

International Journal of Computational Geometry & Applications
 © World Scientific Publishing Company

Shifting coresets: obtaining linear-time approximations for unit disk graphs and other geometric intersection graphs

Guilherme D. da Fonseca

*Universit Clermont Auvergne and LIMOS
 Campus Universitaire des Cézeaux,
 1 rue de la Chebarde, Aubière, France
 fonseca@isima.fr*

Vinícius Gusmão Pereira de Sá

*Instituto de Matemática, Universidade Federal do Rio de Janeiro
 Rua Athos da Silveira Ramos s/n, Rio de Janeiro, Brazil
 vigusmao@dcc.ufrj.br*

Celina Miraglia Herrera de Figueiredo

*COPPE, Universidade Federal do Rio de Janeiro
 Av. Horácio Macedo 2030, Rio de Janeiro, Brazil
 celina@cos.ufrj.br*

ABSTRACT

Numerous approximation algorithms for problems on unit disk graphs have been proposed in the literature, exhibiting a sharp trade-off between running times and approximation ratios. We introduce a variation of the known shifting strategy that allows us to obtain linear-time constant-factor approximation algorithms for such problems. To illustrate the applicability of the proposed variation, we obtain results for three well-known optimization problems. Among such results, the proposed method yields linear-time $(4 + \epsilon)$ -approximations for the maximum-weight independent set and the minimum dominating set of unit disk graphs, thus bringing significant performance improvements when compared to previous algorithms that achieve the same approximation ratios. Finally, we use axis-aligned rectangles to illustrate that the same method may be used to derive linear-time approximations for problems on other geometric intersection graph classes.

Keywords: Approximation algorithm; unit disk graph; geometric intersection graph; core-set.

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.^{1–5} This paper employs a variation of the *shifting strategy* introduced by Hochbaum and Maass three decades ago.⁶ While the shifting strategy usually leads to high running times, our variation, which we call

the *shifting coresets method*, allows us to obtain linear-time constant-approximation algorithms for problems on unit disk graphs. Three algorithms for well-studied optimization problems on unit disk graphs are presented to illustrate the method. A fourth algorithm, for maximum-weight independent sets of axis-aligned rectangles in the plane, is also given, showing that the method applies to other geometric intersection graph classes as well.

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. 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 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.
- *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 on some vertex in C .

The shifting strategy has given rise to geometric PTASs for several problems on geometric intersection graphs.^{6–9} 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 $(1+\epsilon)$ -approximation to the original problem once the solutions to the subproblems are known. Each subproblem is then solved exactly by exploiting the fact that it has constant diameter. For example, it is often possible to 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.¹⁰

Among countless classes of geometric intersection graphs, unit disk graphs are arguably the most studied one, owing largely to their applicability in wireless networks.^{10,11} A *unit disk graph* (UDG) is the intersection graph of unit disks in the plane, and its vertices are usually 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, the fastest of which is geometric and provides a $(1 + \varepsilon)$ -approximation in $n^{O(1/\varepsilon^2)}$ time.^{9,10} 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.¹² 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;¹¹ a 44/9-approximation that uses local improvements and runs in $O(n \log n)$ time;¹² a 4-approximation that uses grids and runs in $O(n^8 \log n)$ time;¹³ and an improved 4-approximation that uses hexagonal grids and runs in $O(n^6 \log n)$ time.¹⁴

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.^{9,10,15} Using a slab decomposition and the longest path in a DAG, Matsui¹⁵ showed how to obtain an approximation ratio of $(1 + (2/\sqrt{3}) + \varepsilon) < 2.16$ in $O(n^2)$ time.^a 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.¹¹ In comparison, the algorithm presented in Section 3 obtains a $(4 + \varepsilon)$ -approximation in linear time. Note that, for the unweighted version, a geometric greedy approach that considers the vertices from left to right can be implemented to give a 3-approximation in $O(n)$ time with floor function and constant-time hashing.¹¹ The high running time of the existing PTASs also motivated the discovery of an $O(n^3)$ -time 2-approximation algorithm for the unweighted version.¹⁷

