Hybrid techniques based on solving reduced problem instances for a longest common subsequence problem

Christian Blum\textsuperscript{a, *}, Maria J. Blesa\textsuperscript{b}

\textsuperscript{a} Artificial Intelligence Research Institute (IIIA-CSI), Campus UAB, Bellaterra, Spain
\textsuperscript{b} Computer Science Department, Universitat Politècnica de Catalunya – BarcelonaTech, Barcelona, Spain

A R T I C L E  I N F O

Article history:
Received 15 June 2017
Received in revised form 8 September 2017
Accepted 4 October 2017
Available online 18 October 2017

Keywords:
Combinatorial optimization
Longest common subsequences
Integer linear programming
Heuristic
Hybrid algorithm

A B S T R A C T

Finding the longest common subsequence of a given set of input strings is a relevant problem arising in various practical settings. One of these problems is the so-called longest arc-preserving common subsequence problem. This NP-hard combinatorial optimization problem was introduced for the comparison of arc-annotated ribonucleic acid (RNA) sequences. In this work we present an integer linear programming (ILP) formulation of the problem. As in the context of rather small problem instances the application of a general purpose ILP solver is not viable due to the size of the model, we study alternative ways based on model reduction in order to take profit from this ILP model. First, we present a heuristic way for reducing the model, with the subsequent application of an ILP solver. Second, we propose the application of an iterative hybrid algorithm that makes use of an ILP solver for generating high quality solutions at each iteration. Experimental results concerning artificial and real problem instances show that the proposed techniques outperform an available technique from the literature.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

In computer science, a string (or sequence) $x$ of length $l_x$ is a finite sequence of characters from a finite alphabet $\Sigma$. In fact, strings are popular data types for representing and storing information. Words and even complete texts, for example, may be stored in a computer in terms of strings. However strings are not only useful in fields such as information and text processing. They arise, in particular, in the field of computational biology. The reason is that most of the genetic instructions involved in the growth, development, functioning and reproduction of living organisms are stored by means of \textit{deoxyribonucleic acid} (DNA) and \textit{ribonucleic acid} (RNA) molecules, which are either double-stranded (DNA) or single-stranded (RNA) sequences of nucleotides. In short, each nucleotide is composed of a nitrogenous base, a five-carbon sugar (ribose or deoxyribose), and at least one phosphate group. Concerning RNA, each nucleotide has one of four different nitrogenous bases: guanine (G), uracil (U), adenine (A), and cytosine (C). As a consequence, any RNA molecule can be represented as a string of symbols from $\Sigma = \{G, U, A, C\}$, which is called the primary structure of a RNA molecule. The primary structure of a RNA molecule is a simplified representation, because RNA molecules fold in space and different nucleotides bind together, for example, by means of hydrogen bonds. Generally, guanine (G) can only bind with cytosine (C) and uracil (U) can only bind with adenine (A). These hydrogen bonds are present in the so-called secondary structure of an RNA molecule; see Fig. 1a for an example.

For computer science purposes, the hydrogen bonds of the secondary structure of an RNA sequence $x$ can be represented by a so-called arc annotation set $P_x$. In technical terms, $P_x$ is an unordered set of pairs of positions of a string $x$.\textsuperscript{1} Each pair $(i_1, i_2) \in P_x$ represents an arc between positions $i_1$ and $i_2$ and is called an arc annotation. The only convention is that $i_1 < i_2$ must hold for any arc $(i_1, i_2) \in P_x$. Finally, $i_1$ is called the left endpoint of arc $(i_1, i_2)$, and $i_2$ is called the right endpoint. A pair $(x, P_x)$ is called an arc-annotated sequence \textsuperscript{2} (or arc-annotated string). Given this definition, note that the secondary structure of an RNA sequence can conveniently be described by an arc-annotated sequence; see Fig. 1b for an example. In fact, arc-annotated sequences have been widely used for this purpose (see, for example, \cite{3}). In particular, arc-annotated sequences have shown to be useful for the structural comparison of RNA sequences. One of the usual measures when comparing two (or more) sequences is the length of their longest common subsequence.

\textsuperscript{1} As a convention, the positions of a string $x$ range from 1 to $l_x$.

