm introduced in Section 4, such a PTAS produces a 4-approximation in roughly $O(n^{10})$ time. The high running times of the existing PTASs have therefore motivated the study of faster constant-factor approximation algorithms. Examples of graph-based algorithms include a 44/9-approximation that runs in $O(n+m)$ time and a 43/9-approximation that runs in $O(n^2 m)$ time [11]. Among the geometric algorithms, we cite the original 5-approximation, which can be implemented in $O(n)$ time if the floor function and constant-time hashing are available [20]; a 44/9-approximation that uses local improvements and runs in $O(n \log n)$ time [11]; a 4-approximation that uses grids and runs in $O(n^8 \log n)$ time [13]; and a recent 4-approximation that uses hexagonal grids and runs in $O(n^6 \log n)$ time [18].

The maximum-weight independent set problem (WIS) also admits PTASs for UDGs, the fastest of which is geometric and attains a $(1+\varepsilon)$-approximation in $O(n^{4\lceil 2/\varepsilon\sqrt{3}\rceil})$ time [17, 22, 23]. While such a PTAS produces a 4-approximation in $O(n^4)$ time, the algorithm presented in Section 3 obtains a $(4+\varepsilon)$-approximation in linear time. Alternatively, a geometric 5-approximation can be obtained in $O(n \log n)$ time by a greedy approach that considers the vertices in decreasing order of weights, or in $O(n+m)$ time in the graph-based version [20]. In contrast, for the unweighted version, a geometric greedy approach that considers the vertices from left to right [20] can be implemented to give a 3-approximation in $O(n)$ time with floor function and constant-time hashing. The high running time of the existing PTASs also motivated the recent discovery of an $O(n^3)$-time 2-approximation algorithm for the unweighted version [12].

The minimum vertex cover problem (VC) for UDGs is a rare example for which a geometric linear-time approximation scheme is known [21]. Previously, two high-complexity PTASs (one geometric and one graph-based) had been proposed [17, 25].

Another important class of geometric intersection graphs is defined by axis-aligned rectangles in the plane. Maximum-weight independent sets in such graphs are widely studied due to their application in data mining, map labeling, and VLSI design [2, 3, 7, 8]. However, the lack of constant-factor approximation algorithms has motivated the investigation of more restricted subclasses, for example the intersection graphs of squares and of unit-height rectangles [3, 8]. Also, PTASs for the WIS exist for the class of fat convex objects [7, 15], which generalizes unit disk graphs and several intersection graphs of rectangles. The fastest PTAS for the WIS on fat convex objects takes $n^{O(1/\varepsilon)}$ time and space, or alternatively $n^{O(1/\varepsilon^2)}$ time with $O(n)$ space [7].

Our results. We introduce a novel method to obtain linear-time approximation algorithms for problems on geometric intersection graphs, which we call the *shifting coresets method* (Section 2). The method is based on approximating the input set of geometric objects, which can be arbitrarily dense, by a *sparse* set of objects, that is, a set of objects such that any sufficiently small square contains at most a constant number of objects.

To obtain efficient algorithms through the application of our method, one needs to investigate the fundamental question of how well a sparse set—generated using only local information—can approximate a denser set for each considered problem. Thus, although the algorithms in this text share the same basic strategy, their analyses differ significantly. For example, the WIS analyses apply advanced graph-coloring theorems, whereas the DS analysis applies geometric packing arguments.

By using the shifting coresets method, we obtain linear-time $(4 + \varepsilon)$-approximation algorithms on UDGs for the WIS (which we tackle first, in Section 3, owing to its greater simplicity) and for the DS (Section 4). The proposed algorithms provide significant improvements when compared not only to existing linear-time algorithms, but also to subquartic-time algorithms (see Table 1 in Section 7). We also include (Section 5) a linear-time $(1 + \varepsilon)$-approximation obtained independently for the VC, which illustrates an indirect application of our method. Finally, we apply our method to obtain a linear-time $(6 + \varepsilon)$-approximation algorithm for the WIS on the class of intersection graphs of axis-aligned rectangles with side lengths in $[1, \lambda]$, for some constant $\lambda \geq 1$.

In Section 7, we discuss open problems and lower bounds to the approximation ratios of our algorithms.

2 The shifting coresets method

The shifting strategy [16] is the core of most of the existing geometric PTASs for problems on geometric intersection graphs. Generally, the shifting strategy reduces the original problem with n objects to a set of subproblems whose inputs have constant diameter and such that the sum of the input sizes is $O(n)$. Such a reduction is based on partitioning the objects according to a number of iteratively shifted grids and takes $O(n)$ time (by using the floor function and constant-time hashing). Exploiting the inputs' constant diameter, each subproblem is solved exactly in polynomial time. The solutions to the subproblems are then combined appropriately (normally in $O(n)$ time) to yield feasible solutions to the original problem, the best of which is returned. The high complexities of these geometric PTASs are due to the exact algorithms that are employed to solve each subproblem.

The shifting coresets method we introduce is based on the shifting strategy. However, it presents a crucial new aspect. Rather than obtaining exact, costly solutions for the subproblems, we solve each subproblem *approximately.* To do that, we employ the coresets paradigm [1], which consists in considering only a constant-size subset of the input.

For a problem whose input is a set P of n objects, the method can be briefly described as follows:

1. Apply the shifting strategy to construct a set of r subproblems with inputs $P_1, \dots, P_r$ such that $\sum_{i=1}^{r} |P_i| = O(n)$ and $\mathrm{diam}(P_i) = O(1)$ for all i.

2. For each subproblem instance P_i, obtain a coreset $Q_i \subseteq P_i$ with $|Q_i| = O(1)$, such that the optimal solution for instance Q_i is an α-approximation to the optimal solution for instance P_i.

3. Solve the problem exactly for each Q_i.

4. Combine the solutions into an $(\alpha + \varepsilon)$-approximation for the original problem.

Coresets for different problems must be devised on a case-by-case basis. For the WIS on UDGs, we create a grid with cells of diameter 0.29 and consider only one disk center of maximum weight inside each cell. For the DS on UDGs, we create a grid with cells of diameter 0.24 and consider only the (at most four) disk centers, inside each cell, with minimum or maximum coordinate in either dimension (breaking ties arbitrarily). We solve the VC on UDGs by breaking each subproblem into two cases. In the first one, the number of input disks is already bounded by a constant, therefore already a coreset. In the second one, we use the same coreset as in the WIS. Finally, we solve the WIS on a class of rectangle intersection graphs by representing each rectangle as a 4-dimensional point and using a 4-dimensional grid with cells of diameter 0.16. Our coreset is then formed by the rectangle of maximum weight inside each 4-dimensional cell.

We assume a real-RAM model of computation with floor function and constant-time hashing (as in [5]) so we can partition the input points into grid cells efficiently, yielding an overall $O(n)$ running time for the algorithms. Without these operations, the running time becomes $O(n \log n)$. We also assume that ε is constant. Otherwise, the running time becomes $2^{O(1/\varepsilon^2)}n$ for the WIS and the DS on UDGs, since their corresponding coresets contain $O(1/\varepsilon^2)$ points. For the VC, the running time becomes $2^{O(1/\varepsilon^3)}n$. As for the WIS on axis-aligned rectangles, if we assume that ε and λ (the maximum allowed side length) are both asymptotic variables, then the running time becomes $2^{O(\lambda^2/\varepsilon^2)}n$.

