Temporal Unit Interval Independent Sets

Danny Hermelin ✉
Department of Industrial Engineering and Management,
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Yuval Itzhaki ✉
Faculty IV, Algorithmics and Computational Complexity, TU Berlin, Germany

Hendrik Molter ✉
Department of Industrial Engineering and Management,
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Rolf Niedermeier ✉
Faculty IV, Algorithmics and Computational Complexity, TU Berlin, Germany

── **Abstract** ──────────────

Temporal graphs have been recently introduced to model changes to a given network that occur throughout a fixed period of time. We introduce and investigate the TEMPORAL Δ INDEPENDENT SET problem, a temporal variant of the well known INDEPENDENT SET problem. This problem is *e.g.* motivated in the context of finding conflict-free schedules for maximum subsets of tasks, that have certain (changing) constraints on each day they need to be performed. We are specifically interested in the case where each task needs to be performed in a certain time-interval on each day and two tasks are in conflict on a day if their time-intervals overlap on that day. This leads us to considering TEMPORAL Δ INDEPENDENT SET on the restricted class of temporal unit interval graphs, *i.e.*, temporal graphs where each layer is unit interval.

We present several hardness results for this problem, as well as two algorithms: The first is a constant-factor approximation algorithm for instances where τ, the total number of time steps (layers) of the temporal graph, and Δ, a parameter that allows us to model some tolerance in the conflicts, are constants. For the second result we use the notion of order preservation for temporal unit interval graphs that, informally, requires the intervals of every layer to obey a common ordering. We provide an FPT algorithm parameterized by the size of minimum vertex deletion set to order preservation.

1 Introduction

Suppose there are n postdocs, each requesting access to your lab in the next τ days for conducting their experiments. Each postdoc submitted at the beginning of the semester an application form which specifies a τ-day schedule for lab experiments, where in each day, the postdoc's schedule specifies a single uninterrupted time-interval for their experiment. All of the n postdocs are very promising, and you wish to make sure that at least k of them are able to conduct their research throughout the entire time period of τ days. However, since the lab is small, it is preferable that no two postdocs share it at the same time. How can you find out if the lab can accept the application of at least k postdocs?

This problem can be classically modeled as an INDEPENDENT SET problem on an undirected graph G with n vertices, one for each postdoc, by which two vertices are connected by an edge if at *any* day the lab sessions of the two corresponding postdocs overlap. Given such a graph G with a vertex set V and an edge set E, a solution to INDEPENDENT SET will comprise a subset of vertices $V' \subseteq V$ such that no two vertices in V' are connected by an edge in E. In case that there exists an independent set V' with a cardinality at least k, there exist k postdocs whose research projects can be scheduled over the next τ days such that there are no conflicting lab sessions on any day.

However, by considering the static graph G comprising all conflicts on its own, we lose all the daily information of each postdoc. This can be a serious hindrance if we are willing to allow some leeway in the way we schedule the lab sessions. For example, given that practically every postdoc in the group is known to skip their lab session every once in a while, one might want to allow some overlaps in the schedule. Thus, one could assume that if the experiments of two postdocs do not overlap in more than Δ consecutive days, then they can still be scheduled together. This leads us to the TEMPORAL Δ INDEPENDENT SET problem which we introduce below.

If we wish to retain the daily information of each postdoc, we naturally have to generalize our graph to a *temporal* graph. Temporal graphs generalize static graphs by adding a discrete temporal dimension to the edge set. Formally, a temporal graph $\mathcal{G} = (V, \mathcal{E}, \tau)$ is an ordered triple consisting of a set V of vertices, a set $\mathcal{E} \subseteq \binom{V}{2} \times \{1, 2, \ldots, \tau\}$ of *time-edges*, and a maximal time label $\tau \in \mathbb{N}$. A temporal graph can be regarded as a set of τ consecutive *time steps*, in which each step is a static graph. For $t \in \{1, \ldots, \tau\}$, we define the t-th layer as $G_t = (V, E_t)$, where $E_t = \{\{u, v\} : (\{u, v\}, t) \in \mathcal{E}\}$. We refer to Casteigts *et al.* [6], Flocchini *et al.* [13], Kostakos [29], Latapy *et al.* [30] and Michail [34] for a more detailed background on temporal graphs.

We next extend the notion of independent sets to temporal graphs. We say a vertex set V' is a Δ-*independent set* in a temporal graph $\mathcal{G} = (V, \mathcal{E}, \tau)$ if V' is an independent set in the edge-*intersection* graph of every Δ consecutive time steps of $\mathcal{G}$. That is, for any pair of distinct vertices $u \neq v \in V'$ and $t \in \{1, \ldots, \tau - \Delta + 1\}$, there exists a $t' \in \{t, \ldots, t + \Delta - 1\}$ such that $\{u, v\} \notin E_{t'}$. We call this edge-intersection graph of every Δ consecutive time steps $G = (V, \bigcup_{i=1}^{\tau - \Delta + 1} \bigcap_{j=i}^{i + \Delta - 1} E_j)$ the *conflict graph*. With this notion in mind, we can now introduce the main problem we deal with in this paper. We give an example in Figure 1.
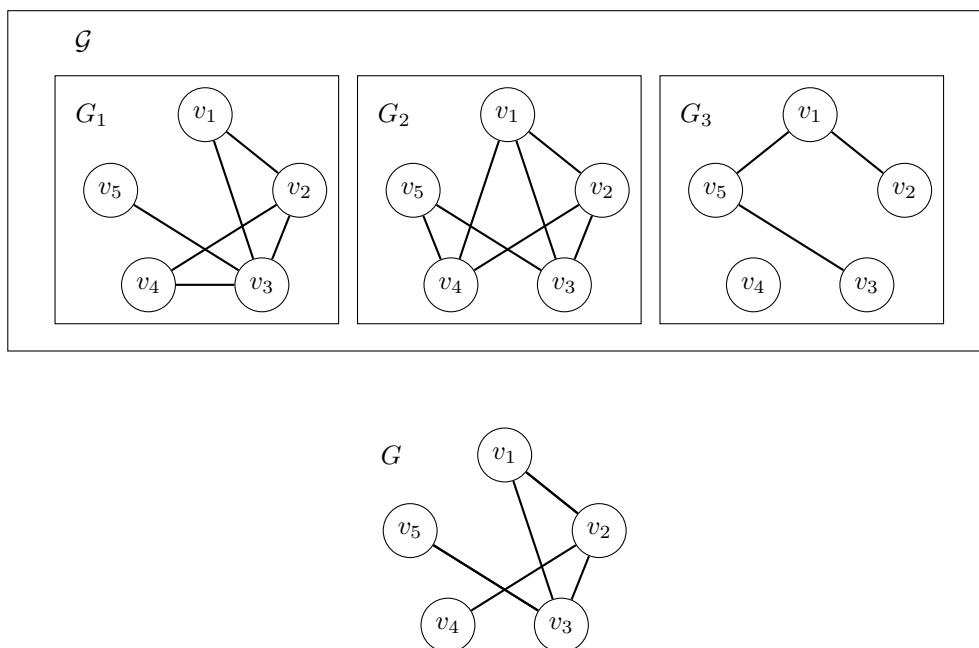
TEMPORAL Δ INDEPENDENT SET
Input: A temporal graph $\mathcal{G} = (V, \mathcal{E}, \tau)$ and two integers $k, \Delta \in \mathbb{N}$.
Question: Is there set $V' \subseteq V$ of vertices such that $|V'| \geq k$ and V' is an independent set in
the conflict graph $G = (V, \bigcup_{i=1}^{\tau - \Delta + 1} \bigcap_{j=i}^{i + \Delta - 1} E_j)$?

