Universität Augsburg

Approximation Algorithms for Intersection Graphs

F. Kammer, T. Tholey und H. Voepel

Report 2009-6 Mai 2009

Institut für Informatik
D-86135 Augsburg

Approximation Algorithms for Intersection Graphs

Frank Kammer, Torsten Tholey, and Heiko Voepel

Institut für Informatik, Universität Augsburg, D-86135 Augsburg, Germany
{kammer,tholey,voepel}@informatik.uni-augsburg.de

Abstract. We introduce three new complexity parameters that in some sense measure how chordal-like a graph is. The similarity to chordal graphs is used to construct simple polynomial-time approximation algorithms with constant approximation ratio for many $\mathcal{NP}$-hard problems, when restricted to graphs for which at least one of our new complexity parameters is bounded by a constant. As applications we present approximation algorithms with constant approximation ratio for maximum weighted independent set, minimum (independent) dominating set, minimum vertex coloring, maximum weighted clique, and minimum clique partition for large classes of intersection graphs.

1 Introduction

Complexity parameters can help to solve many $\mathcal{NP}$-hard problems of theoretical and practical importance on a subclass of instances for which the chosen parameter is very small. Treewidth is one of the classical complexity parameters studied in graph theory. Graphs of bounded treewidth have a treelike structure that allows to generalize efficient algorithms for hard problems on trees to graphs of bounded treewidth. In particular, all decision problems that can be expressed in monadic second-order logic can be solved in polynomial time on graphs of bounded treewidth [2, 7].

In this paper we introduce three new complexity parameters that all generalize in some kind another class of graphs, namely chordal graphs. See Section 2 for a detailed definition of our new complexity parameters. Like trees, chordal graphs have a simple structure that also allows to solve a large number of $\mathcal{NP}$-hard problems. For example, on chordal graphs there are linear time algorithms for minimum vertex coloring (MVC), maximum clique (MC), minimum clique partition (MCP) [13], and for maximum weighted independent set (MWIS) [11]. Therefore, it seems natural to search for a generalization of chordal graphs. In doing so, we obtain new approximation algorithms for the problems above on large graph classes containing many intersection graph classes such as t-interval-graphs, circular arc graphs, disk graphs, and intersection graphs of regular polygons or of arbitrary polygons of so-called bounded fatness. In general, intersection graphs are very useful subclasses of graphs with several practical applications. See [14] or [15] for an overview of different applications of intersection graphs.It is not surprising that, for small graph classes such as unit disk graphs, one can achieve better results than by our new algorithms designed for larger classes of graphs. Nevertheless, also on small graph classes such as disk graphs we obtain new results for some of the problems mentioned above as well as for minimum dominating set (MDS) and minimum independent dominating set (MIDS).

Table 1 summarizes the best previously known and new approximation results for the intersection graphs of disks, regular polygons, fat objects, t-intervals and t-fat-objects. MIS denotes the unweighted version of MWIS and MWC the weighted version of MC. By an *r-regular polygon* we mean a polygon with r corners placed on a circle such that all pairs of consecutive corners of the polygon have the same distance. We assume that $r \in O(1)$. We also define a set $\mathcal{C}$ of geometric objects in $\mathbb{R}^d$—i.e., a set of points in $\mathbb{R}^d$—to be a set of *fat objects* if the following holds. First, let us call the radius of a smallest d-dimensional ball containing the closure of a geometric object S in $\mathbb{R}^d$ the *size* of S. Moreover, let R be the size of the largest object in $\mathcal{C}$. Then, $\mathcal{C}$ is called *fat* if there is a constant c such that, for every d-dimensional ball B of radius r with $0 < r \leq R$, there exist c points (possibly also outside B) such that every B-intersecting object $S \in \mathcal{C}$ of size at least r contains one or more of the c points. We also say that $\mathcal{C}$ has *fatness c*. $\mathcal{C}$ is called a *(c-)restricted set of fat objects* if in the condition above every B-intersecting object in $\mathcal{C}$ (with arbitrary size) contains at least one of the c points. By a *unit* set of objects—in opposite to *arbitrary*—we mean that each object must be a copy of each other object, i.e., of the same size and shape. However, unit and arbitrary objects may be rotated and moved to any position. By an intersection graph G of t-intervals we mean an intersection graph, where each vertex represents a *t-interval*, i.e., the union of t intervals taken from a set S of intervals. By the intersection graph G of *t-fat-objects* we mean an intersection graph, where each vertex represents a *t-fat-object*, i.e., the union of t objects taken from a fat set S of objects. In both cases S is called the *universe* of G.

As usual, disks and regular polygons should be defined in the plane $\mathbb{R}^2$, intervals in $\mathbb{R}$ and fat objects in $\mathbb{R}^d$, where we assume that $d = O(1)$. Concerning the results in table 1 including the hardness results, we assume that—beside an intersection graph itself—a representation of the intersection graph is given. More precisely, for the intersection graph of a set S of 1) disks, 2) r-regular polygons, 3) t-intervals, 4) fat objects, or 5) t-fat-objects, we are given for each element in S in case 1) its radius and the coordinates of its center, in case 2) the coordinates of the center and at least one corner, in case 3) the start and end point of each interval, and in case 4) a representation that, for each pair x, y of objects, each point $p \in \mathbb{R}^d$, and each d-dimensional ball B represented by the coordinates of its center and its radius $r \leq R$, supports the following computations in polynomial time: Decide whether x and y intersect, whether x and B intersect, and whether p is contained in x, determine the size s of x and the center of ball B with a radius s containing the closure of x, and find c points that are contained in every object of size $\geq r$ intersecting B. In case 5) each t-fat-object has a representation of its objects as described in case 4). For many applications, representations as described above are given explicitly.

