# Common Structured Patterns in Linear Graphs: Approximations and Combinatorics

Guillaume Fertin, Danny Hermelin, Romeo Rizzi, Stéphane Vialette

# Common Structured Patterns in Linear Graphs: Approximation and Combinatorics

Guillaume Fertin[1] ⋆, Danny Hermelin[2] ⋆⋆,
Romeo Rizzi[3] ⋆, and Stéphane Vialette[4] ⋆

[1] LINA, Univ. Nantes, 2 rue de la Houssinière, Nantes, France
Guillaume.Fertin@lina.univ-nantes.fr
[2] Dpt. Computer Science, Univ. Haifa, Mount Carmel, Haifa, Israel
danny@cri.haifa.ac.il
[3] DIMI, Univ. Udine, Udine, Italy
Romeo.Rizzi@dimi.uniud.it
[4] LRI, UMR 8623, Univ. Paris-Sud, Orsay, France
Stephane.Vialette@lri.fr

**Abstract.** A linear graph is a graph whose vertices are linearly ordered. This linear ordering allows pairs of disjoint edges to be either preceding ($<$), nesting ($\sqsubset$) or crossing ($\between$). Given a family of linear graphs, and a non-empty subset $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, we are interested in the MCSP problem: Find a maximum size edge-disjoint graph, with edge-pairs all comparable by one of the relations in $\mathcal{R}$, that occurs as a subgraph in each of the linear graphs of the family. In this paper, we generalize the framework of Davydov and Batzoglou by considering patterns comparable by all possible subsets $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$. This is motivated by the fact that many biological applications require considering crossing structures, and by the fact that different combinations of the relations above give rise to different generalizations of natural combinatorial problems. Our results can be summarized as follows: We give tight hardness results for the MCSP problem for $\{<, \between\}$-structured patterns and $\{\sqsubset, \between\}$-structured patterns. Furthermore, we prove that the problem is approximable within ratios: ($i$) $2\mathscr{H}(k)$ for $\{<, \between\}$-structured patterns, ($ii$) $k^{1/2}$ for $\{\sqsubset, \between\}$-structured patterns, and ($iii$) $\mathcal{O}(\sqrt{k \lg k})$ for $\{<, \sqsubset, \between\}$-structured patterns, where $k$ is the size of the optimal solution and $\mathscr{H}(k) = \sum_{i=1}^{k} 1/i$ is the $k$-th harmonic number.

## 1 Introduction

Many biological molecules such as RNA and proteins exhibit a three-dimensional structure that determines most of their functionality. This three dimensional structure can be modeled in two dimensions by an edge-disjoint linear graph, *i.e.*, a graph with linearly ordered vertices that are incident to exactly one edge. The

⋆ Partially supported by the French-Italian Galileo Project PAI 08484VH.
⋆⋆ Partially supported by the Caesarea Edmond Benjamin de Rothschild Foundation Institute (CRI).

corresponding structure-similarity or structure-prediction problems that arise in such contexts usually translate to finding common edge-disjoint subgraphs, or common *structured patterns*, that occur in a family of general linear graphs. Examples of such problems are Longest Common Subsequence [19, 20], Maximum Common Ordered Tree Inclusion [2, 8, 21], Arc-Preserving Subsequence [4, 14, 17], and Maximum Contact Map Overlap [15]. In this paper, we study a general framework for such problems which we call Maximum Common Structured Pattern (MCSP).

The MCSP problem was introduced by Davydov and Batzoglou [10] in the context of (non-coding) RNA secondary structure prediction via multiple structural alignment. There, an RNA sequence of $n$ nucleotides is represented by a linear graph with $n$ vertices, and an edge connects two vertices if and only if their corresponding nucleotides are complementary. A family of linear graphs is then used to represent a family of functionally-related RNAs, and a common structured pattern in such a family is considered to be a probably common secondary structure element of the family. The ordering amongst the vertices of a linear graph allows a pair of disjoint edges in the graph to be either preceding ($<$), nesting ($\sqsubset$), or crossing ($\between$). Since most RNA secondary structures translate to linear graphs with non-crossing edges, Davydov and Batzoglou [10] focused on the variant of MCSP where the common structured pattern is required to be non-crossing. However, there are known RNAs which have secondary structures that translate to linear graphs with a few edge-crossings (pseudo-knotted RNA secondary structures). Also, when predicting proteins rather than RNA structures, the non-crossing restriction becomes an even bigger limitation since the folding structures of proteins are often more complex than those of RNAs. In [16], it is argued that many important protein secondary structure elements like alpha helices and anti-parallel beta sheets exhibit $\{<, \between\}$-structured patterns, *i.e.* patterns which are non-nesting rather than non-crossing.

In this sequel, we present a framework which extends the work of [10] by considering different types of common structured patterns. Following [31], we consider structured patterns that are allowed to have crossing edges, and which might also be restricted to be non-nesting or non-preceding. More specifically, the MCSP problem receives as input a family of linear graphs and a non-empty subset $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, and the goal is to find a maximum common $\mathcal{R}$-structured pattern. We study the combinatorics behind the structures of these different types of patterns, with a focus on approximation algorithms for MCSP.