Temporal interval graphs.   Recall our initial problem of allocating lab space to the n postdocs. Observe that in this setting, each daily conflict graph G_t (corresponding to layer t of the input temporal graph) can also be represented by a set of n intervals, where each each interval indicates a lab session's time-interval of the corresponding scientist on day t. Such a representation is called an *interval representation* of G_t, and it is a unique property of *interval graphs*. As first defined by Hajós [21], a graph belongs to the class of interval graphs if there exists a mapping of its vertices to a set of intervals over a line such that two vertices are adjacent if and only if their corresponding intervals overlap. Interval graphs are used to model many natural phenomena which occur along the line of a one-dimensional axis, and have various applications in scheduling [3], computational biology [26], and more.

Figure 1 An example of TEMPORAL Δ INDEPENDENT SET instance with a temporal graph $\mathcal{G}$ with three layers and $\Delta = 2$. The graph G is the conflict graph of the instance. The vertex subset $V' = \{v_1, v_4, v_5\}$ is a maximum sized solution for this instance.

An important subclass of interval graphs is the class of *unit interval graphs*: A graph G is a *unit interval graph* if it has interval representation where all intervals are of the same length. It is well-known that this graph class is equivalent to the class of *proper interval graphs*, graphs with interval representation where no interval is properly contained in another [36]. The restriction to unit interval graphs is quite natural for our lab allocation problem, since in many cases one can assume that all experiments take roughly the same time. Thus, we will also focus on the TEMPORAL Δ INDEPENDENT SET problem restricted to temporal unit interval graphs.

We focus in this paper on temporal interval graphs, that is temporal graphs $\mathcal{G}$ where each layer is an interval graph. More specifically, most of our work is focused on temporal unit interval graphs, *i.e.*, the case where each layer is a unit interval graph. It should be clear at this point that assuming that all lab sessions take the same amount of time, our lab allocation problem is precisely the TEMPORAL Δ INDEPENDENT SET problem restricted to temporal unit interval graphs.

Geometric interpretation. The restriction of TEMPORAL Δ INDEPENDENT SET to *temporal interval graphs* gives rise to an elegant and useful geometric interpretation. Let us first consider the case of $\Delta = 1$. A *t-track (unit) interval* is a union of t (unit) intervals, one each from t parallel lines, and a *family of t-track (unit) intervals* is a set of t-track (unit) intervals which have intervals all from the same t parallel lines. Clearly, the set of all τ intervals corresponding to a single vertex in G can be represented by a τ-track interval, and so for $\Delta = 1$, the conflict graph G of $\mathcal{G}$ is an intersection graph of a family of τ-track unit intervals. For $\Delta > 1$, we need to use hyperrectangles and hypercubes instead of intervals and unit intervals. We define *t-track hyperrectangles (hypercubes)* and *families of t-track hyperrectangles (hypercubes)* in the natural manner. In this way, we get that our conflict graph G is an intersection graph of a family of $(\tau - \Delta + 1)$-track Δ-dimensional hypercubes.

Order-preserving temporal interval graphs.   Considering our introductory example, it may be a reasonable assumption that some postdocs generally prefer to conduct their experiments in the morning while others prefer to work in the evenings. In this scenario, we have a natural ordering on the time-intervals of the postdocs that stays the same or at least does not change much over the time period of τ days. We use the notion of *order-preserving temporal graphs* to formalize this setting. Order preservation on temporal interval graphs was first introduced by Fluschnik *et al.* [14]. A *temporal interval graph* is said to be *order-preserving* if it admits a vertex ordering $<_V$ such that each of its time steps can be represented by an interval model such that *both* the right-endpoints and left-endpoints are ordered by $<_V$. Fluschnik *et al.* [14] show that the recognition of order-preserving temporal unit interval graphs can be done in linear time, and also offer a metric to measure the distance of a temporal interval graph from being order-preserving, which they call the "shuffle number". It measures the maximum pairwise disagreements in the vertex ordering of any two consecutive layers. We propose an alternative metric to measure the distance to order preservation. Our distance is simply the minimum number k of vertices to be deleted in order to obtain an order-preserving temporal interval graph, and we call it the order-preserving vertex deletion (OPVD) metric.

1.1   Our results

We present both hardness results and positive results regarding TEMPORAL Δ INDEPENDENT SET in temporal unit interval graphs. Our initial hardness results in Section 2.2 are based on adaptations of the hardness result of Marx [31] for INDEPENDENT SET on axis parallel squares in the plane, and Jiang [25] for INDEPENDENT SET on t-track graphs. These hardness results apply for quite restricted instances of temporal interval graphs. For instance, we show that our problem is NP-hard even for $\tau = 2$.

In Section 3, we present an approximation algorithm for MAXIMUM TEMPORAL Δ INDEPENDENT SET (the canonical optimization variant of TEMPORAL Δ INDEPENDENT SET where we want to maximize the independent set size) restricted to temporal unit interval graphs. The algorithm exploits the geometric interpretation of the problem discussed above to achieve an approximation factor of $(\tau - \Delta + 1) \cdot 2^\Delta$ in polynomial time, and can also be extended to the weighted case.

In Section 4, we turn to discuss order-preserving temporal unit interval graphs. As mentioned above, this concept was introduced by Fluschnik *et al.* [14] for temporal interval graphs. We show that computing the OPVD set (*i.e.*, the set of vertices whose removal leaves an order-preserving graph) of a unit interval temporal graph is NP-hard. We complement this result by providing an FPT algorithm for computing an OPVD set when parameterized by the solution size. This leads to an FPT algorithm for Δ INDEPENDENT SET in temporal unit interval graph when parameterized by minimum OPVD set.

1.2   Related Work

By now, there is already a significant body of research related to temporal graphs in general [6, 13, 29, 34], as well as graph problems cast onto the temporal setting [2, 4, 14, 23, 39, 33, 32]. To the best of our knowledge, the problem of TEMPORAL Δ INDEPENDENT SET has not been studied previously, but our definition is highly inspired by the TEMPORAL Δ CLIQUE problem [4, 23, 39].

The classical static INDEPENDENT SET problem is clearly a special case of TEMPORAL Δ INDEPENDENT SET when $\tau = 1$. While INDEPENDENT SET is NP-complete for general undirected graphs [15], it is solvable in linear time on interval graphs and some of their

generalizations [16, 24, 37]. Thus, TEMPORAL Δ INDEPENDENT SET on temporal interval graph is linear time solvable when $\tau = 1$. Moreover, for larger values of τ, the TEMPORAL 1 INDEPENDENT SET problem is a special case of INDEPENDENT SET in τ-interval graphs [19]. Bar-Yehuda *et al.* [3] presented a 2τ approximation algorithm for INDEPENDENT SET in τ-interval graphs, while Fellows *et al.* [12] and Jiang [25] studied this problem from the perspective of parameterized complexity.