	disk	r-reg. polygon	fat objects	t-interval	t-fat-objects
MIS	arbitrary: PTAS [5, 8] unit: PTAS [18] $\mathcal{NP}$-h. [10]	arbitrary: PTAS [5, 8] unit: PTAS [18] $\mathcal{NP}$-h. [10]	arbitrary: PTAS [5, 8] unit: $\mathcal{NP}$-h. [10]	$2t$-PA. [1] $t \geq 3$: $\mathcal{APX}$-h. [17, 25]	arbitrary: $O(t)$-PA. [$*$] $\mathcal{NP}$-h. [10]
MWIS	arbitrary: PTAS [8]	arbitrary: PTAS [8]	arbitrary: PTAS [8]	$2t$-PA. [1]	arbitrary: $O(t)$-PA. [$*$]
MDS	restricted: $O(1)$-PA. [$*$] unit: PTAS [18] $\mathcal{NP}$-h. [6]	restricted: $O(1)$-PA. [$*$] unit: PTAS [18] $\mathcal{NP}$-h. [6]	restricted: $O(1)$-PA. [$*$] unit: $\mathcal{NP}$-h. [6]	t^2-PA. [3] $t \geq 2$: $\mathcal{APX}$-h. [17, 25]	restricted: $O(t)$-PA. [$*$] $t \geq 2$: $\mathcal{APX}$-h. [17, 25]
MIDS	restricted: $O(1)$-PA. [$*$] unit: PTAS [19] $\mathcal{NP}$-h. [6]	restricted: $O(1)$-PA. [$*$] unit: PTAS [19] $\mathcal{NP}$-h. [6]	restricted: $O(1)$-PA. [$*$] unit: $\mathcal{NP}$-h. [6]		restricted: $O(t)$-PA. [$*$] $t \geq 2$: $\mathcal{NP}$-h. [6]
MVC	arbitrary: 5-PA. [**?**,22, 23] unit: 3-PA. [23] 4/3-PA. is $\mathcal{NP}$ -h. [6, 12, 20]	arbitrary: $O(1)$-PA. [$*$] unit: $O(1)$-PA. [23] $\mathcal{NP}$-h. [12, 20]	arbitrary: $O(1)$-PA. [$*$] unit: $\mathcal{NP}$-h. [12, 20]	$2t$-PA. [1]	restricted: $O(t)$-PA. [$*$]
MC	arbitrary: $O(1)$-PA. [$*$] unit: $\in \mathcal{P}$ [6]	arbitrary: $O(1)$-PA. [$*$]	arbitrary: $O(1)$-PA. [$*$]	$\frac{t^2-t+1}{2}$-PA. [3] $O(t)$-PA. [$*$] $t \geq 3$: $\mathcal{NP}$-h. [3]	arbitrary: $O(t)$-PA. [$*$] $t \geq 3$: $\mathcal{NP}$-h. [3]
MWC	arbitrary: $O(1)$-PA. [$*$]	arbitrary: $O(1)$-PA. [$*$]	arbitrary: $O(1)$-PA. [$*$]	$\frac{t^2-t+1}{2}$-PA. [3] $O(t)$-PA. [$*$]	arbitrary: $O(t)$-PA. [$*$]
MCP	arbitrary: $O(1)$ PA. [$*$] unit: 3-PA. [4]	arbitrary: $O(1)$-PA. [$*$]	arbitrary: $O(1)$-PA. [$*$]	$O(\log^2 n/$ $\log(1 + 1/t))$- PA. [$*$]	arbitrary: $O(t)$-PA. [$*$]

Table 1. Approximation results. We use PA. and $\mathcal{NP}$-h. as abbreviation for polynomial-time approximation algorithm and $\mathcal{NP}$-hard. By n we denote the number of vertices of the intersection graph. [$*$] denotes new results shown in this paper.

Other generalized classes of graphs including the intersection graphs of unit disks or r-regular polygons of unit size are graph classes of so-called *polynomially bounded growth* studied by Nieberg, Hurink and Kern [19, 24]. Nieberg et al. presented a PTAS for MWIS, MDS and MIDS for these classes of graphs. However, graphs of polynomally bounded growth do not include the intersection graphs of arbitrary disks, arbitrary r-regular polygons, t-interval graphs, etc.

Note that our results include the first polynomial-time approximation algorithms with constant approximation ratio for maximum clique and minimum clique partition on disk graphs and on intersection graphs of r-regular polygons.

We also present a polynomial-time approximation algorithm with constant approximation ratio for dominating set on the intersection graphs of a restricted set of r-regular polygons. Recently, Erlebach and van Leeuwen [9] presented an approximation algorithm with constant approximation ratio for the same problem on an arbitrary set of r-regular polygons but, in contrast to this paper, they do not allow to rotate the polygons. The results in this paper also imply an approximation algorithm with constant approximation ratio for dominating set on intersection graphs of an arbitrary set of *non-rotated* r-regular polygons. We also improve the approximation ratio of maximum clique on t-interval graphs. In general, our results also extend to intersection graphs of a restricted set of t-fat objects and further classes of graphs not discussed in this paper.

2 New Complexity Parameters

In this section, we introduce three new complexity parameters. For each complexity parameter, we present examples of classes of intersection graphs for which the complexity parameter is bounded by a constant. For a set S of vertices in a graph G, let $G[S]$ be the subgraph of G induced by the vertices of S.

Definition 1 (k-perfectly groupable). *A graph is* k-perfectly groupable *if the neighbors of each vertex v can be partitioned into k sets $S_1, \dots, S_k$ such that $G[S_i]$ is a possibly empty clique for each $i \in \{1, \dots, k\}$.*

By definition, we can find, for each object S of a k-restricted set $\mathcal{C}$ of fat objects, a set $P(S)$ of k points such that every object in $\mathcal{C}$ intersecting S (and hence also a smallest ball containing the closure of S) covers at least one point in $P(S)$. For each S-intersecting object $S' \in \mathcal{C}$, choose one of the points in $P(S)$ as a representative. Then all S-intersecting objects having the same representative in $P(S)$ induce a clique in the intersection graph. Hence, the intersection graph of a k-restricted set of fat objects is k-perfectly groupable. Note also that unit disk graphs and unit square graphs are k-groupable for a suitable constant k. Graphs of maximum degree k are also k-groupable.

Definition 2 (k-perfectly eliminable, k-perfect elimination order, successor). *A graph G is* k-perfectly eliminable *if there is an order $v_1, \dots, v_n$ of the vertices of G such that, for each vertex v_i $(1 \le i \le n)$, the subset of neighbors of v_i contained in $\{v_j \mid j > i\}$ can be partitioned into k sets $S_1, \dots, S_k$ such that $G[S_j]$ is a possibly empty clique for each $j \in \{1, \dots, k\}$. The vertices in $\{v_j \mid j > i, \{v_i, v_j\} \in E(G)\}$ are called the* successors *of v_i and the order above of the vertices in G is called a* k-perfect elimination order.

