New Results for the 2-Interval Pattern Problem

Guillaume Blin, Guillaume Fertin, Stéphane Vialette

New results for the 2-interval pattern problem

Guillaume Blin[1], Guillaume Fertin[1], and Stéphane Vialette[2]

[1] LINA, FRE CNRS 2729
Université de Nantes, 2 rue de la Houssinière
BP 92208 44322 Nantes Cedex 3 - FRANCE
{blin,fertin}@lina.univ-nantes.fr
[2] LRI, UMR CNRS 8623
Faculté des Sciences d'Orsay, Université Paris-Sud
Bât 490, 91405 Orsay Cedex - FRANCE
vialette@lri.fr

Abstract. We present new results concerning the problem of finding a constrained pattern in a set of 2-intervals. Given a set of n 2-intervals $\mathcal{D}$ and a model R describing if two disjoint 2-intervals can be in precedence order ($<$), be allowed to nest ($\sqsubset$) and/or be allowed to cross ($\between$), the problem asks to find a maximum cardinality subset $\mathcal{D}' \subseteq \mathcal{D}$ such that any two 2-intervals in $\mathcal{D}'$ agree with R. We improve the time complexity of the best known algorithm for $R = \{\sqsubset\}$ by giving an optimal $O(n \log n)$ time algorithm. Also, we give a graph-like relaxation for $R = \{\sqsubset, \between\}$ that is solvable in $O(n^2\sqrt{n})$ time. Finally, we prove that the problem is **NP**-complete for $R = \{<, \between\}$, and in addition to that, we give a fixed-parameter tractability result based on the crossing structure of $\mathcal{D}$.

1 Introduction

The general problem of establishing a general representation of structured patterns, *i.e.*, *macroscopic describers* of RNA secondary structures, was considered in [Via02,Via04]. The approach was to set up a *geometric* description of helices by means of a natural generalization of intervals, namely a 2-*interval*. A 2-interval is the disjoint union of two intervals on the line. The geometric properties of 2-intervals provide a possible guide for understanding the computational complexity of finding structured patterns in RNA sequences. Using a model to represent non sequential information allows us for varying restrictions on the complexity of the pattern structure. Indeed, two disjoint 2-intervals, *i.e.*, two 2-intervals that do not intersect in any point, can be in precedence order ($<$), be allowed to nest ($\sqsubset$) and/or be allowed to cross ($\between$). Furthermore, the set of 2-intervals and the pattern can have different restrictions. These different combinations of restrictions alter the computational complexity of the problems, and need to be examined separately. This examination produces efficient algorithms for more restrictive structured patterns, and hardness results for those less restrictive.

There are basically two lines of research our results refer to. The first one is that of arc annotated sequences and the other one is that of protein topolo-

gies. In the context of arc annotated sequences, the Arc-Preserving Subsequence (APS) and Longest Arc-Preserving Common Subsequence (LAPCS) problems are useful in representing the structural information of RNA and protein sequences [Eva99,JLMZ00,GGN02,AGGN02]. The basic idea is to provide a measure for similarity, not only on the sequence level, but also on the structural level. Moreover, a similar problem to compare the three-dimensional structure of proteins is the Contact Map Overlap problem described by Goldman *et al* [GIP99]. Viksna and Gilbert described algorithms for pattern matching and pattern learning in TOPS diagram (formal description of protein topologies) [VD01].

Our results are also related to the independent set problem in different extensions of 2-interval graphs. A graph G is a t-interval graph if there is an intersection model whose objects consist of collections of t intervals, $t \geq 1$, such that G is the intersection graph of this model [TH79,GW79]. From this definition, it is clear that every interval graph is a 1-interval graph. Of particular interest is the class of 2-interval graphs. For example, line graphs, trees and circular-arc graphs are 2-interval graphs. However, West and Shmoys [WS84] have shown that the recognition problem for t-interval graphs is **NP**-complete for every $t \geq 2$ (this has to be compared with linear time recognition of 1-interval graphs). In the context of sequence similarity, [JMT92] contains an application of graphs having interval number at most two. In [BYHN$^+$02], the authors considered the problem of scheduling jobs that are given as groups of non-intersecting segments on the real line. Of particular importance, they showed that the maximum weighted independent set for t-interval graphs ($t \geq 2$) is **APX**-hard even for highly restricted instances Also, they gave a $2t$-approximation algorithm for general instances based on a fractional version of the Local Ratio Technique.

The problem of finding the longest 2-interval pattern in a set of 2-intervals $\mathcal{D}$ with respect to a given abstract model, the so-called 2-Interval Pattern problem, has been introduced by Vialette [Via02,Via04]. Vialette divides the problem in different classes based on the structure of the model and gives for most of them either **NP**-completeness results or polynomial time algorithms. In the present paper, we focus on three classes: the model $\{\sqsubset\}$ over an unlimited support, the model $\{\sqsubset, \between\}$ over a disjoint support and the model $\{<, \between\}$ over a unitary support. We give precise results for these three classes. Those three classes are of importance since each one is a straightforward extension of the Pattern Matching Over Set Of 2-Intervals problem introduced in [Via04] and hence is strongly related, in the context of molecular biology, to pattern matching over RNA secondary structures. The results given in the present paper almost complete the table proposed by Vialette [Via04] (see Table 1) and provide an important step towards a better understanding of the precise complexity of 2-interval pattern matching problems.