The minimum vertex cover problem (VC) for UDGs is a rare example for which a geometric linear-time approximation scheme is known.¹⁸ Previously, two high-complexity PTASs (one geometric and one graph-based) had been proposed.^{9,19}

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.^{7,20–23} However, the lack of polynomial constant-factor approximation algorithms has motivated the investigation of more restricted subclasses, for example the intersection graphs of squares and of unit-height rectangles.^{7,22} Also, PTASs for the WIS exist for the class of fat convex objects, which generalizes unit disk

^aAfter this article, the running time has been reduced to $O(n \log^3 n)$ using advanced data structures.¹⁶

4 *Guilherme D. da Fonseca, Vinícius G. Pereira de Sá, and Celina M. H. de Figueiredo*

previous / new results	WIS		DS	
previous ratio in $O(n \text{ polylog}(n))$ time	5	Ref. 11	4.889	Ref. 12
our approximation ratio in $O(n)$ time	$4 + \varepsilon$	Sec. 3	$4 + \varepsilon$	Sec. 4
previous time for a $(4 + \varepsilon)$ -approximation	$O(n^2)$	Ref. 15	$O(n^6 \log n)$	Ref. 14

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

graphs and several intersection graphs of rectangles.^{21,24} 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.²¹

Our results. We present a method to solve the constant-diameter subproblems that appear when the shifting strategy is used. We call it the *shifting coresets method* (Section 2). By incorporating this method into the shifting strategy idea, we obtain linear-time approximation algorithms for problems on unit disk graphs and other geometric intersection graphs. 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 square of constant size 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 is tackled first, in Section 3, owing to its greater simplicity) and for the DS (Section 4). The proposed algorithms provide significant improvements when compared to existing linear- and near-linear-time algorithms for these problems (see Table 1). The text also includes (Section 5) an alternative linear-time $(1 + \varepsilon)$ -approximation obtained for the VC. We reiterate that in that section we have independently derived a previous result due to Marx, for the sake of illustrating an indirect application of our method.¹⁸ Finally, our method is applied 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 an arbitrary constant $\lambda \geq 1$.

In Section 7, some open problems and lower bounds to the approximation ratios of the presented algorithms are discussed.

2. The shifting coresets method

The shifting strategy is the core of most of the existing geometric PTASs for problems on geometric intersection graphs.⁶ Generally, the shifting strategy reduces the

original problem with n objects to a set of subproblems whose inputs have constant diameter and 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, it solves each subproblem *approximately*. To do that, it employs the coresets paradigm, which consists in considering only a constant-size subset of the input.²⁵ Let $\text{diam}(P)$ denote the maximum distance between two points contained inside objects of a set of objects P . For a problem whose input is a set P of n objects and a parameter $\varepsilon > 0$, 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 for all i , $\text{diam}(P_i)$ is a constant that depends on ε .
- (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 side 0.1. Our coreset is then formed by the rectangle of maximum weight inside each 4-dimensional cell.

Assuming a real-RAM model of computation with floor function and constant-time hashing, it is possible to partition the input points into grid cells efficiently, yielding an overall $O(n)$ running time for the algorithms.²⁶ 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.^b For the VC, the running time becomes $2^{O(1/\varepsilon^3)}n$. As for the WIS on axis-aligned rectangles, if one regards ε and λ (the maximum allowed side length) as asymptotic variables, then the running time becomes $2^{O(\lambda^2/\varepsilon^2)}n$.

Definitions. A grid is said to be *rooted at a point* (x, y) if there is a grid cell with corner at (x, y) . Given a grid cell C and a real parameter $d > 0$, the square region $C_d^- \subset C$, called the *d-contraction* of C , is formed by removing from C the points within L_∞ distance at most d from the boundary of C . Analogously, the square region $C_d^+ \supset C$, called the *d-expansion* of C , is formed by C and all points within L_∞ distance at most d from C . Finally, we denote the collection of all d -expansions of cells of a grid rooted at a point p by $\mathcal{C}_d^+(p)$.