\textsuperscript{2} © 2017 Elsevier B.V. All rights reserved.

https://doi.org/10.1016/j.asoc.2017.10.005
1568-4946/© 2017 Elsevier B.V. All rights reserved.
Fig. 1. (a) Example of the secondary structure of an RNA molecule. (b) The corresponding arc-annotated sequence. The example is reproduced from [1].

Table 1

NP-hard cases of the LAPCS problem. The first two table columns indicate the characterizations of the two input strings, without any order.

<table>
<thead>
<tr>
<th>First characterization</th>
<th>Second characterization</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLIMITED</td>
<td>UNLIMITED</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>UNLIMITED</td>
<td>CROSSING</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>UNLIMITED</td>
<td>NESTED</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>UNLIMITED</td>
<td>CHAIN</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>CROSSING</td>
<td>CROSSING</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>CROSSING</td>
<td>NESTED</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>CROSSING</td>
<td>CHAIN</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>NESTED</td>
<td>NESTED</td>
<td>NP-hard [2,9]</td>
</tr>
<tr>
<td>STEM</td>
<td>STEM</td>
<td>NP-hard [2,9]</td>
</tr>
</tbody>
</table>

First, consider (LCS); see, for example, [4,5]. In this context, given a sequence \(x\) over a finite alphabet \(\Sigma\), sequence \(t\) is called a subsequence of \(x\), if \(t\) can be produced from \(x\) by deleting characters. Given a set of input strings \(\{s_1, \ldots, s_n\}\), the problem of finding the longest common subsequence of all input strings is, in general, NP-hard [6]. The best techniques available nowadays for solving this problem are based on beam search [7] (see [8], for example).

1.1. The LAPCS problem

The longest common subsequence problem in the context of arc-annotated sequences—the longest arc-preserving common subsequence (LAPCS) problem—has first been introduced in [9,2]. Given two input sequences \(x\) and \(y\), the set of possible assignments \(A\) is defined as the set of all \(a_{ij}\)—where \(i \in \{1, \ldots, l_x\}\) and \(j \in \{1, \ldots, l_y\}\)—such that \(x[i]=y[j]\). In other words, \(A\) consists of all \(a_{ij}\) such that at position \(i\) of \(x\) and at position \(j\) of \(y\) there is the same letter. A valid common subsequence of the two input sequences \(x\) and \(y\) can then be represented by a subset \(S \subseteq A\) that fulfills the following conditions:

- **Common subsequence condition**: For any two assignments \(a_{ij}, a_{kl}\), \(a_{ij} \in S\) (where \(a_{ij} \neq a_{kl}\)) it must hold that either \(i < k \land j < l\), or \(i > k \land j > l\).

   In order to translate such a solution into the corresponding common subsequence, the assignments in \(S\) have to be ordered from small to large indices, either according to the first or the second index. Then, the letters corresponding to the assignments must be joined in this order.

   A solution \(S\) that fulfills the common subsequence condition is called arc-preserving if the arcs induced by the solution are preserved:

- **Arc preservation condition**: for any two assignments \(a_{ij}, a_{kl}\) in \(S\) (where \(a_{ij} \neq a_{kl}\) and \(i < k\)) it must hold that \((i,k) \in P_S \iff (j,l) \in P_S\).

Given two arc-annotated input strings \((x, P_x)\) and \((y, P_y)\), the LAPCS problem consists in finding a solution \(S \subseteq A\) that fulfills both the common subsequence and the arc preservation condition and is of maximal cardinality. Note that such a mapping corresponds to the longest arc-preserving common subsequence of \(x\) and \(y\).

In practice, the nature of the arc annotation in the context of RNA sequences generally satisfies some conditions. Given an arc-annotated string \((x, P_x)\), the relative positioning of two arcs \((i_1, i_2)\) and \((j_1, j_2)\) is provided by a number of different characterizations—such as stems, chains, nested sequences, or crossing arcs—of the string. These characterizations are defined as follows:

Fig. 2. Hierarchy of different classifications of arc-annotated sequences.

Fig. 3. Flow diagram of Hyr-Asc (see also Algorithm 1).
دریافت فوری
متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پردادن اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات