A variant of the tandem duplication - random loss model of genome rearrangement

Mathilde Bouvel [a] Dominique Rossin [a]

[a] *LIAFA, Université Paris Diderot, CNRS, Case 7014, 75205 Paris Cedex 13*

Abstract

In (4), Chaudhuri, Chen, Mihaescu and Rao study algorithmic properties of the *tandem duplication - random loss model* of genome rearrangement, well-known in evolutionary biology. In their model, the cost of one step of duplication-loss of width k is α^k for $\alpha = 1$ or $\alpha \geq 2$. In this paper, we study a variant of this model, where the cost of one step of width k is 1 if $k \leq K$ and ∞ if $k > K$, for any value of the parameter $K \in \mathbb{N} \cup \{\infty\}$. We first show that permutations obtained after p steps of width K define classes of pattern-avoiding permutations. We also compute the numbers of duplication-loss steps of width K necessary and sufficient to obtain any permutation of S_n, in the worst case and on average. In this second part, we may also consider the case $K = K(n)$, a function of the size n of the permutation on which the duplication-loss operations are performed.

Key words: Sorting, Permutations, Pattern
PACS:

hal-00206178, version 1 - 16 Jan 2008

1 Introduction

1.1 The model

In the usual models of genome rearrangement, duplications and losses of genes are not taken into account. There were attempts to incorporate them to the classical models, but the consecutive combinatorial complexity of the models so obtained made their study quite difficult. Following (4), we focus on the duplication-loss problem by considering the *tandem duplication - random loss model* of genome rearrangement in which genomes are modified *only* by duplications and losses of genes.

One *step* of tandem duplication - random loss, or duplication-loss for short, consists in (1) the tandem duplication of a contiguous fragment of the genome,

i.e., the duplicated fragment is inserted immediately after the original fragment, and (2) the loss of one of the two copies of every duplicated gene. We assume that the loss occurs immediately after the duplication of genes, which is, on an evolutionary time-scale, a good approximation to reality. The *width* of a step is the number of duplicated genes. See Figure 1 for an example.

$$
\begin{aligned}
1\ 2\ \overbrace{3\ 4\ 5\ 6}\ 7 &\rightsquigarrow 1\ 2\ \overbrace{3\ 4\ 5\ 6}\ \overbrace{3\ 4\ 5\ 6}\ 7 \\
&\quad \text{(tandem duplication)} \\
&\rightsquigarrow 1\ 2\ \not{3}\ 4\ 5\ \not{6}\ 3\ \not{4}\ \not{5}\ 6\ 7 \\
&\quad \text{(random loss)} \\
&\rightsquigarrow 1\ 2\ 4\ 5\ 3\ 6\ 7
\end{aligned}
$$

Fig. 1. Example of one step of tandem duplication - random loss of width 4

From a formal point of view, a genome consisting of n genes is modelled by a permutation $\pi \in S_n$ of the set of integers $\{1, 2, \ldots, n\}$. In (4), the authors define the cost of a duplication-loss step of width k to be α^k, $\alpha \geq 1$ being a constant parameter. They suggest that other cost functions can be considered, and in particular affine functions. In this paper, we consider a *piecewise constant* cost function: the cost of a step of width k is 1 if $k \leq K$ and is infinite for $k > K$, for some fixed parameter $K \in \mathbb{N} \cup \{\infty\}$. Obviously, for this model to be meaningful, we assume that $K \geq 2$. We also consider the possibility that $K = K(n)$ is dependent on the size n of the permutation on which the duplication-loss operations are performed. Both models are generalizations of the *whole genome duplication - random loss model*: it corresponds to the case $\alpha = 1$ in the model of (4), $K = \infty$ or $K = K(n) = n$ in our model.

Many models of evolution of permutations are inspired by computational biology issues: see (2), (5), (6), (7) for examples in the literature.

Our model of evolution of permutations can be viewed in the framework of *permuting machines* defined in (1). Such a machine takes a permutation in input, and transforms it into an output permutation, the transformation being subject to satisfy the two properties of independence with respect to the values and of stability with respect to pattern-involvement (see (1) for more details). The important point is that the duplication-loss transformation satisfies these two properties. Thus, one duplication-loss step (in one of the models defined above) corresponds to running an adequate permuting machine once. When we will consider permutations obtained after a sequence of duplication-loss steps, it will correspond to permutations obtained in the output of a combination in series of identical permuting machines.

For ease of exposition in some proofs, we will sometimes use a graphical representation of permutations, as shown in Figure 2.

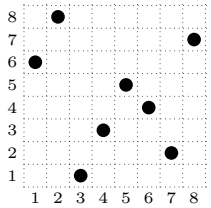


Fig. 2. The graphical representation of $\sigma = 68135427$

1.2 Pattern-avoiding classes of permutations

Though not appearing clearly for the moment, there exist strong links between the duplication-loss model and some pattern-avoiding classes of permutations. Hence, we need to recall a few definitions concerning those classes.

A permutation $\sigma \in S_n$ is a bijective map from $[1..n]$ to itself. The integer n is called the *size* of σ, denoted $|\sigma|$. 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 *position* and σ_i its *value*.

Definition 1 *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 $1 \leq i_1 < i_2 < \dots < i_k \leq n$) 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.*

We write $\pi \prec \sigma$ to denote that π is a pattern of σ.

A permutation σ that does not contain π as a pattern is said to *avoid* π. 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 size n avoiding $\pi_1, \pi_2, \dots, \pi_k$. We say that $S(\pi_1, \pi_2, \dots, \pi_k)$ is a class of pattern-avoiding permutations of *basis* $\{\pi_1, \pi_2, \dots, \pi_k\}$.

Example 2 *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 size 3 of σ is isomorphic to 321, i.e., is decreasing.*

1.3 Outline of the paper

In the tandem duplication - random loss model described above, we will focus on two kinds of problems. First, as hinted before, we will consider permutations obtained after a certain number of duplication-loss steps, that is to say permutations in output of a combination in series of a certain number of permuting machines. For this, we define the class $\mathcal{C}(K,p)$ as follows:

Definition 3 *The class $\mathcal{C}(K,p)$ denotes the class of all permutations obtained from $12\dots n$ (for any n) after p duplication-loss steps of width at most K, for some constant parameters p and K.*

We do not consider the case $K = K(n)$ here.

Be careful that the duplication-loss steps are not reversible, as noticed in (4), and that consequently $\mathcal{C}(K,p)$ is *not* the class of permutations that can be *sorted to* $12\dots n$ in p steps of duplication-loss of width at most K.

Like for the various classes of permutations obtained after a combination in series of permuting machines considered in (1), we obtained combinatorial properties of $\mathcal{C}(K,p)$ in terms of pattern-avoidance. Namely, we show that $\mathcal{C}(K,p)$ is a class of pattern-avoiding permutations. In the case $p = 1$ (Section 2.2), we give a precise description of the basis B of excluded patterns: $B = \{321, 3142, 2143\} \cup D$, D being the set of all permutations of S_{K+1} that do not start with 1 nor end with $K+1$, and containing exactly one descent. In particular, B is of cardinality $3 + 2^{K-1}$ and contains patterns of size at most $K+1$. For the general case (Section 2.3), we cannot get such a precise result but only a bound on the size of the excluded patterns: we show that $\mathcal{C}(K,p)$ is a class of pattern-avoiding permutations whose basis contains patterns of size at most $(Kp+2)^2 - 2$.