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 on unit disk graphs. 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 $w(S) = \sum_{p \in S} w(p)$. Two or more points are considered *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 $\text{diam}(P) = O(1)$, the WIS for the corresponding UDG can be 4-approximated in $O(n)$ time on the real-RAM.*

Proof. Our algorithm proceeds as follows. First, 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, create a grid with cells of diameter $\gamma = 0.29$ and side $\gamma/\sqrt{2}$ (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, build the subset $Q \subseteq P$ as follows. For each non-empty grid cell C , add to Q a point of maximum weight in $P \cap C$. Afterwards, determine the maximum-weight independent set I^* of Q . Since $|Q| = O(1)$, this can be done in constant time. 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 $4 w(I') \geq w(I)$.

^bThe dependency on ε may be conceivably improved to $2^{O(1/\varepsilon)}$ —for instance, by replacing brute-force search with a $2^{O(\sqrt{n})}$ -time exact algorithm via separators. Such a dependency is not too important, though, since the approximation factor does not approach 1. We have therefore opted for the simplest approach.

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 $\|p_1 p_2\| \leq 2$, 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 $p_1 p_2$. This implies no two edges in the drawing can cross. By the Four-Color Theorem,²⁷ S admits a partition into four independent sets S_1, \dots, S_4 . The set I' of maximum weight among S_1, \dots, S_4 must have weight at least $w(I)/4$.^c

Since I^* is the maximum-weight independent set of Q , it follows that I^* is a 4-approximation for the WIS. \square

The following theorem uses the shifting strategy to obtain a $(4 + \varepsilon)$ -approximation for point sets of arbitrary diameter. The proof borrows ideas from Hunt III *et al.*⁹ We present them in a different manner, though, 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 the real-RAM with constant-time hashing and the floor function. Without these operations, it can be done in $O(n \log n)$ time.*

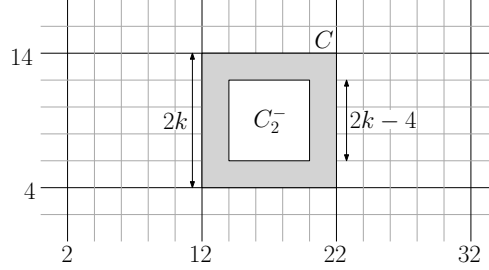
Proof. Let k be the smallest integer such that

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

The algorithm proceeds as follows. For i, j from 0 to $k-1$, create a grid with cells of side $2k$ rooted at $(2i, 2j)$. For each cell C in the grid, run the WIS 4-approximation algorithm from Theorem 1 with point set $P \cap C_2^-$ (the points in P that belong to a 2-contraction of C , see Fig.1), 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 . Return the maximum-weight set $I_{i,j}$ that is found, call it I^* .

To implement the algorithm efficiently, 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 efficiently, one should use the floor function and constant-time hashing, taking $O(n)$ time. If these operations are not available, it is possible to determine the connected components of the graph (using the Delaunay triangulation, for example) and to partition the points of each component into subcells (through

^cNote that the Four-Color Theorem is only used in the argument. No coloring is ever computed by the algorithm.


 Fig. 1. Grid rooted at $(2, 4)$ with cells of side $2k = 10$ and the 2-contraction of a cell.

sorting by x -coordinate, separating into columns, and sorting 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 of such instances, namely $(k - 2)^2$.

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

$$w(I_{i,j}(C)) \geq \frac{1}{4} w(OPT \cap C_2^-).$$

Consequently, by summing over all grid cells,

$$w(I_{i,j}) = \sum_C w(I_{i,j}(C)) \geq \frac{1}{4} \sum_C w(OPT \cap C_2^-) = \frac{1}{4} \sum_{p \in OPT} \rho(p) w(p),$$

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, $E[w(I_{i,j})]$ is bounded by

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

Note that, for all $p \in P$, $\rho(p)$ corresponds to the ratio between the areas of C_2^- and C , namely

$$\rho(p) = \frac{\text{area}(C_2^-)}{\text{area}(C)} = \left(\frac{k-2}{k} \right)^2.$$

