Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Information Processing Letters 90 (2004) 175–179

Information Processing Letters

www.elsevier.com/locate/ipl

A simple algorithm for the constrained sequence problems

Francis Y.L. Chin [a,∗,1], Alfredo De Santis [b], Anna Lisa Ferrara [b], N.L. Ho [a], S.K. Kim [c]

[a] Department of Computer Science and Information Systems, The University of Hong Kong, Pokfulam Road, Hong Kong
[b] Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84081 Baronissi (SA), Italy
[c] Department of Computer Science and Engineering, Chung-Ang University, 221 Huksuk-dong, Dongjak-ku, Seoul 156-756, Republic of Korea

Received 1 November 2003; received in revised form 29 January 2004

Communicated by F. Dehne

Abstract

In this paper we address the constrained longest common subsequence problem. Given two sequences X, Y and a constrained sequence P, a sequence Z is a constrained longest common subsequence for X and Y with respect to P if Z is the longest subsequence of X and Y such that P is a subsequence of Z.

Recently, Tsai [Inform. Process. Lett. 88 (2003) 173–176] proposed an $O(n^2 \cdot m^2 \cdot r)$ time algorithm to solve this problem using dynamic programming technique, where n, m and r are the lengths of X, Y and P, respectively.

In this paper, we present a simple algorithm to solve the constrained longest common subsequence problem in $O(n \cdot m \cdot r)$ time and show that the constrained longest common subsequence problem is equivalent to a special case of the constrained multiple sequence alignment problem which can also be solved with the same time complexity.

Keywords: Constrained longest common subsequence; Algorithms; Dynamic programming; Sequence alignment

1. Introduction

The longest common subsequence (LCS) problem has several applications in many apparently unrelated fields, such as computer science, mathematics, molecular biology, speech recognition, gas chromatography. In molecular biology, LCS is an appropriate measure of the similarity of biological sequences. Indeed, if we wish to compare two strands of DNA or two protein sequences, we may compute the LCS of them. The LCS problem on multiple sequences is NP-hard [4]. However, it may be solved in polynomial time for two sequences.

Many algorithms have been designed using the dynamic programming technique on this problem [3,5,8]. Tsai addressed a variant of the LCS problem, *the constrained longest common subsequence* (CLCS) problem [7]. Given two sequences X, Y and a constrained sequence P, compute the longest common subsequence Z of X and Y such that P is a subsequence of Z. An $O(n^2 \cdot m^2 \cdot r)$ time algorithm based

∗ Corresponding author.
E-mail addresses: chin@csis.hku.hk (F.Y.L. Chin), ads@dia.unisa.it (A. De Santis), ferrara@dia.unisa.it (A.L. Ferrara), nlho@csis.hku.hk (N.L. Ho), skkim@cau.ac.kr (S.K. Kim).
1 This research was supported in part by Hong Kong RGC Grant.

0020-0190/$ – see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.ipl.2004.02.008

on the dynamic programming technique was proposed for this problem by Tsai [7], where n, m and r are the lengths of X, Y and P, respectively.

In this paper, we present a simple algorithm to solve the CLCS problem in $O(n \cdot m \cdot r)$ time. We further show that this CLCS problem is equivalent to a special case of the constrained sequence alignment (CSA) problem which can be solved with the same time complexity.

The rest of this paper is organized as follows. In Section 2, we define formally the CLCS problem and characterize the structure in computing the CLCS. In Section 3, we present a simple dynamic programming algorithm that computes the CLCS of two sequences with respect to a constrained sequence. In Section 4, we show that the CLCS problem is equivalent to a special case of the constrained sequence alignment problem [1,6].

2. Characterization of the constrained LCS problem

A sequence is a string of characters over a set of alphabets Σ. A subsequence Z of a sequence X is obtained by deleting some characters from X (not necessarily contiguous); we also say that X *contains* Z if Z is a subsequence of X. Given two sequences X and Y, Z is a common subsequence of X and Y if Z is a subsequence of both X and Y. Z is the *longest common subsequence* (*LCS*) of X and Y if Z is the longest among all common subsequences of X and Y. For example, "`lm`" and "`rm`" are both the longest common subsequence of "`problem`" and "`algorithm`". Let P be a constrained sequence. We say that Z is the *constrained LCS* of X and Y with respect to P if Z is the longest subsequence of X and Y and Z contains P (i.e., P is a subsequence of Z). For example, "`lm`" is the longest common subsequence of "`problem`" and "`algorithm`" with respect to "`l`".