The paper is organized as follows. In the remaining part of this section we briefly review related work and notations that will be used throughout the paper. In Section 2, we discuss simple structured patterns (*i.e.* $R$-structured patterns, where $R \in \{<, \sqsubset, \between\}$) and $\{<, \sqsubset\}$-structured patterns. Following this, we discuss the more complex $\{<, \between\}$-structured patterns and $\{\sqsubset, \between\}$-structured patterns in Section 3 and Section 4 respectively. In Section 5, we deal with general structured patterns, *i.e.* $\{<, \sqsubset, \between\}$-structured patterns. An overview of the paper, along with some open problems, is given in Section 6.

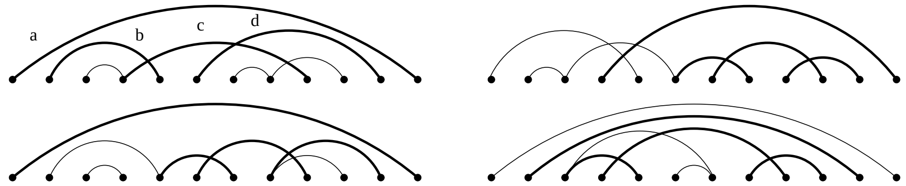


**Fig. 1.** Four linear graphs and a $\{<, \sqsubset, \between\}$-common structured pattern. The occurrence of the structured pattern in each graph is emphasized in bold. Edges **b**,**c**, and **d**, are nesting in edge **a**. Edge **b** precedes edge **d**, and they both cross edge **c**.

### 1.1 Related Work

There are many structural comparison problems that are closely related to MCSP. First, as mentioned previously, MCSP for $\{<, \sqsubset\}$-structured patterns has been studied by Davydov and Batzoglou in [10] under the name Maximum Common Nested Subgraph. Recently, new results concerning this problem appeared in [25]. We discuss the results of both these works in Section 2.

Closely related to MCSP are Arc-Preserving Subsequence [4, 14, 17], and Maximum Contact Map Overlap [15]. Both are concerned with finding maximum common subgraphs in a pair of linear graphs, except that in Arc-Preserving Subsequence the vertices of the linear graphs are assigned letters from some given alphabet, and an occurrence of a common subgraph in each of the linear graphs is required to preserve the letters, as well as their arc structure. Another closely related problem is Pattern Matching over 2-Interval Set [31], where one asks whether a structured pattern occurs in a given 2-interval set, which is a generalization of a linear graph. The 2-Interval Pattern problem [5, 9, 31] asks to find the maximum $\mathcal{R}$-structured pattern, for some given $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, in a single family of 2-interval sets.

There is a well-known bijective correspondence between $\{<, \sqsubset\}$-structured patterns and ordered forests – the nesting relation corresponds to the ancestor/predecessor relationship between the nodes, and the precedence relation corresponds to their order. Hence, MCSP for $\{<, \sqsubset\}$-structured patterns can be viewed as the problem of finding a tree which is included in all trees of a given tree family, the Maximum Common Ordered Tree Inclusion problem. Determining whether a tree is included in another is studied in [2, 8, 21]. Finding the maximum common tree included in a pair of trees can be done using the algorithms given in [22, 29]. The MCSP problem for $\{<, \sqsubset\}$-structured patterns has been studied in [10, 25]. We discuss the results there in Section 2.

Like $\{<, \sqsubset\}$-structured patterns, $\{\sqsubset, \between\}$-structured patterns also correspond to natural combinatorial objects, namely permutations (see Section 4). In [6], the authors studied the problem of determining whether a permutation-pattern occurs in a given permutation, the so called Pattern Matching for Permutations problem. This problem corresponds to determining whether a $\{\sqsubset, \between\}$-structured pattern is a subpattern of another $\{\sqsubset, \between\}$-structured pattern. Bose, Buss, and Lubiw proved that Pattern Matching for Permutations is **NP**-complete [6].

Determining whether a given $\{<, \between\}$-structured pattern occurs in a general linear graph has been studied in [16, 26]. Gramm [16] gave a polynomial-time algorithm for this problem. Recently, Li and Li [26] proved that this algorithm was incorrect and showed the problem was in fact **NP**-complete. Prior to this, Blin *et al.* [5] proved that a generalization of this problem, where the linear graph is replaced by a 2-interval set, is **NP**-complete. Finally, probably the oldest and most famous problem related to MCSP is the Longest Common Subsequence (LCS) [19, 20] problem, where one wishes to find the longest common subsequence in two or more sequences. Important developments of the initial algorithms of [19, 20] can be found in [3, 12, 28]. Maier [27] proved that the LCS problem for multiple sequences is **NP**-hard.