Therefore, by using inequality (1), we obtain

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

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

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

closing the proof. \square

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) on unit disk graphs. 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. A point p dominates a point q if $\|pq\| \leq 2$. Given two sets of points D and P' , the set 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, the *minimum partial dominating set problem* (PDS). Such a generalization is necessary to properly apply the shifting strategy. In the PDS, 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 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 $\pi/2$, it follows 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 Fig. 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

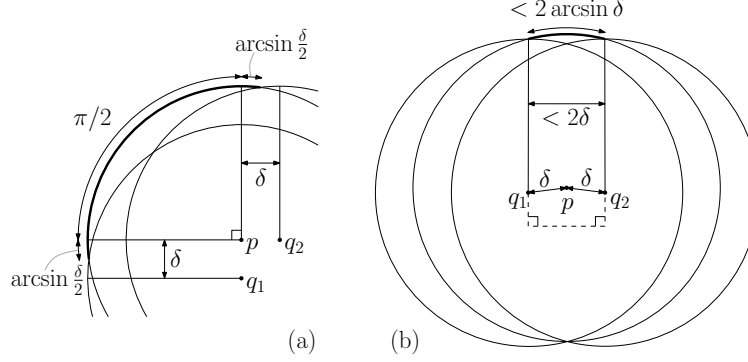


Fig. 2. Proof of Lemma 3.

shown in Fig. 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 $\text{diam}(P) = O(1)$, the PDS can be 4-approximated in $O(n)$ time on the real-RAM.*

Proof. First, determine a bounding box of constant size for P , as in the algorithm for the WIS. Within this bounding box, create a grid with cells of diameter $\gamma = 0.24$ and side $\gamma/\sqrt{2}$ (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, build the subset $Q \subseteq P$ as follows. For each non-empty grid cell, 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 Q contains at most four points per cell, it follows that $|Q| = O(1)$. Now determine the smallest P' -dominating subset $D^* \subseteq Q$. To do that, examine the subsets of Q , from smallest to largest, verifying if all points of P' are dominated, until the dominating set D^* is found and 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 , proceed as follows. For each point $p \in D$, if $p \in Q$, add p to D' . Otherwise, since the set Q contains points of extreme

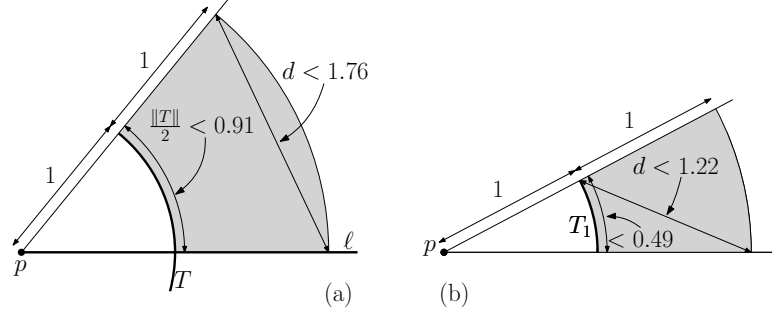


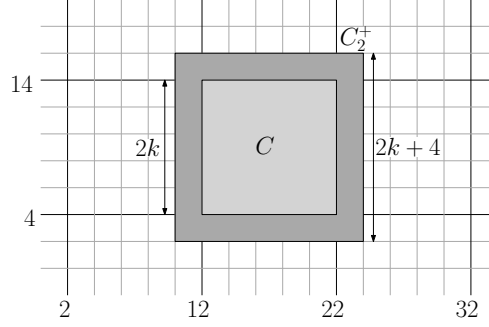
Fig. 3. Proof of Theorem 4.

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 $\angle q_1 p q_2$ is at least $\pi/2$. Add these two points q_1, q_2 to D' .

By Lemma 3, the portion $T = (\partial \odot_p) \setminus (\odot_{q_1} \cup \odot_{q_2})$ of $\partial \odot_p$ consists of one or two circular arcs. 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 $\|p_3 q_3\| \leq \gamma$. 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 $\|p_4 q_4\| \leq \gamma$. 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 Fig. 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 . By the law of cosines, the diameter of each circular crown sector is $d = \sqrt{8 - 8 \cos(\|T\|/2)} < 1.76$. 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 $\|v q_3\| \leq d + \gamma < 2$.