Given a sequence $X = \langle x_1, x_2, \ldots, x_n\rangle$, where character $x_i \in \Sigma$ for any $i = 1, \ldots, n$, we denote the ith prefix of X by $X_i = \langle x_1, x_2, \ldots, x_i\rangle$ for any $i = 1, \ldots, n$. In particular, X_0 denotes the empty sequence. For example, if $X =$ "`algorithm`" then $X_3 =$ "`alg`".

The following theorem characterizes the structure of an optimal solution based on optimal solutions to subproblems, for the constrained LCS problem.

Theorem 2.1. *If $Z = \langle z_1, z_2, \ldots, z_\ell\rangle$ is the constrained LCS of $X = \langle x_1, x_2, \ldots, x_n\rangle$ and $Y = \langle y_1, y_2, \ldots, y_m\rangle$ with respect to $P = \langle p_1, p_2, \ldots, p_r\rangle$, the following conditions hold*:

1. *If $x_n = y_m = p_r$ then $z_\ell = x_n = y_m = p_r$ and $Z_{\ell-1}$ is the constrained LCS of X_{n-1} and Y_{m-1} with respect to P_{r-1}.*
2. *If $x_n = y_m$ and $x_n \neq p_r$ then $z_\ell = x_n = y_m$ and $Z_{\ell-1}$ is the constrained LCS of X_{n-1} and Y_{m-1} with respect to P.*
3. *If $x_n \neq y_m$ then $z_\ell \neq x_n$ implies that Z is a constrained LCS of X_{n-1} and Y with respect to P.*
4. *If $x_n \neq y_m$ then $z_\ell \neq y_m$ implies that Z is a constrained LCS of X and Y_{m-1} with respect to P.*

Proof. As Z is the constrained LCS of X and Y with respect to P, x_n, y_m and z_ℓ are the last characters of X, Y and Z, respectively, we have $z_\ell = x_n = y_m$ if $x_n = y_m$. Assume by contradiction that $z_\ell \neq x_n$, we may append $x_n = y_m$ to Z to obtain a constrained common subsequence of X and Y of length $\ell + 1$, contradicting the hypothesis that Z, of length ℓ, is a constrained LCS of X and Y with respect to P. Therefore, if $x_n = y_m$ then $z_l = x_n = y_m$. This will be used in the proofs of 1 and 2. Now, we prove the four properties.

Proof of 1. Since $x_n = y_m = p_r$, we have $x_n = y_m = p_r = z_\ell$. The prefix $Z_{\ell-1}$ is a common subsequence of X_{n-1} and Y_{m-1} with respect to P_{r-1} of length $\ell - 1$. Now, we show that $Z_{\ell-1}$ is a constrained LCS of X_{n-1} and Y_{m-1} with respect to P_{r-1}. Assume by contradiction that there exists a constrained common subsequence S of X_{n-1} and Y_{m-1} with respect to P_{r-1} whose length is greater than $\ell - 1$. If we append $x_n = y_m = p_r$ to S we obtain a constrained common subsequence of X and Y with respect to P of length greater than ℓ, contradicting the hypothesis that Z is a constrained LCS of X and Y with respect to P.

Proof of 2. Since $x_n = y_m$ and $x_n \neq p_r$, then $x_n = y_m = z_\ell$ and $z_\ell \neq p_r$. As $z_\ell \neq p_r$, the prefix $Z_{\ell-1}$ is a common subsequence of X_{n-1} and Y_{m-1} with respect to P of length $\ell - 1$. Now, we show that $Z_{\ell-1}$ is a constrained LCS of X_{n-1} and Y_{m-1} with respect to P. Assume by contradiction that there exists a constrained common subsequence S of X_{n-1} and Y_{m-1} with respect to P whose length is greater than $\ell-1$. If we append $x_n = y_m$ to S, we obtain a constrained common subsequence of X and Y with respect to P of length greater than ℓ, contradicting the hypothesis that Z is a constrained LCS of X and Y with respect to P.

Proof of 3. Since $z_\ell \neq x_n$, Z is a constrained common subsequence of X_{n-1} and Y with respect to P. Now, we show that Z is a constrained LCS of X_{n-1} and Y with respect to P. Assume by contradiction that there exists a constrained common subsequence S of X_{n-1} and Y with respect to P whose length is greater than ℓ, then S also is a constrained common subsequence of X and Y with respect to P whose length is greater than ℓ. This contradicts the assumption that Z is a constrained LCS of X and Y with respect to P.

