Optimization Problems in Dotted Interval Graphs

Danny Hermelin[1], Julián Mestre[2], and Dror Rawitz[3]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany.
hermelin@mpi-inf.mpg.de
[2] School of Information Technologies, The University of Sydney, Australia.
mestre@it.usyd.edu.au
[3] School of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel.
rawitz@eng.tau.ac.il

Abstract. The class of *D-dotted interval* (D-DI) graphs is the class of intersection graphs of arithmetic progressions with jump (common difference) at most D. We consider various classical graph-theoretic optimization problems in D-DI graphs of arbitrarily, but fixed, D.
We show that Maximum Independent Set, Minimum Vertex Cover, and Minimum Dominating Set can be solved in polynomial time in this graph class, answering an open question posed by Jiang *(Inf. Processing Letters, 98(1):29–33, 2006)*. We also show that Minimum Vertex Cover can be approximated within a factor of $(1+\varepsilon)$ for any $\varepsilon > 0$ in linear time. This algorithm generalizes to a wide class of deletion problems including the classical Minimum Feedback Vertex Set and Minimum Planar Deletion problems.
Our algorithms are based on classical results in algorithmic graph theory and new structural properties of D-DI graphs that may be of independent interest.

1 Introduction

A *dotted interval* $I(s,t,d)$ is an arithmetic progression $\{s, s+d, s+2d, \ldots, t\}$, where s, t and d are positive integers, and the *jump* d divides $t-s$. When $d=1$, the dotted interval $I(s,t,d)$ is simply the interval $[s,t]$ over the positive integer line. This paper is mainly concerned with dotted interval graphs. A *dotted interval graph* is an intersection graph of dotted intervals. Each vertex v is associated a dotted interval I_v and we have an edge (u,v) if $I_u \cap I_v \neq \emptyset$. If the jumps of all intervals are at most D, we call the graph *D-dotted-interval* or D-DI for short. See Figure 1 for an example.

Dotted interval graphs were introduced by Aumann *et al.* [2] in the context of high throughput genotyping. They used dotted intervals to model microsatellite polymorphisms which are used in a genotyping technique called microsatellite genotyping. The respective genotyping problem translates to Minimum Coloring in D-DI graphs of small D. Aumann *et al.* [2] showed that Minimum Coloring in D-DI graphs is NP-hard even for $D = 2$. They also provided

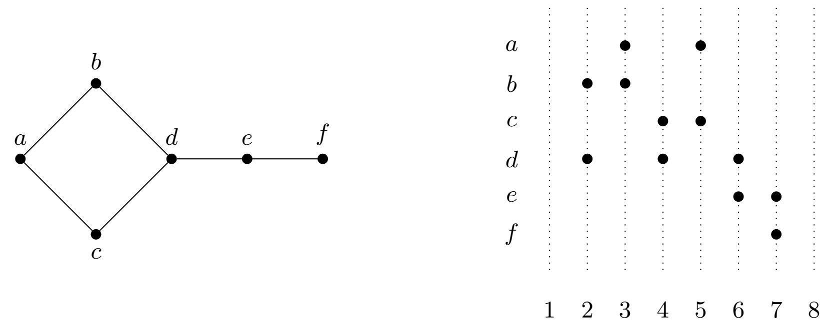

Fig. 1. Example of a 2-DI graph: On the right we have the 2-DI representation of the graph on the left. Notice that graph is clearly not an interval graph since we have hole of length 4.

a $\frac{3}{2}$-approximation algorithm for Minimum Coloring in 2-DI graphs, and a $(\frac{7D}{8} + \Theta(1))$-approximation algorithm for general fixed $D \geq 2$. This algorithm was later improved by Jiang [17], and subsequently also by Yanovsky [21]. The current best approximation ratio for Minimum Coloring is $\frac{2D+4}{3}$ [21].

Since a dotted interval with jump 1 is a regular interval, dotted interval graphs form a natural generalizations of the well-studied class of interval graphs. Interval graphs have been extensively researched in the graph-theoretic community, in particular from the algorithmic viewpoint, because many real-life problems translate to classical graph-theoretic problems in interval graphs, and because its rich structure allows in many cases designing efficient algorithms for these problems. Substantial research effort has been devoted into generalizing such algorithms to larger classes of graphs. Examples include algorithms proposed for circular arc graphs [13, 15], disc graphs [11, 16, 19, 20], rectangle graphs [1, 5, 9], multiple-interval graphs [4, 8], and multiple-subtree graphs [14].

