The Maximum Clique Problem in Multiple Interval Graphs

Mathew C. Francis · Daniel Gonçalves · Pascal Ochem

Received: 10 October 2012 / Accepted: 17 August 2013 / Published online: 11 September 2013
© Springer Science+Business Media New York 2013

Abstract Multiple interval graphs are variants of interval graphs where instead of a single interval, each vertex is assigned a set of intervals on the real line. We study the complexity of the MAXIMUM CLIQUE problem in several classes of multiple interval graphs. The MAXIMUM CLIQUE problem, or the problem of finding the size of the maximum clique, is known to be NP-complete for \( t \)-interval graphs when \( t \geq 3 \) and polynomial-time solvable when \( t = 1 \). The problem is also known to be NP-complete in \( t \)-track graphs when \( t \geq 4 \) and polynomial-time solvable when \( t \leq 2 \). We show that MAXIMUM CLIQUE is already NP-complete for unit 2-interval graphs and unit 3-track graphs. Further, we show that the problem is APX-complete for 2-interval graphs, 3-track graphs, unit 3-interval graphs and unit 4-track graphs. We also introduce two new classes of graphs called \( t \)-circular interval graphs and \( t \)-circular track graphs and study the complexity of the MAXIMUM CLIQUE problem in them. On the positive side, we present a polynomial time \( t \)-approximation algorithm for MAXIMUM WEIGHTED CLIQUE on \( t \)-interval graphs, improving earlier work with approximation ratio \( 4t \).

Keywords \( t \)-interval graphs · \( t \)-track graphs · Unit \( t \)-interval graphs · Unit \( t \)-track graphs · Maximum clique · NP-completeness · Approximation hardness

An extended abstract of this paper has been accepted for publication in the proceedings of WG 2012.

M.C. Francis (✉)
Indian Statistical Institute, Chennai Centre, SETS, MGR Knowledge City, CIT Campus, Taramani, Chennai 600113, India
e-mail: mathew@isichennai.res.in

D. Gonçalves · P. Ochem
LIRMM, CNRS et Université Montpellier 2, 161 rue Ada, 34392 Montpellier Cedex 05, France
e-mail: gonalves@lirmm.fr

P. Ochem
e-mail: ochem@lirmm.fr
1 Introduction

Given a family of sets $F$, a graph $G$ with vertex set $V(G)$ and edge set $E(G)$ is said to be an “intersection graph of sets from $F$” if there exists a function $f : V(G) \to F$ such that for distinct $u, v \in V(G)$, $uv \in E(G) \iff f(u) \cap f(v) \neq \emptyset$. When $F$ is the set of all closed intervals on the real line, it defines the well-known class of interval graphs. A $t$-interval is the union of $t$ intervals on the real line. When $F$ is the set of all $t$-intervals, it defines the class of graphs called $t$-interval graphs. This class was first defined and studied by Trotter and Harary [25]. Given $t$ parallel lines (or tracks), if each element of $F$ is the union of $t$ intervals on different lines, one defines the class of $t$-track graphs. It is easy to see that this class forms a subclass of $t$-interval graphs.

These classes of graphs received a lot of attention, for both their theoretical simplicity and their use in various fields like Scheduling [3, 13] or Computational Biology [2, 8]. West and Shmoys [27] showed that recognizing $t$-interval graphs for $t \geq 2$ is NP-complete.

Given a circle, the intersection graphs of arcs of this circle forms the class of circular arc graphs. We introduce similar generalizations of circular arc graphs. If $G$ has an intersection representation using $t$ arcs on a circle per vertex, then $G$ is called a $t$-circular interval graph. If instead, $G$ has a circular representation using $t$ circles and exactly one arc on each circle corresponding to each vertex of $G$, then $G$ is called a $t$-circular track graph. For all these intersection families of graphs, one can define a subclass where all the intervals or arcs have the same length. We respectively call those subclasses unit $t$-interval, unit $t$-track, unit $t$-circular interval, and unit $t$-circular track graphs. Clearly, the class of $t$-interval graphs (resp. $t$-track graphs) forms a subclass of $t$-circular interval graphs (resp. $t$-circular track graphs); so do their unit length versions. One can see, by imagining cutting the circles to form lines, that $t$-circular interval graphs and $t$-circular track graphs are respectively contained in $(t + 1)$- and $(2t)$-interval graphs. Contrarily to $t$-track and $t$-interval graphs, one can prove that some $t$-circular track graphs (e.g. for $t = 2$ the circulant graph $C^{1,2,4,8}_{4n+1}$ with $n \geq 8$) are not $t$-circular interval graphs. We omit the proof of this statement here.

MAXIMUM WEIGHTED CLIQUE is the problem of deciding, given a graph $G$ with weighted vertices and an integer $k$, whether $G$ has a clique of weight $k$. The case where all the weights are 1 is MAXIMUM CLIQUE. Zuckerman [28] showed that unless P = NP, there is no polynomial time algorithm that approximates the maximum clique within a factor $O(n^{1-\epsilon})$, for any $\epsilon > 0$. MAXIMUM CLIQUE has been studied for many intersection graphs families. It has been shown to be polynomial for interval filament graphs [12], a graph class including circle graphs, chordal graphs and co-comparability graphs. It has been shown to be NP-complete for $B_1$-VPG graphs [22] (intersection of strings with one bend and axis-parallel parts [1]), and for segment graphs [6] (answering a conjecture of Kratochvíl and Nešetřil [21]).

MAXIMUM WEIGHTED CLIQUE is polynomial for interval graphs (folklore) and for circular arc graphs [11, 12, 14]. However, Butman et al. [5] showed that MAXIMUM CLIQUE is NP-complete for $t$-interval graphs when $t \geq 3$. For $t$-track graphs, MAXIMUM WEIGHTED CLIQUE is polynomial-time solvable when $t \leq 2$, but even MAXIMUM CLIQUE is NP-complete when $t \geq 4$ [20]. With respect to parameterized complexity, the MAXIMUM CLIQUE problem for $t$-interval graphs is
not very hard. Indeed, parametrized by both $t$ and the solution size, the problem is FPT even when the graph is given without its $t$-interval representation [9, 15]. On the other hand we show in the following that with respect to the approximation complexity, this problem is not very simple as it is often APX-hard. Butman et al. showed a polynomial-time $\frac{t^2-t+1}{2}$ factor approximation algorithm for MAXIMUM WEIGHTED CLIQUE in $t$-interval graphs. Koenig [20] observed that a similar approximation algorithm with a slightly better approximation ratio $\frac{t^2-t}{2}$ exists for MAXIMUM WEIGHTED CLIQUE in $t$-track graphs. Butman et al. asked the following questions:

- Is MAXIMUM CLIQUE NP-hard in 2-interval graphs?
- Is it APX-hard in $t$-interval graphs for any constant $t \geq 2$?
- Can an algorithm with a better approximation ratio than $\frac{t^2-t+1}{2}$ be achieved for MAXIMUM WEIGHTED CLIQUE in $t$-interval graphs?

We answer all of these questions in the affirmative. As far as the third question is concerned, Kammer and Tholey [19] have already presented an improved polynomial-time approximation algorithm that achieves an approximation ratio of $4t$ for $t$-interval graphs. In this paper (Sect. 3), we present a polynomial time $t$-approximation algorithm for MAXIMUM WEIGHTED CLIQUE in $t$-interval graphs (and thus in $t$-track graphs), $t$-circular interval graphs, and $t$-circular track graphs. Then we show in Sect. 4 that MAXIMUM CLIQUE is APX-complete for many of these families (including 2-interval graphs). In Sect. 5, we show that for some of the remaining classes (including unit 2-interval graphs) MAXIMUM CLIQUE is NP-complete. In Sect. 6 we give some APX-hardness results for several problems restricted to the class of complements of $t$-interval graphs. Finally, we conclude with some remarks and open questions.