When $\tau = \Delta$, TEMPORAL Δ INDEPENDENT SET is a special case of INDEPENDENT SET on intersection graphs of τ-dimensional hyperrectangles. Marx [31] showed that INDEPENDENT SET is NP-complete and W[1]-hard with respect to the solution size when restricted to the intersection graphs of axis-parallel unit squares in the plane. Chlebík and Chlebíková [8] proved, for instance, that MAXIMUM INDEPENDENT SET is APX-hard for intersection graphs of *d*-dimensional *rectangles*, yet on such graphs the optimal solution can be approximated within a factor of *d* [1]. On intersection graphs of *d*-dimensional *squares* MAXIMUM INDEPENDENT SET admits a *polynomial time approximation scheme (PTAS)* for a constant *d* [7, 22, 28].

2   Preliminaries and Basic Results

In this section, we first introduce all temporal graph notation used in this work and then present some basic hardness results for our problem. We use standard notation and terminology from parameterized complexity theory [10].

2.1   Notation and Definitions

Let $a, b \in \mathbb{N}$ such that $a < b$. We use the notation $[a : b]$ as shorthand for $\{x \mid x \in \mathbb{N} \wedge a \le x \le b\}$. We denote $[a : b]$ by $[b]$ when $a = 1$. For $a, b \in \mathbb{R}$ such that $a < b$ we denote with $[a, b] \subseteq \mathbb{R}$ the set of real numbers $\{x \mid x \in \mathbb{R} \wedge a \le x \le b\}$.

Let $G = (V, E)$ denote an undirected graph, where V denotes the set of vertices and $E \subseteq \{\{v, w\} \mid v, w \in V,\ v \ne w\}$ denotes the set of edges. For a graph G, we also write $V(G)$ and $E(G)$ to denote the set of vertices and the set of edges of G, respectively. We denote $n := |V|$. Given an ordering $<_V$ over the vertices V of a graph $G = (V, E)$ in which v_i is the i-th vertex in the ordering, we denote by $V_{[a:b]}$ the set $\{v_i \mid i \in [a : b]\}$ and by $G_{[a:b]}$ the graph induced by $V_{[a:b]}$. We use the notation $\text{index}_{<_V}(v)$ to return the ordinal position of v in $<_V$.

An undirected *temporal graph* $\mathcal{G} = (V, \mathcal{E}, \tau)$ is an ordered triple consisting of a set V of vertices, a set $\mathcal{E} \subseteq \binom{V}{2} \times [\tau]$ of *time-edges*, and a maximal time label $\tau \in \mathbb{N}$. Given a temporal graph $\mathcal{G} = (V, \mathcal{E}, \tau)$, we denote by E_t the set of all edges that are available at time t, that is, $E_t := \{\{v, w\} \mid (\{v, w\}, t) \in \mathcal{E}\}$ and by G_t the t-th *layer* of $\mathcal{G}$, that is, $G_t := (V, E_t)$. We also denote by

- $G_1 \cap G_2$ the *edge-intersection graph* of G_1 and G_2, formally $G_1 \cap G_2 := (V, E_1 \cap E_2)$,
- $G_1 \cup G_2$ the *edge-union graph* of G_1 and G_2, formally $G_1 \cup G_2 := (V, E_1 \cup E_2)$, and
- $\mathcal{G} - V'$ the *temporal graph* induced by $V \setminus V'$, formally $\mathcal{G} - V' := (V \setminus V', \mathcal{E}', \tau)$ with $\mathcal{E}' = \{(\{v, u\}, t) \mid v, u \in V \setminus V' \wedge (\{v, u\}, t) \in \mathcal{E}\}$.

Interval graphs. An undirected graph is an *interval* graph if there exists a mapping from its vertices to intervals on the real line so that two vertices are adjacent if and only if their intervals intersect [17]. Such a representation is called an *intersection model* or an *interval representation*. Formally, given a graph $G = (V, E)$, an *interval representation* is a mapping $\rho : V \to I \subseteq \mathbb{R}$ of each vertex $v \in V$ to an interval $\rho(v)$ such that $E = \{\{v, u\} \mid v, u \in V \wedge \rho(v) \cap \rho(u) \ne \varnothing\}$. We denote by $\text{right}_\rho(u)$ and by $\text{left}_\rho(u)$ the real value of the right and

left endpoints of v's associated interval on the interval representation ρ; the subscript ρ will be omitted if it is clear from the context to which representation we refer. We denote by $\rho(G)$ the entire representation of G.

Unit interval graphs. An interval graph is a *unit interval graph* if it has an interval representation in which all intervals have exactly the same length. It is exactly the class of interval graphs which have a proper interval representation, a representation in which no interval is properly contained within another [36].

2.2   Basic Hardness Results

Since TEMPORAL Δ INDEPENDENT SET generalizes the classic INDEPENDENT SET problem, it is clearly NP-hard [27], W[1]-hard when parameterized by the solution size k [10], and does not admit any efficient approximation algorithms as well. In this section, we show that TEMPORAL Δ INDEPENDENT SET remains computationally hard when restricted to temporal unit interval graphs with a constant number of layers.

We begin with the case that $\Delta = 1$. Here the conflict graph is simply the union of all layers of $\mathcal{G}$. Thus, as mentioned in Section 1, the class of all possible conflict graphs is precisely the class of τ-track unit intervals. Thus, the following hardness result directly follows from the known hardness results for INDEPENDENT SET in 2-track unit interval graph [3, 12, 25].

▶ **Proposition 1.** *TEMPORAL Δ INDEPENDENT SET on temporal unit interval graphs is* NP-*hard,* APX-*hard, and* W[1]-*hard with respect to the solution size k for $\tau \geq 2$ and $\Delta = 1$.*

Now that we have seen hardness for $\tau \geq 2$ and $\Delta = 1$, we proceed with case of $\Delta = \tau$, by which the class of all conflict graphs is precisely the class of τ-dimensional hypercubes (intersection) graphs. Marx [31] showed that INDEPENDENT SET on axis-parallel unit squares in the plane is NP-hard and W[1]-hard with respect to the solution size. For our setting, this result implies the following.

▶ **Proposition 2.** *TEMPORAL Δ INDEPENDENT SET on temporal unit interval graphs is* NP-*hard and* W[1]-*hard with respect to the solution size k for $\tau = \Delta \geq 2$.*

Note however, that the problem admits a PTAS in this case, whenever $\tau = \Delta = \mathcal{O}(1)$ [11].

3   Approximation Algorithms

In this section, we present our polynomial-time approximation algorithm for WEIGHTED MAXIMUM TEMPORAL Δ INDEPENDENT SET on temporal unit interval graphs. We will use the geometric representation of our conflict graph discussed in Section 1. Recall that the conflict graph G has a $(\tau - \Delta + 1)$-track Δ-dimensional hypercube representation. This geometric property of the conflict graph can be exploited in the development of an approximation algorithm, since we can upper-bound the size of the largest independent set in the neighborhood of any vertex of G.

