On the Parameterized Complexity of Multiple-Interval Graph Problems

Michael R. Fellows*[1], Danny Hermelin**[2], Frances Rosamond*[1], and Stéphane Vialette[3]

[1] The University of Newcastle, Callaghan NSW 2308 - Australia
{mike.fellows,frances.rosamond}@cs.newcastle.edu.au
[2] Department of Computer Science, University of Haifa, Mount Carmel, Haifa 31905 - Israel
danny@cri.haifa.ac.il
[3] Laboratoire de Recherche en Informatique (LRI), UMR CNRS 8623
Faculté des Sciences d'Orsay - Université Paris-Sud, 91405 Orsay - France
vialette@lri.fr

Abstract. Multiple-interval graphs are a natural generalization of interval graphs where each vertex may have more than one interval associated with it. Many applications of interval graphs also generalize to multiple-interval graphs, often allowing for more robustness in the modeling of the specific application. With this motivation in mind, a recent systematic study of optimization problems in multiple-interval graphs was initiated. In this sequel, we study multiple-interval graph problems from the perspective of parameterized complexity. The problems under consideration are k-Independent Set, k-Dominating Set, and k-Clique, which are all known to be W[1]-hard for general graphs, and NP-complete for multiple-interval graphs. We prove that k-Clique is in FPT, while k-Independent Set and k-Dominating Set are both W[1]-hard. We also prove that k-Independent Dominating Set, a hybrid of the two above problems, is also W[1]-hard. Our hardness results hold even when each vertex is associated with at most two intervals, and all intervals have unit length. Furthermore, as an interesting byproduct of our hardness results, we develop a useful technique for showing W[1]-hardness via a reduction from the k-Multicolored Clique problem, a variant of k-Clique. We believe this technique has interest in its own right, as it should help in simplifying W[1]-hardness results which are notoriously hard to construct and technically tedious.

* This research has been supported by the Australian Research Council through the Australian Center for Bioinformatics, by the University of Newcastle Parameterized Complexity Research Unit under the auspices of the Deputy Vice-Chancellor for Research, and by a Fellowship to the Durham University Institute for Advanced Studies. The authors also gratefully acknowledge the support and kind hospitality provided by a William Best Fellowship at Grey College while the paper was in preparation.
** Partially supported by the Caesarea Rothschild Institute.

1 Introduction

Interval graphs – the intersection graphs of interval families – is one of the most popular and well-understood graph class in algorithmic graph theory. This is mainely due to the following two reasons:

1. They have numerous applications in various areas, most of which can be modeled by classical graph-theoretic problems. As an example, basic scheduling and storage problems translate to finding minimum colorings and clique covers in appropriate interval graphs [20].
2. Many classical NP-hard problems become polynomial-time solvable when restricted to interval graphs. For example, Independent Set, Dominating Set, and Clique are all polynomial-time solvable when restricted to interval graphs [16, 19].

A natural generalization of interval graphs are *multiple-interval* graphs. These are intersection graphs of families of *multiple intervals*, where a multiple-interval is the union of a finite number of disjoint intervals over the real line. Indeed, many applications that translate to interval graph problems extend naturally to multiple-interval graph problems, often allowing for more robustness in the modeling of the specific application. Scheduled tasks become multi-tasks, storage items require non-linear storage space, and so forth. However, in contrast to interval graphs, many natural NP-hard problems still remain hard when restricted to multiple-interval graphs.

For this reason, there has recently been an effort for the systematic study of multiple-interval problems. For example, Bar-Yehuda *et al.* [4] studied the Maximum Independent Set problem in t-interval graphs (multiple-interval graphs with t intervals associated with each vertex), and gave a $2t$-approximation algorithm for this problem. In [7], Butman *et al.* considered Minimum Vertex Cover, Minimum Dominating Set, and Maximum Clique, and presented algorithms with approximation factors of $2 - 1/t$, t^2, and $(t^2 - t + 1)/2$ for these problems respectively. We mention also [5], where a general variant of the Maximum Independent Set problem in multiple-interval graphs was discussed. Up to date, the above line of research has focused solely on approximation algorithms.

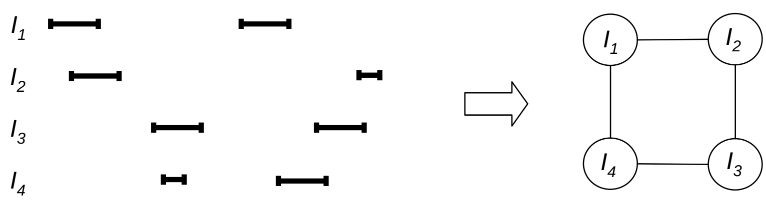

Fig. 1. A 2-interval graph of four 2-intervals. Notice that this simple example already shows that 2-interval graphs are different from interval graphs, as an interval graph cannot have a chordless cycle of length four as an induced subgraph.

In this paper, we study multiple-interval graph problems from the perspective of parameterized complexity. Hence, we are interested in algorithms that compute exact optimal solutions, rather than good approximations, while attempting to confine the inevitable exponential-running time of such algorithms to an input-length independent parameter (in our case the size of solution). The problems we consider are k-Independent Set, k-Dominating Set, and k-Clique, which are all known to be W[1]-hard for general graphs [12, 14], and NP-complete in multiple-interval graphs [4, 7]. In [4, 7], applications that range from scheduling, video on demand, and employee monitoring are discussed for these three problems.

Our results can be summarized as follows. In the main part of the paper we prove that k-Independent Set and k-Dominating Set are both W[1]-hard for 2-interval graphs. Along with the recent work of Marx [28, 29], these are the first known examples of natural W[1]-hard problems that admit constant factor approximations (as Maximum Independent Set and Minimum Dominating Set can both be approximated within a factor of 4 in 2-interval graphs by the results of [4, 7] mentioned above). It is worth mentioning that these two results also hold when all intervals of the given multiple-interval family are required to be of equal length, thus forming a *unitary multiple-interval* family. Also, as an interesting corollary, we get that k-Independent Dominating Set is also W[1]-hard in 2-interval graphs. In the last part of the paper, we prove that unlike the two previous problems, k-Clique in multiple-interval graphs is in FPT, even when t is taken also as a parameter (*i.e.* t, k-Clique).

