Balanced Independent and Dominating Sets on Colored Interval Graphs

Sujoy Bhore
Indian Institute of Science Education and Research, Bhopal, India

Jan-Henrik Haunert
Geoinformation Group, University of Bonn, Bonn, Germany

Fabian Klute
Algorithms and Complexity Group, TU Wien, Vienna, Austria

Guangping Li
Algorithms and Complexity Group, TU Wien, Vienna, Austria

Martin Nöllenburg
Algorithms and Complexity Group, TU Wien, Vienna, Austria

Abstract

We study two new versions of independent and dominating set problems on vertex-colored interval graphs, namely \( f \)-Balanced Independent Set (\( f \)-BIS) and \( f \)-Balanced Dominating Set (\( f \)-BDS). Let \( G = (V, E) \) be a vertex-colored interval graph with a \( k \)-coloring \( \gamma : V \to \{1, \ldots, k\} \) for some \( k \in \mathbb{N} \). A subset of vertices \( S \subseteq V \) is called \( f \)-balanced if \( S \) contains \( f \) vertices from each color class. In the \( f \)-BIS and \( f \)-BDS problems, the objective is to compute an independent set or a dominating set that is \( f \)-balanced. We show that both problems are \( \text{NP} \)-complete even on proper interval graphs. For the \( f \)-BIS problem we design two FPT algorithms, one parameterized by \((f, k)\) for interval graphs and the other parameterized by the vertex cover number for general graphs. Moreover, for the optimization variant of \( 1 \)-BIS on interval graphs, we show that a simple greedy approach achieves approximation ratio 2.

2012 ACM Subject Classification
Theory of computation → Computational geometry; Theory of computation → Fixed parameter tractability; Theory of computation → Problems, reductions and completeness; Theory of computation → Approximation algorithms analysis

Keywords and phrases
Interval graphs, Independent set, \( \text{NP} \)-completeness

Funding
This work was supported by the Austrian Science Fund (FWF) under grant P31119.

1 Introduction

A graph \( G \) is an interval graph if it has an intersection model consisting of intervals on the real line. Formally, \( G = (V, E) \) is an interval graph if there is an assignment of an interval \( I_v \subseteq \mathbb{R} \) for each \( v \in V \) such that \( I_u \cap I_v \) is nonempty if and only if \( \{u, v\} \in E \). A proper interval graph is an interval graph that has an intersection model in which no interval properly contains another \([12]\). Consider an interval graph \( G = (V, E) \) and additionally assume that the vertices of \( G \) are \( k \)-colored by a color assignment \( \gamma : V \to \{1, \ldots, k\} \). We define and study color-balanced versions of two classical graph problems: maximum independent set and minimum dominating set on vertex-colored (proper) interval graphs. In what follows, we define the problems formally and discuss their underlying motivation.

\footnote{We use the term color assignment instead of vertex coloring to avoid any confusion with the general notion of vertex coloring; in particular, a color assignment \( \gamma \) can map adjacent vertices to the same color.}
**f-Balanced Independent Set (f-BIS):** Let \( G = (V, E) \) be an interval graph with a color assignment of the vertices \( \gamma : V \rightarrow \{1, \ldots, k\} \). Find an \( f \)-balanced independent set of \( G \), i.e., an independent set \( L \subseteq V \) that contains exactly \( f \) elements from each color class.

The classic maximum independent set problem serves as a natural model for many real-life optimization problems and finds applications across fields, e.g., computer vision [2], information retrieval [20], and scheduling [24]. Specifically, it has been used widely in map-labeling problems [1, 26, 25, 4], where an independent set of a given set of label candidates corresponds to a conflict-free and hence legible set of labels. To display as much relevant information as possible, one usually aims at maximizing the cardinality or, in the case of weighted label candidates, the total weight of the independent set. This approach may be appropriate if all labels represent objects of the same category. In the case of multiple categories, however, maximizing the cardinality or total weight of the labeling does not reflect the aim of selecting a good mixture of different object types. For example, if the aim was to inform a map user about different possible activities in the user’s vicinity, labeling one cinema, one theater, and one museum may be better than labeling four cinemas. In such a setting, the \( f \)-BIS problem asks for an independent set that contains \( f \) vertices from each object type.

We initiate this study for interval graphs, which is a primary step to understand the behavior of this problem on general intersection graphs. Moreover, solving the problem for interval graphs gives rise to optimal solutions for certain labeling models, e.g., if every label candidate is a rectangle that is placed at a fixed position on the boundary of the map [14].

We also introduce an optimization variant of the 1-BIS problem, which asks for a maximally colorful independent set.

**1-Max-Colored Independent Set (1-MCIS):** Let \( G = (V, E) \) be an interval graph with a color assignment of the vertices \( \gamma : V \rightarrow \{1, \ldots, k\} \). The objective is to find a 1-max-colored independent set of \( G \), i.e., an independent set \( L \subseteq V \), whose vertices contain a maximum number of colors and \( L \) contains at most one element from each color class.

The second problem we discuss is defined as follows.

**f-Balanced Dominating Set (f-BDS):** Let \( G = (V, E) \) be an interval graphs with a color assignment of the vertices \( \gamma : V \rightarrow \{1, \ldots, k\} \). Find an \( f \)-balanced dominating set, i.e., a subset \( D \subseteq V \) such that every vertex in \( V \setminus D \) is adjacent to at least one vertex in \( D \), and \( D \) contains exactly \( f \) elements from each color class.