Finally, if T consists of two circular arcs T_1, T_2 centered in p , then start by adding those same points q_1, q_2 to D' , as if T consisted of only one arc. Then, if necessary, 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

Fig. 4. Grid rooted at $(2, 4)$ with cells of side 10 and the 2-expansion of a cell.

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 Fig. 3(b).

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

The following theorem uses the shifting strategy 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 the 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

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

The algorithm proceeds as follows. For i, j from 0 to $k-1$, create a grid with cells of side $2k$ rooted at $(2i, 2j)$ and, for each cell C in the grid, use Theorem 4 to 4-approximate the PDS with point sets $P \cap C_2^+$ (the points of P that belong to the 2-expansion of C , see Fig. 4) and $P' \cap C$, 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 . 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 again 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

$$|D_{i,j}(C)| \leq 4 |OPT \cap C_2^+|.$$

Consequently, by summing over all grid cells,

$$|D_{i,j}| = \sum_C |D_{i,j}(C)| \leq 4 \sum_C |OPT \cap C_2^+|.$$

We can now bound $E[|D_{i,j}|]$, since

$$\frac{E[|D_{i,j}|]}{4} \leq E \left[\sum_C |OPT \cap C_2^+| \right] = E \left[\sum_{p \in OPT} |C_2^+(p)| \right] = \sum_{p \in OPT} E[|C_2^+(p)|]$$

by the linearity of expectation. Note that since the expected size of $C_2^+(p)$ is the same for all points, it corresponds to the ratio between the areas of C_2^+ and C , namely

$$E[|C_2^+(p)|] = \frac{\text{area}(C_2^+)}{\text{area}(C)} = \left(\frac{k+2}{k} \right)^2.$$

Therefore, by using inequality (2), we obtain

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

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

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

closing the proof. \square

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.

In the *minimum distance d -dominating set problem* (MDDS), the input consists of 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 $d = 1$. Our algorithm for the DS can be generalized to $(4 + \varepsilon)$ -approximate the MDDS in linear time for constant d . In contrast, the greedy heuristic that gives a 5-approximation to the DS gives a $\Theta(d^2)$ -approximation to the MDDS.

5. VC for UDGs

In this section, we show how to obtain a linear-time approximation scheme to the VC on unit disk graphs. 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 from the two previous problems, the size of a minimum vertex cover for a point set of constant diameter is not 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. We remark that a different linear-time approximation scheme to the VC was presented by Marx.¹⁸

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 $\text{diam}(P) = O(1)$, the VC can be $(1 + \varepsilon)$ -approximated in $O(n)$ time on the real-RAM, for constant $\varepsilon > 0$.*

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

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

then n is constant, and the VC can be solved optimally in constant time.

Otherwise, use Theorem 1 to obtain a 4-approximation I to the maximum independent set (other constant factor approximations can be used, adjusting the threshold accordingly). We now show that $V \setminus I$ 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 $\text{diam}(P) + 2$ by the area of a unit disk,

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

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

$$n \leq \frac{4\varepsilon + 3}{3} |V_{OPT}|. \quad (3)$$

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

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

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

Using the shifting strategy the following result ensues.

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 the 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

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

The algorithm proceeds as follows. For i, j from 0 to $k-1$, create a grid with cells of side $2k$ rooted at $(2i, 2j)$ and, for each cell C in the grid, use Theorem 6 to $(1+\varepsilon/2)$ -approximate the VC for $P \cap C_2^+$, 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 . 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, once again 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

$$|V_{i,j}(C)| \leq \left(1 + \frac{\varepsilon}{2}\right) |OPT \cap C_2^+|.$$

Consequently, by summing over all grid cells,

$$|V_{i,j}| = \sum_C |V_{i,j}(C)| \leq \left(1 + \frac{\varepsilon}{2}\right) \sum_C |OPT \cap C_2^+|.$$

