Minimum Vertex Cover in Rectangle Graphs

Reuven Bar-Yehuda[1], Danny Hermelin[2], and Dror Rawitz[3]

[1] Department of Computer Science, Technion IIT, Haifa 32000, Israel
reuven@cs.technion.ac.il
[2] Max-Planck-Institut für Informatik, Saarbrücken, Germany
hermelin@mpi-inf.mpg.de
[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 $R_1, R_2 \in \mathcal{R}$. 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.

1 Introduction

In this paper we are concerned with the Minimum Rectangle Vertex Cover problem: Given a set $\mathcal{R} = \{R_1, \ldots, R_n\}$ 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 [1], and probably even prior to that. Karp proved that the problem is NP-complete in his famous list of fundamental NP-complete problems [2], while Garey and Johnson extensively used Minimum Vertex Cover as an intermediate problem in many of their early NP-completeness reductions [3]. Since then, Minimum Vertex Cover played a pivoting role in the development of both approximation algorithms [4,5], and the theory of parameterized complexity [6], the two main disciplines for coping with the widespread phenomena of NP-hardness.

M. de Berg and U. Meyer (Eds.): ESA 2010, Part I, LNCS 6346, pp. 255–266, 2010.

From the perspective of approximation algorithms, Minimum Vertex Cover has many polynomial-time algorithms achieving an approximation ratio of 2 [7,8,9,10], the first of these given in Nemhauser and Trotter's fundamental paper [10]. Moreover, the problem is known to be approximable in within $2-\frac{\lg\lg n}{2\lg n}$ [11,12], within $2-\frac{\ln\ln n}{\ln n}(1-o(1))$ [13] and even within $2-\Theta(\frac{1}{\sqrt{\log n}})$ [14]. On the other hand, it is also known that Minimum Vertex Cover is inapproximable within a factor of 1.36, unless P=NP [15]. 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 [16]. In planar graphs, the problem is known to admit a polynomial-time approximation scheme (PTAS) [17,18,19], and even an efficient PTAS (EPTAS) due to Baker's seminal framework for NP-hard planar graph problems [20].

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 [21,22], automated label placement [23,24,25], and in network resource allocation with advance reservation for line topologies [21,26], which also apply to Minimum Vertex Cover in rectangle graphs (see below). Fowler *et al.* [27] showed that Maximum Independent Set in rectangle graphs is NP-hard, implying the NP-hardness of Minimum Vertex Cover in rectangle graphs. Asano [28] 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 [23,29,30,22]. Lewin-Eitan *et al.* [26] 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 [21] 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 [31]. There are also many special cases in which Maximum Rectangle Independent Set admits a polynomial-approximation scheme (PTAS) [23,29,32,33].

In contrast to the vast amount of research devoted to 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.* [27] 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 [23] imply that Minimum Rectangle Vertex Cover has a PTAS when all rectangles have equal height, while [32] gives a PTAS when all rectangles are squares. Erlebach *et al.* [33] gave an explicit PTAS for Minimum Rectangle

Vertex Cover in bounded aspect-ratio rectangle families (without using the Nemhauser and Trotter procedure), and an EPTAS for Minimum Rectangle Vertex Cover in unit squares and squares is implied by [34] and [35], respectively. Finally, we mention the work by Chan and Har-Peled [31] who devised a PTAS for 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 Minimum Rectangle Vertex Cover in non-crossing rectangle families.

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 [16] gave a polynomial-time algorithm for both of these problem in chordal graphs, intersection graphs of subgraphs of a tree. Apostolico *et al.* [36] 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 [37], while Golumbic and Hammer [38] gave a polynomial-time algorithm for intersection graphs of arcs on a circle which was later improved in [39]. A good survey of many generalizations of these results can be found in [40,41]. Hochbaum and Maass, and later Chlebík and Chlebíková, considered intersection graphs of d-dimensional boxes in $\mathbb{R}^d$ [42,43], while Erlebach *et al.* [33] considered intersection graphs of general fat objects in the plane. In [44,45], approximation algorithms were suggested for Maximum Independent Set and Minimum Vertex Cover in the class of multiple-interval graphs.

Applications and motivation. Automated label placement is a central problem in geographic information systems which has been extensively studied in various settings [23,24,25]. The basic problem is to place labels around points in a geographic maps, where the labels are often assumed to be rectangles [23] 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 Minimum Rectangle Vertex Cover.

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.