The dominating set problem is another fundamental problem in theoretical computer science, which also finds applications in various fields of science and engineering [8, 15]. Several variants of the dominating set problem have been considered over the years: \( k \)-tuple dominating set [10], Liar’s dominating set [9], independent dominating set [10], and more. The colored variant of the dominating set problem has been considered in parameterized complexity, namely, red-blue dominating set, where the objective is to choose a dominating set from one color class that dominates the other color class [11]. Instead, our \( f \)-BDS problem asks for a dominating set of a vertex-colored graph that contains \( f \) vertices of each color class. Similar to the independent set problem, we primarily study this problem on vertex-colored interval graphs, which can be of independent interest.

**Related Work** In the field of interval scheduling [7], the colored variant of the maximum independent set problem on interval graphs has been well studied. This colored interval
scheduling problem is known as the Job Interval Scheduling Problem (JISP), introduced by Nakajima and Hakimi [18]. Let \( T = \{ t_1, t_2, \ldots, t_n \} \) be a set of \( n \) independent tasks. Each task \( t_i \) has a task duration \( d_i \) and a set of \( k_i \) possible starting times \( \{ s_{i1}, s_{i2}, \ldots, s_{ik_i} \} \). Note that all duration and starting times of all tasks are assumed to be integers. The problem JISP is to find a starting time for each task so that each task can be executed by a single processor. If there is some fixed integer \( k \) such that \( k_i \leq k \) for \( i \in \{ 1, 2, \ldots, n \} \), the problem is referred to as JISP\(_k\) for short. Constructing a colored interval graph, where each possible execution of a task \( t_i \) starting at \( s_{ij} \) corresponds to a half-open interval \( [s_{ij}, s_{ij} + d_i] \) of color \( i \), the problem JISP is equivalent to finding a 1-BIS in the constructed interval graph. Spieksma and Crama [22] showed that JISP3 is NP-complete even if each duration \( d_i \) equals 2. Spieksma [21] proved that the optimization variant of JISP, namely finding an optimal schedule that finishes a maximum number of jobs, is APX-hard, and provided a 2-approximation algorithm.

### Paper Structure

While there exist polynomial-time algorithms for the maximum independent set problem and the minimum dominating set problem on interval graphs [13, 9], it turns out their colored variants \( f\)-BIS and \( f\)-BDS are much more resilient and NP-complete even for proper interval graphs and \( f = 1 \) (Section 2). Then, we restrict our attention to the \( f\)-BIS problem. In Section 3, we complement the complexity result of the \( f\)-BIS problem with two FPT algorithms, one for interval graphs and parameterized by \((f,k)\) and the other parameterized by the graph’s vertex cover number and for general graphs. Section 4 introduces an \( O(n \log n)\)-time 2-approximation algorithm for the optimization problem 1-MCIS on interval graphs.

### 2 Complexity Results

In this section we show that \( f\)-BIS and \( f\)-BDS are NP-complete even if the given graph \( G \) is a proper interval graph and \( f = 1 \). It is readily seen that \( f\)-BIS and \( f\)-BDS are in the class \( \text{NP} \), since for a given potential solution \( V' \subseteq V \) it can be checked in polynomial time whether \( V' \) is an independent (dominating) set and contains \( f \) vertices of each color. To show \( \text{NP}\)-hardness of these two problems, we give reductions from two restricted, but still \( \text{NP}\)-complete versions of 3SAT, respectively. In the following, for two problems \( A \) and \( B \), we write \( A \leq_p B \) if there is a polynomial-time reduction from \( A \) to \( B \).

#### 2.1 Complexity of \( f\)-Balanced Independent Set

In a 3-bounded 3SAT formula, each variable is allowed to appear in at most three clauses and clauses have two or three literals. The 3-bounded 3SAT problem is NP-complete [23]. This section is devoted to showing NP-completeness of 1-BIS by giving a reduction from 3-bounded 3SAT.

▶ **Lemma 2.1.** 3-bounded 3SAT \( \leq_p \) 1-BIS.

**Proof.** Let \( \phi \) be a 3-bounded 3SAT formula with variables \( x_1, \ldots, x_n \) and clause set \( \mathcal{C} = \{ C_1, \ldots, C_m \} \). From \( \phi \) we now construct an equivalent instance of 1-BIS consisting of a proper interval graph \( G = (V, E) \) and a color assignment \( \gamma \) of \( V \). We choose the set of colors to contain exactly \( m \) colors, one for each clause in \( \mathcal{C} \) and we number these colors from 1 to \( m \). We add a vertex \( u_{i,j} \in V \) for each occurrence of a variable \( x_i \) in a clause \( C_j \) in \( \phi \). Furthermore, we insert an edge \( \{ u_{i,j}, u_{i,j'} \} \in E \) whenever \( u_{i,j} \) was inserted because
of a positive occurrence of $x_i$ and $u_{i,j'}$ was inserted because of a negative occurrence of $x_i$. Finally, we color each vertex $u_{i,j} \in V$ with color $j$. See Figure 1 for an illustration. The graph $G$ created from $\phi$ is a proper interval graph as it consists only of disjoint paths of length at most three. It can clearly be constructed in polynomial time and space.

\[
\begin{array}{cccc}
\text{Positive} & u_{1,1} & u_{1,4} & u_{2,1} & u_{2,4} & u_{3,2} & u_{3,4} & u_{4,1} & u_{4,3} \\
\text{Negative} & x_1 & x_2 & x_3 & x_4 & \\
\end{array}
\]

\[
(x_1 \lor x_2 \lor x_4) \land (x_1 \lor x_3 \lor x_4) \land (x_3 \lor x_4) \land (x_1 \lor x_2 \lor x_3)
\]

**Figure 1** The constructed 1-BIS instance depicted as interval representation with the vertex colors being the colors of the intervals.

