

Minimum Vertex Cover in Rectangle Graphs

Reuven Bar-Yehuda¹, Danny Hermelin^{*2}, and Dror Rawitz³

¹ Department of Computer Science, Technion IIT, Haifa 32000, Israel.
`reuven@cs.technion.ac.il`

² Department of Computer Science, University of Haifa, Mount Carmel, Haifa 31905, Israel.
`danny@cri.haifa.ac.il`

³ School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel.
`rawitz@eng.tau.ac.il`

Abstract. We consider the MINIMUM VERTEX COVER problem in intersection graphs of axis-parallel rectangles on the plane. We present two algorithms: The first is an EPTAS for non-crossing rectangle families, rectangle families \mathcal{R} where $R_1 \setminus R_2$ is connected for every pair of rectangles $R_1, R_2 \in \mathcal{R}$. The second algorithm achieves a factor of $(1.5 + \varepsilon)$ in general rectangle families, for any fixed $\varepsilon > 0$, and works also for the weighted variant of the problem. Both algorithms exploit the plane properties of axis-parallel rectangles in a non-trivial way.

Keywords: Approximation algorithms, graph algorithms, vertex cover, axis-parallel rectangles, intersection graphs, arrangement graphs.

* Supported by the Adams Fellowship of the Israel Academy of Sciences and Humanities.

1 Introduction

In this paper we are concerned with the MINIMUM RECTANGLE VERTEX COVER problem: Given a set $\mathcal{R} = \{R_1, \dots, R_n\}$ of (weighted) axis-parallel rectangles in the plane, find a minimum size (weight) subset of rectangles in \mathcal{R} whose removal leaves the remaining rectangles in \mathcal{R} pairwise disjoint, *i.e.* no pair of remaining rectangles share a common point. This problem is a special case of the classical MINIMUM VERTEX COVER problem, which asks to find a minimum weight subset of vertices in a given graph, whose removal leaves the graph without edges. When the input graph is a *rectangle graph*, an intersection graph of axis parallel rectangles in the plane, and the rectangle representation of the input graph is given alongside the input, MINIMUM VERTEX COVER becomes MINIMUM RECTANGLE VERTEX COVER.

MINIMUM VERTEX COVER is one of the most extensively studied combinatorial problems in computer science, a study dating back to König’s classical early 1930s result [37], and probably even prior to that. Karp proved that the problem is NP-complete in his famous list of fundamental NP-complete problems [35], while Garey and Johnson extensively used MINIMUM VERTEX COVER as an intermediate problem in many of their early NP-completeness reductions [24]. Since then, MINIMUM VERTEX COVER played a pivoting role in the development of both approximation algorithms [31, 44], and the theory of parameterized complexity [19, 21, 42], the two main disciplines for coping with the widespread phenomena of NP-hardness.

From the perspective of approximation algorithms, MINIMUM VERTEX COVER has many polynomial-time algorithms achieving an approximation ratio of 2 [4, 15, 30, 41], the first of these given in Nemhauser and Trotter’s fundamental paper [41]. Moreover, the problem is known to be approximable within $2 - \frac{\lg \lg n}{2 \lg n}$ [5, 40], within $2 - \frac{\ln \ln n}{\ln n}(1 - o(1))$ [29] and even within $2 - \Theta(\frac{1}{\sqrt{\log n}})$ [34]. On the other hand, it is also known that MINIMUM VERTEX COVER is inapproximable within a factor of $10\sqrt{5} - 21 \approx 1.36$, unless $P=NP$ [17]. There are however many natural special-case graph classes for which one can improve on this barrier. For instance, in the class of interval graphs, which can be thought of as one dimensional analogs of rectangle graphs, MINIMUM VERTEX COVER is polynomial-time solvable [25]. In planar graphs, the problem is known to admit a polynomial-time approximation scheme (PTAS) [39], and even an efficient PTAS (EPTAS) due to Baker’s seminal framework for NP-hard planar graph problems [3].

The dual problem of MINIMUM RECTANGLE VERTEX COVER is the MAXIMUM RECTANGLE INDEPENDENT SET problem: Given a family of axis-parallel rectangles in the plane, find a maximum size (or weight) subset of pairwise disjoint rectangles. This problem has been extensively studied in the computational geometry community, and has several applications in data mining [10, 36], automated label placement [1, 18, 23], and in network resource allocation with advance reservation for line topologies [10, 38], which also apply to MINIMUM VERTEX COVER in rectangle graphs (see Section ??). Fowler *et al.* [22] showed that MAXIMUM INDEPENDENT SET in rectangle graphs is NP-complete, implying the NP-completeness of MINIMUM VERTEX COVER in rectangle graphs. Chlebík and Chlebíková [14] showed that MAXIMUM INDEPENDENT SET and MINIMUM VERTEX COVER remain NP-hard even in intersection graphs of unit squares. There have been several $O(\lg n)$ approximation algorithms independently suggested for this problem [1, 7, 12, 36]. Lewin-Eitan *et al.* [38] devised a $4q$ -approximation algorithm for the problem, where q is the size of the maximum clique in the input graph. Recently, Chalermsook and Chuzhoy [10] were able to break the $\lg n$ barrier of approximation by devising a $O(\lg \lg n)$ randomized approximation algorithm. A simpler $O(\lg n / \lg \lg n)$ -approximation algorithm was given in [13]. There are also many special cases

in which MAXIMUM RECTANGLE INDEPENDENT SET admits a polynomial-approximation scheme (PTAS) [1, 7, 11, 20].

