



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 $|\Sigma| = 2$. Also we show 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 **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 Σ , 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

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 Σ), 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 Σ is a couple $A = (S, P)$ where S is a sequence of length n on Σ 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 $\{1, \dots, |S_1|\}$ to subset of $\{1, \dots, |S_2|\}$. 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,

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

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 Σ , 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:

$$\forall (i_1, i_2), (i_3, i_4) \in P, i_1 \neq i_4, i_2 \neq i_3, \text{ and } i_1 = i_3 \Leftrightarrow i_2 = i_4.$$

2. no crossing:

$$\forall (i_1, i_2), (i_3, i_4) \in P, i_1 \in [i_3, i_4] \Leftrightarrow i_2 \in [i_3, i_4].$$

3. no nesting:

$$\forall (i_1, i_2), (i_3, i_4) \in P, i_1 \leq i_3 \Leftrightarrow i_2 \leq i_4.$$

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;

- CHAIN — restrictions 1, 2 and 3;
- PLAIN — restriction 4.

The problem LAPCS is varied by these different levels of restrictions as $\text{LAPCS}(x, y)$ which is problem LAPCS with S_1 having restriction level x and S_2 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 ($c \geq 1$):

INSTANCE: An alphabet Σ , 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: Find a longest common subsequence of S_1 and S_2 that is arc-preserving. The allowed matches are those between fragments at the same location.

The c -DIAGONAL LAPCS problem, ($c \geq 0$), is an extension of the c -FRAGMENT LAPCS problem, where base $S_2[i]$ is allowed only to match bases in the range $S_1[i - c, i + c]$.

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.

	unlimited	crossing	nested	chain	plain
unlimited	NP-h [25]	NP-h [25]	NP-h [25]	?	?
crossing	—	NP-h [25]	NP-h [25]	?	?
nested	—	—	NP-h [25]	?	?

Figure 1: **NP**-completeness results for c -DIAGONAL LAPCS (with $c \geq 1$) and c -FRAGMENT LAPCS (with $c \geq 2$)

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 Σ , 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 1-FRAGMENT LAPCS(UNLIMITED, PLAIN) and 0-DIAGONAL LAPCS(UNLIMITED, PLAIN) are **NP**-complete.*

Proof. It is easy to see that 1-FRAGMENT LAPCS(UNLIMITED, PLAIN) = 0-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 1-FRAGMENT LAPCS(UNLIMITED, PLAIN).

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

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

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 $\langle G = (V, E), V = \{1, 2, \dots, n\}, k \rangle$ 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. $k > n$

- $S_1 = S_2 = \mathbf{a}$
- $P_1 = P_2 = \emptyset$
- $\langle (S_1, P_1), (S_2, P_2), k \rangle$

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

Case II. $k \leq n$

- $S_1 = S_2 = (\mathbf{ba}^n\mathbf{b})^n$
- Let $\alpha < \beta$. Then

$$(\alpha, \beta) \in P_1 \Leftrightarrow [\exists i \in \{1, 2, \dots, n\} \exists j \in \{1, 2, \dots, n\}$$

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

$$\wedge \beta = (j - 1)(n + 2) + i + 1)] \vee$$

$$\vee [\exists i \in \{1, 2, \dots, n\} (\alpha = (i - 1)(n + 2) + 1 \wedge \beta = i(n + 2))],$$

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

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

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

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

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

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

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

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

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

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 ,

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

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

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

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

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 $l-1 \notin I$. 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

$$((n+2)(i-1)+l, t) \notin P_1$$

for each t . Since

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

by definition of P_2 ,

$$((n+2)(i-1)+l, t) \notin P_2$$

for each t . If

$$((n+2)(i-1)+l, (n+2)(l-2)+i+1) \in P_1,$$

then in view of $l-1 \notin I$,

$$((n+2)(l-2)+i+1, (n+2)(l-2)+i+1) \notin M.$$

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 . \square

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 **NP**-complete for some fixed alphabet Σ 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 **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