It remains to prove that $\phi$ is satisfiable if and only if $G = (V, E)$ has a 1-BIS with respect to the color assignment $\gamma$.

Suppose $V' \subseteq V$ is a 1-balanced independent set of $G$. We construct a variable assignment for $x_1, \ldots, x_n$ as follows. By definition we find for each color $j$ precisely one vertex $u_{i,j} \in V'$. If $u_{i,j}$ was inserted for a positive occurrence of $x_i$, then we set $x_i$ to true and otherwise $x_i$ to false. Moreover, all variables $x_i$ with $i \in \{1, \ldots, n\}$ for which we do not find a corresponding interval in $V'$ are also set to false. Since $V'$ is an independent set in $G$ this assignment is well defined. Now assume it was not satisfying, then there exists a clause $C_j$ for which none of its literals evaluates to true. Hence, none of the at most three vertices corresponding to the literals in $C_j$ is in $V'$. Recall that there is a one-to-one correspondence between clauses and colors in the instance of 1-balanced independent set we created. Yet, $V'$ does not contain a vertex of that color, a contradiction.

For the opposite direction assume we are given a satisfying assignment of the 3 bounded 3SAT formula $\phi(x_1, \ldots, x_n)$ with clauses $C = \{C_1, \ldots, C_m\}$. Furthermore let $G = (V, E)$ be the graph with a color assignment of the vertices $\gamma$ constructed from $\phi$ as described above. We find a 1-balanced independent set of $G$ from the given assignment as follows. For each clause $C_j \in C$ we choose one of its literals that evaluates to true and add the corresponding vertex $v_j \in V$ to the set of vertices $V'$. Since there is a one-to-one correspondence between the colors and the clauses of the assignment is satisfying, $V'$ clearly contains one vertex per color.

It remains to show that $V'$ is an independent set of vertices in $G$. Assume for contradiction that there are two vertices $v_i, v_i' \in V'$ and $\{v_{i,j}, v_{i,j'}\} \in E$. Then, by construction of $G$, we know that $i = i'$ and further that $v_{i,j}, v_{i,j'}$ correspond to one positive and one negative occurrence of $x_i$ in $\phi$. By the construction of $V'$ this implies a contradiction to the assignment being consistent.

Evidently 1-BIS is in the class $\text{NP}$. Together with Lemma 2.1 this implies the following theorem.

**Theorem 2.2.** The $f$-balanced independent set problem on a graph $G = (V, E)$ with a color assignment of the vertices $\gamma: V \rightarrow \{1, \ldots, k\}$ is $\text{NP}$-complete, even if $G$ is a proper interval graph and $f = 1$.

**Remark 2.3.** Since the constructed graph $G$ consists of a set of disjoint paths of length at most three, it is easy to construct a set $I$ of right-open intervals of length 2 such that $G$ represents the intersections of $I$. Moreover, each color class consists of at most three intervals. Therefore, we independently prove the $\text{NP}$-completeness of the Job Interval Scheduling
Positive
Negative
Positive
Negative
Positive
Negative
t1
t2
f1
f2
hf
ht
c1
f
c2
t
c2
f

2.2 Complexity of \( f \)-Balanced Dominating Set

In a 2P2N-3SAT formula, each variable appears exactly twice positively and twice negatively. The 2P2N-3SAT problem is \( \text{NP}\)-complete \[27\]. In this section, we show \( \text{NP}\)-hardness of 1-BDS by a reduction from 2P2N-3SAT.

**Construction** Let \( \phi \) be a 2P2N-3SAT formula with variables \( x_1, \ldots, x_n \) and clause set \( \mathcal{C} = \{C_1, \ldots, C_m\} \). For variable \( x_i \) in \( \phi \) we denote with \( \mathcal{C}_{x_i} = \{C^i_1, C^i_2, C^i_3, C^i_4\} \) the four clauses \( x_i \) appears in, where \( C^i_1, C^i_2 \) are clauses with positive occurrences of \( x_i \), and \( C^i_3, C^i_4 \) are clauses containing negative occurrences of \( x_i \). We construct a vertex-colored graph \( G = (V, E) \) from \( \phi \) as follows. For each variable \( x_i \) we introduce six vertices \( t_1, t_2, f_1, f_2, h_t, \) and \( h_f \) and for each clause \( C_j \) with occurrences of variables \( x_{j_1}, x_{j_2}, x_{j_3} \), we add up to three vertices \( c_k \) for each \( k \in \{j_1, j_2, j_3\} \). (In case a clause has less than three literals we add only one or two vertices.) If the connection to the variable is clear, we also write \( c^i_1, c^i_2, c^i_3 \), and \( c^i_4 \) for the vertices introduced for this variable’s occurrences in the clauses \( C^i_1, C^i_2, C^i_3, C^i_4 \), respectively. Furthermore, we add for each variable \( x_i \) the edges \( \{h_t, t_1\}, \{h_t, t_2\}, \{h_t, f_1\}, \{h_f, f_2\} \), as well as for each clause \( C_j \) all possible edges between the three vertices introduced for \( C_j \). For each variable \( x_i \) we introduce five colors, namely \( z^i_1, z^i_2, z^i_3, z^i_4, \) and \( z^i_5 \). We set \( \gamma(h_t) = \gamma(h_f) = z_h \). Finally, we set \( \gamma(t_1) = \gamma(c^i_1) = z^i_1 \). Equivalently for \( t_2, f_1, \) and \( f_2 \). See Figure 2 for an example. In total we create \( 6n + 3m \) many vertices and \( 4n + 3m \) many edges, thus the reduction is polynomial. All variable and clause gadgets are independent components and only consist of paths of length three and triangles, hence \( G \) is a proper interval graph.