As a byproduct of our W[1]-hardness proofs, we develop a general technique for providing such results, which we call the "k-Multicolored Clique *reduction technique*". In this technique, one reduces from the k-Multicolored Clique problem, a variant of k-Clique where the graph is colored with k colors and the solution sought is required to be composed of different colored vertices. In a nutshell, the vertex-coloring in k-Multicolored Clique allows for an almost systematic gadget-construction and helps in eliminating several technical details from the overall proof. As an example, our original W[1]-hardness proof for k-Independent Set was nine pages long before applying our technique. We believe this technique has interest in its own right since it should help in simplifying other W[1]-hardness results in different settings. We therefore devote a large portion of this paper to explaining this technique.

The rest of the paper is organized as follows. In the remainder of this section, we briefly discuss basic concepts from parameterized complexity, quickly define notation and terminology used for working with multiple-intervals, and then swiftly go through some relevant work which is related to ours. In Section 2, we outline the k-Multicolored Clique reduction technique. Then, in Sections 3 and 4 respectively, we use this technique to show that k-Independent Set and k-Dominating Set are W[1]-hard already for 2-interval graphs. In Section 5, we discuss the k-Clique problem, and present our fixed-parameter algorithm for it. Finally, we discuss possible directions for future work in Section 6.

1.1 Parameterized complexity

Parameterized complexity is a refinement to classical complexity theory in which one takes into account not only the total input length n, but also other aspects of the problem encoded in a parameter k. In doing so, one attempts to confine the super-polynomial running time needed for solving many natural problems strictly to the parameter. Our terminology follows the general reference text of Downey and Fellows [11].

A *parameterized problem* (or *parameterized language*) is a subset $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed alphabet, Σ^* is the set of all finite length strings over Σ, and $\mathbb{N}$ is the set of natural numbers. In this way, an input (x, k) to a parameterized language consists of two parts, where the second part k is the *parameter*. A parameterized problem L is *fixed-parameter tractable* if there exists an algorithm which on a given input $(x, k) \in \Sigma^* \times \mathbb{N}$, decides whether $(x, k) \in L$ in $f(k) \cdot poly(n)$ time, where f is an arbitrary computable function solely in k, and *poly* is a polynomial in the total input length $n = |x| + k$. Such an algorithm is said to run in FPT-*time*, and FPT is the class of all parameterized problems that can be solved by an FPT-time algorithm (*i.e.* all problems which are fixed-parameter tractable).

A formal framework for proving *fixed-parameter intractability* was developed over the years, using the notion of *parameterized reductions* [12, 13]. A (many-to-one) parameterized reduction from a parameterized problem L to another parameterized problem L' is an FPT-time computable mapping that maps an instance $(x, k) \in \Sigma^* \times \mathbb{N}$ to an instance $(x', k') \in \Sigma^* \times \mathbb{N}$ with k' bounded by some function solely in k and with $(x, k) \in L \iff (x', k') \in L'$. The W-hierarchy of parameterized-intractable problem classes is defined via the k-WEIGHTED SATISFACTION problem for bounded depth circuits: Given an integer $t \in \mathbb{N}^+$, the class W[t] is the class of all problems parameterized-reducible to the k-WEIGHTED SATISFACTION problem for circuits of weft t, where the *weft* of a circuit is the maximum number of gates having in-degree greater than 2 in any path from an input to the output. For our purposes, we focus only on W[1]. It is known that if FPT = W[1], then n variable 3-SAT can be solved in time $2^{o(n)}$ [9].

1.2 Multiple-interval notation and terminology

A *t-interval* I is the union of t disjoint intervals of the real line, and we write $I = (i_1, i_2, \ldots, i_t)$ with $i_1, i_2, \ldots, i_t$ disjoint intervals, and $I = \bigcup_{p=1}^{t} i_p$. Given a pair of t-intervals $I = (i_1, i_2, \ldots, i_t)$ and $J = (j_1, j_2, \ldots, j_t)$, these two t-intervals *intersect* if they share a common point, *i.e.* if $(\bigcup_{p=1}^{t} i_p) \cap (\bigcup_{q=1}^{t} j_q) \neq \emptyset$. Two non-intersecting t-intervals are said to be *disjoint*.

Let $\mathcal{F} = \{I_1, \ldots, I_n\}$ be a family of t-intervals. The *underlying family of intervals* of $\mathcal{F}$, denoted $\mathcal{I}(\mathcal{F})$, is the family of all intervals belonging to t-intervals in $\mathcal{F}$. That is, $\mathcal{I}(\mathcal{F}) = \{i \in \{i_1, i_2, \ldots, i_t\} \mid I = (i_1, i_2, \ldots, i_t) \in \mathcal{F}\}$. The *intersection graph* $\Omega_{\mathcal{F}}$ of $\mathcal{F}$, is a graph with a one-to-one correspondence between its vertices and $\mathcal{F}$ such that two vertices are connected in $\Omega_{\mathcal{F}}$ if their corresponding t-intervals in $\mathcal{F}$ intersect. For a graph $G = \Omega_{\mathcal{F}}$, we say that G is a *t-interval graph* to emphasize the fact that it is an intersection graph of family of t-intervals for some given $t \in \mathbb{N}^+$. The family $\mathcal{F}$ is called a t-interval *representation* of G.

We are concerned with independent sets, dominating sets, and cliques in t-interval graphs. In terms of t-interval families, a subset $\mathcal{S}$ of a t-interval family $\mathcal{F}$ is an *independent set* of $\mathcal{F}$, if it is pairwise disjoint. A subset $\mathcal{D} \subseteq \mathcal{F}$ is a *dominating set*, if for any t-interval in $\mathcal{F} \setminus \mathcal{D}$ there is a t-interval in $\mathcal{D}$ which intersects (dominates) it. A subset $\mathcal{K} \subseteq \mathcal{F}$ is called a *clique*, if it is pairwise intersecting. Given a parameter k, the k-INDEPENDENT SET, k-DOMINATING SET, and k-CLIQUE problems ask to decide whether $\mathcal{F}$ has a independent set, dominating set, and clique of size k respectively.

