Approximating the 2-Interval Pattern Problem

Maxime Crochemore, Danny Hermelin, Gad M. Landau, Stéphane Vialette

Approximating the 2-Interval Pattern problem

Maxime Crochemore*, Danny Hermelin**,
Gad M. Landau***, and Stéphane Vialette†

Abstract. We address the problem of approximating the 2-Interval Pattern problem over its various models and restrictions. This problem, which is motivated by RNA secondary structure prediction, asks to find a maximum cardinality subset of a 2-interval set with respect to some prespecified model. For each such model, we give varying approximation quality depending on the different possible restrictions imposed on the input 2-interval set.

1 Introduction

In the context of RNA secondary structure prediction, Vialette [11] proposed a geometric representation of a helix in an RNA single stranded molecule by means of a natural generalization of an interval, namely a 2-*interval*. A 2-interval is the union of two disjoint intervals defined over a single line. In [11], intervals and 2-intervals represent respectively sequences of contiguous bases and possible pairings between such sequences in the RNA secondary structure. The goal is to find a maximum disjoint subset of the given set of 2-intervals, restricted to prespecified geometrical constrains, so as to serve as a valid approximation of the actual secondary structure of the given RNA.

Throughout the paper, a 2-interval is denoted by $D = (I, J)$ where I and J are two (closed) intervals defined over a single line such that $I < J$, *i.e.*, I is completely to the left of J. Two 2-intervals $D_1 = (I_1, J_1)$ and $D_2 = (I_2, J_2)$ are *disjoint*, if both 2-intervals share no common point, *i.e.*, $(I_1 \cup J_1) \cap (I_2 \cup J_2) = \emptyset$. For such disjoint pairs of 2-intervals, three natural binary relations are of special interest.

Definition 1 (Relations between 2-intervals). *Let $D_1 = (I_1, J_1)$ and $D_2 = (I_2, J_2)$ be two disjoint 2-intervals. Then*

* Institut Gaspard-Monge, Université de Marne-la-Vallée, France, and Department of Computer Science, King's Collage, London, UK. Email: maxime.crochemore@univ-mlv.fr. Partially supported by CNRS, France, and the French Ministry of Research through ACI NIM.

** Department of Computer Science, University of Haifa, Israel. Email: danny@cri.haifa.ac.il.

*** Department of Computer Science, University of Haifa, Israel, and Department of Computer and Information Science, Polytechnic University, NY, USA. Email: landau@cs.haifa.ac.il. Partially supported by the Israel Science Foundation grant 282/01.

† Laboratoire de Recherche en Informatique (LRI), Université Paris-Sud, France. Email: vialette@lri.fr.

– $D_1 < D_2$ *(D_1* precedes *D_2), if* $I_1 < J_1 < I_2 < J_2$.
– $D_1 \sqsubset D_2$ *(D_1 is* nested *in D_2), if* $I_2 < I_1 < J_1 < J_2$.
– $D_1 \between D_2$ *(D_1* crosses *D_2), if* $I_1 < I_2 < J_1 < J_2$.

A pair of 2-intervals D_1 and D_2 is *R-comparable* for some $R \in \{<, \sqsubset, \between\}$, if either $(D_1, D_2) \in R$ or $(D_2, D_1) \in R$. A set of 2-intervals $\mathcal{D}$ is $\mathcal{R}$-comparable for some $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, $\mathcal{R} \neq \emptyset$, if any pair of distinct 2-intervals in $\mathcal{D}$ is R-comparable for some $R \in \mathcal{R}$. The non-empty subset $\mathcal{R}$ is called a *model*. Note that any two disjoint 2-intervals are R-comparable for some $R \in \{<, \sqsubset, \between\}$. Equivalently, any pairwise disjoint subset of $\mathcal{D}$ is $\{<, \sqsubset, \between\}$-comparable. In [3, 11], the 2-Interval Pattern problem is defined as follows:

Definition 2 (The 2-Interval Pattern **problem).** *Let $\mathcal{D}$ be a set of* 2*-intervals and let $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, $\mathcal{R} \neq \emptyset$, be a given model. The* 2-Interval Pattern *problem asks to find a maximum cardinality $\mathcal{R}$-comparable subset of $\mathcal{D}$.*

By the above definition, any solution for the 2-Interval Pattern problem over a model $\mathcal{R}$ corresponds to a secondary structure constrained by $\mathcal{R}$. Let $\mathcal{D}$ be a set of 2-intervals and let $\mathcal{S}(\mathcal{D}) = \{I, J : D = (I, J) \in \mathcal{D}\}$ be the set of intervals *involved* in $\mathcal{D}$. Several biologically motivated restrictions on $\mathcal{D}$ and $\mathcal{S}(\mathcal{D})$ are of interest.