Let $\mathcal{C}$ be a set of fat objects $S_1, \dots, S_n$ ordered by non-decreasing size. Let k be the fatness of $\mathcal{C}$. Then, for each object S_i, we can find k points such that every S_i-intersecting object in $\{S_{i+1}, \dots, S_n\}$ contains one of the k points. If, for $i \in \{1, \dots, n\}$, we define v_i to be the vertex representing S_i in the intersection graph G of $\mathcal{C}$, then $v_1, \dots, v_n$ defines a k-perfect elimination order. Therefore,

G is k-perfectly eliminable. Also note that disk graphs and square graphs are k-perfectly eliminable for a suitable constant k. Moreover, chordal graphs are exactly the 1-perfectly eliminable graphs.

Definition 3 (k-perfectly orientable). *A graph G is called* k-perfectly orientable *if each edge $\{u, v\}$ of G can be assigned to exactly one of its endpoints u and v such that, for each vertex w, the vertices connected to w by edges assigned to w can be partitioned into k sets $S_1, \ldots, S_k$ such that $G[S_i]$ is a possibly empty clique for each $i \in \{1, \ldots, k\}$. We write $a(\{u, v\}) = u$ if $\{u, v\}$ is assigned to u.*

We now show that the intersection graph $G = (V, E)$ of a set of t-fat-objects $\mathcal{C}$ with a universe of fatness c is $(t{\cdot}c)$-perfectly orientable. Let $V = \{v_1, \ldots, v_n\}$ and, for each $i \in \{1, \ldots, n\}$, let $\mathcal{S}_i$ be the union of t objects $S_{i,1}, \ldots, S_{i,t}$ represented by v_i. Choose, for each edge $\{v_i, v_j\}$ with $i < j$, a pair $\{k, l\}$ of indices such that $S_{i,k}$ and $S_{j,l}$ intersect. Assign $\{v_i, v_j\}$ to v_i if the size of $S_{i,k}$ is smaller than the size of $S_{j,k}$ and to v_j otherwise. Then, for each vertex v_i one can find $t \cdot c$ points such that each $\mathcal{S}_i$-intersecting t-fat-object $\mathcal{S}_j$ with $\{v_i, v_j\}$ being assigned to v_i must intersect S_i in at least one of the $t \cdot c$ points. Therefore, the set of vertices being endpoints of edges assigned to v_i can be partitioned into $\leq t \cdot c$ cliques. This proves that G is $(t \cdot c)$-perfectly orientable. Note also that the intersection graphs of t-intervals are $2t$-orientable. For these graphs, an edge $\{v_i, v_j\}$ with $i < j$ is assigned to v_i if the t-interval represented by v_j intersects one of $2t$ endpoints of the intervals whose union is represented by v_i.

We next present explicit upper bounds for our complexity parameters on some special intersection graphs. Before that let us define the *inball* and the *outball* of a geometric object S to be a ball with largest radius contained in the closure of S and the ball with smallest radius containing the closure of S, respectively. The *center* of S is the center of its outball. Due to space limitations the remaining proofs of this section are only part of the appendix.

Lemma 4. *An intersection graph of t-squares, i.e., of unions of t (not necessarily axis-parallel) squares, is*

a) 10-perfectly groupable if $t = 1$ and if the squares are of unit size,
b) 10-perfectly eliminable if $t = 1$, and c) $10t$-perfectly orientable.

Lemma 5. *Let r be a fixed constant and G be an intersection graph, where each vertex represents a union of t polygons taken from a universe of non-rotated r-regular polygons. Then G is $(t \cdot r)$-perfectly orientable.*

Lemma 6. *Let G be the intersection graph of some geometric objects. If the objects are convex and if, additionally, there is a constant k such that, for each object, the ratio between its size and the radius of its inball is bounded by k, then G is $(\frac{3}{2}\sqrt{d\pi}(k+1))^d/\Gamma(d/2+1)$-perfectly eliminable. If the ratio between the largest size of the objects and the radius of a smallest inball of the objects is bounded by a constant k', G is $(\frac{3}{2}\sqrt{d\pi}(k'+1))^d/\Gamma(d/2+1)$-perfectly groupable (even in the case of non-convex objects).*

3 Relations between the New Complexity Parameters

In the following we study the relations of our new complexity parameters to each other and to the so-called treewidth.

Observation 7 *k-perfectly groupable graphs are k-perfectly eliminable since any ordering of the vertices defines a k-perfect elimination order. Conversely, an n-vertex star, i.e., an n-vertex tree with $n-1$ leaves, is not k-perfectly groupable for all $k < n-1$, but it is 1-perfectly eliminable.*

Lemma 8. *A k-perfectly eliminable graph is also k-perfectly orientable, but for every $n \in \mathbb{N}$ with $n \geq 9$, there exists a 2-perfectly orientable graph with n vertices that is not f-perfectly eliminable for all $f < \lfloor \sqrt{n}/3 \rfloor$.*

Proof. Let G be a k-perfectly eliminable graph with a k-perfect elimination order $v_1, \ldots, v_n$. If all edges incident to a vertex v and one of its successors are assigned to v, the endpoints $\neq v$ of the edges assigned to v can be partitioned into k sets $S_1, \ldots, S_k$ such that $G[S_i]$ is a possibly empty clique for every $i \in \{1, \ldots, k\}$. In other words, G is k-perfectly orientable.