1.3 Related work

Multiple-interval graphs have been studied extensively from the graph-theoretic aspect. We briefly list some of the main results. The class of graphs with maximum degree Δ are a subclass of $\lceil(\Delta + 1)/2\rceil$-interval graphs [22]. From this it follows that INDEPENDENT SET and DOMINATING SET are NP-complete already for 2-interval graphs, since these include graphs of maximum degree 3 [17]. Every graph with n vertices is a $\lceil(n+1)/4\rceil$-interval graph, and the complete bipartite graph $K_{\lfloor n/2\rfloor,\lceil n/2\rceil}$ is an extremal example of this [21]. The class of planar graphs is a subclass of 3-interval graphs [30]. Finally, the problem of determining whether a given graph is t-interval is NP-complete for $t \geq 2$ [32]. It is therefore usually necessary to assume that a t-interval graph is given along with its t-interval representation.

The systematic study of multiple-interval graph problems was initiated by [4, 7]. However, many variants of multiple-interval graph problems have been studied previous (or in parallel) to these two papers, and in particular by the computational biology community. For instance, Bafna *et al.* [3]

studied the problem of finding the maximum weight subset of non-overlapping local alignments between two genomic sequences. This problem translates to finding a maximum weight independent set in a restricted subclass of 2-interval graphs. In [2], Aumann *et al.* studied another restricted subclass of t-interval graphs in the context of high throughput genotyping. In [6, 10, 31], 2-intervals were used to model secondary structure of RNA sequences, and secondary structure prediction scenarios were modeled by the INDEPENDENT SET problem in 2-interval graphs and related subclasses. Zhao *et al.* [33] studied tree decompositions of 2-interval graphs in the same context. We mention also variants of multiple-interval covering problems that were studied by the combinatorial optimization and discrete geometry communities [18, 24–27].

Finally, we cite the closely related work of Marx [28, 29] who studied the parameterized complexity status of k-INDEPENDENT SET and k-DOMINATING SET in various geometric intersection graphs. In particular, Marx showed that k-INDEPENDENT SET and k-DOMINATING SET are both W[1]-hard in intersection graphs of unit squares, discs, and line segments [28, 29].

2 The k-Multicolored Clique Reduction Technique

Both our W[1]-hardness results for k-INDEPENDENT SET and k-DOMINATING SET are obtained by what we call the "k-MULTICOLORED CLIQUE *reduction technique*". In this technique, one shows W[1]-hardness by a reduction from the k-MULTICOLORED CLIQUE problem, a multicolored variant of k-CLIQUE. In the following we elaborate on this technique, giving details on the general gadgetry and *modus operandi* used in it. We begin with a formal definition of our core problem:

k-MULTICOLORED CLIQUE

Instance: A graph G and a vertex-coloring $c : V(G) \to \{1, 2, \ldots, k\}$ for G.

Question: Does G have a clique including vertices of all k colors ? That is, are there $v_1, \ldots, v_k \in V(G)$ such that for all $1 \leq i < j \leq k$: $\{v_i, v_j\} \in E(G)$ and $c(v_i) \neq c(v_j)$?

Parameter: k.

The vertex-coloring in k-MULTICOLORED CLIQUE is what helps in simplifying details when using it in reductions. Indeed, multicolored versions exists for almost all natural subset problems, and using the powerful *color-coding technique* of Alon, Yuster, and Zwick [1], one can show that they are (under Turing reductions, not many-to-one) as hard as their uncolored counterparts. However, due to the key-role that k-CLIQUE plays in almost all fundamental W[1]-hardness results of parameterized complexity (see *e.g.* [8, 15]), it is not surprising that we chose to use it in our technique. Furthermore, we can easily prove that it is W[1]-complete under many-to-one reductions.

Lemma 1. *k-MULTICOLORED CLIQUE is W[1]-complete.*

Proof. The proof is via a reduction from k-CLIQUE. Given an instance (G, k) for k-CLIQUE, we construct a graph G' by taking k copies $v_1, \ldots, v_k$ of each vertex v of G, and then coloring each vertex v_i with color $i \in \{1, 2, \ldots, k\}$. We then add an edge in G' between two vertices u_i and v_j, $i \neq j$, iff u and v are connected in G. It is straightforward to verify that G has a k-clique iff G' has a k-multicolored clique. ◻

The main idea in the k-MULTICOLORED CLIQUE reduction technique is to exploit the additional structure given by the vertex-coloring when constructing the reduction. This can usually be done by using two types of gadgets – selection and validation. Before giving details, let us first introduce some notation that will be used for working with vertex-colored graphs: Let G be a graph giving along with a vertex-coloring $c : V(G) \to \{1, 2, \dots, k\}$. For a given color $c \in \{1, 2, \dots, k\}$, we let V_c denote vertices in G colored c, *i.e* $V_c = \{v \in V(G) \mid c(v) = c\}$. Similarly, for a given pair of distinct colors, we let $E_{\{c_1,c_2\}}$ denote the subset of edges $\{u, v\} \in E(G)$ with $\{c(u), c(v)\} = \{c_1, c_2\}$. It is important to to observe that we can assume w.l.o.g. that c is a proper coloring, *i.e.* there are no edges $\{u, v\} \in E(G)$ with $c(u) = c(v)$. Any such edge can safely be removed from G.

- *Selection:* The role of the selection gadgets is to codify the selection of k vertices of distinct colors, and/or the selection of $\binom{k}{2}$ edges of distinct pairs of colors, that form a k-multicolored clique. The gadgets will usually be grouped by colors, and pairs of colors in case of edge-selection. In this way, the c vertex-selection gadget, $c \in \{1, 2, \dots, k\}$, will be constructed in such a way that the intersection of any valid solution with the elements of the gadget will correspond to some vertex $v \in V_c(G)$. Similarly, the intersection of any valid solution with the $\{c_1, c_2\}$ edge-selection gadget, $\{c_1, c_2\} \subseteq \{1, 2, \dots, k\}$, will correspond to some edge $e \in E_{\{c_1,c_2\}}(G)$. Here is one place where the vertex-coloring comes in handy, since when constructing the gadgets, one does not have to worry about details such as identical vertex selection in different selection gadgets, and so forth. This leaves the role of the validation gadgets simple and clean.