Proof of 4. The proof is similar to proof of 3. □

The next theorem shows a characterization of the constrained LCS problem when no constrained common subsequence exists.

Theorem 2.2. *If there is no constrained common subsequence of* $X = \langle x_1, x_2, \ldots, x_n\rangle$ *and* $Y = \langle y_1, y_2, \ldots, y_m\rangle$ *with respect to* $P = \langle p_1, p_2, \ldots, p_r\rangle$, *the following conditions hold*:

1. *If* $x_n = y_m = p_r$ *then there is no constrained common subsequence of* X_{n-1} *and* Y_{m-1} *with respect to* P_{r-1}.
2. *There is no constrained common subsequence of the two sequences* X' *and* Y' *with respect to* P, *for each of the following three cases*:
   - $X' = X_{n-1}$ *and* $Y' = Y_{m-1}$;
   - $X' = X_{n-1}$ *and* $Y' = Y$;
   - $X' = X$ *and* $Y' = Y_{m-1}$.

Proof of 1. Assume by contradiction that there is no constrained common subsequence of X and Y with respect to P but there exists a constrained common subsequence Z of X_{n-1} and Y_{m-1} with respect to P_{r-1}. Since $x_n = y_m = p_r$ then the concatenation of x_n to Z is a constrained common subsequence of X and Y with respect to P. Contradiction.

Proof of 2. Assume by contradiction that there is no constrained common subsequence of X and Y with respect to P but there exists a constrained common subsequence Z of X' and Y' with respect to P. This is a contradiction because Z is also a constrained common subsequence of X and Y with respect to P.

3. A simple algorithm

Given two sequences X, Y and a constrained sequence P, whose lengths are n, m and r, respectively, we define $L(i, j, k)$ as the constrained LCS length of X_i and Y_j with respect to P_k, for any $0 \leqslant i \leqslant n$, $0 \leqslant j \leqslant m$ and $0 \leqslant k \leqslant r$. In particular, $L(n, m, r)$ gives the length of the constrained LCS of X and Y with respect to P. We design an algorithm that computes the CLCS of X and Y with respect to P in $O(n \cdot m \cdot r)$ time.

If either $i < k$ or $j < k$, there is no constrained common subsequence for X_i and Y_j with respect to P_k. We represent this condition by denoting $L(i, j, k) = -\infty$, where ∞ represents a large number, greater than the maximum value of n and m. If $i = 0$ or $j = 0$ and $k = 0$, the CLCS for X_i and Y_j with respect to P_0 has length 0. The characterization of the structure of a solution for the CLCS problem based on solutions to subproblems shown in Section 2, yields the following recursive relation, for any $0 \leqslant i \leqslant n$, $0 \leqslant j \leqslant m$ and $0 \leqslant k \leqslant r$,

$$L(i,j,k) = \begin{cases} 1 + L(i-1, j-1, k-1) \\ \quad \text{if } i, j, k > 0 \text{ and } x_i = y_j = p_k, \\ 1 + L(i-1, j-1, k) \\ \quad \text{if } i, j > 0,\ x_i = y_j \text{ and} \\ \quad (k = 0 \text{ or } x_i \neq p_k), \\ \max(L(i-1, j, k), L(i, j-1, k)) \\ \quad \text{if } i, j > 0 \text{ and } x_i \neq y_j \end{cases} \tag{1}$$

with boundary conditions,

$$L(i, 0, 0) = L(0, j, 0) = 0$$

and

$$L(0, j, k) = L(i, 0, k) = -\infty,$$

for $i = 0, \ldots, n$, $j = 0, \ldots, m$ and $k = 1, \ldots, r$.

This is a generalization of the recurrence formula that computes the length of an LCS between two sequences [2], indeed, if $k = 0$, it holds that

$$L(i,j,0) = \begin{cases} 1 + L(i-1, j-1, 0) \\ \quad \text{if } i, j > 0,\ x_i = y_j, \\ \max(L(i-1, j, 0), L(i, j-1, 0)) \\ \quad \text{if } i, j > 0,\ x_i \neq y_j, \\ 0 \quad \text{if } i = 0 \text{ or } j = 0. \end{cases} \tag{2}$$

Constructing the constrained LCS. The CLCS of X and Y with respect to P can be constructed by backtracking through the computation path from $L(n,m,r)$ to $L(0,0,0)$. Let Z, the initial CLCS, be an empty sequence. If the value of $L(i, j, k)$ is computed from $L(i-1, j-1, k)$ or $L(i-1, j-1, k-1)$, prepend the character $x_i (= y_j)$ to Z. Repeat backtracking until reaching $L(0,0,0)$, and Z is the CLCS of X and Y with respect to P. Recovering the computation path of the CLCS takes at most $O(n + m + r)$ steps.