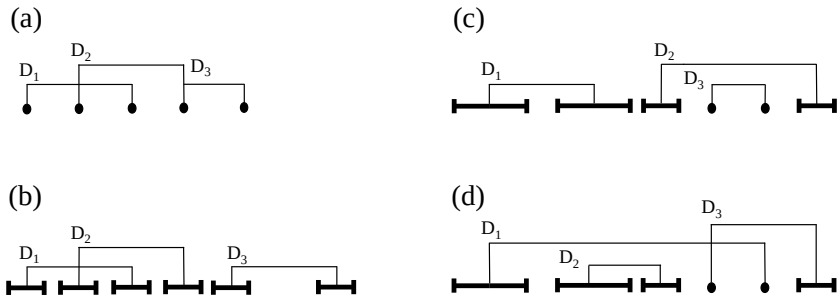

Fig. 1. Restrictions for 2-interval sets. Intervals are represented by dark lines or circles and 2-intervals are represented by a thin line connecting two intervals. (a) A point 2-interval set where $D_1 \between D_2$ and $D_1 < D_3$. D_2 and D_3 are not disjoint and thus are not comparable by any relation. (b) A unitary 2-interval set where $D_1 \between D_2$, $D_1 < D_3$, and $D_2 < D_3$. (c) A balanced 2-interval set where $D_3 \sqsubset D_2$. The entire set is $\{<, \sqsubset\}$-comparable. (d) An unlimited $\{<, \sqsubset, \between\}$-comparable 2-interval set.

Definition 3. *Let $\mathcal{D}$ be a set of* 2*-intervals and let $\mathcal{S}(\mathcal{D})$ be the set of intervals involved in $\mathcal{D}$.*

– *$\mathcal{D}$ is a* point 2-interval set *if all intervals in $\mathcal{S}(\mathcal{D})$ are pairwise disjoint (note that in this case, all intervals in $\mathcal{S}(\mathcal{D})$ may be considered as points).*

– $\mathcal{D}$ *is a* unitary 2-interval set *if* $\mathcal{S}(\mathcal{D})$ *consists of intervals of unit length.*
– $\mathcal{D}$ *is a* balanced 2-interval set *if any* 2*-interval in* $\mathcal{D}$ *is a pair of two intervals of equal length.*
– $\mathcal{D}$ *is an* unlimited 2-interval set *if none of the above restrictions are imposed.*

The left part of Table 1 depicts the current state of the art for the 2-Interval Pattern problem in terms of exact algorithms. In [11], the 2-Interval Pattern problem over the $\{\sqsubset, \between\}$ and $\{<, \sqsubset, \between\}$ models is proved to be **NP**-hard even for unitary 2-interval sets. The proof for the $\{<, \sqsubset, \between\}$ model is obtained as a direct consequence of the **APX**-hardness result for the Maximum Independent Set problem for t-interval graphs given in [2]. The results in [2] also provide approximation algorithms for this model. In [3], an **NP**-hardness result for the $\{<, \between\}$ model restricted to unitary 2-interval sets is given. The time complexity for this same model when the input is restricted to point 2-interval sets is still unknown [11, 3]. These results imply that in practical terms, secondary structures containing pseudoknots are hard to predict in our suggested mathematical model. This is consistent with previously known **NP**-hardness results for RNA secondary structures prediction in other models considering arbitrary pseudoknots [1, 8, 9].

Classical complexity				
Model	Unl.	Bal.	Uni.	Pnt.
$\{<, \sqsubset, \between\}$	**NP**-C [11, 2]			$\mathcal{O}(n\sqrt{n})$ [11]
$\{\sqsubset, \between\}$	**NP**-C [11]			$\mathcal{O}(n^2\sqrt{n})$ [3]
$\{<, \between\}$	**NP**-C [3]			?
$\{<, \sqsubset\}$	$\mathcal{O}(n^2)$ [11]			
$\{\between\}$	$\mathcal{O}(n^2 \log n)$ [11]			
$\{\sqsubset\}$	$\mathcal{O}(n \log n)$ [3]			
$\{<\}$	$\mathcal{O}(n \log n)$ [11]			

Approximation factors				
Model	Unl.	Bal.	Uni.	Pnt.
$\{<, \sqsubset, \between\}$ (Section 2)	4^a	4^b	3^c	–
$\{\sqsubset, \between\}$ (Section 3)	4^a	4^d	3^e	–
$\{<, \between\}$ (Section 4)	6^b	5^b	3^c	2^c

[a] Polynomial-time [2].
[b] $\mathcal{O}(n^2)$ time algorithm.
[c] $\mathcal{O}(n \lg n)$ time algorithm [2].
[d] $\mathcal{O}(n^3)$ time algorithm.
[e] $\mathcal{O}(n^2 \lg n)$ time algorithm.

Table 1. The 2-Interval Pattern problem over it's various models and restrictions. Left part: Classical complexity results for the 2-Interval Pattern problem, where $n = |\mathcal{D}|$. Right part: The approximation factors we obtain in this paper.

In this paper we focus on the three **NP**-hard models of the 2-Interval Pattern problem. More specifically, we design constant factor approximation algorithms for the $\{<, \sqsubset, \between\}$, $\{\sqsubset, \between\}$, and $\{<, \between\}$ models. The approximation factors obtained by all our algorithms vary depending on the restriction imposed on the input set of 2-intervals (see Table 1). Furthermore we suggest a new restriction, namely balanced 2-interval sets. By definition, unitary 2-interval sets are also balanced but the converse is not necessarily true. Consequently, the above mentioned **NP**-hardness results also hold for the balanced case, and moreover, balanced 2-interval sets introduce a new combinatorial object which requires

particular consideration. Furthermore, the balanced restriction is very natural in the biological setting of the problem.

This paper is organized as follows. In Section 2, we consider the 2-Interval Pattern problem over the general model, *i.e.*, the $\{<, \sqsubset, \between\}$ model. We describe in Section 3 an approximation algorithm for the problem over the $\{\sqsubset, \between\}$ model. Finally, in Section 4, the $\{<, \between\}$ model is considered, and different approximation algorithms are introduced for all possible restrictions imposed on the input.

