Minimum Vertex Cover in Rectangle Graphs*

Reuven Bar-Yehuda†
reuven@cs.technion.ac.il

Danny Hermelin‡
hermelin@mpi-inf.mpg.de

Dror Rawitz§
rawitz@eng.tau.ac.il

August 12, 2010

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 \( R \) where \( R_1 \setminus R_2 \) is connected for every pair of rectangles \( R_1, R_2 \in R \). This algorithm extends to intersection graphs of pseudo-disks. 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.

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

*An extended abstract was accepted to the 18th Annual European Symposium on Algorithms (ESA), 2010.
†Department of Computer Science, Technion IIT, Haifa 32000, Israel.
‡Max-Planck-Institut für Informatik, Saarbrücken, Germany.
§School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel.
1 Introduction

In this paper we are concerned with the Minimum Rectangle Vertex Cover problem: Given a set $R = \{R_1, \ldots, R_n\}$ of (weighted) axis-parallel rectangles in the plane, find a minimum size (weight) subset of rectangles in $R$ whose removal leaves the remaining rectangles in $R$ 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 [40], and probably even prior to that. Karp proved that the problem is NP-complete in his famous list of fundamental NP-complete problems [38], while Garey and Johnson extensively used Minimum Vertex Cover as an intermediate problem in many of their early NP-completeness reductions [27]. Since then, Minimum Vertex Cover played a pivoting role in the development of both approximation algorithms [34, 50], and the theory of parameterized complexity [23], 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 [6, 19, 33, 45], the first of these given in Nemhauser and Trotter’s fundamental paper [45]. Moreover, the problem is known to be approximable in within $2 - \frac{\lg \lg n}{2\lg n}$ [8, 44], within $2 - \frac{\ln \ln n}{\ln n} (1-o(1))$ [32] and even within $2 - \Theta\left(\frac{1}{\sqrt{\log n}}\right)$ [37]. 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 [21]. 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 [28]. In planar graphs, the problem is known to admit a polynomial-time approximation scheme (PTAS) [42, 7, 17], and even an efficient PTAS (EPTAS) due to Baker’s seminal framework for NP-hard planar graph problems [5].

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 [13, 39], automated label placement [2, 22, 26], and in network resource allocation with advance reservation for line topologies [13, 41], which also apply to Minimum Vertex Cover in rectangle graphs (see below). Fowler et al. [25] showed that Maximum Independent Set in rectangle graphs is NP-hard, implying the NP-hardness of Minimum Vertex Cover in rectangle graphs. Asano [4] 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 [2, 10, 15, 39]. Lewin-Eitam et al. [41] 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 [13] were able to break the $\lg n$ approximation barrier by devising a sophisticated $O(\lg \lg n)$ randomized approximation algorithm. A simpler $O(\lg n/\lg \lg n)$-approximation algorithm was given in [16]. There are also many special
cases in which \textbf{Maximum Rectangle Independent Set} admits a polynomial-approximation scheme (PTAS) \cite{2, 10, 14, 24}.

In contrast to the vast amount of research devoted to \textbf{Maximum Rectangle Independent Set} there has been surprisingly very little focus on the \textbf{Minimum Rectangle Vertex Cover} problem. Nevertheless, some of the results for \textbf{Maximum Rectangle Independent Set} carry through to \textbf{Minimum Rectangle Vertex Cover}. For instance, the result of Fowler \textit{et al.} \cite{25} implies that \textbf{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 \textbf{Maximum Rectangle Independent Set} can be converted into a PTAS for \textbf{Minimum Rectangle Vertex Cover}. Thus, the results in \cite{2} imply that \textbf{Minimum Rectangle Vertex Cover} has a PTAS when all rectangles have equal height, while \cite{14} gives a PTAS when all rectangles are squares. Erlebach \textit{et al.} \cite{24} gave an explicit PTAS for \textbf{Minimum Rectangle Vertex Cover} in bounded aspect-ratio rectangle families (without using the Nemhauser and Trotter procedure), and an EPTAS for \textbf{Minimum Rectangle Vertex Cover} in unit squares and squares is implied by \cite{43} and \cite{49}, respectively. Finally, we mention the work by Chan and Har-Peled \cite{16} who devised a PTAS for \textbf{Maximum Independent Set} in families of pseudo-disks, which are families of regions on the plane such that the boundaries of every pair of regions intersect at most twice. This result implies a PTAS for \textbf{Minimum Rectangle Vertex Cover} in non-crossing rectangle families.