A second point of view is to examine how many steps of a given width are necessary to obtain any permutation of S_n starting from $12\dots n$. Namely in Section 3 we fix a width K (constant, or $K = K(n)$) and a size n and search for the number p such that any permutation of S_n can be obtained from $12\dots n$ in at most p duplication-loss steps of width at most K. We describe an algorithm computing a possible scenario of duplications and losses for any $\pi \in S_n$, this scenario involving $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2})$ duplication-loss steps in the worst case and on average. We also show that $\Omega(\log n + \frac{n^2}{K^2})$ steps are necessary (in the worst case and on average) to obtain any permutation of S_n from $12\dots n$. These upper and lower bounds coincide in most cases.

2 Characterization with excluded patterns

Before focusing on the classes $\mathcal{C}(K,1)$ and $\mathcal{C}(K,p)$ defined for our model, we will get back to the simpler whole genome duplication - random loss model (corresponding to $K = \infty$ in our model, but defined previously by other authors). We will not prove new theorems, but will interprete the existing results from the pattern-avoidance point of view.

2.1 The whole genome duplication - random loss model through the pattern-avoidance prism

Let us recall that in the whole genome duplication - random loss model, any duplication-loss step has cost 1, so that we can consider *w.l.o.g* that the duplicated fragment is the whole permutation at any step. The cost of obtaining a permutation $\sigma \in S_n$ from the identity is just the minimal number of steps of a duplication-loss scenario transforming $12\dots n$ into σ.

A statistics of permutations that matters for our purpose is their number of *descents*.

Definition 4 *Given a permutation σ of size n, we say that there is a* descent *(resp.* ascent*) at position i, $1 \leq i \leq n-1$, if $\sigma_i > \sigma_{i+1}$ (resp. $\sigma_i < \sigma_{i+1}$). We write $desc(\sigma)$ the number of descents of the permutation σ.*

Example 5 *For example, $\sigma = 524316$ has 3 descents, namely at positions 1, 3 and 4.*

A permutation σ of size n has at most $n-1$ descents, the case of $n-1$ descents exactly corresponding to the reversed identity permutation $n(n-1)\dots 21$. It is also of common knowledge that the average number of descents among permutations of size n is $\frac{n-1}{2}$.

In (4), the authors prove the following theorem.

Theorem 6 *Let $\sigma \in S_n$. In the whole genome duplication - random loss model, $\lceil \log_2(desc(\sigma)+1) \rceil$ steps are necessary and sufficient to obtain σ from $12\dots n$.*

It is equivalent to say that the permutations that can be obtained in at most p steps in the whole genome duplication - random loss model are exactly those whose number of descents is at most $2^p - 1$.

Now, we can notice that the property of being obtainable in at most p steps is

stable for the pattern-involvement relation $\prec$: if σ can be obtained in at most p steps, and if $\pi \prec \sigma$, then π can also be obtained in at most p steps. Indeed, it is enough to perform the same duplication-loss scenario on σ, keeping track only of the elements of σ that form an occurrence of π. This stability for $\prec$ implies that the class of permutations obtainable in at most p steps is a class of pattern-avoiding permutations, whose excluded patterns are the minimal (again in the sense of $\prec$) permutations that cannot be obtained in p steps.

Then, by Theorem 6, the excluded patterns are the minimal permutations with 2^p descents. We initiated a study of the minimal permutations with d descents in (3). However, it is simple to notice that a permutation with d descents and minimal for this criterion has size at most $2d$, since it does not contain to consecutive ascents by minimality. An immediate consequence is that the number of excluded patterns is finite.

This allows us to state the following version of Theorem 6:

Theorem 7 *The permutations that can be obtained in at most p steps in the whole genome duplication - random loss model form a class of pattern-avoiding permutations. The excluded patterns are the permutations with exactly 2^p descents that are minimal (in the sense of $\prec$) for this criterion. These excluded patterns are in finite number.*

In (3), we will give a simpler description and some properties of these minimal permutations with d descents.

2.2 Permutations obtained in one step of width K

As an introduction to the study of $\mathcal{C}(K, p)$, we deal in this section with the simpler case of the class $\mathcal{C}(K) = \mathcal{C}(K, 1)$ of permutations obtained from $12 \dots n$ in one duplication-loss step of width at most K. Assume in this section that the parameter $K \geq 2$ is fixed. Throughout this section, when referring to a duplication-loss step, we always mean duplication-loss step of width K, except when otherwise explicitly stated.

It is easily noticed that any permutation of $\mathcal{C}(K)$ cannot have more than one descent. Conversly, any permutation of size at most K having exactly one descent belongs to $\mathcal{C}(K)$.

Although it is a technical point of importance in the proof of Theorem 9, the following proposition comes straightforward:

Proposition 8 *The permutations of size $K+1$ that do not belong to $\mathcal{C}(K)$ and having exactly one descent are exactly those of S_{K+1} with one descent that*

do not start with 1 *nor end with* $K+1$.

PROOF. Let $\sigma = \sigma_1\sigma_2\ldots\sigma_{K+1}$ be a permutation of size $K+1$ that does not belong to $\mathcal{C}(K)$ but has exactly one descent. Now, if $\sigma_1 = 1$, then $\overline{\sigma} = \sigma_2\ldots\sigma_{K+1}$ is a permutation (of $\{2,3,\ldots,K+1\}$) of size K having one descent, and therefore $\overline{\sigma}$ can be obtained from $23\ldots K+1$ in one duplication-loss step. Applying the same transformation to $123\ldots K+1$ will then produce σ, contradicting that $\sigma \notin \mathcal{C}(K)$. The same reasoning holds when $\sigma_{K+1} = K+1$. So σ does not start with 1 nor end with $K+1$.

Now if σ is a permutation of size $K+1$ having exactly one descent, that does not start with 1 nor end with $K+1$, we claim that σ cannot be obtained from $12\ldots K+1$ in one duplication-loss step. This is because no duplication-loss step of width K can move both 1 and $K+1$ in $12\ldots K+1$.

Theorem 9 *The class* $\mathcal{C}(K)$ *of permutations obtained from* $12\ldots n$ *(for some* $n \geq 1$*) in one duplication-loss step of width* K *is a class* $S(B)$ *of pattern-avoiding permutations whose basis* B *is composed of* $3 + 2^{K-1}$ *patterns of size at most* $K+1$*. Namely* $B = \{321, 3142, 2143\} \cup D$*,* D *being the set of all permutations of* S_{K+1} *that do not start with* 1 *nor end with* $K+1$*, and containing exactly one descent.*

Example 10 $\mathcal{C}(4) =$
$S(321, 3142, 2143, 23451, 23514, 24513, 34512, 25134, 35124, 45123, 51234)$

PROOF. We prove the reversed statement: $\sigma \notin S(B)$ if and only if σ cannot be obtained from an identity permutation in one duplication-loss step of width K.