2 Approximation algorithms for the $\{<, \sqsubset, \between\}$ model.

We begin by considering the 2-Interval Pattern problem over the general model, *i.e.*, the $\{<, \sqsubset, \between\}$ model. Recall that in this case, given an input set of 2-intervals $\mathcal{D}$, the problem asks to find a maximum $\{<, \sqsubset, \between\}$-comparable subset of $\mathcal{D}$, which is equivalent to finding a maximum pairwise disjoint subset of $\mathcal{D}$.

For point 2-intervals sets, 2-Interval Pattern can be solved in polynomial time by maximum matching [11]. For unitary 2-interval sets, the problem is already **APX**-hard [2], and therefore is **APX**-hard also for balanced and unlimited 2-interval sets. Furthermore, the results in [2] also yield approximation algorithms for our case which directly imply the following.

Proposition 1 ([2]). *The* 2-Interval Pattern *problem over the* $\{<, \sqsubset, \between\}$ *model can be approximated within a factor of 4 when restricted to unlimited* 2*-interval sets, and a factor of 3 when restricted to unitary interval sets.*

The algorithm given in [2] that solves the case of unitary 2-interval sets can be executed in $\mathcal{O}(n \lg n)$ time, where n is the size of the input set of 2-intervals. However, the algorithm for unlimited 2-interval sets uses linear programming techniques, which in practice are very often too time costly. Clearly, balanced 2-interval sets lie between the two cases and are arguably the most biologically important case. In the rest of this section we describe a quadratic time 4-approximation algorithm for balanced 2-intervals sets.

Given any balanced 2-interval set $\mathcal{D}$, let the *smallest* 2-interval in $\mathcal{D}$ be the 2-interval with the shortest left (or right, as they are both of equal length) interval among all left intervals involved in $\mathcal{D}$. The algorithm we suggest is a simple greedy strategy that repeatedly picks the smallest 2-interval in the input, adds it to the solution, and omits all other 2-intervals in the input which intersect it. A schematic description of this algorithm, which we call Bal-$\{<, \sqsubset, \between\}$-Approx, is given in Figure 2.

Lemma 1. *Algorithm Bal-*$\{<, \sqsubset, \between\}$*-Approx achieves an approximation factor guarantee of 4 for the* 2-Interval Pattern *problem over the general model, restricted to balanced* 2*-interval sets.*

Proof. Let $\mathcal{D}$ be the set of remaining 2-intervals at any arbitrary iteration of Bal-$\{<, \sqsubset, \between\}$-Approx, and let $D_0 \in \mathcal{D}$ be the smallest 2-interval at this iteration. Since D_0 is the smallest 2-interval in $\mathcal{D}$, no interval involved in $\mathcal{D}$ can be properly

Algorithm Bal-$\{<, \sqsubset, \between\}$-Approx($\mathcal{D}$)

Data : A set of balanced 2-intervals $\mathcal{D}$.
Result : A $\{<, \sqsubset, \between\}$-comparable subset of $\mathcal{D}$.
begin
  while $\mathcal{D} \neq \emptyset$ **do**
    1. Let D_0 be the smallest 2-interval in $\mathcal{D}$.
    2. Add D_0 to the solution.
    3. Omit D_0 and all 2-intervals intersecting D_0 from $\mathcal{D}$.
  end
  return *the* 2*-intervals chosen for the solution.*
end

Fig. 2. A schematic description of algorithm Bal-$\{<, \sqsubset, \between\}$-Approx.

contained in the left or right interval of D_0. Thus, there can be at most four disjoint intervals involved in $\mathcal{D}$, which intersect D_0 at this given iteration. It follows that at this iteration, at most four 2-intervals in the optimal solution are omitted from $\mathcal{D}$. Applying this argument for all iterations of the algorithm yields the desired approximation factor guarantee. □

Time complexity. Given an input set of 2-intervals $\mathcal{D}$ of size n, algorithm Bal-$\{<, \sqsubset, \between\}$-Approx can be implemented straightforwardly to run in $\mathcal{O}(n^2)$ time.

3 An approximation algorithm for the $\{\sqsubset, \between\}$ model.

We next consider the 2-Interval Pattern problem over the $\{\sqsubset, \between\}$ model. Recall that the 2-Interval Pattern problem over this model is **NP**-complete even for unitary 2-interval sets [11]. In the following we introduce a single algorithm which achieves different constant approximation factors for unitary, balanced and unlimited 2-interval sets. More specifically, we describe an algorithm which uses the algorithms described in the previous section as sub-procedures, choosing the specific algorithm according to the restriction imposed on the input. Our algorithm is a direct generalization of the algorithm devised in [3] for the 2-Interval Pattern problem over the $\{\sqsubset, \between\}$ model, restricted to point 2-interval sets. As in [3], the notion of *interval graphs* is used extensively throughout the section. An interval graph is an intersection graph of a finite family of intervals, all defined over a single line [7, 10].