- *Validation:* The role of the validation gadgets is to ensure that the selection codified by the selection gadgets is compatible. Thus, validation gadgets might be used to ensure that the k selected vertices are in fact adjacent, or that the $\binom{k}{2}$ selected edges are in fact over the same set of k vertices. Usually, validation gadgets will somehow "overlap" with the selection gadgets, in such a way so that they can be used to ensure the compatibility of selection. One trick that seems to come in handy is to view the edges of the graph also as directed edges, and to use gadgets representing the directed edges (u, v) and (v, u) to validate the selection of the vertices colored $c(u)$ and $c(v)$, and/or the edge colored $\{c(u), c(v)\}$.

This may seem rather vague, as well as it should be, as reductions are rather ubiquitous creatures. Nevertheless, we hope our two examples in the following sections will help clarify matters.

3 *k*-Independent Set

We begin our discussion on parameterized problems for multiple-interval graphs with the k-Independent Set problem. Recall that given a multiple-interval graph G along with its t-interval representation $\mathcal{F}$, and a parameter $k \in \mathbb{N}^+$, the k-Independent Set problem asks whether $\mathcal{F}$ has a subset of k pairwise disjoint multiple-intervals. Also recall that general k-Independent Set (*i.e.* k-Independent Set for arbitrary graphs) is W[1]-complete [14]. We show that k-Independent Set is W[1]-complete already for 2-interval graphs, using the k-Multicolored Clique reduction technique outlined above. Before giving details on the appropriate gadgets, we give a short sketch of our construction.

Let (G, c, k) be an instance of k-Multicolored Clique. Without loss of generality, we can assume that there are no edges between vertices of the same color (*i.e.* c is a proper coloring), and that any clique in G is in fact a multicolored clique. We denote by $(G', \mathcal{F}, k')$, with $\mathcal{F}$ a family of 2-intervals and $G' = \Omega_{\mathcal{F}}$, the instance of k-Independent Set for 2-interval graphs we construct from (G, c, k). The main idea of our construction is to represent each edge $\{u, v\} \in E(G)$ by three 2-intervals: two for each "direction", $I_{(u,v)}, I_{(v,u)} \in \mathcal{F}$, and one undirected, $I_{\{u,v\}} \in \mathcal{F}$. In addition, we represent each vertex $v \in V(G)$ by a 2-interval $I_v \in \mathcal{F}$. In this way, the number of 2-intervals in $\mathcal{F}$ will equal exactly $|V(G)| + 3|E(G)|$. Furthermore, the 2-intervals in $\mathcal{F}$ will be constructed in such a way that any k-clique in G will correspond to $k + 3\binom{k}{2}$ pairwise disjoint 2-intervals in $\mathcal{F}$, corresponding to all edges (directed and undirected) and all vertices in the clique. Thus, we set $k' = k+3\binom{k}{2}$. To guarantee that k-cliques and only k-cliques in G correspond to k'-independent sets in $\mathcal{F}$, we organize the 2-intervals into appropriate selection and validation gadgets, as explained in further detail below.

3.1 Gadgetry

Vertex-selection: Let $c \in \{1, \ldots, k\}$ be some color. We view the c vertex-selection gadget as a "table" of intervals with $|V_c(G)|$ rows and $k+1$ columns. The rows of the table are associated with vertices in $V_c(G)$, and the columns, apart from the first and last, correspond to colors in $\{1, \ldots, k\} \setminus \{c\}$. The first and last intervals of each row form the vertex 2-interval $I_v \in \mathcal{F}$, where v is the vertex in $V_c(G)$ associated with that row. The other intervals in the row are each associated with a distinct color in $\{1, \ldots, k\} \setminus \{c\}$, and are used for validation. For a given $c' \in \{1, \ldots, k\} \setminus \{c\}$, we will refer to the interval associated with color c' in the row associated with $v \in V_c(G)$, as the *color c' interval of v*. The intervals in the table are constructed in such a way so that intervals in the same column are pairwise intersecting Furthermore, an offset is created so that intervals also intersect intervals which are above them (*i.e.* lower row-index) in the next column. (see Fig. 2).

Fig. 2. A schematic example of the selection gadgets used in our construction for k-Independent Set. On the left is an example of a vertex-selection gadget, and on the right is an edge-selection gadget.

Edge-selection: For a given pair of distinct colors $\{c_1, c_2\} \subseteq \{1, \ldots, k\}$, we construct the $\{c_1, c_2\}$ edge-selection gadget in similar fashion. We again view the gadget as a table of intervals, this time having $|E_{\{c_1,c_2\}}|$ rows and 4 columns, with each row associated with a unique edge $\{u, v\} \in E_{\{c_1,c_2\}}$. The first and last interval of each row form the edge 2-interval $I_{\{u,v\}} \in \mathcal{F}$, where $\{u, v\}$ is the edge associated with this row. The second and third intervals are again used for validation, one belonging to the directed edge 2-interval $I_{(u,v)}$ and the other belonging to $I_{(v,u)}$. The intervals in the table are again organized in such a way so that each column is pairwise intersecting, along with an offset as done in the vertex-selection gadgets.

Validation: Validating the selection of k vertices and $\binom{k}{2}$ edges is done using directed edge 2-intervals. These have intervals both in vertex-selection and edge-selection gadgets. The 2-interval $I_{(u,v)} \in \mathcal{F}$, for an ordered pair of adjacent vertices $u, v \in V(G)$, is composed of one interval from the row associated with $\{u, v\}$ in the $\{c(u), c(v)\}$ edge-selection gadget, along with the color $c(v)$ interval of u in the $c(u)$ vertex-selection gadget (see Fig. 3).

