文字列集合に対する多様な最長共通部分列の発見

志田 祐仁 [1]　　有村 博紀 [2]　　小林 靖明 [2]

[1] 北海道大学大学院情報科学院
[2] 北海道大学大学院情報科学研究院
E-mail:shida.yuto.d4@elms.hokudai.ac.jp,
{arim,koba}@ist.hokudai.ac.jp

概 要

本論文では，文字列解析における最長共通部分列（LCS）問題を拡張し，m 本の入力文字列からなる集合 $\mathcal{S}$ の指数個ある LCS 全体を考えて，それらから選択した k 本の LCS の組合せで，多様性尺度を最大化するものを求める問題を考察する．多様性尺度として，k 本からとった全ての文字列対のハミング距離の最小の最大化問題（MMD(LCS)）と総和の最大化問題（MSD(LCS)）を考える．主結果として，文字列数 m と解サイズ k が定数のときは，LCS 間のハミング距離に関する最小多様性最大化問題 MMD(LCS) と総和多様性最大化問題 MSD(LCS) の両方が多項式時間計算可能であることを示す．関連した結果として，解サイズ k が非限定の場合には，上記の二つの問題は共に NP 困難になることが知られており，提案手法は，文字列データ解析に関わる多様性問題に関する効率良い厳密解法として有用と思われる．

1　はじめに

1.1　研究背景

近年，多様な解を効率的に求める方法への注目が高まっている [4, 6]．本論文では，遺伝情報解析や系列解析においてよく知られた問題である，**最長共通部分列**（*longest common subsequence, LCS*）を求める問題を拡張して，ただ一つの LCS でなく，k 個の多様な LCS 集合を求める問題を考察する．LCS は，複数の系列データ間の共通性を調べるためのパターンや類似性の根拠として用いられることが多い．本稿の手法で，多様な LCS の集合が得られることで，解析における複数の選択肢を得ることができ，実際の系列解析に有用な手段となると期待される．

1.2　本稿の枠組み

文献 [1, 4, 6] にしたがって，多様解発見の枠組みを導入する．一般に，任意の集合 $\mathcal{X}$ と距離関数 d を考える．本稿では，$\mathcal{X}$ はある最適化問題 X の解集合と仮定する．本稿では解集合の多様性の尺度として，d に関する最小および総和の多様性制約 $D_d^{\min}$ と D_d^{sum} を用いる:

$$D_d^{\min}(A) := \min_{i<j} d(Z_i, Z_j),$$
$$D_d^{\mathrm{sum}}(A) := \sum_{i<j} d(Z_i, Z_j).$$

ここに，$A = \{Z_1, \ldots, Z_k\} \subseteq \mathcal{X}$ は，$\mathcal{X}$ の任意の部分集合である．便宜上，$|A| \leqslant 1$ のとき，$D_d^{\min}(A) = \infty$ と $D_d^{\mathrm{sum}}(A) = 0$ と定める．

このとき，**最小多様性（最大多様性）の最大化問題**（*Max-Min (Max-Sum) diversification*）は，入力として，（非明示的に与えられた）許容解集合 $\mathcal{X}$，$\mathcal{X}$ 上の距離関数 $d : \mathcal{X}^2 \to \mathbb{R}_{\geqslant 0}$，非負整数 $k \geqslant 1$ と $\Delta \geqslant 0$ が与えられたとき，タスクとして，要素数制約 $|A| = k$ と，d に関する最小多様性制約 $D_d^{\min}(A) \geqslant \Delta$ （総和多様性制約 $D_d^{\mathrm{sum}}(A) \geqslant \Delta$）を満たす部分集合 $A \subseteq \mathcal{X}$ を求める問題である．

1.3　研究目的

ある文字列 Z が別の文字列 Y からいくつかの文字を削除することによって得られるとき，Z は Y に（**非連続な**）**部分列**として含まれるという．$\mathcal{S}$ の**最長共通部分列**（*longest common subsequence, LCS*） は，$\mathcal{S}$ 中のすべての文字列に部分文字列として含まれる文字列のうち， 長さが最長のものをいう．

一般に，$m \geqslant 2$ 本の入力文字列からなる $\mathcal{S}$ は，入力文字列長 n の指数個の最長共通部分列をもつ．入力文字列の本数 m が非限定のとき，一つの最長共通部分列を求める問題（**最長共通部分列問題**）は，NP 困難であることが知られている [5]．一方で m が定数ならば，一つの最長共通部分列は多項式時間で求められる [9]．

本稿では，解集合として，入力文字列集合 $\mathcal{S}$ のすべての LCS からなる集合 $LCS(\mathcal{S})$ と，LCS 間の距離として文字

列のハミング距離 d_{HD} を仮定し，次のように定義される最小多様性（総和多様性）の最大化問題を考える．

定義 1（**ハミング距離 d_{HD} に関する LCS の最小多様性（総和多様性）の最大化問題**，MMD(LCS) (MSD(LCS))）. 入力として，文字列集合 $\mathcal{S} = \{X_1, \ldots, X_m\} \subseteq \Sigma^*$ $(m \geqslant 1)$ と，非負整数 $k \geqslant 1, \Delta \geqslant 0$ を受け取り，タスクとして，$D^{\min}_{d_{HD}}(A) \geqslant \Delta$（$D^{\mathrm{sum}}_{d_{HD}}(A) \geqslant \Delta$, resp.）を満たすサイズ $|A| = k$ の集合 $A \subseteq LCS(\mathcal{S})$ があるかを判定する．

明示的な有限集合 $\mathcal{X}$ に対する多様性最大化問題と，ここで考察する多様性問題との最も大きな違いは，後者では対象となる集合 $LCS(\mathcal{S})$ の要素数が，入力サイズ $||\mathcal{S}||$ に対して指数的に大きいことである．そのため，単純な解の探索では，効率良いアルゴリズムを構成できない．