2 Preliminaries

Consider a circle $C$ of length $l$ with a distinguished point $O$. The coordinate of a point $p \in C$ is the length of the arc going clockwise from $O$ to $p$. Given two reals $p$ and $q$, $[p, q]$ is the arc of $C$ going clockwise from the point with coordinate $p$ to the one with coordinate $q$. Whenever we consider arcs on a circle of length $l$, coordinates are understood to be modulo $l$.

A representation of a $t$-interval graph $G$ is a set of $t$ functions, $I_1, \ldots, I_t$, assigning each vertex in $V(G)$ to an interval of the real line. For $t$-track graphs we have $t$ lines $L_1, \ldots, L_t$, and each $I_i$ assigns intervals from $L_i$. Similarly, for a representation of $t$-circular interval graphs (resp. $t$-circular track graphs) we have a circle $C$ (resp. $t$ circles $C_1, \ldots, C_t$) and $t$ functions $I_i$, assigning each vertex in $V(G)$ to an arc of $C$ (resp. of $C_i$).

3 Approximation Algorithms

The first approximation algorithms for the MAXIMUM WEIGHTED CLIQUE in $t$-interval graphs and $t$-track graphs [5, 20] are based on the fact that any $t$-interval
representation (resp. \( t \)-track representation) of a clique admits a transversal (i.e. a set of points touching at least one interval of each vertex) of size \( \tau = t^2 - t + 1 \) (resp. \( \tau = t^2 - t \)) [18]. Scanning the representation of a graph \( G \) from left to right (in time \( O(tn) \)), one passes through the points of the transversal of a maximum weighted clique \( K \) of \( G \) with weight \( w \). If one defines the weight of a point to be the sum of the weights of all the intervals passing through that point, then the sum of the weights of all the points in the transversal of \( K \) is at least \( w \). This means that there is at least one point in the transversal whose weight is at least \( w/\tau \). The intervals passing through this point form a clique of weight at least \( w/\tau \) in \( G \). Thus, this gives an \( O(tn) \)-time \( \tau \)-approximation. Butman et al. improved this ratio by 2 by considering every pair of points in the representation. The intervals at these points induce a co-bipartite graph, for which computing the maximum weighted clique is polynomial (as computing a maximum weighted independent set of a bipartite graph is polynomial). Then one can see that this gives a polynomial time \( (\tau/2) \)-approximation algorithm. This actually gives a polynomial time exact algorithm for the MAXIMUM WEIGHTED CLIQUE in 2-track graphs [20], as \( \tau = 2 \) in this case. This technique gives interesting approximation factors for other classes. A representation is balanced if for each vertex, all its intervals (or arcs) have the same length.

\textbf{Remark 1} In any balanced \( t \)-interval (resp. \( t \)-track, \( t \)-circular interval, or \( t \)-circular track) representation of a clique, the \( 2t \) interval extremities of the vertex with the smallest intervals form a transversal. Thus \( \tau = 2t \), and in those classes of graphs, MAXIMUM WEIGHTED CLIQUE admits a polynomial time \( t \)-approximation algorithm.

Actually, earlier observations by Bar-Yehuda et al. [3] could have led to the conclusion that the trivial algorithm (finding the point of the representation belonging to the maximum number of intervals) is actually a \( 2t \)-approximation. This can be deduced from the proofs of Theorems 1 and 2 below.

Recently, Kammer and Tholey [19] improved the (known) approximation ratios from roughly \( t^2/2 \) to \( 4t \), using the new notion of \( k \)-perfectly orientable graphs. These graphs are the graphs that admit an orientation and a \( k \)-colouring of the edges such that for any vertex \( v \) and any colour \( i \), the vertex \( v \) together with its out-neighbors in colour \( i \) (i.e. restricted to \( i \)-coloured arcs) induce a clique. To achieve their \( 4t \)-approximation, Kammer and Tholey first proved that \( t \)-interval graphs are \( 2t \)-perfectly orientable, and then proved that MAXIMUM WEIGHTED CLIQUE in \( k \)-perfectly orientable graphs can be \( 2k \)-approximated in polynomial time. We improve both results in the following theorems.

\textbf{Theorem 1} Every \( t \)-interval graph (resp. \( t \)-track graph, \( t \)-circular interval graph, and \( t \)-circular track graph) is \( t \)-perfectly orientable.

\textbf{Proof} Let us prove the theorem for \( t \)-interval graphs, the proofs for the other classes are exactly the same. Let \( G \) be a \( t \)-interval graph with representation \( I_1, \ldots, I_t \). For any vertex \( u \in V(G) \), let the endpoints of its intervals be denoted \( u_i \) and \( u'_i \) in such a way that \( I_i(u) = [u_i, u'_i] \). For any edge \( uv \) there exists an \( i \) and a \( j \in [t] \) such that
the point \( u_i \) belongs to \( I_j(v) \), or such that the point \( v_j \) belongs to \( I_i(u) \). One can thus orient and \( t \)-colour the edges of \( G \) in such a way that \( uv \) goes from \( u \) to \( v \) in colour \( i \) if \( u_i \in I_j(v) \) for some \( j \). Note that several orientations and colourings are possible when several intervals of \( u \) and \( v \) intersect. Finally, one just has to notice that, for any vertex \( u \) and any colour \( i \), \( u \) and its out-neighbors in colour \( i \) intersect at point \( u_i \), and thus induce a clique in \( G \). □

In [19], the authors designed several approximation algorithms for \( t \)-perfectly orientable graphs. For example, they proved that \( t \)-perfectly orientable graphs admit a polynomial time \( 2t \)-approximation for MAXIMUM WEIGHTED INDEPENDENT SET, MINIMUM VERTEX COLORING, and MAXIMUM WEIGHTED CLIQUE. For the first two problems, this implies \( 2t \)-approximation algorithms for \( t \)-interval graphs. This reaches the best known approximation ratios [3]. Actually this is not surprising, as the algorithms in [19] are strongly inspired by the ones in [3]. For the MAXIMUM WEIGHTED CLIQUE, we shall show in Theorem 2 that one can achieve a better approximation ratio.

**Theorem 2** For any \( t \geq 2 \), there is a polynomial time \( t \)-approximation algorithm for MAXIMUM WEIGHTED CLIQUE on \( t \)-perfectly orientable graphs.

**Proof** Let \( G \) be a weighted \( t \)-perfectly orientable graph with weight function \( w(u) \) on its vertices. We assume that the edges of \( G \) are conveniently oriented and \( t \)-coloured (as described in the definition of \( t \)-perfectly orientable graphs). Let \( K \) be a maximum weighted clique of \( G \).