Given a 2-interval $D = (I, J)$, let $C(D)$ denote the smallest interval that covers $\mathcal{D}$, *i.e.*, $C(D) = [l(I) : r(J)]$ where $l(I)$ and $r(J)$ are the left and right endpoints of I and J, respectively. Blin *et al.* [3] called $C(D)$ the *covering interval* of D. They also observed that any pair of disjoint 2-intervals are $\{\sqsubset, \between\}$-comparable if and only if their corresponding covering intervals intersect. Thus, given a set of 2-intervals $\mathcal{D}$, and the set $\mathcal{C}(\mathcal{D})$ of all covering intervals of 2-intervals in $\mathcal{D}$, any $\{\sqsubset, \between\}$-comparable subset $\mathcal{D}' \subseteq \mathcal{D}$ corresponds to a pairwise intersecting subset of $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{D})$. However, the converse is not true as a pair of non-disjoint

2-intervals have corresponding intersecting covering intervals as well. Hence, a pairwise intersecting subset of $\mathcal{C}(\mathcal{D})$ can contain corresponding 2-intervals which are non-disjoint in $\mathcal{D}$.

Let $\mathcal{D}$ be the input set of 2-intervals and $\mathcal{C}(\mathcal{D})$ be the set of covering intervals of all 2-intervals in $\mathcal{D}$. First, we construct the interval graph $\Omega_{\mathcal{C}(\mathcal{D})}$ of $\mathcal{C}(\mathcal{D})$. Since $\Omega_{\mathcal{C}(\mathcal{D})}$ is an interval graph, it has at most $|V(\Omega_{\mathcal{C}(\mathcal{D})})| = |\mathcal{D}|$ maximal (in containment order) cliques, and all these maximal cliques can be computed in polynomial time [6]. Note that any pair of 2-intervals with covering intervals in a maximal clique, are either nesting or crossing (but not preceding), or they are non-disjoint. Now, let OPT denote a maximum cardinality $\{\sqsubset, \between\}$-comparable subset of $\mathcal{D}$ and let $\mathcal{C}(OPT)$ be the set of covering intervals of OPT. The subgraph of $\Omega_{\mathcal{C}(\mathcal{D})}$ which corresponds to $\mathcal{C}(OPT)$ is a clique, and is thus a subset of a maximal clique in $\Omega_{\mathcal{C}(\mathcal{D})}$. Furthermore, any 2-interval with a covering interval in this clique and not in OPT is necessarily non-disjoint with at least one of the 2-intervals in OPT.

Observation 1. *Let OPT denote the maximum $\{\sqsubset, \between\}$-comparable subset of $\mathcal{D}$. Then OPT is a maximum pairwise disjoint subset of a set of 2-intervals $\mathcal{D}'$, $OPT \subseteq \mathcal{D}' \subseteq \mathcal{D}$, such that $\mathcal{C}(\mathcal{D}')$, the covering intervals of OPT, corresponds to a maximal clique in $\Omega_{\mathcal{C}(\mathcal{D})}$.*

Given the 2-intervals which corresponds to a maximal clique in $\Omega_{\mathcal{C}(\mathcal{D})}$, one can use the algorithms in Section 2 to find an approximation of the maximum pairwise disjoint subset of these 2-intervals. A detailed schematic description of our algorithm, which is called $\{\sqsubset, \between\}$-Approx, is given in Figure 3.

Algorithm $\{\sqsubset, \between\}$-Approx($\mathcal{D}$)

Data : A set of 2-intervals $\mathcal{D}$.
Result : A $\{\sqsubset, \between\}$-comparable subset of $\mathcal{D}$.
begin
  1. Construct $\mathcal{C}(\mathcal{D})$, the set of covering intervals of all 2-intervals in $\mathcal{D}$.
  2. Construct $\Omega_{\mathcal{C}(\mathcal{D})}$, the interval graph of $\mathcal{C}(\mathcal{D})$.
  3. Compute all maximal cliques of $\Omega_{\mathcal{C}(\mathcal{D})}$ using [6].
  4. foreach *maximal clique C of $\Omega_{\mathcal{C}(\mathcal{D})}$* **do**
    (a) Compute $\mathcal{D}_C \subseteq \mathcal{D}$, the 2-intervals with covering intervals in C.
    (b) Approximate the maximum pairwise disjoint subset of $\mathcal{D}_C$, using the algorithms described in the previous section.
  end
  return *the largest pairwise disjoint subset found in step 4(b).*
end

Fig. 3. A schematic description of algorithm $\{\sqsubset, \between\}$-Approx.

Lemma 2. *Algorithm $\{\sqsubset, \between\}$-Approx is a 4-approximation (3-approximation) algorithm for the* 2-Interval Pattern *problem for unlimited (unitary) 2-interval sets.*

Proof. Immediate from the above discussion and from Proposition 1 and Lemma 1. □

Time complexity. The number of sub-procedure invocations in step 4(b) of $\{\sqsubset, \between\}$-Approx is bounded by $\mathcal{O}(n)$ where n denotes the size of the input set. Also, generating all maximal cliques of $\Omega_{\mathcal{C}(\mathcal{D})}$ can be done in $\mathcal{O}(n^2)$ time. Hence, we have a super-quadratic running time of $\mathcal{O}(n^2 \lg n)$ for unitary 2-interval sets and a $\mathcal{O}(n^3)$ running time for balanced 2-interval sets. For unlimited 2-interval sets, the running time of $\{\sqsubset, \between\}$-Approx is polynomial [2].

4 Approximation algorithms for the $\{<, \between\}$ model.

We now turn to considering the 2-Interval Pattern problem over the $\{<, \between\}$ model. Recall that the problem is known to be **NP**-hard for unitary 2-interval sets, while for point 2-interval sets the problem is not known to be in **P** [3]. Thus, in the following section we consider all possible restrictions for the $\{<, \between\}$ model. More specifically, we design a 3-approximation algorithm for unitary 2-interval sets which is also a 2-approximation algorithm for point 2-interval sets. We later slightly modify this algorithm to obtain a 5-approximation algorithm for balanced 2-interval sets. Finally, we introduce a different more complex modification which yields a 6-approximation algorithm for unlimited 2-interval sets.

