Journal of Discrete Algorithms 8 (2010) 273–281

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Weighted LCS

Amihood Amir [a,b,*,1], Zvi Gotthilf [a], B. Riva Shalom [a]

[a] Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel
[b] Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, United States

ARTICLE INFO

Article history:
Received 7 October 2008
Accepted 1 February 2010
Available online 19 February 2010

Keywords:
String algorithms
Approximation algorithms
NP-hard problem
Position weight matrix

ABSTRACT

The Longest Common Subsequence (LCS) of two strings A, B is a well studied problem having a wide range of applications. When each symbol of the input strings is assigned a positive weight the problem becomes the *Heaviest Common Subsequence* (*HCS*) problem. In this paper we consider a different version of weighted LCS on *Position Weight Matrices* (*PWM*). The Position Weight Matrix was introduced as a tool to handle a set of sequences that are not identical, yet, have many local similarities. Such a weighted sequence is a 'statistical image' of this set where we are given the probability of every symbol's occurrence at every text location. We consider two possible definitions of LCS on PWM. For the first, we solve the LCS problem of z sequences in time $O(zn^{z+1})$. For the second, we consider the log-probability version of the problem, prove $\mathcal{NP}$-hardness and provide an approximation algorithm.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The *Longest Common Subsequence* problem, whose first famous dynamic programming solution appeared in 1974 [14], is one of the classical problems in Computer Science. The widely known string version appears in Definition 1.

Definition 1. The *String Longest Common Subsequence* (*LCS*) *Problem*:

Input: Two strings A, B of length n over alphabet Σ.
Output: The length of the longest subsequence common to both strings.

For example, for $A = abcddabef$ and $B = efbadeaab$, $LCS(A, B)$ is 4, where a possible such subsequence is $adab$.

The LCS problem has been very well studied. For a survey, see [5]. The main motivation for the problem is as a measure of string similarity. An immediate example from computational biology is measuring the commonality of two DNA molecules or proteins, which may yield functional similarity between them. The well known dynamic programming solution [7] requires time $O(n^2)$, for two strings of length n each. The problem had also been investigated on more general structures such as trees and matrices [2], run-length encoded strings [4], and more.

Another structure, useful in molecular biology, is the weighted sequence. This is defined as a sequence $S = s_1, \ldots, s_{|S|}$ where a value is associated to every s_i, $i = 1 \ldots |S|$. Comparing two weighted sequences we need a weight function W assigning a value to every possible match between a character from the first and another from the second sequence. The

* Corresponding author at: Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel. Tel.: +972 3 531 8770.
E-mail addresses: amir@cs.biu.ac.il (A. Amir), gotthiz@cs.biu.ac.il (Z. Gotthilf), gonenr1@cs.biu.ac.il (B.R. Shalom).
[1] Partly supported by ISF grant 35/05, and a Binational Israel–Korea grant.

1570-8667/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jda.2010.02.001

LCS variant for these weighted sequences aims at maximizing the *weight* of the common subsequence rather than its *length* as defined below:

Definition 2. The *Heaviest Common Subsequence* (*HCS*) *Problem*:

Input: Two strings $A = a_1 \dots a_n$, $B = b_1 \dots b_m$ of length n over alphabet Σ and a weight function $W : a_i \times b_j \to N$.
Output: A common subsequence of length l $a_{i_1} \dots a_{i_l} = b_{j_1} \dots b_{j_l}$ maximizing the sum $\sum_{k=1}^{l} W(a_{i_k}, b_{j_k})$.

Note that in contrast to sequence alignment, where we have a single weight for the matching of two characters, in the HCS problem the weight of the match depends on the position of the symbols in the input sequences *as well as* on the characters themselves.

Recently another model of weighted sequences was introduced in which, at each position of the sequence, any symbol of the alphabet can occur with a certain probability. To prevent ambiguity, we refer to such sequences as *p-weighted sequences*, though in the literature they are both named weighted sequences.

