Journal of Discrete Algorithms 26 (2014) 98–105

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

An efficient dynamic programming algorithm for the generalized LCS problem with multiple substring exclusive constraints

Yingjie Wu [c], Lei Wang [a], Daxin Zhu [b], Xiaodong Wang [b,c,*]

[a] Microsoft AdCenter, Bellevue, WA 98004, USA
[b] Faculty of Mathematics & Computer Science, Quanzhou Normal University, China
[c] Faculty of Mathematics & Computer Science, Fuzhou University, China

ARTICLE INFO

Article history:
Received 24 March 2013
Received in revised form 4 November 2013
Accepted 24 January 2014
Available online 18 February 2014

Keywords:
Dynamic programming
Algorithm
Generalized LCS problem
Multiple substring exclusion

ABSTRACT

In this paper, we consider a generalized longest common subsequence problem with multiple substring exclusive constraints. For the two input sequences X and Y of lengths n and m, and a set of d constraints $P = \{P_1, \ldots, P_d\}$ of total length r, the problem is to find a common subsequence Z of X and Y excluding each of constraint string in P as a substring and the length of Z is maximized. The problem was declared to be NP-hard [7], but we finally found that this is not true. A new dynamic programming solution for this problem is presented in this paper. The correctness of the new algorithm is proved. The time complexity of our algorithm is $O(nmr)$.

1. Introduction

In this paper, we consider a generalized longest common subsequence problem with multiple substring exclusive constraints. The longest common subsequence (LCS) problem is a well-known measurement for computing the similarity of two strings, and it is crucial in various applications. In this problem, we are interested in a longest sequence which is a subsequence of both sequences. The problem is well studied and is used in many applications, like DNA and protein analysis, text information retrieval, file comparing, music information retrieval, or spelling correction.

The most referred algorithm, proposed by Wagner and Fischer [29], solves the LCS problem by using a dynamic programming algorithm in quadratic time. Other advanced algorithms were proposed in the past decades [3,2,4,16,17,19,21].

If the number of input sequences is not fixed, the problem to find the LCS of multiple sequences has been proved to be NP-hard [23]. Some approximate and heuristic algorithms were proposed for these problems [6,25].

There are also a lot of generalizations of this similarity measure. One of the recent variants of the LCS problem, the constrained longest common subsequence (CLCS) which was first addressed by Tsai [27], has received much attention. It generalizes the LCS measure by introduction of a third sequence, which allows to extort that the obtained CLCS has some special properties [26]. For two given input sequences X and Y of lengths m and n, respectively, and a constrained sequence P of length r, the CLCS problem is to find the common subsequences Z of X and Y such that P is a subsequence of Z and the length of Z is the maximum.

* Corresponding author at: Faculty of Mathematics & Computer Science, Quanzhou Normal University, China.
E-mail address: wangxiaodong@qztc.edu.cn (X. Wang).

http://dx.doi.org/10.1016/j.jda.2014.01.005
1570-8667/

Table 1
The GC-LCS problems.

Problem	Input	Output
SEQ-IC-LCS	X, Y, and P	The longest common subsequence of X and Y including P as a subsequence
STR-IC-LCS	X, Y, and P	The longest common subsequence of X and Y including P as a substring
SEQ-EC-LCS	X, Y, and P	The longest common subsequence of X and Y excluding P as a subsequence
STR-EC-LCS	X, Y, and P	The longest common subsequence of X and Y excluding P as a substring

Table 2
The Multiple-GC-LCS problems.

Problem	Input	Output
M-SEQ-IC-LCS	X, Y, and a set of constraints $P = \{P_1, \ldots, P_d\}$	The longest common subsequence of X and Y including each of constraint $P_i \in P$ as a subsequence
M-STR-IC-LCS	X, Y, and a set of constraints $P = \{P_1, \ldots, P_d\}$	The longest common subsequence of X and Y including each of constraint $P_i \in P$ as a substring
M-SEQ-EC-LCS	X, Y, and a set of constraints $P = \{P_1, \ldots, P_d\}$	The longest common subsequence of X and Y excluding each of constraint $P_i \in P$ as a subsequence
M-STR-EC-LCS	X, Y, and a set of constraints $P = \{P_1, \ldots, P_d\}$	The longest common subsequence of X and Y excluding each of constraint $P_i \in P$ as a substring

The most referred algorithms were proposed independently [5,8], which solve the CLCS problem in $O(mnr)$ time and space by using dynamic programming algorithms. Some improved algorithms have also been proposed [11,18]. The LCS and CLCS problems on the indeterminate strings were discussed in [20]. Moreover, the problem was extended to the one with weighted constraints, a more generalized problem [24].

Recently, a new variant of the CLCS problem, the restricted LCS problem, was proposed [13], which excludes the given constraint as a subsequence of the answer. The restricted LCS problem becomes NP-hard when the number of constraints is not fixed.

Some more generalized forms of the CLCS problem, the generalized constrained longest common subsequence (GC-LCS) problems, were addressed independently by Chen and Chao [7]. For the two input sequences X and Y of lengths n and m, respectively, and a constraint string P of length r, the GC-LCS problem is a set of four problems which are to find the LCS of X and Y including/excluding P as a subsequence/substring, respectively. The four generalized constrained LCS [7] can be summarized in Table 1.

For the four problems in Table 1, $O(mnr)$ time algorithms were proposed [7]. However, their algorithm for STR-EC-LCS is not correct. In a recent paper, a correct $O(mnr)$ time dynamic programming algorithm was proposed [30]. For all four variants in Table 1, $O(r(m+n) + (m+n)\log(m+n))$ time algorithms were proposed by using the finite automata [12]. Recently, a quadratic algorithm to the STR-IC-LCS problem was proposed [10], and the time complexity of [12] was pointed out not correct.

The four GC-LCS problems can be generalized further to the cases of multiple constraints. In these generalized cases, the single constrained pattern P will be generalized to a set of d constraints $P = \{P_1, \ldots, P_d\}$ of total length r, as shown in Table 2.

The problem M-SEQ-IC-LCS has been proved to be NP-hard in [14]. The problem M-SEQ-EC-LCS has also been proved to be NP-hard in [13,28]. In addition, the problems M-STR-IC-LCS and M-STR-EC-LCS were also declared to be NP-hard in [7], but without a proof. The exponential-time algorithms for solving these two problems were also presented in [7].

We will discuss the problem M-STR-EC-LCS in this paper. The failure functions in the Knuth–Morris–Pratt algorithm [22] for solving the string matching problem have been proved very helpful for solving the STR-EC-LCS problem. It has been found by Aho and Corasick [1] that the failure functions can be generalized to the case of keyword tree to speedup the exact string matching of multiple patterns. This idea can be very helpful in our dynamic programming algorithm. This is the main idea of our new algorithm. A polynomial time algorithm is presented for the M-STR-EC-LCS problem based on this observation and disproves that M-STR-EC-LCS problem is NP-hard.

The organization of the paper is as follows.

In the following 4 sections we describe our presented dynamic programming algorithm for the M-STR-EC-LCS problem.

In Section 2 the preliminary knowledge for presenting our algorithm for the M-STR-EC-LCS problem is discussed. In Section 3 we give a new dynamic programming solution for the M-STR-EC-LCS problem with time complexity $O(nmr)$, where n and m are the lengths of the two given input strings, and r is the total length of d constraint strings. In Section 4 we discuss the issues to implement the algorithm efficiently. Some concluding remarks are given in Section 5.