The remainder of the paper is organized as follows. In Section 2 we briefly review the terminology introduced in [Via04]. In Section 3, we improve the time complexity of the best known algorithm for model $R = \{\sqsubset\}$ over an unlimited support. In Section 4, we give a graph-like relaxation for model $\{\sqsubset, \between\}$ that is

solvable in polynomial time. In Section 5, we prove that the 2-interval pattern problem for model $R = \{<, \between\}$ is **NP**-complete even when restricted to unitary support thereby answering an open problem posed in [Via04]. In addition to that latter result, we give in Section 6 a fixed-parameter tractability result based on the crossing structure of $\mathcal{D}$.

2 Preliminaries

An interval and a 2-interval represent respectively a sequence of contiguous bases and pairings between two intervals, *i.e.*, *stems*, in RNA secondary structures. Thus, 2-intervals can be seen as *macroscopic describers* of RNA structures.

Formally, a 2-*interval* is the disjoint union of two intervals on a line. We denote it by $D = (I_1, J_1)$ where I_1 and J_1 are intervals such that $I_1 < J_1$ (here $<$ is the strict precedence order between intervals); in that case we write also $\mathsf{Left}(D) = I_1$ and $\mathsf{Right}(D) = J_1$. If $[x : y]$ and $[x' : y']$ are two intervals such that $[x : y] < [x' : y']$, we will sometimes write $D = ([x : y], [x' : y'])$ to emphasize on the precise definition of the 2-interval D. Let $D_1 = (I_1, J_1)$ and $D_2 = (I_2, J_2)$ be two 2-intervals. They are called *disjoint* if $(I_1 \cup J_1) \cap (I_2 \cup J_2) = \emptyset$ (*i.e.*, involved intervals do not intersect). The *covering interval* of a 2-interval D, written $\mathsf{Cover}(D)$, is the least interval covering both $\mathsf{Left}(D)$ and $\mathsf{Right}(D)$.

Of particular interest is the relation between two disjoint 2-intervals $D_1 = (I_1, J_1)$ and $D_2 = (I_2, J_2)$. We will write $D_1 < D_2$ if $I_1 < J_1 < I_2 < J_2$, $D_1 \sqsubset D_2$ if $I_2 < I_1 < J_1 < J_2$ and $D_1 \between D_2$ if $I_1 < I_2 < J_1 < J_2$. Two 2-intervals D_1 and D_2 are τ-comparable for some $\tau \in \{<, \sqsubset, \between\}$ if $D_1 \tau D_2$ or $D_2 \tau D_1$. Let $\mathcal{D}$ be a set of 2-intervals and $R \subseteq \{<, \sqsubset, \between\}$ be non-empty. The set $\mathcal{D}$ is *R-comparable* if any two distinct 2-intervals of $\mathcal{D}$ are τ-comparable for some $\tau \in R$. Throughout the paper, the non-empty subset R is called a *model*. Clearly, if a set of 2-intervals $\mathcal{D}$ is R-comparable then $\mathcal{D}$ is a set of disjoint 2-intervals. The *support* of a set of 2-intervals $\mathcal{D}$, written $\mathsf{Support}(\mathcal{D})$, is the set of all *simple* intervals involved in $\mathcal{D}$, *i.e.*, $\mathsf{Support}(\mathcal{D}) = \bigcup_{D \in \mathcal{D}} (\mathsf{Left}(D) \cup \mathsf{Right}(D))$. The *leftmost* (resp. *rightmost*) element of a set of disjoint 2-intervals $\mathcal{D}$ is the 2-interval $D_i \in \mathcal{D}$ such that $\mathsf{Left}(D_i) < \mathsf{Left}(D_j)$ (resp. $\mathsf{Right}(D_j) < \mathsf{Right}(D_i)$) for all $D_j \in \mathcal{D} - D_i$. Observe that it could be the case that D_i is both the leftmost and rightmost element of $\mathcal{D}$ (this is indeed the case if $|\mathcal{D}| = 1$ or if $D_j \sqsubset D_i$ for all $D_j \in \mathcal{D} - D_i$). Some parameters can be defined. The *width* of $\mathcal{D}$, written $\mathsf{Width}(\mathcal{D})$, is the size of a maximum cardinality $\{<\}$-comparable subset of $\mathcal{D}$, the *height* of $\mathcal{D}$, written $\mathsf{Height}(\mathcal{D})$, is the size of a maximum cardinality $\{\sqsubset\}$-comparable subset of $\mathcal{D}$ and the *depth* of $\mathcal{D}$, written $\mathsf{Depth}(\mathcal{D})$, is the size of a maximum cardinality $\{\between\}$-comparable subset of $\mathcal{D}$. Observe that these three parameters can be computed in polynomial time [Via04]. Finally, the *forward crossing number* of $\mathcal{D}$, written $\mathsf{FCrossing}(\mathcal{D})$, is defined by $\mathsf{FCrossing}(\mathcal{D}) = \max_{D_i \in \mathcal{D}} |\{D_j : D_i \between D_j\}|$. Clearly, $\mathsf{Depth}(\mathcal{D}) \leq \mathsf{FCrossing}(\mathcal{D})$.