In contrast to the vast amount of research devoted for MAXIMUM RECTANGLE INDEPENDENT SET there has been surprisingly very little focus on the MINIMUM RECTANGLE VERTEX COVER problem. Nevertheless, some of the results for MAXIMUM RECTANGLE INDEPENDENT SET carry through to MINIMUM RECTANGLE VERTEX COVER. For instance, the result of Fowler *et al.* [22] implies that MINIMUM RECTANGLE VERTEX COVER is NP-hard. Also, by applying the Nemhauser and Trotter Theorem (see Section 2) as a preprocessing step, any PTAS for MAXIMUM RECTANGLE INDEPENDENT SET can be converted into a PTAS for MINIMUM RECTANGLE VERTEX COVER. Thus, the results in [1] imply that MINIMUM RECTANGLE VERTEX COVER has a PTAS when all rectangles have equal height, while [11] gives a PTAS when all rectangles are squares. Erlebach *et al.* [20] gave an explicit PTAS for MINIMUM RECTANGLE VERTEX COVER in bounded aspect-ratio rectangle families without using the Nemhauser and Trotter procedure. Finally, we mention the work by Chan and Har-Peled [13] who devised a PTAS for MAXIMUM INDEPENDENT SET in families of pseudo-discs, which implies a PTAS for MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families.

1.1 Related Work

MINIMUM VERTEX COVER and its dual counterpart MAXIMUM INDEPENDENT SET have been previously studied in many geometric intersection graphs other than rectangle graphs. Gavril [25] gave a polynomial-time algorithm for both of these problem in chordal graphs, intersection graphs of subgraphs of a tree. Apostolico *et al.* [2] gave a polynomial-time algorithm for these two problems in intersection graphs of chords on a circle, which were later improved by Cenek and Stuart [9], while Golubic and Hammer [27] gave a polynomial-time algorithm for intersection graphs of arcs on a circle which was later improved in [33]. A good survey of many generalizations of these results can be found in [26, 28]. Hochbaum and Maass considered intersection graphs of d -dimensional cubes in \mathbb{R}^d [32], while Erlebach *et al.* [20] considered intersection graphs of general fat objects in the plane. In [6, 8], approximation algorithms were suggested for MAXIMUM INDEPENDENT SET and MINIMUM VERTEX COVER in the class of multiple-interval graphs.

1.2 Applications and Motivation

Automated label placement is a central problem in geographic information systems which has been extensively studied in various settings [1, 18, 23]. The basic problem is to place labels around points in a geographic maps, where the labels are often assumed to be rectangles [1] which are allowed to be positioned at specific places adjacent to their corresponding points in the map. The usual criterion for a legal placement is that all rectangles are pairwise disjoint. Subject to this constraint, a natural optimization criteria is to minimize the number of labels to be removed so as the remaining labels form a legal placement. This is exactly the MINIMUM RECTANGLE VERTEX COVER problem.

MINIMUM RECTANGLE VERTEX COVER can also be used to model shared-resource scheduling scenarios where the requests are given in advance to the system. Consider the typical critical-section scheduling problem occurring in all modern operating system: A set of programs request access to a shared resource in memory for read\write purposes. The goal of the operating system is to serve as many requests as possible, so long as no two programs access the same memory entries simultaneously, to avoid obvious data-consistency hazards. In a simplified variant of this problem,

one can assume that all programs have a single request to fixed array of registers in memory, and this request occurs during a fixed interval of their running time. If these requests are known beforehand to the operating system, the problem of minimizing the number of programs not to be served can naturally be modeled as MINIMUM RECTANGLE VERTEX COVER, by using the x -axis to measure the shared memory array, and the y -axis to measure program execution-time.

1.3 Results and Techniques

In this paper we present two approximation algorithms for the MINIMUM RECTANGLE VERTEX COVER problem. For a pair of rectangles R_1 and R_2 in our input set of rectangles \mathcal{R} , we say that R_1 and R_2 *cross* if they intersect but neither rectangle contains a corner of the other rectangle. This is equivalent to requiring that $R_1 \setminus R_2$ is connected. We say that \mathcal{R} is non-crossing if there is no pair of crossing rectangles in \mathcal{R} . Our first algorithm is an EPTAS for MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families:

Theorem 1. *Given any $\varepsilon > 0$, MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families can be approximated within $(1 + \varepsilon)$ in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time.*

(We mention that MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families is NP-hard according to [14]).

Theorem 1 generalizes the PTAS result of Agarwal *et al.* [1] and Chan [11] for squares and equal height rectangles, and it also handles several families of rectangles which cannot be handled by the PTAS of Erlebach *et al.* [20]. In terms of time complexity, our algorithm dramatically improves on all these algorithms, and also on the algorithm of Chan and Har-Peled [13], since all these algorithms have running-times of the form $n^{\text{poly}(1/\varepsilon)}$.

The novelty behind the algorithm in Theorem 1 lies in its usage of the *arrangement graph* \square of the input set of rectangles \mathcal{R} . This graph is defined by considering all intersection points occurring on boundary of rectangles as vertices, and the boundary curves connecting them as edges. By its definition, the arrangement graph of a rectangle family is planar and 4-regular, and thus has very convenient structure. However, there is no immediate way to translate approximate vertex-covers in the arrangement graph $A_{\mathcal{R}}$ of \mathcal{R} , to vertex covers in the corresponding rectangle graph $G_{\mathcal{R}}$. Nevertheless, we show that we can translate tree-decompositions in $A_{\mathcal{R}}$ to tree-decompositions in $G_{\mathcal{R}}$ of roughly the same width, and this allows with some technical effort to simulate Baker's algorithm [3]. We believe that the arrangement graph can be a useful tool in other intersection-graph problems.