Assume $\sigma \notin S(B)$. Then there exists $b \in B$ such that $b \prec \sigma$. If $b = 321$, 3142 or 2143, then σ has at least 2 descents and cannot be obtained in one duplication-loss step. Otherwise, using Proposition 8, there exists $\rho \in S_{K+1}$ such that $\rho \prec \sigma$ and $\rho \notin \mathcal{C}(K)$. Now if σ could be obtained in one duplication-loss step, then so would be ρ, yielding a contradiction. So $\sigma \notin \mathcal{C}(K)$.

Conversly, assume that $\sigma \notin \mathcal{C}(K)$. If σ contains at least 2 descents, then σ contains an occurrence of 321 or 3142 or 2143, since these three are the minimal permutations (in the sense of the relation $\prec$) with 2 descents. And consequently, $\sigma \notin S(B)$. Thus we may assume that σ has exactly one descent. We decompose $\sigma \in S_n$ into $\sigma = 12\ldots p_1\widehat{\sigma}p_2(p_2+1)\ldots n$, where $\widehat{\sigma}$ is a permutation of the set $\{p_1+1, p_1+2\ldots, p_2-1\}$ that does not start with p_1+1 nor end with p_2-1, and contains exactly one descent. This decomposition is shown in Figure 3. We denote by $\widehat{K}$ the size of $\widehat{\sigma}$. Since $\sigma \notin \mathcal{C}(K)$, necessarily $\widehat{K} \geq K+1$ or we would get a contradiction. If $\widehat{K} = K+1$, we get that $\widehat{\sigma}$ is

an occurrence of some pattern of $D \subset B$ in σ. As a consequence, $\sigma \notin S(B)$. What is left to prove is that this extends to the case $\widehat{K} > K + 1$. We just need to show that we can remove elements in $\widehat{\sigma}$ without violating any of the properties below:

- the permutation does not start with its smallest element
- the permutation does not end with its greatest element
- the permutation has exactly one descent

until we get a permutation of size $K+1$. At that point $\widehat{\sigma}$ contains an occurrence of a pattern in D, and so does σ, and we get that $\sigma \notin S(B)$. Now, because of the conditions on $\widehat{\sigma}$, the only descent in $\widehat{\sigma}$ necessarily goes from the greatest to the smallest element in $\widehat{\sigma}$, ensuring that it is possible to remove elements without violating any of the properties above (see Figure 3).

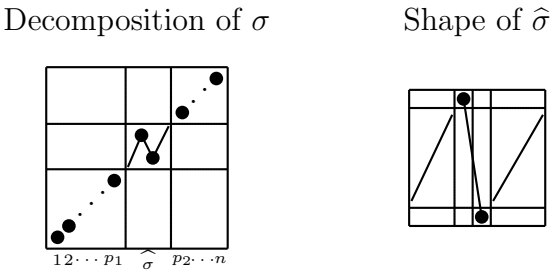


Fig. 3. Decomposition $\sigma = 12\ldots p_1\widehat{\sigma}p_2(p_2+1)\ldots n$ on the graphical representation of σ, and shape of $\widehat{\sigma}$

2.3 Permutations obtained in p steps of width K

As for the case of $\mathcal{C}(K,1)$ in Section 2.2, we prove (Theorem 19) in this section that the class $\mathcal{C}(K,p)$ of all permutations obtained from an identity permutation after p duplication-loss steps of width at most K is a class of pattern-avoiding permutations. However, we do not get a precise description of the basis of this class, but only an upper bound on the size of the excluded patterns. As in the previous section, when referring to a duplication-loss step, we always mean duplication-loss step of width K, except when otherwise explicitely stated.

To prove the announced result, we will need a few more notations and technical lemmas.

The *vector* from i to j in a permutation σ consists of all elements whose positions lie between the positions of i and j, i and j being included. The *size* of a vector is the number of elements in it. For example, the vector from 7 to 2 in the permutation 4123576 is $\overleftarrow{2357}$, and has size 4.

Definition 11 *Let σ be a permutation of S_n. The* value-position vector associated with $i \in [1..n]$ *(vp-*vector *for short) is the vector of σ going from i to σ_i, if i is not a fixpoint of σ. In the case $i = \sigma_i$, the vp-vector associated with i is empty.*

It should appear in this definition that the *vp*-vector associated with i, going from the element of σ which has value i to the element of σ at position i, represents the necessary move for i to reach its position in the sorted permutation $12\ldots n$. As it can be seen on Figure 4, on the graphical representation of permutations used throughout the paper, the *vp*-vector associated with i is an arrow going horizontally from the element at ordinate i to the diagonal.

We can also notice that a non-empty *vp*-vector contains at least two elements.

To take into account all the moves necessary to sort σ to $12\ldots n$, it is convenient to introduce the *value-position domain*:

Definition 12 *Let σ be a permutation of S_n. The* value-position domain *of σ (vp-*domain *for short) is composed of all elements of σ appearing in at least one vp-vector.*

These two definitions are illustrated on Figure 4.

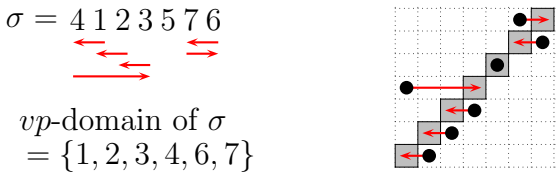


Fig. 4. *vp*-vectors and *vp*-domain for $\sigma = 4123576$, in the usual and in the graphical representations

Now, observe that for any permutation, the *vp*-vectors are reversible in the sense that reversing all the arrows will give a set of vectors that represent the moves of elements that are necessary to "unsort" $12\ldots n$ into σ. It is easily seen from Definitions 11 and 12 and this remark that for any permutation $\sigma \in \mathcal{C}(K, p)$, any element belonging to the *vp*-domain of σ also belongs to at least one of the duplication-loss steps used to obtain σ from $12\ldots n$. Consequently, the *vp*-domain of σ contains at most Kp elements.

Lemma 13 *Consider a permutation σ, and the permutation τ obtained from σ by the removal of some element j. Then for any element $i \neq j$ such that $i \neq \sigma_i$, either this element becomes a fixpoint in τ or the size of the vp-vector associated with this element in τ remains constant, is increased of 1 or is diminished of 1 with respect to the size of the vp-vector associated with i in σ.*

PROOF. It is easily seen on the graphical representation of σ. Any element that does not lie just above or just below the diagonal cannot become a fixpoint when removing an element j. For elements that do not becom fixpoints, the horizontal distance to the diagonal can only change of 0, 1 or -1 when removing some element j (see Figure 5).