In [Via04], Vialette proposed two restrictions on the support:

1. all the intervals of the support are of the same size;

2. all the intervals of the support are disjoint, *i.e.*, if two intervals $I, I' \in \mathsf{Support}(\mathcal{D})$ overlap then $I = I'$.

Using restrictions on the support allows us for varying restrictions on the complexity of the 2-interval set structure, and hence on the complexity of the problems. These two restrictions involve three levels of complexity:

- UNLIMITED: no restrictions
- UNITARY: restriction 1
- DISJOINT: restrictions 1 and 2

Given a set of 2-intervals $\mathcal{D}$, a model $R \subseteq \{<, \sqsubset, \between\}$ and a positive integer k, the 2-INTERVAL PATTERN problem consists in finding a subset $\mathcal{D}' \subseteq \mathcal{D}$ of cardinality greater than or equal to k such that $\mathcal{D}'$ is R-comparable. For the sake of brevity, the 2-INTERVAL PATTERN problem with respect to a model R over an unlimited (resp. unitary, disjoint) support is abbreviated in 2-IP-UNL-R (resp. 2-IP-UNI-R, 2-IP-DIS-R).

Vialette proved in [Via04] that 2-IP-UNI-$\{<, \sqsubset, \between\}$ and 2-IP-UNI-$\{\sqsubset, \between\}$ are **NP**-complete. Moreover, he gave polynomial algorithms for the problem with respect to the models $\{<\}$, $\{\sqsubset\}$, $\{\between\}$ and $\{<, \sqsubset\}$ (cf. Table 1).

In this article, we answer three open problems and we improve the complexity of another one as shown in Table 1. Moreover, we show that 2-IP-UNI-$\{<, \between\}$ is fixed parameter tractable when parameterized by the forward crossing number of $\mathcal{D}$.

2-INTERVAL PATTERN PROBLEM			
	SUPPORT		
MODEL	UNLIMITED	UNITARY	DISJOINT
$\{<, \sqsubset, \between\}$	**NP**-complete		$O(n\sqrt{n})$[MV80]
$\{\sqsubset, \between\}$	**NP**-complete		$O(n^2\sqrt{n})$ $\star$
$\{<, \sqsubset\}$	$O(n^2)$		
$\{<, \between\}$	**NP**-complete $\star$		?
$\{<\}$	$O(n \log n)$		
$\{\sqsubset\}$	$O(n \log n)$ $\star$ $\bullet$		
$\{\between\}$	$O(n^2 \log n)$		

Table 1. 2-INTERVAL PATTERN problem complexity where $n = |\mathcal{D}|$. When not specified, the complexity comes from [Via04]. $\star$ contributions of the present paper. $\bullet$ improvement of the existing complexity (which was $O(n^2)$ in [Via04]).

3 Improving the Complexity of 2-IP-UNL-$\{\sqsubset\}$

The problem of finding the largest $\{\sqsubset\}$-comparable subset in a set of 2-intervals was considered in [Via04]. Observing that this problem is equivalent to finding a largest clique in a comparability graph (a linear time solvable problem [Gol80]), an $O(n^2)$ time algorithm was thus proposed. We improve that result by giving an optimal $O(n \log n)$ time algorithm for finding a largest $\{\sqsubset\}$-comparable subset in a set of 2-intervals.

The inefficiency of the algorithm proposed in [Via04] lies in the effective construction of a comparability graph. We show that this construction can be avoided by considering trapezoids in place of 2-intervals. Recall that a *trapezoid graph* is the intersection graph of a finite set of trapezoids between two parallel lines [DGP88] (it is easily seen that trapezoid graphs generalize both interval graphs and permutation graphs). Analogously to 2-intervals, we will denote by $T = ([x : y], [x' : y'])$ the trapezoid with upper interval $[x : y]$ and lower interval $[x' : y']$.

Proposition 1. 2-IP-UNL-$\{\sqsubset\}$ *is solvable in* $O(n \log n)$ *time.*

Proof. Let $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ be a collection of 2-intervals of the real line. Construct a collection of trapezoids $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ between two parallel lines as follows. For each 2-interval $D_i = ([x : y], [x' : y']) \in \mathcal{D}$, we add the trapezoid $T_i = ([x : y], [-y' : -x'])$ to $\mathcal{T}$.

Claim 1. For all $1 \leq i \leq j \leq n$, the 2-intervals D_i and D_j are $\{\sqsubset\}$-comparable if and only if the trapezoids T_i and T_j are non-intersecting.

Proof (of Claim 1). Let $D_i = ([x_i : y_i], [x'_i : y'_i])$ and $D_j = ([x_j : y_j], [x'_j : y'_j])$ be two 2-intervals of $\mathcal{D}$ and $T_i = ([x_i : y_i], [-y'_i : -x'_i])$ and $T_j = ([x_j : y_j], [-y'_j : -x'_j])$ be the two corresponding trapezoids in $\mathcal{T}$. Suppose that D_i and D_j are $\{\sqsubset\}$-comparable. Without loss of generality, we may assume $D_j \sqsubset D_i$. Thus, we have $y_i < x_j$ and $y'_j < x'_i$. It follows immediately that $-x'_i < -y'_j$, and hence the two trapezoids T_i and T_j are non-intersecting. The proof of the converse is identical. □

Clearly, the collection $\mathcal{T}$ can be constructed in $O(n)$ time. Based on a geometric representation of trapezoid graphs by boxes in the plane, Felsner *et al.* [FMW97] have designed a $O(n \log n)$ algorithm for finding a maximum cardinality subcollection of non-intersecting trapezoids in a collection of trapezoids, and the proposition follows. □

Based on Fredman's bound for the number of comparisons needed to compute maximum increasing subsequences in permutation [Fre75], Felsner *et al.* [FMW97] argued that their $O(n \log n)$ time algorithm for finding a maximum cardinality subcollection of non-intersecting trapezoids in a collection of trapezoids is optimal. Then it follows from Proposition 1 that our $O(n \log n)$ time algorithm for finding a maximum cardinality $\{\sqsubset\}$-comparable subset in a set of 2-intervals is optimal as-well.