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:

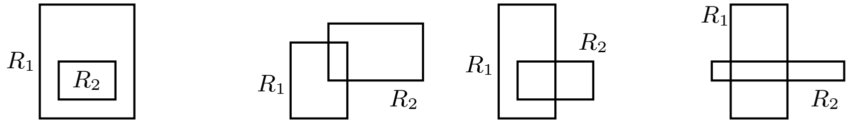
Theorem 1. *Given any $\varepsilon > 0$,* Minimum Rectangle Vertex Cover *in non-crossing rectangle families can be approximated within $(1+\varepsilon)$ in $2^{\text{poly}(1/\varepsilon)} \cdot \text{poly}(n)$ time.*

We mention that Minimum Rectangle Vertex Cover in non-crossing rectangle families is NP-hard according to [28]. Theorem 1 generalizes the PTAS result of Agarwal *et al.* [23] and Chan [32] 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.* [33]. In terms of time complexity, our algorithm dramatically improves on all these algorithms, and also on the algorithm of Chan and Har-Peled [31], since all there algorithms have running-times of the form $n^{\text{poly}(1/\varepsilon)}$. Furthermore, our algorithm easily extends to intersection graphs of pseudo-disks, which is the class of graphs considered in [31].

The novelty behind the algorithm in Theorem 1 lies in its usage of the *arrangement graph* [46] 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 [20]. 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 [20] 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.*



(a) Containment. (b) Corner. (c) Crossing.

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

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 [10]. 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. Several proofs are omitted due to space limitations.

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 (Fig. 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 (Fig. 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 (Fig. 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 $w : V(G) \rightarrow \mathbb{Q}$. 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 *$\alpha$-approximate vertex cover* of G is a vertex cover C with $|C| \leq \alpha \cdot \textsc{opt}$ (or $w(C) \leq \alpha \cdot \textsc{opt}$ in the weighted case), where $\textsc{opt}$ is the size (weight) of a minimum 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 [10] that allows us to focus on graphs whose entire vertex-set already constitutes a good approximate vertex-cover:

Theorem 3 (Nemhauser&Trotter [10]). *There is a polynomial-time algorithm that given a vertex-weighted graph G, computes a vertex set $V \subseteq V(G)$ such that: (i) V is a 2-approximate vertex-cover of $G[V]$, and (ii) any α-approximate vertex cover of $G[V]$ 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 [47].

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

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

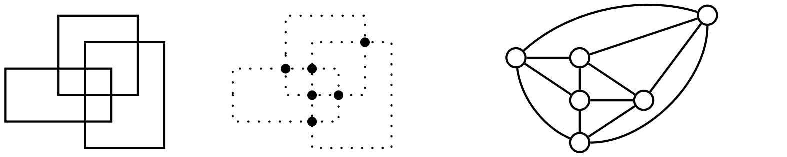

(a) Rectangle Set. (b) The joints. (c) Resulting arrangement graph.

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

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* [46] 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).

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

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 [20] 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.* [48]), gives an EPTAS for Minimum Vertex Cover in planar graphs.

Lemma 4 (Baker [20]). *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* $\mathrm{tw}(G - V_i) \leq 3k$ *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 $G_{\mathcal{R}}$. 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 $U_1, \ldots, U_k$ with the following properties: (1) $\bigcup_i U_i = V(G)$; (2) $\sum_i |U_i| \leq c_1 \cdot n$; and (3) $\mathrm{tw}(G - U_i) \leq c_2 \cdot k$ for every i. Then one can compute a vertex cover of G within a factor of $(1 + \varepsilon)$ in $2^{\mathrm{poly}(1/\varepsilon)} \cdot \mathrm{poly}(n)$ time, for any given $\varepsilon > 0$.*

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 $A_{\mathcal{R}}$ to a tree-decompositions of $G_{\mathcal{R}}$ of approximately the same width.

Lemma 6. $\mathrm{tw}(G_{\mathcal{R}}) \leq 2 \cdot \mathrm{tw}(A_{\mathcal{R}}) + 1$.

Proof. Let $(\mathcal{T}, \mathcal{X})$ is a tree-decomposition of $A_{\mathcal{R}}$ whose width it $\mathrm{tw}(A_{\mathcal{R}})$. Now let $\mathcal{X}' = \{\mathcal{R}(X) : X \in \mathcal{X}\}$, and let $\mathcal{T}'$ be a tree over $\mathcal{X}'$ with an edge $\{\mathcal{R}(X_1), \mathcal{R}(X_2)\}$ for every edge $\{X_1, X_2\}$ in $\mathcal{T}$. We show that $(\mathcal{T}', \mathcal{X}')$ is a tree-decomposition of $G_{\mathcal{R}}$, namely 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 $j_1, i_1, \ldots, i_r, j_2$ connecting j_1 to j_2 in $A_{\mathcal{R}}$, where $i_1, \ldots, i_r$ are all joints of R. Since $(\mathcal{T}, \mathcal{X})$ 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 $\mathcal{T}$, 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 $\mathcal{T}'$ contains R, and since R, $\mathcal{R}(X_1)$, and $\mathcal{R}(X_2)$ were chosen arbitrarily, this shows that for each $R \in \mathcal{R}$: $\{\mathcal{R}(X) \in \mathcal{X}' : R \in X\}$ is connected in $\mathcal{T}'$. Thus, both requirements of Definition 1 are fulfilled by $(\mathcal{T}', \mathcal{X}')$.