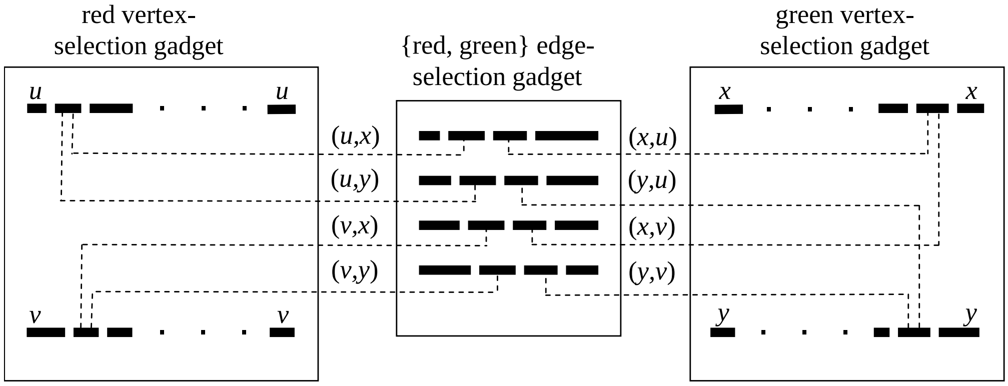

Fig. 3. A schematic example of how the directed edge 2-intervals are constructed. The example depicts the construction of a cycle (u, x, v, y) with alternating red and green colors.

3.2 The construction

The entire construction of $\mathcal{F}$ consists of constructing all gadgets in disjoint regions of the real line so that no two intervals in different gadgets intersect. This gives us the following 2-interval family:

$$\mathcal{F} = \Big\{ I_v \,\Big|\, v \in V(G) \Big\} \cup \Big\{ I_{\{u,v\}}, I_{(u,v)}, I_{(v,u)} \,\Big|\, \{u, v\} \in E(G) \Big\}.$$

As mentioned previously, we set $k' = k + 3\binom{k}{2}$. It is is clear that our construction can be carried out in FPT-time.

We next argue that G has k-multicolored clique if and only if $\mathcal{F}$ has a k'-independent set. For the first direction, suppose $K \subseteq V(G)$ is a k-multicolored clique in G. The reader is invited to check that the following subset of 2-intervals

$$\mathcal{S} = \Big\{ I_v \in \mathcal{F} \,\Big|\, v \in K \Big\} \cup \Big\{ I_{\{u,v\}}, I_{(u,v)}, I_{(v,u)} \in \mathcal{F} \,\Big|\, u, v \in K, u \neq v \Big\}$$

is a k'-independent set in $\mathcal{F}$.

Conversely, suppose $\mathcal{S}$ is a k'-independent set in $\mathcal{F}$. Observe first that by our construction, $\mathcal{S}$ can include at most one vertex 2-interval I_v with $c(v) = c$, for any color class c, and at most one undirected and two directed edge 2-intervals for any pair of distinct colors. Since $\mathcal{S}$ is of size k', it has to include exactly one 2-interval of each of the above types. It follows that for every color $c \in \{1, \dots, k\}$, there are $k + 1$ 2-intervals in $\mathcal{S}$ with intervals in the c vertex-selection gadget, and for every pair of distinct colors $\{c_1, c_2\} \subseteq \{1, \dots, k\}$, there are three 2-intervals in $\mathcal{S}$ with intervals in the $\{c_1, c_2\}$ edge-selection gadget.

Now, by construction of the c vertex-selection gadget, any $k + 1$ pairwise disjoint 2-intervals that have intervals in this gadget, must all have intervals from the same row. Since all intervals from a given row are associated with the same vertex $u \in V_c(G)$, we have for any $I_u, I_{(x,y)} \in \mathcal{S}$: $c(u) = c(x) \Rightarrow u = x$. Furthermore, by construction of the $\{c_1, c_2\}$ edge-selection gadget, any three pairwise disjoint 2-intervals that have intervals in this gadget, must all have intervals from the same row. Since all intervals in a given row are associated with the same edge (ordered and unordered), we have for any $I_{\{u,v\}}, I_{(x,y)} \in \mathcal{S} : \{c(u), c(v)\} = \{c(x), c(y)\} \Rightarrow \{u, v\} = \{x, y\}$. Combining these two facts together, we get $I_u, I_v \in \mathcal{S} \Rightarrow I_{\{u,v\}} \in \mathcal{S}$.

It is now not difficult to see that the subset of vertices $K = \{v \in V(G) \,|\, I_v \in \mathcal{S}\}$ induces a k-multicolored clique in G. First, since $\mathcal{S}$ includes a vertex 2-interval I_v with $c(v) = c$ for all

$c \in \{1, \ldots, k\}$, K indeed includes k vertices of distinct colors. Second, for any pair of vertices $u, v \in K$, we have

$$u, v \in K \Rightarrow I_u, I_v \in \mathcal{S} \Rightarrow I_{\{u,v\}} \in \mathcal{S} \Rightarrow \{u, v\} \in E(G),$$

and so the theorem below follows.

Theorem 1. k-Independent Set *in 2-interval graphs is* W[1]*-complete.*

4 k-Dominating Set

We next turn to consider the k-Dominating Set problem in multiple-interval graphs. Recall that given a t-interval graph G along with its t-interval representation $\mathcal{F}$, k-Dominating Set asks to decide whether there is a subset of k t-intervals $\mathcal{D} \subseteq \mathcal{F}$ where for each t-interval $I \in \mathcal{F} \setminus \mathcal{D}$ there is a t-interval $I' \in \mathcal{D}$ with $I \cap I' \neq \emptyset$ (I' dominates I). For general graphs, this problem is known to be W[2]-complete [13]. We show that it is W[1]-hard already for 2-interval graphs via the k-Multicolored Clique reduction technique.

