Arc-preserving subsequences of arc-annotated sequences

Vladimir Yu. POPOV
Department of Mathematics and Mechanics
Ural State University
620083 Ekaterinburg, RUSSIA
email: Vladimir.Popov@usu.ru

Abstract. Arc-annotated sequences are useful in representing the structural information of RNA and protein sequences. The longest arc-preserving common subsequence problem has been introduced as a framework for studying the similarity of arc-annotated sequences. In this paper, we consider arc-annotated sequences with various arc structures. We consider the longest arc preserving common subsequence problem. In particular, we show that the decision version of the 1-FRAGMENT LAPCS(crossing,chain) and the decision version of the 0-DIAGONAL LAPCS(crossing,chain) are NP-complete for some fixed alphabet Σ such that |Σ| = 2. Also we show that if |Σ| = 1, then the decision version of the 1-FRAGMENT LAPCS(unlimited, plain) and the decision version of the 0-DIAGONAL LAPCS(unlimited, plain) are NP-complete.

1 Introduction

Algorithms on sequences of symbols have been studied for a long time and now form a fundamental part of computer science. One of the very important problems in analysis of sequences is the longest common subsequence (LCS) problem. The computational problem of finding the longest common subsequence of a set of k strings has been studied extensively over the last thirty years (see [5, 19, 21] and references). This problem has many applications.

Computing Classification System 1998: F.1.3
Mathematics Subject Classification 2010: 68Q15
Key words and phrases: longest common subsequence, sequence annotation, NP-complete
When $k = 2$, the longest common subsequence is a measure of the similarity of two strings and is thus useful in molecular biology, pattern recognition, and text compression [26, 27, 34]. The version of LCS in which the number of strings is unrestricted is also useful in text compression [27], and is a special case of the multiple sequence alignment and consensus subsequence discovery problem in molecular biology [11, 12, 32].

The k-unrestricted LCS problem is NP-complete [27]. If the number of sequences is fixed at $k$ with maximum length $n$, their longest common subsequence can be found in $O(n^{k-1})$ time, through an extension of the pairwise algorithm [21]. Suppose $|S_1| = n$ and $|S_2| = m$, the longest common subsequence of $S_1$ and $S_2$ can be found in time $O(nm)$ [8, 18, 35].

Sequence-level investigation has become essential in modern molecular biology. But to consider genetic molecules only as long sequences consisting of the 4 basic constituents is too simple to determine the function and physical structure of the molecules. Additional information about the sequences should be added to the sequences. Early works with these additional information are primary structure based, the sequence comparison is basically done on the primary structure while trying to incorporate secondary structure data [2, 9]. This approach has the weakness that it does not treat a base pair as a whole entity. Recently, an improved model was proposed [13, 14].

Arc-annotated sequences are useful in describing the secondary and tertiary structures of RNA and protein sequences. See [13, 4, 16, 22, 23] for further discussion and references. Structure comparison for RNA and for protein sequences has become a central computational problem bearing many challenging computer science questions. In this context, the longest arc preserving common subsequence problem (LAPCS) recently has received considerable attention [13, 14, 22, 23, 25]. It is a sound and meaningful mathematical formalization of comparing the secondary structures of molecular sequences. Studies for this problem have been undertaken in [5, 16, 1, 3, 6, 7, 10, 15, 20, 28, 29, 30, 33].

2 Preliminaries and problem definitions

Given two sequences $S$ and $T$ over some fixed alphabet $\Sigma$, the sequence $T$ is a subsequence of $S$ if $T$ can be obtained from $S$ by deleting some letters from $S$. Notice that the order of the remaining letters of $S$ bases must be preserved. The length of a sequence $S$ is the number of letters in it and is denoted as $|S|$. For simplicity, we use $S[i]$ to denote the $i$th letter in sequence $S$, and $S[i, j]$ to
Arc-preserving subsequences

denote the substring of $S$ consisting of the $i$th letter through the $j$th letter.