▶ **Lemma 3.** *Let G be the conflict graph of MAXIMUM TEMPORAL Δ INDEPENDENT SET on a temporal unit interval graph. For each vertex it holds that the largest independent set in the graph induced by $N[v]$ has at most $2^{\Delta}(\tau - \Delta + 1)$ vertices.*
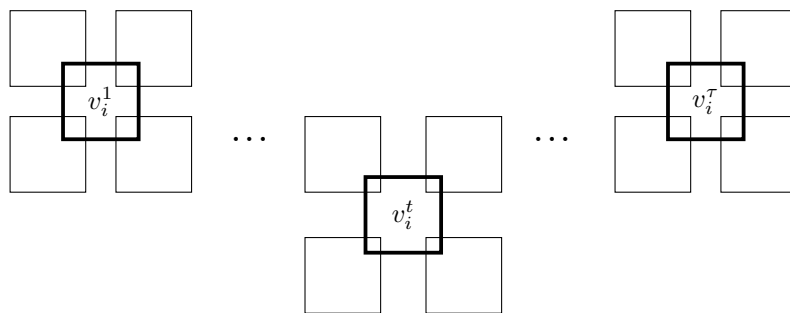
Figure 2 A set of τ axis parallel Δ-hypercubes cannot be intersected by more than $2^\Delta \tau$ disjoint axis parallel Δ hypercubes.

Proof. Let $\rho(G)$ denote the $\tau - \Delta + 1$-track Δ-dimensional hypercube family representation of G. Furthermore, for a vertex v of G, let $\rho(v) \in \rho(G)$ denote the $\tau - \Delta + 1$-track Δ-dimensional hypercube corresponding to v in $\rho(G)$. Consider a single hypercube C of $\rho(v)$. Observe that any other hypercube in $\rho(G)$ intersecting C must include a corner of C. Consequently, there are at most 2^Δ disjoint hypercubes in $\rho(G)$ intersecting C, and so at most $2^\Delta(\tau - \Delta + 1)$ disjoint $\tau - \Delta + 1$-track hypercubes intersect $\rho(v)$ (see Figure 2). The lemma thus follows. ◀

Lemma 3 allows to adopt a greedy approach for solving WEIGHTED MAXIMUM TEMPORAL Δ INDEPENDENT SET on temporal unit interval graphs. We iteratively put the largest-weight vertex into the independent set and remove its neighborhood afterwards. This gives rise to Algorithm 1 below.

Algorithm 1 Maximum Neighborhood Weight Algorithm.

Input: An undirected graph $G = (V, E)$
Output: An independent set
    while V is not empty **do**
        Pick vertex v with maximum $w(v)$ and add it to S.
        Remove $N[v]$ from V.
    end while
    return S

We show that the greedy strategy described in Algorithm 1 terminates within $\mathcal{O}(n^2)$ time, and achieves an approximation factor of $(\tau - \Delta + 1) \cdot 2^\Delta$.

▶ **Theorem 4.** *WEIGHTED MAXIMUM TEMPORAL Δ INDEPENDENT SET can be approximated within a factor of $(\tau - \Delta + 1) \cdot 2^\Delta$ in $\mathcal{O}(n^2)$ time on temporal unit interval graphs.*

Proof. Let $\mathcal{G}$ be the input temporal unit interval graph of WEIGHTED MAXIMUM TEMPORAL Δ INDEPENDENT SET. We can compute the conflict graph G in linear time. We know that G is a $(\tau - \Delta + 1)$-track Δ-dimensional hypercube intersection graph. Given $G = (V, E)$ with weight function w as an input, the output of Algorithm 1 will be obviously an independent set. From Lemma 3 we know that it cannot have smaller weight than $2^\Delta(\tau - \Delta + 1)$ times of the optimal solution's weight. This is due to the fact that at each step of Lemma 3 we add one vertex to the solution and remove its neighbors from the candidate set. The optimal solution is of course an independent set, hence at each step we remove from the candidate set no more than $2^\Delta(\tau - \Delta + 1)$ vertices which are members of the optimal solution. Since we

pick in each step a maximum weight vertex, assuming the worst case in which we throw out a $2^\Delta(\tau - \Delta + 1)$-sized set of vertices with equal weight, we cannot have our solution's weight smaller than this ratio. Suppose that Algorithm 1 terminated after p steps, this means that there are p vertices in the solution and no more than $2^\Delta p(\tau - \Delta + 1)$ vertices in the optimal solution. ◀

4   Order-Preserving Temporal Interval Graphs

In this section, we investigate the computational complexity of TEMPORAL Δ INDEPENDENT SET on so-called order-preserving temporal unit interval graphs. The concept of order preservation was introduced by Fluschnik *et al.* [14]. In Section 4.1, we show that TEMPORAL Δ INDEPENDENT SET can be solved in polynomial time on order-preserving temporal graphs and along the way we generalize the concept to temporal (non-unit) interval graphs. In Section 4.2, we show how to solve TEMPORAL Δ INDEPENDENT SET on non-order-preserving temporal interval graphs via a "distance-to-triviality" parameterization [18]. To this end, we also give an FPT algorithm to compute a minimum vertex deletion set to order preservation with the set size as a parameter. Finally in Section 4.3, we show that computing a minimum vertex deletion set to order preservation is NP-hard.

4.1   A Polynomial-Time Algorithm for Order-Preserving Temporal Interval Graphs

We say an interval graph *agrees on* or is *compatible with* a total order if has an interval intersection model where the right endpoints of the intervals agree with the total order. Formally, an interval graph G agrees on $<_V$ if there exists an interval representation ρ for G such that for every two vertices $\forall v, u \in V$ whose ranking fulfills $v <_V u$, it holds that $\text{right}_\rho(v) < \text{right}_\rho(u)$. We call such ordering *right-endpoints (RE) orderings*. Clearly, any right-endpoints ordering is also a *left-endpoints* ordering of the mirrored intersection model.

▶ **Definition 5.** *A temporal interval graph is* order-preserving *if all of its layers agree on a single RE ordering.*

   Order-preserving temporal unit interval graphs can be recognized in linear time and a corresponding vertex ordering can be computed in linear time as well [14]. The computational complexity of recognizing order-preserving temporal *interval* graphs remains open.
   In the following, we show that RE orderings are preserved under both intersection and union of interval graphs. This means that the conflict graph of an RE order-preserving temporal graph is an interval graph that as well agrees on the RE ordering. We demonstrate this claim for interval graphs in Lemmas 6 and 7. We start with showing that the intersection of two interval graphs that agree on an RE ordering is again an interval graph that agrees on the ordering.

▶ **Lemma 6.** *Let G_1 and G_2 be interval graphs that agree on the total ordering $<_V$. Then $G_1 \cap G_2$ is an interval graph that agrees on $<_V$.*

Proof. Given two interval graphs which agree on an RE ordering, we can normalize their representations such that for each vertex, the right endpoints in both representations are the same. To compute an intersection model for their intersection graph, we can map each vertex to the intersection of their intervals in both representations. We then show that this mapping is an interval representation of the edge intersection graph.

Let V be a vertex set of size n and let $<_V$ be a total ordering on V such that for all $i, j \in [n]$ it holds $j < i \Leftrightarrow v_j <_V v_i$. Since both G_1 and G_2 agree on $<_V$, we can normalize their interval representations so that the right endpoint of the interval associated with each vertex lies on a natural number between 1 and n according to its ordinal position in $<_V$, formally $\text{index}_{<_V}(v_i) = i$. Alternatively, we can say that for an interval graph G_t an interval representation ρ_t exists such that each $v \in V$ is mapped to an interval of the form $\rho_t(v) = [a_v, \text{index}_{<_V}(v)]$ with $a_v \in \mathbb{R}$. In this normalized representation, the left endpoint of the interval lies on the real line between two natural numbers and is by definition smaller than the right endpoint, that is, $a_v < \text{index}_{<_V}(v)$.