The second algorithm we present in this paper applies for general rectangle families, and can handle also weights. This algorithm exploits the observation that the rectangles of a triangle-free rectangle graph can be partitioned into two classes, where no pair of rectangles cross in each class. This, in combination with Theorem 1 and the fact that we can clean all triangles from our input graph at cost of a 1.5 factor to the approximation guarantee, gives us Theorem 2 below for the unweighted case. For the weighted case, we use the additional observation that triangle-free non-crossing rectangle graphs are planar, and so we can use Baker's algorithm [3] directly.

Theorem 2. *Given any $\varepsilon > 0$, MINIMUM RECTANGLE VERTEX COVER can be approximated within a factor of $1.5 + \varepsilon$ in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time.*

2 Preliminaries

We denote our input set of axis-parallel rectangles in the plane by $\mathcal{R} = \{R_1, \dots, R_n\}$. We assume that each rectangle R is specified by two intervals $R := (X, Y)$, where X is the projection of R on the x -axis, and Y is the projection of R on the y -axis. We assume w.l.o.g. that \mathcal{R} is in *general position*, i.e. that all intervals in the specification of \mathcal{R} have different endpoints. The *boundary* of a rectangle R is the set of all points with minimum and maximum x -coordinate values, and minimum and maximum y -coordinate values.

Two rectangles $R_1 = (X_1, Y_1)$ and $R_2 = (X_2, Y_2)$ *intersect*, denoted $R_1 \cap R_2 \neq \emptyset$, if they share a common point, i.e. if $X_1 \cap X_2 \neq \emptyset$ and $Y_1 \cap Y_2 \neq \emptyset$. Two non-intersecting rectangles are said to be *disjoint*. There are three possible types of intersections between two rectangles R_1 and R_2 :

1. *Containment intersection*: R_1 contains R_2 . In this case the boundaries of R_1 and R_2 do not intersect.
2. *Corner intersection*: R_1 contains one or two corners of R_2 . In this case the boundaries of R_1 and R_2 intersect exactly twice.
3. *Crossing intersection*: the intersection of R_1 and R_2 does not involve any corners. In this case, the boundaries of R_1 and R_2 intersect four times.

See examples in Figure 1.

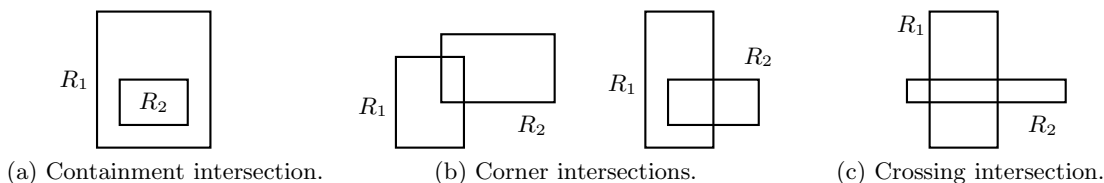


Fig. 1: Possible intersections between two rectangles R_1 and R_2 .

Given a graph G , we use $V(G)$ and $E(G)$ to denote its vertex and edge set, respectively. For a given vertex-subset $V \subseteq V(G)$, we let $G[V]$ denote the subgraph of G induced by V , i.e. the subgraph with vertex-set V and edge-set $\{\{u, v\} \in E(G) : u, v \in V\}$. We write $G - V$ to denote the induced subgraph $G[V(G) \setminus V]$. We will also be considering vertex-weighted graphs, i.e. graphs G equipped with a weight-function $w : V(G) \rightarrow \mathbb{Q}$. A *vertex cover* of G is a subset of vertices $C \subseteq V(G)$ such that $\{u, v\} \cap C \neq \emptyset$ for any edge $\{u, v\} \in E(G)$. For a non-negative real $\alpha \in \mathbb{R}^{\geq 0}$, an α -*approximate vertex cover* of G is a vertex cover C with $|C| \leq \alpha \cdot \text{OPT}$ (or $w(C) \leq \alpha \cdot \text{OPT}$ in the weighted case), where OPT is the weight of a minimum cardinality (weight) vertex cover of G . The *intersection graph* $G_{\mathcal{R}}$ corresponding to our input set of rectangles \mathcal{R} is the graph with vertex-set $V(G_{\mathcal{R}}) := \mathcal{R}$, and edge-set $E(G_{\mathcal{R}}) := \{\{R_1, R_2\} : R_1 \cap R_2 \neq \emptyset\}$.

We will be using an important tool due to Nemhauser and Trotter [41] that allows us to focus on graphs whose entire vertex-set already constitutes a good approximate vertex-cover:

Theorem 3 (Nemhauser&Trotter [41]). *There is a polynomial-time algorithm that on given a (vertex-weighted) graph G , computes an induced subgraph G' of G such that:*

- (i) $V(G')$ is a 2-approximate vertex-cover of G , and
- (ii) any α -approximate vertex cover of G' can be converted in polynomial-time to an α -approximate vertex cover of G .

Finally, we will be using the notion of treewidth and tree-decomposition of graphs, introduced in the form below by Robertson and Seymour [43].

Definition 1 (Tree Decomposition, Treewidth [43]). A tree decomposition of a graph G is a pair $(\mathcal{T}, \mathcal{X})$, where $\mathcal{X} \subseteq 2^{V(G)}$ is family of vertex subsets of G , and \mathcal{T} is a tree over \mathcal{X} , satisfying the following conditions:

1. $\bigcup_{X \in \mathcal{X}} G[X] = G$, and
2. $\mathcal{X}_v = \{X \in \mathcal{X} : v \in X\}$ is connected in \mathcal{T} for all $v \in V(G)$.

