Theoretical Computer Science 593 (2015) 26–41

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Approximation algorithms for sorting by length-weighted prefix and suffix operations

Carla Negri Lintzmayer [a,*,1], Guillaume Fertin [b], Zanoni Dias [a]

[a] Institute of Computing, University of Campinas, Campinas, São Paulo, 13083-852, Brazil
[b] Laboratoire d'Informatique de Nantes-Atlantique, UMR CNRS 6241, Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3, France

ARTICLE INFO

Article history:
Received 17 March 2015
Received in revised form 14 May 2015
Accepted 22 May 2015
Available online 30 May 2015
Communicated by M. Crochemore

Keywords:
Genome rearrangements
Length-weighted operations
Prefix and suffix operations
Reversals and transpositions
Approximation algorithms

ABSTRACT

The traditional approach for the problems of sorting permutations by rearrangements is to consider that all operations have the same unitary cost. In this case, the goal is to find the minimum number of allowed rearrangements that are needed to sort a given permutation, and numerous efforts have been made over the past years regarding these problems. On the other hand, a long rearrangement (which is in fact a mutation) is more likely to disturb the organism. Therefore, weights based on the length of the segment involved may have an important role in the evolutionary process. In this paper we present the first results regarding problems of sorting permutations by length-weighted operations that consider rearrangement models with prefix and suffix variations of reversals and transpositions, which are the two most common types of genome rearrangements. Our main results are $O(\lg^2 n)$-approximation algorithms for 10 such problems.

1. Introduction

One of the challenges of modern science is trying to understand how evolution happened, considering that new organisms arise from mutations that occurred in others. The evolutionary distance between two organisms may be inferred through the genome rearrangement distance between them, which considers a number of rearrangement events that occurred in the transformation of the genome of one organism into the genome of another. A genome rearrangement is a large scale mutation that modifies the order of the segments of the genome, and two of the most commonly studied types of rearrangement events are the reversal (which inverts the order of a sequence of genes) and the transposition (which exchanges the position of two consecutive sequences of genes).

Due to the parsimony principle, the common approach is to consider that the minimum distance between two genomes, or the minimum number of rearrangements needed to transform one into the other, is a reasonable representation of the evolutionary distance between them [1].

In order to compare the genomes, we represent them as sequences of segments which are shared between them. Computationally, we use permutations for this representation. An unsigned permutation of size n is represented as $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$, where $\pi_i \in \{1, 2, \ldots, n\}$ for all $1 \leq i \leq n$ and $\pi_i \neq \pi_j$ for all $i \neq j$. A signed permutation of size n

* Corresponding author.
E-mail address: carlanl@ic.unicamp.br (C.N. Lintzmayer).
[1] Part of the work was done while at Université de Nantes.

http://dx.doi.org/10.1016/j.tcs.2015.05.039
0304-3975/

is denoted the same way as the unsigned, but $\pi_i \in \{-n, -(n-1), \ldots, -1, 1, 2, \ldots, n\}$ for all $1 \leq i \leq n$ and $|\pi_i| \neq |\pi_j|$ for all $i \neq j$. We consider their *extended* version, in which there are elements $\pi_0 = 0$ and $\pi_{n+1} = n+1$ that are fixed.

We represent the *identity* permutation (the goal of the sorting) as $\iota_n = (1\ 2\ \ldots\ n)$. The *inverse* permutation, π^{-1}, is a permutation for which $\pi^{-1}_{\pi_i} = i$ and it satisfies $\pi^{-1} \cdot \pi = \iota_n$, where "$\cdot$" represents composition between permutations, i.e., $\pi \cdot \sigma = (\pi_{\sigma_1}\ \pi_{\sigma_2}\ \ldots\ \pi_{\sigma_n})$.

We also use permutations to represent rearrangements. Therefore, we can use composition to indicate the occurrence of a rearrangement in a genome. A *reversal* $\rho(i, j)$, for $1 \leq i < j \leq n$, is a rearrangement that inverts the segment that goes from position i to position j, transforming π into $\pi \cdot \rho(i, j) = (\pi_1\ \ldots\ \pi_{i-1}\ \underline{\pi_j\ \pi_{j-1}\ \ldots\ \pi_{i+1}\ \pi_i}\ \pi_{j+1}\ \ldots\ \pi_n)$. A *prefix reversal* $\rho_p(j)$ is a reversal $\rho(1, j)$, $1 < j \leq n$, while a *suffix reversal* $\rho_s(i)$ is $\rho(i, n)$, $1 \leq i < n$.

A *signed reversal* $\bar{\rho}(i, j)$, for $1 \leq i \leq j \leq n$, is a rearrangement that inverts the segment that goes from position i to position j while changing the signs of each element of the segment, transforming π into $\pi \cdot \bar{\rho}(i, j) = (\pi_1\ \ldots\ \pi_{i-1}\ \underline{-\pi_j\ -\pi_{j-1}\ \ldots\ -\pi_{i+1}\ -\pi_i}\ \pi_{j+1}\ \ldots\ \pi_n)$. A *signed prefix reversal* $\bar{\rho}_p(j)$ is a signed reversal $\bar{\rho}(1, j)$, $1 \leq j \leq n$, while a *signed suffix reversal* $\bar{\rho}_s(i)$ is a signed reversal $\bar{\rho}(i, n)$, $1 \leq i \leq n$.

