Longest Common Separable Pattern between Permutations

Mathilde Bouvel, Dominique Rossin, Stéphane Vialette

LONGEST COMMON SEPARABLE PATTERN BETWEEN PERMUTATIONS

MATHILDE BOUVEL, DOMINIQUE ROSSIN, AND STÉPHANE VIALETTE

Abstract. In this article, we study the problem of finding the longest common separable pattern between several permutations. We give a polynomial-time algorithm when the number of input permutations is fixed and show that the problem is NP-hard for an arbitrary number of input permutations even if these permutations are separable.

On the other hand, we show that the NP-hard problem of finding the longest common pattern between two permutations cannot be approximated better than within a ratio of $\sqrt{Opt}$ (where Opt is the size of an optimal solution) when taking common patterns belonging to pattern-avoiding classes of permutations.

1. Introduction and basic definitions

A permutation π is said to be a pattern within a permutation σ if σ has a subsequence that is order-isomorphic to π. Within the last few years, the study of the *pattern containment* relation on permutations has become a very active area of research in both combinatorics and computer science. In combinatorics, much research focused on closed classes of permutations, *i.e.*, permutations that are closed downwards under forming subpermutations. A huge literature is devoted to this subject. To cite only a few of a plethora of suitable examples, Knuth considered permutations that do not contain the pattern 312 [15], Lovàsz considered permutations that do not contain the pattern 213 [17] and Rotem those that do not contain 231 nor 312 [19].

Surprisingly enough, there is considerably less research on algorithmic aspects of pattern involvement. Actually, it appears to be a difficult problem to decide whether a permutation occurs as a pattern in another permutation. Indeed, the problem in this general version is NP-complete [5]. The case of *separable patterns*, *i.e.*, permutations that contain neither the subpattern 3142 nor 2413, was proved to be solvable in $\mathcal{O}(kn^6)$ time and $\mathcal{O}(kn^4)$ space in [5], where k is the length of the pattern and n is the length of the target permutation. The design of efficient algorithms for the recognition of a fixed pattern in a permutation is considered in [2] and in particular a $\mathcal{O}(n^5 \log n)$ time algorithm is given for finding separable patterns. L. Ibarra subsequently improved the complexity for separable patterns to $\mathcal{O}(kn^4)$ time and $\mathcal{O}(kn^3)$ space in [14]. Beside separable patterns, only a few restricted cases were considered. A $\mathcal{O}(n \log \log n)$ time algorithm is presented in [10] for finding the longest increasing or decreasing subpermutation of a permutation of length n.

In the present paper we continue this line of research on separable patterns by considering the problem of finding a maximum length common separable pattern to a set of permutations, *i.e.*, given a set of permutations, find a longest separable permutation that occurs as a pattern in each input permutation. Of particular importance in this context, we do not impose here the input permutations to be separable.

This paper is organized as follows. In the remainder of Section 1, we briefly discuss basic notations and definitions that we will use throughout. In Section 2, we give a polynomial-time algorithm for finding the largest common separable pattern that appears as a pattern in a fixed number of permutations. Section 3 is devoted to proving hardness of the problem. Finally, some inapproximation issues are presented in Section 4.

1.1. **Permutations.** A permutation $\sigma \in S_n$ is a bijective map from $[1..n]$ to itself. The integer n is called the *length* of σ. We denote by σ_i the image of i under σ. A permutation can be seen as a word $\sigma_1\sigma_2\dots\sigma_n$ containing exactly once each letter $i \in [1..n]$. For each entry σ_i of a permutation σ, we call i its *index* and σ_i its *value.*

Definition 1 (Pattern in a permutation)**.** *A permutation $\pi \in S_k$ is a* pattern *of a permutation $\sigma \in S_n$ if there is a subsequence of σ which is order-isomorphic to π; in other words, if there is a subsequence $\sigma_{i_1}\sigma_{i_2}\dots\sigma_{i_k}$ of σ (with $1 \le i_1 < i_2 < \dots < i_k \le n$) such that $\sigma_{i_\ell} < \sigma_{i_m}$ whenever $\pi_\ell < \pi_m$.*
We also say that π is involved *in σ and call $\sigma_{i_1}\sigma_{i_2}\dots\sigma_{i_k}$ an* occurrence *of π in σ.*

A permutation σ that does not contain π as a pattern is said to *avoid* π. Classes of permutations of interest are the *pattern-avoiding classes of permutations*: the class of all permutations avoiding the patterns $\pi_1, \pi_2 \dots \pi_k$ is denoted $S(\pi_1, \pi_2, \dots, \pi_k)$, and $S_n(\pi_1, \pi_2, \dots, \pi_k)$ denotes the set of permutations of length n avoiding $\pi_1, \pi_2, \dots, \pi_k$.

Example 1. *For example $\sigma = 142563$ contains the pattern 1342, and 1563, 1463, 2563 and 1453 are the occurrences of this pattern in σ. But $\sigma \in S(321)$: σ avoids the pattern 321 as no subsequence of length 3 of σ is isomorphic to 321, i.e., is decreasing.*

If a pattern π has an occurrence $\sigma_{i_1}\sigma_{i_2}\dots\sigma_{i_k}$ in a permutation σ of length n, let I and V be two subintervals of $[1..n]$ such that $\{i_1, i_2, \dots, i_k\} \subseteq I$ and $\{\sigma_{i_1}, \sigma_{i_2}, \dots, \sigma_{i_k}\} \subseteq V$; then we say that π has an occurrence in σ in the intervals I of indices and V of values, or that π is a pattern of σ using the intervals I of indices and V of values in σ.

Among the pattern-avoiding classes of permutations, we are particularly interested here in the separable permutations.

Definition 2 (Separable permutation)**.** *The class of separable permutations, denoted $\mathcal{S}ep$, is $\mathcal{S}ep = S(2413, 3142)$.*

There are numerous characterizations of separable permutations, for example in terms of permutation graphs [5], of interval decomposition [21, 4, 6], or with ad-hoc structures like the separating trees [5, 14]. Separable permutations have been widely studied in the last decade, both from a combinatorial [20, 11] and an algorithmic [3, 5, 14] point of view.