Thus, computing and constructing the CLCS takes $O(n \cdot m \cdot r)$ time and space.

4. CLCS and constrained sequence alignment

In this section, we show that the CLCS problem is in fact a special case of the *constrained sequence alignment* (CSA) problem [1,6].

Let $X = \langle x_1, x_2, \ldots, x_n\rangle$ and $Y = \langle y_1, y_2, \ldots, y_m\rangle$ be two sequences over Σ, with lengths n and m, respectively. We define the *sequence alignment* of X and Y as two equal-length sequences $X' = \langle x'_1, x'_2, \ldots, x'_{n'}\rangle$ and $Y' = \langle y'_1, y'_2, \ldots, y'_{n'}\rangle$ such that $|X'| = |Y'| = n'$, where $n' \geqslant n, m$, and removing all space characters "-" from X' and Y' gives X and Y, respectively, with the assumption that no $x'_i = y'_i =$ "-" for any $1 \leqslant i \leqslant n'$. For a given distance function $\delta(x', y')$ which measures the *mutation* distance between two characters, where $x', y' \in \Sigma \cup \{\text{-}\}$, the *alignment score* of two length-n' sequences X' and Y' is defined as

$$\sum_{1 \leqslant i \leqslant n'} \delta(x'_i, y'_i).$$

In the *constrained sequence alignment* (*CSA*) *problem*, we are given, in addition to the inputs of the sequence alignment problem, a constrained sequence $P = \langle p_1, p_2, \ldots, p_k\rangle$, where P is a common subsequence of X and Y. A solution of the CSA problem is $\left[\begin{smallmatrix} X' \\ Y' \end{smallmatrix}\right]$, an alignment of X and Y, such that when X' is placed on top of Y', each character in P appears in an entire column of the alignment and in the same order as P, i.e., there exists a list of integers $\langle c_1, c_2, \ldots, c_r\rangle$ where $1 \leqslant c_1 < c_2 < \cdots < c_r \leqslant n'$, and for all $1 \leqslant k \leqslant r$, we have $x'_{c_k} = y'_{c_k} = p_k$. The CSA problem is to find X' and Y' with minimum alignment score when given two sequences X, Y, a constrained sequence P and a distance function $\delta(x', y')$.

The CSA problem can also be solved in $O(n \cdot m \cdot r)$ time and space [1]. Next, we show that the CLCS problem is equivalent to the CSA problem of X and Y with respect to P, using the distance function $\delta(x', y')$, where $x', y' \in \Sigma \cup \{\text{-}\}$,

$$\delta(x', y') = \begin{cases} -1 & \text{if } x' = y' \text{ (match)} \\ 0 & \text{if } x' \neq y' \text{ (insertion, deletion} \\ & \text{or replacement).} \end{cases} \tag{3}$$

The distance function $\delta(x', y')$ in Eq. (3) favors matching characters, and does not penalize mismatching characters or insertion of spaces. Therefore, when the CSA alignment score is $-s$, there are s matchings in X and Y with respect to P.

Theorem 4.1. *Given two sequences X, Y and a constrained sequence P, the CLCS of problem is equivalent to the CSA problem when the distance function $\delta(x', y')$ given in Eq.* (3) *is used.*

Proof. Let $\left[\begin{smallmatrix} X' \\ Y' \end{smallmatrix}\right]$ be the CSA solution with the minimum alignment score, $n' = |X'| = |Y'|$. By the definition of CSA with the distance function $\delta(x', y')$, $\left[\begin{smallmatrix} X' \\ Y' \end{smallmatrix}\right]$ has the minimum alignment score only if X' and Y' have the most number of matches (i.e., $x'_i = y'_i$) and every character in P appears as a column in $\left[\begin{smallmatrix} X' \\ Y' \end{smallmatrix}\right]$. Let Z' be the subsequence of X' and Y', containing *only* the matching characters in X' and Y'. Obviously, Z' is a common subsequence of X' and Y' containing P. Thus, Z' is the CLCS of X and Y with respect to P.