Let us choose an arbitrary $n \in \mathbb{N}$ with $n \geq 9$ and let $k = \lfloor \sqrt{n}/3 \rfloor$. We now construct a 2-perfectly orientable graph G with n vertices that is not f-perfectly eliminable for any $f < k$. The vertices of this graph are divided into 3 disjoint sets S_0, S_1 and S_2 of size k^2 and, if $n - 3k^2 > 0$, a further set $R = V_G \setminus (S_1 \cup S_2 \cup S_3)$ of isolated vertices. Each set S_i ($i \in \{0, 1, 2\}$) is divided into k subsets $S_{i,1}, \ldots, S_{i,k}$ of size k. For each $i \in \{0, 1, 2\}$ and each $j \in \{1, \ldots, k\}$, we introduce edges between each pair of vertices contained in the same subset $S_{i,j}$ and assign each of these edges arbitrarily to one of its endpoints. Let us define a numbering on the vertices of $S_{i,j}$ such that we can refer to the l-th vertex of $S_{i,j}$. For each $i \in \{0, 1, 2\}$ and each $j, l \in \{1, \ldots, k\}$, we additionally introduce edges between the l-th vertex u of $S_{i,j}$ and all vertices of $S_{(i+1) \bmod 3, l}$. We assign them to u. The constructed graph G is 2-perfectly orientable since the endpoints of an edge assigned to a vertex u being the l-th vertex of a subset $S_{i,j}$ belong to one of the two cliques induced by the vertices of $S_{i,j}$ and $S_{(i+1) \bmod 3, l}$. However, u is also adjacent to k vertices in $S_{(i-1) \bmod 3}$. Since there is no edge between a vertex in $S_{(i-1) \bmod 3, j_1}$ and a vertex in $S_{(i-1) \bmod 3, j_2}$ for $j_1 \neq j_2$, G cannot be f-perfectly eliminable for any $f < k$. □

Definition 9 (tree decomposition, bag, (tree)width). *A* tree decomposition *for a graph $G = (V, E)$ is a pair (T, B), where $T = (V_T, E_T)$ is a tree and B is a mapping that maps each node w of T to a subset of V—called the* bag *of w—such that*

1. *$\cup_{w \in V_T} G[B(w)] = G$, and*
2. *$B(x) \cap B(y) \subseteq B(w)$ for all $w \in V_T$ on the path from $x \in V_T$ to $y \in V_T$ in T.*

The width *of (T, B) is $\max_{w \in V_T} \{|B(w)| - 1\}$. The* treewidth of a graph *is the width of a tree decomposition for the graph having smallest width.*

Lemma 10. *All graphs of treewidth k are k-perfectly eliminable and therefore also k-perfectly orientable.*

Proof. Let $G = (V, E)$ be a graph and (T, B) a tree decomposition of G of width k. Note that w.l.o.g. we can find a leaf of T whose bag contains a vertex v that is not contained in any other bag of (T, B). Otherwise, delete a leaf of T as long as each vertex that is contained in a bag of a leave occurs in two or more bags of (T, B). Consequently, v has at most k neighbors. We can choose v as the first vertex of a k-perfect elimination order and recursively determine a k-perfect elimination order on $G[V \setminus \{v\}]$. $\square$

Observation 11 *The n-vertex clique is an example for an 1-perfectly groupable graph G that does not have treewidth $n-2$. Conversely, the n-vertex star is a graph of treewidth 1 that is not $(n-2)$-perfectly groupable.*

4 Recognition Problems

In this section we show that for a given graph it is $\mathcal{NP}$-hard to decide, whether one of our new complexity parameters is bounded by a constant. All results are obtained by a reduction from the $\mathcal{NP}$-hardness of minimum clique partition.

Lemma 12. *Given a constant k and a graph G, it is $\mathcal{NP}$-hard to decide whether G is k-perfectly groupable.*

Proof. Given a graph G as instance of the minimum clique partition problem, we replace $G = (V, E)$ by $G' = (V \cup \{v\}, E \cup \{\{v, w\} \mid w \in V\})$ for a new vertex $v \notin V$. If G' is k-perfectly groupable, the neighbors of v can be partitioned into k possibly empty cliques. This means G has a clique partition of size k. If G has a clique partition of size k, the neighbors of v in G' can be partitioned into k possibly empty cliques. The same is true for each other vertex w of G' if we cover the neighbors of w by the k cliques covering G and add v to one of these cliques. Hence, G is k-perfectly groupable. $\square$

Lemma 13. *Given a constant k and a graph G, it is $\mathcal{NP}$-hard to decide whether G is k-perfectly eliminable.*

Proof. Given a graph G as instance of the minimum clique partition problem, we construct a graph G' on which we want to find a k-perfect elimination order. G' is obtained from G by adding $k+1$ new vertices to G and connecting each new vertex to each vertex of G. If G has a clique-partition of size k, construct an ordering of the vertices of G' beginning with the $k+1$ new vertices. Then all successors are vertices in G and hence can be covered by k possibly empty cliques. Therefore G' is k-perfectly eliminable.

In the reverse direction a k-perfect elimination order $v_1, \ldots, v_n$ cannot start with a vertex of G since it is adjacent to all new vertices. Thus, the successors of v_1 contain all vertices of G and G must have a clique partition of size k. $\square$

Lemma 14. *Given a constant k and a graph G, it is $\mathcal{NP}$-hard to decide whether G is k-perfectly orientable.*

Proof. Given an n-vertex graph $G = (V, E)$ as instance of the minimum clique partition problem, we add a set V' of $nk + 1$ new vertices to G and connect each new vertex to each vertex in V. Let G' be the resulting graph. If G has a clique partition of size at most k, assign all incident edges of a vertex $v' \in V'$ to v' and edges $e \in E$ to an arbitrary endpoint of e. Then the endpoints of edges assigned to a vertex v induce k possibly empty cliques, i.e., G' is k-perfectly orientable.

Conversely, let us assume that G' is k-perfectly orientable and let a be a suitable assignment of the edges to their endpoints. For each vertex $v \in V$ at most k of the $nk + 1$ new edges incident to v can be assigned to v since there are no edges between two vertices of V'. Thus, there is at least one $v' \in V'$ with all its edges assigned to itself. Thus, G must have a clique partition of size $\leq k$. $\square$

5 Algorithms

We present polynomial time approximation algorithms for several $\mathcal{NP}$-hard problems on graph classes with one of our complexity parameters bounded by a constant. We always implicitly assume that we are given an explicit *representation* of a graph as a k-perfectly groupable, eliminable, or orientable graph G. By that we mean that we are given, for each vertex v, a partition of its neighbors, of its successors, and of the vertices connected to v by edges assigned to v, respectively, into k sets $S_1, \ldots, S_k$ such that $G[S_i]$ is a possibly empty clique for all $i \in \{1, \ldots, k\}$; in the case of a k-perfectly eliminable graph, additionally, a k-perfect elimination order, and in the case of a k-perfectly orientable graph, additionally, for each vertex the edges assigned to it. These representations are sufficient even for intersection graphs. We do not need the explicit representations as intersection graphs described in Section 1 but we can use them to construct our new representations in polynomial time.