Let ρ be a mapping from the vertex set V to a set of points on $\mathbb{R}$ such that it holds $\rho(v) = \rho_1(v) \cap \rho_2(v)$. To show that ρ is an interval representation of $G_1 \cap G_2$ we first show that $\rho(v)$ is a continuous interval for any $v \in V$, and that for any two vertices $v, u \in V$ it holds that $\rho(v) \cap \rho(u) \neq \varnothing \Leftrightarrow \rho_1(v) \cap \rho_1(u) \neq \varnothing \wedge \rho_2(v) \cap \rho_2(u) \neq \varnothing$.

By definition $\rho_1(v)$ and $\rho_2(v)$ are both intervals on the real line, they are therefore convex sets. As the mapping $\rho(v)$ is an intersection of two convex sets, it must as well be a convex set and therefore it is interval on the real line.

Let $v_j <_V v_i$, we show that if $\rho(v_i) \cap \rho(v_j) \neq \varnothing$ then both $\rho_1(v_i) \cap \rho_1(v_j) \neq \varnothing$ and $\rho_2(v_i) \cap \rho_2(v_j) \neq \varnothing$ must hold. As the interval representations ρ_1 and ρ_2 are both normalized, it immediately follows that $\text{right}_{\rho_1}(v_i) = \text{right}_{\rho_2}(v_i) = i$. Since both are closed intervals we know that either $\rho_1(v) \subseteq \rho_2(v)$ or $\rho_2(v) \subseteq \rho_1(v)$. Without loss of generality, let $\rho_1(v) \subseteq \rho_2(v)$; it follows that $\rho(v) = \rho_1(v)$. This means that if $\rho_2(v_i) \cap \rho_2(v_j) = \varnothing$, then also $\rho_1(v_i) \cap \rho_1(v_j) = \varnothing$ and therefore $\rho(v_i) \cap \rho(v_j) = \varnothing$. Regardless, it must hold that $j \in \rho_1(v_j) \cap \rho_2(v_j)$ as both interval representations of v_j have j as the right endpoint; it follows that $j \in \rho(v_j)$. This shows that if $j \in \rho(v_i)$ then $j \in \rho_1(v_i)$ and $j \in \rho_2(v_i)$. Therefore, for any $v_j <_V v_i$, if $\rho(v_i) \cap \rho(v_j) \neq \varnothing$, then both $\rho_1(v_i) \cap \rho_1(v_j) \neq \varnothing$ and $\rho_2(v_i) \cap \rho_2(v_j) \neq \varnothing$.

Suppose that $\rho(v_i) \cap \rho(v_j) = \varnothing$, but both $\rho_1(v_i) \cap \rho_1(v_j) \neq \varnothing$ or $\rho_2(v_i) \cap \rho_2(v_j) \neq \varnothing$. This contradicts that $v_j <_V v_i$ because if $\rho_1(v_i)$ contains any point $a \in \rho_1(v_j)$ with $a < j$, then it must contain also j because $\rho_1(v_i)$ is convex.

We have therefore an interval representation ρ which represents the graph $G_1 \cap G_2$ because $\rho(v) \cap \rho(u) \Leftrightarrow \{v, u\} \in E_1 \wedge \{v, u\} \in E_2$ for any $v, u \in V$. Notice that $G_1 \cap G_2$ agrees on $<_V$ because $\text{right}_\rho(v_i) = i$. ◀

Next, we show that the union of two interval graphs agreeing on an RE ordering yields an interval graph that also agrees on the ordering.

▶ **Lemma 7.** *Let G_1 and G_2 be interval graphs that agree on the total ordering $<_V$. The union $G = G_1 \cup G_2$ is an interval graph that agrees on $<_V$.*

Proof. The main concept of the proof is analogous to the one for Lemma 6. Given two interval graphs which agree on an RE ordering, we can normalize their representations such that for each vertex, the right endpoints in both representations are the same. To compute an intersection model for their *union* graph, we can map each vertex to the *union* of their intervals in both representations. We then show that this mapping is an interval representation of the edge-*union* graph.

Let ρ be a mapping from the vertex set V to a set of points on $\mathbb{R}$ such that it holds $\rho(v) = \rho_1(v) \cup \rho_2(v)$. To show that ρ is an interval representation of $G_1 \cup G_2$ we first show that $\rho(v)$ is a continuous interval for any $v \in V$, and that for any two vertices $v, u \in V$ it holds that $\rho(v) \cap \rho(u) \neq \varnothing \Leftrightarrow \rho_1(v) \cap \rho_1(u) \neq \varnothing \vee \rho_2(v) \cap \rho_2(u) \neq \varnothing$.

By definition $\rho_1(v)$ and $\rho_2(v)$ are both closed and normalized intervals on the real line such that $\text{right}_{\rho_1}(v_i) = \text{right}_{\rho_2}(v_i) = i$. As we observed in Lemma 6, since both intervals have the same right endpoint it holds that either $\rho_1(v) \subseteq \rho_2(v)$ or $\rho_2(v) \subseteq \rho_1(v)$. Without loss of generality, assume that $\rho_2(v) \subseteq \rho_1(v)$, it follows that $\rho(v) = \rho_1(v) = \rho_1(v) \cup \rho_2(v)$.
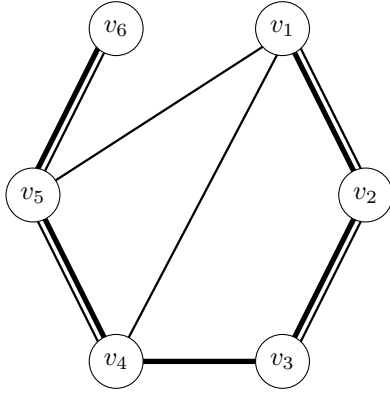
(a) Graph representations of the two interval graphs
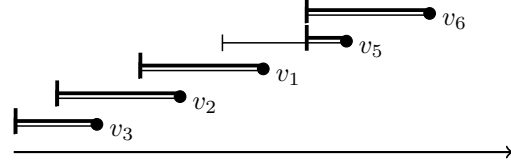
(b) Two intersection models of $G_1 - \{v_4\}$ and $G_2 - \{v_4\}$ which are compatible with $<_{V'}$

▨ **Figure 3** Two interval graphs, G_1 (thick) and G_2 (thin), that *do not* have a common RE ordering. The vertex subset $\{v_4\}$ is an OPVD set of the temporal graph $\mathcal{G} = [G_1, G_2]$ as both $G_1 - \{v_4\}$ and $G_2 - \{v_4\}$ agree on $<_{V'} = [v_3, v_2, v_1, v_5, v_6]$. Two compatible interval representations are illustrated in (b).

Suppose that $\rho(v_i) \cap \rho(v_j) \neq \varnothing$, but both $\rho_1(v_i) \cap \rho_1(v_j) = \varnothing$ or $\rho_2(v_i) \cap \rho_2(v_j) = \varnothing$. This contradicts that $v_j <_V v_i$ because if $\rho(v_i)$ contains any point $a \in \rho(v_j)$ with $a < j$, then it must contain also j because $\rho(v_i)$ is convex. If $j \in \rho(v_i)$ then trivially $j \in \rho_1(v_i)$.

If $\rho(v_i) \cap \rho(v_j) = \varnothing$ but either $\rho_1(v_i) \cap \rho_1(v_j) \neq \varnothing$ or $\rho_2(v_i) \cap \rho_2(v_j) \neq \varnothing$, then it contradicts the fact that either $\rho_1(v_i) \subseteq \rho_2(v_i) = \rho(v_i)$ or that $\rho_2(v_i) \subseteq \rho_1(v_i) = \rho(v_i)$.