Given two sequences $S_1$ and $S_2$ (over some fixed alphabet $\Sigma$), the classic longest common subsequence problem asks for a longest sequence $T$ that is a subsequence of both $S_1$ and $S_2$.

An arc-annotated sequence of length $n$ on a finite alphabet $\Sigma$ is a couple $A = (S, P)$ where $S$ is a sequence of length $n$ on $\Sigma$ and $P$ is a set of pairs $(i_1, i_2)$, with $1 \leq i_1 < i_2 \leq n$. In this paper we will then call an element of $S$ a base. A pair $(i_1, i_2) \in P$ represents an arc linking bases $S[i_1]$ and $S[i_2]$ of $S$. The bases $S[i_1]$ and $S[i_2]$ are said to belong to the arc $(i_1, i_2)$ and are the only bases that belong to this arc.

Given two annotated sequences $S_1$ and $S_2$ with arc sets $P_1$ and $P_2$ respectively, a common subsequence $T$ of $S_1$ and $S_2$ induces a bijective mapping from a subset of $\{1, \ldots, |S_1|\}$ to subset of $\{1, \ldots, |S_2|\}$. The common subsequence $T$ is arc-preserving if the arcs induced by the mapping are preserved, i.e., for any $(i_1, j_1)$ and $(i_2, j_2)$ in the mapping,

$$(i_1, i_2) \in P_1 \iff (j_1, j_2) \in P_2.$$ 

The LAPCS problem is to find a longest common subsequence of $S_1$ and $S_2$ that is arc-preserving (with respect to the given arc sets $P_1$ and $P_2$) [13].

**LAPCS:**

**INSTANCE:** An alphabet $\Sigma$, annotated sequences $S_1$ and $S_2$, $S_1, S_2 \in \Sigma^*$, with arc sets $P_1$ and $P_2$ respectively.

**QUESTION:** Find a longest common subsequence of $S_1$ and $S_2$ that is arc-preserving.

The arc structure can be restricted. We consider the following four natural restrictions on an arc set $P$ which are first discussed in [13]:

1. no sharing of endpoints:
   $$\forall (i_1, i_2), (i_3, i_4) \in P, i_1 \neq i_4, i_2 \neq i_3, \text{ and } i_1 = i_3 \iff i_2 = i_4.$$

2. no crossing:
   $$\forall (i_1, i_2), (i_3, i_4) \in P, i_1 \in [i_3, i_4] \iff i_2 \in [i_3, i_4].$$

3. no nesting:
   $$\forall (i_1, i_2), (i_3, i_4) \in P, i_1 \leq i_3 \iff i_2 \leq i_3.$$

4. no arcs:
   $$P = \emptyset.$$

These restrictions are used progressively and inclusively to produce five distinct levels of permitted arc structures for LAPCS:

- **UNLIMITED** — no restrictions;
- **CROSSING** — restriction 1;
- **NESTED** — restrictions 1 and 2;
The problem LAPCS is varied by these different levels of restrictions as LAPCS(x, y) which is problem LAPCS with S1 having restriction level x and S2 having restriction level y. Without loss of generality, we always assume that x is the same level or higher than y.

We give the definitions of two special cases of the LAPCS problem, which were first studied in [25]. The special cases are motivated from biological applications [17, 24].

**The c-fragment LAPCS problem (c ≥ 1):**

**Instance:** An alphabet Σ, annotated sequences S1 and S2, S1, S2 ∈ Σ∗, with arc sets P1 and P2 respectively, where S1 and S2 are divided into fragments of lengths exactly c (the last fragment can have a length less than c).

**Question:** Find a longest common subsequence of S1 and S2 that is arc-preserving. The allowed matches are those between fragments at the same location.

The c-diagonal LAPCS problem, (c ≥ 0), is an extension of the c-fragment LAPCS problem, where base S2[i] is allowed only to match bases in the range S1[i−c, i+c].

The c-diagonal LAPCS and c-fragment LAPCS problems are relevant in the comparison of conserved RNA sequences where we already have a rough idea about the correspondence between bases in the two sequences.

