The Inapproximability of Domination Problems in Circle Graphs *

Mirela Damian\textsuperscript{a} \hspace{1cm} Sriram V. Pemmaraju\textsuperscript{b}

\textsuperscript{a}Department of Computer Science, Villanova University, Villanova, PA 19085, U.S.A. e-mail: mirela.damian@villanova.edu
\textsuperscript{b}Department of Computer Science, University of Iowa, Iowa City, IA 52242, U.S.A. e-mail: sriram@cs.uiowa.edu

Abstract

A graph $G = (V, E)$ is called a \textit{circle graph} if there is a one-to-one correspondence between vertices in $V$ and a set $C$ of chords in a circle such that two vertices in $V$ are adjacent if and only if the corresponding chords in $C$ intersect. A subset $V'$ of $V$ is a \textit{dominating set} of $G$ if for all $u \in V$ either $u \in V'$ or $u$ has a neighbor in $V'$. In addition, if no two vertices in $V'$ are adjacent, then $V'$ is called an \textit{independent dominating set}. Keil (Discrete Applied Mathematics, 42 (1993), 51-63) shows that the minimum dominating set problem is NP-complete even for circle graphs. He leaves open the complexity of the minimum independent dominating set problem. In this paper we show that the minimum dominating set problem is APX-hard for circle graphs; in other words, there is a constant $\delta > 0$ such that if there is a $(1 + \delta)$-approximation algorithm for minimum dominating set problem on circle graphs, then $P = NP$. We also show that the minimum independent dominating set problem on circle graphs is NP-complete. Furthermore, we show that for any $\varepsilon > 0$, there does not exist an $n^{1-\varepsilon}$-approximation algorithm for the minimum independent dominating set problem on $n$-vertex circle graphs, unless $P = NP$. Several other related domination problems on circle graphs are also shown to be as hard to approximate.

\textit{Key words:} Approximation algorithms, circle graphs, dominating set, independent set, NP-completeness.

* Parts of the minimum independent dominating set hardness result have appeared in the Proceedings of the 10th International Symposium on Algorithms and Computation (ISAAC 99)

Preprint submitted to Elsevier Preprint 27 July 2003
<table>
<thead>
<tr>
<th>Class of graphs</th>
<th>MDS</th>
<th>MIDS</th>
<th>MCDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>circle graphs</td>
<td>NPc [18]</td>
<td>NPc [this paper]</td>
<td>NPc [18]</td>
</tr>
</tbody>
</table>

Fig. 1. Complexity of three domination problems when restricted to different classes of graphs.

1 Introduction

For a graph $G = (V, E)$, a subset $V'$ of $V$ is a dominating set of $G$ if for all $u \in V$ either $u \in V'$ or $u$ has a neighbor in $V'$. In addition, if no two vertices in $V'$ are adjacent, then $V'$ is called an independent dominating set; if the subgraph of $G$ induced by $V'$, denoted $G[V']$, is connected, then $V'$ is called a connected dominating set; if $G[V']$ has no isolated nodes, then $V'$ is called a total dominating set; and if $G[V']$ is a clique, then $V'$ is called a dominating clique. Garey and Johnson [14] mention that problems of finding a minimum cardinality dominating set (MDS), minimum cardinality independent dominating set (MIDS), minimum cardinality connected dominating set (MCDS), minimum cardinality total dominating set (MTDS), and minimum cardinality dominating clique (MDC) are all NP-complete for general graphs.

Restrictions of these problems to various classes of graphs have been studied extensively [15]. Figure 1 shows the computational complexity of three of the problems mentioned above when restricted to different classes of graphs. P indicates the existence of a polynomial time algorithm and NPC indicates that the problem is NP-complete. Some of the references mentioned in the table are to original papers where the corresponding result first appeared, while some are to secondary sources. It should be noted that this list is far from being comprehensive and that these problems have been studied for several other classes of graphs. In this paper we focus on the last row in the above table, corresponding to domination problems on circle graphs.

A graph $G = (V, E)$ is called a circle graph if there is a one-to-one correspon-
dence between vertices in $V$ and a set $C$ of chords in a circle such that two vertices in $V$ are adjacent if and only if the corresponding chords in $C$ intersect. $C$ is called the chord intersection model for $G$. Equivalently, the vertices of a circle graph can be placed in one-to-one correspondence with the elements of a set $I$ of intervals such that two vertices are adjacent if and only if the corresponding intervals overlap, but neither contains the other. $I$ is called the interval model of the corresponding circle graph. Representations of a circle graph as a graph or as a set of chords or as a set of intervals are equivalent via polynomial time transformations. So, without loss of generality, in specifying instances of problems, we assume the availability of the representation that is most convenient.

The complexity of MDS on circle graphs was first mentioned as being unknown in Johnson's NP-completeness column [17]. Little progress was made towards resolving this problem until Elmallah, Stewart, and Culberson defined the class of $k$-polygon graphs [11]. These are the intersection graphs of straight-line chords inside a convex $k$-sided polygon. The class of circle graphs is the (infinite) union of the classes of $k$-polygon graphs for all $k \geq 3$. For fixed $k$, Elmallah et al. were able to provide a polynomial time algorithm for MDS. Finally, Kei [18] resolved the complexity of MDS on circle graphs by showing that it is NP-complete. In the same paper, Kei showed that for circle graphs MCDS and MTDS are also NP-complete, but MDC has a polynomial algorithm. He left the status of MDS for circle graphs open. In this paper we show that MIDS is also NP-complete on circle graphs. We then study the approximability of the four NP-complete dominating set problems on circle graphs and show the results enumerated below.

For any $\alpha > 1$ ($0 \leq \alpha < 1$), an $\alpha$-approximation algorithm for a minimization (maximization) problem is a polynomial time algorithm that guarantees that the ratio of the cost of the solution to the cost of the optimal over all instances of the problem does not exceed (fall below) $\alpha$. The class $APX$ is the class of optimization problems, each of which has an $\alpha$-approximation algorithm for some constant $\alpha$. A polynomial time approximation scheme (PTAS) is a family $F$ of approximation algorithms such that for each $\varepsilon > 0$, there is an algorithm $A_\varepsilon$ in $F$ that is a $(1+\varepsilon)$-approximation algorithm and has running time that is polynomial in the input size. An optimization problem is said to be $APX$-hard if a PTAS for the problem implies that every problem in $APX$ has a PTAS. Just like showing a problem NP-complete means that it is highly unlikely that there exists a polynomial time algorithm for the problem, showing that a problem is $APX$-hard means that it is highly unlikely that there will be a PTAS for the problem.

**Our results:**

1. MDS, MCDS, and MTDS are all $APX$-hard.
(2) MIDS is NP-complete.
(3) For any \( \varepsilon > 0 \), there does not exist an \( n^{1-\varepsilon} \)-approximation algorithm for MIDS on an \( n \)-vertex circle graph, unless \( P = NP \).

In Section 4 we present hardness of approximation results for related problems — for example, we show that MIDS on bipartite graphs has no \( n^{1-\varepsilon} \)-approximation algorithm, unless \( P = NP \).

The APX-hardness results for MDS, MCDS, and MTDS complement the approximation algorithms for these problems, presented in [9]. There we presented a \( (2+\varepsilon) \)-approximation algorithm from MDS and \( (3+\varepsilon) \)-approximation algorithms for MCDS and MTDS. Our hardness results for MIDS, even when restricted to classes of graphs such as circle graphs and bipartite graphs, is extremely hard to approximate. Arora and Lund [1] partition optimization problems into 4 classes depending on the approximation ratio that is hard to achieve for them. Class IV contains the problems that are hardest to approximate — problems such as MAXIMUM CLIQUE and MINIMUM CHROMATIC NUMBER for which the approximation ratios \( n^{1/2-\varepsilon} \) and \( n^{1/5-\varepsilon} \) are hard to achieve for any \( \varepsilon > 0 \). Our third result listed above shows that MIDS on circle graphs belongs to this class.

2 Intractability of MIDS on circle graphs

A first result in this section shows that the problem of finding a minimum independent dominating set on circle graphs is NP-complete. This was one of the open questions in [18]. In a second result we strengthen the NP-completeness proof to show that for any \( \varepsilon > 0 \), there does not exist an \( n^{1-\varepsilon} \)-approximation algorithm for MIDS on an \( n \)-vertex circle graph, unless \( P = NP \). The decision problem formulation of the minimum independent dominating set problem on circle graphs (CMIDS) is as follows.

Minimum Independent Dominating Set for Circle Graphs (CMIDS)

INPUT: An interval representation of a circle graph \( G = (V, E) \) and a positive integer \( k \).

QUESTION: Is there an independent dominating set of \( G \) of size at most \( k \)?