### 1.2 Terminology and basic definitions

For a graph $G$, we denote $V(G)$ as the set of vertices and $E(G)$ as the set of edges. The *order* and the *size* of $G$ stand for $|V(G)|$ and $|E(G)|$, respectively. A *linear graph* of order $n$ is a vertex-labeled graph where each vertex is labeled by a distinct label from $\{1, 2, \ldots, n\}$. Thus, it can be viewed as a graph with vertices embedded on the integral line, yielding a total order amongst them. In case of linear graphs, we write an edge between vertices $i$ and $j$, $i < j$, as the pair $(i, j)$. Two edges of a linear graph are *disjoint* if they do not share a common vertex. A linear graph $G$ is said to be *edge-disjoint* if it is composed of disjoint edges, *i.e.* if $G$ is a matching. Of particular interest are the relations between pairs of disjoint edges [31]: Let $e = (i, j)$ and $e' = (i', j')$ be two disjoint edges in a linear graph $G$; we write (i) $e < e'$ ($e$ *precedes* $e'$) if $i < j < i' < j'$, (ii) $e \sqsubset e'$ ($e$ is *nested* in $e'$) if $i' < i < j < j'$ and (iii) $e \between e'$ ($e$ and $e'$ *cross*) if $i < i' < j < j'$.

Two edges $e$ and $e'$ are *R-comparable*, for some $R \in \{<, \sqsubset, \between\}$, if $eRe'$ or $e'Re$. For a subset $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, $\mathcal{R} \neq \emptyset$, $e$ and $e'$ are said to be $\mathcal{R}$*-comparable* if $e$ and $e'$ are $R$-comparable for some $R \in \mathcal{R}$. A set of edges $E$ (or a linear graph $G$ with $E(G) = E$) is $\mathcal{R}$*-comparable* if any pair of distinct edges $e, e' \in E$ are $\mathcal{R}$-comparable. A *subgraph* of a linear graph $G$ is a linear graph $H$ which can be obtained from $G$ by a series of vertex and edge deletions, where a deletion of vertex $i$ results in removing vertex $i$ and all edges incident to it from the graph, and then relabeling all vertices $j$ with $j > i$ to $j - 1$. An edge-disjoint subgraph of a linear graph is called a *structured-pattern*. For a family of linear graphs $\mathcal{G} = G_1, \ldots, G_n$, a *common structured pattern* of $\mathcal{G}$ is an edge-disjoint linear graph $H$ that is a subgraph of $G_i$, for all $1 \leq i \leq n$. Following the above notation, $H$ is called an $\mathcal{R}$*-structured pattern*, for some non-empty $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, if $E(H)$ is $\mathcal{R}$*-comparable*.

**Definition 1.** *Given a family of linear graphs $\mathcal{G} = G_1, \ldots, G_n$ and a subset $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, $\mathcal{R} \neq \emptyset$, the* Maximum Common Structured Pattern (MCSP) *problem asks to find a maximum-size common $\mathcal{R}$-structured pattern of $\mathcal{G}$.*

We will use the following terminology to describe special edge-disjoint linear graphs. A linear graph is called a *sequence* if it is $\{<\}$-comparable, a *tower* if it is $\{\sqsubset\}$-comparable, and a *staircase* if it is $\{\between\}$-comparable. We define the *width* (resp. *height* and *depth*) of a linear graph to be the size of the maximum cardinality sequence (resp. tower and staircase) subgraph of the graph. A $\{<, \sqsubset\}$-comparable linear graph with the additional property that any two maximal towers in it do not share an edge is called a *sequence of towers*. Similarly, a $\{<, \between\}$-comparable linear graph is a *sequence of staircases* if any two maximal staircases do not share an edge. A *tower of staircases* is a $\{\sqsubset, \between\}$-comparable linear graph where any pair of maximal staircases do not share an edge, and a *staircase of towers* is a $\{\sqsubset, \between\}$-comparable linear graph where any pair of maximal towers do not share an edge. A sequence of towers (resp. sequence of staircases, tower of staircases, and staircase of towers) is *balanced* if all of its maximal towers (resp. staircases, staircases, and towers) are of equal size. Figure 2 illustrates an example of the above types of linear graphs.

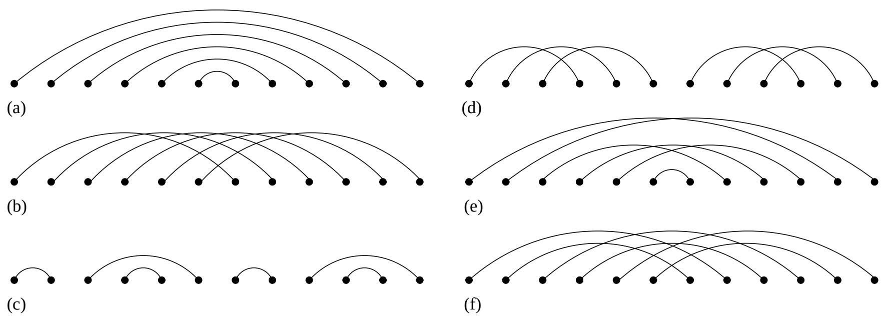


**Fig. 2.** Examples of restricted edge-disjoint linear graphs: (a) a tower of height 6, (b) a staircase of depth 6, (c) a sequence of towers of width 4 and height 2, (d) a balanced sequence of staircases of width 2 and depth 3, (e) a tower of staircases of height 3 and depth 3 and (f) a balanced staircase of towers of height 2 and depth 3.