In this paper we study the computational complexity of classical graph-theoretic optimization problems in D-DI graphs. Note that as any graph G is a D-DI graph for large enough D [2], we are interested in studying D-DI graphs for small D; more precisely, we assume $D = O(1)$. Apart from the Minimum Coloring problem, Aumann *et al.* [2] also considered the Maximum Clique problem in D-DI graphs, and showed that this problem is fixed parameter tractable with respect to D. Jiang [17] studied the problem of Maximum Independent Set in D-DI graphs. He presented a simple $\frac{3}{2}$-approximation algorithm for 2-DI graphs, and a $(\frac{5D}{6} + O(\log d))$-approximation algorithm or D-DI graphs. The question of whether Maximum Independent Set in D-DI graphs, for constant D, is NP-hard was left open by Jiang. He also pointed out that the complexity of other classical graph theoretical problems, such as Minimum Vertex Cover and Minimum Dominating Set, remain open in D-DI graphs.

In this paper we focus mainly on three classical graph-theoretic optimization problems: Maximum Independent Set, Minimum Dominating Set, and Minimum Vertex Cover. We present an $O(Dn^D)$-time algorithm for Maxi-

MUM INDEPENDENT SET and MINIMUM VERTEX COVER in D-DI graphs with n vertices, and give an $O(D^2 n^{O(D^2)})$-time algorithm for MINIMUM DOMINATING SET. Thus, we show that both these problems are polynomial-time solvable in D-DI graphs for fixed D. It is interesting to note that a similar situation occurs in circular-arc graphs, which also generalize interval graphs, where MAXIMUM INDEPENDENT SET and MINIMUM DOMINATING SET can be solved in linear time [15] and MINIMUM COLORING is NP-hard [12]. (However, Aumann *et al.* [2] show that there is a 2-DI graph that is not a circular arc graph, and that for every $D \geq 1$, there is a circular arc graph that is not a D-DI graph.) We also present a linear-time $(1+\varepsilon)$-approximation algorithm for MINIMUM VERTEX COVER in D-DI graphs. This algorithm can be generalized to a wide range of deletion problems which include among many the classical MINIMUM FEEDBACK VERTEX SET and MINIMUM PLANAR DELETION problems. We assume that the D-DI representation of the input graph is given to us.

2 Preliminaries

2.1 Definitions and Notation

For $i, j \in \mathbb{Z}$ such that $i < j$, we define $[i, j] := \{i, i+1, \ldots, j-1, j\}$.

Given a dotted interval $I = \{s, s+d, s+2d, \ldots, t\}$, we denote its starting and finishing points by $s(I)$ and $t(I)$, respectively. The jump of I is denoted by $d(I)$, and the *offset* of I is defined as $o(I) := s(I) \bmod d(I)$.

Given a set of dotted intervals $\mathcal{I} = \{I_1, \ldots, I_n\}$, we assume that the intervals are ordered by starting point, namely that $s(I_i) \leq s(I_{i+1})$, for every i. Dotted intervals with the same starting point are ordered arbitrarily. Given a dotted interval I_i, we define $\mathcal{I}_{<i} := \{I_j : j < i\}$. Given a point p, and a set of dotted intervals $\mathcal{S} \subseteq \mathcal{I}$, let $\mathcal{S}_p \subseteq \mathcal{S}$ contain the dotted intervals that start at or before p and end at or to the right of p, namely $\mathcal{S}_p := \{I \in \mathcal{S} : p \in [s(I), t(I)]\}$. (Note that it is possible $I \in \mathcal{S}_p$ and $p \notin I$.)

Let $G = (V, E)$ be an undirected graph; for any subset $A \subseteq V$, we use $G[A] = (A, \{(u,v) \in E : u \in A, v \in A\})$ to denote the graph induced by A. Let $w : V \to \mathbb{R}^+$ be a vertex weight function; for any $A \subseteq V$, we use the shorthand notation $w(A) = \sum_{u \in A} w(u)$. A subset $A \subseteq V$ is said to be *independent* if no two vertices in A are connected by an edge in E; the MAXIMUM INDEPENDENT SET problem is to find an independent set of maximum weight. A subset $A \subseteq V$ is said to be *dominating* if every vertex $v \subseteq V$ has at least one neighbor in A; the MINIMUM DOMINATING SET problem is to find a dominating set of minimum weight. A subset $A \subseteq V$ is said to be a *vertex cover* if every edge in E has at least one endpoint in A; the MINIMUM VERTEX COVER is to find a vertex cover of minimum weight.

2.2 Simple Observations

Let $\mathcal{I}$ be a representation of a D-DI graph G, and denote $\ell(D) = \operatorname{lcm}\{2, \ldots, D\}$, the least common multiple of the numbers $2, \ldots, D$.