3 WIS on UDGs

In this section, we show how to apply the shifting coresets method to obtain a linear-time $(4 + \varepsilon)$-approximation to the WIS. We start by presenting a 4-approximation for point sets of constant diameter, and then we use the shifting strategy to obtain the desired $(4 + \varepsilon)$-approximation.

Given a point p and a set S of points, let $w(p)$ denote the weight of p, and let $w(S) = \sum_{p \in S} w(p)$. We say two or more points are *independent* if their minimum distance is strictly greater than 2.

Theorem 1. *Given a set P of n points with real weights as input, with* $\mathrm{diam}(P) = O(1)$, *the WIS for the corresponding UDG can be 4-approximated in $O(n)$ time in the real-RAM.*

Proof. Our algorithm proceeds as follows. First, we find the points of P with minimum or maximum coordinates in either dimension. That defines a bounding box of constant size for P. Within this bounding box, we create a grid with cells of diameter $\gamma = 0.29$ (any value $\gamma < (2-\sqrt{2})/2$ suffices). Note that the number of grid cells is constant, and therefore the points of P can be partitioned among the grid cells in $O(n)$ time (even without using the floor function or hashing). Then, we build the subset $Q \subseteq P$ as follows. For each non-empty grid cell C, we add to Q a point of maximum weight in $P \cap C$. Afterwards, we determine the maximum-weight independent set I^* of Q. Since $|Q| = O(1)$, this can be done in constant time. We return the solution I^*.

Next, we show that I^* is indeed a 4-approximation. We argue that, given an independent set $I \subseteq P$, there is an independent set $I' \subseteq Q$ with $4\ w(I') \geq w(I)$. Given a point $p \in P$, let $q(p)$ denote the point from Q that is contained in the same grid cell as p. Consider the set $S = \{q(p) : p \in I\}$. Note that $w(q(p)) \geq w(p)$ and $w(S) \geq w(I)$. The set S may not be independent, but since I is independent, the minimum distance in S is at least $2 - 2\gamma = 1.42 > \sqrt{2}$. We claim that the unit disk graph formed by S is a planar graph. To prove the claim, we show that a planar drawing can be obtained by connecting the points of S within distance at most 2 by straight line segments. Given a pair of points p_1, p_2 with distance $\|p_1p_2\| \leq 2$, the Pythagorean Theorem shows that a unit disk centered within distance greater than $\sqrt{2}$ from both p_1 and p_2 cannot intersect the segment p_1p_2. By the Four-Color Theorem [4], S admits a partition into four independent sets $S_1, \ldots, S_4$. The set I' of maximum weight among $S_1, \ldots, S_4$ must have weight at least $w(I)/4$.[1]

Since I^* is the maximum-weight independent set of Q, we have that I^* is a 4-approximation for the WIS. □

The following theorem uses the shifting strategy to obtain a $(4 + \varepsilon)$-approximation for point sets of arbitrary diameter. The proof uses the ideas from [17], presented in a different manner and including details about an efficient implementation of the strategy.

Theorem 2. *Given a set P of n points in the plane as input, the WIS for the corresponding UDG can be $(4+\varepsilon)$-approximated in $O(n)$ time on a real-RAM with constant-time hashing and the floor function. Without these operations, it can be done in $O(n \log n)$ time.*

Proof. Let k be the smallest integer such that

$$\left(\frac{k-2}{k}\right)^2 \geq \frac{4}{4+\varepsilon}. \tag{1}$$

Throughout this proof, we consider grids with square cells of side $2k$. We say a grid is *rooted at* a point (x, y) if there is a grid cell with corner at (x, y). Given a cell C, the square region $C^- \subset C$, called the *contraction* of C, is formed by removing from C the points within distance at most 2 from the boundary of C. Figure 1 illustrates these concepts.

The algorithm proceeds as follows. For i, j from 0 to $k - 1$, we create a grid with cells of side $2k$ rooted at $(2i, 2j)$. For each cell C in the grid, we run the WIS 4-approximation algorithm from Theorem 1 with point set $P \cap C^-$, obtaining a solution $I_{i,j}(C)$. Then, the independent set $I_{i,j}$ is constructed as the union of the independent sets $I_{i,j}(C)$ for all grid cells C. We return the maximum-weight set $I_{i,j}$ that is found, call it I^*.

To implement the algorithm efficiently, we create a subgrid of subcells of side 2, assigning each point to the subcell that contains it. In order to partition the n points into subcells, we use the floor function and constant-time hashing, taking $O(n)$ time. If these

[1]Note that the Four-Color Theorem is only used in the argument. No coloring is ever computed by the algorithm.

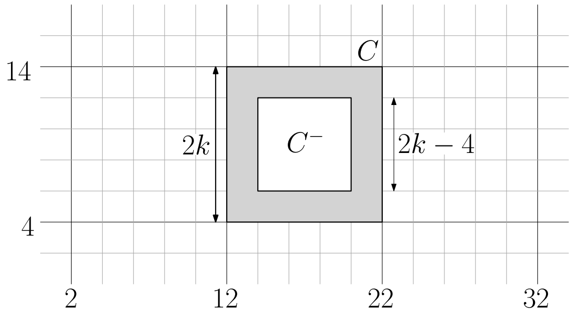


Figure 1: Grid rooted at $(2, 4)$ with $k = 5$ and the contraction of a cell

operations are not available, we determine the connected components of the graph (using the Delaunay triangulation, for example) and for each component we partition the points into subcells by sorting them by x coordinate, separating them into columns, and then sorting the points inside each column by y coordinate. The non-empty subcells are stored in a balanced binary search tree. This process takes $O(n \log n)$ time due to sorting, Delaunay triangulation, and binary search tree operations. Given the partitioning of the point set into subcells, each input to the WIS algorithm can be constructed as the union of a constant number of subcells. Finally, the total size of the constant-diameter WIS instances is $O(n)$, since each point from the original point sets appears in a constant number—a function of the fixed ε—of such instances.