Let Z be a CLCS solution of X and Y with respect to P and $|Z| = \ell$. By definition of CLCS, P is a subsequence of Z and Z is a common subsequence of both X and Y. To obtain an optimal solution for the CSA problem of X and Y with respect to P, we can construct X' and Y' by inserting spaces into X and Y respectively, such that every character in Z appears in the same position in X' and Y'. Using the distance function $\delta(x', y')$, the alignment score of $\left[\begin{smallmatrix}X'\\Y'\end{smallmatrix}\right]$ is $-\ell$, i.e., the alignment $\left[\begin{smallmatrix}X'\\Y'\end{smallmatrix}\right]$ has ℓ matching columns. Since P is a subsequence of Z, there is a column matching in $\left[\begin{smallmatrix}X'\\Y'\end{smallmatrix}\right]$ for every character in P. As Z is a CLCS of X and Y with respect to P, the optimal CSA solution for X and Y with respect to P has at most ℓ matches, i.e., with the minimum alignment score $-\ell$. Hence, $\left[\begin{smallmatrix}X'\\Y'\end{smallmatrix}\right]$ is the optimal CSA solution for X and Y with respect to P. $\square$

5. Conclusions

In this paper, we have addressed the constrained longest common subsequence problem proposed by Tsai [7]. An $O(n^2 \cdot m^2 \cdot r)$ time algorithm based on the dynamic programming technique was proposed to compute a CLCS for X and Y with respect to P, where n, m and r are the lengths of X, Y and P, respectively. We have described a simple $O(n \cdot m \cdot r)$ time algorithm to solve this problem and have also showed that the CLCS problem is indeed a special case of the CSA problem.

It is not difficult to show that this problem can also be solved in $O(\min(n, m) \cdot r)$ space based on Hirschberg's Algorithm [3]. We define $L^r(i, j, k)$ as the length of CLCS of the suffices of X, Y and P starting from the ith, jth and kth positions, respectively. WLOG, assume $n \geqslant m$, then Z, the solution for the CLCS problem, can be defined as

$$\max_{0\leqslant j\leqslant m, 0\leqslant k\leqslant r}\Big(L\big(n/2, j, k\big) + L^r\big(n/2+1, j+1, k+1\big)\Big)$$

which can be found in $O(n \cdot m \cdot r)$ time (say $Knmr$ time, where K is a constant) and $O(m \cdot r)$ space. Assume the maximum value occurs when $j = j'$ and $k = k'$, then we can further solve the two subproblems $L(\frac{1}{2}n, j', k')$ and $L^r(\frac{1}{2}n + 1, j' + 1, k' + 1)$ in $\frac{1}{2}Kmnr$ time and $O(m \cdot r)$ space. Continuing this recursively, we can solve the CLCS problem in the same $O(n \cdot m \cdot r)$ time and $O(m \cdot r)$ space complexities.

References

[1] F.Y.L. Chin, N.L. Ho, T.W. Lam, P.W.H. Wong, M.Y. Chan, Efficient constrained multiple sequence alignment with performance guarantee, in: Proceedings of the IEEE Computer Society Conference on Bioinformatics, IEEE Computer Society, 2003, pp. 337–346.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.

[3] D.S. Hirschberg, Algorithms for the longest common subsequence problem, J. ACM 24 (1977) 664–675.

[4] D. Mayer, The complexity of some problems on subsequences and supersequences, J. ACM 25 (1978) 322–336.

[5] W.J. Masek, M.S. Paterson, A faster algorithm computing string edit distances, J. Comput. System Sci. 20 (1980) 18–31.

[6] C.Y. Tang, C.L. Lu, M.D.T. Chang, Y.T. Tsai, Y.J. Sun, K.M. Chao, J.M. Chang, Y.H. Chiou, C.M. Wu, H.T. Chang, W.I. Chou, Constrained multiple sequence alignment tool development and its application to RNase family alignment, J. Bioinform. Comput. Biol. 1 (2003) 267–287.

[7] Y.T. Tsai, The constrained longest common subsequence problem, Inform. Process. Lett. 88 (2003) 173–176.

[8] R.A. Wagner, M.J. Fischer, The string-to-string correction problem, J. ACM 21 (1974) 168–173.