We define two operations of concatenation on permutation patterns:

Definition 3 (Pattern concatenation)**.** *Consider two patterns π and π' of respective lengths k and k'. The positive and negative concatenations of π and π' are* *defined respectively by:*

$$\pi \oplus \pi' = \pi_1 \cdots \pi_k (\pi'_1 + k) \cdots (\pi'_{k'} + k)$$
$$\pi \ominus \pi' = (\pi_1 + k') \cdots (\pi_k + k')\pi'_1 \cdots \pi'_{k'}$$

The following property, whose proof is straightforward with separating trees, is worth noticing for our purpose:

Property 1. *If both π and π' are separable patterns, then $\pi \oplus \pi'$ and $\pi \ominus \pi'$ are also separable. Conversely, any separable pattern π of length at least 2 can be decomposed into $\pi = \pi_1 \oplus \pi_2$ or $\pi = \pi_1 \ominus \pi_2$ for some smaller but non-empty separable patterns π_1 and π_2.*

1.2. **Pattern problems on permutations.** The first investigated problem on patterns in permutations is the *Pattern Involvement* Problem:
Problem 1 (Pattern Involvement Problem)**.**
Input*: A pattern π and a permutation σ.*
Output*: A boolean indicating whether π is involved in σ or not.*

It was shown to be NP-complete in [5]. However, in [5] the authors also exhibit a particular case in which it is polynomial-time solvable: namely when the pattern π in input is a separable pattern.

Another problem of interest is the *Longest Common Pattern* Problem (LCP for short):
Problem 2 (LCP Problem)**.**
Input*: A set X of permutations.*
Output*: A pattern of maximal length occurring in each $\sigma \in X$.*

This problem is clearly NP-hard in view of the complexity of Problem 1. We showed in [6] that it is solvable in polynomial time when $X = \{\sigma_1, \sigma_2\}$ with σ_1 a separable permutation (or more generally, when the length of the longest *simple permutation* [8, 7, 9] involved in σ_1 is bounded).

In this paper, we will consider a restriction of Problem 2. For any (pattern-avoiding) class $\mathcal{C}$ of permutations, we define the *Longest Common $\mathcal{C}$-Pattern* Problem (LC$\mathcal{C}$P for short):
Problem 3 (LC$\mathcal{C}$P Problem)**.**
Input*: A set X of permutations.*
Output*: A pattern of $\mathcal{C}$ of maximal length occurring in each $\sigma \in X$.*

In particular, we focus in this paper on the *Longest Common Separable Pattern* Problem (LC$\mathcal{S}ep$P) which in fact is LC$\mathcal{C}$P where $\mathcal{C} = \mathcal{S}ep$.

To our knowledge, complexity issues of the LC$\mathcal{C}$P Problem are still unexplored. We will show in this paper that the LC$\mathcal{S}ep$P Problem is NP-hard in general, but solvable in polynomial-time when the cardinality of the set X of permutations in input is bounded by any constant K. However the method we use in our polynomial-time algorithm for solving LC$\mathcal{S}ep$P on K permutations is specific to separable patterns and cannot be extended to any class $\mathcal{C}$ of pattern-avoiding permutations.

Some classes $\mathcal{C}$ of permutations are known for which even the *Recognition* Problem (*i.e.*, deciding if a permutation belongs to $\mathcal{C}$) is NP-hard, so that LC$\mathcal{C}$P on K

permutations must be NP-hard for those classes. [1] gives the example of the class of 4-stack sortable permutations.

However, we are not aware of any example of *finitely based* pattern-avoiding classes of permutations (with a finite number of excluded patterns) for which the Recognition Problem is NP-hard. Thus an open question is to know if the LC$\mathcal{C}$P problem for K permutations is polynomial-time solvable for any finitely based $\mathcal{C}$, or to exhibit such a class $\mathcal{C}$ for which this problem is NP-hard.

2. Polynomial algorithm for the longest common separable pattern between a finite number of permutations

In [5], the authors show that the problem of deciding whether a permutation π is a pattern of a permutation σ is NP-complete. A consequence is that the problem of finding a longest common pattern between two or more permutations in NP-hard. However, they describe a polynomial-time algorithm for solving the Pattern Involvement Problem when the pattern π is *separable.* This algorithm uses dynamic programming, and processes the permutation according to one of its separating trees.

With the same ideas, we described in [6] a polynomial-time algorithm for finding a longest common pattern between two permutations, provided that one of them is separable. Notice that a longest common pattern between two permutations, one of them being separable, is always separable.

In this section, we generalize the result obtained in [6] giving a polynomial-time algorithm for finding a longest common *separable* pattern between K permutations, K being any fixed integer, $K \geq 1$. Notice that we make no hypothesis on the K input permutations.

Like in [5] and [6], our algorithm will use dynamic programming. However, since we do not have a separability hypothesis on any of the permutations, we cannot design an algorithm based on a separating tree associated to one of the permutations in input. To compute a longest common separable pattern between the input permutations, we will only consider sub-problems corresponding to K-tuples of intervals of indices and values, one such pair of intervals for each permutation.

Namely, let us consider K permutations $\sigma^1, \dots, \sigma^K$, of length $n_1, \dots, n_K$ respectively, and denote by n the maximum of the n_q's, $1 \leq q \leq K$. For computing a longest common separable pattern between $\sigma^1, \dots, \sigma^K$, we will consider a dynamic programming array M of dimension $4K$, and when our procedure for filling in M ends, we intend that $M(i_1, j_1, a_1, b_1, \dots, i_K, j_K, a_K, b_K)$ contains a common separable pattern π between $\sigma^1, \dots, \sigma^K$ that is of maximal length among those using, for any $q \in [1..K]$, intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values in σ^q. If we are able to fill in M in polynomial time, with the above property being satisfied, the entry $M(1, n_1, 1, n_1, \dots, 1, n_K, 1, n_K)$ will contain, at the end of the procedure, a longest common separable pattern between $\sigma^1, \dots, \sigma^K$.

Algorithm 1 shows how the array M can indeed be filled in in polynomial time. In Algorithm 1, *Longest* is the naive linear-time procedure that runs through a set S of patterns and returns a pattern in S of maximal length.