To prove that the returned solution I^* is indeed a $(4 + \varepsilon)$-approximation, we use a probabilistic argument. Let i, j be picked uniformly at random from $0, \dots, k - 1$ and let OPT denote the optimal solution. For every cell C, we have

$$w(I_{i,j}(C)) \geq \frac{1}{4}\, w(OPT \cap C^-).$$

Consequently, by summing over all grid cells,

$$w(I_{i,j}) = \sum_C w(I_{i,j}(C)) \geq \frac{1}{4} \sum_C w(OPT \cap C^-) = \frac{1}{4} \sum_{p \in OPT} \rho(p)\, w(p),$$

where $\rho(p)$ denotes the probability that a given point p is contained in some contracted cell. Because such probability is the same for all points, we can bound $\mathrm{E}[w(I_{i,j})]$ by writing

$$\mathrm{E}[w(I_{i,j})] \geq \frac{1}{4}\, \rho(p) \sum_{p \in OPT} w(p) = \frac{1}{4}\, \rho(p)\, w(OPT).$$

Note that, for all $p \in P$, $\rho(p)$ corresponds to the ratio between the areas of C^- and C, namely

$$\rho(p) = \frac{\mathrm{area}(C^-)}{\mathrm{area}(C)} = \left(\frac{k-2}{k}\right)^2.$$

Therefore, by using inequality (1), we obtain

$$\mathrm{E}[w(I_{i,j})] \geq \frac{1}{4} \left(\frac{k-2}{k}\right)^2 w(OPT) \geq \frac{1}{4+\varepsilon}\, w(OPT).$$

Since I^* has maximum weight among the independent sets $I_{i,j}$, it follows that $w(I^*)$ is at least as large as their average weight. Therefore, I^* satisfies

$$w(I^*) \geq \mathrm{E}[w(I_{i,j})] \geq \frac{1}{4+\varepsilon}\, w(OPT),$$

closing the proof. $\square$

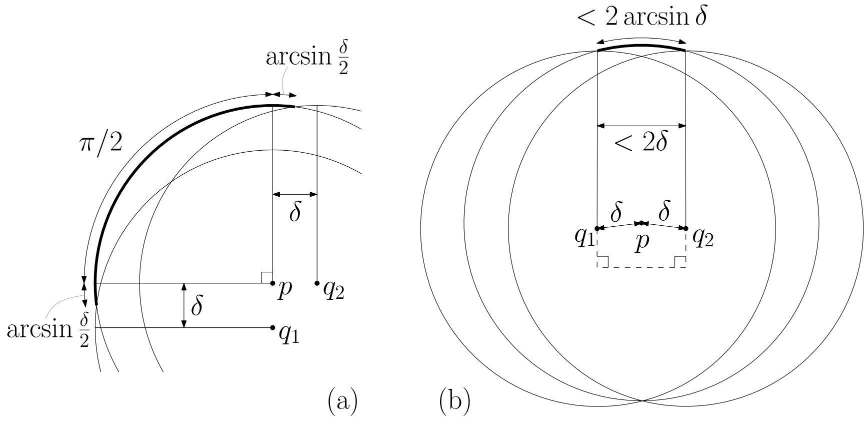


Figure 2: Proof of Lemma 3

4 DS for UDGs

In this section, we show how to apply the shifting coresets method to obtain a linear-time $(4+\varepsilon)$-approximation to the DS (in fact, a generalization of it). We start by presenting a 4-approximation for point sets of constant diameter, and then we use the shifting strategy to obtain the desired $(4+\varepsilon)$-approximation. We say that a point p dominates a point q if $\|pq\| \leq 2$. Given two sets of points D and P', we say that D is a P'*-dominating set* if every point in P' is dominated by some point in D.

We now define a more general version of the DS, which we refer to as the *minimum partial dominating set problem* (PDS). Such a generalization is necessary to properly apply the shifting strategy. In the PDS, we are given a set P of n points and also a subset $P' \subseteq P$. The goal is to find the smallest P'-dominating subset $D \subseteq P$.

In order to analyze our algorithm, we prove a geometric lemma that shows that the set-theoretic difference between a unit circle and two unit disks that are sufficiently close to it and form a sufficiently big angle consists of one or two "small" arcs. Given a point p, let $\bigcirc_p$ denote the unit disk centered at p, and $\partial\bigcirc_p$ denote its boundary circle.

Lemma 3. *Given $\delta > 0$ and three points $p, q_1, q_2 \in \mathbb{R}^2$ with (i) $\|pq_1\| \leq \delta$, (ii) $\|pq_2\| \leq \delta$, and (iii) the smallest angle $\angle q_1pq_2$ is greater than or equal to $\pi/2$, we have that:*

(1) the portion $T = (\partial\bigcirc_p) \setminus (\bigcirc_{q_1} \cup \bigcirc_{q_2})$ of the boundary $\partial\bigcirc_p$ consists of one or two circular arcs;

(2) if T consists of one circular arc, then the arc length is less than or equal to $\pi/2 + 2\arcsin(\delta/2)$; and

(3) if T consists of two circular arcs, then each arc length is less than $2\arcsin\delta$.

Proof. Statement (1) is clearly true. We start by proving statement (2). The arc length $\|T\|$ is maximized as the angle $\angle q_1pq_2$ decreases while the distances $\|pq_1\|, \|pq_2\|$ are kept constant, therefore it suffices to consider the case in which $\angle q_1pq_2 = \pi/2$. The arc T centered at p can be decomposed into three arcs by rays in directions q_1p and q_2p, as shown in Figure 2(a). The central arc measures $\pi/2$, while each of the other two arcs measures $\arcsin(\delta/2)$, proving statement (2).

Next, we prove statement (3). Let T_1, T_2 denote the two arcs that form T with $\|T_1\| \geq \|T_2\|$. The arc length $\|T_1\|$ is maximized in the limit when $\|T_2\| = 0$, as shown in Figure 2(b). The rays connecting q_1 and q_2 to the two extremes of T_1 are parallel, and therefore $\|T_1\| < 2\arcsin\delta$. $\square$

We are now able to prove the following theorem, which presents our 4-approximation algorithm for point sets of constant diameter.

Theorem 4. *Given two sets of points P and P' as input, with $P' \subseteq P$, $|P| = n$, and* $\operatorname{diam}(P) = O(1)$*, the PDS can be* 4*-approximated in $O(n)$ time in the real-RAM.*

Proof. First, we determine a bounding box of constant size for P, as we did in the algorithm for the WIS. Within this bounding box, we create a grid with cells of diameter $\gamma = 0.24$ (any positive γ satisfying

$$\gamma + \sqrt{8 - 8\cos\left(\frac{\frac{\pi}{2} + 2\arcsin(\frac{\gamma}{2})}{2}\right)} < 2$$

suffices) Note that the number of grid cells is constant, and therefore the points of P can be partitioned among the grid cells in $O(n)$ time (even without using the floor function or hashing). Then, we build the subset $Q \subseteq P$ as follows. For each non-empty grid cell, we add to Q the (at most four) extreme points inside the cell, i.e., those presenting minimum or maximum coordinate in either dimension. Ties are broken arbitrarily. Since there is a constant number of grid cells and we include in Q at most four points per cell, we have $|Q| = O(1)$. Afterwards, we determine the smallest P'-dominating subset $D^* \subseteq Q$. To do that, we examine the subsets of Q, from smallest to largest, verifying if all points of P' are dominated, until we find the dominating set D^*, which is returned as the approximate solution. Since Q has a constant number of points, this procedure takes $O(n)$ time.

Now we show that the returned solution D^* is indeed a 4-approximation. We argue that, given a P'-dominating set $D \subseteq P$, there is a P'-dominating set $D' \subseteq Q$ with $|D'| \leq 4\,|D|$. To build the set D' from D, we proceed as follows. For each point $p \in D$, if $p \in Q$, we add p to D'. Otherwise, since the set Q contains points of extreme coordinates in both x and y axes, in the cell of p, there are two points $q_1, q_2 \in Q$ such that (i) $\|pq_1\| \leq \gamma$, (ii) $\|pq_2\| \leq \gamma$, and (iii) the smallest angle $\angle q_1pq_2$ is at least $\pi/2$. We add these two points q_1, q_2 to D'.

By Lemma 3, the portion $T = (\partial\bigcirc_p) \setminus (\bigcirc_{q_1} \cup \bigcirc_{q_2})$ of $\partial\bigcirc_p$ consists of one or two circular arcs. We first consider the case in which T consists of one circular arc. Let R be the set of points from P' which are dominated by p, but not by q_1 or q_2. If R is empty, then no extra point needs to be added to D'. Otherwise, the line ℓ which contains p and bisects T separates R into two (possibly empty) sets R_1, R_2. If $R_1 \neq \emptyset$, let p_3 be an arbitrary point of R_1. Since Q contains a point in the same cell as p_3, there is a point q_3 with $\|p_3q_3\| \leq \gamma$. We add the point q_3 to D'. Analogously, if $R_2 \neq \emptyset$, let p_4 be an arbitrary point of R_2 and let $q_4 \in Q$ be a point with $\|p_4q_4\| \leq \gamma$. We add the point q_4 to D'.

We now show that the four points $q_1, q_2, q_3, q_4 \in Q$ dominate all points dominated by p. Consider a point v that is dominated by p but not by q_1 or q_2. The point v must be inside the circular crown sector depicted in Figure 3(a) and described as follows. Because v is dominated by p, we have $\|pv\| \leq 2$. By Lemma 3, the arc length $\|T\| < 1.82$. Also, $\|pv\| > 1$, because otherwise the unit circles centered at p and v would intersect forming an arc of length at least $2\pi/3$, which is greater than $\|T\|$, in which case v is dominated by q_1 or q_2. Finally, since v is closer to p than it is to q_1 or q_2, it follows that v must be between the lines that connect p to the endpoints of T. This circular crown sector is bisected by the line ℓ. Using the law of cosines, we calculate the diameter of each circular crown sector as $d = \sqrt{8 - 8\cos(\|T\|/2)} < 1.76$. Therefore, for any point v inside the circular crown sector, the point q_3 (or q_4, analogously) that is within distance at most γ from a point inside the same sector dominates v, as $\|vq_3\| \leq d + \gamma < 2$.

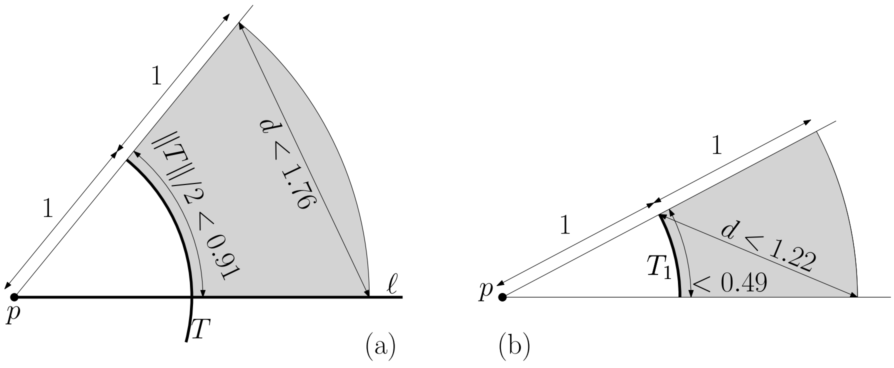


Figure 3: Proof of Theorem 4

Finally, if T consists of two circular arcs T_1, T_2 centered in p, then we start by adding those same points q_1, q_2 to D', as if T consisted of only one arc. Then, if necessary, we add new points q_3, q_4 to D' as follows. The points that are dominated by p but not by q_1 or q_2 must be within distance 1 of either T_1 or T_2. Let p_3, p_4 be arbitrary points that are within distance 1 of T_1 or T_2, respectively, but are not dominated by q_1 or q_2. If such points p_3, p_4 exist, then there are two points q_3, q_4 in Q that are within distance at most γ from respectively p_3, p_4. By Lemma 3, the largest arc among T_1, T_2 measures at most 0.49. The proof that all points dominated by p are dominated by q_1, q_2, q_3, or q_4 is analogous to the case in which T consists of a single arc, using the circular crown sector illustrated in Figure 3(b).

Since D^* is minimum among all subsets of Q that are P'-dominating sets, D^* is a 4-approximation for the PDS. □

The following theorem uses the shifting strategy [17] to obtain a $(4+\varepsilon)$-approximation for point sets of arbitrary diameter.

Theorem 5. *Given two sets of points P and P' as input, with $P' \subseteq P$ and $|P| = n$, the PDS can be $(4+\varepsilon)$-approximated in $O(n)$ time on a real-RAM with constant-time hashing and the floor function. Without these operations, it can be done in $O(n \log n)$ time.*

Proof. Let k be the smallest integer such that

$$\left(\frac{k+2}{k}\right)^2 \leq 1 + \frac{\varepsilon}{4}. \tag{2}$$

Throughout this proof, we consider grids with square cells of side $2k$. We say a grid is *rooted at* a point (x, y) if there is a grid cell with corner at (x, y). Given a cell C, the square region C^+, called the *expansion* of C, is formed by C and all points within L_∞ distance at most 2 from C. Figure 4 illustrates these concepts.

The algorithm proceeds as follows. For i, j from 0 to $k-1$, we create a grid with cells of side $2k$ rooted at $(2i, 2j)$ and, for each cell C in the grid, we use Theorem 4 to 4-approximate the PDS with point sets $P \cap C^+, P' \cap C$, obtaining a solution $D_{i,j}(C)$. The dominating set $D_{i,j}$ is constructed as the union of the dominating sets $D_{i,j}(C)$ for all grid cells C. We return the smallest dominating set $D_{i,j}$ that is found, call it D^*.