1.4 主結果

本稿の主結果として，入力文字列数 $m \geqslant 2$ と解数 $k \geqslant 2$ が定数のとき，MMD(LCS) と MSD(LCS) 問題の両方を厳密に解く多項式時間アルゴリズムを与える（定理 1）．さらに，提案アルゴリズムに基づいて，制約を見たす解集合 A が存在すれば，実際に A を出力することができる（補題 6）．

提案アルゴリズムは，解数 k とハミング距離の下限 $\Delta \leqslant n$ に関して，$O(k^2\Delta^{k^2}n^{2km})$ 時間と $O(\Delta^{k^2}n^{km})$ 領域で動作する．そのために，全ての LCS を表現する多項式サイズの辺ラベル付き DAG（または非決定性有限オートマトン）の効率的な構築法と，k 本の LCS の組合せがもつ距離行列（打ち切り相互ハミング重み行列）に関する動的計画法を開発し，効率良い計算を実現している．

最近，解数 $k \geqslant 2$ が非限定の場合に，入力文字列数 $m \geqslant 2$ が定数であっても，MMD(LCS) と MSD(LCS) 問題の両方の NP 困難性が示されている [10]．これらから，計算量上の仮定 $P \neq NP$ のもとで，これらの問題を厳密に解く多項式時間アルゴリズムは存在しない．今後の課題として，LCS の多様解問題に対する固定パラメータアルゴリズムや近似アルゴリズムの有無は興味深い未解決問題である．

1.5 関連研究

多様性最大化問題．集合の**分散度**（*dispersion*）は古くから研究されている多様性尺度である．とくに，1.2節で導入する最小および総和の多様性制約 $D^{\min}_d$ と D^{sum}_d は代表的な分散度である [2]．Hanaka ら [6] は，組合せ最適化問題における多様性最大化問題を効率的に解くための一般的な枠組みを提案している．

LCS の DAG については，関連研究として [3] らが，$m = 2$ に限定した場合に入力文字列対 $\mathcal{S} = \{X_1, X_2\}$ の極大共通列すべての集合 $MCS(\mathcal{S})$ を表現する多項式サイズの決定性有限オートマトンを与えている．ただし，一般の $m \geqslant 2$ の場合の構成法は未解決問題である．

2 準備

$\mathbb{N} = \{0, 1, \ldots, \}$ と，$\mathbb{R}$，$\mathbb{R}_{\geqslant 0} = \{\, x \in \mathbb{R} \mid x \geqslant 0 \,\}$ で，それぞれ，全ての非負整数，全ての実数，全ての非負実数からなる集合を表す．任意の非負整数 $0 \leqslant i \leqslant j \leqslant n$ に対して，$[n]$ と $[i\,..\,j]$ はそれぞれ，集合 $\{1, \ldots, n\}$ と整数の区間 $\{i, i+1, \ldots, j\}$ を表す．任意の集合 $\mathcal{X}$ 上の値の k-項組 $x \in \mathcal{X}^k$ に対して，その第 $i \in [k]$ 成分 a_i を x_i と書く．すなわち，$x = (x_1, \ldots, x_k)$ である．有向グラフ中のパスは，辺で互いに接続された頂点の列 $\pi = (v_0, \ldots, v_h)$ とし，その長さを辺数 h で定める．

文字列．$\Sigma = \{a, b, \ldots\}$ を文字のアルファベットとする．長さ 0 の空文字列を ε で表す．長さ $n \geqslant 0$ の文字列 $S = a_1 \ldots a_n \in \Sigma^n$ の i 番目の文字を $S[i] = a_i$ で表す．任意の添字 $i, j \in [n]$ に対して，もし $i \leqslant j$ なら，添字 i から j までの連続部分文字列を $S[i\,..\,j] = a_i a_{i+1} \ldots a_j$ で表し，$i > j$ の場合は $S[i\,..\,j] = \varepsilon$ とする．S が Σ 上の長さ n の文字列であることを，$S[1\,..\,n] \in \Sigma^n$ と表す．以降では，添字を許して，文字を $a, b, \ldots$ と，文字列を $X, Y, \ldots$，文字列の集合を $\mathcal{S}, \ldots$，要素を $p, q, \ldots$，要素の組を $\vec{p}, \vec{q}, \ldots$，等と表記する．

最長共通部分列．Σ 上の $m \geqslant 1$ 個の文字列からなる任意の文字列集合を $\mathcal{S} = \{X_1, \ldots, X_m\} \subseteq \Sigma^*$ とおく．集合 $\mathcal{S}$ の**最長共通部分列**（*longest common subsequence, LCS*）は，最長の共通部分列の任意の一つをいう．このとき，最長共通部分列の長さを $lcs(\mathcal{S}) = \max\{|Z| : Z \in CS(\mathcal{S})\} \in [0\,..\,\min_{X \in \mathcal{S}} |X|]$ で表す．$\mathcal{S} = \{X_1, \ldots, X_m\}$ の共通部分列全体と最長共通部分列全体を，それぞれ，$CS(\mathcal{S})$ と $LCS(\mathcal{S}) = \{\, Z \in CS(\mathcal{S}) : |Z| = lcs(\mathcal{S}) \,\}$ と表す．

例 1. 図 1に，アルファベット $\Sigma = \{A, B, C, D, E\}$ 上の長さ $n = 9$ の二つの入力文字列 $X = X_1 = ABABCDDEE$ と $Y = Y_1 = ABCBAEEDD$ のすべての共通部分列を示す．そのうち，最長共通部分列に下線を引く．