Observation 1 *Let $I, J \in \mathcal{I}$ be two dotted intervals, and let $i \in I, J$. If $t(I), t(J) \geq i + \ell(D)$, then $i + \ell(D) \in I, J$.*

Given a dotted interval I and an integer i, let

$$I(i) = \{j : j \in I \text{ and } j < i\} \cup \{j - \ell(i) : j \in I \text{ and } j \geq i + \ell(D)\} \ .$$

Namely $I(i)$ is obtained from I by removing the points in $I \cap [i, i+\ell(D)-1]$ and gluing the two parts of I back together by moving the points in $I \cap [i+\ell(D), t(I)]$ to the left. Let $\mathcal{I}(i) = \{I(i) : I \in \mathcal{I}\}$.

Observation 2 *Let i be an integer such that $[i, i + 2\ell(D) - 1]$ does not contain any starting or finishing point of a dotted interval from $\mathcal{I}$. Then $\mathcal{I}(i)$ is also a representation of G.*

Given an arbitrary D-DI representation, we could apply the above observation repeatedly to obtain an equivalent representation where all intervals with length more than $2\ell(D)$ contain at least one end-point of some dotted interval.

Observation 3 *Any D-DI graph G has a representation $\mathcal{I}$ such that*

$$\max_{I \in \mathcal{I}} t(I) - \min_{I \in \mathcal{I}} s(I) \leq 4\, n\, \ell(D) \ . \tag{1}$$

Hence we may assume that the endpoints of dotted intervals in $\mathcal{I}$ are in $\{1, \ldots, N\}$, where $N \leq 4n \cdot \ell(D)$. By our assumption that $D = O(1)$ this means that $N = O(n)$. We also note that given a representation $\mathcal{I}$ of a D-DI graph G, a representation of G satisfying (1) can be computed in polynomial time.

3 Maximum Independent Set

In this section we present a dynamic programming algorithm for Maximum Independent Set on D-DI graphs that runs in $O(D\, n^D)$ time, for any D. The algorithm can be thought of as a generalization of the well known algorithm for maximum independent set on interval graphs.

The dynamic programming algorithm for Maximum Independent Set in interval graphs is based on the following property. Given an interval I_i and an independent set $\mathcal{S} \subseteq \mathcal{I}_{<i}$, let I be the interval with the right-most end point in $\mathcal{S}$. If $\mathcal{S}' \subseteq \mathcal{I} \setminus \mathcal{I}_{<i}$ is a maximum weight subset such that $\{I\} \cup \mathcal{S}'$ is independent, then $\mathcal{S}'$ is a maximum weight subset such that $\mathcal{S} \cup \mathcal{S}'$ is independent. Namely, $\mathcal{S}$ can be represented by a single interval for the purpose of finding the best completion of $\mathcal{S}$ from $\mathcal{I} \setminus \mathcal{I}_{<i}$. Furthermore, checking whether $\mathcal{S} \cup \{I_i\}$ is an independent set can done by checking if I_i intersects I. Our algorithm is based on an extension of this property for D-DI graphs.

First, we show that finding a maximum weight completion of $\mathcal{S}$ from $\mathcal{I} \setminus \mathcal{I}_{<i}$ amounts to finding a maximum weight completion of $\mathcal{S}_{s(I_i)}$ from $\mathcal{I} \setminus \mathcal{I}_{<i}$.

Lemma 4. *Let $I_i \in \mathcal{I}$ be a dotted interval, and let $\mathcal{S} \subseteq \mathcal{I}_{<i}$ be an independent set. Also, let $\mathcal{S}' \subseteq \mathcal{I} \setminus \mathcal{I}_{<i}$ be an independent set. If $\mathcal{S}'$ is a maximum weight subset such that $\mathcal{S}_{s(I_i)} \cup \mathcal{S}'$ is independent, then $\mathcal{S}'$ is a maximum weight subset of $\mathcal{I} \setminus \mathcal{I}_{<i}$ such that $\mathcal{S} \cup \mathcal{S}'$ is independent.*

Proof. Consider an interval $J \in \mathcal{I} \setminus \mathcal{I}_{<i}$. Any dotted interval $I \in \mathcal{S}$ intersects J then it must satisfy $s(I_i) \in [s(I), t(I)]$, which means that $I \in \mathcal{I}_{s(I_i)}$. It follows that if $\mathcal{S}_{s(I_i)} \cup \{J\}$ is independent, then $\mathcal{S} \cup \{J\}$ is also independent. □

Suppose we are considering the addition of I_i to an independent set $\mathcal{S} \subseteq \mathcal{I}_{<i}$. Clearly, dotted intervals in $\mathcal{S}$ that terminate before $s(I_i)$ may be ignored. We show that, from the view point of I_i, only up to $d-1$ dotted intervals are needed to represent an independent set $\mathcal{S} \subseteq \mathcal{I}_{<i}$.