To prove that the returned solution is indeed a $(4+\varepsilon)$-approximation, we use a probabilistic argument. Let i, j be picked uniformly at random from $0, \ldots, k-1$ and let OPT denote the optimal solution. For every cell C, we have

$$|D_{i,j}(C)| \leq 4 \, \left|OPT \cap C^+\right|.$$

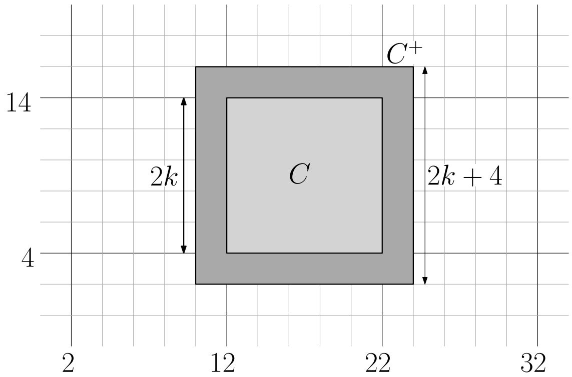


Figure 4: Grid rooted at $(2, 4)$ with $k = 5$ and the expansion of a cell

Consequently, by summing over all grid cells,

$$|D_{i,j}| = \sum_C |D_{i,j}(C)| \le 4 \sum_C \left| OPT \cap C^+ \right|.$$

We now bound $\mathrm{E}[|D_{i,j}|]$. To do that, we define $\mathcal{C}^+(p)$ as the collection of all cell expansions containing a point p, so we can write

$$\frac{\mathrm{E}[|D_{i,j}|]}{4} \le \mathrm{E}\left[\sum_C \left| OPT \cap C^+ \right|\right] = \mathrm{E}\left[\sum_{p \in OPT} |\mathcal{C}^+(p)|\right] = \sum_{p \in OPT} \mathrm{E}\left[|\mathcal{C}^+(p)|\right]$$

by the linearity of expectation. Note that the expected size of $\mathcal{C}^+(p)$, for all $p \in P$, corresponds to the ratio between the areas of C^+ and C, namely

$$\mathrm{E}\left[|\mathcal{C}^+(p)|\right] = \frac{\mathrm{area}(C^+)}{\mathrm{area}(C)} = \left(\frac{k+2}{k}\right)^2.$$

Therefore, by using inequality (2), we obtain

$$\mathrm{E}[|D_{i,j}|] \le 4 \left(\frac{k+2}{k}\right)^2 |OPT| \le 4 \left(1 + \frac{\varepsilon}{4}\right) |OPT| = (4 + \varepsilon)\ |OPT|.$$

Since the smallest among the dominating sets $D_{i,j}$ has no more than their average number of elements, the set D^* returned by the algorithm satisfies

$$|D^*| \le \mathrm{E}[|D_{i,j}|] \le (4 + \varepsilon)\ |OPT|,$$

closing the proof. □

The DS is the special case of the PDS in which $P' = P$, and thus it can be $(4 + \varepsilon)$-approximated in linear time by the same algorithm.

Our algorithm can be generalized to $(4 + \varepsilon)$-approximate the minimum distance d dominating set for constant d. In the *minimum distance d dominating set problem*, we are given a graph and an integer d, and the goal is to find a minimum subset of vertices such that all graph vertices are within distance at most d from a vertex in the subset (here d is the graph distance, that is, the number of edges in a shortest path). The DS is a special case in which $d = 1$. In contrast, the greedy heuristic that gives a 5-approximation to the DS gives an $f(d)$-approximation to the distance d version, where $f(d)$ is a quadratic function.

5 VC for UDGs

In this section, we show how to obtain a linear-time approximation scheme to the VC. We start by presenting an approximation scheme for point sets of constant diameter, and then we use the shifting strategy to generalize the result to arbitrary diameter. Differently than in the previous two problems, the size of a minimum vertex cover for a point set of constant diameter is not upper-bounded by a constant. Therefore, strictly speaking, a coreset for the problem does not exist. Nevertheless, it is possible to use coresets to approach the problem indirectly.

Given a graph $G = (V, E)$ with n vertices, it is well known that I is an independent set if and only if $V \setminus I$ is a vertex cover. While a maximum independent set corresponds to a minimum vertex cover, a constant approximation to the maximum independent set does not necessarily correspond to a constant approximation to the minimum vertex cover. However, in certain cases, an even stronger correspondence holds, as we show in the following proof.

Theorem 6. *Given a set P of n points as input, with* $\mathrm{diam}(P) = O(1)$*, the VC can be $(1+\varepsilon)$-approximated in $O(n)$ time in the real-RAM, for constant $\varepsilon > 0$.*

Proof. Our algorithm considers two cases, depending on the value of n. If

$$n < \left(1 + \frac{3}{4\varepsilon}\right) \frac{(\mathrm{diam}(P) + 2)^2}{4},$$

then n is constant, and we can solve the VC optimally in constant time.

Otherwise, we use Theorem 1 to obtain a 4-approximation I to the maximum independent set. We now show that $V = P \setminus I$ is a $(1+\varepsilon)$-approximation to the minimum vertex cover. Let I_{OPT}, V_{OPT} respectively be the maximum independent set and the minimum vertex cover. Note that $|V| = n - |I|$ and $|V_{OPT}| = n - |I_{OPT}|$. By a simple packing argument, dividing the area of a disk of diameter $\mathrm{diam}(P) + 2$ by the area of a unit disk,

$$|I_{OPT}| \leq \frac{(\mathrm{diam}(P) + 2)^2}{4},$$

and consequently

$$n \geq \left(1 + \frac{3}{4\varepsilon}\right) |I_{OPT}| = \left(1 + \frac{3}{4\varepsilon}\right) (n - |V_{OPT}|).$$

Manipulating the previous inequality, we obtain

$$n \leq \frac{4\varepsilon + 3}{3}\, |V_{OPT}|. \tag{3}$$

Since I is a 4-approximation to I_{OPT},

$$|V| = n - |I| \leq n - \frac{|I_{OPT}|}{4} = \frac{4n - |I_{OPT}|}{4} = \frac{3n + |V_{OPT}|}{4}. \tag{4}$$

Combining (3) and (4), we can write $|V| \leq (1+\varepsilon)|V_{OPT}|$, as desired. □

Using the shifting strategy we obtain the following result.

Theorem 7. *Given a set P of n points in the plane as input, the VC for the corresponding UDG can be $(1+\varepsilon)$-approximated in $O(n)$ time on a real-RAM with constant-time hashing and the floor function, for constant $\varepsilon > 0$. Without these operations, it can be done in $O(n \log n)$ time.*

Proof. Let k be the smallest integer such that

$$\left(\frac{k+2}{k}\right)^2 \leq \frac{1+\varepsilon}{1+\frac{\varepsilon}{2}}. \tag{5}$$

Throughout this proof, we consider grids with square cells of side $2k$. We say a grid is *rooted at* a point (x, y) if there is a grid cell with corner at (x, y). Given a cell C, the square region C^+, called the *expansion* of C, is formed by C and all points within L_∞ distance at most 2 from C.

The algorithm proceeds as follows. For i, j from 0 to $k-1$, we create a grid with cells of side $2k$ rooted at $(2i, 2j)$ and, for each cell C in the grid, we use Theorem 6 to $(1+\varepsilon/2)$-approximate the VC for $P \cap C^+$, obtaining a solution $V_{i,j}(C)$. The vertex cover $V_{i,j}$ is constructed as the union of the vertex covers $V_{i,j}(C)$ for all grid cells C. We return the smallest vertex cover $V_{i,j}$ that is found, call it V^*.

