Matrix Tightness: A Linear-Algebraic Framework for Sorting by Transpositions

Tzvika Hartman[1] and Elad Verbin[2]

[1] Dept. of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel
hartmat@cs.biu.ac.il
[2] School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel
eladv@post.tau.ac.il

Abstract. We study the problems of sorting signed permutations by reversals (SBR) and sorting unsigned permutations by transpositions (SBT), which are central problems in computational molecular biology. While a polynomial-time solution for SBR is known, the computational complexity of SBT has been open for more than a decade and is considered a major open problem.

In the first efficient solution of SBR, Hannenhalli and Pevzner [HP99] used a graph-theoretic model for representing permutations, called the *interleaving graph*. This model was crucial to their solution. Here, we define a new model for SBT, which is analogous to the interleaving graph. Our model has some desirable properties that were lacking in earlier models for SBT. These properties make it extremely useful for studying SBT.

Using this model, we give a linear-algebraic framework in which SBT can be studied. Specifically, for matrices over any algebraic ring, we define a class of matrices called *tight matrices*. We show that an efficient algorithm which recognizes tight matrices over a certain ring, $\mathbb{M}$, implies an efficient algorithm that solves SBT on an important class of permutations, called simple permutations. Such an algorithm is likely to lead to an efficient algorithm for SBT that works on all permutations.

The problem of recognizing tight matrices is also a generalization of SBR and of a large class of other "sorting by rearrangements" problems, and seems interesting in its own right as. We give an efficient algorithm for recognizing tight symmetric matrices over any field of characteristic 2. We leave as an open problem to find an efficient algorithm for recognizing tight matrices over the ring $\mathbb{M}$.

1 Introduction

One of the most promising ways to understand evolution between species is to reconstruct their evolutionary history based on genome rearrangements. In the last decade, a large body of work was devoted to a family of computational problems, called *genome rearrangement problems*. Genomes are represented by permutations, where each element stands for a gene. The basic task is, given two

F. Crestani, P. Ferragina, and M. Sanderson (Eds.): SPIRE 2006, LNCS 4209, pp. 279–290, 2006.

permutations, to find a shortest sequence of rearrangement operations (such as reversals, transpositions, translocations, etc.) that transforms one permutation into the other. Assuming (without loss of generality) that one of the permutations is the identity permutation, the problem is to find the shortest way of sorting a permutation using a given rearrangement operation (or set of operations). In this paper we mainly address the problem of sorting signed permutations by reversals and the problem of sorting (unsigned) permutations by transpositions. For more background on genome rearrangements refer to [Pev00, SEM02].

A *signed permutation* is a permutation with + or − on every element, which represent the direction of the corresponding gene. A *reversal* reverses the order of the elements in a segment and flips their signs. The problem of *sorting signed permutations by reversals* (SBR) is the problem of transforming a given signed permutation to the positive identity permutation using a minimum number of reversals.

A *transposition* is a rearrangement operation in which a segment is cut out of the permutation and pasted in a different location. The problem of *sorting unsigned permutations by transpositions* (SBT) is the problem of transforming a given unsigned permutation to the identity permutation using a minimum number of transpositions.[1]

Hannenhalli and Pevzner, in their seminal paper [HP99], gave a polynomial time algorithm for SBR. Subsequent works gave algorithms with better running times, and simplified the underlying theory [BH96, KST00, Ber01, KV03, TS04]. The computational complexity of SBT on the other hand is still open. There are several 1.5-approximation algorithms [BP98, Chr99, HS06], and the best algorithm to date has approximation ratio 1.375 [EH05].

To obtain a polynomial time algorithm for SBR, Hannenhalli and Pevzner used a labelled graph called the *interleaving graph* [HP99]. Each vertex of this graph is labelled either *black* or *white.* The interleaving graph models the effect of a reversal on a permutation as a graph operation on a vertex. In this operation, which we call *clicking a black vertex* v, we eliminate v while (1) replacing the subgraph induced by the neighbors of v by its complement, and (2) flipping the color of each neighbor of v.

There is a basic lower bound for the reversal distance called the *cycle lower bound* [BP96]. A central subproblem of SBR is to characterize the permutations whose reversal distance is equal to the cycle lower bound. We call these permutations *tight.* Hannenhalli and Pevzner proved that a permutation is tight if and only if each connected component of its interleaving graph contains a black vertex. This leads to an efficient algorithm for finding a minimum sorting sequence for tight permutations. They also showed how to find a minimal sorting sequence for a permutation which is not tight.

We believe that a similar approach should be used to solve SBT. Indeed there are models for SBT that try to capture the effect of performing a transposition in a way similar to the clicking operation in the interleaving graph model (e.g.

[1] See Section 3 for an explanation of why we study SBR on *signed* permutations, while SBT is studied on *unsigned* permutations.