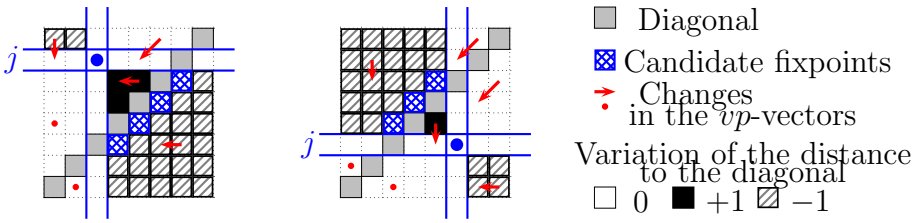


Fig. 5. Variation of the size of *vp*-vectors due to the removal of an element j above or below the diagonal.

Lemma 14 *For any permutation σ, there is at least one element j such that the permutation τ obtained from σ by the removal of j contains at most one more fixpoint than σ.*

PROOF. It is convenient to introduce the *quasi-diagonal elements* of σ, defined as follows. i is a quasi-diagonal element of σ if $\sigma_{i-1} = i$ or $\sigma_{i+1} = i$. These two cases correspond respectively to elements of σ lying just above or just below the diagonal in the graphical representation of σ. Any element of σ that may become a fixpoint in τ is necessarily a quasi-diagonal element.

If there is no quasi-diagonal element, then we can remove any element j to obtain a permutation τ that does not have more fixpoints than σ. If there are some, then we pick j among the quasi-diagonal elements. We claim that at most one fixpoint is create while removing j. The argument is simple. Suppose j is such that $\sigma_{j-1} = j$, the other case being similar. Then the only fixpoint that may appear is $j - 1$, if $\sigma_j = j - 1$. This should appear clearly on Figure 6.

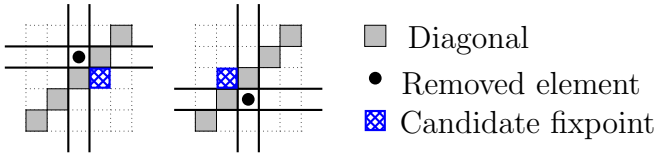


Fig. 6. The only fixpoint that can appear when removing a quasi-diagonal element.

Lemma 15 *Consider a permutation $\sigma \notin \mathcal{C}(K,p)$ such that for any strict pattern τ of σ, $\tau \in \mathcal{C}(K,p)$. Then the vp-domain of σ is of size at most $2Kp+2$.*

PROOF. By Lemma 14, we can choose some $\tau \prec \sigma$ with $|\tau|+1 = |\sigma|$ and such that τ has at most one more fixpoint than σ. Call j the element deleted in σ to obtain τ. By a previous remark, since $\tau \in \mathcal{C}(K,p)$, the *vp*-domain of τ is of size at most Kp, and is therefore composed of at most Kp *vp*-vectors. Each of these *vp*-vectors in τ yields a *vp*-vector in σ, whose size is smaller or equal or possibly increased by 1. Let us denote by $\overrightarrow{V}$ the set of *vp*-vectors of σ obtained from a *vp*-vector of τ. Then the number of elements of σ that belong to a *vp*-vector of $\overrightarrow{V}$ is at most $2Kp$. However $\overrightarrow{V}$ is not yet the *vp*-domain of σ. We must complete it with up to two *vp*-vectors: the one associated with the element j deleted, and the one associated with the fixpoint of τ that was not a fixpoint in σ, if such a point exists. If such an element exists, then it is a quasi-diagonal element in σ and its *vp*-vector (denoted $\overrightarrow{v}$) in σ is necessarily of size 2, so that $\overrightarrow{V} \cup \{\overrightarrow{v}\}$ has total size at most $2Kp+2$. Now it is easily observed that any element of σ belonging to one *vp*-vector necessarily belongs to at least two *vp*-vectors (this can be seen as a "balance condition"). Consequently, all the elements of the *vp*-vector associated with j are already covered by a vector of $\overrightarrow{V} \cup \{\overrightarrow{v}\}$, so that the *vp*-domain of σ is exactly the set of elements covered by $\overrightarrow{V} \cup \{\overrightarrow{v}\}$. Therefore, its size is at most $2Kp+2$.

Lemma 16 *Consider a permutation $\sigma \notin \mathcal{C}(K,p)$ of size $n > (Kp+2)^2 - 2$ such that for any strict pattern τ of σ, $\tau \in \mathcal{C}(K,p)$. Then σ is of the form $\sigma = Ii(i+1)\dots(i+Kp)J$ with I a permutation of $[1..i-1]$ and J a permutation of $[i+Kp+1..n]$. It is possible that I or J is empty.*

PROOF. By Lemma 15, the *vp*-domain of σ is of size at most $2Kp+2$. We can decompose σ into *free windows* of consecutive elements outside the *vp*-domain of σ, separated by windows of consecutive elements of the *vp*-domain. Now, there are at most $Kp+1$ windows of consecutive elements of the *vp*-domain, and consequently, there are at most $Kp+2$ free windows in σ. Since σ is of size $n > (Kp+2)^2 - 2 = (Kp+2)Kp+2Kp+2$, at least one of the free windows of σ has size strictly greater than Kp, *i.e.,* contains at least $Kp+1$ elements. By definition, these elements do not belong to the *vp*-domain of σ, and hence they allow the decomposition of σ into $\sigma = Ii(i+1)\dots(i+Kp)J$ with I a permutation of $[1..i-1]$ and J a permutation of $[i+Kp+1..n]$. Figure 7 represent the decomposition of σ used in this proof.

Lemma 17 *Consider a permutation $\sigma = \sigma'(j+1)(j+2)\dots n$ where σ' is a permutation of $[1\dots j]$. If σ is obtainable after p duplication steps of size at*

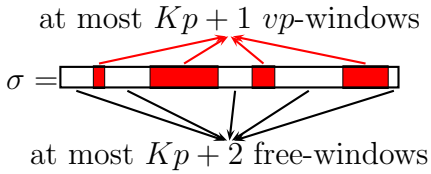


Fig. 7. Proof of Lemma 16

most K then σ is obtainable after p duplication steps of size at most K such that the duplicated window for each step does not intersect

PROOF. The key idea is to consider the first sequence $s_1, s_2, \ldots, s_p$ of duplication-loss steps and create a new sequence $s'_1, s'_2, \ldots, s'_p$ such that :

- Each step s'_i concerns only elements of $[1..j]$.
- After every step s'_i, the elements $1, 2, \ldots, j$ are in the same order than after performing steps $s_1, s_2, \ldots, s_i$.

Then the proof is by induction on the number of steps. If there is only one step then the proof is straighforward. Suppose now that the above statement is true until $p-1$ steps. Then for the last step, we use our hypothesis for $p-1$ so that we have operations $s'_1, s'_2, \ldots, s'_{p-1}$ respecting the above conditions. For s'_n, only notice that the elements of $[1 \ldots j]$ involved in s_n are also in a window of size K in the permutation obtained after s'_{j-1} and in the same relative order by our induction hypothesis which proves the existence of s'_n.