To prove that the returned solution is indeed a $(1+\varepsilon)$-approximation, we use a probabilistic argument. Let i, j be picked uniformly at random from $0, \ldots, k-1$ and let OPT denote the optimal solution. For every cell C, we have

$$|V_{i,j}(C)| \leq \left(1+\frac{\varepsilon}{2}\right) \left|OPT \cap C^+\right|.$$

Consequently, by summing over all grid cells,

$$|V_{i,j}| = \sum_C |V_{i,j}(C)| \leq \left(1+\frac{\varepsilon}{2}\right) \sum_C \left|OPT \cap C^+\right|.$$

We now bound $\mathrm{E}[|V_{i,j}|]$. To do that, we define $\mathcal{C}^+(p)$ as the collection of all cell expansions containing a point p, so we can write

$$\frac{\mathrm{E}[|V_{i,j}|]}{1+\frac{\varepsilon}{2}} \leq \mathrm{E}\left[\sum_C \left|OPT \cap C^+\right|\right] = \mathrm{E}\left[\sum_{p\in OPT} |\mathcal{C}^+(p)|\right] = \sum_{p\in OPT} \mathrm{E}\left[|\mathcal{C}^+(p)|\right]$$

by the linearity of expectation. The expected size of $\mathcal{C}^+(p)$, for all $p \in P$, corresponds to the ratio between the areas of C^+ and C, namely

$$\mathrm{E}\left[|\mathcal{C}^+(p)|\right] = \frac{\mathrm{area}(C^+)}{\mathrm{area}(C)} = \left(\frac{k+2}{k}\right)^2.$$

Therefore, by using inequality (5), we obtain

$$\mathrm{E}[|V_{i,j}|] \leq \left(1+\frac{\varepsilon}{2}\right)\left(\frac{k+2}{k}\right)^2 |OPT| \leq (1+\varepsilon)\ |OPT|.$$

Since the smallest among the vertex covers $V_{i,j}$ has no more than their average number of elements, the set V^* returned by the algorithm satisfies

$$|V^*| \leq \mathrm{E}[|V_{i,j}|] \leq (1+\varepsilon)\ |OPT|,$$

closing the proof. □

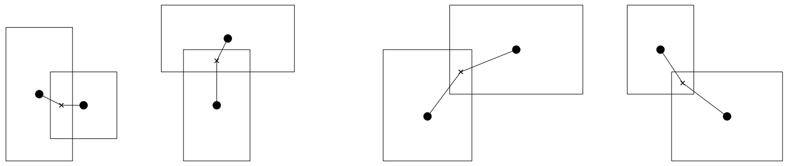


Figure 5: Examples of (a) type-1 and (b) type-2 edges

6 WIS for Rectangles

In this section, we consider the case in which the input is no longer a set of points, but a set of rectangles instead. Let λ be a constant and S a set of axis-aligned rectangles $R_1, \ldots, R_n$ in the plane, such that each rectangle R_q, for $q = 1, \ldots, n$, has width and height between 1 and λ, and weight $w(R_q)$. Let G be the intersection graph of S. We apply the shifting coresets method to obtain a linear-time $(6+\varepsilon)$-approximation algorithm to the maximum-weight independent set of G.

We define the *overlap* of two rectangles R_q, R_s as the minimum (horizontal or vertical) translation distance necessary to make the interiors of R_q and R_s disjoint. The following lemma bounds the chromatic number of the intersection graph of rectangles with a small overlap.

Lemma 8. *If S is a set of axis-aligned rectangles such that*

(1) the width and height of each rectangle is at least 1*, and*

(2) overlap$(R_q, R_s) < 1/3$ *for every two distinct rectangles* R_q, R_s*,*

then the intersection graph G of S is 6*-colorable.*

Proof. Let S be a set as required and G its intersection graph. A graph is 1-*planar* if it can be drawn on the plane in a way that each edge intersects at most one other edge. Borodin [6] showed that 1-planar graphs are 6-colorable. We prove the lemma by providing such a 1-planar drawing for G.

For each rectangle $R_q \in S$ we draw the vertex v_q on the center of R_q. Given two intersecting rectangles R_q, R_s, the edge $v_q v_s$ is drawn as two straight line segments, connecting v_q to the center of the rectangle $R_q \cap R_s$ and then to v_s. We show that at most one other edge may cross the edge $v_q v_s$. We note that the edge $v_q v_s$ is completely inside the region $R_q \cup R_s$.

When two rectangles $R_q, R_s \in S$ intersect one another, there are two possible types of edges corresponding to the relative positions of the rectangles (Figure 5):

1. R_q contains two corners of R_s (or vice-versa); or
2. R_q contains one corner of R_s, and vice-versa.

We show that a type-1 edge $v_q v_s$ cannot possibly be crossed by any other edge. Indeed, if R_q contains two corners of R_s, then the straight segment from v_s to the center of $R_q \cap R_s$ belongs to an axis-aligned line ℓ that bisects R_s. This segment cannot be crossed by any edge $v_{q'} v_{s'}$. Otherwise, the centers of the intersecting rectangles $R_{q'}, R_{s'} \in S$ would belong to distinct halfplanes defined by ℓ, and at least one of these rectangles, say $R_{q'}$, should cross ℓ. Since, by the maximum overlap allowed in S, the length of the segment of ℓ which can be contained in $R_{q'}$ measures at most $2/3$ (i.e., $1/3$ inside R_q plus $1/3$ inside R_s), a

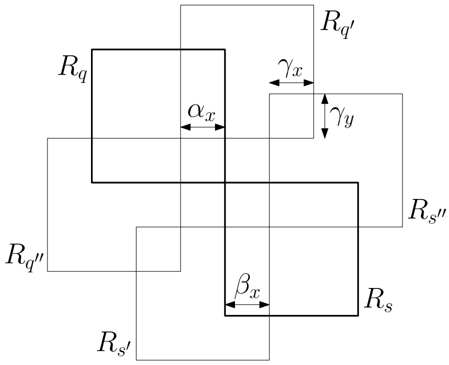


Figure 6: Crossing twice a type-2 edge

contradiction ensues, because $R_{q'}$ measures at least 1 on both dimensions. The straight segment from v_r to the center of $R_q \cap R_s$, on its turn, cannot be crossed by any other edge $v_{q'}v_{s'}$ because, since the edge $v_{q'}v_{s'}$ is drawn completely inside the region $R_q \cup R_s$, one of the rectangles, say $R_{q'}$, should intersect that segment, yet it should not contain any of its points whose distance to the border of R_q that intersects R_s is greater than $1/3$. But now, since both sides of $R_{q'}$ are greater than 1, we have *overlap*$(R_s, R_{q'}) \geq 2/3$, a contradiction.

It remains to show that a type-2 edge v_qv_s can be crossed by at most one other edge (which must also be a type-2 edge). Suppose, for the sake of contradiction, that v_qv_s is crossed by two other edges, $v_{q'}v_{q''}$ and $v_{s'}v_{s''}$, as illustrated in Figure 6. Now, without loss of generality, the maximum allowed overlap implies that:

- the width α_x of $R_q \cap R_{q'}$ satisfies $\alpha_x < 1/3$;
- the width β_x of $R_s \cap R_{s'}$ satisfies $\beta_x < 1/3$;
- the width γ_x and height γ_y of $R_{q'} \cap R_{s''}$ satisfy either $\gamma_x < 1/3$ or $\gamma_y < 1/3$.