## 2 Simple and $\{<, \sqsubset\}$-Structured Patterns

A structured pattern is simple if it is an $R$-structured pattern for a single relation $R \in \{<, \sqsubset, \between\}$. We begin our study by considering the MCSP problem for simple structured patterns, and for $\{<, \sqsubset\}$-structured patterns. We first discuss the analogy between the relations we defined for disjoint edges in a linear graph, and well-studied relations defined for families of intervals. We show that known algorithms on interval families can be used to solve MCSP for simple structured patterns in polynomial-time. Following this, we discuss results presented in [10, 25] for MCSP for $\{<, \sqsubset\}$-structured patterns.

For a given linear graph $G$ of size $m$, let $\mathcal{I}(G) = \{[i, j] \,|\, (i, j) \in E(G)\}$ be the family of intervals obtained by considering each edge of $G$ as an interval of the line, closed between both its endpoints. A pair of $\{<\}$-comparable edges in $E(G)$ correspond to a pair of disjoint intervals in $\mathcal{I}(G)$, a pair of $\{\sqsubset\}$-comparable edges correspond to a pair of nesting intervals, and a pair of $\{\between\}$-comparable edges correspond to a pair of overlapping intervals. Note that this correspondence is bi-directional only if $G$ is edge-disjoint, since a pair of edges sharing a vertex can correspond to a pair of nesting or overlapping intervals. Nevertheless, we can always modify $\mathcal{I}(G)$ in such a way, so that all intervals have unique endpoints, and so that any pair of intervals who shared an endpoint now become non-nesting (resp. non-overlapping). A maximum pairwise disjoint subset of intervals can be computed in linear time using standard dynamic-programming, assuming the interval family is given in a sorted manner [18] (which we can provide in linear time in our case using bucket sorting). A maximum pairwise nesting subset can be computed in $\mathcal{O}(m \lg\lg m)$ in an interval family of $m$ intervals (see for example the algorithm in [7]), and a maximum pairwise overlapping subset in $\mathcal{O}(m^{1.5})$ time [30].

**Lemma 1.** *Let $G$ be a linear graph of size $m$. Then there exists a $\mathcal{O}(m)$ (resp. $\mathcal{O}(m \lg\lg m)$ and $\mathcal{O}(m^{1.5})$) time algorithm for finding the largest $\{<\}$-comparable (resp. $\{\sqsubset\}$-comparable and $\{\between\}$-comparable) subgraph of $G$.*

**Theorem 1.** *The* MCSP *problem for $\{<\}$-structured patterns (resp. $\{\sqsubset\}$-structured patterns and $\{\between\}$-structured patterns) is solvable in $\mathcal{O}(nm)$ (resp. $\mathcal{O}(nm \lg\lg m)$ and $\mathcal{O}(nm^{1.5})$) time, where $n = |\mathcal{G}|$ and $m = \max_{G \in \mathcal{G}} |E(G)|$.*

We next consider $\{<, \sqsubset\}$-structured patterns. The MCSP problem for this type of patterns was considered by [10, 25], in the context of multiple RNA structural alignment.

**Theorem 2 ([25]).** *The* MCSP *problem for $\{<, \sqsubset\}$-structured patterns is* **NP**-*hard even if each input linear graph is a sequence of towers of height at most* 2.

Note, however, that the problem MCSP is polynomial-time solvable in case the number of input linear graphs is a constant [25]. Also, it is proven in [25] that MCSP for $\{<, \sqsubset\}$-structured patterns is approximable with ratio $\lg k + 1$, where $k$ is the size of the optimal solution.

**Theorem 3 ([25]).** *The* MCSP *problem for $\{<, \sqsubset\}$-structured patterns is approximable within ratio $\mathcal{O}(\lg k)$ in $\mathcal{O}(nm^2)$ time, where $k$ is the size of an optimal solution, $n = |\mathcal{G}|$, and $m$ is the maximum size of any graph in $\mathcal{G}$.*

## 3 $\{<, \between\}$-Structured Patterns

We now turn to consider MCSP for $\{<, \between\}$-structured patterns. We begin by proving a tight hardness result for the problem. Following this, we present an approximation algorithm for the problem which achieves a ratio of $2\mathscr{H}(k)$ in $\mathcal{O}(nm^3 \log^2 m)$ time, where $k$ is the size of an optimal solution, $\mathscr{H}(k) = \sum_{i=1}^{k} 1/i$, $n = |\mathcal{G}|$, and $m$ is the maximum size of any graph in $\mathcal{G}$.

**Theorem 4.** *The* MCSP *problem for $\{<, \between\}$-structured patterns is* **NP**-*hard even if each input linear graph is a sequence of staircases of depth at most* 2.