Lemma 5. *Let $I_i \in \mathcal{I}$ be a dotted interval, and let $\mathcal{S} \subseteq \mathcal{I}_{<i}$ be an independent set. $\mathcal{S} \cup \{I_i\}$ is independent if and only if (i) $\mathcal{S}_{s(I_i)} \cup \{I_i\}$ is independent, and (ii) $|\mathcal{S}_{s(I_i)}| < D$.*

Proof. Any dotted interval $I \in \mathcal{S}$ intersecting I_i must satisfy $s(I_i) \in [s(I), t(I)]$. In addition, observe that any dotted interval $I \in \mathcal{S}_{s(I_i)}$ must contain at least one point in $[s(I_i) - D + 1, s(I_i))$. Hence, $|\mathcal{S}_{s(I_i)}| < D$. □

Our dynamic programming algorithm is based on Lemmas 4 and 5. The dynamic programming table Π is constructed as follows. A state is a pair of a dotted interval I_i and an independent set $\mathcal{P} \subseteq \mathcal{I}_{s(I_i)}$ of size at most $D-1$. The entry $\Pi(I_i, \mathcal{P})$ stands for the maximum weight of an independent set $\mathcal{S}' \subseteq \mathcal{I} \setminus \mathcal{I}_{<i}$ such that $\mathcal{S}' \cup \mathcal{P}$ is independent. Observe that the optimum is given by $\Pi(I_1, \emptyset)$. The size of the table is $O(n^D)$.

In the base case, we have

$$\Pi(I_n, \mathcal{P}) = \begin{cases} 0 & \mathcal{P} \cup \{I_n\} \text{ is not independent} \\ w(I_n) & \text{otherwise} . \end{cases}$$

For $i < n$, if $\mathcal{P} \cup \{I_i\}$ is not an independent set we have

$$\Pi(I_i, \mathcal{P}) = \Pi(I_{i+1}, \mathcal{P} \cap \mathcal{I}_{s(I_{i+1})}) .$$

On the other hand, if $\mathcal{P} \cup \{I_i\}$ is an independent set, then there are two options. If there exists an index $k > i$ for which the size of $(\mathcal{P} \cup \{I_i\}) \cap \mathcal{I}_{s(I_k)}$ is less than D, then we have

$$\Pi(I_i, \mathcal{P}) = \min \left\{ \ \Pi(I_{i+1}, \mathcal{P} \cap \mathcal{I}_{s(I_{i+1})}) \ , \ w(I_i) + \Pi(I_k, (\mathcal{P} \cup \{I_i\}) \cap \mathcal{I}_{s(I_k)}) \ \right\} \ ,$$

where $k > i$ is the smallest index for which the size of $(\mathcal{P} \cup \{I_i\}) \cap \mathcal{I}_{s(I_k)}$ is less than D. If such an index does not exist, then

$$\Pi(I_i, \mathcal{P}) = \min \left\{ \ \Pi(I_{i+1}, \mathcal{P} \cap \mathcal{I}_{s(I_{i+1})}) \ , \ w(I_i) \ \right\} \ .$$

The correctness of the algorithm is implied by Lemmas 4 and 5. Hence it remains to show that the running time of the algorithm is $O(D\,n^D)$. We do so by proving that the running time of computing an entry of Π is $O(D)$. First, checking whether $\mathcal{P} \cup \{I_i\}$ is an independent set takes $O(D)$ time. Filtering out dotted intervals from $\mathcal{P}$ or $\mathcal{P} \cup \{I_i\}$ that do not belong to $\mathcal{I}_{s(I_{i+1})}$ or to $\mathcal{I}_{s(I_k)}$ also requires $O(D)$ time. Also, finding k, if necessary, can be done in $O(D)$ time.

The computation of $\Pi(I_1, \emptyset)$ can be modified to compute a corresponding independent set using standard techniques. The complement of an independent set is a vertex cover, so the complement of the set returned by our algorithm is minimum weight vertex cover. Hence, we get the following theorem.

Theorem 1. *There is an $O(Dn^D)$-time algorithm for* Maximum Independent Set *and* Minimum Vertex Cover *in D-DI graphs with n vertices.*

Notice that our algorithm runs in $O(n)$ time when $D = 1$, so Theorem 1 can be viewed as a strict generalization of the classical linear time algorithm for Maximum Independent Set in interval graphs.

4 Dominating Set

Using a similar approach to the one used for Maximum Independent Set in D-DI graphs, we can solve the Minimum Dominating Set problem in D-DI graphs in $O(D^2 n^{O(D^2)})$ time, for any D.