Before giving the details of the proof of our algorithm for finding a longest common separable pattern, we state and prove two lemmas. They should help

Algorithm 1 Longest common separable pattern between K permutations

```
INPUT: $K$ permutations $\sigma^1, \ldots, \sigma^K$ of length $n_1, \ldots, n_K$ respectively

CREATE AN ARRAY $M$:
for any integers $i_q$, $j_q$, $a_q$ and $b_q \in [1..n_q]$, for all $q \in [1..K]$ do
    $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K) \leftarrow \epsilon$
end for

FILL IN $M$:
for any integers $i_q$, $j_q$, $a_q$ and $b_q \in [1..n_q]$, $i_q \leq j_q$, $a_q \leq b_q$, for all $q \in [1..K]$,
by increasing values of $\sum_q (j_q - i_q) + (b_q - a_q)$ do
    if $\exists q \in [1..K]$ such that $i_q = j_q$ or $a_q = b_q$ then
        if $\forall q \in [1..K], \exists h_q \in [i_q..j_q]$ such that $\sigma^q_{h_q} \in [a_q..b_q]$ then
            $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K) \leftarrow 1$
        else
            $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K) \leftarrow \epsilon$
        end if
    else
        /*$\forall q \in [1..K], i_q < j_q$ and $a_q < b_q$ /*
        $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K) \leftarrow Longest(S_\oplus \cup S_\ominus \cup S)$ where
  $S_\oplus = \{M(i_1, h_1 - 1, a_1, c_1 - 1, \ldots, i_K, h_K - 1, a_K, c_K - 1) \oplus M(h_1, j_1, c_1, b_1,$
        $\ldots, h_K, j_K, c_K, b_K) : i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]\}$
  $S_\ominus = \{M(i_1, h_1 - 1, c_1, b_1, \ldots, i_K, h_K - 1, c_K, b_K) \ominus M(h_1, j_1, a_1, c_1 - 1,$
        $\ldots, h_K, j_K, a_K, c_K - 1) : i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]\}$
  $S = \{1\}$ if $\forall q \in [1..K], \exists h_q \in [i_q..j_q]$ such that $\sigma^q_{h_q} \in [a_q..b_q]$,
    $= \{\epsilon\}$ otherwise.
    end if
end for

OUTPUT: $M(1, n_1, 1, n_1, \ldots, 1, n_K, 1, n_K)$
```

understanding how common separable patterns can be merged, or on the contrary split up, to exhibit other common separable patterns. We are also interested in the stability of the maximal length property when splitting up patterns.

Lemma 1. *Let π_1 be a common separable pattern between $\sigma^1, \ldots, \sigma^K$ that uses the intervals $[i_q..h_q - 1]$ of indices and $[a_q..c_q - 1]$ (resp. $[c_q..b_q]$) of values in σ^q, for all $q \in [1..K]$.*

Let π_2 be a common separable pattern between $\sigma^1, \ldots, \sigma^K$ that uses the intervals $[h_q..j_q]$ of indices and $[c_q..b_q]$ (resp. $[a_q..c_q - 1]$) of values in σ^q, for all $q \in [1..K]$.

Then $\pi = \pi_1 \oplus \pi_2$ (resp. $\pi = \pi_1 \ominus \pi_2$) is a common separable pattern between $\sigma^1, \ldots, \sigma^K$ that uses the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values in σ^q, for all $q \in [1..K]$.

Proof. We give a proof for $\pi = \pi_1 \oplus \pi_2$ (the case $\pi = \pi_1 \ominus \pi_2$ being similar).

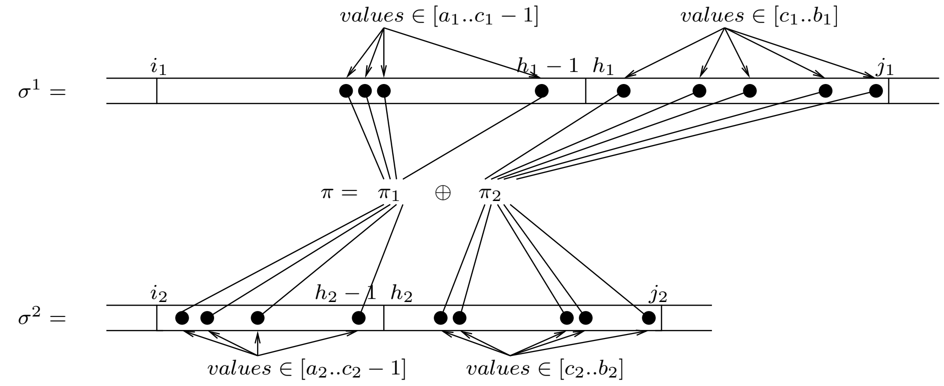

Figure 1. Proof of lemma 1 for $K = 2$

Fix some $q \in [1..K]$. By hypothesis, there exist occurrences of π_1 and π_2 in σ^q, the occurrence of π_1 using the intervals $[i_q..h_q - 1]$ of indices and $[a_q..c_q - 1]$ of values, and the occurrence of π_2 using the intervals $[h_q..j_q]$ of indices and $[c_q..b_q]$ of values. It is then easily noticed (see Figure 1) that all the elements used in one of these occurrences form an occurrence of the pattern $\pi = \pi_1 \oplus \pi_2$ in σ^q in the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values. This argument holds for any $q \in [1..K]$ and hence π is a common separable pattern between $\sigma^1, \dots, \sigma^K$ using the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values in σ^q, for all $q \in [1..K]$.

□ □

Lemma 2. *Let π be a common separable pattern of maximal length between $\sigma^1, \dots, \sigma^K$ among those using the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values in σ^q, for all $q \in [1..K]$.*

If $\pi = \pi_1 \oplus \pi_2$ (resp. $\pi = \pi_1 \ominus \pi_2$), with π_1 and π_2 non-empty *separable patterns, then there exist indices $(h_q)_{q\in[1..K]}$ and values $(c_q)_{q\in[1..K]}$, with $i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]$, such that:*

i) π_1 is a common separable pattern of maximal length between $\sigma^1, \dots, \sigma^K$ among those using the intervals $[i_q..h_q - 1]$ of indices and $[a_q..c_q - 1]$ (resp. $[c_q..b_q]$) of values in σ^q, for all $q \in [1..K]$, and

ii) π_2 is a common separable pattern of maximal length between $\sigma^1, \dots, \sigma^K$ among those using the intervals $[h_q..j_q]$ of indices and $[c_q..b_q]$ (resp. $[a_q..c_q - 1]$) of values in σ^q, for all $q \in [1..K]$.

Proof. Again, consider the case $\pi = \pi_1 \oplus \pi_2$ (the case $\pi = \pi_1 \ominus \pi_2$ being similar).