**Theorem 1** CMIDS is NP-complete.

**Proof:** Clearly CMIDS is in NP. The proof that the problem is NP-hard is organized in two parts. In the first part we present a polynomial time reduction from 3SAT to CMIDS. Let \( \phi \) be an arbitrary instance of 3SAT. Let \( X = \{x_1, x_2, \cdots, x_q\} \) be the set of boolean variables and let \( C = \{C_1, C_2, \cdots, C_m\} \) be the set of clauses in \( \phi \). Let \( I \) be the instance of CMIDS produced by the
reduction from $\phi$ and let $G(I)$ be the overlap graph of $I$. In the second part we choose an integer $k$ and show that $G(I)$ has an independent dominating set of size $k$ or less if and only if $\phi$ is satisfiable.

In the following we describe the reduction from 3SAT to CMIDS. Without loss of generality we assume that no two literals involving the same variable appear more than once in a clause. We consider first each clause $C_j$ separately and create six intervals with types $a$, $b$, $f$, $t$, $p$ and $s$ associated with each literal that appears in $C_j$. Including the type $t$ interval in the independent dominating set will correspond to setting the associated literal to true and including the type $f$ interval in the dominating set will correspond to setting the associated literal to false.

For each clause $C_j$ we create three pairs of intervals of types $u$ and $w$. The purpose of each such pair is to dominate the type $a$ and type $b$ intervals associated with two out of the three literals that appear in $C_j$. The key idea of our proof is as follows. Suppose that $C_j$ is satisfied by some assignment of truth values to variables. Then we can construct an independent dominating set that contains a type $t$ interval corresponding to a true literal in $C_j$ and a pair of type $u$, type $w$ intervals associated with $C_j$. The type $t$ interval dominates one pair of type $a$, type $b$ intervals while the other two pairs of type $a$, type $b$ intervals are dominated by the type $u$, type $w$ pair. In this manner all the type $a$ and type $b$ intervals associated with $C_j$ are dominated. Including the type $t$ interval in our independent dominating set will correspond to the clause $C_j$ being satisfied by a literal associated to this interval.

Finally we connect clauses together using type $tt$, $ff$, $tf$ and $ft$ intervals so as to maintain consistency of truth values of literals throughout all clauses. Figure 2 shows all intervals associated with a two clause and four variable formula in which $C_1 = \{x_1, x_2, \bar{x}_3\}$ and $C_2 = \{\bar{x}_1, x_2, \bar{x}_4\}$.

In the following we present our construction in detail. Let us consider an arbitrary clause $C_j$. For each variable $x_i$ that appears in $C_j$ we construct two independent intervals $a^i_j = [y_j + 12i, z_j + 5i]$ and $b^i_j = [y_j + 12i + 1, z_j + 5i]$ where $y_j = 12(q + 2)j$ and $z_j = y_j + 12(q + 1)$. The purpose of the offset $y_j$ is to ensure that the intervals associated with different clauses are disjoint. Note that since $b^i_j \subset a^i_j$, these two intervals do not overlap. Next we construct truth setting intervals $t^i_j = [y_j + 12i - 1, y_j + 12i + 5]$ and $f^i_j = [y_j + 12i + 4, y_j + 12i + 10]$ associated with the variable $x_i$. The intervals $t^i_j$ and $f^i_j$ dominate each other, so at most one of them can appear in any independent dominating set. Furthermore $t^i_j$ dominates both $a^i_j$ and $b^i_j$ and $f^i_j$ does not dominate either. The truth setting intervals will help us in determining the correspondence between a satisfying truth assignment and an independent dominating set of size $k$ or less.
Fig. 2. The intervals constructed from the two clause and four variable formula (z1).
To ensure that exactly one of the intervals $t^j_i$ and $f^j_i$ associated with a literal involving $x_i$ appears in any “small enough” independent dominating set, we associate with $x_i$ two other independent intervals $p^j_i = [y_i + 12i + 3, y_i + 12i + 6]$ and $s^j_i = [y_j + 12i + 2, y_j + 12i + 7]$ adjacent to both $t^j_i$ and $f^j_i$ and to no other interval in $I$. Later we will argue that any “small enough” independent dominating set must contain either of the intervals $t^j_i$ or $f^j_i$ in order to dominate $p^j_i$ and $s^j_i$. Otherwise we would need to include both $p^j_i$ and $s^j_i$ in our independent dominating set and the set would become too large.

We now construct intervals associated to the clause $C_j$. The clause $C_j$ is satisfied if and only if at least one of the three literals involved in $C_j$ is true. This will correspond to at least one of the intervals $t^j_i$ appearing in the independent dominating set. Such an interval will dominate the intervals $a^j_i$ and $b^j_i$ associated to the literal involving $x_i$. In order to dominate the other four type $a_j$ and type $b_j$ intervals associated to the other two literals in $C_j$ we add six more intervals of type $u_i$ and $w_i$ to $I$ as follows. Suppose that $C_j$ consists of three literals corresponding to three variables $x_i$, $x_k$ and $x_l$, with $i < k < l$. We create the intervals $u^j_i = [z_j, z_j + 5i + 3]$, $u^k_j = [z_j + 5k + 1, z_j + 5i + 3]$, $u^j_k = [z_j + 1, z_j + 5k + 3]$, $w^j_i = [z_j + 2, z_j + 5k + 2]$, $w^k_i = [z_j + 5i + 1, z_j + 5i + 2]$ and $w^j_k = [z_j + 5i + 2, z_j + 5k + 1]$, where $z_j$ is as defined above. Note that the two intervals $u^j_i$ and $u^k_j$ are independent and dominate the remaining 4 intervals of type $u_i$ and type $w_i$. Furthermore, $u^j_i$ and $w^j_i$ dominate the following 4 intervals of type $a_j$ and $b_j$: $a^j_i$, $b^j_i$, $a^j_k$, and $b^j_k$. The interval pair $u^j_k$ and $w^j_k$ and the interval pair $u^k_i$ and $w^k_i$ have similar properties.

If we repeat the construction described above for each clause $C_j$, $1 \leq j \leq m$, we obtain $m$ pairwise disjoint subsets of intervals, each subset associated to one clause. A last piece of our construction is a gadget that forces a variable to have the same truth value throughout all clauses. For this we create intervals of type $tt$, $ff$, $tf$ and $ft$ that connect clauses together as follows. For any $j$ and $k$, $1 \leq j < k \leq m$, if $x_i$ appears in two clauses $C_j$ and $C_k$ or if $\overline{x}_i$ appears in both $C_j$ and $C_k$ we create two intervals $tf^j_{jk}$ and $ft^j_{jk}$ such that $tf^j_{jk}$ overlaps $t^j_i$ and $f^k_{i}$ and $ft^j_{jk}$ overlaps $f^j_i$ and $t^k_{i}$. Furthermore, if $x_i$ appears in $C_j$ and $\overline{x}_i$ appears in $C_k$ or if $\overline{x}_i$ appears in $C_j$ and $x_i$ appears in $C_k$, we create two intervals $tt^j_{jk}$ and $ff^j_{jk}$ such that $tt^j_{jk}$ overlaps $t^j_i$ and $t^k_{i}$ and $ff^j_{jk}$ overlaps $f^j_i$ and $f^k_{i}$. We now specify the intervals of type $tt$, $ff$, $tf$, and $ft$. If $x_i$ appears in both $C_j$ and $C_k$ or $\overline{x}_i$ appears in both $C_j$ and $C_k$, then we add $tf^j_{jk} = [y_j + 12i + 1, y_k + 12i + 8]$, $f^j_{tjk} = [y_j + 12i + 9, y_k + 12i]$ to the set $I$. If $x_i$ appears in one of $C_j$ or $C_k$ and $\overline{x}_i$ appears in the other, then we add $tt^j_{jk} = [y_j + 12i + 1, y_k + 12i + 8]$, $f^j_{tjk} = [y_j + 12i + 9, y_k + 12i + 8]$ to the set $I$.

This completes the construction of $I$. We now set $k = 5m$ and prove the following lemma, which completes the proof of the theorem.

**Lemma 2** $G$ has an independent dominating set of size $k$ if and only if $\phi$ is
satisfiable.

**Proof.** Given a satisfying truth assignment $A$ for $\phi$ we show how to construct a set $D$ of size $k$ of independent intervals that dominate all intervals in $I - D$. If $x_i$ is true in $A$ we include in $D$, $t_j^i$ for each clause $C_j$ in which $x_i$ appears and $f_j^i$ for each clause $C_j$ in which $\overline{x}_i$ appears. If $x_i$ is false in $A$ we include in $D$, $f_j^i$ for each clause $C_j$ in which $x_i$ appears and $t_j^i$ for each clause in which $\overline{x}_i$ appears. This ensures that all the type $p^i$ and type $s^i$ intervals have been dominated. Note that so far $3m$ intervals have been included in $D$.