In \( K \) there is a vertex \( u \) with at least as much weight on its out-neighbors in \( K \) as on its in-neighbors in \( K \). Indeed, this comes from the fact that in any oriented graph \( D \) with positive weights on its vertices, there is a vertex with at least as much weight on its out-neighbors as on its in-neighbors. This can be seen as follows. Assign to each oriented edge \( uv \) of \( D \) the weight \( w(uv) = w(u)w(v) \). Since in \( D \), the sum of the total weight of out-edges over all vertices is equal to the sum of the total weight of in-edges over all vertices, there is a vertex \( u \) in \( D \) that has at least as much weight on its out-edges as on its in-edges. Therefore, \( \sum_{v \in N^+(u)} w(uv) \geq \sum_{v \in N^-(u)} w(uv) \), which gives us \( \sum_{v \in N^+(u)} w(u)w(v) \geq \sum_{v \in N^-(u)} w(u)w(v) \). This leads to \( w(u) \sum_{v \in N^+(u)} w(v) \geq w(u) \sum_{v \in N^-(u)} w(v) \), and therefore, \( \sum_{v \in N^+(u)} w(v) \geq \sum_{v \in N^-(u)} w(v) \). Therefore, in any oriented graph with positive weights on its vertices, there is a vertex with at least as much weight on its out-neighbors as on its in-neighbors. Since \( K \) was an oriented graph with positive weights on its vertices, we can assume that there is a vertex \( u \) in \( K \) with \( w(N^+_K(u)) \geq w(N^-_K(u)) \).

Since every vertex in \( K \) is a neighbour of \( u \), \( w(N^+_K(u)) \geq \frac{1}{t}(w(K) - w(u)) \). Let \( i \) be the colour such that the sum of the weights of the out-neighbours of \( u \) in that colour is the largest. As the arcs outgoing from \( u \) are coloured with \( t \) colours, the weight of the out-neighbors of \( u \) in colour \( i \) is at least \( \frac{1}{t}(w(K) - w(u)) \).

By definition, the vertex \( u \) and its out-neighbors (including the out-neighbors outside \( K \)) in colour \( i \) induce a clique of \( G \). Thus considering every vertex and every colour, one will consider the couple \( (u, i) \) and find a clique of weight at least...
Algorithmica (2015) 71:812–836

$w(u) + \frac{1}{2t}(w(K) - w(u)) > \frac{1}{2} w(K)$. That is $\frac{1}{2t}$ of the optimal, hence we have a $2t$-approximation.

For the $t$-approximation let us define colour $j \neq i$ as the colour other than $i$ such that the sum of the weights of the out-neighbours of $u$ in colour $j$ is the largest. As the arcs outgoing from $u$ are coloured with $t$ colours, the weight of the out-neighbours of $u$ in colour $i$ and $j$ is at least $\frac{2}{2t}(w(K) - w(u)) = \frac{1}{t}(w(K) - w(u))$.

Note that the graph induced by some vertex $v$ and its out-neighbors in any two given colours $i'$, $j'$ is co-bipartite (as it can be covered by two cliques). Recall that as noted by [20] computing the maximum weighted clique of a co-bipartite graph is polynomial (as computing a maximum weighted independent set of a bipartite graph is polynomial). Thus one can in polynomial time consider every vertex $v$ and every pair of colours $i'$, $j'$ and compute the maximum weighted clique of the graph induced by $v$ and its out-neighbors in colour $i'$ and $j'$. By doing this, one will consider vertex $u$ and colours $i$, $j$ and find the maximum weighted clique in the subgraph of $G$ formed by $u$ and its out-neighbours in colours $i$ and $j$. This clique will surely have weight at least $w(u) + \frac{1}{t}(w(K) - w(u)) > \frac{1}{t} w(K)$. That is $\frac{1}{t}$ of the optimal, hence we have our polynomial $t$-approximation algorithm. □

We thus have the following corollary by Theorem 1.

**Corollary 1** There is a polynomial time $t$-approximation algorithm for MAXIMUM WEIGHTED CLIQUE on $t$-interval graphs, $t$-track graphs, $t$-circular interval graphs, and $t$-circular track graphs.

In [19], the authors introduced a family of graph classes, the intersection graphs of $t$ objects with fatness $c$. These classes generalize several graph classes such as $t$-disk graphs (with fatness parameter $c = 6$) and $t$-square (not necessarily axis parallel) graphs (with fatness parameter $c = 10$). They proved that the intersection graphs of $t$ fat objects (with fatness $c$) are $tc$-perfectly orientable. Thus by Theorem 2 MAXIMUM WEIGHTED CLIQUE is $tc$-approximable on these graphs.

### 4 APX-Hardness in Multiple Interval Graphs

The complement of a graph $G$ is denoted by $\overline{G}$. Given a graph $G$ on $n$ vertices and $m$ edges, with $V(G) = \{x_1, \ldots, x_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$, and a positive integer $w$, we define $\text{Subd}_w(G)$ to be the graph obtained by subdividing each edge of $G$ $w$ times. If $e_k \in E(G)$ and $e_k = x_i x_j$ where $i < j$, we define $l(k) = i$ and $r(k) = j$ (as if $x_i$ and $x_j$ were respectively the left and right ends of $e_k$). In the following we subdivide edges 2 or 4 times. In $\text{Subd}_2(G)$ (resp. $\text{Subd}_4(G)$), the vertices subdividing $e_k$ are $a_k$ and $b_k$ (resp. $a_k, b_k, c_k, d_k$) and they are such that $(x_{l(k)}, a_k, b_k, x_{r(k)})$ (resp. $(x_{l(k)}, a_k, b_k, c_k, d_k, x_{r(k)})$) is the subpath of $\text{Subd}_2(G)$ (resp. $\text{Subd}_4(G)$) corresponding to $e_k$. To prove APX-hardness results we need the following structural theorem, which is of independent interest.

**Theorem 3** Given any graph $G$,
Subd$_4$(G) is a 2-interval graph,
Subd$_2$(G) is a unit 3-interval graph,
Subd$_2$(G) is a 3-track graph,
Subd$_2$(G) is a unit 4-track graph,
Subd$_2$(G) is a unit 2-circular interval graph (and thus a 2-circular interval graph),
Subd$_2$(G) is a 2-circular track graph, and
Subd$_2$(G) is a unit 4-circular track graph.

Furthermore, such representations can be constructed in linear time.

Since MAXIMUM INDEPENDENT SET is APX-hard even when restricted to degree bounded graphs [4, 24], Chlebík and Chlebíková [7] observed that MAXIMUM INDEPENDENT SET is APX-hard even when restricted to $2k$-subdivisions of 3-regular graphs for any fixed integer $k \geq 0$. Taking the complement graphs, we thus have that MAXIMUM CLIQUE is APX-hard even when restricted to the set $C_{2k} = \{\text{Subd}_2^k(G) \mid \text{any graph } G\}$, for any fixed integer $k \geq 0$. Thus, since MAXIMUM CLIQUE is approximable for all the graph classes considered in Theorem 3, we clearly have the next result.

**Theorem 4** MAXIMUM CLIQUE is APX-complete for:
- 2-interval graph,
- unit 3-interval graph,
- 3-track graph,
- unit 4-track graph,
- unit 2-circular interval graph (and thus for 2-circular interval graphs),
- 2-circular track graph, and
- unit 4-circular track graph.

Note that recently, Jiang [16] gave an alternative proof of the fact that MAXIMUM CLIQUE is APX-complete for 3-track graphs by refining the technique used in [5].

**Remark 2** To prove that MAXIMUM CLIQUE is NP-hard on $B_1$-VPG graphs (defined in [1]), Middendorf and Pfeiffer [22] proved that for any graph $G$, Subd$_2^k(G) \in B_1$-VPG. One can thus see that MAXIMUM CLIQUE is actually APX-hard for this class of graphs, and for the larger class of string intersection graphs.

We prove Theorem 3 in the following subsections.

### 4.1 2-Interval Graphs

**Theorem 5** Given any graph $G$, Subd$_4^4(G)$ is a 2-interval graph and a 2-interval representation for it can be constructed in linear time.