The width of \mathcal{T} is $\max_{X \in \mathcal{X}} |X| - 1$. The treewidth of G , denoted $tw(G)$, is the minimum width over all tree decompositions of G .

3 An EPTAS for Non-Crossing Rectangle Graphs

In this section we present an EPTAS for MINIMUM RECTANGLE VERTEX COVER in unweighted non-crossing rectangle families. Let \mathcal{R} denote our input set of unweighted non-crossing rectangles. The first step of our algorithm is to clean \mathcal{R} from containment intersections and pairwise intersecting subsets of size greater than some $q \geq 2$ to be chosen later. This can be done using standard techniques, and allows us to gain substantial structure at a small cost to the approximation factor of our algorithm.

Lemma 1. *Suppose that MINIMUM RECTANGLE VERTEX COVER in corner-intersecting rectangle families with no $q+1$ pairwise intersecting rectangles can be approximated within a factor of α . Then MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families can be approximated within a factor of $\max\{\alpha, 1 + 1/q\}$.*

Proof. Let \mathcal{R} be any non-crossing rectangle family. First observe that if \mathcal{R} has two rectangles R_1 and R_2 such that R_1 is contained in R_2 , then (in the unweighted case) there is an optimal vertex cover of $G_{\mathcal{R}}$ that includes R_2 . Indeed, if \mathcal{C} is a vertex-cover of $G_{\mathcal{R}}$, then $\mathcal{C} \cap \{R_1, R_2\} \neq \emptyset$, and $\mathcal{C}' := (\mathcal{C} \setminus \{R_1\}) \cup \{R_2\}$ is also a vertex-cover of $G_{\mathcal{R}}$ with $|\mathcal{C}'| \leq |\mathcal{C}|$. Second, note that if $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_r$ form disjoint cliques of size $q+1$ in $G_{\mathcal{R}}$, then any vertex-cover of $G_{\mathcal{R}}$ must include at least $r \cdot q$ rectangles from $\bigcup_{1 \leq i \leq r} \mathcal{R}_i$. Thus, by including all rectangles from these cliques in our solution, we deviate (again, in the unweighted case) by at most a factor of $1 + 1/q$ from the optimum.

Using these two observations, we proceed as follows: We compute two sets of rectangles $\mathcal{P}, \mathcal{Q} \subseteq \mathcal{R}$. The set \mathcal{P} includes all rectangles of \mathcal{R} that contain other rectangles of \mathcal{R} , but are not contained in any of rectangle of \mathcal{R} , and \mathcal{Q} is a greedy packing of disjoint $(q+1)$ -cliques in $G_{\mathcal{R} \setminus (\mathcal{P} \cup \mathcal{P}')}$, where \mathcal{P}' denotes all rectangles contained in rectangles of \mathcal{P} . Observe that both these sets can be computed in $O(n^2)$ time, \mathcal{P} by obvious brute force, and \mathcal{Q} by the fact that every clique is represented by some rectangle corner as $\mathcal{R} \setminus \mathcal{P}$ has only corner intersections. We then apply the α -approximation algorithm assumed in the lemma to obtain a α -approximate vertex-cover \mathcal{C} for $\mathcal{R} \setminus (\mathcal{P} \cup \mathcal{P}' \cup \mathcal{Q})$. According to the two observations above, $\mathcal{P} \cup \mathcal{Q} \cup \mathcal{C}$ is a $\max\{\alpha, 1 + 1/q\}$ -approximate vertex-cover of $G_{\mathcal{R}}$. \square

Due to Lemma 1 above, we can henceforth assume that \mathcal{R} contains only corner intersections, and that the maximum clique in $G_{\mathcal{R}}$ is of size at most q . We also apply the Nemhauser&Trotter algorithm (Theorem 3) on $G_{\mathcal{R}}$ after applying Lemma 1, and so we assume that \mathcal{R} is a 2-approximate vertex cover of $G_{\mathcal{R}}$.

The main idea of algorithm is as follows. We will construct the so-called *arrangement graph* $A_{\mathcal{R}}$ of \mathcal{R} which is build by considering all intersection points occurring on boundaries of rectangles as vertices, and the boundary curves connecting them as edges. By this construction, $A_{\mathcal{R}}$ is a planar graph, and as such, it has very specific structure. The most tempting approach is to use Baker's EPTAS for MINIMUM VERTEX COVER in planar graphs on $A_{\mathcal{R}}$, and to convert the $(1 + \varepsilon)$ -approximate vertex-cover of $A_{\mathcal{R}}$ to a $(1 + \varepsilon')$ -approximate vertex-cover of $G_{\mathcal{R}}$. Unfortunately, this attempt fails, since the natural transformation from vertices of $A_{\mathcal{R}}$ to rectangles of $G_{\mathcal{R}}$ produces the entire set of rectangles \mathcal{R} on any vertex-cover of $A_{\mathcal{R}}$. We therefore take an alternative route. The basic idea is to mimic Baker's algorithm by using the observation that tree-decompositions of $A_{\mathcal{R}}$ correspond to tree-decompositions of $G_{\mathcal{R}}$ of roughly the same width. Thus, instead of applying Baker's algorithm on $A_{\mathcal{R}}$ as a black-box, we can simulate it steps directly on $G_{\mathcal{R}}$. Using then an extension of Baker's analysis, we can show that this approach indeed gives us our the desired $(1 + \varepsilon)$ approximation factor.