A *transposition* $\tau(i, j, k)$, for $1 \leq i < j < k \leq n+1$, is a rearrangement that exchange the segment that goes from position i to $j-1$ with the segment that goes from position j to $k-1$, transforming π into $\pi \cdot \tau(i, j, k) = (\pi_1\ \ldots\ \pi_{i-1}\ \underline{\pi_j\ \pi_{j-1}\ \ldots\ \pi_{k-1}}\ \underline{\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}}\ \pi_k\ \ldots\ \pi_n)$. A *prefix transposition* $\tau_p(j, k)$ is a transposition $\tau(1, j, k)$, $1 < j < k \leq n+1$, while a *suffix transposition* $\tau_s(i, j)$ is a transposition $\tau(i, j, n+1)$, $1 \leq i < j < n+1$.

Over the past years, many combinatorial problems have been motivated by genome rearrangements. Given a *rearrangement model*, which establishes the rearrangements that are allowed during the sorting, and given a permutation, one wants to find a sequence of *operations* in the rearrangement model that sorts the permutation and has the minimum cost.

Normally, the sorting problems consider that all operations have the same unitary cost. Therefore, as we said before, the traditional approach is to find the minimum *number* of operations needed to sort the permutation. Next, we show some known results over these types of problems.

Sorting by Reversals and Sorting by Transpositions are two well studied NP-hard [2,3] problems and their best-known algorithms have approximation factor of 1.375 [4,5]. On the contrary, Sorting by Signed Reversals is polynomial, as shown by Hannenhalli and Pevzner [6] with an $O(n^4)$ algorithm. Improvements have been made ever since in order to decrease this time so that the best known complexity is $O(n \lg n)$ [7]. Also, there is an $O(n)$ algorithm that is capable of computing the minimum number of reversals without showing the sequence of transformations itself [8].

Walter et al. [9] considered a variation in which transpositions are allowed along with reversals and gave a 3-approximation algorithm for the unsigned version and a 2-approximation algorithm for the signed version. The best known algorithm for the unsigned version, however, is a $2k$-approximation [10] where k is the approximation factor of the algorithm for cycle decomposition of the cycle graph [11]. The best known value for k is $1.4167 + \epsilon$ [12].

Regarding unsigned permutations, Gu et al. [13] considered a third event along with reversals and transpositions, the *transreversal*, which is a transposition in which one of the segments is reversed, and also gave a 2-approximation algorithm when the three are allowed. Lin and Xue [14] yet considered a fourth event that they called *revrev*, which consists in a transposition where the two segments are reversed. When considering the four events, they gave a 1.75-approximation algorithm but the best-known approximation factor is 1.5 [15].

Rearrangements that affect only segments from the beginning of the genome are called *prefix* rearrangements. The Pancake Flipping problem, or Sorting by Prefix Reversals, was recently proved to be NP-hard [16] and the best known algorithm for it has an approximation factor of 2 [17]. The signed version of this problem was introduced by Gates and Papadimitriou [18], its computational complexity remains open, and its best known approximation algorithm also has a factor of 2 [19].

Dias and Meidanis [20] introduced the Sorting by Prefix Transpositions problem, and they gave a 2-approximation algorithm for it, which is the best-known factor to the moment. In 2010, Sharmin et al. [21] considered the variation Sorting by Prefix Reversals and Prefix Transpositions, in which prefix reversals and prefix transpositions are allowed simultaneously, and gave a 3-approximation algorithm. They also considered Sorting by Prefix Reversals and Prefix Transreversals, presenting a 2-approximation algorithm for it. For Sorting by Prefix Reversals and Prefix Transpositions, Dias and Dias [22] recently presented an asymptotic 2-approximation algorithm, which is the best known so far.

Lintzmayer and Dias [23–25] introduced new problems for which signed and unsigned *suffix* rearrangements are allowed along with their prefix versions, and presented approximation algorithms for them.

However, a very long reversal is more likely to disturb the organism and preliminary results show that in some cases the reversals that happened during evolution indeed tend not to be very long [26], which indicates that weights based on the size/length of the segment involved may have an important role in the evolutionary process.

Because of this, some authors [27–32] started to consider length-weighted rearrangements. We consider that a reversal $\rho(i, j)$ has length $\ell = j - i + 1$ and a transposition $\tau(i, j, k)$ has length $\ell = k - i + 1$. Now let $f(\ell)$ be the cost of a rearrangement of length ℓ. If a sequence of q rearrangements sorts a permutation and the lengths of these rearrangements are $\ell_1, \ell_2, \ldots, \ell_q$, then the cost of this sequence is $f(\ell_1) + f(\ell_2) + \ldots + f(\ell_q)$. Let π be a permutation (signed or unsigned) and let β be a rearrangement model. For the variations of length-weighted sorting problems, the goal is to find $c_\beta(\pi)$, which is the minimum cost of a sorting sequence for π that uses operations of β.

The value of $f(\ell)$ is usually considered as ℓ^α [28–30]. Note that the traditional approach is to consider $\alpha = 0$, but in this paper we will consider only $\alpha = 1$. Pinter and Skiena [27] gave an $O(\lg^2 n)$-approximation algorithm for the problem