Let (G, c, k) be an instance of k-Multicolored Clique where w.l.o.g. the are no edges between two vertices of the same color. We use $(G', \mathcal{F}, k')$ to denote the instance we construct for k-Dominating Set from (G, c, k), where $\mathcal{F}$ is a 2-interval family and $G' = \Omega_{\mathcal{F}}$. The construction we use is very similar to the one used in Section 3. We again represent each vertex $v \in V(G)$ by a 2-interval $I_v \in \mathcal{F}$, and each edge $\{u, v\} \in E(G)$ by two directed edge 2-intervals $I_{(u,v)}, I_{(v,u)} \in \mathcal{F}$ and one undirected edge 2-interval $I_{\{u,v\}} \in \mathcal{F}$. The 2-intervals are constructed in such a way so that a k'-dominating set in $\mathcal{F}$ will be forced to include exactly k vertex 2-intervals and $\binom{k}{2}$ undirected edge 2-intervals. These will correspond to k vertices and $\binom{k}{2}$ edges which form a k-multicolored clique in G. Thus, we set $k' = k + \binom{k}{2}$. In the spirit of the k-Multicolored Clique reduction technique, the selection of vertices and edges is codified in appropriate selection gadgets, and the validation of compatibility of this selection is done via validation gadgets.

4.1 Gadgetry

Vertex-selection: For each color $c \in \{1, \ldots, k\}$, we construct a c vertex-selection gadget which will codify the selection of the c colored vertex in a multicolored clique of G. The gadget is quite simple, and consists of $1 + |V_c(G)|$ disjoint intervals – one interval associated with the color c, and one interval associated with each vertex $v \in V_c(G)$. The vertex 2-interval I_v then consists of the interval associated with v and the interval associated with c. In this way, all vertex 2-intervals corresponding to vertices of a given color are pairwise intersecting. In addition, we add a pair of disjoint "dummy" 2-intervals that intersect the interval associated with the color c and nothing else (see Fig. 4). The role of these dummy 2-interval is to assure that at least one 2-interval with an interval in the c vertex selection gadget will be selected in any k'-dominating set of $\mathcal{F}$.

Edge-selection: For each pair of distinct colors $\{c_1, c_2\} \subseteq \{1, \ldots, k\}$, we construct a $\{c_1, c_2\}$ edge-selection gadget which is very similar to the $\{c_1, c_2\}$ edge-selection gadget used for k-Independent Set (see Section 3.1). Again, the gadget consists of $|E_{\{c_1,c_2\}}|$ rows, this time with only 3 columns, where each row is associated with a unique edge $\{u, v\} \in E_{\{c_1,c_2\}}$ – the first and last interval belonging to $I_{\{u,v\}}$, and the second to both $I_{(u,v)}$ and $I_{(v,u)}$. The gadget also consists of a pair of disjoint "dummy" 2-intervals, one intersecting all intervals in the first column of the gadget, and the other intersecting all intervals in the last column.

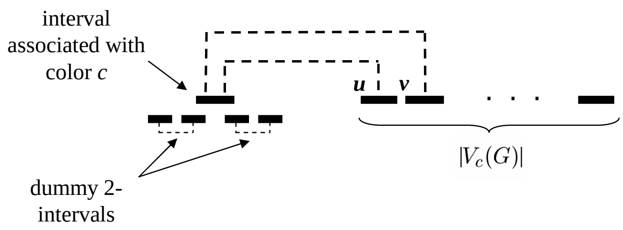

Fig. 4. An example of the vertex-selection gadget used in k-Dominating Set.

Validation: Validating the compatibility of the selection of k vertices and $\binom{k}{2}$ edges is done using directed edge 2-intervals which have intervals both in vertex-selection and edge-selection gadgets. The 2-interval $I_{(u,v)} \in \mathcal{F}$, for an ordered pair of adjacent vertices $u, v \in V(G)$, is composed of the second interval from the row associated with $\{u, v\}$ in the $\{c(u), c(v)\}$ edge-selection gadget, along with the interval associated with u in the $c(u)$ vertex-selection gadget (see Fig. 5).

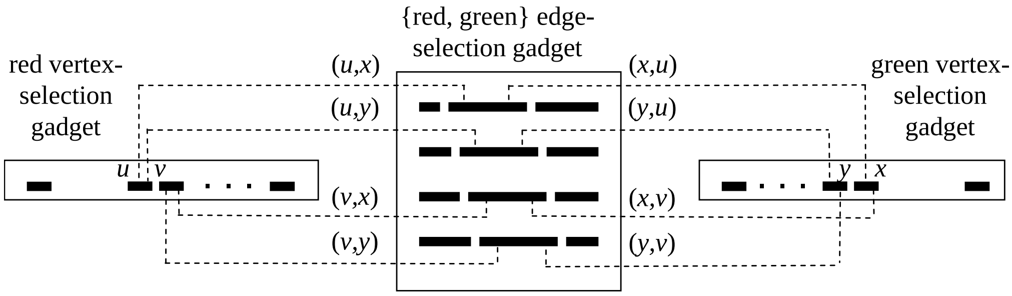

Fig. 5. A schematic example of how the directed edge 2-intervals are used for validation. The example depicts the construction of a cycle (u, x, v, y) with alternating red and green colors. Dummy 2-intervals are omitted from the figure.

4.2 The construction

The entire construction of $\mathcal{F}$ consists of constructing all selection and validation gadgets in disjoint regions of the real line so that no two intervals in different gadgets intersect. This gives us the following 2-interval family:

$$\mathcal{F} = \left\{ I_v \;\middle|\; v \in V(G) \right\} \cup \left\{ I_{\{u,v\}}, I_{(u,v)}, I_{(v,u)} \;\middle|\; \{u, v\} \in E(G) \right\} \cup DUMMIES$$

where $DUMMIES$ is the set of $2k+2\binom{k}{2}$ dummy 2-intervals that are placed in each vertex-selection and edge-selection gadget. To complete the construction, we set $k' = k + \binom{k}{2}$.

Now suppose G has a k-multicolored clique $K \subseteq V(G)$. Then it is easy to see that the following subset of 2-intervals

$$\mathcal{D} = \left\{ I_v \;\middle|\; v \in K \right\} \cup \left\{ I_{\{u,v\}} \;\middle|\; u, v \in K, u \neq v \right\}$$

is a k'-dominating set of $\mathcal{F}$.