![Figure 2](image-url) Illustrations of three variable gadgets and a clause gadget as interval representations.

To establish the correctness of our reduction for 1-BDS we first introduce a canonical type of solutions for the graphs produced by our reduction. We call \( V_D \) canonical, if for each variable \( x_i \) we either find \( \{h_t, f_1, f_2, c^i_1, c^i_2\} \subset V_D \) or \( \{h_f, t_1, t_2, c^i_1, c^i_2\} \subset V_D \). If for a variable \( x \in X \) and a 1-balanced dominating set \( V_D \subseteq V \) we find one of the two above sets in \( V_D \), we say \( x \) is in canonical form in \( V_D \). The next lemma shows that if \( G \) has a 1-balanced dominating set we can turn it into a canonical one.

**Lemma 2.4.** Let \( G = (V,E) \) be a graph generated from a 2P2N-3SAT formula \( \phi \) with clause set \( \mathcal{C} = \{C_1, \ldots, C_m\} \) as above and \( V_D \subseteq V \) a 1-balanced dominating set, then \( V_D \) can be transformed into a canonical 1-balanced dominating set in \( O(|V|) \) time.

**Proof.** Let \( x \) be not in canonical form in \( V_D \). Since \( V_D \) is a 1-balanced dominating set we know that either \( h_t \) or \( h_f \) of \( x \) is in \( V_D \). Without loss of generality assume that \( h_t \in V_D \). Consequently, we find that \( f_1, f_2 \in V_D \) and \( c^i_1, c^i_2 \notin V_D \). Now, we obtain the set \( V_D^r \) from \( V_D \) by removing any occurrence of \( t_1 \) or \( t_2 \) from \( V_D \) and inserting all missing elements of \( \{c^i_1, c^i_2\} \).
Clearly $x$ is in canonical form in $V_D'$. We need to show that $V_D'$ is still a 1-balanced dominating set. It is straight forward to verify that every color appears exactly once in $V_D'$ if $V_D$ was 1-balanced. Now assume there was a vertex $u \in V$ that is not dominated by any vertex in $V_D'$. Yet, we at most deleted $t_1$ and $t_2$ in $V_D'$ but since $h_t \in V_D'$ both and all their neighbors are dominated. As our operations only affected vertices introduced for $x$ and occurrences of $x$ we can simply iterate this process for each variable until every variable $x_i$ is in canonical form.

\[\blacktriangleleft\]

**Theorem 2.5.** The $f$-balanced dominating set problem on a graph $G = (V, E)$ with a color assignment of the vertices $\gamma : V \to \{1, \ldots, k\}$ is NP-complete, even if $G$ is a proper interval graph and $f = 1$.

**Proof.** Let $G = (V, E)$ be constructed from a 2P2N-3SAT formula $\phi$ with clause set $C = \{C_1, \ldots, C_m\}$ as above and let $V_D$ be a 1-balanced dominating set of $G$. By Lemma 2.4 we can assume $V_D$ is canonical. We construct an assignment of the variables in $\phi$ by setting $x_i$ to true if its $h_i \in V_D$ and to false otherwise. Assume this assignment was not satisfying, i.e., there exists a clause $C_j \in C$ such that none of the literals in $C_j$ evaluates to true. For each positive literal of $C_j$ we then get that the corresponding variable $x_i$ was set to false. Hence, $h_f \in V_D$ for $x_i$ and consequently $c_i^1, c_i^2 \notin V_D$. Equivalently for each negative literal we find $h_t \in V_D$ and $c_i^1, c_i^2 \notin V_D$. As a result we find that none of the vertices introduced for literals in $C_j$ is in $V_D$ and especially that none of them is dominated as they are each others only neighbors. Yet, $V_D$ is a 1-balanced dominating set by assumption, a contradiction.

In the other direction, assume we are given a satisfying assignment of a 2P2N-3SAT formula $\phi$ with clause set $C = \{C_1, \ldots, C_m\}$. Furthermore, let $G = (V, E)$ be the graph constructed from $\phi$ as above. We form a canonical 1-balanced dominating set $V_D \subseteq V$ of $G$ in the following way. For every variable $x_i$ that is set to true in the assignment we add $\{h_t, f_1, f_2, c_i^1, c_i^2\}$ to $V_D$ and for every variable $x_i$ that is set to false we add $\{h_f, t_1, t_2, c_i^1, c_i^2\}$. This clearly is a 1-balanced set and it is canonical. It remains to argue that it dominates $G$. For the vertices introduced for variables this is clear, since we pick either $h_t$ or $h_f$, as well as $f_1, f_2$ or $t_1, t_2$ for every variable $x_i$. Now, assume there was a clause $C_j \in C$ and none of the vertices introduced for literals in $C_j$ was in $V_D$. Then, by construction of $V_D$, we find that for any positive (negative) occurrence of a variable $x_i$ in $C_j$ the variable $x_i$ was set to false (true). A contradiction to the assignment being satisfying.

\[\blacktriangleleft\]
index of the rightmost interval with its right endpoint left of \(I_i\)'s left endpoint. If no such interval exists for some interval \(I_i\), we set \(\text{prev}(I_i) = 0\).

