A-17　変更制約付き最長共通部分列問題に対する多項式時間アルゴリズム

高雄 奏摩*，新竹 優駿*，江藤 宏*，宮野 英次*，斎藤 寿樹*，塩田 拓海* **
（*九州工業大学，**日本学術振興会特別研究員 DC）

１．はじめに

2 つ文字列 $S=(s_1,s_2,\dots,s_n)$ と $T=(t_1,t_2,\dots,t_m)$ に対し，S の部分列かつ T の部分列である共通部分列のうち最長のもの**最長共通部分列(LONGEST-COMMON SUBSEQUENCE PROBLEM, LCS)**という．例えば，$S=$ abcdefと $T=$ fcaedebf のLCSは adef であり長さは4となる．また，LCSの長さを求める最適化問題を最長共通部分列問題(LCS問題)という．LCS問題は，入力列がS とT の2つの場合，動的計画法を用いて$O(|S||T|)$ の時間計算量で解くことができる[1]．

最適化問題に対して初期解を与え，変更を加えることで最適解を求める増分最適化問題がある[2]．本研究では2 つの文字列に対するLCS問題を増分最適化問題として扱う，変更制約付き最長共通部分列問題 (BOUNDED-DELETION LONGEST-COMMON SUBSEQUENCE PROBLEM，BD-LCS問題) を考える．以降，列 X の長さを $|X|$，i 文字目を $X[i]$，i 文字目から j 文字目の部分文字列を $X[i{:}j]$ と表記する．

変更制約付き最長共通部分列問題 (BD-LCS 問題)
入力：2 つの文字列 S,T，初期共通部分列 Z，S に対する添字列 $A(a_0,a_1,\dots,a_{|Z|-1})$，$T$ に対する添字列 $B(b_0,b_1,\dots,b_{|Z|-1})$，非負整数 k.
条件：$S[a_i]=T[b_i]=Z[i]$ $(0\le i\le |Z|-1)$
目的：k 個以下の$(S[a_i],T[b_i])$ の組みを削除するとき，最長共通部分列の長さを求める．

例として図 1 の場合を考える．このときのBD-LCS問題に対する解は，組み $(S[4],T[0])$ を削除し，新たに 4 つの組みを追加した abcde であり長さは 5 となる．本稿では，BD-LCS問題に対し$O(|S||T||Z|)$ の時間計算量で解く多項式時間アルゴリズムを示す．

２．LCS問題に対する既存アルゴリズム

2 つの文字列 S, T に対して，動的計画法を用いてLCS問題を解くアルゴリズム[1]を説明する．二次元配列 dp を導入する．ここで，$\mathrm{dp}[i][j]$ を $S[0{:}i]$と $T[0{:}j]$ のLCSの長さと定義する $(0\le i\le |S|,\ 0\le j\le |T|)$．このとき $\mathrm{dp}[i+1][j+1]$ は，下記の漸化式を用いて求めることができる．

$$\mathrm{dp}[i+1][j+1]=\begin{cases}\max(\mathrm{dp}[i+1][j],\mathrm{dp}[i][j+1]) & ,\ (S[i]\neq T[j])\\ \max(\mathrm{dp}[i][j]+1,\mathrm{dp}[i+1][j],\mathrm{dp}[i][j+1]), & (S[i]=T[j])\end{cases}$$

この漸化式における初期条件は $\mathrm{dp}[0][j]=\mathrm{dp}[i][0]=0$ である．これは，S もしくは T の部分文字列と空文字列のLCSの長さが0 であることから求まる．

上記の漸化式を用いて，二次元配列 dp の全ての値が計算できる．また，$\mathrm{dp}[|S|][|T|]$ にはS と T のLCSの長さが格納され，これがLCS問題の解となる．