Using these lemmas, we state and prove a key proposition:

Proposition 18 *Consider a permutation $\sigma \notin \mathcal{C}(K,p)$. Then either σ is of size at most $(Kp+2)^2-2$, or there exists a strict pattern τ of σ that does not belong to $\mathcal{C}(K,p)$.*

PROOF. Consider a permutation $\sigma \notin \mathcal{C}(K,p)$ such that any strict pattern τ of σ belongs to $\mathcal{C}(K,p)$. We want to show that σ is of size $n \leq (Kp+2)^2-2$. Let us assume the contrary. By Lemma 16, there exist $i \in [1..n]$, I a permutation of $[1..i-1]$ and J a permutation of $[i+Kp+1..n]$ such that $\sigma = Ii(i+1)\ldots(i+Kp)J$. Let us denote $\widehat{\sigma}$ the permutation $\widehat{\sigma} = Ii(i+1)\ldots(i+Kp-1)(J-1)$, where $(J-1)$ is the permutation of $[i+Kp..n-1]$ obtained from J by subtracting 1 to every element of J. $\widehat{\sigma}$ is a strict pattern of σ, hence $\widehat{\sigma} \in \mathcal{C}(K,p)$. Consider a shortest sequence of duplication-loss steps of width at most K that produces $\widehat{\sigma}$ from $12\ldots(n-1)$. This sequence has at most p steps, each of width at most K. It implies that the total distance crossed by the elements that are duplicated is at most Kp. Consequently, it is not possible to bring an element of I and an element of $J-1$ in two consecutive positions. So it is necessary that the duplication-loss steps of the scenario we

consider are *internal* to I and $J-1$. We can reproduce these steps in I and J to obtain σ from $12\dots n$ in at most p duplication-loss steps of width at most K, contradicting that $\sigma \notin \mathcal{C}(K,p)$.

It is then quite easy to prove Theorem 19:

Theorem 19 *The class $\mathcal{C}(K,p)$ of all permutations obtained from an identity permutation after p duplication-loss steps of width at most K is a class of pattern-avoiding permutations whose basis is finite and contains only patterns of size at most $(Kp+2)^2-2$.*

PROOF. We set $B = \{\pi : \pi \notin \mathcal{C}(K,p) \text{ and } |\pi| \leq (Kp+2)^2-2\}$ and show that $S(B) = \mathcal{C}(K,p)$.

Consider $\sigma \notin \mathcal{C}(K,p)$. If $|\sigma| \leq (Kp+2)^2-2$, then $\sigma \in B$ and $\sigma \notin S(B)$. Otherwise, if $|\sigma| > (Kp+2)^2-2$, then by Proposition 18, there exists a strict pattern τ of σ that does not belong to $\mathcal{C}(K,p)$. Reasoning by induction on the size of the permutations, we deduce from $\tau \notin \mathcal{C}(K,p)$ that $\tau \notin S(B)$. A direct consequence is that $\sigma \notin S(B)$. This proves that $S(B) \subseteq \mathcal{C}(K,p)$.

Conversely, consider $\sigma \in \mathcal{C}(K,p)$. Then any pattern τ of σ is also obtainable from an identity permutation in at most p steps of width at most K (using the sequence of duplication-loss steps associated with σ), *i.e.,* $\tau \in \mathcal{C}(K,p)$. Then σ does not contain an occurrence of any pattern of B, *i.e.,* $\sigma \in S(B)$. This shows that $\mathcal{C}(K,p) \subseteq S(B)$, ending the proof of the theorem.

3 Number of steps of width K to obtain any permutation of size n

The *whole genome duplication - random loss* model is studied in (4), and the authors describe a method to compute an optimal duplication-loss scenario, *i.e.,* a scenario of duplications (of the whole genome in this case) and losses whose number of steps is minimal.

Our model with bounded size duplication operations reduces to the *whole genome duplication - random loss* case when $K = n$ and thus to a radix-sort algorithm as shown in (4) and to a bubble-sort when $K = 2$. Thus we give some algorithm whose complexity matches the two extremal cases and shows some continuity between the two sorting algorithms.

It is worth noticing that any scenario in our model can be viewed as a whole genome duplication - random loss scenario. Consequently, the number of steps of an optimal whole genome duplication - random loss scenario is a lower

bound to the number of steps of an optimal scenario in our duplication-loss model.

It is also easy to see that, when considering permutations of size at most K, our model and the whole genome duplication - random loss model coincide. Indeed, we will use for our purpose the procedure of (4), which is given in Algorithm 1. We omit the proof of correctness and optimality of this algorithm. See (4) for details.

Algorithm 1 An optimal whole genome duplication - random loss scenario from $12\dots K$ to $\sigma \in S_K$

1: $\pi = 12\dots K$
2: Partition σ into maximal increasing substrings, from left to right
3: Each element of $[1..K]$ appearing in the i^{th} maximal increasing substring gets as a label the binary representation of i
4: **for** $j = 1$ to $\lceil \log_2(desc(\sigma) + 1) \rceil$ **do**
5: Perform a duplication-loss step on π that keeps in the first copy of π exactly the elements whose label has a 0 in its j^{th} least significant bit
6: **end for**

In order to examine every bit of the labels given to the elements of $[1..K]$, the number of steps in the loop on line 4 is $\lceil \log_2(\text{number of maximal increasing substrings of } \sigma) \rceil = \lceil \log_2(desc(\sigma) + 1) \rceil$. A consequence is that the number of steps in an optimal whole genome duplication - random loss scenario from $12\dots n$ to σ is $\Theta(\log n)$ in the worst case and on average (see equation (1) for the average case).

Note that the same algorithm can be used to compute an optimal whole genome duplication - random loss scenario from $i_1 i_2 \dots i_k$, with $k \leq K$ and $i_1 < i_2 < \dots < i_k$, to any permutation of $\{i_1, i_2, \dots, i_k\}$.

3.1 Upper bound

In this section, we provide an algorithm that computes, for any permutation $\sigma \in S_n$ in input, a possible scenario of duplications and losses to obtain σ from $12\dots n$. We will restrict ourselves to duplication-loss steps of width at most K, so that the number of duplication-loss steps corresponds to the cost of the scenario in our cost model. We are interested in the number of duplication-loss steps of the scenario produced by the algorithm, in the worst case, and on average. It provides an upper bound on the number of duplication-loss steps that are necessary to obtain a permutation. The algorithm we use is described in Algorithm 2.

A few keys to understand Algorithm 2 are the following remarks.