Throughout the section, we will use the notion of *trapezoid graph* [4, 5]. Consider two intervals, I' and J', defined over two distinct horizontal lines. The trapezoid $T = (I', J')$ is the convex set of points bounded by I' and J', and the two arcs connecting the right and left endpoints of I' and J'. We call I' and J' the *bottom interval* and *top interval* of T respectively. A family of trapezoids is a finite set of trapezoids which are all defined over the same two horizontal lines. The above definitions imply, that two distinct trapezoids $T_1 = (I'_1, J'_1)$ and $T_2 = (I'_2, J'_2)$ in a family of trapezoids are disjoint, *i.e.*, they contain no common point, if and only if ($I'_1 < I'_2$ and $J'_1 < J'_2$) or ($I'_2 < I'_1$ and $J'_2 < J'_1$) holds. If T_1 and T_2 are indeed disjoint, then one trapezoid is completely to left of the other, say for instance T_1, and this is denoted by $T_1 < T_2$. Finally, a trapezoid graph is an intersection graph of a family of trapezoids.

4.1 Point and unitary 2-interval sets.

We begin the discussion in this section by first describing an approximation algorithm for point and unitary 2-interval sets. We call this initial algorithm $\{<, \between\}$-Approx. The general outline of $\{<, \between\}$-Approx consists of the following stages: First $\mathcal{T}(\mathcal{D})$, a family of trapezoids representing each 2-interval in $\mathcal{D}$ is constructed. Next, the maximum pairwise disjoint subset of $\mathcal{T}(\mathcal{D})$ is computed using the algorithm proposed in [5]. Finally, trapezoids in this subset which correspond to non-disjoint 2-intervals in $\mathcal{D}$ are omitted, and the filtered solution is outputted.

Definition 4 (Corresponding trapezoid family). *Let $\mathcal{D}$ be a set of* 2*-intervals, and let α and β be two distinct horizontal lines such that α is below β. The* corresponding trapezoid family of $\mathcal{D}$*, denoted $\mathcal{T}(\mathcal{D})$, is defined as the family containing a single trapezoid $T = (I', J') \in \mathcal{D}$ for each* 2*-interval $D = (I, J) \in \mathcal{D}$, where I' is defined over α, J' is defined over β, and $I' = I$ and $J = J'$.*

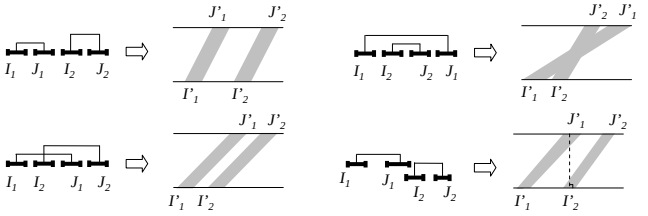

Fig. 4. $\{<, \between\}$-comparable 2-intervals correspond to disjoint trapezoids but the converse is not necessarily true.

Let $\mathcal{D}$ be a set of 2-intervals and let $\mathcal{T}(\mathcal{D})$ be the corresponding trapezoid family of $\mathcal{D}$. It is not difficult to see that $\{<, \between\}$-comparable 2-intervals in $\mathcal{D}$ correspond to disjoint trapezoids in $\mathcal{T}(\mathcal{D})$, while $\{\sqsubset\}$-comparable 2-intervals in $\mathcal{D}$ correspond to intersecting trapezoids in $\mathcal{T}(\mathcal{D})$ (see Figure 4).

Observation 2. *Any two disjoint* 2*-intervals in $\mathcal{D}$ are $\{<, \between\}$-comparable if and only if their corresponding trapezoids in $\mathcal{T}(\mathcal{D})$ are disjoint.*

Felsner *et al.* [5] presented an $\mathcal{O}(n \lg n)$ algorithm for finding a maximum disjoint subset in a family of n trapezoids. Unfortunately, this alone does not suffice in our case since there may be disjoint trapezoids in $\mathcal{T}(\mathcal{D})$ which correspond to non-disjoint 2-intervals in $\mathcal{D}$. (see Figure 4).

Definition 5 (Clashing intervals). *Let $I' = [l(I'), r(I')]$ and $J' = [l(J'), r(J')]$ be two distinct intervals defined over two distinct horizontal lines such that $l(I') \leq l(J')$. The two intervals I' and J' clash, if either $l(I') \leq l(J') \leq r(J') \leq r(I')$ or $l(I') \leq l(J') \leq r(I') \leq r(J')$.*

Definition 6 (Clashing trapezoids). *Let $T_1 = (I'_1, J'_1)$ and $T_2 = (I'_2, J'_2)$ be two distinct trapezoids in a family of trapezoids. The two trapezoids T_1 and T_2 clash, if either I'_1 and J'_2 clash or I'_2 and J'_1 clash.*

Observation 3. *Any pair of* 2*-intervals in $\mathcal{D}$ are $\{<, \between\}$-comparable if and only if their corresponding trapezoids in $\mathcal{T}(\mathcal{D})$ are disjoint and do not clash.*

Observation 3 is the heart of algorithm $\{<, \between\}$-Approx. Note that the number of maximal (in containment order) pairwise disjoint subsets of $\mathcal{T}(\mathcal{D})$ can be exponential, so exhaustively searching through all such subsets for a maximum

non-clashing subset is unfeasible. Let $\mathcal{T}'$ be the maximum pairwise disjoint subset of $\mathcal{T}(\mathcal{D})$. Since the optimal solution $OPT \subseteq \mathcal{D}$ also corresponds to a pairwise disjoint non-clashing subset of trapezoids, we must have $|OPT| \leq |\mathcal{T}'|$. Next we show how to obtain a a pairwise non-clashing subset of $\mathcal{T}'$ which is no more than a constant factor smaller then OPT, in case $\mathcal{D}$ is either a point or unitary 2-interval set. Namely, we find a subset of $\mathcal{T}'$ which is an approximation of OPT.