For each \(\kappa \in \{1, \ldots, k\}\), let \(\mathbf{e}_\kappa\) denote the \(k\)-dimensional unit vector of the form \((0, \ldots, 0, 1, 0, \ldots, 0)\), where the element at the \(\kappa\)-th position is 1 and the rest are 0. For a subset \(\mathcal{I}' \subseteq \mathcal{I}\) we define a cardinality vector as the \(k\)-dimensional vector \(C_{\mathcal{I}'} = (c_1, \ldots, c_k)\), where each element \(c_i\) represents the number of intervals of color \(i\) in \(\mathcal{I}'\). We say \(C_{\mathcal{I}'}\) is valid if all \(c_i \leq f\) and the set \(\mathcal{I}'\) is independent.

The key observation here is that there are at most \(O((f + 1)^k)\) many different valid cardinality vectors as there are only \(k\) colors and we are interested in at most \(f\) intervals per color. In the following let \(U_j, j \in \{1, \ldots, n\}\), be the union of all valid cardinality vectors of the first \(j\) intervals in \(\mathcal{I}\). Furthermore, we store for each \(u \in U_j\) one representative interval set with \(u\) as its cardinality vector.

Let \(U_0 = \{(0, \ldots, 0)\}\) in the beginning. To compute an \(f\)-balanced independent set the algorithm simply iterates over all right endpoints of the intervals in \(\mathcal{I}\) and in the \(i\)-th step computes \(U_i\) as the set consisting of all valid cardinality vectors of the union \(\{u + \mathbf{e}_\kappa(I_i) \mid u \in U_{\text{prev}(I_i)}\} \cup U_{i-1}\). Correspondingly, we update the representative interval sets for \(U_i\). Finally, we check the cardinality vectors in \(U_n\) and return \(\text{true}\) in case there is one cardinality vector \(w \in U_n\) with entries being all \(f\) and \(\text{false}\) otherwise. Moreover, the representative interval set of \(w\) is an \(f\)-balanced independent set.

\[\blacktriangleright\textbf{Theorem 3.1.}\ Let \(G = (V, E)\) be an interval graph with a color assignment of the vertices \(\gamma : V \to \{1, \ldots, k\}\). We can compute an \(f\)-balanced independent set of \(G\) or determine that no such set exists in \(O(n \log n + k(f + 1)^k n)\) time.\]

\[\textbf{Proof.}\ Let \(\mathcal{I} = \{I_1, \ldots, I_n\}\) be an interval representation of \(G\) on which we execute our algorithm. For \(U_0\) the set just contains the valid cardinality vector with all zeros which is clearly correct. Let \(U_{i-1}\) be the set of valid cardinality vectors computed after step \(i - 1\). Now, in step \(i \leq n\) we calculate the set \(U_i\) as the union of \(U_{i-1}\) and the potential new solutions based on independent sets of intervals containing \(I_i\). Assume \(\mathcal{I}' \subseteq \{I_1, \ldots, I_n\}\) is an independent set of intervals such that its cardinality vector \(C_{\mathcal{I}'}\) is valid, but there is no valid cardinality vector \(C_{\mathcal{I}_x} \in U_i\) such that \(C_{\mathcal{I}_x}\) is larger or equal in every component than \(C_{\mathcal{I}'}\). Since \(U_{i-1}\) contained all valid cardinality vectors for the intervals in \(\{I_1, \ldots, I_{i-1}\}\) we know that \(C_{\mathcal{I}_x}\) is such that \(I_i \in \mathcal{I}'\). Yet, the set \(U_{\text{prev}(I_i)}\) contained all valid cardinality vectors for the set of intervals \(\{I_1, \ldots, I_{\text{prev}(I_i)}\}\). Since \(I_i\) has overlaps with all intervals in \(\{I_{\text{prev}(I_i)}, \ldots, I_{I_{i-1}}\}\) and hence cannot be in any independent set with any such interval we can conclude that \(C_{\mathcal{I}_x} - \mathbf{e}_\kappa(I_i) \in U_{\text{prev}(I_i)}\). Thus, we also find \(C_{\mathcal{I}_x} \in U_i\), a contradiction.