If $\gamma_x < 1/3$, then the width of $R_{q'}$ is $\alpha_x + \beta_x + \gamma_x < 1$, a contradiction. If, on the other hand, $\gamma_y < 1/3$, then an analogous contradiction ensues on the y direction. □

Given a set S of rectangles, the *diameter* $\text{diam}(S)$ is the maximum distance between two vertices of the rectangles in S. We are now able to prove the following theorem, which presents our 6-approximation algorithm for sets of constant diameter.

Theorem 9. *Given a set S of n axis-aligned weighted rectangles as input, such that $\text{diam}(S) = O(1)$ and each rectangle in S has width and height at least 1, the WIS on the intersection graph of S can be 6-approximated in $O(n)$ time in the real-RAM.*

Proof. We represent each rectangle $R_q \in S$ by four real values (x_q, y_q, w_q, h_q), corresponding respectively to the x and y coordinates of its center, its width, and its height. The set S can thus be seen as a constant-diameter set of points in $\mathbb{R}^4$. We create a 4-dimensional grid with cells of diameter $\delta = 0.16$ (any positive $\delta < 1/6$ suffices), and we define the set S' by choosing the element of S with maximum weight inside each non-empty grid cell. Note that since $\text{diam}(S) = O(1)$, we have $|S'| = O(1)$, so we can compute the maximum-weight independent set among the points of S' in constant-time by brute force. We return such a set.

To prove the returned solution is indeed a 6-approximation, we show that, given an independent set $I \subseteq S$, there is an independent set $I' \subseteq S'$ such that $|I'| \geq |I|/6$. Let

$J' \subseteq S'$ be the set of rectangles obtained by selecting, for each rectangle $R_i \in I$, the rectangle $R'_i \in S'$ whose corresponding 4-dimensional point lies inside the same grid cell as that of R_i. Note that since S' contains the maximum weight rectangle inside each grid cell, we have $w(J') \geq w(I)$, even though J' may not be an independent set. However, the rectangles in J' overlap by less than 1/3 and we can use Lemma 8 to partition J' into 6 independent sets. Let I' be set of maximum weight among the partitions. Since $6w(I') \geq w(J')$, we have $6w(I') \geq w(I)$, proving the theorem. □

Using the shifting strategy, we extend the result for sets of arbitrary diameter.

Theorem 10. *Let $\lambda \geq 1$ be a constant. Given a set S of n axis-aligned weighted rectangles as input, such that each rectangle in S has width and height between 1 and λ, the WIS can be $(6+\varepsilon)$-approximated in $O(n)$ time on a real-RAM with constant-time hashing and the floor function. Without these operations, it can be done in $O(n \log n)$ time.*

Proof. Let k be the smallest multiple of λ such that

$$\left(\frac{k-\lambda}{k}\right)^2 \geq \frac{6}{6+\varepsilon}, \tag{6}$$

and let $k' = k/\lambda$.

Throughout this proof, we consider grids with square cells of side k and use the same strategy as in the proof of Theorem 2, with some small modifications. To make sure that the union of two independent sets, each belonging to a different contraction of a cell, is still an independent set, we obtain the contraction C^- of C by removing from C all its points within distance at most $\lambda/2$ from the boundary of C.

For i, j from 0 to $k'-1$, we create a grid with cells of side k rooted at $(\lambda i, \lambda j)$. For each cell C in the grid, we use Theorem 9 to obtain a 6-approximation $I_{i,j}(C)$ for the WIS whose input consists of all rectangles whose centers belong to C. The independent set $I_{i,j}$ is the union of all $I_{i,j}(C)$. We return the maximum-weight set $I_{i,j}$ that is found, call it I^*.

To prove that the returned solution I^* is indeed a $(4+\varepsilon)$-approximation, we use a probabilistic argument. Let i, j be picked uniformly at random from $0, \ldots, k-1$ and let OPT denote the optimal solution. For every cell C, we have

$$w(I_{i,j}(C)) \geq \frac{1}{6}\, w(OPT \cap C^-).$$

Consequently, by summing over all grid cells,

$$w(I_{i,j}) = \sum_C w(I_{i,j}(C)) \geq \frac{1}{6} \sum_C w(OPT \cap C^-) = \frac{1}{6} \sum_{R_q \in OPT} \rho(R_q)\, w(R_q),$$

where $\rho(R_q)$ denotes the probability that the center of a given rectangle R_q is contained in some contracted cell. Because such probability is the same for all rectangles, we can bound $\mathrm{E}[w(I_{i,j})]$ by writing

$$\mathrm{E}[w(I_{i,j})] \geq \frac{1}{6}\, \rho(R_q) \sum_{R_q \in OPT} w(R_q) = \frac{1}{6}\, \rho(R_q)\, w(OPT).$$

Note that, for all $R_q \in S$, $\rho(R_q)$ corresponds to the ratio between the areas of C^- and C, namely

$$\rho(R_q) = \frac{\mathrm{area}(C^-)}{\mathrm{area}(C)} = \left(\frac{k-\lambda}{k}\right)^2.$$

previous / new results	WIS		DS	
previous approximation ratio in $o(n^4)$ time	5	[20]	4.889	[11]
our approximation ratio in $O(n)$ time	$4+\varepsilon$		$4+\varepsilon$	
previous time for the same approximation	$O(n^4)$	[22]	$O(n^6 \log n)$	[18]

Table 1: Comparison of new and previous approximation algorithms for UDGs

Therefore, by using inequality (6), we obtain

$$\mathrm{E}[w(I_{i,j})] \geq \frac{1}{6}\left(\frac{k-\lambda}{k}\right)^2 w(OPT) \geq \frac{1}{6+\varepsilon}\, w(OPT).$$

Since I^* has maximum weight among the independent sets $I_{i,j}$, it follows that $w(I^*)$ is at least as large as their average weight. Therefore, I^* satisfies

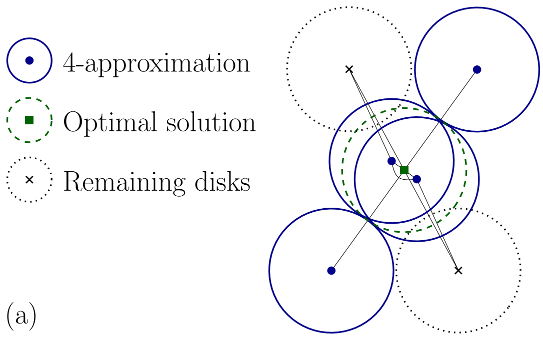
$$w(I^*) \geq \mathrm{E}[w(I_{i,j})] \geq \frac{1}{6+\varepsilon}\, w(OPT),$$

closing the proof. □

7 Conclusion

We introduced the shifting coresets method to obtain linear-time approximation algorithms for problems on geometric intersection graphs. The central idea of the method is a technique to obtain approximate solutions when the inputs are point sets of constant diameter. For the WIS and the DS on UDGs, the proposed algorithms provide improved approximation ratios when compared not only to existing linear-time algorithms, but also to sub-quartic-time algorithms, as shown in Table 1.