We now show that all type $tf^i$, $ff^i$, $tt^i$, and $ff^i$ intervals are also dominated by the intervals included in $D$ so far. Consider an interval $tf_{jk}^i$. The presence of this interval in $I$ implies one of four possibilities: (a) $x_i$ is true and $x_i$ appears in both $C_j$ and $C_k$, (b) $x_i$ is true and $\overline{x}_i$ appears in both $C_j$ and $C_k$, (c) $x_i$ is false and $x_i$ appears in both $C_j$ and $C_k$, and (d) $x_i$ is false and $\overline{x}_i$ appears in both $C_j$ and $C_k$. If $x_i$ is true and it appears in both $C_j$ and $C_k$, then $t_j^i$ dominates $tf_{jk}^i$. Otherwise if $x_i$ is true and $\overline{x}_i$ appears in both $C_j$ and $C_k$, then $f_k^i$ dominates $tf_{jk}^i$. If $x_i$ is false and it appears in both $C_j$ and $C_k$, then $f_j^i$ dominates $tf_{jk}^i$. Otherwise if $x_i$ is false and $\overline{x}_i$ appears in both $C_j$ and $C_k$, then $t_j^i$ dominates $tf_{jk}^i$. So $tf_{jk}^i$ is dominated by one of the intervals already included in $D$ in each of the four possible cases. In a similar manner it can be shown that the intervals of type $ft^i$, $tt^i$, and $ff^i$ are all dominated by the intervals already included in $D$.

Let us consider a clause $C_j$ that contains three literals corresponding to variables $x_i$, $x_k$ and $x_l$, where $i < k < l$. Since $A$ is a satisfying truth assignment, at least one of the literals involving $x_i$, $x_k$ and $x_l$ must be true. Let us assume without loss of generality that the literal involving $x_i$ is true and so we have included $t_j^i$ in $D$. In this case we add $u_j^l$ and $w_j^l$ to $D$, thus ensuring that all type $a_j$, $b_j$, $u_j$ and $w_j$ intervals have been dominated. So we have added two more intervals associated with the clause $C_j$ to $D$. Repeating this for each clause will add $2m$ intervals to the independent set and will also ensure that all intervals in $G$ are dominated. Thus we have an independent dominating set of size $5m$ as required.

We now establish the other direction of Lemma 2: if there is a dominating independent set $D \subseteq I$ of size $k$, then there is a truth assignment to the variables in $X$ that satisfies $\phi$. Recall that associated with each variable $x_i$ that appears in a clause $C_j$ there is a pair of intervals: $s_j^i$ and $p_j^i$. Since no interval in $I$ is adjacent to two type $p$ intervals or two type $s$ intervals and since only $t_j^i$ and $f_j^i$ are adjacent to $p_j^i$ and $s_j^i$, $D$ must contain, for each $i$ and $j$, either $t_j^i$ or $f_j^i$ or both $s_j^i$ and $p_j^i$. If for each $i$ and $j$, $D$ contains $t_j^i$ or $f_j^i$ (but not both, since $t_j^i$ and $f_j^i$ are adjacent to each other), then exactly these $3m$ type $t$ and type $f$ intervals in $D$ suffice to dominate all the type $p$ and type $s$
intervals. This leaves exactly $2m$ intervals to dominate the rest of intervals in $I$. If for some $i$ and $j$, $D$ contains $p^i_j$ and $s^i_j$ then more than $3m$ intervals in $D$ are necessary to dominate the type $p$ and type $s$ intervals. This leaves fewer than $2m$ intervals to dominate the rest of the intervals in $I$.

There is no interval in $I$ adjacent to type $u$ intervals or type $w$ intervals associated with different clauses. In other words, for any $j \neq k$, no interval in $I$ is adjacent to two intervals of types $u_j$ and $w_k$, or $u_j$ and $w_k$, or $w_j$ and $w_k$. Thus the set of intervals in $D$ which dominate the type $u$ and type $w$ intervals associated with different clauses are pairwise disjoint. Also no interval in $I$ is adjacent to all the six type $u_j$ and type $w_j$ intervals associated with a clause $C_j$. Therefore $D$ must contain at least two intervals per clause in order to dominate all the type $u$ and type $w$ intervals. Combined with the argument in the previous paragraph, this means exactly $3m$ intervals in $D$ dominate the type $p$ and type $s$ intervals and the remaining $2m$ intervals dominate the type $u$ and type $w$ intervals. This further implies that for each variable $x_i$ appearing in a clause $C_j$, either $t^i_j$ or $f^i_j$ (but, not both) belong to $I$.

We first show that for each clause $C_j$, $D$ contains at least one type $t_j$ interval, one type $u_j$ interval, and one type $w_j$ interval. Suppose that the three variables that appear in $C_j$ are $x_i$, $x_k$, and $x_l$, where $i < k < l$. As shown before, $D$ must contain two intervals which dominate all type $u_j$ and type $w_j$ intervals. The only intervals adjacent to a type $u_j$ or a type $w_j$ interval are the type $a_j$, $b_j$, $u_j$ and $w_j$ intervals. No two type $a_j$ intervals or two type $b_j$ intervals can occur in $D$, since they are not independent. Suppose that $D$ contains an interval $a^i_j$. This interval is adjacent to at most four out of the six intervals of type $u_j$ or $v_j$. This leaves at least one type $u_j$, at least one type $w_j$ interval, and $b^j_j$ undominated. It is easy to see that independent of which type $t_j$ and which type $f_j$ intervals are included in $D$, it is not possible for one interval to dominate all of these undominated intervals. Thus we have shown that $D$ cannot contain a type $a_j$ interval (since the above argument can be used to show that $D$ cannot contain $a^k_j$ or $a^l_j$ as well). Likewise it can be shown that $D$ does not contain any of the type $b_j$ intervals. Hence $D$ must contain two independent intervals of type $u_j$ or $w_j$ in order to dominate all the type $u_j$ and type $w_j$ intervals. From the structure of $I$ it can be seen that such two intervals must be of the form $u^j_k$ and $w^j_k$. Besides dominating all type $u_j$ and $w_j$ intervals, any pair $(u^j_k, w^j_k)$ of intervals also dominates $a^i_j$, $b^j_j$, $a^k_j$ and $b^l_j$. The other two intervals of type $a_j$ and $b_j$ associated with the clause $C_j$ must be dominated by a type $t$ interval.

The type $t$ and type $f$ intervals have thus accounted for $3m$ of the intervals in $D$ and the two type $u$ and type $w$ intervals per clause account for the remaining $2m$ intervals in $D$. Since all $5m$ intervals in $D$ have been accounted for, we have to show that all intervals in $I - D$ are dominated. The above argument has established that all the type $p$, type $s$, type $t$, type $f$, type $a$, type $b$, type
$u$, and type $w$ intervals have been dominated. This only leaves the type $tt$, type $tf$, type $ff$, and type $ft$ intervals. In the following we observe that these intervals are dominated if the truth values included in $D$ are “consistent” across variables and clauses.

(a) if $x_i$ (or $\overline{x_i}$) appears in two different clauses $C_j$ and $C_k$ and $D$ contains $t_j^i$ (respectively, $f_j^i$), then $D$ must also contain $t_k^i$ (respectively, $f_k^i$) in order to dominate $ft^i_{jk}$ (respectively, $tf^i_{jk}$).

(b) if $x_i$ appears in $C_j$, $\overline{x_i}$ appears in $C_k$ and $D$ contains $t_j^i$ (respectively, $f_j^i$), then $D$ must also contain $f_k^i$ (respectively, $t_k^i$) in order to dominate $ff^i_{jk}$ (respectively, $tt^i_{jk}$).

Now we can construct a satisfying truth assignment $A$ for $\phi$ as follows. If $D$ contains a type $t^i$ interval associated to a literal involving $x_i$, then we set this literal to true in $A$. Similarly, if $D$ contains a type $f^i$ interval associated with a literal involving $x_i$, then we set this literal to false in $A$. As mentioned above, $D$ must contain those type $t$ and type $f$ intervals that correspond to a literal having the same truth value throughout all clauses. Since for any clause $C_j$ there exists $i$ such that $t_j^i$ occurs in $D$, we conclude that any clause $C_j$ contains a true literal and therefore $A$ satisfies $\phi$ as required.

In the next result, we strengthen the above NP-completeness proof to show that the CMIDS problem is extremely hard even to approximate.

**Theorem 3** For any $\varepsilon > 0$, CMIDS does not have an $n^{1-\varepsilon}$-approximation algorithm, unless $P = NP$. Here $n$ is the number of intervals in the input to CMIDS.