Lemma 15. *On k-perfectly groupable graphs minimum dominating set and minimum independent dominating set can be k-approximated in polynomial time.*

Proof. As a k-approximative solution on a k-perfectly groupable graph G we output a maximal—not necessarily maximum—independent set S of G. To prove correctness, let us consider a minimum (independent) dominating set S_{opt} of G. For all $v \in S \setminus S_{\text{opt}}$, there must be a neighbor of v in S_{opt}. However, each such neighbor cannot cover more than k vertices of S, since G is k-perfectly groupable. Consequently, S is an independent dominating set of size at most $k|S_{\text{opt}}|$. $\square$

Lemma 16. *Maximum weighted independent set, maximum weighted clique, minimum vertex coloring, and minimum clique partition are k-approximable on k-perfectly eliminable and on k-perfectly groupable graphs in polynomial time.*

Proof (Maximum Weighted Independent Set). A k-approximative solution on a k-perfectly eliminable graph $G = (V, E)$ with weight function $w : V \to \mathbb{R}$ can be computed as follows: First remove all vertices with weights ≤ 0 since they can be excluded from a solution. If the remaining graph G' is the empty graph, return the empty set. Otherwise, choose the vertex v that among the remaining vertices appears first in the k-perfect elimination order of G. Decrease the weight of v and its neighbors by $w(v)$. Compute an independent set I for the graph G' with the new weight function w' recursively. If a neighbor of v is in I, return I and, otherwise, return $I \cup \{v\}$. In both cases, if we reincrease the weights of v and its neighbors, the weight of our solution increases by $w(v)$. Note that the weight of any other independent set can increase by at most $k \cdot w(v)$ since all neighbors of v in G' are successors of v in G and thus they can be partitioned into $\leq k$ cliques. Thus, the difference between the weights of optimal solutions for G with weight function w and for G' with weight function w' is bounded by $k \cdot w(v)$. Since this is true for all recursive steps, the solution obtained has approximation ratio k. The algorithm terminates since each recursion sets the weight of ≥ 1 vertices to 0. Also note that a representation of G' as a k-perfectly eliminable graph can be computed in polynomial time from such a representation of G. $\square$

Proof (Maximum Weighted Clique). Given a k-perfectly eliminable graph, choose, for each node v, a clique C_v of maximal weight among the cliques obtained from one of the k possibly empty cliques induced by the successors of v by adding v to these cliques. Return the clique with maximal weight among the cliques in $\{C_v \mid v \in V\}$. This solution has approximation ratio k since a maximum weighted clique C_{opt} must also contain a vertex v with C_{opt} consisting exclusively of v and a subset of its successors. $\square$

Proof (Minimum Vertex Coloring). Let $v_1, \ldots, v_n$ be a k-perfect elimination order of the given graph. Let, for each vertex v, $\text{Succ}(v)$ be the set of successors of v. Iterate over the vertices in reverse k-perfect elimination order. Color a vertex v with the lowest number in $\{1, \ldots, n\}$ different from the numbers already assigned to its neighbors. Then we obtain a coloring with $\max_v\{1 + |\text{Succ}(v)|\}$ numbers. Note that $\max_v\{1 + |\text{Succ}(v)|/k\}$ are necessary, since there is a clique of this size induced by a vertex and a subset of its successors. $\square$

Proof (Minimum Clique Partition). Given a graph G and a k-perfect elimination order $v_1, \ldots, v_n$ for G, we first compute the graph G' obtained by removing v_1 and its neighbors from G. We then solve the problem recursively on G'. The graph induced by the removed vertices can be partitioned into a set S of $\leq k$ cliques. We output S together with the solution obtained for G' as a solution for G. Note that v_1 is not incident to any vertex of G', i.e., all vertices of v_1-containing cliques are removed. This guarantees that the difference between the clique partition numbers of G and G' is ≥ 1. Therefore, we obtain a clique partition that uses at most k times as many cliques as an optimal partition. $\square$

Lemma 17. *On k-perfectly orientable n-vertex graphs, there are polynomial-time algorithms with approximation ratio*

1. *$2k$ for maximum weighted independent set, minimum vertex coloring and maximum weighted clique.*

2. $O(\log^2 n/\log(1+1/k))$ *for minimum clique partition.*

For the following proofs let $G=(V,E)$ be a k-perfectly orientable n-vertex graph and, for each $u \in V$, let $D_{u,1}=(V_{u,1},E_{u,1}),\ldots,D_{u,k}=(V_{u,k},E_{u,k})$ be k possibly empty cliques with pairwise disjoint vertex sets and with $V_{u,1}\cup\ldots\cup V_{u,k}$ being the set of vertices connected to u by edges assigned to u. Moreover, for $i \in \{1,\ldots,k\}$, let $C_{u,i}=(V'_{u,i},E'_{u,i})$ be the clique $G[V_{u,i}\cup\{u\}]$ and $\mathcal{C}=\{C_{u,i}\,|\,u \in V, 1 \le i \le k\}$.

Proof (Maximum Weighted Independent Set). Our proof follows the ideas in [1]. W.l.o.g. G has no vertex with weight ≤ 0. Otherwise remove such vertices and their adjacent edges. Assume $V=\{1,\ldots,n\}$ and define $w(i)$ $(1 \le i \le n)$ to be the weight of vertex i. Let w be the vector $(w(1),\ldots,w(n))$ and $x=(x_1,\ldots,x_n)^{\mathrm{T}}$ be an optimal solution of the following integer linear program (P). Then $S=\{i\,|x_i=1\}$ is a solution for the maximum weighted independent set problem on G.

$$\begin{array}{rll} (P): \ \max & wx & \\ \textbf{s.t.} & \sum_{v\in C} x_v \le 1 & \forall C \in \mathcal{C} \\ & x_v \in \{0,1\} & \forall v \in V \end{array}$$