We can now bound $E[|V_{i,j}|]$, since

$$\frac{E[|V_{i,j}|]}{1 + \frac{\varepsilon}{2}} \leq E\left[\sum_C |OPT \cap C_2^+|\right] = E\left[\sum_{p \in OPT} |\mathcal{C}_2^+(p)|\right] = \sum_{p \in OPT} E[|\mathcal{C}_2^+(p)|]$$

by the linearity of expectation. Note that since the expected size of $\mathcal{C}_2^+(p)$ is the same for all points, it corresponds to the ratio between the areas of \mathcal{C}_2^+ and C , namely

$$E[|\mathcal{C}_2^+(p)|] = \frac{\text{area}(\mathcal{C}_2^+)}{\text{area}(C)} = \left(\frac{k+2}{k}\right)^2.$$

Therefore, by using inequality (5), we obtain

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

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

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

closing the proof. \square

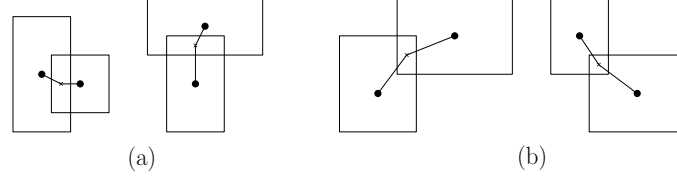


Fig. 5. Examples of (a) type-1 and (b) type-2 edges.

6. WIS for Rectangles of Bounded Size

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, \dots, R_n in the plane, such that each rectangle R_q , for $q = 1, \dots, n$, has width and height between 1 and λ , and weight $w(R_q)$. Let G be the intersection graph of S . The shifting coresets method is applied to obtain a linear-time $(6 + \varepsilon)$ -approximation algorithm to the maximum-weight independent set of G .

The *overlap* of two rectangles R_q, R_s is defined as the minimum (horizontal or vertical) translation distance of a single rectangle R_q 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) *$\text{overlap}(R_q, R_s) < 1/3$ for every two distinct rectangles $R_q, R_s \in S$,*

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 showed that 1-planar graphs are 6-colorable.²⁸ We prove the lemma by providing such a 1-planar drawing for G .

For each rectangle $R_q \in S$, draw 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 $v_q v_s$. 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 (Fig. 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

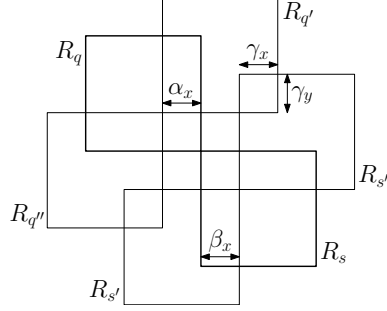


Fig. 6. Crossing twice a type-2 edge.

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 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, it follows that $\text{overlap}(R_s, R_{q'}) \geq 2/3$, 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 Fig. 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. \square

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*

18 *Guilherme D. da Fonseca, Vinícius G. Pereira de Sá, and Celina M. H. de Figueiredo*

the intersection graph of S can be 6-approximated in $O(n)$ time on the real-RAM.

Proof. Represent each rectangle $R_q \in S$ by four real values (x_q, y_q, w_q, h_q) , corresponding 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 . Create a 4-dimensional grid with cells of side $\delta = 0.1$ (any positive $\delta < 1/9$ suffices), and 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 the maximum-weight independent set among the points of S' can be computed in constant-time by brute force. 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. We claim that the rectangles in J' overlap by less than $1/3$, hence Lemma 8 can be employed to partition J' into 6 independent sets. Let I' be set of maximum weight among the partitions. Since $6w(I') \geq w(J')$, it follows that $6w(I') \geq w(I)$, proving the theorem.

To prove the claim, consider two disjoint rectangles $R_1 = (x_1, y_1, w_1, h_1), R_2 = (x_2, y_2, w_2, h_2) \in I$. Let $R'_1 = (x'_1, y'_1, w'_1, h'_1), R'_2 = (x'_2, y'_2, w'_2, h'_2)$ be the corresponding rectangles in J' . Because the grid cells have side $\delta < 1/9$, we have that all the following quantities are less than $1/9$, for $i = 1, 2$: $|x_i - x'_i|, |y_i - y'_i|, |w_i - w'_i|, |h_i - h'_i|$. The possible horizontal overlap between R'_1 and R'_2 may come from two sources: a displacement (by less than $1/9$ for each rectangle) and a change in size, which moves the boundary of each rectangle by less than $1/18$. Therefore, the maximum horizontal overlap is at most $2/9 + 2/18 = 1/3$. The same argument bounds the maximum vertical overlap. \square

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 the 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

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

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

Throughout this proof, 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 employ the contraction $C_{\lambda/2}^-$ of C .