\textbf{Related work.} \textbf{Minimum Vertex Cover} and its dual counterpart \textbf{Maximum Independent Set} have been previously studied in many geometric intersection graphs other than rectangle graphs. Gavril \cite{28} gave a polynomial-time algorithm for both of these problem in chordal graphs, intersection graphs of subgraphs of a tree. Apostolico \textit{et al.} \cite{3} 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 \cite{12}, while Golumbic and Hammer \cite{30} gave a polynomial-time algorithm for intersection graphs of arcs on a circle which was later improved in \cite{36}. A good survey of many generalizations of these results can be found in \cite{29, 31}. Hochbaum and Maass, and later Chlebík and Chlebíková, considered intersection graphs of $d$-dimensional boxes in $\mathbb{R}^d$ \cite{18, 35}, while Erlebach \textit{et al.} \cite{24} considered intersection graphs of general fat objects in the plane. In \cite{9, 11}, approximation algorithms were suggested for \textbf{Maximum Independent Set} and \textbf{Minimum Vertex Cover} in the class of multiple-interval graphs.

\textbf{Applications and motivation.} Automated label placement is a central problem in geographic information systems which has been extensively studied in various settings \cite{2, 22, 26}. The basic problem is to place labels around points in a geographic maps, where the labels are often assumed to be rectangles \cite{2} 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 \textbf{Minimum Rectangle Vertex Cover}.

\textbf{Minimum Rectangle Vertex Cover} can also be used to model shared-resource scheduling scenarios where 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, 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.

Finally, Muthukrishnan et al. [10, 39] describe applications for rectangle packing and covering problems that arise in data mining applications. The general idea is that database mining systems generate association rules for a given database, which can be viewed in many cases as axis parallel rectangles, e.g. the rule “Age $\in [25 \ldots 45] \land$ Balance $\in [15K \ldots 40K] \rightarrow$ Car-Loan = Yes”. Given thresholds on confidence and support, the database mining systems tag each association rule with a weight that shows their gain or value. Following that, database decision systems choose a subset of these rules for further development, such as marketing. One natural formulation of this task is to remove a subset of rectangles with smallest total value so as the remaining rectangles are pairwise disjoint. This is exactly the Minimum Vertex Cover in rectangle graphs problem.

**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 for every $R_1, R_2 \in \mathcal{R}$. (We assume w.l.o.g. that the rectangles are in general position.) 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 $\epsilon > 0$, Minimum Rectangle Vertex Cover in non-crossing rectangle families can be approximated within $(1 + \epsilon)$ in $2^{\text{poly}(1/\epsilon)} \cdot \text{poly}(n)$ time.

We mention that Minimum Rectangle Vertex Cover in non-crossing rectangle families is NP-hard according to [4]. Theorem 1 generalizes the PTAS result of Agarwal et al. [2] and Chan [14] 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. [24]. In terms of time complexity, our algorithm dramatically improves on all these algorithms, and also on the algorithm of Chan and Har-Peled [16], since all there algorithms have running-times of the form $n^{\text{poly}(1/\epsilon)}$. Furthermore, our algorithm easily extends to intersection graphs of pseudo-disks, which is the class of graphs considered in [16].

The novelty behind the algorithm in Theorem 1 lies in its usage of the arrangement graph [1] 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 a 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 [5]. 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 to 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 [5] 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.

**Overview.** The remainder of the paper is organized as follows. Section 2 contains notation and terminology and a short explanation about the Nemhauser-Trotter reduction [45]. Our EPTAS for MINIMUM RECTANGLE VERTEX COVER in non-crossing rectangle families is given in Section 3, and our $(1.5 + \varepsilon)$-approximation algorithm for MINIMUM RECTANGLE VERTEX COVER is given in Section 4.

## 2 Preliminaries

We denote our input set of axis-parallel rectangles in the plane by $\mathcal{R} = \{R_1, \ldots, R_n\}$. 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 $R_1$ contains all corners of $R_2$, and the boundaries of $R_1$ and $R_2$ do not intersect (Figure 1a).

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 (Figure 1b).

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 (Figure 1c).