Fix some $q \in [1..K]$. By hypothesis, $\pi = \pi_1 \oplus \pi_2$ has an occurrence in σ^q in the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values. By definition of positive pattern concatenation, this occurrence splits into two occurrences of π_1 and π_2 respectively (see again Figure 1). More precisely, there exist $h_q \in [i_q + 1..j_q]$ and $c_q \in [a_q + 1..b_q]$ such that π_1 (resp. π_2) has an occurrence in the intervals $[i_q..h_q - 1]$ (resp. $[h_q..j_q]$) of indices and $[a_q..c_q - 1]$ (resp. $[c_q..b_q]$) of values. This argument holding for all $q \in [1..K]$, it becomes clear that π_1 (resp. π_2) is a common separable pattern between $\sigma^1, \dots, \sigma^K$ that uses the intervals $[i_q..h_q - 1]$ (resp. $[h_q..j_q]$) of indices and

$[a_q..c_q-1]$ (resp. $[c_q..b_q]$) of values in σ^q, for all $q \in [1..K]$. What remains to prove is that π_1 and π_2 are of maximal length among all such patterns.

Assume that π_1 is *not* of maximal length among the common separable patterns between $\sigma^1, \ldots, \sigma^K$ using interval $[i_q..h_q-1]$ of indices and interval $[a_q..c_q-1]$ of values in σ^q, for all $q \in [1..K]$. Then, there exists π_1^{long}, a common separable pattern between $\sigma^1, \ldots, \sigma^K$ using interval $[i_q..h_q-1]$ of indices and interval $[a_q..c_q-1]$ of values in σ^q, for all $q \in [1..K]$, such that $|\pi_1^{long}| > |\pi_1|$. Now by Lemma 1, $\pi_1^{long} \oplus \pi_2$ is a common separable pattern between $\sigma^1, \ldots, \sigma^K$ using the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values in σ^q, for all $q \in [1..K]$. And we have $|\pi_1^{long} \oplus \pi_2| > |\pi_1 \oplus \pi_2| = |\pi|$, contradicting the maximality of π. So π_1 is a common separable pattern of maximal length between $\sigma^1, \ldots, \sigma^K$ among those using interval $[i_q..h_q-1]$ of indices and interval $[a_q..c_q-1]$ of values in σ^q, for all $q \in [1..K]$. In the same way, we prove that π_2 is a common separable pattern of maximal length between $\sigma^1, \ldots, \sigma^K$ among those using interval $[h_q..j_q]$ of indices and interval $[c_q..b_q]$ of values in σ^q, for all $q \in [1..K]$, ending the proof of the lemma. □ □

Proposition 1. *Algorithm 1 is correct: it outputs a longest common separable pattern between the K permutations in input.*

Proof. Consider the array M returned by Algorithm 1. We show by induction on $\sum_q (j_q - i_q) + (b_q - a_q)$ that $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K)$ contains a common separable pattern π between $\sigma^1, \ldots, \sigma^K$ that is of maximal length among those using, for any $q \in [1..K]$, intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values in σ^q.

First, there is no loss of generality in assuming that $i_q \leq j_q$ and $a_q \leq b_q$ for all $q \in [1..K]$, since otherwise the above statement is clearly true (indeed, $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K)$ contains ϵ and either $[i_q..j_q] = \emptyset$ or $[a_q..b_q] = \emptyset$ for some $q \in [1..K]$).

If $\sum_q (j_q - i_q) + (b_q - a_q) = 0$, then $i_q = j_q$ and $a_q = b_q$ for all $q \in [1..K]$. Consequently, the pattern we would like $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K)$ to contain is a longest common separable pattern between $\sigma^1, \ldots, \sigma^K$ that uses only index $i_q = j_q$ and value $a_q = b_q$ in σ^q for all $q \in [1..K]$. Such a pattern is either ϵ or 1. And it is 1 if and only if $\forall q \in [1..K], \sigma^q_{i_q} = a_q$ that is to say if and only if $\forall q \in [1..K], \exists h_q \in [i_q..j_q]$ such that $\sigma^q_{h_q} \in [a_q..b_q]$. On lines 8 to 12 of Algorithm 1, we see that in this case $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K)$ is set correctly.

If $\sum_q (j_q - i_q) + (b_q - a_q) > 0$, we must consider two subcases:

If $\exists q \in [1..K]$ such that $i_q = j_q$ or $a_q = b_q$, let us call π a common separable pattern between $\sigma^1, \ldots, \sigma^K$ that is of maximal length among those using, for any $q \in [1..K]$, intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values in σ^q. Then, just as before, π is either 1 or ϵ, because in at least one of the permutations, say σ^q, the occurrence of π can use at most one index (if $i_q = j_q$) or at most one value (if $a_q = b_q$). And again, lines 8 to 12 of Algorithm 1 show that $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K)$ contains 1 or ϵ. More precisely, the condition on line 9 ensures that it contains 1 exactly when 1 has an occurrence in σ^q in the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values, for any $q \in [1..K]$.

It remains to consider the recursive case when $\forall q \in [1..K], i_q < j_q$ and $a_q < b_q$. In this case, consider π a common separable pattern π between $\sigma^1, \ldots, \sigma^K$ that is of maximal length among those such that, for any $q \in [1..K]$, π has an occurrence in σ^q in the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values. Since π is separable, then either $\pi = \epsilon$, or $\pi = 1$, or $\pi = \pi_1 \oplus \pi_2$, or $\pi = \pi_1 \ominus \pi_2$ where π_1 and π_2 are smaller but non-empty separable patterns. Now consider π_{algo} a longest pattern in the set $S_\oplus \cup S_\ominus \cup S$ where

$$\begin{aligned}
S_\oplus &= \{M(i_1, h_1-1, a_1, c_1-1, \dots, i_K, h_K-1, a_K, c_K-1) \oplus M(h_1, j_1, c_1, b_1,\\
&\quad \dots, h_K, j_K, c_K, b_K) : i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]\}\\
S_\ominus &= \{M(i_1, h_1-1, c_1, b_1, \dots, i_K, h_K-1, c_K, b_K) \ominus M(h_1, j_1, a_1, c_1-1,\\
&\quad \dots, h_K, j_K, a_K, c_K-1) : i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]\}\\
S &= \{1\} \text{ if } \forall q \in [1..K], \exists h_q \in [i_q..j_q] \text{ such that } \sigma^q_{h_q} \in [a_q..b_q],\\
&= \{\epsilon\} \text{ otherwise.}
\end{aligned}$$