A recent result [26] implies that MCSP for $\{<, \between\}$-structured patterns is hard even if $\mathcal{G}$ consists of only two graphs of unlimited structure. We next show that one can approximate the maximum common $\{<, \between\}$-structured pattern of $\mathcal{G}$ within ratio $2\mathscr{H}(k)$. The first ingredient of our proof is to observe that every $\{<, \between\}$-structured pattern contains a sequence of staircases of substantial size. The second ingredient consists in showing that any sequence of staircases contains a balanced subgraph of substantial size.

**Lemma 2.** *Let $H$ be a $\{<, \between\}$-comparable linear graph. There exists a partition $E(H) = E_{RED} \cup E_{BLUE}$ such that both $H[E_{RED}]$ and $H[E_{BLUE}]$ are sequences of staircases.*

**Lemma 3.** *Let $H$ be a sequence of staircases of size $k$. Then $H$ contains a balanced sequence of staircases with at least $\frac{k}{\mathscr{H}(k)}$ edges.*

As a direct corollary of Lemmas 2 and 3, we obtain:

**Corollary 1.** *Any $\{<, \between\}$-comparable linear graph of size $k$ contains as a subgraph a balanced sequence of staircases of size at least $\frac{k}{2\mathscr{H}(k)}$.*

What is left is to show that, given a set of linear graphs, one can find in polynomial-time the size of the largest balanced sequence of staircases that occurs in each input linear graph. For this particular purpose, we present Algorithm Bal-Seq-Staircase in Figure 3.

---

Algorithm Bal-Seq-Staircase($G, w, d$).

---

**Data** : A linear graph $G$ of size $m$, and two positive integers $d$ and $w$.
**Result** : `true` iff $G$ contains a balanced sequence of staircases of width $w$ and depth $d$.
**begin**
  **1.** $E' \leftarrow \emptyset$
  **2.** **for** $i = 1, 2, \ldots, m-1$ **do**
    **(a)** Let $j$ be the smallest integer such that $G[i, \ldots, j]$ contains as a subgraph a staircase of size $d$ (set $j = \infty$ if no such integer exists).
    **(b)** **if** $j \neq \infty$ **then** $E' \leftarrow E' \cup \{(i, j)\}$.
  **end**
  **3.** Compute $H$, the maximum $\{<\}$-comparable subgraph of $G' = (V(G), E')$.
  **4.** **if** $|E(H)| \geq w$ **then return** `true` **else return** `false`.
**end**

---

**Fig. 3.** Algorithm Bal-Seq-Staircase for finding a balanced sequence of staircases of width $w$ and depth $d$ in a linear graph. For a linear graph $G \in \mathcal{G}$, and two integers $i$ and $j$ with $1 \leq i < j \leq |V(G)|$, $G[i, \ldots, j]$ stands for the subgraph of $G$ obtained by deleting all vertices labeled $k$ with $k < i$ or $j < k$.

**Lemma 4.** *Algorithm* Bal-Seq-Staircase$(G, w, d)$ *runs in* $\mathcal{O}(m^{2.5} \log m)$ *time and returns* `true` *if and only if* $G$ *contains a balanced sequence of staircases of width* $w$ *and depth* $d$*.*

**Theorem 5.** *The* MCSP *problem for* $\{<, \between\}$*-structured patterns is approximable within ratio* $2\mathscr{H}(k)$ *in* $\mathcal{O}(nm^{2.5} \log^2 m)$ *time, where* $k$ *is the size of an optimal solution,* $n = |\mathcal{G}|$*, and* $m$ *is the maximum size of any graph in* $\mathcal{G}$*.*

## 4 $\{\sqsubset, \between\}$-Structured Patterns

We next consider $\{\sqsubset, \between\}$-structured patterns. We begin by proving a hardness result, analogous to Theorem 4, which states that MCSP for $\{\sqsubset, \between\}$-structured patterns is **NP**-hard even if the input consists of towers of staircases of depth at most 2. However, unlike the approach we used for $\{<, \between\}$-structured patterns, we cannot use towers of staircases to obtain very good approximations of maximum common $\{\sqsubset, \between\}$-structured patterns. We show that there exists a $\{\sqsubset, \between\}$-comparable linear graph of size $k$ which does not contain a tower of staircases of size $\varepsilon\sqrt{k}$ for some constant $\varepsilon$. On the other hand, such a graph must contain either a tower or a staircase with at least $\sqrt{k}$ edges.

**Theorem 6.** *The* MCSP *problem for* $\{\sqsubset, \between\}$*-structured patterns is* **NP***-hard even if each input linear graph is a tower of staircases of depth at most* 2*.*