Given a graph $G$, we use $V(G)$ and $E(G)$ to denote its vertex set 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 $w : V(G) \to \mathbb{Q}$. A vertex cover of $G$ is a subset of vertices $C \subseteq V(G)$ such that $\{\{u, v\} : u, v \in G\}$ for any edge $\{u, v\} \in E(G)$. For a non-negative real $\alpha \in \mathbb{R}_{\geq 0}$, an $\alpha$-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 $\text{oPT}$ is the size (weight) of a minimum vertex cover of $G$. The intersection
Figure 1: Examples of possible intersections between two rectangles $R_1$ and $R_2$.

graph $G_R$ corresponding to our input set of rectangles $\mathcal{R}$ is the graph with vertex-set $V(G_R) = \mathcal{R}$,
and edge-set $E(G_R) = \{\{R_1, R_2\} : R_1 \cap R_2 \neq \emptyset\}$. 

We will be using an important tool due to Nemhauser and Trotter [45] that allows us to focus on graphs whose entire vertex-set already constitutes a good approximate vertex-cover:

**Theorem 3** (Nemhauser&Trotter [45]). There is a polynomial-time algorithm that given a vertex-weighted graph $G$, computes a vertex set $V \subseteq V(G)$ such that:

1. $V$ is a $2$-approximate vertex-cover of $G[V]$.
2. Any $\alpha$-approximate vertex cover of $G[V]$ can be converted in polynomial-time to an $\alpha$-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 [47].

**Definition 1** (Treewidth [47]). A tree decomposition of a graph $G$ is a pair $(T, \mathcal{X})$, where $\mathcal{X} \subseteq 2^{V(G)}$ is a family of vertex subsets of $G$, and $T$ is a tree over $\mathcal{X}$, satisfying the following conditions:

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

The width of $T$ is $\max_{X \in \mathcal{X}} |X| - 1$. The treewidth of $G$, denoted $\text{tw}(G)$, 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. This algorithm extends to intersection graphs of pseudo-disks.

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 constant $q \geq 2$ 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 $\alpha$. Then **Minimum Rectangle Vertex Cover** in non-crossing rectangle families can be approximated within a factor of $\max\{\alpha, 1 + 1/q\}$ in polynomial-time.
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 $R_1 \subseteq R_2$, then (in the unweighted case) we may assume w.l.o.g. that any vertex cover of $G_\mathcal{R}$ includes $R_2$. Indeed, if $\mathcal{C}$ is a vertex-cover of $G_\mathcal{R}$ that does not contain $R_2$, then $R_1 \in \mathcal{C}$, and therefore $\mathcal{C}' = (\mathcal{C} \setminus \{R_1\}) \cup \{R_2\}$ is also a vertex-cover of $G_\mathcal{R}$. Second, note that if $Q_1, Q_2, \ldots, Q_r$ are cliques of size $q + 1$ in $G_\mathcal{R}$ such that $Q_i \cap Q_j = \emptyset$ for every $i \neq j$, then any vertex-cover of $G_\mathcal{R}$ must include at least $r \cdot q$ rectangles from $\bigcup_{1 \leq i \leq r} Q_i$. Thus, by including all rectangles from these cliques in our solution, we deviate (again, in the unweighted case) by at most a factor of $1 + \frac{1}{q}$ 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 $\mathcal{R}$, and $\mathcal{Q}$ is a greedy packing of disjoint $(q + 1)$-cliques in $G_\mathcal{R}\setminus \mathcal{P}$. Observe that both these sets can be computed in polynomial-time, $\mathcal{P}$ by obvious brute force, and $\mathcal{Q}$ by the fact that every clique in $\mathcal{R}$ is represented by some intersection point of two rectangles (or even by a rectangle corner as $\mathcal{R} \setminus \mathcal{P}$ has only corner intersections). We then apply the $\alpha$-approximation algorithm assumed in the lemma to obtain a $\alpha$-approximate vertex-cover $\mathcal{C}$ for $\mathcal{R} \setminus (\mathcal{P} \cup \mathcal{Q})$. According to the two observations above, $\mathcal{P} \cup \mathcal{Q} \cup \mathcal{C}$ is a max$\{\alpha, 1 + \frac{1}{q}\}$-approximate vertex-cover of $G_\mathcal{R}$. \[ \square \]

Due to Lemma 1, 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 its 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

In this section we present several properties of arrangement graphs [1] of rectangle families. An intersection of two rectangle boundaries is called a joint. The arrangement graph $A_\mathcal{R}$ of a rectangle family $\mathcal{R}$ is the multi-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 $A_\mathcal{R}$ if and only if $u$ 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 and 4-regular (see example in Fig. 2).
Figure 2: A set $\mathcal{R}$ of corner intersecting 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.** $|\mathcal{R}(J)| \leq 2|J|$ for any set of joints $J \subseteq V(A_{\mathcal{R}})$.