Our algorithm for Minimum Dominating Set is based on the following idea. Let $\mathcal{S}$ be a dominating set of $\mathcal{I}$ and consider the set $\mathcal{S}_{<i} = \mathcal{S} \cap \mathcal{I}_{<i}$. Clearly, $\mathcal{S}_{<i}$ covers some dotted intervals from $I_{<i}$, but it may be the case that there are dotted intervals in $\mathcal{I}_{<i}$ that do not intersect $\mathcal{S}_{<i}$. Such dotted intervals must end at or after $s(I_i)$. Furthermore, $\mathcal{S}_{<i}$ may cover dotted intervals in $\mathcal{I} \setminus \mathcal{I}_{<i}$.

Given a dotted interval I_i and a subset $\mathcal{S} \subseteq \mathcal{I}_{<i}$, we say that $\mathcal{S}' \subseteq \mathcal{I} \setminus \mathcal{I}_{<i}$ is a *completion* of $\mathcal{S}$ if $\mathcal{S} \cup \mathcal{S}'$ is a dominating set of $\mathcal{I}$. Notice that it may be the case that such a completion for $\mathcal{S}$ does not exist. Also, given a set $\mathcal{T}$, we say that $I \in \mathcal{T}$ is a *left representative of $\mathcal{T}$ with jump $d(I)$ and offset $o(I)$* if

$$t(I) = \min\{t(I') : I' \in \mathcal{T} \text{ and } d(I') = d(I) \text{ and } j(I') = j(I)\} \ .$$

Similarly, I is a *right representative of $\mathcal{T}$ with jump $d(I)$ and offset $o(I)$* if

$$t(I) = \max\{t(I') : I' \in \mathcal{T} \text{ and } d(I') = d(I) \text{ and } j(I') = j(I)\} \ .$$

The set of left and right representatives of $\mathcal{T}$ is denoted by $\mathcal{T}^L$ and $\mathcal{T}^R$, and contain one representative from each jump-offset pair realized by intervals in $\mathcal{T}$.

Lemma 6. *Let $I_i \in \mathcal{I}$ be a dotted interval, let $\mathcal{S} \subseteq \mathcal{I}_{<i}$, and let $\mathcal{T} \subseteq \mathcal{I}_{<i}$ be the subset of dotted intervals that are not covered by $\mathcal{S}$. If $\mathcal{S}' \subseteq \mathcal{I} \setminus \mathcal{I}_{<i}$ is a minimum weight subset such that $\mathcal{S}^R_{s(I_i)} \cup \mathcal{S}'$ dominates $\mathcal{T}^L \cup (\mathcal{I} \setminus \mathcal{I}_{<i})$, then $\mathcal{S}'$ is a minimum weight completion of $\mathcal{S}$. Furthermore, $|\mathcal{S}^R_{s(I_i)}| + |\mathcal{T}^L| \leq \frac{1}{2}D(D+1)$.*

Proof. First notice that if $\mathcal{S}$ covers a dotted interval $I \in \mathcal{I} \setminus \mathcal{I}_{<i}$, then $\mathcal{S}^R$ must also cover I. Also, if $\mathcal{S}'$ covers $\mathcal{T}^L$, then it must cover $\mathcal{T}$.

Finally, observe that $\mathcal{T} \subseteq \mathcal{I}_{s(I_i)}$ since otherwise $\mathcal{S}$ cannot be completed. It follows that it cannot be that $I \in \mathcal{S}^R_{s(I_i)}$ and $J \in \mathcal{T}^L$ represent the same jump and offset, since in this case I covers J. Hence, $\mathcal{S}^R_{s(I_i)} \cup \mathcal{T}^L$ contain at most one representative for each pair of jump and offset, and there are $\frac{1}{2}D(D+1)$ such pairs. □

The dynamic programming table Π is constructed as follows. A state is a triple of a dotted interval I_i, and sets $\mathcal{P}$ and $\mathcal{Q}$ such that

- $\mathcal{P}, \mathcal{Q} \subseteq \mathcal{I}_{s(I_i)}$.
- $\mathcal{P} \cap \mathcal{Q} = \emptyset$.
- $\mathcal{P} \cup \mathcal{Q}$ contain at most one dotted interval for every pair of jump and offset.

The entry $\Pi(I_i, \mathcal{P}, \mathcal{Q})$ stands for the minimum weight subset S' such that $\mathcal{P} \cup \mathcal{S}'$ dominates $(\mathcal{I} \setminus \mathcal{I}_{<i}) \cup \mathcal{Q}$. Observe that the optimum is given by $\Pi(I_1, \emptyset, \emptyset)$. The size of the table is $n^{O(D^2)}$.