We have therefore an interval representation ρ which represents the graph $G_1 \cup G_2$ because $\rho(v) \cap \rho(u) \Leftrightarrow \{v, u\} \in E_1 \vee \{v, u\} \in E_2$ for any $v, u \in V$. Notice that $G_1 \cup G_2$ agrees on $<_V$ because $\mathrm{right}_\rho(v_i) = i$. ◀

Using both Lemmas 6 and 7 we arrive at the following corollary.

▶ **Corollary 8.** *Let $\mathcal{G}$ be an order-preserving temporal interval graph of a* Temporal Δ Independent Set *instance. Then the conflict graph of $\mathcal{G}$ is an interval graph.*

Since Independent Set can be solved in linear time on interval graphs [16, 37], we immediately arrive at our main result of this subsection.

▶ **Theorem 9.** Temporal Δ Independent Set *on order-preserving temporal interval graphs is solvable in linear time.*

4.2 An FPT-Algorithm for Vertex Deletion to Order Preservation

Now we generalize Theorem 9 and show how to solve Temporal Δ Independent Set on *almost* order-preserving temporal unit interval graphs, that is, graphs that most of their vertices agree on a common ordering. To this end, we define a *distance* of a temporal graph to order preservation. This distance is measured by the size of the minimum vertex set that obstructs the compatibility of a total RE order of a temporal interval graph. We define it as follows and give an illustration in Figure 3.

▶ **Definition 10** (OPVD). *Let $\mathcal{G} = (V, \mathcal{E}, \tau)$ be a temporal interval graph. A* vertex deletion set for order preservation *(OPVD) is a set of vertices $V' \subseteq V$ such that $\mathcal{G} - V'$ is order-preserving.*

The size of the minimum OPVD set measures how many vertices obstruct a total RE order for a temporal interval graph. We denote the cardinality of the minimum OPVD by ℓ. A brute-force algorithm checks every subset of the vertex set to find a solution to TEMPORAL Δ INDEPENDENT SET. Given an ℓ-sized OPVD set we can brute-force the power set of the OPVD (which has size 2^ℓ) and then check against the rest of the order-preserving graph in polynomial time.

▶ **Theorem 11.** *TEMPORAL Δ INDEPENDENT SET can be decided in $2^\ell \cdot n^{\mathcal{O}(1)}$ time when given a size-ℓ OPVD set of the input temporal graph.*

Proof. The idea is as follows. Given an order-preserving vertex deletion set S of size ℓ, we brute-force its power set. Let G_{op} be the conflict graph of $\mathcal{G} - S$. The graph G_{op} is, by definition, an interval graph. For each subset X of S we compute G_X as $G_{\mathrm{op}} - N_{G_{\mathrm{op}}}(X)$, the neighbors of X from the conflict interval graph. As G_X is an interval graph, we can compute a maximum independent set V' of G_X in linear time, then check in quadratic time whether $X \cup V'$ is an independent set of size k in the conflict graph of $\mathcal{G}$. If $X \cup V'$ is an independent set of size at least k, then we have a yes-instance.

Any independent set of size k must clearly be divisible into two subsets, a subset of $X \subseteq S$ (that includes the trivial subset) and a subset of $V \setminus S$. Any independent set on $\mathcal{G}$ must be also an independent set on the subgraph induced by $V \setminus S$. If we exhaust all of the subsets of S and do not find an independent set of size at least $k - |X|$ on $\mathcal{G} - (S \cup N_{G_{\mathrm{op}}}[X])$ for $X \subseteq S$, then we can conclude that such set does not exist. In such case the instance is a no-instance. The power set of S is of size 2^ℓ, which means it takes $2^\ell \cdot n^{\mathcal{O}(1)}$ time to exhaust all subsets of S. ◀

Since the FPT algorithm for TEMPORAL Δ INDEPENDENT SET parameterized by the minimum OPVD ℓ behind Theorem 11 requires access to an ℓ-sized OPVD set, we present an FPT-algorithm to compute a minimum OPVD for a given temporal unit interval graph. We do this by providing a reduction to the so-called CONSECUTIVE ONES SUBMATRIX BY COLUMN DELETIONS problem, for which efficient algorithms are known [9, 35].

Before we describe the reduction, we give an alternative characterization of order-preserving temporal unit interval graphs. We will use this characterization in our FPT-algorithm to compute a minimum OPVD. As we show in the next lemma, a temporal unit interval graph $\mathcal{G}$ is order-preserving if and only if its vertices vs. maximal cliques matrix has the so-called *consecutive ones property*[1] (C1P). Note that it is known that the vertices vs. neighborhoods matrix also has the consecutive ones property in this case [14].

▶ **Lemma 12.** *A temporal unit interval graph is order-preserving if and only if its* vertices vs. maximal cliques matrix *has the consecutive ones property.*

Proof. Testing for the consecutive ones property for a matrix can be done in linear time [38]. To test a temporal unit interval graph for order preservation we compute its vertices vs. maximal cliques matrix and test it for the consecutive ones property. We say that $\mathcal{G}$'s set of maximal cliques $\mathcal{C}$ is the union of sets of maximal cliques of each layer of $\mathcal{G}$. The vertices vs. maximal cliques matrix M is a binary matrix in which $M_{i,j} = 1$ if and only if the vertex $v_i \in V$ is a member of $C_j \in \mathcal{C}$. It is left to show that a temporal unit interval graph $\mathcal{G}$ is order-preserving if and only if its vertices vs. maximal cliques matrix has the consecutive ones property.

---

[1] A 0-1-matrix has the *consecutive ones property* if there exists a permutation of the columns such that in each row all ones appear consecutively.

($\Rightarrow$) If $\mathcal{G}$ is order-preserving, then there exists an ordering $<_V$ such that every layer has an interval representation ρ in which the right endpoints of all intervals agree on $<_V$. Let M's columns be ordered by $<_V$. If M is not in its petrie form [2], then it must mean that there exists a clique C in $\mathcal{C}$ whose members are not consecutive in $<_V$. In other words, there exist $u, v, w \in V$ such that $u <_V w <_V v$, for which $u, v \in C$ and $w \notin C$. Since u and v are adjacent and $u <_V v$, we know that $\text{left}(v) < \text{right}(u)$. We know also that the length of $\rho(v)$ is exactly 1. This definitely means that v and w intersect because $\text{right}(w) \in [\text{right}(u), \text{right}(v)]$. However since $w \notin C$, w and u cannot be adjacent. This is a contradiction since $\text{right}(v) - \text{right}(u) < 1$ and $\text{right}(w) - \text{left}(w) = 1$. If $\mathcal{G}$ is order-preserving, then M must have the consecutive ones property.

($\Leftarrow$) If M has the consecutive ones property, then there exists an ordering $<_V$ so that the vertices vs. maximal cliques matrix M_t of every layer $G_t \in \mathcal{G}$ is in its petrie form, when its columns are permuted according to $<_V$. Let $\mathcal{I}_t(v)$ be the union of all maximal cliques in layer G_t which contain v. We know that for every $v \in V$ the vertices of $\mathcal{I}_t(v)$ are consecutive in $<_V$ [5]. Let $\text{index}(v)$ be the index of v in $<_V$ and let $\rho_{G_t}(v) = [\min\{\text{index}(u) \mid u \in \mathcal{I}_t(v)\} - 1 + \text{index}(v) \cdot \varepsilon, \text{index}(v)]$, for some $0 < \varepsilon < 1/|V|$. First, note that no two intervals are contained in each other. This means that there is an equivalent interval representation where all intervals have unit length [36].