3.1 The arrangement graph

An intersection of two rectangle boundaries is called a *joint*. The *arrangement graph* $A_{\mathcal{R}}$ of \mathcal{R} is the planar graph that is defined as follows: The vertex set of $A_{\mathcal{R}}$ is the set of joints. The edge set of $A_{\mathcal{R}}$ consists of the rectangle boundary fragments, namely $\{u, v\}$ is an edge in $A_{\mathcal{R}}$ iff u and v are two joints located on the boundary of some rectangle such that no other joint is located on the boundary between them. It is not difficult to see that the arrangement graph defined as above is in fact planar (see example in Figure 2).

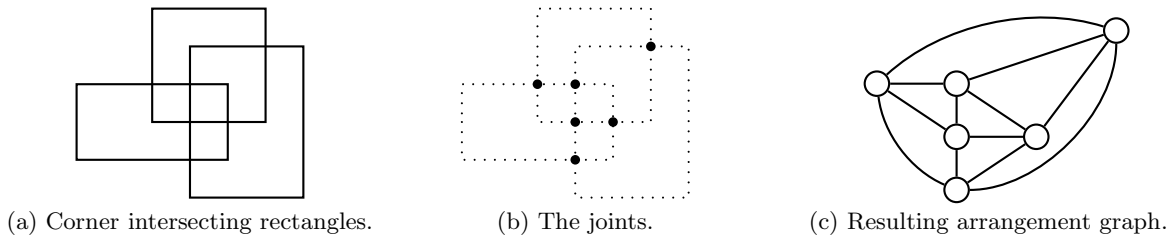


Fig. 2: A set \mathcal{R} of corner intersection rectangles and its arrangement graph $A_{\mathcal{R}}$.

For a given subset of joints $J \subseteq V(A_{\mathcal{R}})$, the set of rectangles that is induced by J is defined by $\mathcal{R}(J) := \{R \in \mathcal{R} : \exists j \in J \text{ s.t. } j \text{ is on the boundary of } R\}$. The following lemma is immediate from the fact that \mathcal{R} is in general position.

Lemma 2. $|\mathcal{R}(J)| = 2 \cdot |J|$ for any set of joints $J \subseteq V(A_{\mathcal{R}})$.

Lemma 3. $|V(A_{\mathcal{R}})| \leq 4q \cdot |\mathcal{R}|$.

Proof. Recall that we assume \mathcal{R} to have only corner intersections and no $q+1$ pairwise intersecting rectangles. Now, if two rectangles R_1 and R_2 are corner intersecting, then the boundaries of R and R' intersect exactly twice. Hence, $|V(A_{\mathcal{R}})| = 2|E(G_{\mathcal{R}})|$. Furthermore, since each corner can be involved in at most q intersections, we have $\sum_{R \in \mathcal{R}} \deg(R) \leq 4q$, where $\deg(R)$ denotes the number of rectangles intersecting R . Thus, $|E(G_{\mathcal{R}})| \leq 2q \cdot |\mathcal{R}|$, and so $|V(A_{\mathcal{R}})| \leq 2 \cdot |E(G_{\mathcal{R}})| \leq 4q \cdot |\mathcal{R}|$. \square

3.2 Baker's Algorithm

Our algorithm for MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families simulates Baker's classical algorithm for MINIMUM VERTEX COVER in planar graphs [3] on the arrangement graph $A_{\mathcal{R}}$ of \mathcal{R} . The main idea behind Baker's approach is the observation that given a planar graph G and any positive integer k , one can partition the vertex-set of G into k classes such that deleting each the vertices in class results in subgraph of treewidth at most $3k$ (see Lemma 4 below). Combining this observation along with the well-known algorithm for MINIMUM VERTEX COVER in bounded treewidth graphs (see *e.g.* [42]), gives an EPTAS for MINIMUM VERTEX COVER in planar graphs.

Lemma 4 (Baker [3]). *Given any planar graph G and any integer k , one can partition $V(G)$ in polynomial-time w.r.t. both $|V(G)|$ and k , into k subsets V_1, \dots, V_k such that $tw(G - V_i) \leq 3k$.*

In order to properly simulate Baker's approach on MINIMUM RECTANGLE VERTEX COVER we will need a slightly more general framework. In particular, our algorithm will not necessarily produce a partition of the vertex-set of $G_{\mathcal{R}}$. Also, our algorithm will produce vertex-sets whose deletion results in a subgraph of $G_{\mathcal{R}}$ with treewidth slightly more than $3k$. Nevertheless, it is not difficult to show that a slight relaxation of these two requirements does not alter Baker's analysis too much:

Lemma 5. *Let G be a graph with n vertices, and let c_1 and c_2 be two fixed positive integers. Suppose that there is a polynomial-time algorithm that, given G and a positive integer k , produces vertex-sets V_1, \dots, V_k with the following properties:*

1. $\bigcup_i V_i = V(G)$.
2. $\sum_i |V_i| \leq c_1 \cdot n$.
3. $\forall i : tw(G - V_i) \leq c_2 \cdot k$.

Then one can compute an vertex-cover of G within a factor of $(1 + \varepsilon)$ in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time, for any given $\varepsilon > 0$.