距離尺度．任意の集合を $\mathcal{X}$ とし，任意の非負値関数を $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geqslant 0}$ とする．このとき，d が**準距離関数**（*semi-metric*）とは，任意の $x, y \in \mathcal{X}$ に対して，次(i) $d(x, y) = 0 \iff x = y$（分離性）

$$\epsilon, A, B, C, D, E,$$
$$AA, AB, AC, AD, AE, BA, \ldots, CD, CE, DD, EE,$$
$$ABA, ABB, ABC, ABD, \ldots, CEE,$$
$$ABAD, ABAE, ABBD, \ldots, BCEE,$$
$$\underline{ABADD}, \underline{ABAEE}, \underline{ABBDD},$$
$$\underline{ABBEE}, \underline{ABCDD}, \underline{ABCEE}$$

図 1: 入力文字列 $X = ABABCDDEE$ と $Y = ABCBAEEDD$ のすべての共通部分列．最長共通部分列には下線を引いている．

と (ii) $d(x,y) = d(y,x)$（対称性）を満たすときをいう．準距離 d が**距離関数**（*metric*）であるとは，任意の $x, y, z \in \mathcal{X}$ に対して，次をいう: (iii) $d(x,y) \leqslant d(x,y) + d(y,z)$（三角不等式）．

任意の $p \in \mathbb{N}$ と任意の $x, y \in \mathbb{R}^p$ に対して，$\mathbb{R}^p$ 上の距離関数である ℓ_1 距離 d_{ℓ_1} と ℓ_2 距離 d_{ℓ_2} を，次で定義する:

$$d_{\ell_1}(x,y) := ||x-y||_1 = \textstyle\sum_{i\in[p]} |x_i - y_i|,$$
$$d_{\ell_2}(x,y) := ||x-y||_2 = \left\{\textstyle\sum_{i\in[p]} (x_i - y_i)^2\right\}^{1/2}.$$

文字列間の距離として，ハミング距離を次のように導入する．文字の不一致関数 $d_\Sigma : \Sigma^2 \to \mathbb{R}_{\geqslant 0}$ を，任意の $\forall a, b \in \Sigma$ に対し，$\delta_\Sigma(a,b) := \mathbb{1}[a \neq b] \in \{0,1\}$ と定める．次に，任意の $\ell \in \mathbb{N}$ と文字列 $X, Y \in \Sigma^\ell$ に対し，Σ^ℓ 上の**ハミング距離**（*Hamming distance*）を，

$$d_{HD}(X[1..\ell], Y[1..\ell]) = \sum_{i\in[\ell]} \delta_\Sigma(X[i], Y[i]) \in [0..\ell]$$

と定める．関数 $d_{HD}(X,Y)$ は Σ^ℓ 上の距離関数となる．

本稿で考察する問題．本稿では，ハミング距離 d_{HD} に関する $LCS(\mathcal{S})$ の最小多様性最大化問題（MMD(LCS)）と総和最小多様性最大化問題（MSD(LCS)）を考察する．問題の定義に関しては，定義 2（1.3節）を参照されたい．

3 最長共通部分列の DAG

本節では，m 個の入力文字列 $X_1, \ldots, X_m$ の LCS の集合をコンパクトに表現する多項式サイズの DAG の構成方法を与える．ここでは，この DAG を LCS グラフと呼ぶ．LCS グラフは，$LCS(\mathcal{S})$ を受理する非決定性有限オートマトンである．NFA についての詳細は，[7] 等の教科書を参照されたい．

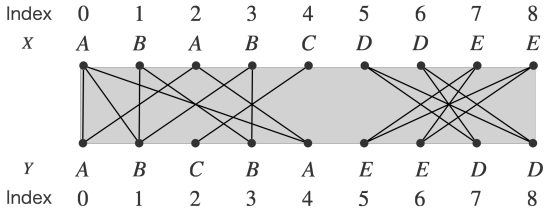

図 2: 例 1の入力文字列 $X = X_1$ と $Y = Y_1$ に対応する $m = 2$ 部グラフ．頂点の脇の数字は，文字列中の位置を表す．各線分は，同じ文字をもつ二つの位置 $\vec{i} = (i_1, i_2)$ を結ぶ辺を表す．

Σ 上の**非決定性有限オートマトン**（*nondeterministic finite automaton, NFA*）を，組 $N = (V, E, s, t)$ で表す．ここに，V は頂点集合であり，$E \subseteq V \times (\Sigma \cup \{\varepsilon\}) \times V$ は文字または空文字 ε をラベルにもつ有向辺の集合である．s と $t \in V$ は，それぞれ，初期状態と終了状態を表す．NFA の部分クラスとして，Σ 上の**空遷移をもたない非決定性有限オートマトン**（*NFA without ε-transitions*）は，非決定性有限オートマトン $N' = (V, E, s, t)$ で，集合 $E \subseteq V \times \Sigma \times V$ の有向辺が文字ラベルのみをもつ場合をいう．非決定性オートマトン N が受理する言語を，$\mathcal{L}(N) \subseteq \Sigma^*$ で表す．

3.1 文字列集合の LCS の計算

はじめに復習として，**動的計画法**（*DP*）を用いて，$LCS(\mathcal{S})$ 中の一つの LCS を計算する Irving [9] のアルゴリズムを説明する．これは，定数 $m \geqslant 2$ と入力文字列集合 $\mathcal{S} = \{X_1, \ldots, X_m\}$ に対して，動的計画法を用いて $\mathcal{S}$ の LCS の一つを，文字列の最大長さ n の多項式時間と領域で求める．

補題 1 (Irving [9])**.** $\mathcal{S}$ の LCS の一つ $Z \in LCS(\mathcal{S})$ は，$O(n^m)$ 時間と領域で計算可能である．