**Proof** Recall that each edge $e_k = x_i x_j$ of $G$ where $i < j$, corresponds to the path $(x_i, a_k, b_k, c_k, d_k, x_j)$ in Subd$_4^4(G)$. We define the representation $\{I_1, I_2\}$ of Subd$_4^4(G)$ as follows (see also Fig. 1). For $1 \leq i \leq n$ and $1 \leq k \leq m$:
Algorithmica (2015) 71:812–836

Fig. 1 The 2-interval representation of $\text{Subd}_4(G)$

$I_1(a_k) = [0,m(l(k) - 1) + k - 1]$
$I_1(x_i) = [mi, mn + mi]
$I_2(a_k) = [mn + ml(k) + 1, 4mn + m - ml(k) - k + 1]$
$I_1(b_k) = [m(l(k) - 1) + k, mn + m - k]$
$I_1(c_k) = [mn + m - k + 1, 3mn + m - mr(k) - k + 1]$
$I_1(d_k) = [3mn + m - mr(k) - k + 2, 4mn + mr(k)]$
$I_2(b_k) = [4mn + m - ml(k) - k + 2, 5mn + k]$
$I_2(x_i) = [4mn + mi + 1, 5mn + mi + 1]$
$I_2(d_k) = [5mn + mr(k) + k + 1, 6mn + m + 1]$
$I_2(c_k) = [5mn + k + 1, 5mn + mr(k) + k]$

Figure 1 (and the other figures of this kind) should be understood in the following way. The leftmost block labeled $a_k$ corresponds to the intervals $I_1(a_k)$, and its shape, together with the label $(l(k), k)$ on the arrow mean that,

– the left ends of the intervals $I_1(a_k)$ are the same (coordinate 0), and that
– the right ends of the intervals $I_1(a_k)$ are ordered (from left to right) accordingly to $l(k)$, and in case of equality, accordingly to $k$. 

Here we can see that this block is close to the blocks $I_1(b_k)$, and $I_1(x_i)$.

The left ends of the intervals $I_1(b_k)$ are also ordered (from left to right) accordingly to $(l(k), k)$. Such a situation means that $I_1(a_k)$ intersects $I_1(b_{k'})$ if and only if $(l(k), k) > (l(k'), k')$ (i.e. when $l(k) > l(k')$, or when $l(k) = l(k')$ and $k > k'$). Note that since, between $I_2(a_k)$ and $I_2(b_k)$ we have the opposite situation, for any vertex $a_k$, $a_k$ is adjacent to every $b_{k'}$, except $b_k$.

The left ends of the intervals $I_1(x_i)$ are ordered (from left to right) accordingly to $i$. Such a situation means that $I_1(a_k)$ intersects $I_1(x_i)$ if and only if $l(k) > i$. Note that since, between $I_1(x_i)$ and $I_2(a_k)$ we have the opposite situation, for any vertex $a_k$, $a_k$ is adjacent to every $x_{i(k)}$, except $x_{l(k)}$.

Note the dashed lines joining the left edge of the block $I_1(b_k)$ to the left edge of the block $I_1(x_i)$. This means that these blocks are overlaid one on top of the other, with their left edges coinciding. See Fig. 2 for an example in which the magnified view of the individual intervals in the three leftmost blocks of Fig. 1 are shown.
Fig. 2 A part of a graph and the arrangement of the intervals corresponding to the shown vertices in the three leftmost blocks in Fig. 1.
We claim that $I_1$ and $I_2$ together form a valid 2-interval representation for $\overline{\text{Subd}_4(G)}$. One can check it with Fig. 1, but we give a full proof for this first construction. For any two vertices $u$ and $v$ of $\overline{\text{Subd}_4(G)}$, we will show that $uv$ is an edge of $\overline{\text{Subd}_4(G)}$ if and only if $I_1(u) \cup I_2(u)$ intersects $I_1(v) \cup I_2(v)$. We first consider the case where $uv$ is an edge.

**Case** $u = x_i$ and $v = x_j$:

$[mn, mn + m] \subseteq I_1(x_i) \cap I_1(x_j)$.

**Case** $u = x_i$ and $v = a_k$, where $l(k) \neq i$:

If $l(k) > i$, then $mi \in I_1(a_k) \cap I_1(x_i)$. If on the other hand, $l(k) < i$, then $mn + mi \in I_1(x_i) \cap I_2(a_k)$.

**Case** $u = x_i$ and $v = b_k$:

$mn \in I_1(x_i) \cap I_1(b_k)$.

**Case** $u = x_i$ and $v = c_k$:

$mn + m \in I_1(x_i) \cap I_1(c_k)$.

**Case** $u = x_i$ and $v = d_k$, where $r(k) \neq i$:

If $r(k) > i$, then $4mn + mi + m \in I_1(d_k) \cap I_2(x_i)$ and if $r(k) < i$, then $5mn + mi + 1 \in I_2(x_i) \cap I_2(d_k)$.

**Case** $u = a_k$ and $v = a_{k'}$:

$0 \in I_1(a_k) \cap I_1(a_{k'})$.

**Case** $u = a_k$ and $v = b_{k'}$, where $k \neq k'$:

If $l(k') < l(k)$, then $ml(k) - 1 \in I_1(a_k) \cap I_1(b_{k'})$ and if $l(k) < l(k')$, then $4mn - ml(k) + 1 \in I_2(a_k) \cap I_2(b_{k'})$. Suppose $l(k) = l(k')$. Now, if $k' < k$, then $m(l(k) - 1) + k - 1 \in I_1(a_k) \cap I_1(b_{k'})$ and if $k' > k$, then $4mn + m - ml(k) - k + 1 \in I_2(a_k) \cap I_2(b_{k'})$.

**Case** $u = a_k$ and $v = c_{k'}$:

$2mn + 1 \in I_2(a_k) \cap I_1(c_{k'})$.

**Case** $u = a_k$ and $v = d_{k'}$:

$3mn + 1 \in I_2(a_k) \cap I_1(d_{k'})$.

**Case** $u = b_k$ and $v = b_{k'}$:

$mn \in I_1(b_k) \cap I_1(b_{k'})$.

**Case** $u = b_k$ and $v = c_{k'}$, where $k \neq k'$:

If $k < k'$, then $mn + m - k \in I_1(b_k) \cap I_1(c_{k'})$.

**Case** $u = b_k$ and $v = d_{k'}$:

$4mn + 1 \in I_2(b_k) \cap I_1(d_{k'})$.

**Case** $u = c_k$ and $v = c_{k'}$:

$[mn + m, 2mn + 1] \subseteq I_1(c_k) \cap I_1(c_{k'})$.

**Case** $u = c_k$ and $v = d_{k'}$, where $k \neq k'$:

If $r(k) < r(k')$, then $3mn - mr(k) + 1 \in I_1(c_k) \cap I_1(d_{k'})$ and if $r(k') < r(k)$, then $5mn + mr(k) + 1 \in I_2(c_k) \cap I_2(d_{k'})$. Suppose $r(k) = r(k')$. Now, if $k < k'$, $3mn - mr(k) + 1 \in I_1(c_k) \cap I_1(d_{k'})$ and if $k' < k$, then $5mn + mr(k) + k \in I_2(c_k) \cap I_2(d_{k'})$.

**Case** $u = d_k$ and $v = d_{k'}$:

$6mn + m + 1 \in I_2(d_k) \cap I_2(d_{k'})$.

Let us now consider the case where $uv$ is not an edge. In particular, let us show that $I_1(u) < I_1(v) < I_2(u) < I_2(v)$, where $[u, u'] < [v, v']$ means that $u' < v$.