We next want to consider the relaxation (P') of (P) obtained by replacing the constraint $x_v \in \{0,1\}$ by $x_v \ge 0$ for all $v \in V$. Let x be an optimal solution of (P') and, for each vertex $v \in V$, define $N[v]$ to be the set consisting of v and its neighbors. We now show that there is a vertex $v \in V$ with $\sum_{u\in N[v]} x_u \le 2k$. Recall that $a(\{v,w\})=v$ means that $\{v,w\}$ is assigned to v. By the first constraint of (P') we have

$$\forall v \in V: k\,x_v + \sum_{\substack{\{v,u\}\in E,\\ a(\{v,u\})=v}} x_u \le k \quad (*) \Rightarrow nk \ge \sum_{v\in V}\sum_{\substack{\{v,u\}\in E,\\ a(\{v,u\})=v}} x_u = \sum_{v\in V}\sum_{\substack{\{v,u\}\in E,\\ a(\{v,u\})=u}} x_u$$

$$\Rightarrow \exists v: \sum_{\substack{\{v,u\}\in E,\\ a(\{v,u\})=u}} x_u \le k \overset{(*)}{\Rightarrow} \exists v: \sum_{u\in N[v]} x_u \le 2k.$$

A $2k$-approximative solution for the MWIS on G is computed as follows. Find a solution x for the linear program (P') [26] and a vertex $\tilde{v}$ with $\sum_{u\in N[\tilde{v}]} x_u \le 2k$. Then, decrease the weight of $\tilde{v}$ and its neighbors by $w(\tilde{v})$ and find recursively an independent set S' of approximation ratio $2k$ for G with the decreased weight function denoted by w'. Set $S=S'\cup\{\tilde{v}\}$ if $S'\cap N[\tilde{v}]=\emptyset$ and $S=S'$, otherwise. Return S. For analyzing the approximation ratio, let OPT and OPT' be optimal solutions of (P') with respect to w and w', respectively. Since $w(OPT) \le w'(OPT)+2kw(\tilde{v}) \le w'(OPT')+2kw(\tilde{v})$ and since $w(S) \ge w'(S')+w(\tilde{v})$, we can conclude that the vector x with $x_v=1$ for all $v \in S$ and $x_v=0$ for all $v \in V\setminus S$ is a $2k$-approximative solution of (P') and hence also for (P). □

Proof (Minimum Vertex Coloring). Construct an order $v_1,\ldots,v_n$ of the vertices such that, for each vertex v_i $(i \in \{1,\ldots,n\})$, at least half of the edges in

$G[\{v_i, \ldots, v_n\}]$ being adjacent to v_i are assigned to v_i. We now want to color the vertices $v_n, \ldots, v_1$ in this order with numbers in $\{1, \ldots, n\}$. We color each vertex $v \in V$ with the smallest number different from the colors of all already colored neighbors of v. Define C_v as in the part of the proof of Lemma 16 for maximum weighted clique. Then, a vertex v obtains a color smaller or equal $2k(|C_v| - 1) + 1$, whereas an optimal coloring must color v and its neighbors with at least $|C_v|$ colors. Thus, the coloring obtained is a $2k$-approximation.

Proof (Maximum Weighted Clique). As $2k$-approximative solution, return the clique $C \in \mathcal{C}$ of maximal weight. Let us compare the weight of C with the weight of a maximal clique C_{OPT} of G. The subgraph of G induced by the vertices of C_{OPT} contains at least one vertex u for which the sum of the weights of the neighbors not being endpoints of edges assigned to u does not exceed the sum of the weights of the neighbors being endpoints of edges assigned to u. Thus, the weight of C is at most a factor $2k$ smaller than the weight of C_{OPT}. □

Proof (Minimum Clique Partition). As part of our computations, we want to find in polynomial time a minimal number of cliques in $\mathcal{C}$ such that the union of their vertex sets is V. Unfortunately, this is an instance of the $\mathcal{NP}$-hard set cover problem. However, we can find a subset of the cliques in $\mathcal{C}$ that covers V and that is at most a factor $O(\log |V|)$ larger than the minimal number of cliques necessary [21]. We return this subset as an approximative solution. We achieve approximation ratio $O(\log^2 |V| / \log \frac{2k}{2k-1}) = O(\log^2 |V| / \log(1 + \frac{1}{k}))$ since there is a clique partition of V using only cliques in $\mathcal{C}$ that uses $O(\log |V| / \log \frac{2k}{2k-1})$ as many cliques as a minimum clique partition C_{OPT} of $q \leq n$ arbitrary cliques $C_1, \ldots, C_q$ of G: Choose a vertex v of C_1 such that in the subgraph of G induced by the vertices of C_1 at least half of the edges adjacent to v are assigned to v. Remove v and the clique among $D_{v,1}, \ldots, D_{v,k}$ containing the largest number of not already deleted vertices in C_1. This reduces the number of vertices of C_1 by a factor of at least $\frac{1}{2k}$. Repeat this step recursively until, after $O(\log |V| / \log \frac{2k}{2k-1})$ steps, C_1 contains no vertex anymore. More precisely, when choosing a vertex v for which at least half of the adjacent edges are assigned to v, only count the edges not already being deleted. If we do the same for the remaining cliques, we obtain a clique partition with $O(q \log |V| / \log \frac{2k}{2k-1})$ cliques. □

References

1. R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira, Scheduling split intervals *SIAM J. Comput.* **36** (2006), 1–15.
2. H. L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, *SIAM J. Comput.* **25** (1996), 1305–1317.
3. A. Butman, D. Hermelin, M. Lewenstein, and D. Rawitz, Optimization problems in multiple-interval graphs, Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), 268–277.
4. M. R. Cerioli, L. Faria, T. O. Ferreira, and F. Protti, On minimum clique partition and maximum independent set on unit disk graphs and penny graphs: complexity and approximation, *Electronic Notes in Discrete Mathematics* **18** (2004), 73–79.