Irving [9] のアルゴリズムは，次の漸化式に基づいて，任意の $\vec{i} = (i_1, \ldots, i_m) \in [0..n]^m$ に対して，サイズ $\prod_{i=i}^{i} (|X_i|+1)$ の表 $L(i_1, \ldots, i_m) \in [0..n]$ を再帰的に計算する:

$$L(0, i_2, i_3, \ldots, i_m) = \cdots = L(i_1, i_2, \ldots, i_{m-1}, 0)$$
$$:= 0, \qquad \forall (i_1, \ldots, i_m) \in [0..n]^m,$$
$$L(i_1, \ldots, i_m)$$
$$:= \max \begin{cases} L(i_1 - 1, \ldots, i_m - 1) + 1, \\ \qquad \text{if } X_1[i_1] = \cdots = X_m[i_m], \\ \max\limits_{1 \leqslant h \leqslant m} L(i_1, \ldots, i_{h-1}, i_h - 1, i_{h+1}, \ldots, i_m), \\ \qquad \text{otherwise}. \end{cases}$$

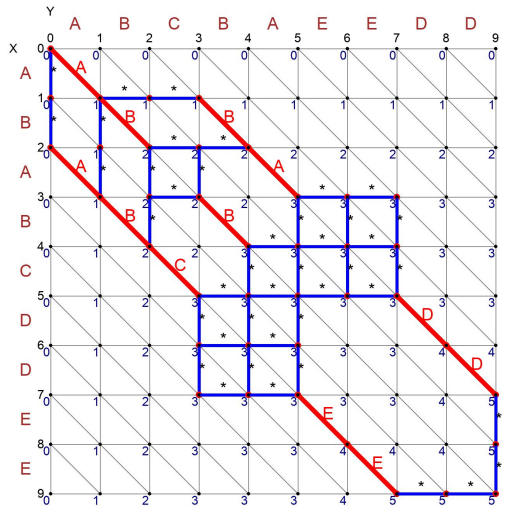

図 3: 例 1の入力文字列 $X = X_1$ と $Y = Y_1$ に対する空遷移つきの LCS グラフ N_{eps} の例

漸化式の構成から，任意の m-項組 $\vec{i} = (i_1, \ldots, i_m) \in \prod_{i=i}^{i}[0..|X_i|]$ に対して，値 $L(\vec{i})$ は正しく定義されることは明らかである．重要な点として，表のマス目 $L(i_1, \ldots, i_m)$ には，m 本の入力文字列の接頭辞 $X_1[1..i_1], \ldots, X_m[1..i_m]$ の LCS の長さが格納される．さらに，全ての m-項組のなす格子を，左上から右下に走査することで，表 L は $O(n^m)$ 時間と領域で計算できる．

上記の漸化式の直感的な説明を与える．図 2（$m = 2$ の場合）に示すように，入力文字列 $X_1, \ldots, X_m$ を上から順に水平な線分上に並べて，それぞれの文字列中の位置 $0, \ldots, |X_i|$ を第 i 番目の点集合（部）とし，隣り合った線分上で，同じ文字（色）をもつ点同志を辺で結ぶと，図のような m-部グラフ $\mathcal{B}_{\mathcal{S}} = (V_{\mathcal{S}} = \biguplus_{i=i}^{i} U_i, E_{\mathcal{S}})$ ができる．

ここに，$\mathcal{B}_{\mathcal{S}}$ に含まれる互いに非交差な線分の集合をトレース（*trace*）と呼ぶ．非交差性から，トレースの線分は添字の昇順で一意に整列できる．このとき，長さ $\ell \geqslant 0$ のトレース $T = (\vec{i_1}, \ldots, \vec{i_\ell}) \subseteq [n]^m$ は，同じ長さの $\mathcal{S}$ の共通部分列 $Z = (X_i[i_{i,1}], \ldots, X_i[i_{i,\ell}]) \in \Sigma^\ell$ と一対一に対応する．したがって，最大数の線分を含むトレースが LCS に対応する．そこで，各組 $(i_1, \ldots, i_m)$ に対応する接頭辞 $X_1[1..i_1], \ldots, X_m[1..i_m]$ が含む最大のトレースを再帰的に求める．

これらの観察をもとに，Irving [9] のアルゴリズムが表 L を正しく計算することがわかる [9]．以上の議論から，補題 1が示された．

3.2 空遷移を許した LCS グラフ

前節で説明した Irving のアルゴリズム [9] の計算過程から，$\mathcal{S}$ の LCS の集合 $LCS(\mathcal{S})$ を表す空遷移を許した NFA である N_{eps} を多項式時間で構築できる．この N_{eps} は，図 3に示すような格子型の形状をもつ．次の補題を示す．

補題 2. 任意の文字列集合 $\mathcal{S} = \{X_1, \ldots, X_m\}$ に対して，その言語 $\mathcal{L}(G)$ が $LCS(\mathcal{S})$ を表わす高々 n^m 個のノードと n^m 本の有向辺をもつ空遷移を許した非決定性有限オートマトン $N_{\text{eps}} = (V, E, s, t)$ が存在する．

証明. Σ 上の m 本の同じ長さ $n \geqslant 0$ の文字列の任意の集合 $\mathcal{S} = \{X_1, \ldots, X_m\}$ を考える．$\mathcal{S}$ の LCS の長さを $\ell := lcs(X_1, \ldots, X_m) \geqslant 0$ とおく．補題 1に基づいて，$\mathcal{S}$ の LCS グラフを，一意の入口 s と出口 t を持つ空遷移を許した NFA $N = (V, E, s, t)$ として，次のように構築する．

(1) 頂点集合 $V := \prod_{i=i}^{i}[|X_i|]$ を，全ての m 項組 $\vec{i} = (i_1, \ldots, i_m)$ からなる集合とする．

(2) ラベル付き有向辺の集合 $E \subseteq V \times (\Sigma \cup \{\varepsilon\}) \times V$ を次のように構築する．任意の m-項組 $\vec{i} = (i_1, \ldots, i_m), \vec{j} = (j_1, \ldots, j_m) \in V$ に対して，$e = (\vec{j}, c, \vec{i}) \in E$ であるのは，次の (a) または (b) が成立するとき，そのときだけとする:

(a) 次の条件が成立する: (a1) $\vec{j} = (i_1 - 1, \ldots, i_m - 1)$ かつ，(a2) $c = X_1[i_1] = \cdots = X_m[i_m]$，(a3) $L(i_1, \ldots, i_m) = L(i_1 - 1, \ldots, i_m - 1) + 1$.