**Case** $u = x_i$ and $v = a_k$, where $l(k) = i$:
I_1(a_k) < I_1(x_i) < I_2(a_k) < I_2(x_i).

Case u = x_i and v = d_k, where r(k) = i:
I_1(x_i) < I_1(d_k) < I_2(x_i) < I_2(d_k).

Case u = a_k and v = b_k:
I_1(a_k) < I_1(b_k) < I_2(a_k) < I_2(b_k).

Case u = b_k and v = c_k:
I_1(b_k) < I_1(c_k) < I_2(b_k) < I_2(c_k).

Case u = c_k and v = d_k:
I_1(c_k) < I_1(d_k) < I_2(c_k) < I_2(d_k).

Therefore, we have a valid 2-interval representation of \( Subd_4(G) \) and this representation can obviously be constructed in linear time.

\[ \text{□} \]

4.2 Unit 3-Interval Graphs

**Theorem 6** Given any graph \( G \), \( Subd_4(G) \) is a unit 3-interval graph and a unit 3-interval representation for it can be constructed in linear time.

**Proof** Recall that each edge \( e_k = x_i.x_j \) of \( G \) where \( i < j \), corresponds to the path \((x_i, a_k, b_k, x_j)\) in \( Subd_2(G) \). We define \( I_1 \), \( I_2 \) and \( I_3 \) as follows (see also Fig. 3). Here again, \( 1 \leq i \leq n \) and \( 1 \leq k \leq m \).

\[
\begin{align*}
I_1(a_k) &= [m(l(k) - 1) + k, m(l(k) - 1) + m^2 + k] \\
I_1(x_i) &= [mi + m^2 + k + 1, ml(k) + 2m^2 + k + 1] \\
I_2(b_k) &= [mr(k) + 3m^2 + k + 2, mr(k) + 4m^2 + k + 2] \\
I_2(x_i) &= [mi + 4m^2 + m + 3, ml(k) + 5m^2 + m + 3] \\
I_2(a_k) &= [ml(k) + 5m^2 + m + k + 3, ml(k) + 6m^2 + m + k + 3] \\
I_3(b_k) &= [ml(k) + 6m^2 + m + k + 4, ml(k) + 7m^2 + m + k + 4] \\
I_3(a_k) &= [15m^2, 16m^2] \\
I_3(x_i) &= [17m^2, 18m^2]
\end{align*}
\]

Clearly, this representation can be constructed in linear time. It is easy to see that \( I_1 \), \( I_2 \) and \( I_3 \) assign intervals of length \( m^2 \) to the vertices of \( Subd_2(G) \). Using similar arguments as in the proof of the previous theorem, the validity of this unit 3-interval representation can be easily verified. One can also easily check in the figure that this is a valid unit 3-interval representation of \( Subd_2(G) \). \[ \text{□} \]
4.3 3-Track Graphs

**Theorem 7** Given any graph $G$, $\text{Subd}_2(G)$ is a 3-track graph and a 3-track representation for it can be constructed in linear time.

**Proof** We define a 3-track representation for $\text{Subd}_2(G)$ as follows (see also Fig. 4). For $1 \leq i \leq n$ and $1 \leq k \leq m$:

\[
\begin{align*}
I_1(a_k) &= [0, l(k)] \\
I_1(x_i) &= [i + 1, n + i + 1] \\
I_1(b_k) &= [n + r(k) + 2, 2n + 3] \\
I_2(x_i) &= [0, i] \\
I_2(a_k) &= [l(k) + 1, n + k] \\
I_2(b_k) &= [n + k + 1, m + n + 2] \\
I_3(a_k) &= [0, m + 1 - k] \\
I_3(b_k) &= [m + 2 - k, m + r(k)] \\
I_3(x_i) &= [m + i + 1, m + n + 2]
\end{align*}
\]

This representation can be constructed in linear time and it can be verified easily, either by arguments as before or using Fig. 4, that this is a valid 3-track representation of $\text{Subd}_2(G)$. □

4.4 Unit 4-Track Graphs

**Theorem 8** Given any graph $G$, $\text{Subd}_2(G)$ is a unit 4-track graph and a unit 4-track representation for it can be constructed in linear time.

**Proof** We define $I_1$, $I_2$, $I_3$ and $I_4$ as follows (see also Fig. 5). As usual, $1 \leq i \leq n$ and $1 \leq k \leq m$.

\[
\begin{align*}
I_1(a_k) &= [m(l(k) - 1) + k, m(l(k) - 1) + m^2 + k] \\
I_1(x_i) &= [mi + m^2 + 1, mi + 2m^2 + 1]
\end{align*}
\]
Fig. 5 The unit 4-track representation of $\text{Subd}_2(G)$

$I_1(b_k) = [2m^2 + mr(k) + k + 1, 3m^2 + mr(k) + k + 1]$
$I_2(b_k) = [m(r(k) - 1) + k, m(r(k) - 1) + m^2 + k]$
$I_2(x_i) = [mi + m^2 + 1, mi + 2m^2 + 1]$
$I_2(a_k) = [2m^2 + ml(k) + k + 1, 3m^2 + ml(k) + k + 1]$
$I_3(a_k) = [k, k + m^2]$
$I_3(b_k) = [k + m^2 + 1, k + 2m^2 + 1]$
$I_3(x_i) = [5m^2, 6m^2]$
$I_4(b_k) = [k, k + m^2]$
$I_4(a_k) = [k + m^2 + 1, k + 2m^2 + 1]$
$I_4(x_i) = [5m^2, 6m^2]$

This representation can be constructed in linear time and it is easy to verify that $I_1$, $I_2$, $I_3$ and $I_4$ assign intervals of length $m^2$ to the vertices of $\text{Subd}_2(G)$. One can also easily check in Fig. 5, or verify using arguments as before, that this is a valid unit 4-track representation of $\text{Subd}_2(G)$. □

4.5 Unit 2-Circular Interval Graphs

**Theorem 9** Given any graph $G$, $\text{Subd}_2(G)$ is a unit 2-circular interval graph and a unit 2-circular interval representation for it can be constructed in linear time.

**Proof** Let $C$ be a circle of circumference $6m^2 + 2m + 4$. The mappings $I_1$ and $I_2$, which map $V(G)$ to arcs on this circle, are defined as follows (see also Fig. 6).

$I_1(b_k) = [ml(k) + 6m^2 + m + k + 4, m(l(k) - 1) + m^2 + k]$  
$I_1(a_k) = [m(l(k) - 1) + m^2 + k + 1, m(l(k) - 1) + 2m^2 + k + 1]$  
$I_1(x_i) = [mi + 2m^2 + 2, mi + 3m^2 + 2]$
Fig. 6 The unit 2-circular interval representation of $\text{Subd}_2(G)$

Fig. 7 The 2-circular track representation of $\text{Subd}_2(G)$

$I_2(b_k) = [mr(k) + 3m^2 + k + 2, mr(k) + 4m^2 + k + 2]$

$I_2(x_i) = [mi + 4m^2 + m + 3, mi + 5m^2 + m + 3]$

$I_2(a_k) = [ml(k) + 5m^2 + m + k + 3, ml(k) + 6m^2 + m + k + 3]$

Note that this representation is almost the same as the unit 3-interval representation given for $\text{Subd}_2(G)$ in the proof of Theorem 6, the only difference being that $I_1(b_k)$ and $I_3(b_k)$ have now been fused to form $I_1(b_k)$ of the unit 2-circular interval representation being constructed. This representation can be constructed in linear time and it is easy to verify that the arcs have length $m^2$. One can also easily check in the figure, or argue as in the proof of Theorem 5, that this is a valid unit 2-circular interval representation of $\text{Subd}_2(G)$. □

4.6 2-Circular Track Graphs

**Theorem 10** Given any graph $G$, $\text{Subd}_2(G)$ is a 2-circular track graph and a 2-circular track representation for it can be constructed in linear time.