Next we consider the running time. The key observation is that there are at most \((f + 1)^k\) different valid cardinality vectors. Checking the validity can be done in \(O(1)\) time for each new vector as only one entry changes. Computing the sets \(U_i\) can be done in time \(O(k(f + 1)^k)\), by storing the cardinality vectors in lexicographically sorted order for each set. Keeping the sets in sorted order does not require any extra running time, as \(U_0\) is clearly sorted in the beginning (it only contains one element) and we only increase the same entry for each vector in \(U_{\text{prev}(I_i)}\) when forming the union, thus not changing their ordering. Hence, the set \(U_{\text{prev}(I_i)}\) and \(U_{i-1}\) can be assumed to be sorted in lexicographic order. Consequently, by merging from smallest to largest element the set \(U_i\) is again lexicographically sorted after the union. Furthermore, we can easily discard double entries by comparing also against the vector we inserted last into \(U_i\). Finally, we have to sort the intervals themselves. Using standard sorting algorithms this works in \(O(n \log n)\) time. Altogether, this results in a running time of \(O(n \log n + k(f + 1)^kn)\). \[\blacktriangleright\]
3.2 An FPT Algorithm Parameterized by the Vertex Cover Number

Here we will give an alternative FPT algorithm for \( f \)-BIS, this time parameterized by the vertex cover number \( \tau(G) \) of \( G \), i.e., the size of a minimum vertex cover of \( G \). This alternative algorithm works on general graphs, not only on interval graphs. In the following, for a vertex set \( S \subseteq V \), \( N(S) \) denotes the neighborhood of \( S \) consisting of all vertices adjacent to a vertex of \( S \).

**Lemma 3.2.** Let \( G = (V,E) \) be a graph. Consider a vertex cover \( V_c \) in \( G \) and its complement \( V_{ind} = V \setminus V_c \). Then any maximal independent set \( M \) of \( G \) can be constructed from \( V_{ind} \) by adding the subset \( M \cap V_c \) of \( V_c \) and removing its neighborhood in \( V_{ind} \), namely \( M = (V_{ind} \cup (M \cap V_c)) \setminus N(M \cap V_c) \).

**Proof.** For a fixed but arbitrary maximal independent set \( M \), in the following, we denote the set \((V_{ind} \cup (M \cap V_c)) \setminus N(M \cap V_c)\) as \( M_{\text{swap}} \).

We first prove the independence of \( M_{\text{swap}} \). Note that by the definition of a vertex cover \( V_{ind} \) is an independent set. Furthermore, the set \((M \cap V_c)\), as a subset of the independent set \( M \), is also independent. Then, in the union \( V_{ind} \cup (M \cap V_c) \) of these two independent sets, any adjacent pair of vertices must contain one vertex in \( M \cap V_c \) and one in \( V_{ind} \). Hence, after removing all the neighboring vertices of \( M \cap V_c \), the set \( M_{\text{swap}} \) is independent.

Next we prove that \( M \subseteq M_{\text{swap}} \). Assume there exists one vertex \( v_m \) in \( M \) but not in \( M_{\text{swap}} \). Since \( v_m \in M \) it must also be in the set \( V_{ind} \cup (M \cap V_c) \). With the assumption that \( v_m \notin M_{\text{swap}} \), we get that \( v_m \) must be in \( N(M \cap V_c) \). Consequently, \( v_m \) is in the independent set \( M \) and is at the same time a neighbor of vertices in \( M \), a contradiction.

Finally we prove \( M = M_{\text{swap}} \). We showed above that \( M_{\text{swap}} \) is an independent set and also \( M \subseteq M_{\text{swap}} \). Since \( M \) is a maximal independent set by assumption we get \( M = M_{\text{swap}} \).

**Lemma 3.3.** Let \( G = (V,E) \) be a graph with vertex cover number \( \tau(G) \). There are \( O(2^{\tau(G)}) \) maximal independent sets of \( G \).

**Proof.** Consider a minimum vertex cover \( V_c \) in \( G \) and its complement \( V_{ind} = V \setminus V_c \). Note that since \( V_c \) is a (minimum) vertex cover, \( V_{ind} \) is a (maximum) independent set. Furthermore, any maximal independent set \( M \) of \( G \) can be constructed from \( V_{ind} \) by adding \( M \cap V_c \) and removing its neighborhood in \( V_{ind} \), namely \( M = (V_{ind} \cup (M \cap V_c)) \setminus N(M \cap V_c) \) by Lemma 3.2.

Thus there are \( O(2^{\tau(G)}) \) maximal independent sets of \( G \).

**Theorem 3.4.** Let \( G = (V,E) \) be an interval graph with a color assignment of the vertices \( \gamma: V \to \{1, \ldots, k\} \). We can compute an \( f \)-balanced independent set of \( G \) or determine that no such set exists in \( O(2^{\tau(G)} \cdot n) \) time.

**Proof.** According to Lemma 3.3, there are \( O(2^{\tau(G)}) \) maximal independent sets of \( G \). The basic idea is to enumerate all the \( O(2^{\tau(G)}) \) maximal independent sets and compute their maximum balanced subsets. Enumerating all maximal independent sets of an interval graph takes \( O(1) \) time per output \([19]\). Given an arbitrary independent set of \( G \) we can compute an \( f \)-balanced independent subset in \( O(n) \) time or conclude that no such subset exists. Therefore, the running time of the algorithm is \( O(2^{\tau(G)} \cdot n) \).

4 A 2-Approximation for the 1-Max-Colored Independent Set

We note that the NP-completeness of 1-BIS implies that 1-MCIS is an NP-hard optimization problem. Moreover, the APX-hardness of Job Interval Scheduling Problem \([21]\) implies...
that 1-MCIS does not have a PTAS approach as well. Spieksma [21] showed that a simple sweep-line algorithm can provide approximation ratio 2 for JISP. Now we first review this sweep-line algorithm and show that this algorithm provides ratio 2 for 1-MCIS problem.

First, we sort the intervals from left to right based on their right end-points. Then, our algorithm scans the intervals from left to right, and at each step selects greedily an interval of a distinct color such that no interval of the same color has been selected before. Moreover, we maintain a solution array $S$ of size $k$ to store the selected intervals.