4 A Polynomial Time Algorithm for 2-IP-DIS-$\{\sqsubset, \between\}$

In this section, we give a $O(n^2\sqrt{n})$ time algorithm for the 2-IP-DIS-$\{\sqsubset, \between\}$ problem, where n is the cardinality of the set of 2-intervals $\mathcal{D}$. Recall that given a set of 2-intervals $\mathcal{D}$ over a disjoint support, the problem asks to find the size of a maximum cardinality $\{\sqsubset, \between\}$-comparable subset $\mathcal{D}' \subseteq \mathcal{D}$. Observe that the

2-IP-Dis-$\{\sqsubset, \between\}$ problem has an interesting formulation in terms of constrained matchings in general graphs: Given a graph G together with a linear ordering π of its vertices, the 2-IP-Dis-$\{\sqsubset, \between\}$ problem is equivalent to finding a maximum cardinality matching $\mathcal{M}$ in G with the property that for any two distinct edges $\{u, v\}$ and $\{u', v'\}$ of $\mathcal{M}$ neither $\max\{\pi(u), \pi(v)\} < \min\{\pi(u'), \pi(v')\}$ nor $\max\{\pi(u'), \pi(v')\} < \min\{\pi(u), \pi(v)\}$ occur.

Roughly speaking, our algorithm is based on a three-step procedure. First, the interval graph of all the covering intervals of 2-intervals in $\mathcal{D}$ is constructed. Next, all the maximal cliques of that graph are efficiently computed. Finally, for each maximal clique we construct a new graph and find a solution using a maximum cardinality matching algorithm. The size of a best solution found in the third step is thus returned. Clearly, the efficiency of our algorithm relies upon an efficient algorithm for finding all the maximal cliques in the intersection of the covering intervals. We now proceed with the details of our algorithm.

Let $\mathcal{D} = \{D_i : 1 \leq i \leq n\}$ be a set of 2-intervals. Consider the set $\mathcal{C}_\mathcal{D}$ composed of all the covering intervals of the 2-intervals in $\mathcal{D}$, *i.e.*, $\mathcal{C}_\mathcal{D} = \{\mathsf{Cover}(D) : D \in \mathcal{D}\}$. Now, let $\Omega(\mathcal{C}_\mathcal{D})$ be the interval graph associated with $\mathcal{C}_\mathcal{D}$. The graph $\Omega(\mathcal{C}_\mathcal{D})$ has a vertex v_i for each interval $\mathsf{Cover}(D_i)$ in $\mathcal{C}_\mathcal{D}$ and two vertices v_i and v_j of $\Omega(\mathcal{C}_\mathcal{D})$ are joined by an edge if the two associated intervals $\mathsf{Cover}(D_i)$ and $\mathsf{Cover}(D_j)$ intersect. Most in the interest in the interval graph $\Omega(\mathcal{C}_\mathcal{D})$ stems from the following lemma.

Lemma 1. *Let $\mathcal{D}$ be a set of 2-intervals and $\mathcal{D}'$ be a $\{\sqsubset, \between\}$-comparable subset of $\mathcal{D}$. Then, $\{v_i : D_i \in \mathcal{D}'\}$ induces a complete graph in $\Omega(\mathcal{C}_\mathcal{D})$.*

Proof. Let D_i and D_j be two distinct 2-intervals of $\mathcal{D}'$. Since D_i and D_j are $\{\sqsubset, \between\}$-comparable then it follows that either intervals $\mathsf{Cover}(D_i)$ and $\mathsf{Cover}(D_j)$ overlap or one interval is included in the other. In both cases, intervals $\mathsf{Cover}(D_i)$ and $\mathsf{Cover}(D_j)$ intersect and hence vertices v_i and v_j are joined by an edge in $\Omega(\mathcal{C}_\mathcal{D})$. Therefore $\{v_i : D_i \in \mathcal{D}'\}$ induces a complete graph in $\Omega(\mathcal{C}_\mathcal{D})$. □

Observe that the converse is false since the intersection of two 2-intervals in $\mathcal{D}$ results in an edge in $\Omega(\mathcal{C}_\mathcal{D})$, and hence two 2-intervals associated to two distinct vertices in the maximal clique C may not be $\{\sqsubset, \between\}$-comparable. However, thanks to Lemma 1 we now only need to focus on maximal cliques of $\Omega(\mathcal{C}_\mathcal{D})$. Several problems that are **NP**-complete on general graphs have polynomial time algorithms for interval graphs. The problem of finding all the maximal cliques of a graph is one such example. Indeed, an interval graph $G = (V, E)$ is a chordal graph and as such has at most $|V|$ maximal cliques [FG65]. Furthermore, all the maximal cliques of a chordal graph can be found in $O(n + m)$ time, where $n = |V|$ and $m = |E|$, by a modification of Maximum Cardinality Search (MCS) [TY84,BP93].