## 3 Previous results

It is shown in [25] that the 1-fragment LAPCS(crossing, crossing) and 0-diagonal LAPCS(crossing, crossing) are solvable in time O(n). An overview on known NP-completeness results for c-diagonal LAPCS and c-fragment LAPCS is given in Figure 1.

<table>
<thead>
<tr>
<th></th>
<th>unlimited</th>
<th>crossing</th>
<th>nested</th>
<th>chain</th>
<th>plain</th>
</tr>
</thead>
<tbody>
<tr>
<td>nested</td>
<td>—</td>
<td>—</td>
<td>NP-h [25]</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Figure 1: NP-completeness results for c-diagonal LAPCS (with c ≥ 1) and c-fragment LAPCS (with c ≥ 2)
4 The c-FRAGMENT LAPCS(UNLIMITED,PLAIN) and the c-DIAGONAL LAPCS(UNLIMITED,PLAIN) problem

Let us consider the decision version of the c-FRAGMENT LAPCS problem.

**Instance:** An alphabet $\Sigma$, a positive integer $k$, annotated sequences $S_1$ and $S_2$, $S_1, S_2 \in \Sigma^*$, with arc sets $P_1$ and $P_2$ respectively, where $S_1$ and $S_2$ are divided into fragments of lengths exactly $c$ (the last fragment can have a length less than $c$).

**Question:** Is there a common subsequence $T$ of $S_1$ and $S_2$ that is arc-preserving, $|T| \geq k$? (The allowed matches are those between fragments at the same location).

Similarly, we can define the decision version of the c-DIAGONAL LAPCS problem.

**Theorem 1** If $|\Sigma| = 1$, then c-FRAGMENT LAPCS(UNLIMITED, PLAIN) and c-DIAGONAL LAPCS(UNLIMITED, PLAIN) are NP-complete.

**Proof.** It is easy to see that c-FRAGMENT LAPCS(UNLIMITED, PLAIN) = c-DIAGONAL LAPCS(UNLIMITED, PLAIN).

Let $G = (V, E)$ be an undirected graph, and let $I \subseteq V$. We say that the set $I$ is independent if whenever $i, j \in I$ then there is no edge between $i$ and $j$. We make use of the following problem:

**Independent Set (IS): Instance:** A graph $G = (V, E)$, a positive integer $k$.

**Question:** Is there an independent set $I$, $I \subseteq V$, with $|I| \geq k$?

IS is NP-complete (see [31]).

Let us suppose that $\Sigma = \{a\}$. We will show that IS can be polynomially reduced to problem c-FRAGMENT LAPCS(UNLIMITED, PLAIN).

Let $(G = (V, E), V = \{1, 2, \ldots, n\}, k)$ be an instance of IS. Now we transform an instance of the IS problem to an instance of the c-FRAGMENT LAPCS(UNLIMITED, PLAIN) problem as follows.

- $S_1 = S_2 = a^n$.
- $P_1 = E, P_2 = \emptyset$.
- $(S_1, P_1), (S_2, P_2), k$.

First suppose that the graph $G$ has an independent set $I$ of size $k$. By definition of independent set, $(i, j) \notin E$ for each $i, j \in I$. For a given subset $I$, let

$$M = \{(i, i) : i \in I\}.$$  

Since $I$ is an independent set, if $(i, j) \in E = P_1$ then either $(i, i) \notin M$ or
\((j, j) \notin M\). This preserves arcs since \(P_2\) is empty. Clearly, \(S_1[i] = S_2[i]\) for each \(i \in I\), and the allowed matches are those between fragments at the same location. Therefore, there is a common subsequence \(T\) of \(S_1\) and \(S_2\) that is arc-preserving, \(|T| = k\), and the allowed matches are those between fragments at the same location.