By induction hypothesis, each entry of M that appears in the set $S_\oplus \cup S_\ominus$ is a common separable pattern between $\sigma^1, \dots, \sigma^K$ whose occurrence in σ^q, for any $q \in [1..K]$, uses indices and values in the prescribed intervals, and that is of maximal length among all such patterns. Notice also that $S = 1$ if and only if 1 has an occurrence in σ^q in the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values, for all $q \in [1..K]$. An immediate consequence of those two facts and of Lemma 1 is that π_{algo} is a common separable pattern between $\sigma^1, \dots, \sigma^K$ which has, for any $q \in [1..K]$, an occurrence in σ^q in the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values. What is left is to prove that $|\pi_{algo}| = |\pi|$. This is clear when $\pi = \epsilon$ or 1. So assume that $\pi = \pi_1 \oplus \pi_2$, the case $\pi = \pi_1 \ominus \pi_2$ being very similar. Since $\pi = \pi_1 \oplus \pi_2$ has an occurrence in each σ^q in the intervals $[i_q..j_q]$ of indices and $[a_q..b_q]$ of values, by Lemma 2, there exists indices $(h_q)_{q \in [1..K]}$ and values $(c_q)_{q \in [1..K]}$, with $i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]$, such that π_1 has an occurrence in each σ^q in the intervals $[i_q..h_q-1]$ of indices and $[a_q..c_q-1]$ of values and π_2 has an occurrence in each σ^q in the intervals $[h_q..j_q]$ of indices and $[c_q..b_q]$ of values. Lemma 2 also states that π_1 and π_2 are of maximal length among the common separable patterns in the given intervals of indices and values. So by induction hypothesis, $|M(i_1, h_1-1, a_1, c_1-1, \dots, i_K, h_K-1, a_K, c_K-1)| = |\pi_1|$ and $|M(h_1, j_1, c_1, b_1, \dots, h_K, j_K, c_K, b_K)| = |\pi_2|$. Consequently, $|\pi| = |\pi_1 \oplus \pi_2| = |M(i_1, h_1-1, a_1, c_1-1, \dots, i_K, h_K-1, a_K, c_K-1)| + |M(h_1, j_1, c_1, b_1, \dots, h_K, j_K, c_K, b_K)| \leq |\pi_{algo}|$. The inequality $|\pi| \geq |\pi_{algo}|$ being obvious by maximality of π, we conclude that $|\pi| = |\pi_{algo}|$. This ends the proof in the case $\pi = \pi_1 \oplus \pi_2$. For the case $\pi = \pi_1 \ominus \pi_2$, the proof follows the exact same steps, with π_1 having an occurrence in each σ^q in the intervals $[i_q..h_q-1]$ of indices and $[c_q..b_q]$ of values and π_2 having an occurrence in each σ^q in the intervals $[h_q..j_q]$ of indices and $[a_q..c_q-1]$ of values. □ □

Proposition 2. *Algorithm 1 runs in time $\mathcal{O}(n^{6K+1})$ and space $\mathcal{O}(n^{4K+1})$.*

Proof. Algorithm 1 handles an array M of size $\mathcal{O}(n^{4K})$, where each cell contains a pattern of length at most n, so that the total space complexity is $\mathcal{O}(n^{4K+1})$. For filling in one entry $M(i_1, j_1, a_1, b_1, \dots, i_K, j_K, a_K, b_K)$, if $\exists q \in [1..K]$ such that $i_q = j_q$ or $a_q = b_q$ (lines 9 to 13 of Algorithm 1), the time complexity is $\mathcal{O}(n^K)$. If no such q exists (line 15 of Algorithm 1), the time complexity needed to fill in $M(i_1, j_1, a_1, b_1, \dots, i_K, j_K, a_K, b_K)$, using the entries of M previously computed, is $\mathcal{O}(n^{2K+1})$. Indeed, we search for an element of maximal length among $\mathcal{O}(n^{2K})$ elements, each element being computed in $\mathcal{O}(n)$-time as the concatenation of two

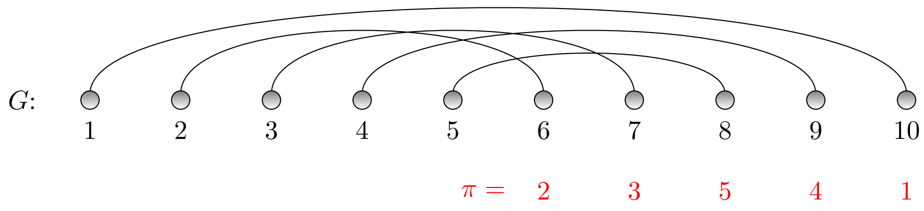

Figure 2. Shown here is the correspondence between the permutation $\pi = 2\ 3\ 5\ 4\ 1$ and the associated matching diagram G.

previously computed entries of M. Consequently, the total time complexity to fill in M is $\mathcal{O}(n^{6K+1})$. □ □

A consequence of propositions 1 and 2 is:

Theorem 1. *For any fixed integer K, the problem of computing a longest common separable pattern between K permutations is in P.*

We may wonder whether a longest common separable pattern between two permutations σ^1 and σ^2 (computed in polynomial time by Algorithm 1) is a good approximation of a longest common pattern between σ^1 and σ^2 (whose computation is NP-hard). Section 4 gives a negative answer to this question, by the more general Corollary 1.

3. Hardness result

We proved in the preceding section that the LC$\mathcal{S}ep$P problem is polynomial-time solvable provided a constant number of input permutations. We show here that the C$\mathcal{S}ep$P problem (the general decision version of LC$\mathcal{S}ep$P), is NP-complete.
Problem 4 (C$\mathcal{S}ep$P Problem)**.**
Input*: A set X of permutations and an integer k.*
Output*: A boolean indicating if there a separable pattern of length k occurring in each $\sigma \in X$.*

Actually, we will prove more, namely that the C$\mathcal{S}ep$P problem is NP-complete even if each input permutation is separable. An immediate consequence is the NP-hardness of LC$\mathcal{S}ep$P. For ease of exposition, our proof is given in terms of matching diagrams.