(b) ある $h \in [m]$ に対して，次の条件が成立する: (b1) $\vec{j} = (i_1, \ldots, i_{h-1}, i_h - 1, i_{h+1}, \ldots, i_m)$，かつ，(b2) $c = \varepsilon$，(b3) $L(i_1, \ldots, i_m) = L(i_1, \ldots, i_{h-1}, i_h - 1, i_{h+1}, \ldots, i_m)$.

(3) 入り口を $\vec{s} = (0^{(1)}, \ldots, 0^{(m)})$ とおき，出口を $\vec{t} = (|X_1|, \ldots, |X_m|) \in V$ とおく．

上記で構成した空遷移を許した NFA は，所望の性質を満たすことが示せる．これより，補題 2が示された． □

例として，図 3に，長さ $n = 9$ の入力文字列 $X = X_1$ と $Y = Y_1$ （図 2）に対する空遷移つきの LCS グラフ N_{eps} を示す．図からは，このグラフが頂点集合 $V = [0..9]^2$ と水平，対角，垂直の 3 方向のタイプの有向辺をもつ格子状の形状をもつことがわかる．

3.3 空遷移をもたない LCS グラフ

本小節では，前節で示した補題 2を用いて，次の補題を示す．

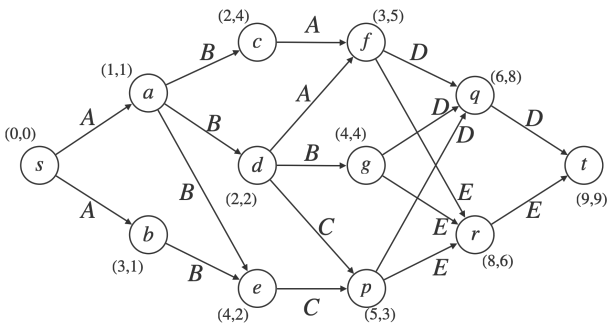

図 4: 例 1の入力文字列 $X_1 = ABABCDDEE$ と $Y_1 = ABCBAEEDD$ に対して，N_{eps} から空遷移除去で得られた LCS グラフ N_{noeps}．図において，円それぞれの中の英小文字 $a, b, c, \ldots$ は，頂点 ID を表し，その脇のラベル (i_1, i_2) は対応する添字の対を，辺それぞれの中央の英大文字 $A, B, \ldots$ はその文字ラベルを表す．

補題 3. 任意の文字列集合 $\mathcal{S} = \{X_1, \ldots, X_m\}$ に対して，N_{eps} から，次の性質を満たす空遷移をもたない非決定性有限オートマトン $N_{\mathrm{noeps}} = (V, E, s, t)$ を多項式時間で構築できる:

(1) 言語 $\mathcal{L}(G)$ が $LCS(\mathcal{S})$ を表わす．
(2) 高々 n^m 個のノードと n^{m+1} 本の有向辺をもつ
(3) さらに，頂点集合 V は，ℓ 個の互いに素な部分集合（階層と呼ぶ）$V_0 = \{s\}, V_1, \ldots, V_\ell = \{t\}$ に分割される．
(4) 全ての辺は，隣接した階層 V_{h-1} と V_h の間の頂点間を状態遷移する．

証明. N_{noeps} の構成には，よく知られた NFA の ε-遷移除去アルゴリズム [8] を用いればよい．詳細は省略する．N_{noeps} は全ての LCS をちょうど受理することから，初期状態 s から任意の状態（頂点）v へ到達するパスの文字ラベルの長さは同一である．これより，s を始点とする幅優先探索を用いて，所望の階層分割が得られる． □

任意の $h \in [\ell]$ について，任意の頂点 $v \in V_h$ のレベル（*level*）を $\mathrm{level}(v) = h$ と定める．入口 s から任意の頂点 $v \in V$ に至る有向パスの全体を $\mathsf{Path}_{N_{\mathrm{noeps}}}(v)$ と表す．図 4に，例として，入力文字列 $X = X_1$ と $Y = Y_1$ (図 2) に対する空遷移除去で得られた LCS グラフ N_{noeps} を示す．

4 定数サイズの多様な LCS 集合を求める多項式時間アルゴリズム

本節では，入力文字列数 $m \geqslant 1$ と解数 $k \geqslant 1$ が定数の場合に，MMD_k と MSD_k を入力サイズの多項式時間と領域で解く効率良いアルゴリズムを与える．

4.1 アルゴリズムの概要

前節の方法で，入力集合 $\mathcal{S}$ から構築された空遷移をもたない LCS グラフ $N_{\mathrm{noeps}} = (V, E, s, t)$ が与えられたと仮定する．ここに，LCS 長を $\ell := lcs(\mathcal{S})$ とし，N_{noeps} は $\ell + 1$ 個のレベルをもつとする．このとき，提案アルゴリズムは，次のように定められる仮想的な**マクロ DAG** $\mathcal{D} = (\mathcal{V}, \mathcal{E}, \vec{s}, \vec{t})$ 上で動的計画法を行う．