Let C be a maximal clique of $\Omega(\mathcal{C}_\mathcal{D})$. As observed above, any two 2-intervals associated to two distinct vertices in the maximal clique C may not be $\{\sqsubset, \between\}$-comparable. Let $\mathcal{D}' \subseteq \mathcal{D}$ be the set of all 2-intervals associated to vertices in the maximal clique C. Based on C, consider the graph $G_C = (V_C, E_C)$ defined by $V_C = \mathsf{Support}(\mathcal{D}')$ and $E_C = \{\{I, J\} : D = (I, J) \in \mathcal{D}'\}$. In other words, the

set of vertices of G_C is the support of $\mathcal{D}'$ and the edges of G_C is the 2-interval subset $\mathcal{D}'$ itself viewed as a set of subsets of size 2. Note that the construction of G_C is possible only because $\mathcal{D}'$ has disjoint support. The following lemma is an immediate consequence of the definition of G_C and Lemma 1.

Lemma 2. *Let C be a clique in $\Omega(\mathcal{C}_\mathcal{D})$ and $G_C = (V_C, E_C)$ be the graph constructed as detailed above. Then, $\{(I_{i_1}, J_{i_1}), (I_{i_2}, J_{i_2}), \ldots, (I_{i_k}, J_{i_k})\}$ is a $\{\sqsubset, \emptyset\}$-comparable subset if and only if $\{\{I_{i_1}, J_{i_1}\}, \{I_{i_2}, J_{i_2}\}, \ldots, \{I_{i_k}, J_{i_k}\}\}$ is a matching in G_C.*

Proposition 2. *The* 2-IP-DIS-$\{\sqsubset, \emptyset\}$ *problem is solvable in $O(n^2\sqrt{n})$ time, where n is the number of 2-intervals in $\mathcal{D}$.*

Proof. Consider the algorithm given in Figure 1. Correctness of this algorithm follows from Lemmas 1 and 2. What is left is to prove the time complexity. Clearly, the interval graph $\Omega(\mathcal{C}_\mathcal{D})$ can be constructed in $O(n^2)$ time. All the maximal cliques of $\Omega(\mathcal{C}_\mathcal{D})$ can be found in $O(n+m)$ time, where m is the number of edges in $\Omega(\mathcal{C}_\mathcal{D})$ [TY84,BP93]. Summing up, the first two steps can be done in $O(n^2)$ time since $m < n^2$. We now turn to the time complexity of the loop (in fact the dominant term of our analysis). For each maximal clique C of $\Omega(\mathcal{C}_\mathcal{D})$, the graph G_C can be constructed in $O(n)$ time since $|C| \leq n$. We now consider the computation of a maximal matching in G_C. Micali and Vazirani [MV80] (see also [Vaz94]) gave an $O(\sqrt{|V|}|E|)$ time algorithm for finding a maximal matching in a graph $G = (V, E)$. But G_C has at most n edges (as each edge corresponds to a 2-interval) and hence has at most $2n$ vertices. Then it follows that a maximum matching $\mathcal{M}$ in G_C can be found in $O(n\sqrt{n})$ time. Since $\Omega(\mathcal{C}_\mathcal{D})$ is an interval graph with n vertices, it has at most n maximal cliques [FG65], we conclude that the algorithm as a whole runs in $O(n^2\sqrt{n})$ time. □

Max $\{\sqsubset, \emptyset\}$-Comparable 2-Interval Pattern

Input: A set of 2-intervals $\mathcal{D}$ with disjoint support
Output: The size of a maximum cardinality $\{\sqsubset, \emptyset\}$-comparable subset of $\mathcal{D}$

1. Construct the interval graph $\Omega(\mathcal{C}_\mathcal{D})$
2. Compute all maximal cliques in $\Omega(\mathcal{C}_\mathcal{D})$
3. For each maximal clique C in $\Omega(\mathcal{C}_\mathcal{D})$
   3.1. Construct the graph G_C
   3.2. Compute a maximal matching $\mathcal{M}$ in G_C
   3.3. Store the cardinality of $\mathcal{M}$ in $m(C)$
4. Return $\max\{m(C) : C$ is a maximal clique of $\Omega(\mathcal{C}_\mathcal{D})\}$

Fig. 1. Algorithm Max $\{\sqsubset, \emptyset\}$-Comparable 2-Interval Pattern.

5 2-IP-Uni-$\{<, \not\wr\}$ is NP-complete

Theorem 1 below completes the analysis of 2-IP-Uni-R and 2-IP-Unl-R for any model $R \subseteq \{<, \sqsubset, \not\wr\}$ (see Table 1).

Theorem 1. *The* 2-IP-Uni-$\{<, \not\wr\}$ *problem is* **NP***-complete.*

Proof. The proof is by reduction from the Exact 3-CNF Sat problem. Due to space considerations, the rather technical proof is deferred to the full version of this paper.