In the base case, we have

$$\Pi(I_n, \mathcal{P}, \mathcal{Q}) = \begin{cases} 0 & \mathcal{Q} = \emptyset \text{ and } \mathcal{P} \text{ covers } I_n, \\ w(I_n) & \mathcal{Q} \neq \emptyset \text{ and } I_n \text{ covers } \mathcal{Q}, \\ \infty & \text{otherwise.} \end{cases}$$

For $i < n$, if $\mathcal{Q} \nsubseteq \mathcal{I}_{s(I_i)}$, then

$$\Pi(I_i, \mathcal{P}, \mathcal{Q}) = \infty \ .$$

otherwise,

$$\Pi(I_i, \mathcal{P}, \mathcal{Q}) = \min \left\{ \ \Pi(I_{i+1}, \mathcal{P}_{s(I_{i+1})}, \mathcal{Q}) \ , \ \Pi(I_{i+1}, (\mathcal{P} \cup \{I_i\})_{s(I_{i+1})}, \mathcal{Q}') \ \right\} \ ,$$

where $\mathcal{Q}' \subseteq \mathcal{Q}$ is the subset of dotted intervals that are not covered by I_i.

The correctness of our algorithm is implied by Lemma 6. Computing the value $\Pi(I_i, \mathcal{P}, \mathcal{Q})$ can be done in $O(D^2)$. Hence, the running time of the algorithm is $O(D^2 n^{O(D^2)})$. The computation of $\Pi(I_1, \emptyset)$ can be modified to compute a corresponding independent set using standard techniques.

Theorem 2. *There is an $O(D^2 n^{O(D^2)})$-time algorithm for* Minimum Dominating Set *on D-DI graphs with n vertices.*

5 Deletion Problems

This section presents an EPTAS for a wide class of deletion problems in D-DI graphs. Three classical examples of such problems are Minimum Vertex Cover, Minimum Feedback Vertex Set, and Minimum Planar Deletion. For ease of presentation, we first describe our algorithm for Minimum Vertex Cover, and then later explain how it generalizes to other deletion problems. We begin by recalling the definition of a path decomposition [18]:

Definition 1. *A* path decomposition *of a given graph G is a path $\mathcal{P}$ whose vertices $V(\mathcal{P}) \subseteq 2^{V(G)}$ are subsets of vertices in G, called* bags*, satisfying the following two properties:*

- $\bigcup_{P \in V(\mathcal{P})} G[P] = G$*, and*
- *for every $v \in V$, the set of bags $\{P \in V(\mathcal{P}) : v \in P\}$ induces a subpath in $\mathcal{P}$.*

The width *of the path decomposition $\mathcal{P}$ is $\max_{P \in \mathcal{P}} |P| - 1$. The* pathwidth *of G is the minimum width of any path decomposition of G.*

It is well known that an interval graph with maximum clique size k has pathwidth $k-1$. The next lemma shows that this result generalizes quite nicely to D-DI graphs.

Definition 2. *A clique K in a D-DI graph with dotted interval representation $\mathcal{I}$ is a* point clique *if there exists a point $p \in \mathbb{N}$ which is included in every $I_v \in \mathcal{I}$ with $v \in K$.*

Lemma 7. *A D-DI graph with maximum point clique size k has pathwidth at most $Dk - 1$.*

Proof. Let G be a D-DI graph, and let $\mathcal{I}$ denote a set of dotted intervals corresponding to G. Let K_i denote the set of all vertices whose corresponding dotted interval include the integer $i \in \mathbb{N}$. Define a path decomposition $\mathcal{P} := P_1, \ldots, P_N$ for G by $P_i := \bigcup_{j=i}^{i+D-1} K_j$ for all $i \in \{1, \ldots, N\}$, where N is the maximum integer included in any dotted interval of $\mathcal{I}$. Since G has no clique of size $k+1$, we have $|K_i| \leq k$ for all $i \in \mathbb{N}$. Thus, $|P_i| \leq Dk$ for all $i \in \{1, \ldots, N\}$. We finish the proof by showing that $\mathcal{P}$ is indeed a path decomposition of G.

First observe that any vertex of G is included in some $K_i \subseteq P_i$. Second, since for any edge $\{u, v\} \in E(G)$ we have $i \in I_u \cap I_v$ for some $i \in \{1, \ldots, N\}$, every edge is also completely contained in some K_i, which in turn is contained in P_i; thus, $\bigcup_i G[P_i] = G$. Finally, observe that for any vertex v, if $v \in P_i \cap P_{i+2}$ for any $i \in \{1, \ldots, N-2\}$, then it must be the case that $v \in P_{i+1}$; otherwise, the jump of I_v must be at least $D+1$. Thus, the second condition in Definition 1 is also satisfied, and $\mathcal{P}$ is a path decomposition of width at most $Dk - 1$. □

Aumann *et al.* [2] show that for any $D \in \mathbb{N}$ there exists a finite bipartite graph G which is not a D-DI graph. An interesting corollary of Lemma 7 is that such a statement is true even for trees, a much more restricted class of bipartite graphs.