Definition 4 (Matching Diagram)**.** *A* matching diagram *G of size n is a vertex-labeled graph of order i.e., number of vertices, $2n$ and size i.e., number of edges, n where each vertex is labeled by a distinct label from $\{1, 2, \ldots, 2n\}$ and each vertex $i \in \{1, 2, \ldots, n\}$ (resp. $j \in \{n+1, n+2, \ldots, 2n\}$) is connected by an edge to exactly one vertex $j \in \{n+1, n+2, \ldots, 2n\}$ (resp. $i \in \{1, 2, \ldots, n\}$). We denote the set of vertices and edges of G by $V(G)$ and $E(G)$, respectively.*

It is well-known that matching diagrams of size n are in one-to-one correspondence with permutations of length n (see Figure 2 for an illustration).

Let G and G' be two matching diagrams. The matching diagram G' is said to *occur* in G if one can obtain G' from G by a sequence of edge deletions. More formally, the deletion of the edge (i, j), $i < j$, consists in (1) the deletion of the edge (i, j), (2) the deletion of the two vertices i and j, and (3) the relabeling of all vertices $k \in [i+1..j-1]$ to $k-1$ and all the vertices $k > j$ to $k-2$. Therefore, the decision version of the LC$\mathcal{S}ep$P is equivalent to the following problem: Given a set of matching diagrams and a positive integer k, find a matching diagram of size k which occurs in each input diagram [16].

Clearly, two edges in a matching diagram G are either crossing or nested . Moreover, it is easily seen that an occurrence in G of a matching diagram G' of which all edges are crossing (resp. nested) correspond to an occurrence in the permutation associated with G of an *increasing* (resp. *decreasing*) subsequence.

For the purpose of permutations, convenient matching diagrams are needed. A matching diagram is called a *tower* if it is composed of pairwise nested edges and a *staircase* if it is composed of pairwise crossing edges . A matching diagram is called a *tower of staircases* if its edge set can be partitioned in nested staircases .

Theorem 2. *The* C$\mathcal{S}ep$P *problem is* NP*-complete even if each input permutation is separable.*

Proof. C$\mathcal{S}ep$P is clearly in NP. For proving hardness, we reduce from the Independent-Set problem which is known to be NP-complete [13] . Let G be an arbitrary graph instance of the Independent-Set problem. Write $V(G) = \{1, 2, \ldots, n\}$. We now detail the construction of $n+1$ matching diagrams $G_0, G_1, G_2, \ldots, G_n$, each corresponding to a separable permutation. First the matching diagram G_0 is a tower of n staircases $A_{0,1}, A_{0,2}, \ldots, A_{0,n}$, each of size $n+1$ (see Figure 3, middle part; staircases are represented by shaded forms), *i.e.*,

$$\forall j,\ 1 \le j \le n, \qquad |A_{0,j}| = n+1.$$

Each matching diagram G_i, $1 \le i \le n$, is composed of two crossing towers of n staircases each referred as $A_{i,1}, A_{i,2}, \ldots, A_{i,n}$ and $B_{i,1}, B_{i,2}, \ldots, B_{i,n}$ (see Figure 3, bottom part), and defined as follows:

$$\forall i,\ 1 \le i \le n,\ \ \forall j,\ 1 \le j \le n, \qquad |A_{i,j}| \begin{cases} n+1 & \text{if } i \neq j \\ n & \text{if } i = j \end{cases}$$

$$\forall i,\ 1 \le i \le n,\ \ \forall j,\ 1 \le j \le n, \qquad |B_{i,j}| \begin{cases} n+1 & \text{if } (i,j) \notin E(G) \\ n & \text{if } (i,j) \in E(G). \end{cases}$$

It is simple matter to check that all matching diagrams G_i, $0 \le i \le n$, correspond to separable permutations and that our construction can be carried on in polynomial time. This ends our construction.

We claim that there exists an independent set $V' \subseteq V(G)$ of size k in G if and only if there exists a matching diagram G_{sol} of size $n^2 + k$ that occurs in each input matching diagram G_i, $0 \le i \le n$.

Suppose that there exists an independent set $V' \subseteq V(G)$ of size k in G. Consider as a solution G_{sol} the tower of n staircases $C_1, C_2, \ldots, C_n$ (see Figure 3, top part) of total size $n^2 + k$, where the size of each staircase is defined according to the following: $|C_i| = n$ if $i \notin V'$ and $|C_i| = n + 1$ if $i \in V'$. We claim that $G_{\rm sol}$ occurs in each input matching diagram G_i, $0 \le i \le n$. Indeed, by construction, for any $i \in V(G)$, if $i \notin V'$ then $G_{\rm sol}$ occurs in Side–A of G_i and if $i \in V'$ then $G_{\rm sol}$ occurs in Side–B of G_i. To complete the claim, we note that $G_{\rm sol}$ occurs in G_0 (G_0 is indeed a tower of n staircases, each of size $n + 1$).

Conversely, suppose that there exists a matching diagram $G_{\rm sol}$ of size n^2+k that occurs in each input matching diagram G_i, $0 \le i \le n$. Let us prove the following:

Claim 1. *G_{sol} is a tower of n staircases. Furthermore, for any $i \in [1..n]$, G_{sol} occurs completely in Side–A or completely in Side–B in G_i.*

Proof. Let us first focus on an occurrence of $G_{\rm sol}$ in G_0. Since G_0 is a tower of staircases, then it follows that $G_{\rm sol}$ is a tower of staircases as well. We now prove that $G_{\rm sol}$ is a tower of n staircases. Indeed, suppose, aiming at a contradiction, that $G_{\rm sol}$ is a tower of $n' < n$ staircases. Then it follows that $G_{\rm sol}$ has size at most $(n + 1)(n - 1) = n^2 - 1 < n^2 + k$. This is the desired contradiction, and hence $G_{\rm sol}$ is a tower of n staircases $C_1, C_2, \ldots, C_n$, each of size at most $n + 1$. We now turn to considering an occurrence of $G_{\rm sol}$ in G_i, $1 \le i \le n$. We prove that $G_{\rm sol}$ occurs completely in Side–A or completely in Side–B (see Figure 3). Suppose, for the sake of contradiction, that $G_{\rm sol}$ matches at least one edge in Side–A, say e, and at least one edge in Side–B, say e'. By construction, e and e' are crossing edges in G_i, and hence e and e' are matched by two edges that belong to the same staircase in $G_{\rm sol}$ ($G_{\rm sol}$ is indeed a tower of staircases). We now observe that any edge in Side–A crosses the edge e' and any edge in Side–B crosses the edge e. Then it follows that the occurrence of $G_{\rm sol}$ in G_i induces a single staircase. But $G_{\rm sol}$ is a tower of n staircases, each of size at most $n+1$. A contradiction. Therefore, in any matching diagram G_i, $1 \le i \le n$, $G_{\rm sol}$ occurs completely in Side–A or completely in Side–B. □