Now suppose that there is a common subsequence \(T\) of \(S_1\) and \(S_2\) that is arc-preserving, \(|T| = k\), and the allowed matches are those between fragments at the same location. In this case there is a valid mapping \(M\), with \(|M| = k\). Since \(c = 1\), it is easy to see that if \((i, j) \in M\) then \(i = j\). Let

\[ I = \{i : (i, i) \in M\}. \]

Clearly,

\[ |I| = |M| = k. \]

Let \(i_1\) and \(i_2\) be any two distinct members of \(I\). Then let \((i_1, j_1), (i_2, j_2) \in M\). Since

\[ i_1 = j_1, i_2 = j_2, i_1 \neq i_2, \]

it is easy to see that \(j_1 \neq j_2\). Since \(P_2\) is empty, \((j_1, j_2) \notin P_2\), so \((i_1, i_2) \notin P_1\). Since \(P_1 = E\), the set \(I\) of vertices is a size \(k\) independent set of \(G\). \(\square\)

5 The c-FRAGMENT LAPCS(crossing,chain) and the c-DIAGONAL LAPCS(crossing,chain) problem

**Theorem 2** If \(|\Sigma| = 2\), then 1-FRAGMENT LAPCS(crossing,chain) and 0-DIAGONAL LAPCS(crossing,chain) are \(NP\)-complete.

**Proof.** It is easy to see that 1-FRAGMENT LAPCS(crossing,chain) = 0-DIAGONAL LAPCS(crossing,chain).

Let us suppose that \(\Sigma = \{a, b\}\). We will show that IS can be polynomially reduced to problem 1-FRAGMENT LAPCS(crossing,chain).

Let \(\langle G = (V, E), V = \{1, 2, \ldots, n\}, k\rangle\) be an instance of IS. Note that IS remains \(NP\)-complete when restricted to connected graphs with no loops and multiple edges. Let \(G = (V, E)\) be such a graph. Now we transform an instance of the IS problem to an instance of the 1-FRAGMENT LAPCS(crossing,chain) problem as follows.
There are two cases to consider.

Case I. $k > n$

- $S_1 = S_2 = a$
- $P_1 = P_2 = \emptyset$
- $((S_1, P_1), (S_2, P_2), k)$

Clearly, if $I$ is an independent set, then $I \subseteq V$ and $|I| \leq |V| = n$. Therefore, there is no an independent set $I$, with $|I| \geq k$.

Since $k > n$ and $n \in \{1, 2, \ldots\}$, it is easy to see that $k > 1$. Since $S_1 = S_2 = a$ and $P_1 = P_2 = \emptyset$, $T = a$ is the longest arc-preserving common subsequence. Therefore, there is no an arc-preserving common subsequence $T$ such that $|T| \geq k$.

Case II. $k \leq n$

- $S_1 = S_2 = (a^nb^n)$
- Let $\alpha < \beta$. Then

$$(\alpha, \beta) \in P_1 \iff \exists i \in \{1, 2, \ldots, n\} \exists j \in \{1, 2, \ldots, n\}$$

$$(i, j) \in E \land \alpha = (i - 1)(n + 2) + j + 1 \land$$

$$\land \beta = (j - 1)(n + 2) + i + 1) \lor$$

$$\lor \exists i \in \{1, 2, \ldots, n\} (\alpha = (i - 1)(n + 2) + 1 \land \beta = i(n + 2)),$$

$$(\alpha, \beta) \in P_2 \iff \exists i \in \{1, 2, \ldots, n\}$$

$$(\alpha = (i - 1)(n + 2) + 1 \land \beta = i(n + 2)).$$

- $((S_1, P_1), (S_2, P_2), k(n + 2))$

First suppose that $G$ has an independent set $I$ of size $k$. By definition of independent set, $(i, j) \notin E$ for each $i, j \in I$. For a given subset $I$, let

$$M = \{(j, j) : j = (n + 2)(i - 1) + 1, i \in I,$$

$$l \in \{1, 2, \ldots, n + 2\}\}.$$

Let $(j, j) \in M$, and there exist $i$ such that $j = (n + 2)(i - 1) + 1$. By definition of $M$,

$$((n + 2)(i - 1) + 1, (n + 2)(i - 1) + 1) \in M \iff$$

$$\iff ((n + 2)i, (n + 2)i) \in M.$$
By definition of $P_{l}$, $((n + 2)(i - 1) + 1, (n + 2)i) \in P_{l}$ where $l = 1, 2$. Let $(j, j) \in M$, and there exist $i$ such that $j = (n + 2)i$. By definition of $M$,

$$((n + 2)i, (n + 2)i) \in M \Leftrightarrow ((n + 2)(i - 1) + 1, (n + 2)(i - 1) + 1) \in M.$$ 

By definition of $P_{1}$,

$$((n + 2)(i - 1) + 1, (n + 2)i) \in P_{1}$$

where $l = 1, 2$. Let $(j, j) \in M$, and

$$j = (n + 2)(i - 1) + l$$

where $1 < l < n + 2$. By definition of $M$, $i \in I$. Since $I$ is an independent set, if $(i, l - 1) \in E$ then $l - 1 \not\in I$. Since

$$1 < l < n + 2,$$

by definition of $P_{1}$, either

$$((n + 2)(i - 1) + l, (n + 2)(l - 2) + i + 1) \in P_{1}$$

or

$$((n + 2)(i - 1) + l, t) \not\in P_{1}$$

for each $t$. Since

$$1 < l < n + 2,$$

by definition of $P_{2}$,

$$((n + 2)(i - 1) + l, t) \not\in P_{2}$$

for each $t$. If

$$((n + 2)(i - 1) + l, (n + 2)(l - 2) + i + 1) \in P_{1},$$

then in view of $l - 1 \not\in I$,

$$((n + 2)(l - 2) + i + 1, (n + 2)(l - 2) + i + 1) \not\in M.$$ 

This preserves arcs. Since $|I| = k$, it is easy to see that

$$|M| = k(n + 2).$$
Clearly, \( S_1[i] = S_2[i] \) for each \( i \in I \), and the allowed matches are those between fragments at the same location. Therefore, there is a common subsequence \( T \) of \( S_1 \) and \( S_2 \) that is arc-preserving, \( |T| = k(n + 2) \), and the allowed matches are those between fragments at the same location.

Now suppose that there is a common subsequence \( T \) of \( S_1 \) and \( S_2 \) that is arc-preserving, \( |T| = k \), and the allowed matches are those between fragments at the same location. In this case, there is a valid mapping \( M \), with \( |M| = k \). Since \( c = 1 \), it is easy to see that if \( (i, j) \in M \) then \( i = j \). Let \( I = \{ i : (i, i) \in M \} \). Clearly, \( |I| = |M| = k \). Let \( i_1 \) and \( i_2 \) be any two distinct members of \( I \). Then let \( (i_1, j_1), (i_2, j_2) \in M \). Since \( i_1 = j_1, i_2 = j_2, i_1 \neq i_2 \), it is easy to see that \( j_1 \neq j_2 \). Since \( P_2 \) is empty, \( (j_1, j_2) \notin P_2 \), so \( (i_1, i_2) \notin P_1 \). Since \( P_1 = E \), the set \( I \) of vertices is a size k independent set of \( G \).

6 Conclusions

In this paper, we considered two special cases of the LAPCS problem, which were first studied in [25]. We have shown that the decision version of the 1-fragment LAPCS (crossing, chain) and the decision version of the 0-diagonal LAPCS (crossing, chain) are \( \text{NP} \)-complete for some fixed alphabet \( \Sigma \) such that \( |\Sigma| = 2 \). Also we have shown that if \( |\Sigma| = 1 \), then the decision version of the 1-fragment LAPCS (unlimited, plain) and the decision version of the 0-diagonal LAPCS (unlimited, plain) are \( \text{NP} \)-complete. This results answers some open questions in [16] (see Table 4.2. in [16]).

Acknowledgements

The work was partially supported by Grant of President of the Russian Federation MD-1687.2008.9 and Analytical Departmental Program “Developing the scientific potential of high school” 2.1.1/1775.

References


*Received: November 17, 2010* Revised: March 11, 2011