For i, j from 0 to $k' - 1$, create a grid with cells of side k rooted at $(\lambda i, \lambda j)$. For each cell C in the grid, 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 $I_{i,j}(C)$. Return the maximum-weight set $I_{i,j}$ that is found, call it I^* .

We now prove that the returned solution I^* is indeed a $(6 + \varepsilon)$ -approximation. 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

$$w(I_{i,j}(C)) \geq \frac{1}{6} w(OPT \cap C_{\lambda/2}^-).$$

Consequently, by summing over all grid cells,

$$w(I_{i,j}) = \sum_C w(I_{i,j}(C)) \geq \frac{1}{6} \sum_C w(OPT \cap C_{\lambda/2}^-) = \frac{1}{6} \sum_{R_q \in OPT} \rho(R_q) w(R_q),$$

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, $E[w(I_{i,j})]$ can be bounded by

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

Note that, for all $R_q \in S$, $\rho(R_q)$ corresponds to the ratio between the areas of $C_{\lambda/2}^-$ and C , namely

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

Thus, inequality (6) yields

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

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

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

closing the proof. \square

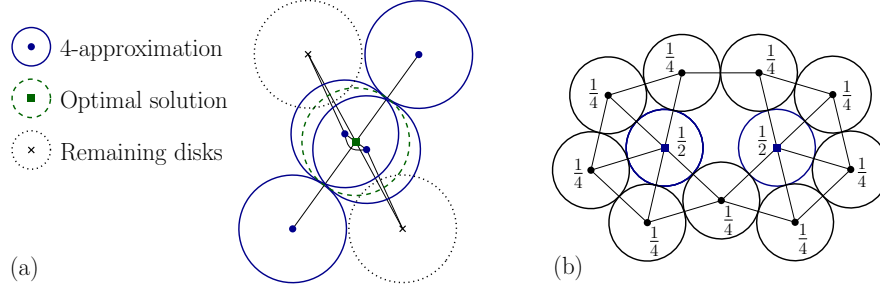


Fig. 7. (a) Example in which the approximation ratio for the DS is exactly 4. (b) Coin graph used in the example in which the approximation ratio for the WIS is 3.25.

7. Conclusion

This paper introduced the method of shifting coresets, which, combining the shifting strategy and the coresets paradigm, has allowed us to obtain improved linear-time approximations for problems on unit disk graphs. The method is applicable to other geometric intersection graph classes as well. The central idea of the method is the creation of coresets 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 to existing linear-time algorithms, as shown in Table 1 in the Introduction.

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, Fig. 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 Fig. 7(b), in 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 Fig. 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 Fig. 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 .

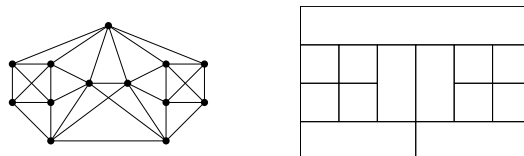


Fig. 8. A 5-chromatic graph in the class from Lemma 8, and its representation by slightly-overlapping rectangles.

Several open problems remain. Is it possible to obtain an approximation ratio better than 4 in $O(n \log n)$ time for the WIS on UDGs, or at least for the unweighted version? Can the linear-time approximation scheme for 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 our method be used 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? 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? Finally, is it possible to use similar ideas to derive improved linear-time approximations for problems on other classes of graphs such as planar graphs, bounded treewidth graphs, etc.?