Definition 3. (See [9].) A p-weighted sequence $A = a_1 \dots a_n$ over alphabet Σ, is a sequence of sets a_i, $1 \leqslant i \leqslant n$. Every a_i is a set of pairs $(s_j, \pi_i(s_j))$, where $s_j \in \Sigma$ and $\pi_i(s_j)$ is the probability of having symbol s_j at location i.

Formally, $a_i = \{(s_j, \pi_i(s_j)) \mid s_j \neq s_l \text{ for } j \neq l, \text{ and } \sum_j \pi_i(s_j) = 1\}$.

The concept of p-weighted sequences was introduced as a tool for motif discovery and local alignment. A weighted sequence is called in the biological literature a "*Position Weight Matrix*" (PWM) [12]. A p-weighted sequence of length m is a $|\Sigma| \times m$ matrix that reports the frequency of each symbol in a finite alphabet Σ for every possible location.

The first usage of PWM sequences was for relative short sequences, for example binding sites, sequences resulting from multiple alignment, etc. Iliopoulos et al. [9] considered building very large Position Weight Matrices that correspond, for example, to complete chromosome sequences that have been obtained using a whole-genome shotgun strategy [13]. By keeping all the information the whole-genome shotgun produces, it is possible to ferret out information that has been previously undetected after being faded during the consensus step. This concept is true for other applications where local similarities are thus encoded. Therefore, the necessity of developing adequate algorithms for p-weighted sequences increases.

It is natural to extend the LCS definition to p-weighted strings as a means of measuring their similarity. However the PWM model deals with probabilities, thus values smaller than 1 are multiplied as a subsequence is extended. The heaviest common p-weighted subsequence will always be of length 1, since every added symbol reduces the total weight. Therefore, we define a **new** but related problem named *Longest Common Weighted Subsequence*, in which the weight is allowed to decrease till a certain bound, and under this restriction the longest common subsequence is sought.

The bound is set according to the certainty level required in the application. Since we consider two p-weighted sequences, we differentiate between their probabilities by denoting π_i^A the probability of occurring at the ith location of sequence A. The formal definition appears below.

Definition 4. The *Longest Common Weighted Subsequence* (*LCWS*) *Problem*:

Input: Two p-weighted strings A, B of length n over alphabet Σ, and a constant α, $0 < \alpha \leqslant 1$.
Output: The maximal l such that there is a common subsequence of length l, $a_{i_1} \dots a_{i_l} = b_{j_1} \dots b_{j_l}$, where $\prod_{y=1}^{l} (\pi_{i_y}^A(a_{i_y}) \cdot \pi_{j_y}^B(b_{j_y})) \geqslant \alpha$.

Though the *LCWS* problem seems natural for the position weighted matrices input, in case the probabilities of the characters of one input sequence are far from being uniformly distributed, the results may be biased and not reflect a real relation between the weighted sequences. In order to prevent this effect and obtain informative results we suggest an additional definition to the *LCWS* problem, Longest Common Weighted Subsequence with two thresholds, referred to as *LCWS*2. In the *LCWS*2 problem, a separate probability bound is set for each of the p-weighted sequences.

Definition 5. The *Longest Common Weighted Subsequence* 2 (*LCWS*2) *Problem*:

Input: Two p-weighted strings A, B of length n over alphabet Σ, and constants α_1, α_2, $0 < \alpha_i \leqslant 1$.
Output: The maximal l such that there is a common subsequence of length l, $a_{i_1} \dots a_{i_l} = b_{j_1} \dots b_{j_l}$, where $\prod_{y=1}^{l} \pi_{i_y}^A(a_{i_y}) \geqslant \alpha_1$ and $\prod_{y=1}^{l} \pi_{j_y}^B(b_{j_y}) \geqslant \alpha_2$.

In this paper, we consider the *log-probability* version of this problem. We define the *Longest Common Integer Weighted Subsequence* 2 (*LCIWS*2) *Problem* in Section 4, and proves that it is $\mathcal{NP}$-hard.

Download English Version:

https://daneshyari.com/en/article/431086

Download Persian Version:

https://daneshyari.com/article/431086

Daneshyari.com