Algorithm 2 A duplication-loss scenario from $12\dots n$ to $\sigma \in S_n$

1: $\pi \leftarrow 12\dots n$
2: **for** $i = 1$ to $\lceil \frac{n-K}{\lfloor K/2 \rfloor} \rceil$ **do**
3: Let $L^i = \{\sigma_j : n - i\lfloor K/2 \rfloor + 1 \leq j \leq n - (i-1)\lfloor K/2 \rfloor\}$
4: Perform duplication-loss steps on π to move from left to right the elements of L^i to the positions $n - i\lfloor K/2 \rfloor + 1$ to $n - (i-1)\lfloor K/2 \rfloor$ of π, without changing their respective order
5: **end for**
6: **for** $i = 1$ to $\lceil \frac{n-K}{\lfloor K/2 \rfloor} \rceil$ **do**
7: Perform Algorithm 1 on the window of π between the indices $n - i\lfloor K/2 \rfloor + 1$ and $n - (i-1)\lfloor K/2 \rfloor$
8: **end for**
9: Perform Algorithm 1 on the window of π between the indices 1 and $n - \lceil \frac{n-K}{\lfloor K/2 \rfloor} \rceil \lfloor K/2 \rfloor$

The set L^i of values defined at line 3 represents the rightmost $\lfloor K/2 \rfloor$ elements of σ not yet examined. The algorithm consists in two different loops, the first one corresponding to lines 2 to 5 of the algorithm and the second one from line 6 to 8. At the end of the first loop (line 5), π is decomposed into windows of width $\lfloor K/2 \rfloor$ (except the leftmost one which is of width at most K) ; and each of these windows is an increasing sequence containing exactly the same elements as the window of σ corresponding to the same indices. In the second loop, we consider these windows from right to left and since there are of width less than K, we can call Algorithm 1 (that implements whole genome duplication-random loss) on each window successively to transform π into σ.

An example is given with $\sigma = 2\,10\,1\,7\,6\,5\,8\,9\,3\,4$ and $K = 6$. We first cut σ in chunks of size $\lfloor K/2 \rfloor = 3$ and obtain $2\,10\,1\,|\,7\,6\,5\,|\,8\,9\,3\,|\,4$. Then the first loop of the algorithm (step 2 to 5) starts from $1\,2\,3\,4\,5\,6\,7\,8\,9\,10$ and takes the elements in increasing order to the same chunk the belong to in σ. This gives $1\,2\,10\,|\,5\,6\,7\,|\,3\,8\,9\,|\,4$. Then the second loop sorts each chunk separately to obtain σ using the radix sort Algorithm 1 introduced in (4).

Notice here that we use in the second loop (except for the leftmost window) only duplication-loss steps of width $\lfloor K/2 \rfloor$. An improvement we considered is to use whole genome duplication - random loss scenarios on windows of width K, that are nonetheless increasing sequences. Unfortunately, we were not able to analyse how many duplication-loss steps there are in a scenario produced by such an algorithm.

We now analyse the number of steps of the scenario produced by Algorithm 2.

Proposition 20 *The number of duplication-loss steps of a scenario produced*

by Algorithm 2 on a permutation of size n is at most $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2})$ asymptotically.

PROOF. Suppose we are at iteration i of the first loop. We have to move the $\lfloor K/2 \rfloor$ elements of L^i to their positions (from $n-i\lfloor K/2\rfloor+1$ to $n-(i-1)\lfloor K/2\rfloor$) by duplication-loss steps of width at most K. The worst situation is when the elements of L^i are at the begining of π. But in this case, we can move to the right the elements of L^i by $\lceil K/2 \rceil$ positions at every duplication-loss step, until they reach their position. The total number of duplication-loss steps in this first process is then at most

$$\sum_{i=1}^{\lceil \frac{n-K}{\lfloor K/2 \rfloor} \rceil} \left\lceil \frac{n - i\lfloor K/2 \rfloor}{\lceil K/2 \rceil} \right\rceil = \Theta(\frac{n^2}{K^2}).$$

Consider now the second loop of Algorithm 2. In each window of size $\lfloor K/2 \rfloor$, it performs at most $\lceil \log \lfloor K/2 \rfloor \rceil$ duplication-loss steps (line 7) and in the leftmost window (line 9), at most $\lceil \log K \rceil$ by the result of (4). Consequently the number of duplication-loss steps produced by the second loop is

$$\left\lceil \frac{n-K}{\lfloor K/2 \rfloor} \right\rceil \lceil \log \lfloor K/2 \rfloor \rceil + \lceil \log K \rceil = \Theta(\frac{n}{K}\log K).$$

We finally get that the total number of duplication-loss steps in a scenario produced by Algorithm 2 is at most $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2})$ asymptotically in the worst case.

It is easily noticed that this worst case corresponds to the *reversed identity* permutation $n(n-1)\ldots 21$. This corresponds to our intuition of a worst case situation in this context. We can also notice that $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2}) = \Theta(\frac{n^2}{K^2})$ for "small" values of K, namely as long as $K = o(\frac{n}{\log n})$. If on the contrary $\frac{n}{\log n} = o(K)$ then $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2}) = \Theta(\frac{n}{K}\log K)$. When $K = \Theta(\frac{n}{\log n})$, the two terms are of the same order.

We can also compute the average number of duplication-loss steps of a scenario produced by Algorithm 2.

Proposition 21 *The number of duplication-loss steps of a scenario produced by Algorithm 2 on a permutation of size n is on average $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2})$ asymptotically.*

PROOF. First, we introduce a few notations. Consider σ a permutation of size n, and decompose it from right to left into $p = \left\lceil \frac{n-K}{\lfloor K/2 \rfloor} \right\rceil + 1$ windows of

width $\lfloor K/2 \rfloor$, except the leftmost one, whose width is $n - \left\lceil \frac{n-K}{\lfloor K/2 \rfloor} \right\rceil \lfloor K/2 \rfloor \leq K$. We denote $\sigma = \sigma^1 \sigma^2 \dots \sigma^p$ this decomposition.

Now, let us denote $c(\sigma)$ the number of duplication-loss steps produced in the first loop of Algorithm 2 on σ. And in particular, we denote $c_p(\sigma)$ the number of such steps produced by the first iteration of this loop, *i.e.*, the number of steps to move the elements of L^1 at the end of the permutation. For computing the average number of such steps, we consider $u_n = \sum_{\sigma \in S_n} c(\sigma)$. It is simple to conceive that

$$\begin{aligned} u_n &= \sum_{\sigma \in S_n} c_p(\sigma) + c(\sigma^1 \dots \sigma^{p-1}) \\ &= \sum_{\sigma \in S_n} c_p(\sigma) + n(n-1)\dots(n - \lfloor K/2 \rfloor + 1) \sum_{\sigma \in S_{n-\lfloor K/2 \rfloor}} c(\sigma) \\ &= \sum_{\sigma \in S_n} c_p(\sigma) + \frac{n!}{(n - \lfloor K/2 \rfloor)!} u_{n-\lfloor K/2 \rfloor}. \end{aligned}$$

Let us focus on $\sum_{\sigma \in S_n} c_p(\sigma)$. Figure 8 should convince the reader that

$$\frac{n + 1 - \lfloor K/2 \rfloor - \min(\sigma^p)}{K} \leq c_p(\sigma) \leq \frac{n + 1 - \lfloor K/2 \rfloor - \min(\sigma^p)}{\lfloor K/2 \rfloor}.$$

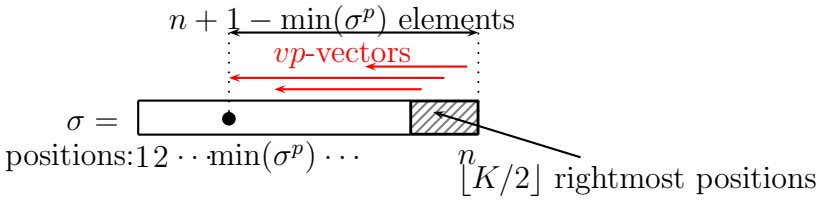


Fig. 8. Bounding $c_p(\sigma)$

Now, we can notice that the number of permutations σ of size n such that $\min(\sigma^p) = i$ is $\binom{n-i}{\lfloor K/2 \rfloor - 1} (n - \lfloor K/2 \rfloor)! \lfloor K/2 \rfloor !$. This yields

$$
\begin{aligned}
&\sum_{\sigma\in S_n} n+1-\lfloor K/2\rfloor-\min(\sigma^p)\\
&=\sum_{i=1}^{n-\lfloor K/2\rfloor+1}(n+1-\lfloor K/2\rfloor-i)\binom{n-i}{\lfloor K/2\rfloor-1}(n-\lfloor K/2\rfloor)!\lfloor K/2\rfloor!\\
&=(n-\lfloor K/2\rfloor)!\lfloor K/2\rfloor!\sum_{i=\lfloor K/2\rfloor-1}^{n-1}(i+1-\lfloor K/2\rfloor)\binom{i}{\lfloor K/2\rfloor-1}\\
&=(n-\lfloor K/2\rfloor)!\lfloor K/2\rfloor!\lfloor K/2\rfloor\sum_{i=\lfloor K/2\rfloor}^{n-1}\binom{i}{\lfloor K/2\rfloor}\\
&=(n-\lfloor K/2\rfloor)!\lfloor K/2\rfloor!\lfloor K/2\rfloor\binom{n}{\lfloor K/2\rfloor+1}.
\end{aligned}
$$

Consequently,

$$
\begin{aligned}
\sum_{\sigma\in S_n} c_p(\sigma)&\leq(n-\lfloor K/2\rfloor)!\lfloor K/2\rfloor!\binom{n}{\lfloor K/2\rfloor+1}\\
\sum_{\sigma\in S_n} c_p(\sigma)&\geq\frac{\lfloor K/2\rfloor}{K}(n-\lfloor K/2\rfloor)!\lfloor K/2\rfloor!\binom{n}{\lfloor K/2\rfloor+1}\\
&\geq\frac{1}{3}(n-\lfloor K/2\rfloor)!\lfloor K/2\rfloor!\binom{n}{\lfloor K/2\rfloor+1},
\end{aligned}
$$

giving after a few computations

$$
\frac{1}{3}\frac{n-\lfloor K/2\rfloor}{\lfloor K/2\rfloor+1}+\frac{u_{n-\lfloor K/2\rfloor}}{(n-\lfloor K/2\rfloor)!}\leq\frac{u_n}{n!}\leq\frac{n-\lfloor K/2\rfloor}{\lfloor K/2\rfloor+1}+\frac{u_{n-\lfloor K/2\rfloor}}{(n-\lfloor K/2\rfloor)!}.
$$

Therefore, we consider two sequences (v_n) and (w_n) satisfying the relations $v_n=\frac{1}{3}\frac{n-\lfloor K/2\rfloor}{\lfloor K/2\rfloor+1}+v_{n-\lfloor K/2\rfloor}$ and $w_n=\frac{n-\lfloor K/2\rfloor}{\lfloor K/2\rfloor+1}+w_{n-\lfloor K/2\rfloor}$ respectively if $n>K$, and $v_n=w_n=\frac{u_n}{n!}$ for any $n\leq K$. Then we have $v_n\leq\frac{u_n}{n!}\leq w_n\forall n\in\mathbb{N}$. We can solve the recurrence equations for v_n and w_n; and if we write $n=\lceil\frac{n-K}{\lfloor K/2\rfloor}\rceil\lfloor K/2\rfloor+r$ (then $\lfloor K/2\rfloor\leq r\leq K$), we get:

$$
\begin{aligned}
v_n&=\frac{1}{3}\sum_{i=1}^{\lceil\frac{n-K}{\lfloor K/2\rfloor}\rceil}\frac{n-i\lfloor K/2\rfloor}{\lfloor K/2\rfloor+1}+v_r\\
&=\frac{1}{3(\lfloor K/2\rfloor+1)}\left\lceil\frac{n-K}{\lfloor K/2\rfloor}\right\rceil\left(n-\lfloor K/2\rfloor\frac{\lceil\frac{n-K}{\lfloor K/2\rfloor}\rceil+1}{2}\right)+v_r\\
&=\Theta(\frac{n^2}{K^2})
\end{aligned}
$$

and

$$w_n = \sum_{i=1}^{\lceil \frac{n-K}{\lfloor K/2 \rfloor} \rceil} \frac{n - i\lfloor K/2 \rfloor}{\lfloor K/2 \rfloor + 1} + w_r = \Theta(\frac{n^2}{K^2})$$

Consequently, the average number of duplication-loss steps produced by the first loop of Algorithm 2 on permutations of size n is $\frac{u_n}{n!} = \Theta(\frac{n^2}{K^2})$.

What is left to compute is the average number of duplication-loss steps produced by the second loop of Algorithm 2 on permutations of size n. This number is given by

$$\begin{aligned}
&\frac{1}{n!} \sum_{\sigma \in S_n} \sum_{i=1}^{p} \lceil \log(desc(\sigma^i) + 1) \rceil \\
&= \frac{1}{n!} \Big(\sum_{i=2}^{p} \sum_{\sigma \in S_n} \lceil \log(desc(\sigma^i) + 1) \rceil + \sum_{\sigma \in S_n} \lceil \log(desc(\sigma^1) + 1) \rceil \Big) \\
&= \frac{1}{n!} \Big(\sum_{i=2}^{p} (n - \lfloor K/2 \rfloor)! \binom{n}{\lfloor K/2 \rfloor} \sum_{\sigma \in S_{\lfloor K/2 \rfloor}} \lceil \log(desc(\sigma) + 1) \rceil \\
&\quad + (n - |\sigma^1|)! \binom{n}{|\sigma^1|} \sum_{\sigma \in S_{|\sigma^1|}} \lceil \log(desc(\sigma) + 1) \rceil \Big) \\
&= \frac{1}{\lfloor K/2 \rfloor!} (p-1) \sum_{\sigma \in S_{\lfloor K/2 \rfloor}} \lceil \log(desc(\sigma) + 1) \rceil \\
&\quad + \frac{1}{|\sigma^1|!} \sum_{\sigma \in S_{|\sigma^1|}} \lceil \log(desc(\sigma) + 1) \rceil.
\end{aligned}$$