5. T. M. Chan, Polynomial-time approximation schemes for packing and piercing fat objects, *J. Algorithms* **46** (2003), 178–189.
6. B. N. Clark, C. J. Colbourn, and D. S. Johnson, Unit disk graphs, *Discrete Math.* **86** (1990), 165–177.
7. B. Courcelle, The monadic second-order logic of graphs. I. Recognizable sets of finite graphs, *Inform. and Comput.* **85** (1990), 12–75.
8. T. Erlebach, K. Jansen, and E. Seidel, Polynomial-time approximation schemes for geometric intersection graphs, Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2001), 671–679.
9. T. Erlebach and E. J. van Leeuwen, Domination in geometric intersection graphs, LATIN 2008, Lecture Notes in Computer Science, Vol. 4957, Springer, Berlin, 2008, 747–758.
10. R. J. Fowler, M. S. Paterson, and S. L. Tanimoto, Optimal packing and covering in the plane are NP-complete, *Inform. Process. Lett.* **12** (1981), 133–137.
11. A. Frank, Some polynomial algorithms for certain graphs and hypergraphs, Proc. 5th British Combinatorial Conference (Aberdeen 1975), Congr. Numer. XV (1976), 211-226.
12. M. R. Garey, D. S. Johnson, and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoret. Comput. Sci.* **1** (1976), 237–267.
13. F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, *SIAM J. Comput.* **1** (1972), 180–187.
14. M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
15. M. C. Golumbic, Algorithmic aspects of intersection graphs and representation hypergraphs, *Graphs and Combinatorics* **4** (1988), 307–321.
16. A. Gräf, Coloring and recognizing special graph classes, Technical Report Musikinformatik und Medientechnik Bericht 20/95, Johannes Gutenberg-Universität Mainz, 1995.
17. J. R. Griggs and D. B. West, Extremal values of the interval number of a graph, *SIAM Journal on algebraic and discrete methods* **1** (1980), 1–7.
18. H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns, NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs, *J. Algorithms* **26** (1998), 238–274.
19. J. L. Hurink amd T. Nieberg, Approximating minimum independent dominating sets in wireless networks, *Inform. Process. Lett.* **109** (2008), 155–160.
20. H. Imai and T. Asano, Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane, *J. Algorithms* **4** (1983), 310–323.
21. D. . S. Johnson, Approximation algorithms for combinatorial problems, *J. Comput. System Sci.* **9**, 256–278.
22. E. Malesińska, Graph-theoretical models for frequency assignment problems, PhD thesis, University of Berlin, 1997.
23. M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz, Simple heuristics for unit disk graphs, *Networks* **25** (1995), 59-68.
24. T. Nieberg, J. Hurink, and W. Kern, Approximation Schemes for Wireless Networks, ACM Transactions on Algorithms **4** (2008), Article No. 49.
25. C. H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, *J. Comput. System Sci.* **43** (1991), 425–440.
26. É. Tardos, A strongly polynomial algorithm to solve combinatorial linear programs, *Operations Research* **34** (1986), 250–256.

Appendix

Proof of Lemma 4. For proving case a) and b), let G be the intersection graph of a set S of squares. It remains to show that, for a square Q of minimal side length ℓ, there are 10 points—called the *barriers of* Q—such that every Q-intersecting square Q' of length $\geq \ell$ must cover at least one of them. This fact also proves case c) since the universe of a set of t-squares then must have fatness 10.

We first describe our choice of the 10 barriers of Q. See also the left side of Fig. 1 for the following construction. Let b_1 and b_2 be the two perpendicular bisectors of the sides of Q. Choose two barriers x and y of Q as points on b_1 such that the part of b_1 inside Q is divided into three parts of equal length. We call these points the *inner barriers* of Q. Let C be the curve surrounding Q that consists of all points having distance exactly ℓ to one of the inner barriers and distance at least ℓ to the other inner barrier. The remaining 8 barriers, called *outer barriers*, are almost equidistant points on C. More exactly, 4 outer barriers of Q are placed on the $2+2$ intersection points of C with b_1 and b_2. Choosing the other 4 outer barriers of Q is more sophisticated. Let x' and y' be the two points on b_1 having the same distance to the center of Q as to x and y, respectively. In addition, let $r_1, \ldots, r_4$ be the 4 rays starting from x' and y', respectively, and intersecting a corner of Q but neither b_1 nor b_2. The four remaining outer barriers are placed on the intersection points of C with the rays $r_1, \ldots, r_4$.

By a simple mathematical analysis one can show that the distance between any two consecutive barriers on C is strictly smaller than ℓ. It remains to show that each square of side length at least ℓ intersecting Q also covers one of the barriers of Q. Assume for a contradiction that we can find a square Q' of side length at least ℓ intersecting Q but none of the barriers of Q. W.l.o.g. we can assume that Q' has side length exactly ℓ since otherwise Q' also contains a smaller square intersecting Q. Let $\mathcal{H}$ be the convex hull of the outer barriers and let B be the largest circle contained in $\mathcal{H}$ and having the same center as Q. B, and thus also $\mathcal{H}$, contains at least one corner of Q' since Q' intersects Q and B, and since a simple mathematical analysis shows that each chord of B with length at most l does not intersect Q. We now distinguish two cases.

Case 1: No side of Q' is completely contained in the convex hull $\mathcal{H}$ of the outer barriers. For each pair of consecutive outer barriers p and q on C, let us define $C_{p,q}$ to be the semi-circle inside H with endpoints p, q and hence having a diameter equal to the distance between p and q. See again the left side of Fig. 1. Let z be the corner of Q' inside B with the smallest distance to a point in Q. Note that the two sides of Q' ending in z are not completely contained in $\mathcal{H}$. Consequently, by Thales' theorem and by Q' not containing any barriers there must be two consecutive outer barriers p and q on C such that z is contained in the face enclosed by $C_{p,q}$ and $\overline{pq}$. Again a simple mathematical analysis shows that none of our semi-circles intersects Q. Thus, neither z nor any other point of Q' is covered by Q. Contradiction.

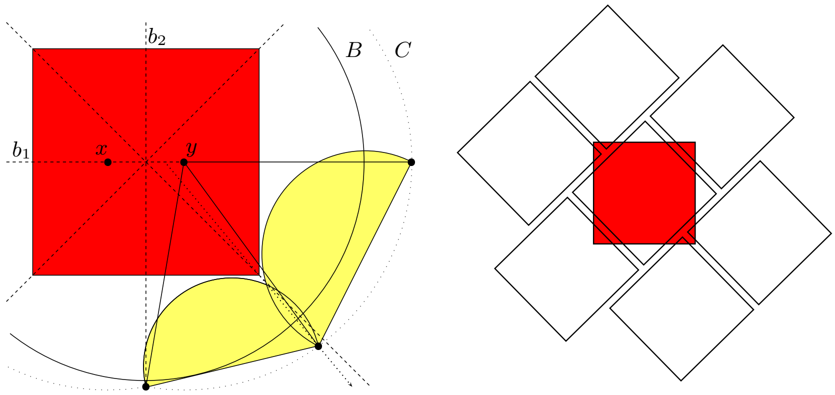

Fig. 1. Left: A square with some barriers. **Right:** A square intersects 7 disjoint squares.