Proof. Choose k to be an integer greater or equal to $2c_1/\varepsilon$, and let V_1, \dots, V_k be the vertex-sets produced by the algorithm assumed in the lemma. We assume G has been preprocessed using the Nemhauser&Trotter algorithm (Theorem 3), and thus $n \leq 2\text{OPT}$, where OPT is the size of the minimum vertex-cover of G . For each i , $1 \leq i \leq k$, let H_i denote the subgraph $H_i := G_i - V_i$, and let OPT_i be the size of the minimum vertex-cover of H_i . Since $tw(H_i) \leq c_2 \cdot k$, a vertex-cover C_i^* for H_i of size OPT_i can be computed in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time. We thus obtain k candidate vertex-covers for G , $C_i := V_i \cup C_i^*$, and we have

$$\sum_i |C_i| = \sum_i (|V_i| + \text{OPT}_i) \leq c_1 n + \sum_i \text{OPT}_i \leq 2c_1 \text{OPT} + k \text{OPT} = (k + 2c_1) \text{OPT}.$$

Therefore, choosing the smallest among the C_i 's, we get a vertex-cover for G of size at most $\min_i |C_i| \leq (1 + 2c_1/k) \text{OPT} \leq (1 + \varepsilon) \text{OPT}$. \square

3.3 Our Algorithm

We are now in position to describe our EPTAS. The key lemma we will need is Lemma 6 below that allows us to convert tree-decompositions of $A_{\mathcal{R}}$ to a tree-decompositions of $G_{\mathcal{R}}$ of approximately the same width.

Lemma 6. *If $(\mathcal{T}, \mathcal{X})$ is a tree-decomposition of $A_{\mathcal{R}}$, then $(\mathcal{T}', \mathcal{X}')$ is a tree-decomposition of $G_{\mathcal{R}}$, where $\mathcal{X}' := \{\mathcal{R}(X) : X \in \mathcal{X}'\}$, and \mathcal{T}' is a tree over \mathcal{X}' with an edge $\{\mathcal{R}(X_1), \mathcal{R}(X_2)\}$ for every edge $\{X_1, X_2\}$ in \mathcal{T} .*

Proof. We show that $(\mathcal{T}', \mathcal{X}')$ satisfies all requirements of Definition 1. First, observe that any rectangle has at least two corresponding joints since we assume there are no isolated rectangles in \mathcal{R} . Furthermore, if two rectangles intersect, then there is a joint $j \in V(A_{\mathcal{R}})$ that corresponds to both these rectangles. Hence, for every edge $\{R_1, R_2\} \in E(G_{\mathcal{R}})$, there is at least one node in \mathcal{X}' which contains both R_1 and R_2 . Thus,

$$\bigcup_{X \in \mathcal{X}'} G_{\mathcal{R}}[X] = G_{\mathcal{R}}.$$

Now suppose there is some rectangle R which is contained in two nodes $\mathcal{R}(X_1)$ and $\mathcal{R}(X_2)$ of \mathcal{T}' . Then R has two joints j_1 and j_2 with $j_1 \in X_1$ and $j_2 \in X_2$. By construction, there is a path $j_1, i_1, \dots, i_r, j_2$ connecting j_1 to j_2 in $A_{\mathcal{R}}$, where i_1, \dots, i_r are all joints of R . Since $(\mathcal{T}, \mathcal{X})$ is a proper tree decomposition of $A_{\mathcal{R}}$, it follows that there is a path $X_1, Y_1, \dots, Y_{r'}, X_2$ connecting X_1 and X_2 in \mathcal{T} , with $Y_i \cap \{j_1, i_1, \dots, i_r, j_2\} \neq \emptyset$ for each i , $1 \leq i \leq s$. Thus, each node in the path $\mathcal{R}(X_1), \mathcal{R}(Y_1), \dots, \mathcal{R}(Y_s), \mathcal{R}(X_2)$ connecting $\mathcal{R}(X_1)$ and $\mathcal{R}(X_2)$ in \mathcal{T}' contains R , and since $R, \mathcal{R}(X_1)$, and $\mathcal{R}(X_2)$ were chosen arbitrarily, this shows that for each $R \in \mathcal{R}$:

$$\{\mathcal{R}(X) \in \mathcal{X}' : R \in X\} \text{ is connected in } \mathcal{T}'.$$

Thus, both requirements of Definition 1 are fulfilled by $(\mathcal{T}', \mathcal{X}')$, and so the lemma is proved. \square

Observe that due to Lemma 2, the tree decomposition $(\mathcal{T}', \mathcal{X}')$ of $G_{\mathcal{R}}$ obtained by in the above lemma has at most twice the width of the given tree-decomposition $(\mathcal{T}, \mathcal{X})$ of $A_{\mathcal{R}}$. Our algorithm consists of the following four steps:

1. Set q and k
2. Apply Lemma 1 so that \mathcal{R} does not have any containment intersections, and no pairwise intersecting subsets of size greater than q .
3. Apply the Nemhauser&Trotter algorithm on $G_{\mathcal{R}}$, and let $\mathcal{R}' \subseteq \mathcal{R}$ denote the resulting subset of rectangles.
4. Construct the arrangement graph $S_{\mathcal{R}'}$ corresponding to \mathcal{R}' , and partition $S_{\mathcal{R}'}$ into k subsets V_1, \dots, V_k using Lemma 4.
5. Use Lemma 5 on $G_{\mathcal{R}}$ with $\mathcal{R}(V_1), \dots, \mathcal{R}(V_k)$.

According to Lemmas 2, 3, and 6 above, $\mathcal{R}(V_1), \dots, \mathcal{R}(V_k)$ indeed satisfies all the three conditions of Lemma 5, thus the above algorithm indeed outputs a $(1 + \varepsilon)$ -approximate vertex-cover of $G_{\mathcal{R}}$ in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time.