The following lemma states that the number of joints is $A_{\mathcal{R}}$ is linear in $|\mathcal{R}|$.

**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 $R_1$ and $R_2$ 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 $\sum_{R \in \mathcal{R}} \deg(R) \leq 4q \cdot |\mathcal{R}|$, where $\deg(R)$ denotes the number of rectangles intersecting $R$. Thus, $|E(G_{\mathcal{R}})| \leq 2q \cdot |\mathcal{R}|$, and so $|V(A_{\mathcal{R}})| \leq 2 \cdot |E(G_{\mathcal{R}})| \leq 4q \cdot |\mathcal{R}|$. 

### 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 [5] 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. [46]), gives an EPTAS for **Minimum Vertex Cover** in planar graphs.

**Lemma 4** (Baker [5]). *Given a planar graph $G$ and an integer $k$, one can partition in polynomial-time $V(G)$ into $k$ subsets $V_1, \ldots, V_k$ such that $\text{tw}(G - V_i) \leq 3k$ for all $i$, $1 \leq i \leq k$.*

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 $G_R$. Also, our algorithm will produce vertex-sets whose deletion results in a subgraph of $G_R$ 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 $U_1, \ldots, U_k$ with the following properties:

1. $\bigcup_i U_i = V(G)$;
2. $\sum_i |U_i| \leq c_1 \cdot n$; and
3. tw$(G - U_i) \leq c_2 \cdot k$ for every $i$.

Then one can compute a 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 $U_1, \ldots, U_k$ 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 $\text{OPT}$ is the size of a minimum vertex cover of $G$. For each $i$, $1 \leq i \leq k$, let $H_i$ denote the subgraph $H_i = G_i - U_i$, and let $\text{OPT}_i$ be the size of a 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 $\text{OPT}_i$ can be computed in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time. We thus obtain $k$ candidate vertex-covers for $G$, $C_i = U_i \cup C_i^*$, and we have

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

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

**3.3 Our Algorithm**

We are now in position to describe our EPTAS. The key lemma we need is Lemma 6 below that allows us to convert tree-decompositions of $A_R$ to a tree-decompositions of $G_R$ of approximately the same width.

**Lemma 6.** tw$(G_R) \leq 2 \cdot \text{tw}(A_R) + 1$.

**Proof.** Let $(T, \mathcal{X})$ is a tree-decomposition of $A_R$ whose width it tw$(A_R)$. Now let $\mathcal{X}' = \{R(X) : X \in \mathcal{X}\}$, and let $T'$ be a tree over $\mathcal{X}'$ with an edge $\{R(X_1), R(X_2)\}$ for every edge $\{X_1, X_2\}$ in $T$. We show that $(T', \mathcal{X}')$ is a tree-decomposition of $G_R$, namely that $(T', \mathcal{X}')$ 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 $j \in V(A_R)$ that corresponds to both these rectangles. Hence, for every edge $\{R_1, R_2\} \in E(G_R)$, there is at least one node in $\mathcal{X}'$ which contains both $R_1$ and $R_2$. Thus, $\bigcup_{X \in \mathcal{X}'} G_R[X] = G_R$. 

8
Now suppose there is some rectangle $R$ which is contained in two nodes $\mathcal{R}(X_1)$ and $\mathcal{R}(X_2)$ of $T'$. 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 $j_1, i_1, \ldots, i_r, j_2$ connecting $j_1$ to $j_2$ in $A_{\mathcal{R}}$, where $i_1, \ldots, i_r$ are all joints of $R$. Since $(T, \mathcal{X})$ is a proper tree decomposition of $A_{\mathcal{R}}$, it follows that there is a path $X_1, Y_1, \ldots, Y_r, X_2$ connecting $X_1$ and $X_2$ in $T$, with $Y_i \cap \{j_1, i_1, \ldots, i_r, j_2\} \neq \emptyset$ for each $i$, $1 \leq i \leq s$. Thus, each node in the path $\mathcal{R}(X_1), \mathcal{R}(Y_1), \ldots, \mathcal{R}(Y_s), \mathcal{R}(X_2)$ connecting $\mathcal{R}(X_1)$ and $\mathcal{R}(X_2)$ in $T'$ contains $R$, and since $R$, $\mathcal{R}(X_1)$, and $\mathcal{R}(X_2)$ were chosen arbitrarily, this shows that for each $R \in \mathcal{R}$: \( \{ \mathcal{R}(X) \in \mathcal{X}': R \in X \} \) is connected in $T'$. Thus, both requirements of Definition 1 are fulfilled by $(T', \mathcal{X}')$.