**Proof.** The proof of the theorem is organized in three parts. In the first part, we present a reduction from 3SAT to CMIDS similar to the one described in Theorem 1. Let $\phi$ be an arbitrary instance of 3SAT with $X = \{x_1, x_2, \ldots, x_q\}$ as the set of boolean variables and $C = \{C_1, C_2, \ldots, C_m\}$ as the set of clauses. Let $J$ be the instance of CMIDS produced by the reduction from $\phi$ and let $G(J)$ be the overlap graph of $J$. The reduction is parameterized by a positive integer $\alpha$ and the size of $J$ depends on the size of $\phi$ and on $\alpha$.

In the second part we show that the reduction has the following two properties. (i) if $\phi$ is satisfiable, then a minimum independent dominating set in $G(J)$ has size $5m$ or less and (ii) if $\phi$ is not satisfiable, then a minimum independent dominating set in $G(J)$ has size greater than $5\alpha m$.

In the third part, we show that for any $\varepsilon > 0$, if $\alpha \leq |J|^{1-\varepsilon}$, then $|J|$ is a polynomial function of $m$. This ensures that the reduction described in the first part of the proof takes polynomial time.
These three pieces together prove the theorem. To see this suppose that for some $\varepsilon > 0$, we have an $n^{1-\varepsilon}$-approximation algorithm $A$ for CMIDS. Then we could use $A$ to solve an arbitrary instance $\phi$ of 3SAT in polynomial time as follows. Part 3 of the proof implies that given $\phi$, we construct $J$ in polynomial time. We then apply $A$ to $G(J)$. If $\phi$ is satisfiable, then the size of a minimum independent dominating set in $G(J)$ is $5m$ or less and therefore the size of the independent dominating set found by $A$ is $5n^{1-\varepsilon}m$ or less. If $\phi$ is not satisfiable, then the size of a minimum independent dominating set in $G(J)$ is greater than $5n^{1-\varepsilon}m$ and therefore the size of the independent dominating set found by $A$ is greater than $5n^{1-\varepsilon}m$. Hence, comparing the size of the set found by $A$ with the value $5n^{1-\varepsilon}m$ resolves the satisfiability of $\phi$ in polynomial time.

In the following we describe the reduction from 3SAT to CMIDS. The set of intervals $J$ is a simple extension of the set $I$ used in the proof of Theorem 1 and we only sketch the modifications here. The reduction is depicted in Figure 3. For an interval $a$, let $l(a)$ denote the left endpoint of $a$ and $r(a)$ denote the right endpoint of $a$. Consider an arbitrary clause $C_j$. For each variable $x_i$ that appears in $C_j$ the set $I$ contains two independent intervals $a_j^i$ and $b_j^i$. Recall that these two intervals overlap exactly the same set of intervals. Corresponding to these two intervals in $I$ we add $r = 5cm$ independent intervals, named $a_j^{i1}, a_j^{i2}, \ldots, a_j^{i r}$ to $J$ such that (i) $a_j^{il} = a_j^i$, $a_j^{il} = b_j^i$ and (ii) $a_j^{il+1} \subset a_j^{il}$ for $l = 1 \cdots r - 1$. This ensures that the intervals $a_j^{il}$ for all $l$ overlap exactly the same set of intervals. Let us call $a_j^{il}$ and $a_j^{ir}$ origina ls and $a_j^{ir}$ for all $l$, $1 < \ell < r$ copies. For each variable $x_i$ that appears in $C_j$, I also contains four more intervals $t_j^{i}, f_j^{i}, s_j^{i}$ and $p_j^{i}$. We include in $J$ the truth setting intervals $t_j^{i}$ and $f_j^{i}$ as they appear in the set $I$. Corresponding to $s_j^{i}$ and $p_j^{i}$ in $I$ we add to $J$, $r$ independent intervals $s_j^{i1}, s_j^{i2}, \ldots, s_j^{ir}$ such that (i) $s_j^{i1} = s_j^{i}$, $s_j^{i1} = p_j^{i}$ and (ii) $s_j^{i1+1} \subset s_j^{i1}$ for $l = 1 \cdots r - 1$. This ensures that the intervals $s_j^{i1}$ for all $l$, overlap exactly the same set of intervals. As before, let us call $s_j^{i1}$ and $s_j^{i1}$ origina ls and $s_j^{ir}$ for all $l$, $1 < \ell < r$ copies. The set $I$ also contains intervals of type $u_j$ and $w_j$ associated with each clause $C_j$. We add these to $J$ as they appear in $I$.

Finally, recall that in the construction used in Theorem 1, clauses that contain common variables are connected together by $tt$, $ff$, $tf$ and $ft$ type intervals in $J$. Corresponding to each of these intervals in $I$ we add $r$ new intervals to $J$ as follows. Without loss of generality we consider intervals of type $tt$. The construction is identical for intervals of type $ff$, $tf$, and $ft$. Suppose that $I$ contains the interval $tt_j^{ik}$. To $J$ we add $r$ independent intervals named $tt_j^{i1}, tt_j^{i2}, \ldots, tt_j^{ir}$ such that (i) $tt_j^{i1} = tt_j^{ik}$, (ii) $tt_j^{i1+1} \subset tt_j^{i1}$, and (iii) the endpoints of the intervals $tt_j^{i1}$ are chosen close enough so that for all $l$, the intervals $tt_j^{il}$ overlap exactly the same set of intervals. As before, we call $tt_j^{i1}$ an original and the rest of the type $tt$ intervals copies. We use a similar nomenclature for the type $ft$, $tf$, and $ff$ intervals. This completes the construction of $J$. We are now ready to prove the following lemma.
\[(\forall x \land \exists x \land \exists x) \lor (\exists x \land \exists x) \land (\exists x \land \exists x)\]

Fig. 3. The intervals constructed from the two clause and four variable formula (x1)
Lemma 4 If $\phi$ is satisfiable, then a minimum independent dominating set in $G(J)$ has size $5m$ or less. If $\phi$ is not satisfiable, then a minimum independent dominating set in $G(J)$ has size greater than $5\alpha m$.

PROOF. Suppose that $\phi$ is satisfiable. As shown in the proof of Theorem 1, we can construct an independent set $D$ of size $5m$ of type $t$, type $f$, type $u$ and type $w$ intervals that dominates all intervals in $I - D$. These intervals in $D$ appear in $J$ as well. Clearly, every interval in $J - D$ that appears in $I - D$ is dominated by some interval in $D$. The intervals in $J - D$ that do not appear in $I - D$ are copies. Since all the originals are dominated and the copies have exactly the same neighbors as the originals, and since none of the originals are in $D$, it follows that the copies are also dominated. This ensures that all intervals in $J - D$ are dominated by intervals in $D$.

Now suppose that $\phi$ is not satisfiable. Let $D$ be a smallest set of independent intervals in $J$ that dominates all intervals in $J - D$. Suppose that $D$ contains an interval of type $a$ - say $a^i_j$. Since intervals in $D$ are independent, $D$ cannot contain any interval that overlaps with $a^i_j$, or $f^i_j$. Since the original and all the copies of type $a$ have exactly the same set of neighbors, to be dominated, all intervals of type $a$ must be in $D$. A similar argument can be made for intervals of type $tt$, $ft$, $tf$, $ff$, and $s$. This means that if $D$ contains any of these intervals, then its size is at least $r = 5\alpha m$. Now suppose that $D$ does not contain an interval of the following types: $a$, $tt$, $tf$, $ft$, $ff$, or $s$. The absence of the type $s$ intervals in $D$ implies that for each appropriate $i$, $j$ pair, $D$ contains $t^i_j$ or $f^i_j$. Thus we have an assignment of truth values to all the literals. The absence of type $tt$, $ff$, $tf$, and $ft$ intervals implies that these truth values are consistent with each other. This implies that for some clause $C_j$ and for all relevant $i$, $f^i_j \in D$. This leaves all the type $a_j$ intervals undominated and these cannot be dominated by the inclusion of the type $u$ and type $w$ intervals in $D$. This implies that at least one type $a_j$ interval has to be included in $D$. But, if one type $a_j$ interval is included in $D$, then all $r = 5\alpha m$ type $a_j$ intervals have to be included in $D$. This implies that if $\phi$ is not satisfiable, any dominating set of $J$ contains at least $5\alpha m$ intervals. \qed

To complete the proof of this theorem, it remains to show that the transformation from $\phi$ to $J$ takes time that is polynomial in the size of $\phi$ when $\alpha \leq |J|^{1-\varepsilon}$. The following lemma states this claim formally.

Lemma 5 For any $\varepsilon > 0$, and any $\alpha \leq |J|^{1-\varepsilon}$, $|J|$ is bounded above by a polynomial function of $m$.