4 General Rectangle Graphs

In this section we present an algorithm for MINIMUM RECTANGLE VERTEX COVER in general rectangle families. Our algorithm achieves an approximation factor of $1.5 + \varepsilon$, for any given $\varepsilon > 0$, in time $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$, and works also for the weighted variant of the problem. We begin with the unweighted case, and with the following lemma which relies on an observation already made by Lewin-Eytan *et al.* [38]. A rectangle family is said to be *triangle-free* if there are no three pairwise intersecting rectangles in the family.

Lemma 7. *Any triangle-free rectangle family can be partitioned into two non-crossing subsets in polynomial time.*

Proof. Let \mathcal{R} be a triangle-free rectangle family. Observe that any two rectangles $R_1 = (X_1, Y_1)$ and $R_2 = (X_2, Y_2)$ are crossing if and only if either

- $X_1 \subset X_2$ and $Y_2 \subset Y_1$, or
- $X_2 \subset X_1$ and $Y_1 \subset Y_2$,

It follows that the crossing relation between rectangles forms a partial order. Thus, by Dilworth's Theorem [16], and due to the fact \mathcal{R} is triangle-free, there exists a partitioning Π of \mathcal{R} into two non-crossing subsets as desired. Furthermore, using one of many classical minimum anti-chain partitioning algorithms (see *e.g.* [26]), one can compute Π in polynomial-time. \square

Observe that Lemma 7 is already enough, along with the results in Section 3, to obtain our desired $1.5 + \varepsilon$ approximation factor. The algorithm proceeds in the following six steps, given a rectangle family \mathcal{R} and $\varepsilon > 0$:

1. Apply Lemma 1 to obtain a triangle-free rectangle family $\mathcal{R}' \subseteq \mathcal{R}$.
2. Apply the Nemhauser&Trotter algorithm to obtain a subset $\mathcal{R}'' \subseteq \mathcal{R}'$ which is a 2-approximate vertex-cover of $G_{\mathcal{R}''}$.
3. Use Lemma 7 to obtain a partitioning $\{\mathcal{R}_1, \mathcal{R}_2\}$ of \mathcal{R}'' , where both \mathcal{R}_1 and \mathcal{R}_2 are non-crossing.
4. Compute an ε -approximate vertex-cover \mathcal{C}_1 of $G_{\mathcal{R}_1}$, and an ε' -approximate vertex-cover \mathcal{C}_2 of $G_{\mathcal{R}_2}$, using $\varepsilon' := \varepsilon/2$ and the EPTAS of Section 3.
5. Use the best of the two vertex-covers $\mathcal{R}_1 \cup \mathcal{C}_2$ and $\mathcal{R}_2 \cup \mathcal{C}_1$ for $G_{\mathcal{R}''}$, along with the Nemhauser&Trotter Theorem to compute a vertex-cover of $G_{\mathcal{R}'}$.
6. Add removed rectangles as required by Lemma 1 to obtain a vertex-cover for $G_{\mathcal{R}}$.

The fact that this algorithm outputs a vertex-cover which is a factor of $1.5 + \varepsilon$ off the optimum follows from a similar analysis used in Lemma 5. Clearly, both $\mathcal{R}_1 \cup \mathcal{C}_2$ and $\mathcal{R}_2 \cup \mathcal{C}_1$ are vertex covers for $G_{\mathcal{R}''}$. Furthermore, letting OPT , OPT_1 , and OPT_2 denote the size of the minimum vertex-covers of $G_{\mathcal{R}''}$, $G_{\mathcal{R}_1}$, and $G_{\mathcal{R}_2}$ respectively, we get:

$$\begin{aligned} |\mathcal{R}_1 \cup \mathcal{C}_2| + |\mathcal{R}_2 \cup \mathcal{C}_1| &\leq |\mathcal{R}| + (1 + \varepsilon)\text{OPT}_1 + (1 + \varepsilon)\text{OPT}_2 \\ &\leq 2\text{OPT} + (1 + \varepsilon)\text{OPT} \\ &= (3 + \varepsilon)\text{OPT} . \end{aligned}$$

Thus, the minimum of both $\mathcal{R}_1 \cup \mathcal{C}_2$ and $\mathcal{R}_2 \cup \mathcal{C}_1$ gives a $(1.5 + \varepsilon)$ -approximate vertex-cover for $G_{\mathcal{R}''}$. Applying the Nemhauser&Trotter Theorem along with Lemma 1 shows that the algorithm above outputs a $(1.5 + \varepsilon)$ -approximate vertex-cover for $G_{\mathcal{R}}$.

For the weighted variant of the problem, we observe that all steps of the algorithm above, apart from Step 4, can be applied also in the weighted case. For the first step we use a weighted version of Lemma 1, which can be obtained by a standard application of the local-ratio technique [5] (see e.g. [8]). All other steps have immediate weighted counterparts. To replace step 4, we use the following observation that the intersection graph of any triangle-free non-crossing rectangle family is planar:

Lemma 8. *If \mathcal{R} is triangle-free and non-crossing, then $G_{\mathcal{R}}$ is planar.*

Proof. First, notice that since \mathcal{R} is triangle-free, $R_i \cap R_j \cap R_k = \emptyset$ for every three rectangles $R_i, R_j, R_k \in \mathcal{R}$. In other words, every point p in the plane is contained in at most two rectangles. Moreover, observe that $R_i \cap R_j$ is a rectangle for every two intersecting rectangles R_i and R_j (see Figure 1).