While the approximation ratio for the WIS and the DS on UDGs is no greater than 4 (for constant-diameter inputs), we only know that the analysis is tight for the DS. Indeed, Figure 7(a) shows a DS instance in which our algorithm does not achieve an approximation ratio better than 4, even if we reduce the grid size and search for extreme points in a larger number of directions. In contrast, for the WIS, the best lower bound we are aware of is 3.25, as shown in the following example. Let P_1 be the weighted point set from Figure 7(b), in



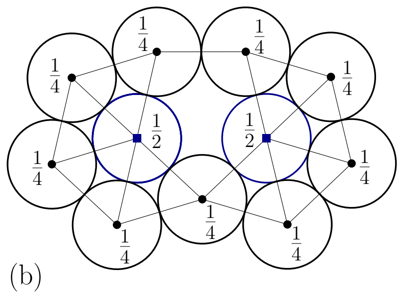


Figure 7: (a) Example in which the approximation ratio for the DS is exactly 4 (b) Coin graph used in the example in which the approximation ratio for the WIS is 3.25

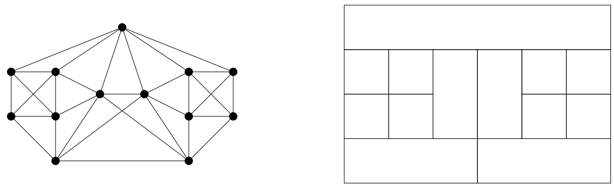


Figure 8: A 5-chromatic graph in the class from Lemma 8, and its representation by slightly-overlapping rectangles

which all adjacent vertices are at distance exactly 2. Create another set P_2 by multiplying the coordinates of the points in P_1 by $1+\varepsilon$, while multiplying their weights by $1-\varepsilon$, for arbitrarily small $\varepsilon > 0$. The set P_2 forms an independent set of weight just smaller than 3.25, while the maximum independent set in P_1 has weight 1. Since each vertex in P_2 has a smaller weight and is arbitrarily close to a vertex of P_1, the vertices of P_2 will be disregarded by the algorithm for the input instance $P_1 \cup P_2$.

The analysis of the 6-approximation ratio for the WIS on rectangle intersection graphs leaves an even bigger gap. The best lower bound we are aware of is 13/3, since the graph illustrated in Figure 8 (with 13 vertices and maximum independent set with size 3) is in the graph class used in Lemma 8. In fact, it is possible that such class is 5-colorable (the same graph in Figure 8 shows it is not 4-colorable, though). We remark that the need for c colors does not mean that the ratio between the total weight of the vertices and the maximum weight of an independent set can be as high as c.

Several open problems remain. Can we obtain an approximation ratio better than 4 in (close to) linear time for the WIS on UDGs, or at least for the unweighted version? Can the linear-time approximation scheme for the VC be generalized for the weighted version? Are the point coordinates really necessary, or is it possible to devise similar graph-based algorithms? Also, can we use our method to obtain better linear-time approximations to related problems on unit disk graphs such as finding the minimum-weight dominating set or the minimum connected dominating set? Finally, is it possible to use the shifting coresets method to obtain a constant approximation for the WIS on disk graphs (of arbitrary radii) in linear-time?

Acknowledgements

The authors would like to thank Raphael Machado, Mickael Montassier, Petru Valicov and Yann Vaxès for the insightful discussions. This research was partially supported by the Brazilian agencies CAPES, CNPq, and FAPERJ. An extended abstract of this paper appeared in the 12th Workshop on Approximation and Online Algorithms (WAOA 2014).

References

[1] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry*. Cambridge Univ. Press, 2005.

[2] P. K. Agarwal and N. H. Mustafa. Independent set of intersection graphs of convex objects in 2D. *Comput. Geom. Theory Appl.*, 34(2):83–95, 2006.

[3] P. K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Comput. Geom. Theory Appl.*, 11(3-4):209–218, 1998.

[4] K. Appel and W. Haken. Solution of the four color map problem. *Scientific American*, 237(4):108–121, 1977.

[5] J. Bentley, D. Stanat, and E. H. Williams Jr. The complexity of finding fixed-radius near neighbors. *Information Processing Letters*, 6(6):209–212, 1977.

[6] O. V. Borodin. A new proof of the 6 color theorem. *J. Graph Theory*, 19(4):507–521, 1995.

[7] T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, 2003.

[8] T. M. Chan. A note on maximum independent sets in rectangle intersection graphs. *Information Processing Letters*, 89:19–23, 2004.

[9] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42:202–208, 2003.

[10] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S.-H. Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proc. 43rd annual ACM Symp. on Theory of Computing (STOC)*, pages 273–282, 2011.

[11] G. D. da Fonseca, C. M. H. de Figueiredo, V. G. Pereira de Sá, and R. C. S. Machado. Efficient sub-5 approximations for minimum dominating sets in unit disk graphs. *WAOA 2012, Theoretical Computer Science*, 540–541:70–81, 2014.

[12] G. K. Das, M. De, S. Kolay, S. C. Nandy, and S. Sur-Kolay. Approximation algorithms for maximum independent set of a unit disk graph. *Information Processing Letters.* In press, doi:10.1016/j.ipl.2014.11.002.

[13] M. De, G. Das, P. Carmi, and S. Nandy. Approximation algorithms for a variant of discrete piercing set problem for unit disks. *International J. of Computational Geometry and Applications*, 6(23):461–477, 2013.

[14] R. Duan and S. Pettie. Linear-time approximation for maximum weight matching. *J. ACM*, 61(1):1–23, 2014.

[15] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric graphs. *SIAM J. Comput.*, 34(6):1302–1323, 2005.

[16] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.

[17] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26:238–274, 1998.

[18] R. K. Jallu, P. R. Prasad, and G. K. Das. Minimum dominating set for a point set in $\mathbb{R}^2$. *preprint*, arXiv:1312.7243, 2014.

[19] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proc. 25th annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 217–226, 2014.

[20] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(2):59–68, 1995.

[21] D. Marx. Efficient approximation schemes for geometric problems. In *Proceedings of the 13th Annual European Conference on Algorithms*, ESA'05, pages 448–459, Berlin, Heidelberg, 2005. Springer-Verlag.

[22] T. Matsui. Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs. In *JCDCG*, volume 1763 of *Lecture Notes in Computer Science*, pages 194–200, 1998.

[23] T. Nieberg, J. Hurink, and W. Kern. Approximation schemes for wireless networks. *ACM Transactions on Algorithms*, 4(4):49:1–49:17, 2008.

[24] D. E. D. Vinkemeier and S. Hougardy. A linear-time approximation algorithm for weighted matchings in graphs. *ACM Transactions on Algorithms*, 1:107–122, 2005.

[25] A. Wiese and E. Kranakis. Local ptas for independent set and vertex cover in location aware unit disk graphs. In *Distributed Computing in Sensor Systems*, pages 415–431, 2008.