6 A Fixed-Parameter Algorithm for 2-IP-Uni-$\{<, \not\wr\}$

According to Theorem 1, finding the largest $\{<, \not\wr\}$-comparable subset in a set of 2-intervals on a unitary support is an **NP**-complete problem. In this section, we give an exact algorithm for that problem with strong emphasis on the crossing structure of the set of 2-intervals. More precisely, we consider the time complexity of the problem with respect to the *forward crossing number* of the input. Indeed, in the context of 2-intervals, one may reasonably expect the forward crossing number to be small compared to the number of 2-intervals. Therefore, a natural direction seems to be the question for the fixed-parameter tractability with respect to parameter $\mathsf{FCrossing}(\mathcal{D})$. In response to that question, we show that the problem can be solved for any support by means of dynamic programming in $O(n \cdot \mathsf{FCrossing}(\mathcal{D}) \cdot 2^{\mathsf{FCrossing}(\mathcal{D})}(\log(n) + \mathsf{FCrossing}(\mathcal{D})))$ time where n is the number of 2-intervals in $\mathcal{D}$, and hence is fixed-parameter tractable with respect to parameter $\mathsf{FCrossing}(\mathcal{D})$.

For any $D_i \in \mathcal{D}$, let $T(D_i)$ denote the size of the largest $\{<, \not\wr\}$-comparable subset $\mathcal{D}' \subseteq \mathcal{D}$ of which the 2-interval D_i is the rightmost element. Furthermore, for any $D_i, D_j \in \mathcal{D}$ such that $D_j \not\wr D_i$, let $T(D_j \mid D_i)$ denotes the size of the largest $\{<, \not\wr\}$-comparable subset $\mathcal{D}' \subseteq \mathcal{D}$ such that (1) the 2-interval D_j is the rightmost element of $\mathcal{D}'$ and (2) the 2-interval D_i is not part of the subset $\mathcal{D}'$ but can safely be added to $\mathcal{D}'$ to obtain a new $\{<, \not\wr\}$-comparable subset of size $|\mathcal{D}'| + 1$.

Clearly, a maximum cardinality $\{<, \not\wr\}$-comparable subset $\mathcal{D}' \subseteq \mathcal{D}$ of which the 2-interval D_i is the rightmost element can be obtained either (1) by adding D_i to a maximum cardinality $\{<, \not\wr\}$-comparable subset $\mathcal{D}'' \subseteq \mathcal{D}$ whose rightmost 2-interval D_j precedes the 2-interval D_i, *i.e.*, $D_j < D_i$, or (2) by adding D_i to a maximum cardinality $\{<, \not\wr\}$-comparable subset $\mathcal{D}'' \subseteq \mathcal{D}$ whose rightmost 2-interval D_j crosses the 2-interval D_i, *i.e.*, $D_j \not\wr D_i$, and such that D_i crosses or precedes any 2-interval of $\mathcal{D}''$. Here is another way of stating these observations:

$$\forall D_i \in \mathcal{D}, \qquad T(D_i) = 1 + \max \begin{cases} \max \{T(D_j) : D_j < D_i\} \\ \max \{T(D_j \mid D_i) : D_j \not\wr D_i\} \end{cases} \tag{1}$$

What is left is thus to compute $T(D_j \mid D_i)$. To this aim, we extend the notation $T(D_j \mid D_i)$ as follows: for any $\{\not\wr\}$-comparable subset $\{D_{i_1}, D_{i_2}, \ldots, D_{i_k}\} \subseteq$

$\mathcal{D}$, $k \geq 1$, satisfying $\mathsf{Right}(D_{i_1}) < \mathsf{Right}(D_{i_2}) < \ldots < \mathsf{Right}(D_{i_k})$, we let $T(D_{i_1} \mid D_{i_2}, \ldots, D_{i_k})$ stand for the size of a largest $\{<, \between\}$-comparable subset $\mathcal{D}' \subseteq \mathcal{D}$ such that (1) the 2-interval D_{i_1} is the rightmost element of $\mathcal{D}'$ and (2) the 2-intervals $\{D_{i_2}, D_{i_3}, \ldots, D_{i_k}\}$ are not part of the subset $\mathcal{D}'$ but can safely be added to $\mathcal{D}'$ to obtain a new $\{<, \between\}$-comparable subset of size $T(D_{i_1} \mid D_{i_2}, \ldots, D_{i_k}) + k - 1$. A straightforward extension of the calculation (1) yields the following recurrence relation for computing the entry $T(D_{i_1} \mid D_{i_2}, \ldots, D_{i_k})$ of the dynamic programming table:

$$T(D_{i_1} \mid D_{i_2}, \ldots, D_{i_k}) = 1 + \max \begin{cases} \max \{T(D_j) \mid D_j \text{ satisfies condition (1)}\} \\ \max \{T(D_j \mid D_{i_1}) \mid D_j \text{ satisfies condition (2)}\} \\ \max \{T(D_j \mid D_{i_1}, D_{i_2}) \mid D_j \text{ satisfies condition (3)}\} \\ \quad \vdots \\ \max \{T(D_j \mid D_{i_1}, D_{i_2}, \ldots, D_{i_k}) \mid D_j \text{ satisfies condition } (k+1)\} \end{cases} \tag{2}$$

where condition (i), $1 \leq i \leq k+1$, is defined as follows:

$$\text{condition } (i) \qquad \begin{cases} D_j \between D_{i_r} & \text{for all } 0 < r < i \qquad \text{(crossing conditions)} \\ D_j < D_{i_s} & \text{for all } i \leq s < k+1 \qquad \text{(precedence conditions)} \end{cases}$$

It follows from the above recurrence relation that entries of the form $T(D_i \mid *)$ depend only on entries of the form $T(D_j \mid *)$ where $D_j < D_i$ or $D_j \between D_i$. From a computational point of view, this implies that the calculation of entries of the form $T(D_i \mid *)$ depends only on of the calculation of entries of the form $T(D_j \mid *)$ where $\mathsf{Right}(D_j) < \mathsf{Right}(D_i)$. The following easy lemma gives an upper-bound on the size of the dynamic programming table T with respect to the forward crossing number of $\mathcal{D}$.