We now turn to approximating MCSP for $\{\sqsubset, \between\}$-structured patterns. First, let us observe the one-to-one correspondence between $\{\sqsubset, \between\}$-structured patterns and permutations. Let $H$ be a $\{\sqsubset, \between\}$-comparable linear graph of size $k$. Then the vertices in $H$ which are left endpoints of edges are labeled $\{1, \ldots, k\}$ and the right endpoints are labeled $\{k+1, \ldots, 2k\}$. The permutation $\pi_H$ corresponding to $H$ is defined by $\pi_H(i) = j - k \iff (i, j) \in E(H)$. Clearly, all $\{\sqsubset, \between\}$-comparable linear graphs have corresponding permutations, and vice versa. It follows from this bijective correspondence, that the number of different $\{\sqsubset, \between\}$-comparable linear graphs of size $k$ is exactly $k!$. Moreover, notice that increasing subsequences in $\pi_H$ correspond to $\{\between\}$-comparable subgraphs of $H$, while decreasing subsequences correspond to $\{\sqsubset\}$-comparable subgraphs. The well known Erdős-Szekeres Theorem [13] states that any permutation on $1, \ldots, k$ contains either an increasing or a decreasing subsequence of size at least $\sqrt{k}$ (see also Lemma 6). Hence, using the algorithms in Lemma 1 for finding the maximum common $\{\sqsubset\}$-structured $\{\between\}$-structured patterns, we obtain the following theorem:

**Theorem 7.** *The* MCSP *problem for model* $\mathcal{M} = \{\sqsubset, \between\}$ *is approximable within ratio* $k^{1/2}$ *in* $\mathcal{O}(nm^{1.5})$ *time, where* $k$ *is the size of an optimal solution* $n = |\mathcal{G}|$*, and* $m = \max_{G \in \mathcal{G}} |E(G)|$*.*

Alon [1] recently showed that towers of staircases cannot be used to obtain a much better approximation algorithm than the one proposed above. To see this, let us count the number of different towers of staircases with $k$ edges. Note that the number of towers of staircases of size $k$ and of height $h$, is exactly the number of different partitions of $\{1,\ldots,k\}$ into $h$ consecutive intervals, *i.e.* $\binom{k}{h-1}$. Hence the total number of towers of staircases of size $k$ equals $\sum_{h=1}^{k}\binom{k}{h-1} = 2^k - 1 < 2^k$. Using this simple observation, the following lemma can be proved.

**Lemma 5 ([1]).** *There exists a $\{\sqsubset, \between\}$-comparable linear graph of size $K = \Omega(k^2)$ which does not contain a tower of staircases of size $k$.*

## 5 General Structured Patterns

In this section we consider MCSP for general, *i.e.*, $\{<, \sqsubset, \between\}$, structured patterns. Since $\{<, \sqsubset, \between\}$-structured patterns generalize all other types of patterns, all hardness results presented in previous sections apply for general structured patterns as well. We present three approximation algorithms with increasing time complexities and decreasing approximation ratios.

Observe that both $<$ and $\sqsubset$ induce partial orders on the edges of a given linear graph. Recall now that a *chain* (resp. *anti-chain*) in a partial order is a subset of pairwise comparable (resp. incomparable) elements. Dilworth's Theorem [11] states that in any partial order, the size of the maximum chain equals the size of the minimum anti-chain partitioning. Therefore, in any partial order on $k$ elements, the size of the maximum chain multiplied by the size of the maximum anti-chain is at least $k$. The following lemma states this property in our terms.

**Lemma 6.** *Let $H$ be a $\{<, \sqsubset, \between\}$-comparable linear graph of size $k$, width $w(H)$, and height $h(H)$. Also, let $hd(H)$ and $wd(H)$ be the sizes of the maximum $\{\sqsubset, \between\}$-comparable and $\{<, \between\}$-comparable subsets of $E(H)$. Then $k \le w(H) \cdot hd(H)$ and $k \le h(H) \cdot wd(H)$.*

An immediate consequence of Lemma 6 is as follows.

**Lemma 7.** *Let $H$ be a $\{<, \sqsubset, \between\}$-comparable linear graph of size $k$. Then $H$ contains a simple structured pattern of size at least $k^{1/3}$.*

Combining the lemma above with the fact that a maximum common simple structured pattern of $\mathcal{G}$ can be found in $\mathcal{O}(nm^{1.5})$ time (Theorem 1), we obtain our first approximation algorithm for general structured patterns.

**Theorem 8.** *The* MCSP *problem for $\{<, \sqsubset, \between\}$-structured patterns is approximable within ratio $\mathcal{O}(k^{2/3})$ in $\mathcal{O}(nm^{1.5})$ time, where $k$ is the size of an optimal solution, $n = |\mathcal{G}|$, and $m = \max_{G \in \mathcal{G}} |E(G)|$.*

It is easily seen that Lemma 7 is tight. One way to obtain an extremal example of this is as follows: Take $k^{1/3}$ balanced towers of staircases, each one of depth $k^{1/3}$ and height $k^{1/3}$, and concatenate them one next to the other into one supergraph of size $k$, reassigning labels accordingly.

**Lemma 8.** *Let $k$ be an integer such that $k^{1/3}$ is also integer. Then there exists an $\{<, \sqsubset, \between\}$-comparable linear graph of size $k$ that does not contain a simple structured pattern of size $\varepsilon\, k^{1/3}$ for any $\varepsilon > 1$.*

Dilworth's theorem does not apply on the crossing relation since it is not transitive. However, an analogous result proven in [23] (see also [24]) implies that for any $\{<, \sqsubset, \between\}$-comparable linear graph $H$, $|E(H)| = \mathcal{O}(d \cdot wh \lg wh)$, where $d$ and $wh$ are sizes of the maximum $\{\between\}$-comparable and $\{<, \sqsubset\}$-comparable subsets of $E(H)$. This yields the following analogous of Lemma 6.