**Proof** We define a 2-circular track representation using circles $C_1$ and $C_2$, each having circumference at least $3n + 1$, and mappings $I_1$ and $I_2$ defined as follows (see also Fig. 7).

$I_1(x_i) = [i, i + n]$

$I_1(a_k) = [l(k) + n + 1, l(k) + 2n]$

$I_1(b_k) = [l(k) + 2n + 1, r(k) - 1]$

$I_2(x_i) = [i, i + n]$

$I_2(b_k) = [r(k) + n + 1, r(k) + 2n]$

$I_2(a_k) = [r(k) + 2n + 1, l(k) - 1]$

Clearly, this representation can be constructed in linear time, and as before, it can be checked that the circles $C_1$ and $C_2$ together with the mappings $I_1$ and $I_2$ form a valid 2-circular track representation of $\text{Subd}_2(G)$. □
Valiant [26] has shown that every planar graph of degree at most 4 can be drawn on a grid of linear size such that the vertices are mapped to points of the grid and the edges to piecewise linear curves made up of horizontal and vertical line segments whose endpoints are also points of the grid. It is immediately clear that every planar graph $G$ has a subdivision $G'$ that is an induced subgraph of a grid graph such that each edge of $G$ corresponds to a path of length at most $O(|V(G)|^2)$ (see Fig. 8).

Note that here, some paths have even length and some have odd length. An even subdivision (resp. odd subdivision) of $G$ is a graph obtained from $G$ by subdividing each edge $e$ of $G$ an even (resp. odd) number of times, and at most $|V(G)|^{O(1)}$ times.

Let $R(w, h)$ be the rectangular grid of height $h$ and width $w$. A path in $R(w, h)$ that contains only vertices from one row of the grid is called a horizontal grid-path and one that contains vertices from only one column is called a vertical grid-path. We denote by $R'(w, h)$ the graph obtained by subdividing each edge of $R(w, h)$ once and by adding paths of length 3 between the newly introduced vertices as shown in Fig. 9.

**Lemma 1** Any planar graph $G$, on $n$ vertices and of maximum degree 4, has an even subdivision that is an induced subgraph of $R'(w, h)$ for some values of $w$ and $h$ that are linear in $n$. 
Fig. 10 Modifying the paths in $H'$ to obtain $H''$: A part of the graph in Fig. 8 is shown.

Proof Let $H$ be the subdivision of $G$ that is an induced subgraph of the grid $R(w, h)$. Let $P_e$ denote the path in $H$ corresponding to an edge $e$ in $G$. We assume that $P_e$ is the union of horizontal and vertical grid-paths of length at least 5. We now transform the grid $R(w, h)$ into $R'(w, h)$ by subdividing each edge once and by adding paths of length 3 between the newly introduced vertices as explained before. Clearly, a 1-subdivision of $H$, which we shall denote by $H'$, is an induced subgraph of $R'(w, h)$. It could be noted that $H'$ is an odd subdivision of $G$, although we do not use this fact in our proof. (This can be seen as follows. Suppose $P'_e$ is the path in $H'$ corresponding to an edge $e$ in $G$. Clearly, $P'_e$ is a subdivision of $P_e$ with the number of vertices in $P'_e$ being equal to the sum of the number of vertices in $P_e$ and the number of edges in $P_e$. But this sum is always odd as the number of vertices in $P_e$ is exactly one more than the number of edges in $P_e$. Therefore, $H'$ is an odd subdivision of $G$.) Let $P'_e$ denote the path in $H'$ corresponding to an edge $e$ of $G$. Note that $P'_e$ consists of 1-subdivisions of vertical and horizontal grid-paths.

For every edge $e$ of $G$, we do the following procedure on $P'_e$ in $H'$ to obtain a new graph $H'':$ we replace one of the subdivided horizontal or vertical grid-paths that make up $P'_e$ to obtain $P''_e$ which has an even number of vertices as shown in Fig. 10. The procedure makes sure that the new graph $H''$ so obtained is an even subdivision of $G$ and is also an induced subgraph of $R'(w, h)$.

Lemma 2 For any $w$ and $h$ the graph $R'(w, h)$ is both a unit 2-interval graph as well as a unit 3-track graph. Thus since those classes are closed under taking induced subgraphs, they also contain the induced subgraphs of $R'(w, h)$.

Proof We construct a graph $Q(w, l)$ which is defined so that by making $l$ large enough, we can find an induced subgraph of $Q(w, l)$ that is isomorphic to $R'(w, h)$. The graph more or less looks like a grid that has been tilted 45 degrees, with some extra vertices added. The vertex set consists of three types of vertices: the “$X$” vertices that are arranged as layers of vertices placed one above the other with each layer having $w$ vertices (there are $l + 1$ “odd” layers, denoted by $X^o$, and $l$ “even” layers, denoted by $X^e$, arranged in an alternating fashion), the “$a$” vertices that connect an
odd layer to the even layer immediately above it, the “$b$” vertices that connect an even layer to the odd layer immediately above it, and the “$c$” and “$d$” vertices that form a path of length 3 connecting an $a$ vertex with a $b$ vertex. We shall now define the graph formally.

$$V(Q(w, l)) = X^o \cup X^e \cup A \cup B \cup C \cup D$$

where $X^o = \{x_1^o, \ldots, x_{w(l+1)}^o\}$, $X^e = \{x_1^e, \ldots, x_w^e\}$, $A = \{a_1, \ldots, a_{2wl}\}$, $B = \{b_1, \ldots, b_{2wl}\}$, $C = \{c_1, \ldots, c_{2wl}\}$ and $D = \{d_1, \ldots, d_{2wl}\}$.

$$E(Q(w, l)) = \bigcup_{i=1}^{wl} \{x_i^oa_{2i-1}, x_i^oa_{2i}\} \cup \{x_w^ob_1\} \cup \bigcup_{i=w+2}^{w(l+1)} \{x_i^ob_2(i-w)-2, x_i^ob_2(i-w)-1\}$$

$$\cup \{x_i^oa_1\} \cup \bigcup_{i=2}^{w(l+1)} \{x_i^ea_{2i-2}, x_i^ea_{2i-1}, x_i^eb_{2i-1}, x_i^eb_{2i}\}$$

$$\cup \bigcup_{i=1}^{2wl-1} \{a_ii, c_ii, d_ii, d_ii+1\}$$

Let $Q'(w, l)$ be the graph induced in $Q(w, l)$ by the vertices in $V(Q(w, l)) \setminus \bigcup_{i=1}^l \{a_{2wi}, b_{2wi}\}$. Figure 11 shows a drawing of the graph $Q'(w, l)$. It can be seen that $R'(w, h)$ is an induced subgraph of $Q'(w, \lceil \frac{w+h}{2} \rceil - 1)$ (see Fig. 12) and therefore also of $Q(w, \lceil \frac{w+h}{2} \rceil - 1)$. Thus, to show that for any $w$ and $h$, $R'(w, h)$ is a unit 2-interval graph and a unit 3-track graph, we only need to show that $Q(w, l)$ for any $w$ and $l$ is both a unit 2-interval graph as well as a unit 3-track graph.