As an important consequence of the claim, there is thus no loss of generality in assuming that each staircase of $G_{\rm sol}$ has size n or $n+1$, and hence $G_{\rm sol}$ is composed of $n-k$ staircases of size n and k staircases of size $n+1$ (since $G_{\rm sol}$ has size n^2+k). Consider now the subset $V' \subseteq V(G)$ defined as follows: $i \in V(G)$ belongs to V' if and only if $|C_i| = n + 1$, *i.e.*, the staircase C_i has size $n + 1$ in $G_{\rm sol}$. According to the above, V' has certainly size k. We shall show that V' is an independent set in G. Indeed, let $i \in V'$ and consider the matching diagram G_i. Since $i \in V'$ then it follows that $|C_i| = n + 1$, and hence $G_{\rm sol}$ occurs in Side–B in G_i (the latter follows from the fact that $|A_{i,i}| = n$). But, by construction, for $1 \le j \le n$, $|B_{i,j}| = n + 1$ if and only if (i, j) is not an edge in G (in particular, $|B_{i,i}| = n + 1$). Hence, the vertex i is not connected to any vertex in V'. Therefore, since the argument applies to each matching diagram G_i with $i \in V'$, V' is an independent set in G. □ □

4. Approximation ratio

In this section, we return to the LC$\mathcal{C}$P Problem for K permutations. As said before, the general LCP Problem is NP-hard as well as the Pattern Involvement Problem [5]. In this section we prove the following result:

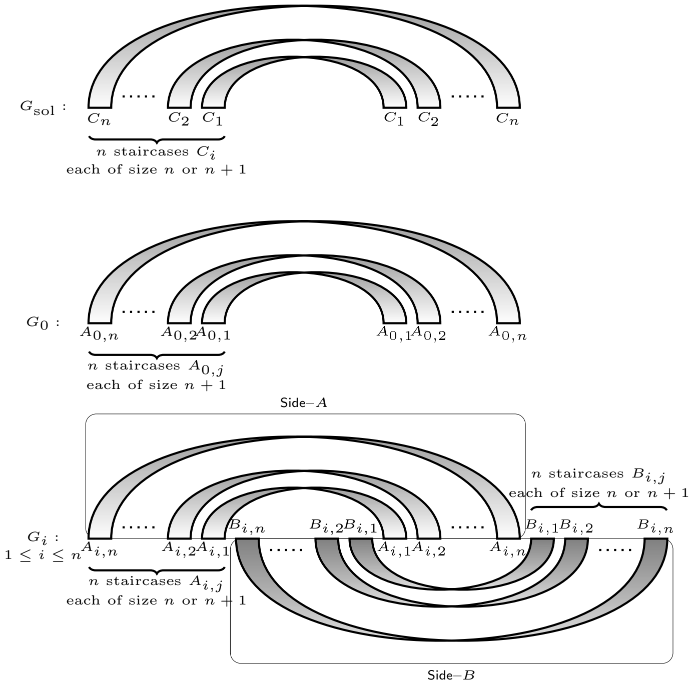

FIGURE 3. Reduction in the proof of Theorem 2

Theorem 3. *For all $\epsilon > 0$ and $\mathcal{C}$, a pattern-avoiding class of permutations, there exists a sequence $(\sigma_n)_{n\in\mathbb{N}}$ of permutations $\sigma_n \in S_n$ such that*

$$|\pi_n| = o\left(n^{0.5+\epsilon}\right)$$

where π_n is the longest pattern of class $\mathcal{C}$ involved in σ_n.

Before proving this result we need the following Lemma.

Lemma 3. *Given a permutation $\pi \in S_k$, the number of permutations $\sigma \in S_n$ such that π is involved in σ is at most $(n-k)!\binom{n}{k}^2$.*

Proof. Let $\pi = \pi_1\pi_2\ldots,\pi_k$ and $\sigma = \sigma_1\sigma_2\ldots\sigma_n$ be such that π is involved in σ. There exist $i_1 < i_2 < \ldots < i_k$ such that $\sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_k}$ is order-isomorphic to π. Then $\sigma = u_1\sigma_{i_1}u_2\sigma_{i_2}\ldots u_k\sigma_{i_k}u_{k+1}$ where u_i is a factor of σ. We will call u_i's the blocks associated to σ, π and an occurrence of π in σ.

Conversely, suppose we are given a permutation $\pi \in S_k$, and $u_1, u_2, \ldots, u_{k+1}$ $k+1$ sequences of distinct numbers in $\{1\ldots n\}$ such that $|u_1|+|u_2|+\ldots+|u_{k+1}| = n-k$. Let $E = \{n_1, n_2, \ldots n_k\}$ the k integers in $\{1\ldots n\}$ which do not appear in a block u_i. We denote by $\bar{j}$ the j^{th} smallest element in E. Then the permutation $\sigma = u_1\overline{\pi_1}u_2\overline{\pi_2}\ldots\overline{\pi_k}u_{k+1}$ is a permutation of S_n and π is involved in σ. For example, if $\pi = 2143$, $n = 9$, $u_1 = 31$, $u_2 = \varnothing$, $u_3 = 8$, $u_4 = 65$ and $u_5 = \varnothing$ then $E = \{2,4,7,9\}$ and $\sigma = 31\mathbf{42}8\mathbf{9}65\mathbf{7}$.

Note that two different lists of blocks could lead to the same permutation σ if π has several occurrences in σ as shown in Figure 4.

$$2431 \longrightarrow \begin{cases} (1, \varnothing, \varnothing, 56, \varnothing) \\ (1, \varnothing, 4, 6, \varnothing) \\ (1, \varnothing, 45, \varnothing, \varnothing) \end{cases} \longrightarrow 1374562$$

FIGURE 4. Relation between pattern, blocks and permutation

Thus, the number of permutations σ such that π is involved in σ is *at most* the number of different lists of blocks. There are $\binom{n}{n-k}$ different choices for the numbers that appear in one of the u_i. Then there are $(n-k)!$ different orders for these numbers. The last step is to cut the word so obtained into $k+1$ (possibly empty) blocks $u_1, u_2, \dots u_{k+1}$. There are $\binom{n}{n-k}$ such choices. Hence we have the claimed formula. □ □

We can now prove Theorem 3.