- [1] J. Alber, J. Gramm, J. Guo, R. Niedermeier, Computing of two sequences with nested arc notations, *Theoret. Comput. Sci.* **312**, 2-3 (2004) 337–358.
 \Rightarrow 36

-
- [2] V. Bafna, S. Muthukrishnan, R. Ravi, Comparing similarity between RNA strings, *Proc. 6th Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Comput. Sci.* **937** (1995) 1–16. [⇒36](#)
 - [3] G. Blin, H. Touzet, How to compare arc-annotated sequences: The alignment hierarchy, *Proc. 13th International Symposium on String Processing and Information Retrieval (SPIRE), Lecture Notes in Comput. Sci.* **4209** (2006) 291–303. [⇒36](#)
 - [4] G. Blin, M. Crochemore, S. Vialette, Algorithmic aspects of arc-annotated sequences, in: *Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications* (ed. M. Elloumi, A. Y. Zomaya), John Wiley & Sons, Inc., Hoboken, NJ, 2011, pp. 171–183. [⇒36](#)
 - [5] H. L. Bodlaender, R. G. Downey, M. R. Fellows, H. T. Wareham, The parameterized complexity of sequence alignment and consensus, *Theoret. Comput. Sci.* **147**, 1-2 (1995) 31–54. [⇒35](#), [36](#)
 - [6] H. L. Bodlaender, R. G. Downey, M. R. Fellows, M. T. Hallett, H. T. Wareham, Parameterized complexity analysis in computational biology, *Computer Applications in the Biosciences* **11**, 1 (1995) 49–57. [⇒36](#)
 - [7] J. Chen, X. Huang, I. A. Kanj, G. Xia, W-hardness under linear FPT-reductions: structural properties and further applications, *Proc. of COCOON*, Kunming, China, 2005, pp. 975–984. [⇒36](#)
 - [8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, Third edition, The MIT Press, Cambridge, Massachusetts, 2009. [⇒36](#)
 - [9] F. Corpet, B. Michot, Rnalign program: alignment of RNA sequences using both primary and secondary structures, *Computer Applications in the Biosciences* **10**, 4 (1994) 389–399. [⇒36](#)
 - [10] P. Damaschke, A remark on the subsequence problem for arc-annotated sequences with pairwise nested arcs, *Inform. Process. Lett.* **100**, 2 (2006) 64–68. [⇒36](#)
 - [11] W. H. E. Day, F. R. McMorris, Discovering consensus molecular sequences, in: *Information and Classification – Concepts, Methods, and Applications* (ed. O. Opitz, B. Lausen, R. Klar), Springer-Verlag, Berlin, 1993, pp. 393–402. [⇒36](#)

-
- [12] W. H. E. Day, F. R. McMorris, The computation of consensus patterns in DNA sequences, *Math. Comput. Modelling* **17**, 10 (1993) 49–52. [⇒36](#)
 - [13] P. A. Evans, *Algorithms and Complexity for Annotated Sequence Analysis*, PhD Thesis, University of Victoria, Victoria, 1999. [⇒36, 37](#)
 - [14] P. A. Evans, Finding common subsequences with arcs and pseudo-knots, *Proc. 10th Annual Symposium on Combinatorial Pattern Matching (CPM'99)*, *Lecture Notes in Comput. Sci.* **1645** (1999) 270–280. [⇒36](#)
 - [15] J. Gramm, J. Guo, R. Niedermeier, Pattern matching for arc-annotated sequences, *ACM Trans. Algorithms* **2**, 1 (2006) 44–65. [⇒36](#)
 - [16] J. Guo, *Exact algorithms for the longest common subsequence problem for arc-annotated sequences*, Master Thesis, Eberhard-Karls-Universität, Tübingen, 2002. [⇒36, 43](#)
 - [17] D. Gusfield, *Algorithm on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, Cambridge, 1997. [⇒38](#)
 - [18] D. S. Hirschberg, *The Longest Common Subsequence Problem*, PhD Thesis, Princeton University, Princeton, 1975. [⇒36](#)
 - [19] D. S. Hirschberg, Recent results on the complexity of common subsequence problems, in: *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison* (ed. D. Sankoff, J. B. Kruskal), Addison-Wesley Publishing Company, Reading/Menlo Park, NY, 1983, pp. 325–330. [⇒35](#)
 - [20] C. S. Iliopouéos, M. S. Rahman, Algorithms for computing variants of the longest common subsequence problem, *Theoret. Comput. Sci.* **395**, 2-3 (2008) 255–267. [⇒36](#)
 - [21] R. W. Irving, C. B. Fraser, Two algorithms for the longest common subsequence of three (or more) strings, *Proc. Third Annual Symposium on Combinatorial Pattern Matching*, *Lecture Notes in Comput. Sci.* **644** (1992) 214–229. [⇒35, 36](#)
 - [22] T. Jiang, G.-H. Lin, B. Ma, K. Zhang, The longest common subsequence problem for arc-annotated sequences, *Proc. 11th Annual Symposium on Combinatorial Pattern Matching (CPM 2000)*, *Lecture Notes in Comput. Sci.* **1848** (2000) 154–165. [⇒36](#)