Lemma 3. *The number of distinct entries of the dynamic programming table T is upper-bounded by $|\mathcal{D}| \cdot 2^{\mathsf{FCrossing}(\mathcal{D})}$.*

Proof. For any 2-interval $D_i \in \mathcal{D}$, the number of distinct $\{\between\}$-comparable subsets of which D_i is the leftmost element is upper-bounded by $2^{\mathsf{FCrossing}(\mathcal{D})}$, and hence there exist at most $2^{\mathsf{FCrossing}(\mathcal{D})}$ distinct entries of the form $T(D_i \mid *)$ in the dynamic programming table T. □

The overall algorithm for finding the size of the largest $\{<, \between\}$-comparable subset in a set of 2-intervals is given in Figure 2. Using a suitable data structure for efficiently searching 2-intervals, we have the following result (proof deferred to the full version of this paper).

Proposition 3. *Algorithm* Max $\{<, \between\}$-Comparable 2-Interval Pattern *returns the size of a maximum cardinality $\{<, \between\}$-comparable subset of a set of 2-intervals $\mathcal{D}$ in $O(n^2 \cdot \mathsf{FCrossing}(\mathcal{D}) \cdot 2^{\mathsf{FCrossing}(\mathcal{D})}(\log(n) + \mathsf{FCrossing}(\mathcal{D})))$ time where n is the number of 2-intervals in $\mathcal{D}$.*

Max $\{<, \between\}$-Comparable 2-Interval Pattern

Input: A set $\mathcal{D}$ of n 2-intervals.
Output: The maximum size of a $\{<, \between\}$-comparable pattern in $\mathcal{D}$.

1. Sort the set $\mathcal{D}$ according to their right interval. For the sake of clarity, let us assume that the ordered 2-intervals set is now given by $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$, *i.e.*, $\mathsf{Right}(D_i) < \mathsf{Right}(D_j)$ implies $i < j$. All ordered subsets considered in the following of the algorithm are to be understood as ordered with respect to that order.
2. For i from 1 to n
   2.1. Fill the entry $T(D_i)$.
   2.2. For all ordered non-empty set $\{D_{i_1}, D_{i_2}, \ldots, D_{i_q}\} \subseteq \mathcal{D}$ such that $\{D_i\} \cup \{D_{i_1}, D_{i_2}, \ldots, D_{i_q}\}$ is an ordered subset of $\{\between\}$-comparable 2-intervals with $\mathsf{Right}(D_i) < \mathsf{Right}(D_{i_1}) < \ldots < \mathsf{Right}(D_{i_q})$, fill the entry $T(D_i \mid D_{i_1}, D_{i_2}, \ldots, D_{i_q})$ according to the recurrence relation (2).
3. Return the largest entry $T(D_i)$

Fig. 2. Algorithm Max $\{<, \between\}$-Comparable 2-Interval Pattern.

Corollary 1. *The* 2-IP-Uni-$\{\sqsubset, \between\}$ *problem is fixed-parameter tractable with respect to parameter* $\mathsf{FCrossing}(\mathcal{D})$.

It remains open, however, whether the 2-IP-Uni-$\{\sqsubset, \between\}$ problem is fixed-parameter tractable with respect to parameter $\mathsf{Depth}(\mathcal{D})$ (recall indeed that $\mathsf{FCrossing}(\mathcal{D}) \geq \mathsf{Depth}(\mathcal{D})$).

7 Conclusion

In the context of structured pattern matching, we considered the problem of finding an occurrence of a given structured pattern in a set of 2-intervals and solved three open problems of [Via04]. We gave an optimal $O(n \log n)$ algorithm for model $R = \{\sqsubset\}$ thereby improving the complexity of the best known algorithm. Also, we described a $O(n^2\sqrt{n})$ time algorithm for model $R = \{\sqsubset, \between\}$ over a disjoint support. Finally, we proved that the problem is **NP**-complete for model $R = \{<, \between\}$ over a unitary support, and in addition to that, we gave a fixed parameter-tractability result based on the crossing structure of the set of 2-intervals. These results almost complete the table of complexity classes for the 2-interval pattern problem proposed by Vialette [Via04] (see Table 1).

An interesting question would be to answer the last remaining open problem in that area, that is to determine whether there exists a polynomial time algorithm for 2-IP-Dis-$\{<, \between\}$, *i.e.*, finding the largest $\{<, \between\}$-comparable subset of a set of 2-intervals over a disjoint support[1]. In the light of Table 1, we conjecture that problem to be polynomial time solvable.