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 $(\mathcal{T}', \mathcal{X}')$ is at most $2\text{tw}(A_{\mathcal{R}}) + 1$, 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 $\mathcal{Q}$ such that $\bigcap_{R \in \mathcal{Q}} R \neq \emptyset$. This is sufficient, since the number of joints in the arrangement graph $A_{\mathcal{D}}$ of a set $\mathcal{D}$ of pseudo-disks, where no point is contained in more than q pseudo-disks, is $O(q \cdot |\mathcal{D}|)$ [49].

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.* [26]. 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.*

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 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 $\mathcal{R}_1 \cup C_2$ and $\mathcal{R}_2 \cup C_1$ are vertex covers for $G_{\mathcal{R}''}$. Furthermore, letting $\textsc{opt}$, $\textsc{opt}_1$, and $\textsc{opt}_2$ denote the size of the minimum vertex-covers of $G_{\mathcal{R}''}$, $G_{\mathcal{R}_1}$, and $G_{\mathcal{R}_2}$ respectively, we get: $|\mathcal{R}_1 \cup C_2| + |\mathcal{R}_2 \cup C_1| \leq |\mathcal{R}| + (1+\varepsilon)\textsc{opt}_1 + (1+\varepsilon)\textsc{opt}_2 \leq 2\textsc{opt} + (1+\varepsilon)\textsc{opt} = (3+\varepsilon)\textsc{opt}$. 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 [11] (see *e.g.* [45]). 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.*

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

1. Kőnig, D.: Gráfok és mátrixok. Matematikai és Fizikai Lapok 38, 116–119 (1931)
2. Karp, R.: Reducibility among combinatorial problems. In: Complexity of Computer Computation, pp. 85–103 (1972)
3. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)

4. Hochbaum, D.: Approximation Algorithms for NP-hard Problems. PWS Publishing Company (1997)
5. Vazirani, V.: Approximation Algorithms. Springer (2003)
6. Downey, R., Fellows, M.: Parameterized Complexity. Springer, Heidelberg (1999)
7. Bar-Yehuda, R., Even, S.: A linear time approximation algorithm for the weighted vertex cover problem. J. Algorithm 2, 198–203 (1981)
8. Clarkson, K.: A modification of the greedy algorithm for vertex cover. Inform. Process. Lett. 16, 23–25 (1983)
9. Hochbaum, D.: Approximation algorithms for the set covering and vertex cover problems. SIAM J. Comput. 11(3), 555–556 (1982)
10. Nemhauser, G.L., Trotter, L.E.: Vertex packings: structural properties and algorithms. Mathematical Programming 8, 232–248 (1975)
11. Bar-Yehuda, R., Even, S.: A local-ratio theorem for approximating the weighted vertex cover problem. Annals of Discrete Mathematics 25, 27–46 (1985)
12. Monien, B., Speckenmeyer, E.: Ramsey numbers and an approximation algorithm for the vertex cover problem. Acta Informatica 22(1), 115–123 (1985)
13. Halperin, E.: Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. SIAM J. Comput. 31(5), 1608–1623 (2002)
14. Karakostas, G.: A better approximation ratio for the vertex cover problem. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1043–1050. Springer, Heidelberg (2005)
15. Dinur, I., Safra, S.: The importance of being biased. In: 34th annual ACM Symposium on Theory of Computing, pp. 33–42 (2002)
16. Gavril, F.: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM J. Comput. 1(2), 180–187 (1972)
17. Lipton, R., Tarjan, R.: Applications of a planar separator theorem. SIAM J. Comput. 9(3), 615–627 (1980)
18. Bar-Yehuda, R., Even, S.: On approximating a vertex cover for planar graphs. In: 14th Annual ACM STOC, pp. 303–309 (1982)
19. Chiba, N., Nishizeki, T., Saito, N.: Applications of the Lipton and Tarjan's planar separator theorem. Journal of information processing 4(4), 203–207 (1981)
20. Baker, B.: Approximation algorithms for NP-complete problems on planar graphs. J. ACM 41(1), 153–180 (1994)
21. Chalermsook, P., Chuzhoy, J.: Maximum independent set of rectangles. In: 20th Annual ACM-SIAM SODA, pp. 892–901 (2009)
22. Khanna, S., Muthukrishnan, S., Paterson, M.: On approximating rectangle tiling and packing. In: 9th annual ACM-SIAM SODA, 384–393 (1998)
23. Agarwal, P.K., van Kreveld, M.J., Suri, S.: Label placement by maximum independent set in rectangles. Comput. Geom. 11(3-4), 209–218 (1998)
24. Doerschler, J., Freeman, H.: A rule-based system for dense-map name placement. Communications of the ACM 35(1), 68–79 (1992)
25. Freeman, H.: Computer name placement. In: Geographical Information Systems: Principles and Applications, pp. 445–456 (1991)
26. Lewin-Eytan, L., Naor, J., Orda, A.: Admission control in networks with advance reservations. Algorithmica 40(4), 293–304 (2004)
27. Fowler, R., Paterson, M., Tanimoto, S.: Optimal packing and covering in the plane are NP-complete. Inform. Process. Lett. 12(3), 133–137 (1981)
28. Asano, T.: Difficulty of the maximum independent set problem on intersection graphs of geometric objects. In: 6th ICTAG (1991)