PROOF. The set $J$ contains intervals of type $a$, type $t$, type $f$, type $s$, type $u$, type $w$, type $tt$, type $tf$, type $ft$ and type $ff$. The number of type $a$ intervals
per clause per literal is \( r = 5 \alpha m \) and therefore the total number of type \( a \) intervals in \( J \) is \( 3rm \). The number of type \( t \) and type \( f \) intervals per clause per literal is 2 and thus the total number of type \( t \) and type \( f \) intervals in \( J \) is \( 6m \). The number of type \( s \) intervals per clause per literal is \( r \) and therefore the total number of type \( s \) intervals in \( J \) is \( 3rm \). The number of type \( u \) and type \( w \) intervals per clause per literal is 6 and therefore the total number of type \( u \) and type \( w \) intervals in \( J \) is \( 6m \). For each \( i, 1 \leq i \leq q \), let \( d_i \) be the number of clauses that a variable \( x_i \) appears in (in positive or negative form). Then the number of type \( tt \), type \( tf \), type \( ft \) and type \( ff \) intervals associated with \( x_i \) is \( d_i(d_i-1)r \). Since \( \Sigma_{i=1}^q d_i = 3m \), the total number of type \( tt \), type \( tf \), type \( ft \) and type \( ff \) intervals in \( J \) is at most \( \frac{3m(3m-1)r}{2} \). Summing up all the intervals in \( J \) yields the inequality

\[
|J| \leq \frac{45m^2}{2}(m + 1)|J|^{1-\varepsilon} + 12m.
\]  

We weaken the right hand side further to obtain \( |J| \leq 45m^2(m + 1)|J|^{1-\varepsilon} \) and solve this for \( |J| \) to get

\[
|J| \leq \left(45m^2(m + 1)\right)^{\frac{1}{\varepsilon}}.
\]  

Therefore, \( |J| \) is bounded above by a polynomial in \( m \) for any fixed \( \varepsilon > 0 \).

\[ \Box \]

It is easy to see that the reduction described at the beginning of the proof takes time proportional to \( |J| \). By showing that when \( \alpha \leq |J|^{1-\varepsilon} \), \( |J| \) is bounded above by a polynomial function of \( m \), we have established that the reduction is a polynomial time reduction.

\[ \Box \]

3 APX-hardness of MDS, MCDS, and MTDS

In this section we show that MDS, MCDS, and MTDS are all APX-hard for circle graphs. In other words, for each of these problems, there is a constant \( \delta > 0 \) such that if there is a \( (1 + \delta) \)-approximation algorithm for the problem, then \( P = NP \). These results complement the \( (2 + \varepsilon) \)-approximation algorithm for MDS on circle graphs and the \( (3 + \varepsilon) \)-approximation algorithms for MCDS and MTDS on circle graphs presented in [9]. Our results are based on a polynomial-time reduction from MAX-3SAT(8); the reductions use ideas from the reductions from SAT presented in [18] that established the NP-completeness of MDS, MCDS, and MTDS. The reductions from MAX-3SAT(8) to MDS,
MCDS and MTDS all have the same overall structure and we only present the reduction from MAX-3SAT(8) to MDS. The problem MAX-3SAT\((k)\) is defined below.

**MAX-3SAT\((k)\)**

**INPUT:** A set \(X = \{x_1, x_2, \ldots, x_n\}\) of variables and a set \(C = \{c_1, c_2, \ldots, c_m\}\) of disjunctive clauses such that each clause contains at most 3 literals and each variable occurs in at most \(k\) clauses.

**OUTPUT:** A truth-assignment to variables in \(X\) that maximizes the number of clauses in \(C\) satisfied.

For any instance \(\phi\) of MAX-3SAT\((k)\), let \(\text{SAT}(\phi)\) denote the largest fraction of clauses in \(\phi\) that can be simultaneously satisfied. For any graph \(G\) let \(\text{DOM}(G)\) denote the size of a minimum dominating set in \(G\). We show the following theorem.

**Theorem 6** There is a polynomial time reduction that takes an instance \(\phi\) of MAX-3SAT\((8)\) with \(n\) variables and \(m\) clauses and constructs a circle graph \(G\) such that

\[
\text{SAT}(\phi) = 1 \Rightarrow \text{DOM}(G) \leq 16n + 2 \\
\text{SAT}(\phi) < \alpha \Rightarrow \text{DOM}(G) > 16n + 2 + (1 - \alpha)m/8
\]

### 3.1 The reduction

Let \(\phi\) be an instance of MAX-3SAT\((8)\); without loss of generality we assume that each variable appears in exactly 8 clauses. Otherwise, we can add dummy clauses to make this happen without changing the approximability of the problem. We now show a polynomial-time reduction that maps \(\phi\) to a circle graph such that the above theorem holds. We construct in polynomial time from \(\phi\), a set \(J\) of chords of a circle. The theorem holds for the circle graph \(G(J)\) induced by the chords in \(J\). Since this reduction is similar to the reduction in [18] (Theorem 2.1) we do not present details such as co-ordinates of endpoints of chords, merely emphasizing intersections between chords. As a running example for the reduction we consider an instance \(\phi\) of MAX-3SAT\((8)\) in which the literal \(x_1\) appears in clauses \(c_1, c_2, c_4\), and \(\overline{x}_1\) appears in \(c_3, c_5, c_6, c_7, c_8\).

The set \(J\) contains \(m\) pairwise non-intersecting chords \(C_1, C_2, \ldots, C_m\) corresponding respectively to the clauses \(c_1, c_2, \ldots, c_m\). The chords \(C_1, C_2, \ldots, C_m\) are placed in counterclockwise order around the circle as shown in Figure 4(a). For each variable \(x_i\) and each clause \(C_j\), the set \(J\) contains a base chord \(B^j_i\), pro-
vided $x_i$ appears (as $x_i$ or as $\overline{x}_i$) in clause $C_j$. Thus the number of base chords in $J$ associated with each $x_i$ is exactly 8. This step of the reduction differs from Keil’s reduction in which for each variable $x_i$ there were $m$ base chords $B_j^i$, independent of the number of clauses $x_i$ appears in. The base chords are all pairwise non-intersecting and are placed as shown in Figure 4(a). Specifically, as we travel counterclockwise around the circle starting at the clause chords, we first encounter the base chords for $x_1$, then the base chords for $x_2$, and so on until we encounter the base chords for $x_n$. Suppose variable $x_i$ appears in clauses $c_{j_t}$, $1 \leq t \leq 8$, $j_1 < j_2 < \cdots < j_8$, then the base chords $B_{j_1}^i, B_{j_2}^i, \ldots, B_{j_8}^i$ appear in this order as we travel counterclockwise around the circle.

For each variable $x_i$, we add to $J$ four upper chords $U_i = \{U_1^i, U_2^i, U_3^i, U_4^i\}$ and four lower chords $L_i = \{L_1^i, L_2^i, L_3^i, L_4^i\}$. Each chord in $U_i \cup L_i$ intersects exactly two base chords corresponding to $x_i$. Suppose variable $x_i$ appears in clauses $c_{j_t}$, $1 \leq t \leq 8$, $j_1 < j_2 < \cdots < j_8$. Then $U_1^i$ intersects $B_{j_1}^i$ and $B_{j_2}^i$; $U_2^i$ intersects $B_{j_3}^i$ and $B_{j_4}^i$; $U_3^i$ intersects $B_{j_5}^i$ and $B_{j_6}^i$; and $U_4^i$ intersects $B_{j_7}^i$ and $B_{j_8}^i$. The lower chords intersect the base chords as follows: $L_1^i$ intersects $B_{j_1}^i$ and $B_{j_5}^i$; $L_2^i$ intersects $B_{j_2}^i$ and $B_{j_6}^i$; $L_3^i$ intersects $B_{j_3}^i$ and $B_{j_7}^i$; and $L_4^i$ intersects $B_{j_4}^i$ and $B_{j_8}^i$. Figure 4(b) shows the placement of the upper and lower chords for variable $x_1$ and how they interact with its base chords.

Note that chords in $U_i$ dominate all base chords corresponding to $x_i$ and similarly chords in $L_i$ dominate all base chords corresponding to $x_i$. For any dominating set $D$ of $G(J)$, $U_i \subseteq D$ and $L_i \cap D = \emptyset$ corresponds to setting $x_i$ true and $L_i \subseteq D$ and $U_i \cap D = \emptyset$ corresponds to setting $x_i$ false.