Proof. We make the proof by contradiction. We first prove that if the result were false, every permutation of length n would contain a pattern of $\mathcal{C}$ of length *exactly* $k = \lceil n^{0.5+\epsilon} \rceil$. Next, we show that the number of permutations of length n containing one permutation of $\mathcal{C} \bigcap S_k$ as a pattern is strictly less than $n!$.

Suppose that there exist $\epsilon > 0$ and $\mathcal{C}$ a pattern-avoiding class of permutations such that for every permutation $\sigma \in S_n$, the longest pattern $\pi \in \mathcal{C}$ of σ has length $|\pi| \geq \lceil |\sigma|^{0.5+\epsilon} \rceil = k$. As $\mathcal{C}$ is closed - every pattern π of a permutation $\sigma \in \mathcal{C}$ is also in $\mathcal{C}$- for every permutation $\sigma \in S_n$ there exists a pattern $\pi \in \mathcal{C}$ of σ whose length is *exactly* $|\pi| = k$.

But the number of permutations in $\mathcal{C} \bigcap S_k$ is at most c^k by [18]. By Lemma 3, for each permutation in $\mathcal{C} \bigcap S_k$, the number of permutations in S_n having this permutation as a pattern is at most $(n-k)!\binom{n}{k}^2$. Thus the number of permutations in S_n having a pattern in $\mathcal{C} \bigcap S_{\geq k}$ is at most $c^k(n-k)!\binom{n}{k}^2$. But with $k = \lceil n^{0.5+\epsilon} \rceil$, $c^k(n-k)!\binom{n}{k}^2 = o\left(n^{n^{1-2\epsilon}}\right) = o(n!)$. Note that a similar proof is given in [12] for finding the smallest permutation containing all patterns of a given length. □ □

Corollary 1. *The* LCP *Problem cannot be approximated with a tighter ratio than* $\sqrt{Opt}$ *by the* LC$\mathcal{C}$P *Problem, where* $\mathcal{C}$ *is a pattern-avoiding class of permutations, and Opt is the size of an optimal solution to the* LCP *Problem.*

Proof. Consider the LCP Problem between σ and σ. Then the optimal solution to the LCP Problem is σ. But the solution to the LC$\mathcal{C}$P Problem is a longest pattern of σ belonging to the class $\mathcal{C}$. By Theorem 3, such a pattern may have size $\sqrt{|\sigma|}$ asymptotically. □ □

REFERENCES

[1] M.H. Albert, R. E. L. Aldred, M. D. Atkinson, H. P. van Ditmarsch, B. D. Handley, C. C. Handley, and J. Opatrny. Longest subsequences in permutations. *Australian J. Combinatorics*, 28:225–238, 2003.

[2] Michael H. Albert, Robert E. L. Aldred, Mike D. Atkinson, and Derek A. Holton. Algorithms for pattern involvement in permutations. In *Proc. International Symposium on Algorithms and Computation (ISAAC)*, volume 2223 of *Lecture Notes in Computer Science*, pages 355–366, 2001.

[3] S. Bérard, A. Bergeron, C. Chauve, and C. Paul. Perfect sorting by reversals is not always difficult. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(1), 2007.

[4] A. Bergeron, C. Chauve, F. de Montgolfier, and M. Raffinot. Computing common intervals of K permutations, with applications to modular decomposition of graphs. pages 779–790. ESA, 2005.

[5] P. Bose, J.F. Buss, and A. Lubiw. Pattern matching for permutations. *Information Processing Letters*, 65(5):277–283, 1998.

[6] M. Bouvel and D. Rossin. The longest common pattern problem for two permutations. *Pure Mathematics and Applications*, 2007. to be published, arXiv:math.CO/0611679.

[7] R. Brignall, S. Huczynska, and V. Vatter. Decomposing simple permutations, with enumerative consequences. arXiv:math.CO/0606186, 2006.

[8] R. Brignall, S. Huczynska, and V. Vatter. Simple permutations and algebraic generating functions. arXiv:math:CO/0608391, 2006.

[9] R. Brignall, N. Ruškuc, and V. Vatter. Simple permutations: decidability and unavoidable substructures. arXiv:math.CO/0609211, 2006.

[10] M.-S Chang and G.-H Wang. Efficient algorithms for the maximum weight clique and maximum weight independent set problems on permutation graphs. *Information Processing Letters*, 43:293–295, 1992.

[11] A. Ehrenfeucht, T. Harj, P. ten Pas, and G. Rozenberg. Permutations, parenthesis words, and Schröder numbers. *Discrete Mathematics*, 190:259–264, 1998.

[12] H. Eriksson, K. Eriksson, S. Linusson, and Wästlund J. Dense packing of patterns in a permutation. In *FPSAC'02*, 2002.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractablility: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.

[14] L. Ibarra. Finding pattern matchings for permutations. *Information Processing Letters*, 61:293–295, 1997.

[15] D.E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, Reading MA, 3rd edition, 1973.

[16] M. Kubica, R. Rizzi, S. Vialette, and T. Walen. On non-coding RNA multiple structural alignment. In *CPM 2006*, 2006.

[17] L. Lovász. *Combinatorial problems and exercices*. North-Holland, 1979.

[18] A. Marcus and G. Tardos. Excluded permutation matrices and the Stanley-Wilf conjecture. *J. Combin. Th. A*, 107:153–160, 2004.

[19] D. Rotem. Stack-sortable permutations. *Discrete Math.*, 33:185–196, 1981.

[20] J. West. Generating trees and the Catalan and Schröder numbers. *Discrete Mathematics*, 146:247–262, 1995.

[21] B.-M. Bui Xuan, M. Habib, and C. Paul. Revisiting T. Uno and M. Yagiura's algorithm. pages 146–155. ISAAC, 2005.

CNRS, Université Paris Diderot, Laboratoire d'Informatique Algorithmique: Fondements et Applications, 2 Place Jussieu, Case 7014,, F-75251 Paris Cedex 05, France

E-mail address: (mbouvel,rossin)@liafa.jussieu.fr

Laboratoire de Recherche en Informatique (LRI), bât.490, Univ. Paris-Sud XI,, F-91405 Orsay cedex, France

E-mail address: Stephane.Vialette@lri.fr