For the converse direction, suppose $\mathcal{D}$ is a k'-dominating set of $\mathcal{F}$. To dominate all dummy 2-intervals in $\mathcal{F}$, $\mathcal{D}$ must include at least one 2-interval with an interval in the c vertex-selection gadget, for each color $c \in \{1, \dots, k\}$, and at least one undirected edge 2-interval with intervals in the $\{c_1, c_2\}$ edge-selection gadget, for each pair of distinct colors $c_1, c_2 \in \{1, \dots, k\}$. Since $k' = k + \binom{k}{2}$, it follows that exactly k vertex 2-intervals (representing vertices of different colors) and $\binom{k}{2}$ undirected edge 2-intervals (representing edges of different pairs of colors) must be included in $\mathcal{D}$.

Now, an undirected edge 2-interval $I_{\{u,v\}} \in \mathcal{D}$ will dominate all 2-intervals with intervals in the $\{c_1, c_2\}$ edge-selection gadget, except $I_{(u,v)}$ and $I_{(v,u)}$. Hence, these must be dominated in the validation gadgets. But this is possible only if both I_u and I_v are in $\mathcal{D}$. It follows that the subset of vertices $K \subseteq V(G)$ defined by $K = \{v \in V(G) \mid I_v \in \mathcal{D}\}$ forms a k-multicolored clique in G.

Theorem 2. *k-Dominating Set for 2-interval graphs is* W[1]*-hard.*

Notice that any k'-dominating set of $\mathcal{F}$ in our construction will be pairwise disjoint. This means that our construction works also for an important variant of k-Dominating Set, the k-Independent Dominating Set problem (also known as the k-Minimal Maximal Independent Set problem).

Theorem 3. *k-Independent Dominating Set for 2-interval graphs is* W[1]*-hard.*

5 k-Clique

We now consider the k-Clique problem in multiple-interval graphs. Recall that given a t-interval graph $G = \Omega_{\mathcal{F}}$ of a family of t-intervals $\mathcal{F}$, and a parameter k, the k-Clique problem for t-interval graphs asks whether there is a subset $\mathcal{K} \subseteq \mathcal{F}$ of size k which is pairwise intersecting. The problem is NP-complete for $t = 3$ [7]. We show that not only is k-Clique in FPT for constant values of t, but also that k, t-Clique is in FPT (*i.e.* when the problem is parameterized by both k and t). Recall that when t is unbounded, k-Clique for t-interval graphs reduces to general k-Clique, as all graphs of n vertices are t-interval graphs for $t \geq n/4$ [21].

Our algorithm for k-Clique is quite simple, but it exemplifies an important strategy in designing fixed-parameter algorithms: Either determine whether a given instance is a "yes" instance in FPT-time, or reduce the instance to a smaller instance by removing at least one of the elements in the instance. The hard part is usually in finding what properties of the given problem allow this strategy. The following lemma, which appeared in a similar form in [4], gives us this property for k-Clique in t-interval graphs.

Lemma 2. *If G is a t-interval graph with no k-clique then G has a vertex of degree less than $2tk$.*

Proof. Let $\mathcal{F}$ be the t-interval family representation of G, and let $G^* = \Omega_{\mathcal{I}(\mathcal{F})}$ be the (1-)interval graph of the underlying interval family $\mathcal{I}(\mathcal{F})$ of $\mathcal{F}$. Also, for any interval $i \in \mathcal{I}(\mathcal{F})$, let $R(i)$ denote all intervals $j \in \mathcal{I}(\mathcal{F})$, $j \neq i$, which include the right endpoint of i. Then $|E(G^*)| = \sum_{i \in \mathcal{I}(\mathcal{F})} |R(i)|$. Furthermore, $|R(i)| < k$, for any $i \in \mathcal{I}(\mathcal{F})$, since otherwise G has a clique of size k. Noting also that $|E(G)| \leq |E(G^*)|$ and that $|V(G^*)| \leq t|V(G)|$, we get

$$|E(G)| \leq |E(G^*)| = \sum_{i \in \mathcal{I}(\mathcal{F})} |R(i)| < k|V(G^*)| \leq tk|V(G)|.$$

It follows that G has average degree less than $2tk$, and the lemma holds. ☐

Let $deg(v)$ denote the number of vertices adjacent to v in G, and let $G - v$ denote the graph obtained by removing v along with all edges incident to v from G. Fig. 6 gives a simple algorithm for deciding whether a t-interval graph G has a k-clique. Correctness of this algorithm is immediate due to Lemma 2. As for its time complexity, notice that line 4 of this algorithm can be implemented in $\mathcal{O}(k^2 \cdot \binom{2tk}{k})$ time. Hence, since all other operations require polynomial time, each recursive call can be performed in FPT-time with respect to both k and t, and as the algorithm has $\mathcal{O}(n)$ recursive calls, it follows that the entire algorithm can be implemented to run in FPT-time. It is worth noticing that our algorithm does not require a t-interval representation of G for its computation, only the guarantee that G is in fact t-interval.

Algorithm Clique(G, k)

Data : A t-interval graph G and an integer k.
Result : YES iff G has a k-clique.
begin
  1. if $|V(G)| < k$ **then return** NO.
  2. Let v be a vertex of minimum degree in G.
  3. If $deg(v) \geq 2tk$ **then return** YES.
  4. if v is in a k-clique of G **then return** YES.
  5. return Clique($G - v, k$).
end

Fig. 6. An algorithm for k-Clique in t-interval graphs.

Theorem 4. *k, t-Clique is in* FPT.

6 Discussion

In this paper, we initiated the study of multiple-interval graph problems under the parameterized complexity framework. We considered three classical problems – k-Independent Set, k-Dominating Set, and k-Clique. We proved that k-Clique is in FPT, while k-Independent Set and k-Dominating Set are both W[1]-hard. Our constructions also implied that the two above problem are W[1]-hard in unitary 2-interval graphs, and also that their hybrid problem, k-Independent Dominating Set, is W[1]-hard for this graph class.