[1] The 2-IP-Dis-$\{<, \between\}$ problem has an immediate formulation in terms of constrained matchings in general graphs: Given a graph G together with a linear ordering π

References

[AGGN02] J. Alber, J. Gramm, J. Guo, and R. Niedermeier, *Towards optimally solving the longest common subsequence problem for sequences with nested arc annotations in linear time*, Proceedings of the 13th Annual Symposium on Combinatorial Pattern Matching (CPM 2002), Lecture Notes in Computer Science, vol. 2373, Springer-Verlag, 2002, pp. 99–114.

[BP93] J.R.S. Blair and B. Peyton, *An introduction to chordal graphs and clique trees*, Graph Theory and Sparse Matrix Computation **56** (1993), 1–29.

[BYHN+02] R. Bar-Yehuda, M.M. Halldorsson, J. Naor, H. Shachnai, and I. Shapira, *Scheduling split intervals*, Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 732–741.

[DGP88] I. Dagan, M.C. Golumbic, and R.Y. Pinter, *Trapezoid graphs and their coloring*, Discrete Applied Mathematics **21** (1988), 35–46.

[Eva99] P. Evans, *Finding common subsequences with arcs and pseudoknots*, Proceedings of the 10th Annual Symposium Combinatorial Pattern Matching (CPM 1999), Lecture Notes in Computer Science, vol. 1645, Springer-Verlag, 1999, pp. 270–280.

[FG65] D.R. Fulkerson and O.A. Gross, *Incidence matrices and interval graphs*, Pacific Journal of Math. **15** (1965), 835–855.

[FMW97] S. Felsner, R. Müller, and L. Wernisch, *Trapezoid graphs and generalizations: Geometry and algorithms*, Discrete Applied Math. **74** (1997), 13–32.

[Fre75] M.L. Fredman, *On computing the length of longest increasing subsequences*, Disrete Mathematics **11** (1975), 29–35.

[GGN02] J. Gramm, J. Guo, and R. Niedermeier, *Pattern matching for arc-annotated sequences*, Proceedings of the the 22nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2002), Lecture Notes in Computer Science, vol. 2556, 2002, pp. 182–193.

[GIP99] D. Goldman, S. Istrail, and C.H. Papadimitriou, *Algorithmic aspects of protein structure similarity*, Proceedings of the 40th Annual Symposium of Foundations of Computer Science (FOCS99), 1999, pp. 512–522.

[Gol80] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, New York, 1980.

[GW79] J.R. Griggs and D.B. West, *Extremal values of the interval number of a graph, I*, SIAM J. Alg. Discrete Methods **1** (1979), 1–7.

[JLMZ00] T. Jiang, G.-H. Lin, B. Ma, and K. Zhang, *The longest common subsequence problem for arc-annotated sequences*, In Proc. 11th Annual Symposium on Combinatorial Pattern Matching (CPM 2000), Lecture Notes in Computer Science, vol. 1848, Springer-Verlag, 2000, pp. 154–165.

[JMT92] D. Joseph, J. Meidanis, and P. Tiwari, *Determining DNA sequence similarity using maximum independent set algorithms for interval graphs*, Proceedings of the Third Scandinavian Workshop on Algorithm Theory (SWAT 92), Lecture Notes in Computer Science, Springer-Verkag, 1992, pp. 326–337.

---

of the vertices of G, the 2-IP-Dis-$\{<, \emptyset\}$ problem is equivalent to finding a maximum cardinality matching $\mathcal{M}$ in G with the property that for any two distinct edges $\{u, v\}$ and $\{u', v'\}$ of $\mathcal{M}$ neither $\min\{\pi(u), \pi(v)\} < \min\{\pi(u'), \pi(v')\}$ and $\max\{\pi(u'), \pi(v')\} < \max\{\pi(u), \pi(v)\}$ nor $\min\{\pi(u'), \pi(v')\} < \min\{\pi(u), \pi(v)\}$ and $\max\{\pi(u), \pi(v)\} < \max\{\pi(u'), \pi(v')\}$ occur.

[MV80] S. Micali and V.V. Vazirani, *An $O(\sqrt{|V||E|})$ algorithm for finding maximum matching in general graphs*, Proceedings of the 21st Annual Symposium on Foundation of Computer Science, IEEE, 1980, pp. 17–27.

[TH79] W.T. Trotter and F. Harary, *On double and multiple interval graphs*, J. Graph Theory **3** (1979), 205–211.

[TY84] R.E. Tarjan and M Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput. **13** (1984), 566–579.

[Vaz94] V.V. Vazirani, *A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{|V||E|})$ maximum matching algorithm*, Combinatorica **14** (1994), no. 1, 71–109.

[VD01] J. Viksna and D.Gilbert, *Pattern matching and pattern discovery algorithms for protein topologies*, Lecture Notes in Computer Science, vol. 2149, Springer, 2001, pp. 98–111.

[Via02] S. Vialette, *Pattern matching over 2-intervals sets*, In Proc. 13th Annual Symposium Combinatorial Pattern Matching (CPM 2002), Lecture Notes in Computer Science, vol. 2373, Springer-Verlag, 2002, pp. 53–63.

[Via04] ———, *On the computational complexity of 2-interval pattern matching*, Theoretical Computer Science **312** (2004), no. 2-3, 223–249.

[WS84] D.B. West and D.B. Shmoys, *Recognizing graphs with fixed interval number is NP-complete*, Discrete Applied Mathematics **8** (1984), 295–305.