We construct a unit 2-interval representation $f$ for $Q(w, l)$ as follows (see also Fig. 13).

$$I_1(a_i) = [2i, 2i + 6n]$$
$$I_1(c_i) = [2i + 6n + 2, 2i + 12n + 2]$$
$$I_1(x_i^e) = [4i + 6n + 1, 4i + 12n + 1]$$
$$I_2(a_i) = [2i + 12n + 4, 2i + 18n + 4]$$
$$I_1(x_i^o) = [4i + 6n - 1, 4i + 12n - 1]$$
$$I_1(d_i) = [2i, 2i + 6n]$$
$$I_1(b_i) = [18n + 6 - 2i, 24n + 6 - 2i]$$
$$I_2(d_i) = [24n + 6 - 2i, 30n + 6 - 2i]$$
$$I_2(b_i) = [30n + 10 - 2i, 36n + 10 - 2i]$$
$$I_2(x_i^e) = [24n + 9 - 4i, 30n + 9 - 4i]$$
$$I_2(x_i^o) = [24n + 11 - 4i + 4w, 36n + 11 - 4i + 4w]$$
$$I_2(c_i) = [30n + 8 - 2i, 36n + 8 - 2i]$$

It is easy to verify that all the intervals have length $6n$. Then one can verify either by arguments similar to those used in the proof of Theorem 5 or using the figure that this is a valid unit 2-interval representation of $Q(w, l)$. Note that this construction is
Fig. 11  The graph $Q'(w, l)$, which is essentially the same as the graph $Q(w, l)$, the only difference between them being that the vertices in $\bigcup_{i=1}^{l} \{a_{2wi}, b_{2wi}\}$ are not present in $Q'(w, l)$. Not all the vertices are labelled due to lack of space. The black circular nodes correspond to vertices in $X^o \cup X^e$, the white circular ones to vertices in $A$ and the gray circular ones to vertices in $B$. The square white nodes are for vertices in $C$ and the square gray nodes for vertices in $D$. 
Fig. 12  Picture showing $R'(6, 4)$ and an induced subgraph of $Q'(6, 4)$. In general, $R'(w, h)$ is an induced subgraph of $Q'(w, l)$ where $l = \lceil \frac{w+h}{2} \rceil - 1$

Fig. 13  Unit 2-interval representation of $Q(w, l)$

slightly more involved than the previous ones. Here the second blocks of $x_i^e$ and $x_i^o$ are slightly shifted. This is due to the fact that each $x_i^e$ is adjacent to every $b_j$, except $b_i$ and $b_{i+1}$, and that each $x_i^o$ is adjacent to every $b_j$, except $b_{i-w-1}$ and $b_{i-w}$. We construct now a unit 3-track representation for $Q(w, l)$ as follows (see also Fig. 14).

$$I_1(b_i) = [i, i + 2n]$$
$$I_1(x_i^e) = [2i + 2n + 1, 2i + 4n + 1]$$
$$I_1(x_i^o) = [2i - 2w + 2n, 2i - 2w + 4n]$$
$$I_1(a_i) = [i + 4n + 4, i + 6n + 4]$$
$$I_1(c_i) = [i + 2n + 3, i + 4n + 3]$$
$$I_1(d_i) = [i + 4n + 4, i + 6n + 4]$$
$$I_2(a_i) = [i, i + 2n]$$
$$I_2(x_i^e) = [2i + 2n, 2i + 4n]$$
$$I_2(x_i^o) = [2i + 2n + 1, 2i + 4n + 1]$$
Fig. 14 Unit 3-track representation of $Q(w,l)$

$I_2(b_i) = [i + 2w + 4n + 4, i + 2w + 6n + 4]$

$I_2(d_i) = [i + 2w + 2n + 4, i + 2w + 4n + 4]$

$I_2(c_i) = [i + 2w + 4n + 5, i + 2w + 6n + 5]$

$I_3(x_i^e) = [2i, 2i + 4n]$

$I_3(x_i^o) = [i + 4n + 2, i + 8n + 2]$

$I_3(a_i) = [i + 8n + 3, i + 12n + 3]$

$I_3(b_i) = [8n + 3 - i, 12n + 3 - i]$

$I_3(d_i) = [4n + 1 - i, 8n + 1 - i]$

$I_3(x_i^e) = [12n + 5 - 2i, 16n + 5 - 2i]$

It is easy to verify that all the intervals have length 4n. Then one can either verify using arguments or check in the figure that this is a valid unit 3-track representation of $Q(w,l)$. Note that here also one has to be careful with the $b_i$’s that $x_i^e$ and $x_i^o$ have to avoid. □

**Theorem 11** MAXIMUM CLIQUE is NP-complete for unit 2-interval and unit 3-track graphs.
Proof It is known that the MAXIMUM INDEPENDENT SET problem is NP-complete even when restricted to planar graphs of degree at most 3 [10]. It is folklore that the instance \((G, k)\) of MAXIMUM INDEPENDENT SET is equivalent to an instance \((H, k + k')\), where \(H\) is an even subdivision of \(G\) with \(|V(G)| + 2k'\) vertices. Thus according to Lemma 1, MAXIMUM INDEPENDENT SET is NP-complete on the class of induced subgraphs of \(R'(w, h)\). MAXIMUM CLIQUE is thus NP-complete on the class of induced subgraphs of \(R'(w, h)\). Finally by Lemma 2 this class of graphs is contained in unit 2-interval and unit 3-track graphs. MAXIMUM CLIQUE is thus NP-complete on these classes. \(\square\)

6 Complements of \(t\)-Interval Graphs

Very recently, Jiang and Zhang studied the class of complements of \(t\)-interval graphs [17]. In particular they proved that MINIMUM (INDEPENDENT) DOMINATING SET parameterized by the solution size is in W[1] for co-2-interval graphs, and they proved that MINIMUM DOMINATING SET is W[1]-hard for co-3-track graphs.

Following the same line of proof as for Theorem 4 we can prove the following APX-hardness results, for this kind of graph classes.

Theorem 12

(i) MINIMUM VERTEX COVER is APX-complete in co-2-interval graphs, and the complement classes of all the classes of Theorem 3.

(ii) For any graph \(G\), Subd\(_3\)(\(G\)) is a co-unit-2-interval, a co-3-track, a co-unit-4-track, and a co-2-circular track graph, and MINIMUM (INDEPENDENT) DOMINATING SET is APX-hard for these classes of graphs.

Proof The first item follows from the fact that MINIMUM VERTEX COVER is 2-approximable [23] and the first item of the following theorem.

Theorem 13 ([7])

(i) MINIMUM VERTEX COVER is APX-complete when restricted to 2\(k\)-subdivisions of 3-regular graphs for any fixed integer \(k \geq 0\).

(ii) The problems MINIMUM DOMINATING SET, and MINIMUM INDEPENDENT DOMINATING SET are APX-complete when restricted to 3\(k\)-subdivisions of graphs of degree at most 3, for any fixed integer \(k \geq 0\).

For the second item the constructions are described in Figs. 15–18. Here an edge \(e_k = x_i x_j\) of \(G\), with \(i < j\), corresponds to a path \((x_{l(k)}, a_k, b_k, c_k, x_{r(k)})\) of Subd\(_3\)(\(G\)). Then Theorem 13(ii) clearly implies the APX-hardness. \(\square\)
7 Concluding Remarks

In our approximation algorithm (as in the previous algorithms) we assume that we are given an interval representation. We wonder what we can do if we are not given such a representation.

Open question. Can MAXIMUM (WEIGHTED) CLIQUE be polynomially $c(t)$-approximated in $t$-interval graphs, for some function $c$, if we are not given an interval representation?

This would be the case if there is an algorithm that computes, given a $t$-interval graph $G$, a $c(t)$-interval representation of $G$. Actually even when we are given a representation, the approximation ratio might be far from the optimal.
Open question. Does there exist an approximation algorithm for MAXIMUM (WEIGHTED) CLIQUE in $t$-interval graphs with a better approximation ratio?