By showing $\{u, v\} \in E_t \Leftrightarrow \rho_t(v) \cap \rho_t(u) \neq \varnothing$ we effectively show that ρ_{G_t} is an interval representation of G_t. If $\{u, v\} \in E_t$, then there must exist a maximal clique C so that $u, v \in C$ and thus $u \in \mathcal{I}_t(v)$ and $v \in \mathcal{I}_t(u)$. Assume $u <_V v$, then $\text{left}_{\rho_{G_t}}(v) \leq \text{right}_{\rho_{G_t}}(u)$ and by that $\rho_{G_t}(u) \cap \rho_{G_t}(v) \neq \varnothing$. If $\{u, v\} \notin E_t$ then there is no clique C so that $u, v \in C$. Assume that $u <_V v$, then $\rho_{G_t}(u) \cap \rho_{G_t}(v) \neq \varnothing$ if and only if $\text{left}_{\rho_{G_t}}(v) \leq \text{right}_{\rho_{G_t}}(u)$. This cannot be because $\text{left}_{\rho_{G_t}}(v)$ is exactly the right endpoint of v's lowest neighbors in $<_V$. If $\text{right}_{\rho_{G_t}}(u)$ is right of v's lowest neighbor's right endpoint, then the vertices of $\mathcal{I}_t(v)$ are not consecutive in $<_V$. This contradicts the fact that M has the consecutive ones property. ◄

Since there is a bijection between the columns of the vertices vs. maximal cliques matrix M of $\mathcal{G}$ and $\mathcal{G}$'s vertices, we can use M as input for the CONSECUTIVE ONES SUBMATRIX BY COLUMN DELETIONS problem and apply existing algorithms for that problem [9, 35] to obtain a vertex-maximal temporal subgraph of $\mathcal{G}$ that is order-preserving. This allows us to obtain the following result.

▶ **Theorem 13.** *A minimum OPVD for a given temporal unit interval graph can be computed in* $10^\ell n^{\mathcal{O}(1)}$ *time, where* ℓ *is the size of a minimum OPVD.*

Proof. In Lemma 12 we have shown that a temporal unit interval graph is order-preserving if and only if its vertices vs. maximal cliques matrix has the consecutive ones property. We provide a reduction to the CONSECUTIVE ONES SUBMATRIX BY COLUMN DELETIONS problem. Formally, in CONSECUTIVE ONES SUBMATRIX BY COLUMN DELETIONS we are given a binary matrix $M \in \{0, 1\}^{m \times n}$ and are asked whether there exists a submatrix M' with the consecutive ones property, such that M' is obtained with not more than ℓ column deletions from M. CONSECUTIVE ONES SUBMATRIX BY COLUMN DELETIONS is known to be FPT with respect to the column deletion set size and it can be decided in $10^\ell n^{\mathcal{O}(1)}$ time [9, 35].

Let $\mathcal{G} = (V, \mathcal{E})$ be a temporal unit interval graph with $\mathcal{C}$ as maximal cliques set. Let M be the vertices vs. maximal cliques matrix of $\mathcal{G}$. If M does not have the consecutive ones property, then we can find a set of ℓ columns in $10^\ell n^{\mathcal{O}(1)}$ time so that when deleted from M,

---

[2] A 0-1-matrix is in its *petrie form* (if it has one) if the columns are permuted in a way such that the ones appear consecutively in all rows.

the resulting matrix M' has the consecutive ones property. The columns of M are mapped to vertices of V, the image of the deleted columns V' is the OPVD set. We can find in linear time an ordering $<'_V$ of M' columns such that M' is in its petrie form. All layers of the graph $\mathcal{G} - V'$ agree on $<'_V$. ◀

This provides us with an efficient algorithm for TEMPORAL Δ INDEPENDENT SET on "almost" ordered temporal unit interval graph. Namely, find in $10^\ell n^{\mathcal{O}(1)}$ time a minimum OPVD set in the input temporal unit interval graph using Theorem 13, then decide in $2^\ell n^{\mathcal{O}(1)}$ time if we have a yes-instance of TEMPORAL Δ INDEPENDENT SET using Theorem 11. Overall, we arrive at the following result.

▶ **Corollary 14.** *TEMPORAL Δ INDEPENDENT SET can be decided in $10^\ell \cdot n^{\mathcal{O}(1)}$ time if the input temporal graph is a temporal unit interval graph, where ℓ is the size of a minimum OPVD of the input temporal graph.*

4.3 NP-Hardness of Vertex Deletion to Order Preservation

Finally, we show that computing a minimum OPVD for a given temporal unit interval graph is NP-hard. This complements Theorem 13 as it implies that we presumably cannot improve Theorem 13 to a polynomial-time algorithm.

▶ **Theorem 15.** *Computing a minimum OPVD for a given temporal unit interval graph is NP-hard.*

Proof. To show NP-hardness, we present a polynomial time many-one reduction from the NP-complete CONSECUTIVE ONES SUBMATRIX BY COLUMN DELETIONS problem [20] to the problem of computing an OPVD of size at most ℓ for a given temporal unit interval graph. Note that this implies NP-hardness of the optimization problem of finding a minimum OPVD.

Formally, in CONSECUTIVE ONES SUBMATRIX BY COLUMN DELETIONS we are given a binary matrix $M \in \{0, 1\}^{m \times n}$ and are asked whether there exists a submatrix M' with the consecutive ones property, such that M' is obtained with not more than ℓ column deletions from M. Note that we can assume w.l.o.g. that there are at least two ones in each row of M, otherwise we can delete the row since its ones are consecutive for all permutations of the columns.

Our reduction works as follows. Given a binary matrix $M \in \{0, 1\}^{m \times n}$ with m rows and n columns, we create a temporal graph $\mathcal{G}$ with n vertices $V = \{1, \ldots, n\}$, one for each column, and m layers, one for each row. In each layer G_t for $1 \le t \le m$, we add an edge between vertices i and j if $M_{t,i} = 1$ and $M_{t,j} = 1$. This finished the construction of $\mathcal{G}$, which can clearly be done in polynomial time.

Next, we argue that $\mathcal{G}$ is a temporal unit interval graph. To this end, note that every layer G_t of $\mathcal{G}$ is a single clique (consisting of vertices i with $M_{t,i} = 1$) and some isolated vertices (the vertices i with $M_{t,i} = 0$). Hence, we can clearly find a unit interval representation for every layer G_t of $\mathcal{G}$.

To prove the correctness of the reduction, we first observe that M is the vertices vs. maximal cliques matrix of $\mathcal{G}$: there is exactly one non-trivial maximal clique in each layer G_t containing the vertices i with $M_{t,i} = 1$. We show ℓ columns can be deleted from M such that the remaining matrix M' has the consecutive ones property if and only if $\mathcal{G}$ admits an OPVD of size ℓ.