- マクロな頂点集合 $\mathcal{V} \subseteq V^k$ は，同じレベルに属する頂点の k-項組全体からなる．すなわち，$\forall \vec{p} = (p_1, \ldots, p_k) \in V^k, \vec{p} \in \mathcal{V} \iff \mathrm{level}(p_1) = \cdots = \mathrm{level}(p_k)$ が成立する．
- 任意のマクロ頂点の対 $\vec{p} = (p_1, \ldots, p_k), \vec{q} == (q_1, \ldots, q_k) \in \mathcal{V}$ と文字の k-項組 $\vec{c} = (c_1, \ldots, c_k) \in \Sigma^k$ に対して次が成立する: $\mathcal{D}$ がマクロ有向辺 $\vec{e} = (\vec{p}, \vec{c}, \vec{q})$ をもつ $\iff \forall i \in [k], (p_i, c_i, q_i) \in E$. マクロ頂点 $\vec{r}$ から出るマクロ辺の集合を $\mathcal{E}^+(\vec{r}) := \{ (\vec{p'}, \vec{c'}, \vec{q'}) \in \mathcal{E} \mid \vec{p} = \vec{r} \}$ で表す．
- マクロな入口 $\vec{s}$ と出口 $\vec{t} \in \mathcal{V}$ は，入口頂点と出口頂点の組 $\vec{s} = (s^{(1)}, \ldots, s^{(k)}), \vec{t} = (t^{(1)}, \ldots, t^{(k)})$ である．

マクロ DAG $\mathcal{D}$ の頂点数と辺数は，それぞれ，$|\mathcal{V}| \leqslant (n-1)^m + 2 = O(n^m)$ と $|\mathcal{E}| \leqslant |\mathcal{V}|^2 = O(n^{2m})$ であり，m が定数のとき，入力長の多項式サイズである．

Algorithm 1に動的計画法の手続きを示す．

(1) この手続きは，マクロ入り口 $\vec{s}$ から開始し，$\mathcal{D}$ 上の幅優先探索を用いて，各マクロ頂点 $\vec{p} \in \mathcal{V}$ を訪問する．
(2) 各マクロ頂点 $\vec{p}$ では，全ての $(\vec{s}, \vec{p})$-パスに対応する「打ち切り相互ハミング重み行列」（次の小節で説明する）と呼ばれるサイズ $(k \times k)$ の行列たちからなる集合 $\mathcal{W}[\vec{p}]$ を計算する．
(3) マクロ出口 $\vec{t}$ まで計算が完了したら，それがもつ打ち切り相互ハミング重み行列たちに関して，d に関する多様性制約 $D_d^\alpha(A) \geqslant \Delta$ を検査し（α は min または sum），判定を出力する．

4.2 打ち切り相互ハミング重み行列

マクロ DAG $\mathcal{D}$ のマクロ入口 $\vec{s}$ から任意のマクロ頂点 $\vec{p} = (p_1, \ldots, p_k)$ に至る，マクロな有向パス $\pi = (\vec{p}_0 = \vec{s}, \ldots, \vec{p}_\ell = \vec{t}) \in \mathcal{V}^\ell$ を考える．ここに，π の隣接マクロ頂点は $e_i = (p_{i-1}, c_i, p_i) \in E$ $(\forall j \in [h])$ で互いに接続する．この π を $(\vec{s}, \vec{p})$-マクロパスと呼び，それら全ての集合を $\mathsf{MPath}(\vec{p})$ で表す．

$\mathsf{MPath}(\vec{p})$ 中の各 $(\vec{s}, \vec{t})$-マクロパス $\pi = (\vec{p}_0, \ldots, \vec{p}_h) \in \mathcal{V}^\ell$ は，すべての成分 $\forall i \in [k]$ にわたって，LCS グラフ N_{noeps} が

含む k 本の対応する長さ h のパスの k-項組 $\mathsf{CPath}_k(\pi_h) = (P_1, \ldots, P_k) \in (V^h)^k$ を表す．これを，**成分パスの k 項組**（*component path k-tuple*）と呼ぶ（詳細な定義は省略）．

ここで，重み行列の族を導入する．$\mathcal{WM}_{k,\Delta}$ で，サイズが $k \times k$ で値域 $[0\,..\,\Delta]$ をもつ上三角行列 $W = (W_{i,j}) \in [0\,..\,\Delta]^{[k]\times[k]}$ の全体を表す．各メンバー $W \in \mathcal{WM}_{k,\Delta}$ を，**打ち切り相互重み行列**（*truncated weight matrix*）と呼ぶ．そのような行列の総数は，定数 k に対して，高々 $|\mathcal{WM}_{k,\Delta}| \leqslant (\Delta+1)^{\binom{k}{2}} = O(\Delta^{k^2})$ 個，つまり Δ の多項式個であることに注意されたい．次に $\mathcal{WM}_{k,\Delta}$ 上の定数と演算を導入しよう．$W_0 \in \mathcal{WM}_{k,\Delta}$ で，$\forall (i,j) \in [k]^2, W_0[i,j] = 0$ となる**空行列**を表す．任意の文字の k-項組 $\vec{c} = (c_1, \ldots, c_k) \in \Sigma^k$ に対して，二値行列 $C(\vec{c}) = (C_{i,j}) \in \mathcal{WM}_{k,\Delta}$ を，$i < j$ ならば $C_{i,j} := \delta_\Sigma(c_i, c_j) \in \{0,1\}$ で，それ以外なら $C_{i,j} = 0$ となる行列とする．相互重み行列 $U, W \in \mathcal{WM}_{k,\Delta}$ の加算を，通常の行列の要素ごとの加算 $U + W$ と定める．

以上の準備のもとに，$\vec{P} = (P_1, \ldots, P_k) \in (V^h)^k$ を長さ h のパスからなる k 項組とする．このとき，$\vec{P}$ の**打ち切り相互ハミング重み行列**（*truncated mutual Hamming weight matrix*）を，次で定める相互重み行列 $\mathsf{WMatrix}(\vec{P}) = (W_{i,j})$ とする:

$$W_{i,j} := \begin{cases} \min\{\,\Delta, d_{HD}(P_i, P_j)\,\} & \text{if } i < j, \\ 0, & \text{otherwise,} \end{cases}$$