For each interval $I_i$ in this order, we check if the color of $I_i$ is still missing in our solution (by checking if $S[\gamma(I_i)]$ is not yet occupied). If yes, we store $I_i$ in $S[\gamma(I_i)]$ and remove all the remaining intervals overlapping $I_i$. Otherwise, if $S[\gamma(I_i)]$ is not empty, we remove $I_i$ and continue scanning the intervals. This process is repeated until all intervals are processed. Then, by using a simple charging argument on the colors in an optimal solution that are missing in our greedy solution, we obtain the desired approximation factor.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure3.png}
\caption{Comparison of a solution $S$ of the algorithm and an optimal solution $O$. Subset $M \subseteq O$ contains two colors (red and blue) missing from $S$, but each interval in $M$ contains the right endpoint of a different interval from $S$.}
\end{figure}

\begin{theorem}
Let $G = (V,E)$ be an interval graph with a color assignment of the vertices $\gamma: V \rightarrow \{1, \ldots, k\}$. In $O(n \log n)$ time, we can compute an independent set with at least $\lceil \frac{c}{2} \rceil$ colors, where $c$ is the number of colors in a 1-max-colored independent set.
\end{theorem}

\begin{proof}
It is clear from the above description that the greedy algorithm finds an independent set. We maintain a solution array $S$, and it is possible to check if an interval of a particular color is already available in $S$ in constant time. Therefore, the entire algorithm runs in $O(n \log n)$ time.

In order to prove the approximation factor, we compare the solution $S$ of our greedy algorithm with a fixed 1-max-colored independent set $O$ (see Figure 3). Let $M = \{I_i \in O \mid \exists I_j \in S \text{ with } \gamma(I_j) = \gamma(I_i)\}$ be the subset of $O$ consisting of intervals of missing colors in $S$. Now, consider an interval $I_m \in M$. There must be at least one interval $I_s \in S$, whose right endpoint is contained in the interval $I_m$. Otherwise, since there is no interval of the same color as $I_m$ in $S$, the greedy algorithm would scan $I_m$ as the interval with the leftmost right endpoint in the process and select it in $S$. Thus, the function $\rho$, which maps each interval $I_m$ in $M$ to an interval $I_s$ in $S$ such that $I_s$ is the rightmost interval in $S$ with its right endpoint is contained in $I_m$, is well-defined. Furthermore, $\rho$ is an injective function because of the independence of the set $M$. Therefore, we can conclude that the cardinality of the set $S$ is greater than or equal to the cardinality of $M$. Note that, $|M| + |S| \geq |O|$. Hence, $S$ has size at least $\lceil \frac{c}{2} \rceil$.
\end{proof}
Correction to an earlier version of this paper. In an earlier version of this work \[6\], we presented a local search approach and claimed it being a PTAS for the 1-MCIS problem. After its publication date, we found the paper by Spieksma \[21\] and noted that our claim was in contradiction with their APX-hardness result of the Job Interval Scheduling Problem. We rechecked our charging argument and found an error in our proof. More precisely, let \( \mathcal{L} \) be the solution of the local search algorithm, let \( \mathcal{O} \) be an optimal solution, and let \( H \) be the graph induced by \( \mathcal{L} \cup \mathcal{O} \) that has one vertex for each interval and two vertices are connected by an edge if the corresponding intervals intersect or belong to the same color class. We distinguish between these two types of edges: the former edges are called interval-edges and the latter are called color-edges. We claimed that “\( H \) is a planar graph” \[5\, \text{Lemma 4}\]. This statement is incorrect; see a counterexample in the following figure. Here, the graph \( H \) induced by \( \mathcal{L} \cup \mathcal{O} \) contains color-edges \((L_i, O_j)\) and \((L_{ij}, O_{ij})\) as well as interval-edges \((L_i, O_{ij})\) and \((L_{ij}, O_j)\) for \(i, j \in \{1, 2, 3\}\) and \(i \neq j\). The color-edges \((O_i, L_i)\) and the internally disjoint paths \((L_i, O_{ij}, L_{ij}, O_j)\) for \(i, j \in \{1, 2, 3\}\) and \(i \neq j\) together yield a \( K_{3,3}\)-minor of \( H \), which contradicts the planarity of \( H \).

5 Conclusions

In this paper, we have studied the \( f \)-Balanced Independent and Dominating set problem for interval graphs. We proved that these problems are \( \text{NP} \)-complete and obtained algorithmic results for the \( f \)-Balanced Independent Set problem. An interesting direction for future work is to obtain algorithmic results for the \( f \)-Balanced Independent Set problem for other geometric intersection graphs, e.g., rectangle intersection graphs, unit disk graphs etc. Our results may help to tackle these problems since algorithms for computing (maximum weighted) independent sets of geometric objects in the plane often use algorithms for interval graphs as subroutines. Another interesting problem is to design approximation or parameterized algorithm for the \( f \)-Balanced Dominating Set problem for interval graphs.