We include in $J$ four more chords associated with each variable $x_i$ that appears in a clause $C_j$. If the literal $x_i$ appears in $C_j$ then we add the chords $w_{j_1}^i, d_{j_1}^i$, 

Fig. 4. $m = 8$ (a) Clause chords and base chords corresponding to variable $x_1$ (b) The lower and and upper chords corresponding to variable $x_1$. 
$f^i_j$ and $g^j_k$ to $J$. These chords induce a simple path from $w^i_j$ to $C_j$ in $G(J)$. The chord $w^i_j$ intersects $B^j_i$ and an upper chord in $U^i$. See Figure 5(a). Thus in any setting containing all the chords in $U^i$, $w^i_j$ is dominated. Dominating $C_j$ with $g^j_k$ corresponds to satisfying $c_j$ by setting $x_i$ to true. Figure 5(b) shows the chords $w^1_2, d^1_2, f^1_2$ and $g^1_2$. Figure 7(a) shows all type $w$, type $d$, type $f$ and type $g$ chords associated with $x_1$.

\[ \text{(a)} \quad \text{(b)} \]

Fig. 5. $x_1$ appears in $C_2$: $J$ contains a sequence of chords $w^1_2, d^1_2, f^1_2, g^1_2$ that induce a path from $w^1_2$ to $C_2$. (a) Here we show how $w^1_2$ interacts with the base chords, lower chords, and upper chords corresponding to $x_1$ (b) Here we show the chords $w^1_2, d^1_2, f^1_2$, and $g^1_2$ relative to the rest of the chords.

If $x_i$ appears in $C_j$, then we include the chords $v^i_j, e^i_j, f^i_j$ and $g^i_j$ in $J$. Again, these chords induce a simple path in the circle graph from $v^i_j$ to $C_j$. The chord $v^i_j$ intersects $B^i_j$ and a lower chord in $L^i$. See Figure 6(a). Thus in any dominating set containing all the chords in $L^i$, $v^i_j$ is dominated. Like before, dominating $C_j$ with $g^i_j$ corresponds to satisfying $C_j$ with $x_i$, but by setting $x_i$ to false. Figure 6(b) shows chords $v^1_6, e^1_6, f^1_6$ and $g^1_6$. Figure 7(a) shows all type $v$, type $e$, type $f$ and type $g$ chords associated with $x_1$.

Note that all type $f$ chords are grouped together as shown in Figure 7(a). Specifically, as we travel counterclockwise from the clause chords, we first encounter all the base chords, and then all the type $f$ chords. The type $f$ chords appear in the same order as the type $B$ chords. First come the 8 type $f^1$ chords, then the 8 type $f^2$ chords, and so on till the 8 type $f^8$ chords. If a variable $x_i$ appears in clauses $c_{j_t}, 1 \leq t \leq 8$, $j_1 < j_2 < \cdots < j_8$, then the 8 type $f^i$ chords $f^i_{j_1}, f^i_{j_2}, \ldots, f^i_{j_8}$ appear in this order as we travel the circle in counterclockwise order. Now we add a pair of chords $p^i_1$ and $p^i_2$ so that $p^i_1$ intersects all the type $d$ and $e$ chords and $p^i_1$ intersects only $p^i_2$. This is shown in Figure 7(a). This construction means for any dominating set $D$ of $G(J)$ that

17
Fig. 6. $\mathcal{F}_1$ appears in $C_6$: $J$ contains a sequence of intervals $v_6^1, e_6^1, f_6^1, g_6^1$ that induce a path from $v_6^1$ to $C_6$. (a) Here we show the interaction of $v_6^1$ with base chords, lower chords, and upper chords of $x_6$. (b) Here we show the chords $v_6^1, e_6^1, f_6^1$ and $g_6^1$ relative to the rest of the chords.

does not contain $p'_1$, there exists a dominating set, no larger, than contains $p'_1$. Including $p'_1$ in a dominating set will enable us to treat all the $d$ and type $e$ chords as dominated and thus type $d$ and type $e$ chords will be included only if they are needed to dominate other chords. Similarly, we add a pair of chords $p_0'$ and $p_0$ such that $p_0'$ intersects all type $g$ chords and $p_0$ intersects exactly $p_0'$. Again, we have that for any dominating set $D$ of $G(J)$ that does not contain $p_0'$, there exists a dominating set, no larger, than contains $p_0'$. This is also shown in Figure 7(a).

Finally, we add to $J$, $4n$ pairs of chords: $p_{is}$ and $p'_{is}$, $1 \leq i \leq n$, $1 \leq s \leq 4$. Each $p_{is}$ intersects exactly one chord, $p'_{is}$. For each $i$, $1 \leq i \leq n$, the chords in $\{p_{is} \mid 1 \leq s \leq 4\}$ collectively dominate all the type $U^i$ and type $L^i$ chords, as shown in Figure 7(b). None of the $p_{is}$ chords intersect the base chords of $x_i$. This completes our construction; see [18] for details such as actual coordinates of endpoints.

The total number of vertices in our circle graph is $|J| = m + 56n + 4$. To see this, observe that $J$ contains $m$ clause chords. For each variable $x_i$, $J$ contains: 8 base chords, 4 upper chords, 4 lower chords, and 32 chords of types $w, v, d, e, f$ and $g$. Thus for each variable we have 48 chords. In addition, we have a total of $4 + 8n$ type $p$ and $p'$ chords for a total of $m + 56n + 4$ chords.
Fig. 7. (a) $p_0'$ dominates all type $g$ type intervals; $p_1'$ dominates all type $d$ and $e$ intervals (b) type $p'$ intervals dominate all type $U$ and $L$ intervals: $p_{11}'$ dominates $U_1^1, L_1^1$; $p_{12}'$ dominates $U_2^1, L_2^1$; $p_{13}'$ dominates $U_3^1, L_3^1$; and $p_{14}'$ dominates $U_4^1, L_4^1$.

3.2 Analysis

Lemma 7 $SAT(\phi) = 1 \Rightarrow DOM(G) \leq 16n + 2$.

PROOF. Since $SAT(\phi) = 1$, there is a satisfying truth assignment $A$ for $\phi$. We construct a dominating set $D$ of size $16n + 2$ using the procedure described in [18], which we briefly sketch here. As mentioned earlier, we can assume without loss of generality that $D$ contains all the type $p'$ chords. There are $4n + 2$ such chords and they dominate all the type $p, U, L, d, e$ and $g$ chords. It remains to dominate the type $B, C, v, w$ and $f$ chords. If $x_i$ is true in $A$, we include in $D$ all type $U^i$ chords; if $x_i$ is false in $A$, we include all $L^i$ chords. Thus we have added $4n$ more chords to $D$ and have dominated all base chords.

Suppose that the literal $x_i$ appears in a clause $c_j$. Then if $x_i$ is true in $A$, $w^i_j$ is dominated by chords in $U^i$, and $d^i_j$ is dominated by a type $p'$ chord. We add the chord $g^i_j$ to $D$ to dominate $f^i_j$. If $x_i$ is false in $A$, we add $d^i_j$ to dominate $w^i_j$ and $f^i_j$. In either case, we use a single chord for the $(i, j)$ pair. Now suppose that the literal $\overline{x}_i$ appears in a clause $c_j$. Then if $x_i$ is false in $A$, $v^i_j$ is dominated by chords in $L^i$, and $e^i_j$ is dominated by a type $p'$ chord. We add the chord $g^i_j$ to $D$ to dominate $f^i_j$. If $x_i$ is false in $A$, we add $e^i_j$ to dominate $v^i_j$ and $f^i_j$. In either case, we a single chord for the $(i, j)$ pair.

Since there are $8n$ possible $(i, j)$ pairs, we add $8n$ additional chords and dominate all the type $v$, type $w$, and type $f$ chords. Since $A$ is a satisfying truth
assignment, every clause \( c_j \) is dominated by a chord \( g^j_i \) for some \( i \). The number of chords we have included in \( D \) is \( 16n + 2 \).

**Lemma 8** \( SAT(\phi) < \alpha \Rightarrow DOM(G) > 16n + 2 + (1 - \alpha)m/8 \).

**Proof.** We prove this by showing that if there is a dominating set \( D \) of \( G(J) \), of size \( |D| \leq 16n + 2 + (1 - \alpha)m/8 \), then there is a truth assignment to variables in \( X \) that satisfies at least \( \alpha m \) of the clauses. For any subset \( J' \subseteq J \) of chords, define the \( D \)-dominating set of \( J' \) as

\[
(J' \cup \{ y \in J \mid y \text{ is a neighbor of some vertex in } J' \}) \cap D.
\]

Let \( D_p, D_B, \) and \( D_f \) be \( D \)-dominating sets respectively for the set of type \( p \) chords, the set of base chords, and the set of type \( f \) chords. There are \( (4n + 2) \) type \( p \) chords, no two of which have a common neighbor and so \( |D_p| \geq (4n + 2) \). There are \( 8n \) base chords, no three of which share a neighbor and so \( |D_B| \geq 4n \). There are \( 8n \) type \( f \) chords, no two of which share a neighbor and so \( |D_f| \geq 8n \). Also, the sets \( D_p, D_B, \) and \( D_f \) are pairwise non-intersecting because no chord in \( J \) intersects a type \( p \) chord and a base chord, or a base chord and a type \( f \) chord, or a type \( f \) chord and a type \( p \) chord. Therefore, the inequality \( |D_p| \geq 4n + 2 \) implies that \( |D_B| + |D_f| \leq 12n + (1 - \alpha)m/8 \).