Assume, without loss of generality, that the rectangles in $\mathcal{R} = \{R_1, \dots, R_n\}$ are numbered in such a way that if $R_i \subseteq R_j$, then $i < j$. We construct a family $\mathcal{R}' = \{R'_1, \dots, R'_n\}$ of rectilinear polygons as follows. For $i < j$ let R_{ij} be the set of points in $R_i \cap R_j$ that are not on the border of R_i . We define $R'_j = R_j \setminus \cup_{i < j} R_{ij}$. Notice that if R_i is contained in R_j , then $R'_i = R_i$. Hence, $R'_i \neq \emptyset$ for every i . Furthermore, observe that R_i and R_j intersect if and only if R'_i and R'_j intersect. Hence, $G_{\mathcal{R}'} \cong G_{\mathcal{R}}$. Finally, notice that intersections in \mathcal{R}' only contain polygon borders. It follows that $G_{\mathcal{R}'}$ is the dual of a planar graph, and thus itself is planar, and therefore $G_{\mathcal{R}}$ is also planar. \square

Thus, according to Lemma 8 above, we can apply Baker's algorithm for MINIMUM VERTEX COVER in planar graphs instead of our EPTAS in step 4 of the algorithm above. Indeed, Baker's algorithm can also handle weights. Thus, by the same analysis given above, we get a $1.5 + \varepsilon$ for the weighted variant of MINIMUM RECTANGLE VERTEX COVER. We mention also that Lemma 8 above can also be used to obtain a 1.5-approximation algorithm for the weighted variant of MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families.

Acknowledgments

We wish to acknowledge Micha Sharir for helpful comments regarding the extension of our algorithm for non-crossing rectangle families to pseudo discs.

References

1. P. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11(3-4):209218, 1998.
2. A. Apostolico, M. Atallah, and S. Hambrusch. New clique and independent set algorithms for circle graphs. *Discrete Applied Mathematics*, 36(1):1–24, 1992.
3. B. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.
4. R. Bar-Yehuda and S. Even. A linear time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2:198–203, 1981.
5. R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–46, 1985.
6. R. Bar-Yehuda, M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. In *Proceedings of the 13th annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 732–741, 2002.
7. P. Berman, B. DasGupta, S. Muthukrishnan, and S. Ramaswami. Efficient approximation algorithms for tiling and packing problems with rectangles. *Journal of Algorithms*, 41, 2001.
8. A. Butman, D. Hermelin, M. Lewenstein, and D. Rawitz. Optimization problems in multiple-interval graphs. In *Proceedings of the 18th annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 268–277, 2007.
9. E. Cenek and L. Stewart. Maximum independent set and maximum clique algorithms for overlap graphs. *Discrete Applied Mathematics*, 131(1):77–91, 2003.
10. P. Chalermsook and J. Chuzhoy. Maximum independent set of rectangles. In *Proceedings of the 20th annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 892–901, 2009.
11. T. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46, 2003.
12. T. Chan. A note on maximum independent sets in rectangle intersection graphs. *Information Processing Letters*, 89(1):19–23, 2004.
13. T. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. In *Proceedings of the 25th annual ACM Symposium on Computational Geometry (SoCG)*, 2009. To appear.
14. M. Chlebík and J. Chlebíková. Approximation hardness of optimization problems in intersection graphs of d -dimensional boxes. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 267–276, 2005.
15. K. Clarkson. A modification of the greedy algorithm for vertex cover. *Information Processing Letters*, 16:23–25, 1983.
16. R. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics Series 2*, 51:161–166, 1950.
17. I. Dinur and S. Safra. The importance of being biased. In *Proceedings of the 34th annual ACM Symposium on Theory Of Computing (STOC)*, pages 33–42, 2002.
18. J. Doerschler and H. Freeman. A rule-based system for dense-map name placement. *Communications of the ACM*, 35(1):68–79, 1992.
19. R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
20. T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34, 2005.
21. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
22. R. Fowler, M. Paterson, and S. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information Processing Letters*, 12(3):133–137, 1981.
23. H. Freeman. Computer name placement. In *Geographical Information Systems: Principles and Applications*, pages 445–456, 1991.
24. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
25. F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.
26. M. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980.
27. M. Golumbic and P. Hammer. Stability in circular arc graphs. *Journal of Algorithms*, 9(3):314–320, 1988.
28. M. Golumbic and A. Trenk. *Tolerance Graphs*. Cambridge University Press, 1985.
29. E. Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM Journal on Computing*, 31(5):1608–1623, 2002.
30. D. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11(3):555–556, 1982.

31. D. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.
32. D. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.
33. W.-L. Hsu and K.-H. Tsai. Linear time algorithms on circular-arc graphs. *Information Processing Letters*, 40(3):123–129, 1991.
34. G. Karakostas. A better approximation ratio for the vertex cover problem. In *32nd International Colloquium on Automata, Languages and Programming*, pages 1043–1050, 2005.
35. R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computation*, pages 85–103, 1972.
36. S. Khanna, S. Muthukrishnan, and M. Paterson. On approximating rectangle tiling and packing. In *Proceedings of the 9th annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 384–393, 1998.
37. D. König. Gráfok és mátrixok. *Matematikai és Fizikai Lapok*, 38:116119, 1931.
38. L. Lewin-Eytan, J. Naor, and A. Orda. Routing and admission control in networks with advance reservations. In *Proceedings of the 5th international workshop on approximation algorithms for combinatorial optimization (APPROX)*, pages 215–228, 2002.
39. R. Lipton and R. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.
40. B. Monien and E. Speckenmeyer. Ramsey numbers and an approximation algorithm for the vertex cover problem. *Acta Informatica*, 22(1):115–123, 1985.
41. G. L. Nemhauser and L. E. Trotter. Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.
42. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
43. N. Robertson and P. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.
44. V. Vazirani. *Approximation Algorithms*. Springer, 2003.