References

1. Pankaj K. Agarwal, Marc J. van Kreveld, and Subhash Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11(3-4):209–218, 1998. [doi:10.1016/S0925-7721(98)00028-5]
2. Egon Balas and Chang Sung Yu. Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, 15(4):1054–1068, 1986. [doi:10.1137/0215075]
3. Sandip Banerjee and Sujoy Bhore. Algorithm and hardness results on liar’s dominating set and \( k \)-tuple dominating set. In Charles J. Colbourn, Roberto Grossi, and Nadia Pisanti, editors, *Combinatorial Algorithms (IWOCA’19)*, volume 11638 of LNCS, pages 48–60. Springer, 2019. [doi:10.1007/978-3-030-25005-8_5]
4. Ken Been, Martin Nöllenburg, Sheung-Hung Poon, and Alexander Wolff. Optimizing active ranges for consistent dynamic map labeling. *Computational Geometry: Theory and Applications*, 43(3):312–328, 2010. [doi:10.1016/j.comgeo.2009.03.006]
5. Sujoy Bhore, Jan-Henrik Haunert, Fabian Klute, Guangping Li, and Martin Nöllenburg. Balanced independent and dominating sets on colored interval graphs. In *European Workshop on Computational Geometry (EuroCG’20)*, pages 66:1–66:6, 2020.
Sujoy Bhore, Jan-Henrik Haunert, Fabian Klute, Guangping Li, and Martin Nöllenburg. Balanced independent and dominating sets on colored interval graphs. In *Proc. the 47th Theory and Practice of Computer Science (SOFSEM’21)*, volume 12607 of *LNCS*, pages 89–103. Springer, 2021. doi:10.1007/978-3-030-67731-2_7

Michael W. Carter and Craig A. Tovey. When is the classroom assignment problem hard? *Operations Research*, 40(4):S28–S39, 1992. doi:10.1287/opre.40.4.528

Gerard J. Chang. Algorithmic aspects of domination in graphs. In *Handbook of Combinatorial Optimization*, pages 1811–1877. Springer, 1998. doi:10.1007/978-1-4613-0303-9_28

Maw-Shang Chang. Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM Journal on Computing*, 27(6):1671–1694, 1998.

Mustapha Chellali, Odile Favaron, Adriana Hansberg, and Lutz Volkmann. $k$-domination and $k$-independence in graphs: A survey. *Graphs and Combinatorics*, 28(1):1–55, 2012. doi:10.1007/s00373-011-1040-3

Valentin Garnero, Ignasi Sau, and Dimitrios M. Thilikos. A linear kernel for planar red–blue dominating set. *Discrete Applied Mathematics*, 217:536–547, 2017. doi:10.1016/j.dam.2016.09.095

Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs*. Elsevier, 2004.

U. I. Gupta, D. T. Lee, and Joseph Y.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12(4):459–467, 1982. doi:10.1002/net.3230120410

Jan-Henrik Haunert and Tobias Hermes. Labeling circular focus regions based on a tractable case of maximum weight independent set of rectangles. In *Proc. 2nd ACM SIGSPATIAL Workshop on MapInteraction*, 2014. doi:10.1145/2677068.2677069.

Teresa W. Haynes, Stephen Hedetniemi, and Peter Slater. *Fundamentals of domination in graphs*, volume 208 of *Pure and applied mathematics*. Dekker, 1998.

Robert W. Irving. On approximating the minimum independent dominating set. *Information Processing Letters*, 37(4):197–200, 1991. doi:10.1016/0020-0190(91)90081-N

C. Lekkerkerker and J. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51(1):45–64, 1962. doi:10.4064/fm-51-1-45-64

Kazuo Nakajima and S. Louis Hakimi. Complexity results for scheduling tasks with discrete starting times. *Journal of Algorithms*, 3(4):344–361, 1982. doi:10.1016/0196-6774(82)90030-X

Yoshio Okamoto, Takeaki Uno, and Ryuhei Uehara. Counting the number of independent sets in chordal graphs. *Journal of Discrete Algorithms*, 6(2):229–242, 2008. doi:10.1016/j.jda.2006.07.006

Panos M Pardalos and Yue Xue. The maximum clique problem. *Journal of Global Optimization*, 4(3):301–328, 1994. doi:10.1007/BF01098364

Frits C. R. Spieksma. Approximating an interval scheduling problem. In Klaus Jansen and Dorit S. Hochbaum, editors, *Proc. Approximation Algorithms for Combinatorial Optimization, International Workshop (APPROX’98)*, volume 1444 of *LNCS*, pages 169–180. Springer, 1998. doi:10.1007/3-540-49270-6_10

Frits C. R. Spieksma and Yves Crama. The complexity of scheduling short tasks with few starting times. Reports in operations research and systems theory M92-06, Maastricht University, 1992.

Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7

René Van Bevern, Matthias Mnich, Rolf Niedermeier, and Mathias Weller. Interval scheduling and colorful independent sets. *Journal of Scheduling*, 18(5):449–469, 2015. doi:10.1007/s10951-014-0398-5

Marc J. van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13(1):21–47, 1999. doi:10.1016/S0925-7721(99)00005-X
26 Frank Wagner and Alexander Wolff. A combinatorial framework for map labeling. In Sue H. Whitesides, editor, *Graph Drawing (GD’98)*, volume 1547 of *LNCS*, pages 316–331. Springer, 1998. doi:10.1007/3-540-37623-2_24

27 Ryo Yoshinaka. Higher-order matching in the linear lambda calculus in the absence of constants is NP-complete. In Jürgen Giesl, editor, *Term Rewriting and Applications (RTA’05)*, volume 3467 of *LNCS*, pages 235–249. Springer, 2005. doi:10.1007/978-3-540-32033-3_18