Since $p = \left\lceil \frac{n-K}{\lfloor K/2 \rfloor} \right\rceil + 1$, we deduce that the average number of duplication-loss steps produced by the second loop of Algorithm 2 on permutations of size n is $\Theta(\frac{1}{\lfloor K/2 \rfloor!} \left(\left\lceil \frac{n-K}{\lfloor K/2 \rfloor} \right\rceil + 1 \right) \sum_{\sigma \in S_{\lfloor K/2 \rfloor}} \lceil \log(desc(\sigma) + 1) \rceil)$. Hence we focus on the computation of $\frac{1}{k!} \sum_{\sigma \in S_k} \lceil \log(desc(\sigma) + 1) \rceil)$ for $k = \lfloor K/2 \rfloor$. By concavity of the log function, since $\frac{1}{k!} \sum_{\sigma \in S_k} desc(\sigma) + 1 = \frac{k+1}{2}$, we get that

$$\frac{1}{k!} \sum_{\sigma \in S_k} \lceil \log(desc(\sigma) + 1) \rceil \geq \frac{1}{k!} \sum_{\sigma \in S_k} \log(desc(\sigma) + 1) \geq \log(\frac{k+1}{2}).$$

Moreover, it is clear that

$$\frac{1}{k!} \sum_{\sigma \in S_k} \lceil \log(desc(\sigma) + 1) \rceil \leq \lceil \log(k) \rceil,$$

so that we deduce that

$$\frac{1}{k!}\sum_{\sigma\in S_k}\lceil\log(desc(\sigma)+1)\rceil=\Theta(\log(k)). \tag{1}$$

Consequently, the average number of duplication-loss steps produced by the second loop of Algorithm 2 on permutations of size n is $\Theta(\lceil\frac{n-K}{\lfloor K/2\rfloor}\rceil\log(\lfloor K/2\rfloor))=\Theta(\frac{n}{K}\log K)$.

Finally, we end the proof concluding that the total number of duplication-loss steps in a scenario produced by Algorithm 2 on a permutation of size n is $\Theta(\frac{n}{K}\log K+\frac{n^2}{K^2})$ on average.

3.2 Lower bound

It is possible to provide very simple lower bounds on the number of duplication-loss steps necessary to obtain a permutation. These lower bounds are given and proved in Propositions 22 and 23 below. They are tight in most cases, however not in any case. Indeed the upper and lower bounds coincide up to a constant factor whenever K is a constant, or when $K=K(n)$, except when $\frac{n}{\log n}\ll K(n)\ll n$.

Proposition 22 *In the worst case, $\Omega(\log n+\frac{n^2}{K^2})$ duplication-loss steps of width K are necessary to obtain a permutation of S_n from $123\ldots n$.*

PROOF. Let us consider first the number of inversions in a permutation that can create a duplication-loss step s of width K. It is easily seen that these new inversions can only involve two elements of s. Call i the number of elements of s that are kept in the first copy. Then the maximum number of inversions that can be created by s is $i(K-i)\leq\frac{K^2}{4}$. Now, a permutation $\sigma\in S_n$ has up to $\frac{n(n-1)}{2}$ inversions, so that at least $\frac{2n(n-1)}{K^2}$ duplication-loss steps are necessary to transform $123\ldots n$ into σ.

To get the other term of the lower bound, we just refer to the result of (4) recalled at the beginning of this section, namely that $\log n$ steps are necessary in the worst case in the whole genome duplication - random loss model, in which duplication-loss operation are less restricted.

Finally, we get a lower bound of $\Omega(\log n+\frac{n^2}{K^2})$ necessary duplication-loss steps to obtain a permutation of S_n from $123\ldots n$ in the worst case.

Proposition 23 *On average,* $\Omega(\log n + \frac{n^2}{K^2})$ *duplication-loss steps of width* K *are necessary to obtain a permutation of* S_n *from* $123\ldots n$.

PROOF. As before, a duplication-loss step can create at most $\frac{K^2}{4}$ inversions in a permutation. But the average number of inversions in a permutation of S_n is $\frac{n(n-1)}{4}$, so that on average at least $\frac{n(n-1)}{K^2}$ duplication-loss steps are necessary to transform $123\ldots n$ into $\sigma \in S_n$.

Again, (4) provides use with the $\Omega(\log n)$ lower bound, referring to the whole genome duplication - random loss model which is more general than ours, so that this bound applies in our context.

We conclude that a lower bound on the average number of duplication-loss steps necessary to obtain a permutation of S_n from $123\ldots n$ is $\Omega(\log n + \frac{n^2}{K^2})$.

4 Conclusion

We discuss the results of Section 3 on the average (or worst case) number of steps of width K to obtain a permutation of size n. It appears that we could not provide lower bounds that coincide with the upper bounds given by our algorithm, but we claim that they are tight in many cases. Indeed, whenever $K = o(\frac{n}{\log n})$, we get that $\frac{n}{K}\log K = o(\frac{n^2}{K^2})$, and consequently the upper bound can be rewritten as $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2}) = \Theta(\frac{n^2}{K^2})$, which coincide up to a constant factor with the lower bound $\Omega(\log n + \frac{n^2}{K^2}) = \Omega(\frac{n^2}{K^2})$. For the case $K = \Theta(\frac{n}{\log n})$, the same argument holds, but the constant factor between the lower and the upper bound might be much greater. Finally, if $K = \Theta(n)$, then $\Theta(\frac{n}{K}\log K + \frac{n^2}{K^2}) = \Theta(\log n)$ and $\Omega(\log n + \frac{n^2}{K^2}) = \Omega(\log n)$, so that upper and lower bounds coincide again.

On the contrary, when $\frac{n}{\log n} \ll K \ll n$, the upper and lower bounds provided do not coincide. We leave as an open question the problem of finding an algorithm that computes a duplication-loss scenario whose number of steps is optimal (on average and in the worst case) up to a constant factor, when the width K of the duplicated windows satisfies $\frac{n}{\log n} \ll K \ll n$.

Several other questions are still open. First of all neither of our algorithms is optimal for a specific permutation and our results are only optimal asymptotically in average and/or in the worst case. It could be interesting to provide algorithms that produce optimal duplication-loss scenarios on any permutation σ, for $K = K(n)$ in order to provide some continuity between the bubble sort (corresponding to $K = 2$) and the radix sort (corresponding to $K(n) = n$).

References

[1] M.H. Albert, R.E.L. Aldred, M.D. Atkinson, H.P. Van Ditmarsch, C.C. Handley, D.A. Hotlon, and D.J. McCaughan. Compositions of pattern restricted sets of permutations. Technical report, University of Otago, 2004. Technical report number OUCS-2004-12.

[2] 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.

[3] M. Bouvel and E. Pergola. Permutations arising in the duplication-loss model. In preparation.

[4] K. Chaudhuri, K. Chen, R. Mihaescu, and S. Rao. On the tandem duplication-random loss model of genome rearrangement. In *SODA*, SODA, pages 564 – 570, 2006.

[5] M.C. Chen and R.C.T. Lee. Sorting by transpositions based on the first increasing substring concept. In *BIBE '04: Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering*, page 553, Washington, DC, USA, 2004. IEEE Computer Society.

[6] Anthony Labarre. A new tight upper bound on the transposition distance. In Rita Casadio and Gene Myers, editors, *Algorithms in Bioinformatics, 5th International Workshop, WABI 2005, Mallorca, Spain, October 3-6, 2005, Proceedings*, volume 3692 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2005.

[7] Anthony Labarre. New bounds and tractable instances for the transposition distance. *IEEE/ACM Trans. Comput. Biology Bioinform*, 3(4):380–394, 2006.