本アルゴリズムは，dp の全ての値の計算に $(|S|+1)\times(|T|+1)$ ステップを要するため，時間計算量は $O(|S||T|)$ となる．

３．BD-LCS問題に対する提案アルゴリズム

BD-LCS問題の**目的**は次のように捉えることができる．

目的：$(|Z|-k)$ 個以上の$(S[a_i],T[b_i])$ の組みからなる，最長共通部分列の長さを求める．

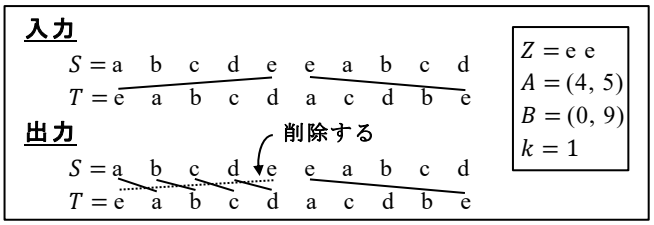

図 1 2 つの文字列 S,T に対する BD-LCS 問題の例

新たに三次元配列 dp を導入する．ここで，$\mathrm{dp}[i][j][\ell]$ を $S[0{:}i]$と $T[0{:}j]$ のLCSのうち，ちょうど ℓ 個の$(S[a_i],T[b_i])$ の組みを用いたものとして定義する．なお，$\mathrm{dp}[i][j][\ell]$ には，ℓ 個以外の組みは格納されない点に注意をする．例えば $\ell=0$ のとき，解として得られるLCSには $(S[a_i],T[b_i])$ の全ての組みが一つも含まれない．

$P=\{(a_0,b_0),(a_1,b_1),\dots,(a_{|Z|-1},b_{|Z|-1})\}$ を添字列のペアの集合とする．このとき $\mathrm{dp}[i+1][j+1][\ell+1]$ は下記の漸化式を用いて求めることができる．

$$\mathrm{dp}[i+1][j+1][\ell+1]=\begin{cases}\max(\mathrm{dp}[i][j][\ell+1],\mathrm{dp}[i+1][j][\ell+1],\\ \quad \mathrm{dp}[i][j+1][\ell+1]), & (S[i]\neq T[j])\\ \max(\mathrm{dp}[i][j][\ell+1]+1,\mathrm{dp}[i+1][j][\ell+1],\\ \quad \mathrm{dp}[i][j+1][\ell+1]), & (S[i]=T[j]\ \text{and}\ (i,j)\notin P)\\ \max(\mathrm{dp}[i][j][\ell]+1,\mathrm{dp}[i+1][j][\ell+1],\\ \quad \mathrm{dp}[i][j+1][\ell+1]), & (S[i]=T[j]\ \text{and}\ (i,j)\in P)\end{cases}$$

この漸化式における初期条件は，$\mathrm{dp}[0][j][\ell]=\mathrm{dp}[i][0][\ell]=0$ である．これは，ℓ の値に関わらず S もしくは T の部分文字列と空文字列のLCSの長さが 0 となるからである．

上記の漸化式を用いて，三次元配列dp の全ての値を計算すると，ちょうど ℓ 個の$(S[a_i],T[b_i])$ の組みからなるLCSの長さが，$\mathrm{dp}[|S|][|T|][\ell]$ に格納される．一方，BD-LCS問題において求める解は，$(|Z|-k)$ 個以上の$(S[a_i],T[b_i])$ の組みからなるLCSの長さである．ゆえに，$|Z|-k\le\ell\le|Z|$としたときの $\mathrm{dp}[|S|][|T|][\ell]$ の最大値が，本問題の解となる．

本アルゴリズムでは，三次元配列 dp の全ての値の計算に $(|S|+1)\times(|T|+1)\times(|Z|+1)$ ステップ，$|Z|-k\le\ell\le|Z|$ の中から $\mathrm{dp}[|S|][|T|][\ell]$ の最大値を求めるのに，k ステップを要する．ゆえに，BD-LCS問題における時間計算量は$O(|S||T||Z|)$ となる．

謝辞

本研究の一部は，JSPS科研費JP24K02902，JP24KJ1816 の助成を受けたものである．

参考文献

[1] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. Journal of the ACM, 21(1), 168-173. ACM, 1974.
[2] Onur Şeref, Ravindra K. Ahuja, and James B. Orlin. Incremental Network Optimization: Theory and Algorithms. Oper. Res., 57(3):586–594, May 2009.