Consider the leftmost trapezoid T_0 of $\mathcal{T}'$ and let D_0 be its corresponding 2-interval in $\mathcal{D}$. By our definition of a 2-interval and of $\mathcal{T}(\mathcal{D})$, any trapezoid in $\mathcal{T}(\mathcal{D})$, has a bottom interval which is completely to the left of its top interval. Thus, T_0 can only clash with trapezoids on its right in $\mathcal{T}'$. Now, if $\mathcal{D}$ is a point 2-interval set, then all 2-intervals with left intervals intersecting the right interval of D_0 have the same left interval, and as $\mathcal{T}'$ is pairwise disjoint, at most one of these has a corresponding trapezoid in $\mathcal{T}'$. Furthermore, if $\mathcal{D}$ is a unitary 2-interval set, intersecting intervals involved in $\mathcal{D}$ must overlap. Thus, any trapezoid in $\mathcal{T}'$ clashing with T_0 corresponds to a 2-interval with a left interval which contains either endpoints, but not both, of the right interval of D_0. Since $\mathcal{T}'$ is pairwise disjoint, there can be at most two such trapezoids in $\mathcal{T}'$.

Algorithm $\{<, \between\}$-Approx first computes $\mathcal{T}'$ the maximum pairwise disjoint subset of $\mathcal{T}(\mathcal{D})$, and then repeatedly adds the leftmost trapezoids in $\mathcal{T}'$ to the solution while omitting all trapezoids which clash with this trapezoid in $\mathcal{T}'$. A schematic description of algorithm $\{<, \between\}$-Approx is given in Figure 5.

Algorithm $\{<, \between\}$-Approx($\mathcal{D}$)

Data : A set of 2-intervals $\mathcal{D}$.
Result : A $\{<, \between\}$-comparable subset of $\mathcal{D}$.
begin
  1. Construct $\mathcal{T}(\mathcal{D})$, the corresponding trapezoid set of $\mathcal{D}$.
  2. Compute $\mathcal{T}' \subseteq \mathcal{T}(\mathcal{D})$, the maximum pairwise disjoint subset of $\mathcal{T}(\mathcal{D})$ [5].
  3. while $\mathcal{T}' \neq \emptyset$ **do**
    (a) Let T_0 be the leftmost trapezoid in $\mathcal{T}'$.
    (b) Add T_0 to the solution.
    (c) Omit T_0 and all trapezoids clashing with T_0 from $\mathcal{T}'$.
  end
  return *the set of* 2*-intervals corresponding to the trapezoids in the solution.*
end

Fig. 5. A schematic description of algorithm $\{<, \between\}$-Approx.

Lemma 3. *Algorithm* $\{<, \between\}$*-Approx is a 3-approximation algorithm (*2*-approximation algorithm) for the* 2-Interval Pattern *problem over the* $\{<, \between\}$ *model restricted to unitary* 2*-interval sets (point* 2*-interval sets).*

Time complexity. Let $|\mathcal{D}| = n$. The family of trapezoids $\mathcal{T}(\mathcal{D})$ can be constructed in $\mathcal{O}(n)$ time, and according to [5], $\mathcal{T}' \subseteq \mathcal{T}(\mathcal{D})$ can be computed in $\mathcal{O}(n \lg n)$ time. In addition, each iteration in step 3 of the algorithm can easily be computed

by scanning $\mathcal{T}'$ a constant number of times. As there are $\mathcal{O}(n)$ iterations all together, it follows that step 3, and consequently algorithm $\{<, \between\}$-Approx, can be computed in $\mathcal{O}(n^2)$ time. In fact, if we sort all the right endpoints of intervals involved in $\mathcal{D}$ in an $\mathcal{O}(n \lg n)$ preprocessing stage, we can compute each iteration of step 3 in linear time with respect to the number of trapezoids omitted. As there is only a constant number of such trapezoids in each iteration, step 3 can be computed in $\mathcal{O}(n)$ time. This yields a total of $\mathcal{O}(n \lg n)$ running time.

4.2 Balanced 2-interval sets.

We next consider balanced 2-interval sets. We show that a slight modification to algorithm $\{<, \between\}$-Approx yields a 5-approximation algorithm for this case. We call this new algorithm Bal-$\{<, \between\}$-Approx. Algorithm Bal-$\{<, \between\}$-Approx differs from $\{<, \between\}$-Approx only by the fact that at each iteration of step 3, instead of choosing the leftmost trapezoid in $\mathcal{T}'$, the smallest trapezoid (*i.e.*, the trapezoid corresponding to the smallest 2-interval) amongst all trapezoids in $\mathcal{T}'$ is chosen for the solution.