ここに $i, j \in [k]$ である．以降，$\mathsf{WMatrix}(\vec{P}) \in \mathcal{WM}_{k,\Delta}$ を単に $\vec{P}$ の**重み行列**と呼ぶ．最後に，各マクロ頂点 $\vec{p} \in \mathcal{V}$ に対し，全ての $(\vec{s}, \vec{p})$-マクロパスの相互重み行列の集合

$$\mathcal{W}[\vec{p}] := \left\{ \mathsf{WMatrix}(\vec{P}) \;\middle|\; \begin{array}{l} \vec{P} = \mathsf{CPath}_k(\pi), \\ \pi \in \mathsf{MPath}(\vec{p}) \end{array} \right\} \in 2^{\mathcal{WM}_{k,\Delta}}$$

を割り当てる．

4.3 動的計画法アルゴリズム

本小節では，前の小節で定めたマクロ $\mathcal{D}$ の各マクロ頂点 $\vec{p} \in \mathcal{V}$ がもつ重み行列の集合 $\mathcal{W}[\vec{p}] \in 2^{\mathcal{WM}_{k,\Delta}}$ を計算する動的計画法のアルゴリズムを与える．$\mathcal{W}[\vec{p}]$ に関する以下の漸化式を考える:

- $h = 0$ のとき： マクロ入口 $\vec{s} = (s^{(1)}, \ldots, s^{(k)}) \in V_0$ に対して，$\mathcal{W}[\vec{s}] := \{W_0\}$ と定める．
- $h \geqslant 1$ のとき：レベル h の任意のマクロ頂点 $\vec{q} \in \mathcal{V}$ と，任意の重み行列 $W' \in \mathcal{W}_{k,\ell}$ に対して，次のように定める: $(a) W' \in \mathcal{W}[\vec{q}] \overset{\text{def}}{\iff} (b) \exists \vec{e} = (\vec{p}, \vec{c}, \vec{q}) \in \mathcal{E}, \exists \vec{p} \in \mathcal{V}, \exists W \in \mathcal{W}[\vec{p}], W' := W + C(\vec{c})$.

Algorithm 1: LCS 長 $\ell = lcs(\mathcal{S})$ をもつ $LCS(\mathcal{S})$ を表す空遷移を持たない LCS グラフ N_{noeps} から，相互ハミング重み行列の表 $\mathcal{W}$ と親ポインタの表 $\mathcal{P}$ の対を計算する動的計画法のアルゴリズム．

```
1  procedure ComputeTable(N = (V, E, s, t, ℓ))
     ▷V = ⋃_{h=0}^{ℓ} V_h
2  begin
3      W ← ∅; P ← ∅;
       ▷Base step for level h = 0
4      W[s] ← {W_0} ;        ▷初期状態 s, ゼロ行列 W_0
       ▷Induction step for level h ⩾ 1
5      for ∀h ∈ [1..ℓ] do
6          for ∀p ∈ (W.key ∩ (V_{h-1})^k) do
7              for ∀(p, c, q) ∈ E^+(p) do
8                  for ∀W ∈ W[p] do   ▷ |W[p]| ⩽ Δ^{k^2}
9                      W' ← W + C(c);
10                     if W[q] = ⊥ then
                           W[q] ← {W'};
11                     else if W' ∉ W[q] then
12                         W[q] ← W[q] ∪ {W'};
13                     P[(q, W')] ← (p, W) ;
                           ▷親を記録
14     return (W, P);
```

漸化式の正当性は，その構成から明らかである．よって，次の補題を得る．

補題 4 (漸化式の正当性)**.** $\mathcal{D}$ に含まれる任意のマクロ頂点 $\vec{p} = (p_1, \ldots, p_k) \in \mathcal{V}$ と任意の重み行列 $W \in \mathcal{WM}_{k,\Delta}$ に対して，以下の条件 (1)–(3) は同値である：

(1) $W \in \mathcal{W}[\vec{p}]$.
(2) $\exists \pi \in \mathsf{MPath}(\vec{p}), W = \mathsf{WMatrix}(\mathsf{CPath}(\pi))$.
(3) LCS グラフ N_{noeps} に含まれるある k 個のパスの組 $P_1 \in \mathsf{Path}(p_1), \ldots, P_k \in \mathsf{Path}(p_k)$ が存在して，$W = \mathsf{WMatrix}(P_1, \ldots, P_k)$ が成立する．

Algorithm 1に，上の $\mathcal{W}$ の漸化式と補題 4 に基づいて，$\mathcal{S}$ の LCS グラフ N_{noeps} から表 $\mathcal{W}$ を計算する動的計画法アルゴリズムを示す．補題 4から，次の補題が直ちに成立する．

補題 5. Algorithm 1は，$\mathcal{S}$ の LCS グラフ N_{noeps} と，正整数 k, Δ から，表 $\mathcal{W}$ を正しく計算する．