Let us call $f(t)$ the better ratio a polynomial algorithm can achieve on $t$-interval graphs (actually $f(t)$ should be an infimum). For any graph $G$ on $n$ vertices, it is easy to construct an $n$-interval representation of $G$. Thus since for any $\epsilon > 0$, one cannot $O(n^{1-\epsilon})$-approximate the MAXIMUM CLIQUE unless $P = NP$ [28], we certainly have $f(t) = \Omega(t^{1-\epsilon})$.

The current status of the complexity of the MAXIMUM CLIQUE problem for the various classes of multiple interval graphs that were studied are shown in Table 1 (where “Unres.” stands for “Unrestricted”). The entries marked “NP-c” and “?” in this table clearly imply the following questions.

Open question. Is MAXIMUM CLIQUE for unit 2-interval graphs, unit 3-track graphs or unit 3-circular track graphs APX-hard, or does it admit a PTAS?
Table 1  The complexity of the MAXIMUM CLIQUE problem for various classes of multiple interval graphs

| t | t-track | t-interval | t-circular track | t-circular interval |
|---|---------|-----------|------------------|--------------------|
| Unit | Unres. | Unit | Unres. | Unit | Unres. | Unit | Unres. |
| 1 | P folklore | P folklore | folklore | folklore | P folklore | [11] | [11] | [11] | [11] |
| 2 | P [20] | P Thm. 11 | NP-c Thm. 5 | APX-c Thm. 10 | APX-c Thm. 9 |
| 3 | NP-c Thm. 11 | APX-c Thm. 7 | APX-c Thm. 6 | APX-c Thm. 5 | APX-c Thm. 11 | Thm. 9 |
| ≥ 4 | APX-c Thm. 8 | APX-c Thm. 7 | APX-c Thm. 6 | APX-c Thm. 5 | APX-c Thm. 8 | Thm. 9 |

Open question. Is MAXIMUM CLIQUE for unit 2-circular track graphs Polynomial or NP-complete?

Koenig [20] explains that 2-track graphs have a polynomial-time algorithm for MAXIMUM CLIQUE because for any 2-track representation of a clique, there is a transversal of size 2 (i.e. two points such that for every vertex, at least one of its intervals contains one of these points). We note that this is not true for unit 2-circular track graphs as the complete graph on 5 vertices has a unit 2-circular track representation in which each circular track induces a cycle on 5 vertices. This representation clearly does not have a transversal of size 2.

Acknowledgements  This work was partially supported by the grant ANR-09-JCJC-0041.

References

1. Asinowski, A., Cohen, E., Golumbic, M.C., Limouzy, V., Lipshteyn, M., Stern, M.: Vertex intersection graphs of paths on a grid. J. Graph Algorithms Appl. 16(2), 129–150 (2012)
2. Aumann, Y., Lewenstein, M., Melamud, O., Pinter, R.Y., Yakhini, Z.: Dotted interval graphs and high throughput genotyping. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ‘05, pp. 339–348. Society for Industrial and Applied Mathematics, Philadelphia (2005)
3. Bar-Yehuda, R., Halldórsson, M.M., Naor, J.S., Shachnai, H., Shapira, I.: Scheduling split intervals. SIAM J. Comput. 36, 1–15 (2006)
4. Berman, P., Fujito, T.: On approximation properties of the independent set problem for degree 3 graphs. In: Algorithms and Data Structures. Lecture Notes in Computer Science, vol. 955, pp. 449–460. Springer, Berlin (1995)
5. Butman, A., Hermelin, D., Lewenstein, M., Rawitz, D.: Optimization problems in multiple-interval graphs. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’07, pp. 268–277. Society for Industrial and Applied Mathematics, Philadelphia (2007)
6. Cabello, S., Cardinal, J., Langerman, S.: The clique problem in ray intersection graphs. In: Algorithms—ESA 2012. Lecture Notes in Computer Science, vol. 7501, pp. 241–252. Springer, Berlin (2012)
7. Chlebík, M., Chlebíková, J.: The complexity of combinatorial optimization problems on d-dimensional boxes. SIAM J. Discrete Math. 21(1), 158–169 (2007)
8. Crochemore, M., Hermelin, D., Landau, G.M., Vialette, S.: Approximating the 2-interval pattern problem. In: Algorithms—ESA 2005. Lecture Notes in Computer Science, vol. 3669, pp. 426–437. Springer, Berlin (2005)
9. Fellows, M., Hermelin, D., Rosamond, F., Vialette, S.: On the parameterized complexity of multiple-interval graph problems. Theor. Comput. Sci. 410, 53–61 (2009)
10. Garey, M.R., Johnson, D.S.: Rectilinear Steiner tree problem is NP-complete. SIAM J. Appl. Math. 6, 826–834 (1977)
11. Gavril, F.: Algorithms for a maximum clique and a maximum independent set of a circle graph. Networks 3, 261–273 (1973)
12. Gavril, F.: Maximum weight independent sets and cliques in intersection graphs of filaments. Inf. Process. Lett. 73(5–6), 181–188 (2000)
13. Hochbaum, D.S., Levin, A.: Cyclic scheduling and multi-shift scheduling: complexity and approximation algorithms. Discrete Optim. 3(4), 327–340 (2006)
14. Hsu, W.L.: Maximum weight clique algorithms for circular-arc graphs and circle graphs. SIAM J. Comput. 14(1), 224–231 (1985)
15. Jiang, M.: On the parameterized complexity of some optimization problems related to multiple-interval graphs. In: Proceedings of CPM 2010. Lecture Notes in Computer Science, vol. 6129, pp. 125–137. Springer, Berlin (2010)
16. Jiang, M.: Clique in 3-track interval graphs is APX-hard (2012). arXiv:1204.2202 [cs.CC]
17. Jiang, M., Zhang, Y.: Parameterized complexity in multiple-interval graphs: domination, partition, separation, irredundancy. Theor. Comput. Sci. 461, 27–44 (2012). doi:10.1016/j.tcs.2012.01.025
18. Kaiser, T.: Transversals of d-intervals. Discrete Comput. Geom. 18, 195–203 (1997)
19. Kammer, F., Tholey, T.: Approximation algorithms for intersection graphs. Algorithmica (2012). doi:10.1007/s00453-012-9671-1
20. König, F.G.: Sorting with objectives. Ph.D. thesis, Technische Universität Berlin (2009)
21. Kratochvíl, J., Nešetřil, J.: INDEPENDENT SET and CLIQUE problems in intersection-defined classes of graphs. Comment. Math. Univ. Carol. 31(1), 85–93 (1990)
22. Middendorf, M., Pfeiffer, F.: The max clique problem in classes of string-graphs. Discrete Math. 108, 365–372 (1992)
23. Monien, B., Speckenmeyer, E.: Ramsey numbers and an approximation algorithm for the vertex cover problem. Acta Inform. 22, 115–123 (1985)
24. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. 43(3), 425–440 (1991)
25. Trotter, W.T., Harary, F.: On double and multiple interval graphs. J. Graph Theory 3(3), 205–211 (1979)
26. Valiant, L.G.: Universality considerations in VLSI circuits. IEEE Trans. Comput. 30(2), 135–140 (1981)
27. West, D.B., Shmoys, D.B.: Recognizing graphs with fixed interval number is NP-complete. Discrete Appl. Math. 8(3), 295–305 (1984)
28. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proceedings of the 38th ACM Symposium on Theory of Computing, STOC ’06, pp. 681–690. ACM, New York (2006)