For notational convenience, let \( \beta = 1 - \alpha \) and let \( \beta_1, \beta_2 \geq 0 \) be reals such that \( |D_B| = 4n + \beta_1 m/8 \) and \( |D_f| = 8n + \beta_2 m/8 \). This implies that \( \beta_1 + \beta_2 \leq \beta \).

For any \( i, 1 \leq i \leq n \), let \( D^i_B \subseteq D_B \) be the \( D \)-dominating set for the base chords corresponding to variable \( x_i \). It is easy to verify that \( |D^i_B| \geq 4 \) and if \( |D^i_B| = 4 \) then \( D^i_B = U^i \) or \( D^i_B = L^i \). Note that the sets \( D^i_B \) are pairwise disjoint for distinct \( i \)'s and therefore \( |D_B| = \sum_{i=1}^{n} |D^i_B| \). Since \( |D_B| = 4n + \beta_1 m/8 \), this implies that for at most \( \beta_1 m/8 \) of the \( i \)'s we have \( |D^i_B| > 4 \), while for the rest of the \( i \)'s we have \( |D^i_B| = 4 \) and therefore \( D^i_B = U^i \) or \( D^i_B = L^i \). For any \( i, 1 \leq i \leq n \), if \( D^i_B = U^i \) assign to \( x_i \) the value true; otherwise if \( D^i_B = L^i \), assign to \( x_i \) the value false. Variables to which truth values have been assigned are called consistent; the remaining variables are called inconsistent. Arbitrarily assign truth values to inconsistent variables. Next we show that this truth assignment satisfies at least \( \alpha m \) of the clauses.

Note that there are at most \( \beta_1 m/8 \) inconsistent variables. These can participate in at most \( \beta_1 m \) clauses and therefore the remaining at least \( (1 - \beta_1)m \) clauses contain only consistent variables. Let \( C' \subseteq C \) denote the subset of clauses that contain only consistent variables. Call any clause in \( C' \) a consistent clause and call any type \( C \) chord that corresponds to a clause in \( C' \) a consistent chord. Let \( t \) be the number of consistent chords in \( D \). It follows that \( t \leq (\beta - \beta_1 - \beta_2)m/8 \), because otherwise,
\[ |D| \geq |D_p| + |D_B| + |D_f| + t \]
\[ > (4n+2) + (4n + \beta_1 m/8) + (8n + \beta_2 m/8) + (\beta - \beta_1 - \beta_2)m/8 \]
\[ = 16n + 2 + \beta m/8 \]

a contradiction. This implies that the number of consistent chords dominated by type \( g \) chords is at least

\[ (1 - \beta_1)m - (\beta - \beta_1 - \beta_2) \frac{m}{8} \geq (1 - \beta)m + \frac{\beta_2 m}{8}. \]

Let \( S \) denote the set of indices \( j \) such that \( c_j \) is a consistent chord dominated by a type \( g \) chord. Now construct a set \( F \) of type \( f \) chords as follows: for each \( j \in S \), pick a \( g^j \) that dominates \( c_j \) (we know such a \( g^j \) exists) and add the chord \( f^j \) to \( F \). For each \( j \in S \), \( F \) contains exactly one \( f^j \) for some \( i \). Also note that \( |F| \geq (1 - \beta)m + \frac{\beta_2 m}{8} \) and all of the chords in \( F \) are dominated by type \( g \) chords. Of the chords in \( F \), at most \( \beta_2 m/8 \) chords can be dominated by 2 or more chords. This is because \( |D_f| = 8n + \beta_2 m/8 \) and no two type \( f \) chords share a neighbor. Hence, there are at least \( (1 - \beta)m = \alpha m \) type \( f \) chords in \( F \) that are dominated only by type \( g \) chords.

Now consider a chord \( f^j_i \in F \), dominated only by a type \( g \) chord. Since \( f^j_i \) is in \( J \), either \( x_i \) or \( \overline{x_i} \) appears in \( c_j \). Suppose that \( x_i \) appears in \( c_j \). Then we have the chords \( w^j_i, d^j_i \) also in \( J \). Since \( f^j_i \) is dominated only by type \( g \) chords, \( d^j_i \not\in D \) and this in turn implies that either \( w^j_i \in D \) or there is an upper chord that dominates it. Since \( C_j \) is a consistent chord, it only contains consistent variables and therefore \( D^i_B = U^i \) or \( D^i_B = L^i \). Hence, \( w^j_i \not\in D \), implying that \( D^i_B = U^i \), which in turn implies that \( x_i \) is assigned true and therefore clause \( c_j \) is satisfied.

A similar argument suffices to show that \( c_j \) is satisfied even in the case when \( \overline{x_i} \) appears in \( c_j \). Therefore, the truth assignment satisfies at least \( \alpha m \) clauses. This completes the proof.

A fundamental consequence of the PCP theorem [2] is the result that there exists an \( \alpha \), \( 0 < \alpha < 1 \) such that it is not possible to distinguish instances \( \phi \) of MAX-3SAT(8) for which \( \text{SAT}(\phi) = 1 \) from instances for which \( \text{SAT}(\phi) < \alpha \), unless \( P = NP \). This along with the “gap-preserving” reduction described above leads to the following result.

**Theorem 9** There exists a \( \delta > 0 \) such that MDS does not have a \((1 + \delta)\)-approximation algorithm, unless \( P = NP \).
Let BMIDS be the minimum independent dominating set problem restricted to bipartite graphs. Irving [16] shows that for any fixed $\alpha > 1$, no polynomial time \( \alpha \)-approximation algorithm exists for BMIDS unless P=NP. We strengthen this result to show that BMIDS is not approximable within a factor of \( n^{1-\varepsilon} \), for any \( \varepsilon > 1 \), unless P = NP.

**Theorem 10** For any \( \varepsilon > 0 \), BMIDS does not have an \( n^{1-\varepsilon} \)-approximation algorithm unless \( P = NP \). Here \( n \) is the number of intervals in the input to BMIDS.

**Proof.** The proof in [16] is based on a polynomial time reduction of SAT to BMIDS. We briefly sketch the main idea here. Given an arbitrary instance \( \phi \) of SAT, with \( X = \{ x_1, x_2, \cdots, x_q \} \) the set of boolean variables and \( C = \{ C_1, C_2, \cdots, C_m \} \) the set of clauses in \( \phi \), a graph \( G = G(F) \) is constructed such that (i) if \( \phi \) is satisfiable, then a minimum independent dominating set in \( G \) has size \( m \) or less, and (ii) if \( \phi \) is not satisfiable, then a minimum independent dominating set in \( G \) has size greater than \( \alpha m \), where \( \alpha > 1 \) is a fixed constant. Thus comparing the size of an independent dominating set found by an \( \alpha \)-approximation algorithm with the value \( \alpha m \) resolves the satisfiability of \( \phi \) in polynomial time. We omit the details of the proof here.

The graph \( G(\phi) \) has two vertices for each variable \( x_i \), \( i = 1 \cdots q \), and \( \alpha m \) vertices for each clause \( C_j \), \( j = 1 \cdots m \). Thus the total number of vertices in \( G \) is \( n = 2q + \alpha m^2 \). Assuming that each clause has exactly three literals, we have \( q \leq 3m \) and therefore \( n \leq 6m + \alpha m^2 \). Now let \( \alpha = n^{1-\varepsilon} \), for any \( \varepsilon > 0 \), and then solve the inequality \( n \leq 6m + n^{1-\varepsilon}m^2 \) for \( n \). The fact that \( m \geq 1 \) and \( n^{1-\varepsilon} \geq 1 \) implies that \( n \leq 12m^2n^{1-\varepsilon} \). So \( n \leq (12m^2)^{1/\varepsilon} \) and this implies that for fixed \( \varepsilon \), \( n \) is bounded above by a polynomial in \( m \). This ensures that the transformation from 3SAT to BMIDS takes time that is polynomial in the size of \( \phi \) when \( \alpha = n^{1-\varepsilon} \) for any \( \varepsilon > 0 \). \( \Box \)

We now determine the complexity of the “bichromatic” minimum independent dominating set problem and show that this problem is also hard to approximate. This problem is motivated by problems in polygon decomposition. In [8] we show that the problem of decomposing a polygon into minimum number of subpolygons each with diameter bounded above by a given constant can be modeled as a circle graph problem as follows. The input to the problem is a circle graph whose vertex set is partitioned into a set of RED and a set of BLUE vertices. The output sought is a minimum size independent set of RED vertices that dominates all the BLUE vertices. See [8] for details.
Let TRUE, INDEP, CONNECT, TOTAL, and CLIQUE refer to properties of a subset of vertices of a graph. In particular, for any graph $G = (V, E)$, any subset of $V$ has the property TRUE, any independent subset of $V$ has the property INDEP, any subset of $V$ that induces a connected subgraph has the property CONNECT, any subset of $V$ that induces a subgraph with no isolated vertices has the property TOTAL, and any subset of $V$ that induces a clique has the property CLIQUE. Letting $\pi \in \{\text{TRUE, INDEP, CONNECT, TOTAL, CLIQUE}\}$ we have the following classes of problems.