Lemma 4. *Algorithm Bal-$\{<, \between\}$-Approx is a 5-approximation factor the* 2-Interval Pattern *problem over the $\{<, \between\}$ model restricted to balanced 2-interval sets.*

Proof. Consider $\mathcal{T}'$ at an arbitrary iteration of step 3 in Bal-$\{<, \between\}$-Approx, and let T_0 be the smallest trapezoid of $\mathcal{T}'$ at this iteration. Also let OPT denote the maximum $\{<, \between\}$-comparable subset of $\mathcal{D}$. Since T_0 is the smallest trapezoid, by a similar argument used in Lemma 1, T_0 clashes at most 4 other trapezoids in $\mathcal{T}'$ at this iteration. Hence, since $|OPT| \leq |\mathcal{T}'|$ prior to step 3, the promised approximation factor is obtained and the above lemma holds. □

Time complexity. Algorithm Bal-$\{<, \between\}$-Approx can be implemented straightforwardly to run in $\mathcal{O}(n^2)$ time, where $n = |\mathcal{D}|$.

4.3 Unlimited 2-interval sets.

The rest of this section is devoted to the 2-Interval Pattern problem over the $\{<, \between\}$ model for unlimited 2-interval sets. We introduce a slightly more delicate modification of $\{<, \between\}$-Approx to obtain a 6-approximation algorithm for the unlimited case. For this, we consider special trapezoid families which have structures that are convenient for our purposes.

Definition 7 (Proper trapezoid family). *A family of trapezoids $\mathcal{T}$ is proper if for any two distinct trapezoids $T_1 = (I'_1, J'_1), T_2 = (I'_2, J'_2)$ in $\mathcal{T}$, $I'_1 \cap I'_2 = \emptyset$ and $J'_1 \cap J'_2 = \emptyset$ holds.*

Definition 8 (Strongly proper trapezoid family). *A proper family of trapezoids $\mathcal{T}$ is strongly proper if for any two distinct trapezoids $T_1 = (I'_1, J'_1), T_2 = (I'_2, J'_2)$ in $\mathcal{T}$, if J'_1 and I'_2 clash then $l(J'_1) \leq l(I'_2) < r(I'_2) \leq r(J'_1)$, where $l(J'_1), r(J'_1)$ and $l(I'_2), r(I'_2)$ are the left and right endpoints of J'_1 and I'_2 respectively.*

Note that by the above definition, any pairwise disjoint family of trapezoids is proper, but the converse is not true. Thus, $\mathcal{T}' \subseteq \mathcal{T}$ computed at step 2 of $\{<, \between\}$-Approx is a proper trapezoid family. Also, computing a strongly proper subset $\mathcal{T}'' \subseteq \mathcal{T}'$ can be done easily by adjusting step 3 of $\{<, \between\}$-Approx. Instead of omitting all trapezoids clashing with the leftmost trapezoid in this iteration, we need only to omit a small subset of these trapezoids. More specifically, let $T_0 = (I'_0, J'_0)$ be the leftmost trapezoid in $\mathcal{T}'$. We only omit trapezoids $T_\alpha = (I'_\alpha, J'_\alpha) \in \mathcal{T}'$ with either $l(I'_\alpha) \leq l(J'_0) \leq r(I'_\alpha)$ or $l(I'_\alpha) \leq r(J'_0) \leq r(I'_\alpha)$ (or both). It is not difficult to see that we obtain a strongly proper trapezoid family $\mathcal{T}'' \subseteq \mathcal{T}'$ if we proceed in this fashion and that $|\mathcal{T}''| \geq \frac{1}{3}|\mathcal{T}'|$.

Definition 9 (Clashing trapezoid graph). *Given a family $\mathcal{T}$ of trapezoids, the clashing trapezoid graph of $\mathcal{T}$, denoted by $G_\mathcal{T}$, is the graph such that each vertex in $V(G_\mathcal{T})$ corresponds to a distinct trapezoid in $\mathcal{T}$, and two vertices are connected by an edge if and only if their corresponding trapezoids clash.*

Lemma 5. *Let $\mathcal{T}$ be a family of trapezoids. If $\mathcal{T}$ is strongly proper then $G_\mathcal{T}$ is a forest.*

Proof. Let $\mathcal{T}$ be a strongly proper family of trapezoids and let $G_\mathcal{T}$ be its corresponding clashing trapezoid graph. Define $G^*_\mathcal{T}$ as the directed graph obtained be orientating the edges of $G_\mathcal{T}$ according to the precedence relation of $\mathcal{T}$. In other words, $V(G^*_\mathcal{T}) = V(G_\mathcal{T})$ and $(T_1, T_2) \in E(G^*_\mathcal{T})$ if and only if $\{T_1, T_2\} \in E(G_\mathcal{T})$ and $T_1 < T_2$ in $\mathcal{T}$. Since $\mathcal{T}$ is strongly proper, every trapezoid in $\mathcal{T}$ clashes with at most one trapezoid on its left, and so the in-degree of every vertex $v \in V(G^*_\mathcal{T})$ is at most one. Hence, any cycle $(v_0, \ldots, v_t, v_0)$ in $G_\mathcal{T}$ is a (directed) cycle in $G^*_\mathcal{T}$. However, in such a case we must have $T_0 < T_t < T_0$ by definition of $G^*_\mathcal{T}$, which is clearly a contradiction. Hence, we conclude that $G^*_\mathcal{T}$, and consequently $G_\mathcal{T}$, contain no cycles, and the above lemma holds. □

It is well known that the maximum independent set in any forest G is of size at least $\frac{1}{2}|V(G)|$ and that this set can be found in linear time with respect to $|V(G)|$. Also, by definition, if $\mathcal{T}''$ is a pairwise disjoint family of trapezoids, then any independent set of $G_{\mathcal{T}''}$ corresponds to a pairwise disjoint non-clashing set of trapezoids, which by Observation 3, corresponds to a $\{<, \between\}$-comparable set of 2-intervals. A schematic description of our algorithm for unlimited 2-intervals sets, called Unl-$\{<, \between\}$-Approx, is given in Figure 6.