Case 2: At least one side of Q' is completely contained in $\mathcal{H}$. Since each pair of consecutive outer barriers on C have a distance smaller than ℓ, the center q of Q' is inside $\mathcal{H}$.

By symmetry, w.l.o.g. we can assume that the distance between q and y is smaller or equal than the distance between q and x. Let $\mathcal{H}'$ be the convex hull of x and the outer barriers having distance $\leq \ell$ to y. On the one hand, for each pair of consecutive barriers q_1 and q_2 on $\mathcal{H}'$, there is at most one corner in the face bounded by $\overline{q_1 q_2}$ and the semi-circle outside $\mathcal{H}'$ with endpoints q_1 and q_2. On the other hand, at least one corner of Q' is outside $\mathcal{H}'$ since the inball of Q', which does not contain y, must intersect the border of $\mathcal{H}'$. Consequently, there are two sides s_1 and s_2 of Q' that have a common corner p outside $\mathcal{H}'$ and that intersect $\mathcal{H}'$ between to outer barriers, say q_1 and q_2.

Let T be the triangle with corners y, q_1 and q_2. Since Q' is a square of side length ℓ, since p is not covered by T and since T is a triangle with two sides of length ℓ and with an s_1-intersecting side of length $\leq \ell$, y has to be inside Q'. Contradiction. □

Observation 18 *Some square graphs are not* 6*-perfectly groupable as shown on the right side of Fig. 1.*

Lemma 19. *The intersection graph of a set of rectangles, all having aspect ratio of α, is $10\lceil\alpha\rceil$-perfectly eliminable.*

Proof. Consider each rectangle as a set of $\lceil\alpha\rceil$ squares. For each rectangle r_1 replaced by squares of a size s_1 one can find $10\lceil\alpha\rceil$ points such that every r_1-intersecting square of size $s_2 \geq s_1$ replacing another rectangle r_2 must cover one of this points. Here we use the fact that the at most $\lceil\alpha\rceil$ squares replacing a single rectangle can all be chosen of the same size. □

Proof of Lemma 5. The intersection of two intersecting non-rotated r-regular polygons must contain at least one of the corners of the two polygons. Let

$V = \{v_1, \ldots, v_n\}$. We assign an edge $\{v_i, v_j\}$ in G with $i < j$ to v_i, if and only if one of the polygons in the union of polygons represented by v_i has a corner contained in the union of polygons represented by v_j. Otherwise, we assign it to v_j. The edges assigned to a vertex v can be partitioned into $t \cdot r$ sets such that the endpoints of the edges of each set induce a clique in G. More precisely, we have one clique for each corner of the t polygons. $\square$

Proof of Lemma 6. For proving the lemma we first show how to find, for a given ball B with radius $\leq R'$ and a real number $r > 0$, a set of points such that every ball b with radius at least r intersecting B must cover at least one of these points. Therefore, let us consider the d-dimensional space, paved with d-dimensional cubes of edge length $s = 2r/\sqrt{d}$ and volume $s^d = 2^d r^d d^{-\frac{d}{2}}$. Then, every ball b of radius at least r must contain at least one of their midpoints, as the cubes' diagonals have length $2r$. Furthermore, the distance between the center of a ball b of radius $\geq r$ intersecting B and B's center is at most $R' + r$. Hence it suffices to pave a ball of radius $R' + 2r$. To do this, we do not need more cubes than completely fit in a ball of radius $R' + 3r$. A ball of radius $R' + 3r$ has volume $(\sqrt{\pi}(R' + 3r))^d / \Gamma(\frac{d}{2} + 1)$ and hence

$$\left\lfloor \frac{(\sqrt{\pi}(R' + 3r))^d}{\Gamma(\frac{d}{2} + 1)} \cdot \frac{1}{2^d r^d d^{-\frac{d}{2}}} \right\rfloor = \left\lfloor \left(\frac{\sqrt{d\pi}}{2} \left(\frac{R'}{r} + 3 \right) \right)^d / \Gamma\left(\frac{d}{2} + 1 \right) \right\rfloor$$

cubes are enough.

Let $\mathcal{S}$ be a set of geometric objects such that G is the intersection graph of $\mathcal{S}$. We first consider the case, where all objects are convex and where there is a k such that, for each object, the ratio between its size and the radius of its inball is bounded by k. Let S_1 be an object of $\mathcal{S}$ with smallest size R and let S_2 be an S_1-intersecting object in $\mathcal{S}$ with size $s_2 \geq R$. Choose S_2' as the image of a dilation of S_2 with an arbitrary point $p \in S_1 \cap S_2$ as center and scaling factor $\lambda = R/s_2 > 0$. Then—as S_2 is convex—every point covered by S_2' is also covered by S_2. Furthermore, the inball of S_2' having radius $r \geq R/k$ must be completely contained in the ball of radius $R' := 3R$ around the center of S_1. Now the considerations above imply that S_2'—and hence S_2—must cover the midpoint of at least one cube of edge length $s = 2r/\sqrt{d}$ completely contained in a ball of radius $R' + 3r$. If we number the vertices of G in an order such that the sizes of objects represented by the vertices do not decrease, we obtain a $(\frac{3}{2}\sqrt{d\pi}(k+1))^d / \Gamma(d/2 + 1))$-perfect elimination order proving the claim.

Finally, let us consider the case, where the objects of $\mathcal{S}$ are not necessarily convex, but the ratio between the largest size of the objects and the radius of a smallest inball of the objects is bounded by a constant k'. Consider intersecting geometric objects S_1 (with size R_1) and S_2 (with size R_2 and inball radius r_2) in $\mathcal{S}$. Then the considerations above imply, that the inball of S_2 must completely lie inside the ball of radius $R' := R_1 + 2R_2$ around the center of S_1. With $\frac{R'}{r_2} = \frac{R_1 + 2R_2}{r_2} \leq 3k'$ the second part of the lemma follows immediately. $\square$