**Lemma 9.** *Let $H$ be a $\{<, \sqsubset, \between\}$-comparable linear graph of size $k$. Then $H$ contains a subgraph of size $\Omega(\sqrt{k/\lg k})$ which is either $\{<, \sqsubset\}$-comparable or $\{\between\}$-comparable.*

Using Lemma 9, the algorithm for finding a maximum structured pattern given in Theorem 1, and the $\mathcal{O}(\lg k)$-approximation algorithm for $\{<, \sqsubset\}$-structured patterns given in Theorem 3, we obtain our second approximation algorithm.

**Theorem 9.** *The* MCSP *problem for $\{<, \sqsubset, \between\}$-structured patterns is approximable within ratio $\mathcal{O}(\sqrt{k \lg^3 k})$ in $\mathcal{O}(nm^2)$ time.*

For our third algorithm, we show that any $\{<, \sqsubset, \between\}$-comparable linear graph contains a subgraph of sufficient size that is either a tower or a balanced sequence of staircases.

**Lemma 10.** *Let $H$ be a $\{<, \sqsubset, \between\}$-comparable linear graph of size $k$. Then $H$ contains either a tower or a balanced sequence of staircases of size $\Omega(\sqrt{k/\lg k})$.*

Applying Lemma 3 and the algorithms for finding the maximum common tower and balanced sequence of staircases in $\mathcal{G}$ given in Theorems 1 and 5, respectively, we obtain the following theorem.

**Theorem 10.** *The* MCSP *problem for $\{<, \sqsubset, \between\}$-structured patterns is approximable within ratio $\mathcal{O}(\sqrt{k \lg k})$ in $\mathcal{O}(nm^{2.5} \lg^2 m)$ time.*

We next consider subgraphs of $\{<, \sqsubset, \between\}$-comparable linear graphs that are comparable by pairs of relations. We show that any $\{<, \sqsubset, \between\}$-comparable linear graph of size $k$ contains such a subgraph of size at least $m^{2/3}$, and that this lower bound is relatively tight. Unfortunately, this result can not be applied for approximation purposes (approximating MCSP for $\{\sqsubset, \between\}$-patterns remains the bottleneck). Nevertheless, we present this result on account of independent interest.

**Lemma 11.** *Let $H$ be a $\{<, \sqsubset, \between\}$-comparable graph of size $k$. Then $H$ has a subgraph of size $\varepsilon\, k^{2/3}$, where $\varepsilon = \frac{\sqrt{17}-1}{8}$, which is either $\{<, \sqsubset\}$-comparable, $\{<, \between\}$-comparable, or $\{\sqsubset, \between\}$-comparable.*

We believe the bound of Lemma 11 to be not the best possible. However, combining Lemmas 6 and 8, we show that the above lemma is relatively tight.

**Lemma 12.** *Let $k$ be an integer such that $k^{1/3}$ is integer. Then there exists a $\{<, \sqsubset, \between\}$-comparable linear graph of size $k$ that contains neither a $\{<, \sqsubset\}$-comparable subgraph, nor a $\{<, \between\}$-comparable subgraph, nor a $\{\sqsubset, \between\}$-comparable subgraph of size least $\varepsilon\, k^{2/3}$ for any $\varepsilon > 1$.*

## 6 Discussion and Open Problems

In this paper we introduced MCSP as a general framework for many structure-comparison and structure-prediction problems, that occur mainly in computational molecular biology. Our framework followed the approach in [31] by analyzing all types of $\mathcal{R}$-structured patterns, $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$. We gave tight hardness results for finding maximum common $\{<, \between\}$-structured patterns and maximum common $\{<, \between\}$-structured patterns. We also proved that MCSP is approximable within ratio: (*i*) $2\mathscr{H}(k)$ for $\{<, \between\}$-structured patterns, (*ii*) $k^{1/2}$ for $\{\sqsubset, \between\}$-structured patterns, and (*iii*) $\mathcal{O}(\sqrt{k \lg k})$ for $\{<, \sqsubset, \between\}$-structured patterns.

There are many questions left open by our study. Below we list some of them. According to Lemma 11, we could improve in terms of approximation ratio on all the algorithms suggested for general structured patterns, if we had a better approximation algorithm for $\{\sqsubset, \between\}$-structured patterns. Is there an approximation algorithm which achieves a better ratio then the simple $\sqrt{k}$ algorithm? On the same note, can lower bounds on the approximation factor of MCSP for $\{<, \sqsubset, \between\}$-structured patterns or $\{\sqsubset, \between\}$-structured patterns be proven? How about $\{<, \sqsubset\}$-structured patterns or $\{<, \between\}$-structured patterns?