Lemma 6. *Algorithm Unl-$\{<, \between\}$-Approx is a 6-approximation algorithm for the* 2-Interval Pattern *problem over the $\{<, \between\}$ model.*

Proof. Let $\mathcal{D}$ be the input set of 2-intervals and let $\mathcal{T}(\mathcal{D})$, $\mathcal{T}'$ and $\mathcal{T}''$ be the trapezoid families as described in the above description of Unl-$\{<, \between\}$-Approx. Also, denote by OPT the maximum $\{<, \between\}$-comparable subset of $\mathcal{D}$. We have $|OPT| \leq |\mathcal{T}'|$ and $|\mathcal{T}'| \leq 3|\mathcal{T}''|$. Let $\alpha(G_{\mathcal{T}''})$ denote the size of the maximal independent set of $G_{\mathcal{T}''}$. Since $G_{\mathcal{T}''}$ is a forest, we have $|V(G_{\mathcal{T}''})| \leq 2\alpha(G_{\mathcal{T}''})$. Accumulating all these inequalities together we get: $|OPT| \leq |\mathcal{T}'| \leq 3|\mathcal{T}''| = 3|V(G_{\mathcal{T}''})| \leq 6\alpha(G_{\mathcal{T}''})$. Thus, the maximum independent set of $G_{\mathcal{T}''}$ is at least of size $\frac{1}{6}|OPT|$, and the promised approximation factor holds. □

Algorithm Unl-$\{<, \not\wr\}$-Approx($\mathcal{D}$)

Data : A set of 2-intervals $\mathcal{D}$.
Result : A $\{<, \not\wr\}$-comparable subset of $\mathcal{D}$.
begin
1. Construct $\mathcal{T}(\mathcal{D})$, the corresponding trapezoid set of $\mathcal{D}$.
2. Compute $\mathcal{T}'$, the maximum pairwise disjoint subset of $\mathcal{T}(\mathcal{D})$.
3. Compute $\mathcal{T}''$, a strongly proper subset of $\mathcal{T}'$, such that $|\mathcal{T}''| \geq \frac{1}{3}|\mathcal{T}'|$.
4. Compute $G_{\mathcal{T}''}$ and the maximum independent set of $G_{\mathcal{T}''}$.
return *the set of* 2*-intervals corresponding to the maximum independent set of* $G_{\mathcal{T}''}$.
end

Fig. 6. A schematic description of algorithm Unl-$\{<, \not\wr\}$-Approx.

Time complexity. Let $|\mathcal{D}| = n$. Steps 1-2 in Unl-$\{<, \not\wr\}$-Approx can be computed in $\mathcal{O}(n \lg n)$ time by a similar analysis of the time complexity of $\{<, \not\wr\}$-Approx. Step 3 can be computed straightforwardly in $\mathcal{O}(n^2)$ time. Finally, step 4 can be computed in $\mathcal{O}(n)$ time since $G_{\mathcal{T}''}$ is a forest. Thus, the whole algorithm can be implemented to run in $\mathcal{O}(n^2)$ time.

References

1. T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104:45-62, 2000.
2. R. Bar-Yehuda, M.M. Halldorsson, J. Naor, H. Shachnai and I. Shapira. Scheduling spit intervals. *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002)*, 732-741.
3. G. Blin, G. Fertin and S. Vialette. New results for the 2-interval pattern problem. *Proceedings of the 15th Annual Symposium on Combinatorial Pattern Matching (CPM 2004)*, Lecture Notes in Computer Science 3109, Springer-Verlag, 311-322.
4. I. Dagan, M.C. Golumbic and R.Y. Pinter. Trapezoid graphs and their coloring. *Discrete Applied Mathematics*, 21:35-46, 1988.
5. S. Felsner, R. Müller and L. Wernisch. Trapezoid graphs and generalizations: Geometry and algorithms. *Discrete Applied Mathematics*, 74:13-32, 1997.
6. F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1:180-187, 1972.
7. M.C. Golumbic. *Algorithmic graph theory and perfect graphs.* Academic Press, New York, 1980.
8. S. Ieong, M.Y. Kao, T.W. Lam, W.K. Sung and S.M. Yiu. Predicting RNA secondary structures with arbitrary pseudoknots by maximizing the number of stacking pairs. *Proceedings of the 2nd Symposium on Bioinformatics and Bioengineering (BIBE 2002)*, 183-190.
9. R.B. Lyngsø, C.N.S. Pedersen. RNA pseudoknot prediction in energy based models. *Journal of Computational Biology*, 7:409-428, 2000.
10. T.A. McKee, F.R. McMorris. *Topics in intersection graph theory.* SIAM monographs on discrete mathematics and applications, 1999.
11. S. Vialette. On the computational complexity of 2-interval pattern matching problems. *Theoretical Computer Science*, 312:335-379, 2004.