($\Rightarrow$) Assume there are ℓ columns that can be deleted from M such that the remaining matrix M' has the consecutive ones property. Then M' corresponds to vertices vs. maximal cliques matrix of $\mathcal{G}'$ which is obtained from $\mathcal{G}$ by removing the ℓ vertices corresponding to the deleted columns of M. By Lemma 12 we have that $\mathcal{G}'$ is an order-preserving temporal unit interval graph. It follows that the removed vertices form an OPVD of size ℓ for $\mathcal{G}$.

($\Leftarrow$) Assume $\mathcal{G}$ admits an OPVD X of size ℓ. Then let M' be the matrix obtained from M by deleting the ℓ columns corresponding to the vertices in X. Now we have that M' is the vertices vs. maximal cliques matrix of $\mathcal{G} - X$, which is an order-preserving temporal unit interval graph. By Lemma 12 we have that M' has the consecutive ones property and hence that (M, ℓ) is a yes-instance of Consecutive Ones Submatrix by Column Deletions. ◀

5   Conclusion

We introduced a naturally motivated temporal version of the classic Independent Set problem, called Temporal Δ Independent Set, and investigated its computational complexity. Herein, we focused on the case where all layers of the input temporal graph are unit interval graphs. After establishing computational hardness results, we showed that Maximum Temporal Δ Independent Set admits a polynomial-time $(\tau - \Delta + 1) \cdot 2^{\Delta}$-approximation. Furthermore, we presented a polynomial-time algorithm for Temporal Δ Independent Set when restricted to so-called order-preserving temporal interval graphs and generalized it to an FPT-algorithm for the vertex deletion distance to order preservation. The latter heavily relies on our result that order preservation is retained under edge-union and edge-intersection, which is of independent interest since it may also be useful in the context of related problem such as Temporal Δ Clique [4, 23, 39].

An immediate future work direction is to generalize our results for temporal (non-unit) interval graphs. For most of our results this question remains open. We believe that our approximation algorithm does not easily adapt. In fact even for two layers it is unclear how to approximate Maximum Temporal Δ Independent Set. Our FPT-algorithm for Temporal Δ Independent Set parameterized by the vertex deletion distance to order preservation generalizes to the non-unit interval case assuming the deletion set is part of the input. We leave for future research how to efficiently compute a minimum vertex deletion set to order preservation for temporal interval graphs.

References

1 Karhan Akcoglu, James Aspnes, Bhaskar DasGupta, and Ming-Yang Kao. Opportunity cost algorithms for combinatorial auctions. In *Computational Methods in Decision-Making, Economics and Finance*, pages 455–479. Springer, 2002.

2 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107:108–123, 2020.

3 Reuven Bar-Yehuda, Magnús M. Halldórsson, Joseph Naor, Hadas Shachnai, and Irina Shapira. Scheduling split intervals. *SIAM Journal on Computing*, 36(1):1–15, 2006.

4 Matthias Bentert, Anne-Sophie Himmel, Hendrik Molter, Marco Morik, Rolf Niedermeier, and René Saitenmacher. Listing all maximal k-plexes in temporal graphs. *Journal of Experimental Algorithmics (JEA)*, 24:1–27, 2019.

5 Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976.

6 Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

7 Timothy M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46(2):178–189, 2003.

8 Miroslav Chlebík and Janka Chlebíková. Approximation hardness of optimization problems in intersection graphs of *d*-dimensional boxes. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05)*, pages 267–276. SIAM, 2005.

9 Michael Dom, Jiong Guo, and Rolf Niedermeier. Approximation and fixed-parameter algorithms for consecutive ones submatrix problems. *Journal of Computer and System Sciences*, 76(3-4):204–221, 2010.

10 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

11 Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34(6):1302–1323, 2005.

12 Michael R. Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.

13 Paola Flocchini, Bernard Mans, and Nicola Santoro. On the exploration of time-varying networks. *Theoretical Computer Science*, 469:53–68, 2013.

14 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020.

15 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

16 Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974.

17 Paul C. Gilmore and Alan J. Hoffman. A characterization of comparability graphs and of interval graphs. *Canadian Journal of Mathematics*, 16:539–548, 1964.

18 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *Proceedings of the 1st International Workshop on Parameterized and Exact Computation (IWPEC '04)*, pages 162–173. Springer, 2004.

19 András Gyárfás and Douglas West. Multitrack interval graphs. *Congressus Numerantium 109*, 1995.

20 Mohammad Taghi Hajiaghayi and Yashar Ganjali. A note on the consecutive ones submatrix problem. *Information processing letters*, 83(3):163–166, 2002.

21 György Hajós. Über eine Art von Graphen. *Internationale Mathematische Nachrichten*, 11(65), 1957.

22 Monika Henzinger, Stefan Neumann, and Andreas Wiese. Dynamic Approximate Maximum Independent Set of Intervals, Hypercubes and Hyperrectangles. In *Proceedings of the 36th International Symposium on Computational Geometry (SoCG '20)*, volume 164 of *LIPIcs*, pages 51:1–51:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

23 Anne-Sophie Himmel, Hendrik Molter, Rolf Niedermeier, and Manuel Sorge. Adapting the Bron–Kerbosch algorithm for enumerating maximal cliques in temporal graphs. *Social Network Analysis and Mining*, 7(1):35:1–35:16, 2017.

24 Wen-Lian Hsu and Jeremy P. Spinrad. Independent sets in circular-arc graphs. *Journal of Algorithms*, 19(2):145–160, 1995.

25 Minghui Jiang. On the parameterized complexity of some optimization problems related to multiple-interval graphs. *Theoretical Computer Science*, 411(49):4253–4262, 2010.

26 Deborah Joseph, Joao Meidanis, and Prasoon Tiwari. Determining DNA sequence similarity using maximum independent set algorithms for interval graphs. In *Proceedings of the 3rd Scandinavian Workshop on Algorithm Theory (SWAT '92)*, pages 326–337. Springer, 1992.

27 Richard M Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

28 Sanjeev Khanna, Shan Muthukrishnan, and Mike Paterson. On approximating rectangle tiling and packing. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 95, page 384. SIAM, 1998.

29 Vassilis Kostakos. Temporal graphs. *Physica A: Statistical Mechanics and its Applications*, 388(6):1007–1023, 2009.

30 Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):61:1–61:29, 2018.

31 Dániel Marx. Efficient approximation schemes for geometric problems? In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA '05)*, pages 448–459. Springer, 2005.

32 George B. Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. Computing maximum matchings in temporal graphs. In *Proceedings of the 37th International Symposium on Theoretical Aspects of Computer Science (STACS '20)*, volume 154 of *LIPIcs*, pages 27:1–27:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

33 George B. Mertzios, Hendrik Molter, and Viktor Zamaraev. Sliding window temporal graph coloring. *Journal of Computer and System Sciences*, 120:97–115, 2021.

34 Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4):239–280, 2016.

35 N.S. Narayanaswamy and R. Subashini. Obtaining matrices with the consecutive ones property by row deletions. *Algorithmica*, 71(3):758–773, 2015.

36 Fred S. Roberts. Indifference graphs. Proof techniques in graph theory. In *Proceedings of the Second Ann Arbor Graph Conference, Academic Press, New York*, 1969.

37 Donald J. Rose, Robert Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

38 Alan Tucker. A structure theorem for the consecutive 1's property. *Journal of Combinatorial Theory, Series B*, 12(2):153–162, 1972.

39 Tiphaine Viard, Matthieu Latapy, and Clémence Magnien. Computing maximal cliques in link streams. *Theoretical Computer Science*, 609:245–252, 2016.