We believe that the importance of our results is twofold. First, we determined the parameterized complexity status of three elegantly defined combinatorial problems that have real life applications. Along with the recent work of Marx [28, 29], our hardness results give the first known examples of natural W[1]-hard problems that admit constant factor approximations. Second, for proving our hardness results, we developed a useful technique for showing W[1]-hardness using a reduction from the k-Multicolored Clique problem, a variant of k-Clique. This technique seems to have interest in its own right, as it should help in simplifying other W[1]-hardness results which tend to be technically tedious and difficult to construct. Below we list a few questions remained unanswered in our work:

– An important subclass of multiple-interval graphs is the class of multitrack-interval graphs. A *t-multitrack-interval* graph is an intersection graph of a multitrack-interval family, where a t-multitrack-interval is the union of t intervals over t different lines [23], and a t-multitrack-interval family is a family of t-multitrack-intervals defined over the same t lines. Both our constructions for k-Dominating Set and k-Independent Set do not hold for 2-multitrack interval graphs. Are k-Dominating Set and k-Independent Set also W[1]-hard for 2-multitrack interval graphs ?

– k-Dominating Set in general graphs is W[2]-complete. Is k-Dominating Set also W[2]-complete for 2-interval graphs ? How about t-interval graphs for any $t \in \mathbb{N}$?

– The parametric time-bound of the algorithm presented in Section 5 is asymptotical to $t^{\mathcal{O}(k \lg k)}$. Can this be improved ? A possible good place to start is to consider the problem for constant values of t, and to attempt to obtain a parametric time-bound of $2^{\mathcal{O}(k)}$.

Another possibly prosperous direction for extending our work would be to study other W[1]-hard graph problems, seeing whether they become fixed-parameter tractable in multiple-interval graphs. Possible candidates could be k-Irredundant Set, k-Perfect Code, k-Vertex Clique Cover, and k-Separating Vertices.

Acknowledgements

The authors would like to thank an anonymous referee who helped improving the hardness result in Section 3.

References

1. N. Alon, R. Yuster, and U. Zwick. Color coding. *Journal of the ACM*, 42(4):844–856, 1995.
2. Y. Aumann, M. Lewenstein, O. Melamud, R.Y. Pinter, and Z. Yakhini. Dotted interval graphs and high throughput genotyping. In *Proceedings of the 16th annual Symposium on Discrete Algorithms (SODA)*, pages 339–348, 2005.
3. V. Bafna, B.O. Narayanan, and R. Ravi. Nonoverlapping local alignments (weighted independent sets of axis parallel rectangles). *Discrete Applied Mathematics*, 71:41–53, 1996.
4. R. Bar-Yehuda, M.M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. In *Proceedings of the 13th annual ACM/SIAM Symposium On Discrete Algorithms (SODA)*, pages 732–741, 2002.
5. R. Bar-Yehuda and D. Rawitz. Using fractional primal-dual to schedule split intervals with demands. In *13th annual European Symposium on Algorithms*, pages 714–725, 2005.
6. G. Blin, G. Fertin, and S. Vialette. Extracting constrained 2-interval subsets in 2-interval sets. *Theoretical Computer Science*, 385:241–263, 2007.
7. A. Butman, D. Hermelin, M. Lewenstein, and D Rawitz. Optimization problems in multiple-interval graphs. In *Proceedings of the 18th annual ACM/SIAM Symposium On Discrete Algorithms (SODA)*, pages 268–278, 2007.
8. L. Cai, J. Chen, R.G. Downey, and M.R. Fellows. On the parameterized complexity of short computation and factorization. *Archive for Mathematical Logic*, 36(4–5):321–337, 1997.
9. L. Cai and D.W. Juedes. Subexponential parameterized algorithms collapse the W-hierarchy. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 273–284, 2001.
10. M. Crochemore, D. Hermelin, G.M. Landau, and S. Vialette. Approximating the 2-interval pattern problem. In *Proceedings of the 13th annual European Symposium on Algorithms (ESA)*, pages 426–437, 2005.
11. R. Downey and M. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
12. R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness. *Congressus Numerantium*, 87:161–187, 1992.
13. R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24:873–921, 1995.
14. R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science*, 141(1–2):109–131, 1995.
15. R.G. Downey and M.R. Fellows. Parameterized computational feasibility. In *Proceedings of Feasible Mathematics II*, pages 273–284, 1995.
16. M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics*, 7:115–130, 1984.
17. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.
18. D. R. Gaur, T. Ibaraki, and R. Krishnamurti. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. *Journal of Algorithms*, 43:138–152, 2002.

19. F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1:180–187, 1972.
20. M.C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980.
21. J.R. Griggs. Extremal values of the interval number of a graph, II. *Discrete Mathematics*, 28:37–47, 1979.
22. J.R. Griggs and D.B. West. Extremal values of the interval number of a graph. *SIAM Journal on Algebraic and Discrete Methods*, 1(1):1–14, 1980.
23. A. Gyráfás and D. West. Multitrack interval graphs. *Congr. Numer.*, 109:109–116, 1995.
24. R. Hassin and N. Megiddo. Approximation algorithms for hitting objects with straight lines. *Discrete Applied Mathematics*, 30(1):29–42, 1991.
25. R. Hassin and D. Segev. Rounding to an integral program. In *4th International Workshop on Efficient and Experimental Algorithms*, pages 44–54, 2005.
26. R. Hassin and A. Tamir. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 10(7):395–402, 1991.
27. D.S. Hochbaum and A. Levin. Cyclical scheduling and multi-shift scheduling: complexity and approximation algorithms. *Discrete Optimization*, 2006. To appear.
28. D. Marx. Efficient approximation schemes for geometric problems? In *Proceedings of the 13th annual European Symposium on Algorithms (ESA)*, pages 448–459, 2005.
29. D. Marx. Parameterized complexity of independence and domination on geometric graphs. In *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC)*, pages 154–165, 2006.
30. E.R. Scheinerman and D.B. West. The interval number of a planar graph: three intervals suffice. *Journal of Combinatorial Theory B*, 35:224–239, 1983.
31. S. Vialette. On the computational complexity of 2-interval pattern matching problems. *Theoretical Computer Science*, 312:335–379, 2004.
32. D.B. West and D.B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discrete Applied Mathematics*, 8:295–305, 1984.
33. J. Zhao, R.L. Malmberg, and L. Cai. Rapid *ab initio* rna folding including pseudoknots via graph tree decomposition. In *Proceedings of the 6th Workshop on Algorithms in BioInformatics (WABI)*, pages 262–273, 2006.