Acknowledgements

The authors would like to thank Raphael Machado, Mickael Montassier, Petru Valicov and Yann Vaxès for the insightful discussions. The proofs of Theorems 2 and 7 use the same techniques as in Hunt III *et al.*, which were in turn motivated by Baker's technique.^{9,29}

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 and J. Pan, Near-linear algorithms for geometric hitting sets and set covers, in *Proc. 30th Symp. on Computational Geometry (SoCG)* (2014), pp. 271–279.
2. 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)* (2011), pp. 273–282.
3. R. Duan and S. Pettie, *J. ACM* **61** (2014) 1.
4. 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)* (2014), pp. 217–226.
5. D. E. D. Vinkemeier and S. Hougardy, *ACM Trans. on Algorithms* **1** (2005) 107.
6. D. S. Hochbaum and W. Maass, *J. ACM* **32** (1985) 130.

22 *Guilherme D. da Fonseca, Vinícius G. Pereira de Sá, and Celina M. H. de Figueiredo*

7. P. K. Agarwal, M. van Kreveld and S. Suri, *Comput. Geom. Theory Appl.* **11** (1998) 209.
8. X. Cheng, X. Huang, D. Li, W. Wu and D.-Z. Du, *Networks* **42** (2003) 202.
9. H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. Ravi, D. J. Rosenkrantz and R. E. Stearns, *J. Algorithms* **26** (1998) 238.
10. T. Nieberg, J. Hurink and W. Kern, *ACM Trans. on Algorithms* **4** (2008) 49:1.
11. M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi and D. J. Rosenkrantz, *Networks* **25** (1995) 59.
12. G. D. da Fonseca, C. M. H. de Figueiredo, V. G. Pereira de Sá and R. C. S. Machado, *Theoretical Computer Science* **540–541** (2014) 70.
13. M. De, G. Das, P. Carmi and S. Nandy, *International J. of Computational Geometry and Applications* **6** (2013) 461.
14. P. Carmi, G. K. Das, R. K. Jallu, S. C. Nandy, P. R. Prasad and Y. Stein, *International J. of Computational Geometry and Applications* **25** (2015) 227.
15. T. Matsui, Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs, in *Proc. 2nd Japan Conference on Discrete and Computational Geometry (JCDCG)* (1998), volume 1763 of *Lecture Notes in Computer Science*, pp. 194–200.
16. G. K. Das, G. D. da Fonseca and R. K. Jallu, Efficient independent set approximation in unit disk graphs, Submitted for publication.
17. G. K. Das, M. De, S. Kolay, S. C. Nandy and S. Sur-Kolay, *Information Proc. Letters* **115** (2015) 439.
18. D. Marx, Efficient approximation schemes for geometric problems, in *Proc. 13th Annual European Symp. on Algorithms (ESA)* (Springer-Verlag, Berlin, Heidelberg, 2005), pp. 448–459.
19. A. Wiese and E. Kranakis, Local PTAS for independent set and vertex cover in location aware unit disk graphs, in *Proc. 4th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)* (2008), volume 5067 of *Lecture Notes in Computer Science*, pp. 415–431.
20. P. K. Agarwal and N. H. Mustafa, *Comput. Geom. Theory Appl.* **34** (2006) 83.
21. T. M. Chan, *J. Algorithms* **46** (2003) 178.
22. T. M. Chan, *Information Proc. Letters* **89** (2004) 19.
23. A. Adamaszek and A. Wiese, Approximation schemes for maximum weight independent set of rectangles, in *Proc. 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (2013), pp. 400–409.
24. T. Erlebach, K. Jansen and E. Seidel, *SIAM J. Comput.* **34** (2005) 1302.
25. P. K. Agarwal, S. Har-Peled and K. R. Varadarajan, Geometric approximation via coresets, in *Combinatorial and Computational Geometry*, eds. J. E. Goodman, J. Pach and E. Welzl (Cambridge Univ. Press, 2005).
26. J. Bentley, D. Stanat and E. H. Williams Jr., *Information Proc. Letters* **6** (1977) 209.
27. K. Appel and W. Haken, *Scientific American* **237** (1977) 108.
28. O. V. Borodin, *J. Graph Theory* **19** (1995) 507.
29. B. S. Baker, *J. ACM* **41** (1994) 153.