-
- [23] T. Jiang, G.-H. Lin, B. Ma, K. Zhang, The longest common subsequence problem for arc-annotated sequences, *J. Discrete Algorithms* **2**, 2 (2004) 257–270. [⇒ 36](#)
 - [24] M. Li, B. Ma, L. Wang, Near optimal multiple alignment within a band in polynomial time, *Proc. Thirty-second Annual ACM Symposium on Theory of Computing (STOC'00)*, Portland, OR, 2000, pp. 425–434. [⇒ 38](#)
 - [25] G. H. Lin, Z. Z. Chen, T. Jiang, J. J. Wen, The longest common subsequence problem for sequences with nested arc annotations, *Proceedings of the 28th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci.* **2076** (2001) 444–455. [⇒ 36](#), [38](#), [43](#)
 - [26] S. Y. Lu, K. S. Fu, A sentence-to-sentence clustering procedure for pattern analysis, *IEEE Transactions on Systems, Man, and Cybernetics* **8**, 5 (1978) 381–389. [⇒ 36](#)
 - [27] D. Maier, The complexity of some problems on subsequences and supersequences, *J. ACM* **25**, 2 (1978) 322–336. [⇒ 36](#)
 - [28] D. Marx, I. Schlotter, Parameterized complexity of the arc-preserving subsequence problem, *Proc. 36th International Workshop on Graph Theoretic Concepts in Computer Science (WG 2010), Lecture Notes in Comput. Sci.* **6410** (2010) 244–255. [⇒ 36](#)
 - [29] A. Ouangraoua, C. Chauve, V. Guignon, S. Hamel, New algorithms for aligning nested arc-annotated sequences, Laboratoire Bordelais de Recherche en Informatique, Research Report RR-1443-08, Université Bordeaux, 2008. [⇒ 36](#)
 - [30] A. Ouangraoua, V. Guignon, S. Hamel, C. Chauve, A new algorithm for aligning nested arc-annotated sequences under arbitrary weight schemes, *Theoret. Comput. Sci.* **412**, 8–10 (2011) 753–764. [⇒ 36](#)
 - [31] C. H. Papadimitriou, *Computational complexity*, Addison-Wesley Publishing Company, Reading/Menlo Park, NY, 1994. [⇒ 39](#)
 - [32] P. A. Pevzner, Multiple alignment, communication cost, and graph matching, *SIAM J. Appl. Math.* **52**, 6 (1992) 1763–1779. [⇒ 36](#)

-
- [33] K. Pietrzak, On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems, *J. Comput. System Sci.* **67**, 4 (2003) 757–771. [⇒ 36](#)
 - [34] D. Sankoff, Matching comparisons under deletion/insertion constraints, *Proc. Natl. Acad. Sci. USA* **69**, 1 (1972) 4–6. [⇒ 36](#)
 - [35] R. A. Wagner, M. J. Fischer, The string-to-string correction problem, *J. ACM* **21**, 1 (1974) 168–173. [⇒ 36](#)

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