## References


1. N. Alon. Private communication, 2006.
2. L. Alonso and R. Schott. On the tree inclusion problem. In *Proc. 18th Mathematical Foundations of Computer Science (MFCS)*, volume 711 of *LNCS*, pages 211–221, 1993.
3. A. Apostolico and C. Guerra. The longest common subsequence problem revisited. *Algorithmica*, 2:315–336, 1987.
4. G. Blin, G. Fertin, R. Rizzi, and S. Vialette. What makes the arc-preserving subsequence problem hard ? In *Proc. 5th International Conference on Computational Science (ICCS)*, volume 3515 of *LNCS*, pages 860–868, 2005.
5. G. Blin, G. Fertin, and S. Vialette. New results for the 2-interval pattern problem. In *Proc. 15th Combinatorial Pattern Matching (CPM)*, volume 3109, 2004.
6. P. Bose, J.F. Buss, and A. Lubiw. Pattern matching for permutations. *IPL*, 65(5):277–283, 1998.
7. M.-S. Chang and F.-G. Wang. Efficient algorithms for the maximum weight clique and maximum weight independent set problems on permutation graphs. *IPL*, 43(6):293–295, 1992.
8. W. Chen. More efficient algorithm for ordered tree inclusion. *J. Algorithms*, 26(2):370–385, 1998.
9. M. Crochemore, D. Hermelin, G.M. Landau, and S. Vialette. Approximating the 2-interval pattern problem. In *Proc. 13th European Symposium on Algorithms (ESA)*, volume 3669 of *LNCS*, pages 426–437, 2005.
10. E. Davydov and S. Batzoglou. A computational model for RNA multiple structural alignment. In *Proc. 15th Combinatorial Pattern Matching (CPM)*, volume 3109 of *LNCS*, pages 254–269, 2004.
11. R.P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics Series 2*, 51:161–166, 1950.

12. D. Eppstein, Z. Galil, R. Giancarlo, and G.F. Italiano. Sparse dynamic programming I: Linear cost functions. *J. ACM*, 39(3):519–545, 1992.
13. P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.
14. P.A. Evans. *Algorithms and complexity for annotated sequence analysis*. PhD thesis, University of Alberta, 1999.
15. D. Goldman, S. Istrail, and C.H. Papadimitriou. Algorithmic aspects of protein structure similarity. In *Proc. 40th Foundations of Computer Science (FOCS)*, pages 512–522, 1999.
16. J. Gramm. A polynomial-time algorithm for the matching of crossing contact-map patterns. *IEEE/ACM Trans. Comp. Biol. and Bioinfo.*, 1(4):171–180, 2004.
17. J. Gramm, J. Guo, and R. Niedermeier. Pattern matching for arc-annotated sequences. In *Proc. 22nd Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, LNCS, pages 182–193, 2002.
18. U.I. Gupta, D.T. Lee, and J.Y-T. Leung. Efficient algorithms for interval graph and circular-arc graphs. *Networks*, 12:459–467, 1982.
19. D.S. Hirschberg. Algorithms for the longest common subsequence problem. *J. ACM*, 24(4):664–675, 1977.
20. J.W. Hunt and T.G. Szymanski. A fast algorithm for computing longest common subsequences. *Communications of the ACM*, 20:350–353, 1977.
21. P. Kilpeläinen and H. Mannila. Ordered and unordered tree inclusion. *SIAM J. Comp.*, 24(2):340–356, 1995.
22. P.N. Klein. Computing the edit-distance between unrooted ordered trees. In *Proc. 6th European Symposium on Algorithms (ESA)*, volume 1461 of *LNCS*, pages 91–102, 1998.
23. A. Kostochka. On upper bounds on the chromatic numbers of graphs. *Transactions of the Institute of Mathematics (Siberian Branch of the Academy of Sciences in USSR)*, 10:204–226, 1988.
24. A. Kostochka and J. Kratochvil. Covering and coloring polygon-circle graphs. *Discrete Mathematics*, 163:299–305, 1997.
25. M. Kubica, R. Rizzi, S. Vialette, and T. Waleń. Approximation of RNA multiple structural alignment. In *Proc. 17th Combinatorial Pattern Matching (CPM)*, volume 4009 of *LNCS*, pages 211–222, 2006.
26. S.C. Li and M. Li. On the complexity of the crossing contact map pattern matching problem. In *Proc. 6th Workshop on Algorithms in Bioinformatics (WABI)*, volume 4175, pages 231–241, 2006.
27. D. Maier. The complexity of some problems on subsequences and supersequences. *J. ACM*, 25(2):322–336, 1978.
28. W.J. Masek and M.S. Paterson. A faster algorithm computing string edit distances. *J. Comp. and Syst. Sc.*, 20(1):18–31, 1980.
29. D. Shasha and K. Zhang. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comp.*, 18(6):1245–1262, 1989.
30. A. Tiskin. Longest common subsequences in permutations and maximum cliques in circle graphs. In *Proc. 17th Combinatorial Pattern Matching (CPM)*, volume 4009 of *LNCS*, pages 270–281, 2006.
31. S. Vialette. On the computational complexity of 2-interval pattern matching problems. *Theoretical Computer Science*, 312(2-3):223–249, 2004.