Corollary 1. *For any $D \in \mathbb{N}$ there is a finite tree T which is not a D-DI graph.*

Proof. Let D be given. Robertson and Seymour [18] argued that for any integer $w \in \mathbb{N}$ there is a finite tree with pathwidth greater than w. By Lemma 7, choosing such a tree for $w := 2D$ gives a tree T which is not a D-DI graph, since the maximum clique size of T is 2. □

Another interesting corollary of Lemma 7 more relevant to our purposes is that Minimum Vertex Cover can be solved optimally in D-DI graphs of maximum (point) clique size k in time $2^{O(Dk)} \cdot n$. This follows from the well known $2^{O(w)} \cdot n$ algorithm for Minimum Vertex Cover in graph of pathwidth at most w (see *e.g.* [6]). Recall that by Observation 3, we can assume that $N = O(n)$ so the path decomposition obtain in the proof of Lemma 7 can be computed in linear time.

Corollary 2. *There is a linear-time algorithm for solving* Minimum Vertex Cover *restricted to D-DI graphs given with representations that have maximum point clique size k.*

Theorem 3. *For any fixed $d \in \mathbb{N}$ and $\varepsilon > 0$, there exists a linear time $(1+\varepsilon)$-approximation algorithm for unweighted* Minimum Vertex Cover *in D-DI graphs.*

Proof. Let G be a given D-DI graph with representation $\mathcal{I}$, and let $k := 1/\varepsilon$. We first greedily compute a maximal set $\mathcal{K} := \{K_1, \ldots, K_t\}$ of pairwise disjoint point cliques of size $k+1$ in G. (Note that there can be several point cliques of size $k+1$ at the same point.) Such a set can be computed in linear time. Let $S_1 := \bigcup \mathcal{K}$, and let $G_1 := G[S_1]$ and $G_2 := G[V \setminus S_1]$. Then G_2 has maximum point clique size k, and so by Corollary 2 we can compute an optimal vertex cover S_2 for G in linear time. Our algorithm outputs the set of vertices $S := S_1 \cup S_2$. Clearly, S is a vertex cover of G. We next argue that S is has size at most $(1+\varepsilon)|\text{OPT}|$, where OPT is a minimum vertex cover of G.

Let OPT_1 and OPT_2 respectively denote minimum vertex covers of the graphs G_1 and G_2. Then $|\text{OPT}_2| = |S_2|$ and $|\text{OPT}_1| + |\text{OPT}_2| \leq |\text{OPT}|$. Observe that for any clique $K \in \mathcal{K}$, we must have $|\text{OPT}_1 \cap K| \geq k$, otherwise OPT_1 would not be a vertex cover of G_1. Since each such K has size $k+1$, we have

$$|S_1| \leq (1 + 1/k)|\text{OPT}_1| = (1+\varepsilon)|\text{OPT}_1|.$$

Thus,

$$|S| = |S_1| + |S_2| \leq (1+\varepsilon)|\text{OPT}_1| + |\text{OPT}_2| \leq (1+\varepsilon)|\text{OPT}|.$$

□

We next consider other deletion problems. For a graph class (property) $\mathcal{G}$, the Minimum $\mathcal{G}$-Deletion problem takes as input a graph G, and the goal is to compute a minimum size subset of vertices S in G such that $G - S \in \mathcal{G}$. We will be interested in this problem for graph classes $\mathcal{G}$ that have finite forbidden subgraph, topological minor, or minor characterizations. We call such a graph class *finitely defined*. For example, if $\mathcal{G}$ is the class of forests (and Minimum $\mathcal{G}$-Deletion is Minimum Feedback Vertex Set) then $\mathcal{G}$ has a finite forbidden minor characterization which consists of the single graph K_3; when $\mathcal{G}$ is the set of all planar graphs then it has forbidden minor characterization consisting of $K_{3,3}$ and K_5.

Let $\mathcal{G}$ be a finitely defined graph class. First, notice that for any positive integer w, the Minimum $\mathcal{G}$-Deletion problem can be solved in linear time when restricted to graphs of treewidth w; this is due to an extension of Courcelle's Theorem [10] due to Borie *et al.* [7]. Second, notice that the clique-deletion technique that is applied in the proof of Theorem 3 can be extended to Minimum $\mathcal{G}$-Deletion. Specifically, this is done by setting $k := (h-1)/\varepsilon$, where h is the minimum number of vertices in any graph of the forbidden characterization of $\mathcal{G}$. Clearly any solution S for Minimum $\mathcal{G}$-Deletion must include at least $k-h+1$ vertices of any clique of size k in the input graph G, since otherwise $G-S$ will contain a graph from the forbidden characterization of $\mathcal{G}$. Using this observation, the argument used in Theorem 3 follows exactly as is.