Finally, observe that due to Lemma 2, $\max_{X' \in \mathcal{X}'} |X'| \leq 2 \max_{X \in \mathcal{X}} |X|$. It follows that the width of $(T', \mathcal{X}')$ is at most $2tw(A_{\mathcal{R}}) + 1$, and we are done.

Our algorithm consists of the following steps:

1. Set $q = \lceil 1/\varepsilon \rceil$ and $k = \lceil 8q/\varepsilon \rceil = \lceil 8/\varepsilon^2 \rceil$.
2. Apply Lemma 1 so that $\mathcal{R}$ does not have any containment intersections and no pairwise intersecting subsets of rectangles of size greater than $q$.
3. Apply the Nemhauser&Trotter Theorem on $G_{\mathcal{R}}$, and let $\mathcal{R}' \subseteq \mathcal{R}$ denote the resulting subset of rectangles.
4. Construct the arrangement graph $A_{\mathcal{R}'}$ corresponding to $\mathcal{R}'$, and partition $A_{\mathcal{R}'}$ into $k$ subsets $V_1, \ldots, V_k$ using Lemma 4.
5. Use Lemma 5 on $G_{\mathcal{R}'}$ with $U_i = \mathcal{R}'(V_i)$, for every $i$.

Observe that the arrangement graph of $\mathcal{R}' \setminus U_i$ is a subgraph of $A_{\mathcal{R}'} - V_i$. Hence, according to Lemmas 2, 3 and 6 above, $U_1, \ldots, U_k$ satisfy the three conditions of Lemma 5. Thus the above algorithm outputs a $(1 + \varepsilon)$-approximate vertex-cover of $G_{\mathcal{R}}$ in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time. This proves Theorem 1.

Finally, we mention that our EPTAS can be modified to deal with intersection graphs of pseudo-disks. Specifically, in Step 2, instead of removing cliques, we remove point cliques, namely subsets of pseudo-disks $Q$ such that $\bigcap_{R \in Q} R \neq \emptyset$. This is sufficient, since the number of joints in the arrangement graph $A_D$ of a set $D$ of pseudo-disks, where no point is contained in more than $q$ pseudo-disks, is $O(q \cdot |D|)$ [48].

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. [41]. 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 (i) $X_1 \subseteq X_2$ and $Y_2 \subseteq Y_1$, or (ii) $X_2 \subseteq X_1$ and $Y_1 \subseteq Y_2$. It follows that the crossing relation between rectangles forms a partial order. Thus, by Dilworth’s Theorem [20], and due to the fact that $\mathcal{R}$ is triangle-free, there exists a partitioning $\Pi$ of $\mathcal{R}$ into two non-crossing subsets as desired. Furthermore, using one of many classical minimum anti-chain partitioning algorithms (see e.g. [29]), one can compute $\Pi$ in polynomial-time.

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 $\varepsilon$-approximate vertex-cover $\mathcal{C}_1$ of $G_{\mathcal{R}_1}$ and an $\varepsilon$-approximate vertex-cover $\mathcal{C}_2$ of $G_{\mathcal{R}_2}$ using 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 the 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 $\text{OPT}$, $\text{OPT}_1$, and $\text{OPT}_2$ denote the size of the minimum vertex-covers of $G_{\mathcal{R}''}$, $G_{\mathcal{R}_1}$, and $G_{\mathcal{R}_2}$ respectively, we get:

$$|\mathcal{R}_1 \cup \mathcal{C}_2| + |\mathcal{R}_2 \cup \mathcal{C}_1| \leq |\mathcal{R}| + (1 + \varepsilon)\text{OPT}_1 + (1 + \varepsilon)\text{OPT}_2$$

$$\leq 2\text{OPT} + (1 + \varepsilon)\text{OPT}$$

$$= (3 + \varepsilon)\text{OPT}.$$ 

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 [8] (see e.g. [11]). 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. 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 $\mathcal{R} = \{R_1, \ldots, R_n\}$ are numbered in such a way that if $R_i \subseteq R_j$, then $i < j$. We construct a family $\mathcal{R}' = \{R'_1, \ldots, R'_n\}$ 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 $R'_i = R_j \cup \cup_{i<j} R_{ij}$. 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.

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.

Acknowledgment

We thank Micha Sharir for helpful discussions.

References