29. Berman, P., DasGupta, B., Muthukrishnan, S., Ramaswami, S.: Efficient approximation algorithms for tiling and packing problems with rectangles. J. Algorithm 41 (2001)
30. Chan, T.: A note on maximum independent sets in rectangle intersection graphs. Inform. Process. Lett. 89(1), 19–23 (2004)
31. Chan, T.M., Har-Peled, S.: Approximation algorithms for maximum independent set of pseudo-disks. In: 25th annual ACM SOCG, pp. 333–340 (2009)
32. Chan, T.M.: Polynomial-time approximation schemes for packing and piercing fat objects. J. Algorithm 46 (2003)
33. Erlebach, T., Jansen, K., Seidel, E.: Polynomial-time approximation schemes for geometric intersection graphs. SIAM J. Comput. 34 (2005)
34. Marx, D.: Efficient approximation schemes for geometric problems? In: 13th annual European Symposium on Algorithms, pp. 448–459 (2005)
35. van Leeuwen, E.J.: Better approximation schemes for disk graphs. In: 10th Scandinavian Workshop on Algorithm Theory, pp. 316–327 (2006)
36. Apostolico, A., Atallah, M., Hambrusch, S.: New clique and independent set algorithms for circle graphs. Discrete Appl. Math. 36(1), 1–24 (1992)
37. Cenek, E., Stewart, L.: Maximum independent set and maximum clique algorithms for overlap graphs. Discrete Appl. Math. 131(1), 77–91 (2003)
38. Golumbic, M., Hammer, P.: Stability in circular arc graphs. J. Algorithm 9(3), 314–320 (1988)
39. Hsu, W.L., Tsai, K.H.: Linear time algorithms on circular-arc graphs. Inform. Process. Lett. 40(3), 123–129 (1991)
40. Golumbic, M.: Algorithmic graph theory and perfect graphs. Academic Press, New York (1980)
41. Golumbic, M., Trenk, A.: Tolerance Graphs. Cambridge University Press, Cambridge (1985)
42. Chlebík, M., Chlebíková, J.: Approximation hardness of optimization problems in intersection graphs of d-dimensional boxes. In: 16th Annual ACM-SIAM SODA 2005, pp. 267–276 (2005)
43. Hochbaum, D., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM 32(1), 130–136 (1985)
44. Bar-Yehuda, R., Halldórsson, M.M., Naor, J., Shachnai, H., Shapira, I.: Scheduling split intervals. SIAM J. Comput. 36(1), 1–15 (2006)
45. Butman, A., Hermelin, D., Lewenstein, M., Rawitz, D.: Optimization problems in multiple-interval graphs. In: 18th annual ACM-SIAM SODA, pp. 268–277 (2007)
46. Agarwal, P.K., Sharir, M.: Arrangements and their applications. In: Handbook of Computational Geometry, pp. 49–119 (1998)
47. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. J. Algorithm 7, 309–322 (1986)
48. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, Oxford (2006)
49. Sharir, M., Agarwal, P.K.: Davenport-Schinzel Sequences and Their Geometric Applications. Cambridge University Press, Cambridge (1995)