Theorem 4. *For any fixed $d \in \mathbb{N}$ and $\varepsilon > 0$, and any finitely defined graph class $\mathcal{G}$, there exists a linear time $(1+\varepsilon)$-approximation algorithm for unweighted* Minimum $\mathcal{G}$-Deletion *in D-DI graphs.*

6 Concluding Remarks

This paper presents algorithms for a number of classical optimization problems in D-DI graphs. We show an $O(Dn^D)$-time algorithm for Maximum Independent Set and Minimum Vertex Cover in D-DI graphs, and give an $O(D^2 n^{O(D^2)})$-time algorithm for Minimum Dominating Set. We also present a linear-time $(1+\varepsilon)$-approximation algorithm for unweighted Minimum Vertex Cover in D-DI graphs, that generalizes to a wide range of deletion problems. We note that for Minimum Vertex Cover and many other problems for this class, our algorithm also works for the general weighted case using the local ratio technique [3] for the clique-deletion process in the proof of Theorem 3. However since the Borie *et al.* [7] extension of Courcelle's Theorem does not work for weighted graphs, Theorem 4 in its generality only applies to uniform weights.

Two main open problems stem from our work. The first is to settle the fix parameter tractability of these problems of the problems considered in this paper, when parameterized by D. In particular, is Minimum Vertex Cover parameterized by D in FPT, or is it W[1]-hard? The second question arises from the fact that our algorithms crucially exploit the D-DI representation of the input graph. Thus, the natural question to ask is whether one can in polynomial-time compute a D-DI representation for a given graph G and a fixed D, or to determine that none exists. This can be done efficiently when $D = 1$ since it reduces to finding an interval representation of a given interval graph. We conjecture that finding a D-DI representation is NP-hard for $D \geq 2$.

References

1. P. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11, 1998.

2. Y. Aumann, M. Lewenstein, O. Melamud, R. Y. Pinter, and Z. Yakhini. Dotted interval graphs and high throughput genotyping. In *16th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 339–348, 2005.
3. R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–46, 1985.
4. R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM J. Comput.*, 36(1):1–15, 2006.
5. R. Bar-Yehuda, D. Hermelin, and D. Rawitz. Minimum vertex cover in rectangle graphs. *Comput. Geom.*, 44(6–7):356–364, 2011.
6. H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybern*, 11(1-2):1–22, 1993.
7. R. Borie, R. Parker, and C. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7:555–581, 1992.
8. A. Butman, D. Hermelin, M. Lewenstein, and D. Rawitz. Optimization problems in multiple-interval graphs. *ACM Transactions on Algorithms*, 6(2), 2010.
9. P. Chalermsook and J. Chuzhoy. Maximum independent set of rectangles. In *20th annual ACM-SIAM SODA*, pages 892–901, 2009.
10. B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
11. T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34, 2005.
12. M. R. Garey, D. S. Johnson, G. L. Miller, and C. H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic and Discrete Methods*, 1(2):216–227, 1980.
13. M. C. Golumbic and P. L. Hammer. Stability in circular arc graphs. *Journal of Algorithms*, 9(3):314–320, 1988.
14. D. Hermelin and D. Rawitz. Optimization problems in multiple subtree graphs. *Discrete Applied Mathematics*, 159(7):588–594, 2011.
15. W.-L. Hsu and K.-H. Tsai. Linear time algorithms on circular-arc graphs. *Inf. Process. Lett.*, 40(3):123–129, 1991.
16. H. H. III, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms*, 26, 1998.
17. M. Jiang. Approximating minimum coloring and maximum independent set in dotted interval graphs. *Inf. Process. Lett.*, 98(1):29–33, 2006.
18. N. Robertson and P. Seymour. Graph minors. I. Excluding a forest. *J. Comb. Theory, Ser. B*, 35(1):39–61, 1983.
19. E. J. van Leeuwen. Approximation algorithms for unit disk graphs. In *31st Workshop on Graph-theoretic concepts in computer science*, pages 351–361, 2005.
20. E. J. van Leeuwen. Better approximation schemes for disk graphs. In *10th Scandinavian Workshop on Algorithm Theory*, pages 316–327, 2006.
21. V. Yanovski. Approximation algorithm for coloring of dotted interval graphs. *Inf. Process. Lett.*, 108(1):41–44, 2008.