**Minimum $\pi$-Dominating Set on Circle Graphs ($\pi$-CMDS)**

**INPUT:** An interval representation of a circle graph $G = (V, E)$ and a natural number $k$.

**QUESTION:** Does $G$ have a dominating set of size no greater than $k$, having property $\pi$?

**Minimum $\pi$-Dominating Set on Bichromatic Circle Graphs ($\pi$-BCMDS)**

**INPUT:** An interval representation of a circle graph $G = (V, E)$, each of whose vertices is colored RED or BLUE, and a natural number $k$.

**QUESTION:** Does $G$ have a RED set of vertices $R$ of size no greater than $k$ such that (i) $R$ has property $\pi$ and (ii) all the BLUE vertices are dominated by $R$.

**Theorem 11** For any $\pi \in \{\text{TRUE, INDEP, CONNECT, TOTAL, CLIQUE}\}$ there is a polynomial time reduction from $\pi$-CMDS to $\pi$-BCMDS.

**PROOF.** Let $(G, k)$, with $G = (V, E)$, be an instance of $\pi$-CMDS. Since we are assuming that the interval representation of $G$ is available, we can view the vertices of $G$ as intervals. Let these intervals be labeled 1 through $n$ and for any interval $i$, $1 \leq i \leq n$, let $l(i)$ and $r(i)$ denote the left endpoint and the right endpoint respectively of interval $i$. Without loss of generality we assume that no two endpoints of intervals are coincident. Let $EP(V)$ denote the set of all endpoints on intervals in $V$. Let $\delta$ be the smallest distance between any pair of endpoints. Refer to Figure 8. In other words $\delta = \min\{|x - y| : x, y \in EP(V) \text{ and } x \neq y\}$. We now map the instance $(G, k)$ of $\pi$-CMDS into an instance $(G', k', C)$ of $\pi$-BCMDS, where $G' = (V', E')$ and $C : V' \to \{\text{RED, BLUE}\}$ is the assignment of colors to vertices. $V'$ is obtained by adding to $V$ intervals labeled $\{n + 1, n + 2, \ldots, 2n\}$ where interval $(n + i)$ has left endpoint $l(i) + \delta/2$ and right endpoint $r(i) + \delta/2$.

Thus we obtain $V'$ by making a copy of the intervals in $V$ and shifting the copies to the right by distance $\delta/2$. To complete the mapping, let $k' = k$ and let $C(i) = \text{RED}$ for $1 \leq i \leq n$ and $C(i) = \text{BLUE}$ for $n < i \leq 2n$. The following two observations about the relationship between edges in $E$ and edges in $E'$ are easy to verify.
Fig. 8. An example of reduction from $\pi$-CMDS to $\pi$-CMDS: RED intervals are input to an instance of $\pi$-CMDS; BLUE intervals are copies of the RED intervals shifted to the right by $\delta/2$.

**Observation 1:** For all $i$, $1 \leq i \leq n$, $\{i, n + i\} \in E'$.

**Observation 2:** For all $i, j$, $1 \leq i, j \leq n$, $\{i, j\} \in E$ iff $\{i, n + j\} \in E'$.

Now suppose that $D \subseteq V$ is a dominating set of $G$ of size no greater than $k$ with property $\pi$. Clearly, $C(i) = $ RED for all intervals $i \in D$. Let $j, n < j \leq 2n$ be a BLUE interval in $G'$. If $j - n \in D$, then by Observation 1, $\{j - n, j\} \in E'$ and so $j$ is dominated by a RED interval $j - n$ in $D$. Otherwise, if $(j - n) \not\in D$, then $(j - n)$ has a neighbor $k$, $1 \leq k \leq n$, in $D$. By Observation 2, $\{k, j\} \in E'$ and hence $j$ is dominated by a RED interval in $D$. Thus $D$ is a RED set that dominates all the BLUE intervals. It is clear that if a set of vertices $D \in V$ has property $\pi$ in $G$ then $D$ has property $\pi$ in $G'$ as well.

Now suppose that $D$ is a RED set of intervals with property $\pi$ that dominates all the BLUE intervals. Consider any interval $i$, $1 \leq i \leq n$, that is not in $D$. The interval $(n + i)$ is a BLUE interval and therefore has a neighbor $k$ in $D$. By Observation 2, $k$ is also a neighbor of $i$. Hence, $D$ is a dominating set of $G$. It is clear that if a set of vertices $D \in V$ has property $\pi$ in $G'$ then $D$ has property $\pi$ in $G$ as well. □

**Corollary 12** The problem $\pi$-BCMDS is NP-complete for all $\pi$ belonging to \{TRUE, INDEP, CONNECT, TOTAL\}. Furthermore, for $\pi =$ INDEP and for any $\varepsilon > 0$, there exists no $n^{1-\varepsilon}$-approximation algorithm for the problem $\pi$-BCMDS on $n$-vertex bichromatic circle graphs, unless $P = NP$.

We now show, somewhat surprisingly, that not only is it hard to find a smallest RED independent set that dominates all the BLUE vertices, but it is hard even to determine whether there exists a RED independent set that dominates all the BLUE vertices. We place this result in context by defining the following class of problems.

24
Existence of $\pi$-Dominating Set on Bichromatic Circle Graphs ($\pi$-BCEDS)

INPUT: An interval representation of a circle graph $G = (V, E)$, each of whose vertices is colored either RED or BLUE.

QUESTION: Is there a set of RED vertices with property $\pi$ that dominates all the BLUE vertices?

Let $V_R$ be the set of RED vertices in $G$ and let $G[V_R]$ denote the subgraph of $G$ induced by $V_R$. $\pi$-BCEDS is trivial when $\pi = \text{TRUE}$ because, in this case, the answer is yes if and only if $V_R$ dominates the set of all the BLUE vertices. Similarly, $\pi$-BCEDS is trivial when $\pi = \text{CONNECT}$. In this case the answer to $\pi$-BCEDS is yes if and only if the vertices of some connected component of $G[V_R]$ dominate all the BLUE vertices. $\pi$-BCMDS, for $\pi = \text{CLIQUE}$, can be solved using a slight modification of Keil’s polynomial time algorithm for finding a minimum dominating clique in a circle graph [18]. This implies that $\pi$-BCEDS is also polynomial time solvable for $\pi = \text{CLIQUE}$. When $\pi = \text{TOTAL}$, $\pi$-BCEDS can be solved by deleting all isolated vertices from $G[V_R]$ and determining whether the remaining vertices in $G[V_R]$ dominate all the BLUE vertices. All of these observations contrast with what the following theorem proves. The proof is an adaptation of the proof of Theorem 1 and we merely sketch the modifications.

**Theorem 13** $\pi$-BCEDS is NP-complete for $\pi = \text{INDEP}$. 

**Proof.** For this proof we fix $\pi = \text{INDEP}$. $\pi$-BCEDS is obviously in NP. We show that $\pi$-BCEDS is NP-complete by reducing 3SAT to $\pi$-BCEDS in polynomial time. Let $\phi$ be an arbitrary instance of 3SAT and consider the layout of intervals $I$ as in the proof of Theorem 1. Color RED all the type $t$, $f$, $u$ and $w$ intervals. Color BLUE all the remaining intervals. An argument similar to the one in the proof of Theorem 1, shows that the overlap graph of intervals in $I$ has an independent set of RED vertices that dominate all BLUE vertices if and only if $\phi$ is satisfiable. 

5 Conclusions

Our results show that domination problems on circle graphs are harder than may have been expected. We have shown elsewhere that MDS, MCDS, and MTDS on circle graphs have constant-factor approximation algorithms [9]. Together these results constitute a fairly complete understanding of domination problems for circle graphs.
Acknowledgment: We thank J. Mark Keil for sharing preliminary ideas on the NP-completeness proof of CMIDS

References