例 2. 図 5に，Algorithm 1の実行例を示す．これは，6 本

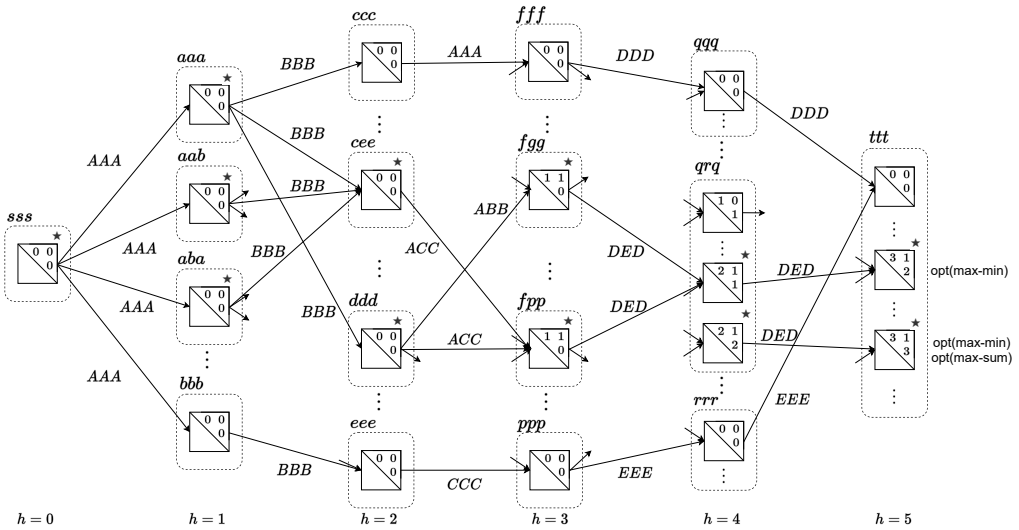

図 5: 例 1の入力文字列 $X = ABABCDDEE$ と $Y = ABCBAEEDD$ を与えた場合に，6 本の LCS から，$k = 3$ 本の多様な LCS 組をもとめる提案アルゴリズムの実行例．最小多様性スコア $D_d^{\min}$ を与えるマクロパスについて，それに含まれる重み行列の右上に ★ 印を付与している．マクロ頂点 ttt において，多様性スコア $D_d^{\min}$ と D_d^{sum} それぞれに関する最適解を示す重み行列に，ラベル opt(max-min) と opt(max-min) を付与している．

の LCS を含む LCS グラフ上で，$k = 3$ 本の多様な LCS からなる集合をもとめる．

Algorithm 1が表 $\mathcal{W}$ を計算した後で，MMD(LCS) 問題に関しては，マクロ出口 $\vec{t}$ を受け取り，次のコードを実行することで，条件 $D_d^{\min}(A) \geqslant \Delta$ を判定できる．

```
1 for ∀W ∈ W[t⃗] do
2     d_min ← min_{1⩽i<j⩽k} W[i,j] ;      ▷(*) 値の更新
3     if d_min > Δ then return d_min;
```

一方，上の手続きの 2 行目 (*) の値の更新をコード “ $d_{\min} \leftarrow \sum_{1\leqslant i<j\leqslant k} W[i,j]$ ” に代えることで，MSD(LCS) 問題の条件 $D_d^{\min}(A) \geqslant \Delta$ を判定できる．

補題 5から，次の定理が得られる．

定理 1. 任意の定数 $m, k \geqslant 1$ を仮定する．このとき，m 本の任意の入力文字列の集合 $\mathcal{S}$ に対して，MMD(LCS) 問題と MSD(LCS) 問題は，$O(k^2\Delta^{k^2}n^{2km})$ 時間と $O(\Delta^{k^2}n^{km})$ 領域で解ける．ここに，n は入力文字列の最大長であり，$\Delta = O(n)$ は多様性の下限値である．

証明. Algorithm 1の正当性は，漸化式の正当性から導かれる．LCS グラフの頂点集合 V について，補題 3より，$|V| \leqslant (n-1)^m + 2 = O(n^m)$ であり，頂点の k-項組全体の集合 $\mathcal{V}$ のサイズは高々 $|\mathcal{V}| \leqslant |V|^k$ である．

Algorithm 1において，実行全体をみると，はじめに 5–6 行目の for-loop と 8 行目の for-loop によって，各マクロ辺 $\vec{e}$ をちょうど一度ずつ試す．ここで，$|\mathcal{E}| \leqslant |\mathcal{V}|^2 \leqslant |V|^{2k}$ である．次に，11 行目では，親 $\vec{p}$ がもつ重み行列リスト $\mathcal{W}[\vec{p}]$ に含まれる行列全てに対して 12–16 行目の処理を行う．ここで，値域 $[0\,..\,\Delta]$ をもつサイズ $k \times k$ の重み行列の異なり数は $|\mathcal{WM}_{k,\Delta}| \leqslant \Delta^{k^2}$ である．よって，各リストの長さは高々 $|\mathcal{W}[\vec{p}]| \leqslant |\mathcal{WM}_{k,\Delta}| \leqslant \Delta^{k^2}$ であり，行列の要素数は k^2 である．

以上の議論より，アルゴリズムの総実行時間は，$O(|\mathcal{E}|) \times O(|\mathcal{WM}_{k,\Delta}|) \times O(k^2) = O(|\mathcal{V}|^k) \times O(\Delta^{k^2}) \times O(k^2) = O(k^2\Delta^{k^2}n^{2km})$ で与えられる．領域量は，頂点の k-項組の総数が $|V|^k = n^{km}$ 個であり，重み行列リスト $\mathcal{W}[\vec{p}]$ の長さが高々 Δ^{k^2} であることから，$O(\Delta^{k^2}n^{km})$ となり，定理が示された． □

定理 1から，m と k が定数のとき，MMD(LCS) 問題と MSD(LCS) 問題は入力サイズ $||\mathcal{S}||$ に対する多項式時間で解けることがわかる．

4.4 解の出力

上の問題は判定問題であったが，Algorithm 1を修正して，MMD と MSD の多様性を最適化する k 個の LCS 集合を同じ計算量で出力することができる．

補題 6. このとき，任意の入力文字列の集合 $\mathcal{S}$ と $\Delta \geqslant 0$ に対して，Algorithm 1 が真（多様解がある）と出力したとき，多様解の一つ A を定理 1と同じ時間と領域を用いて計算可能である．

証明. Algorithm 1が真と判定したとき，マクロ出口がもつ根拠となった重み行列の任意の一つを $W_{\mathrm{opt}} \in \mathcal{W}[\vec{t}]$ とおく．このとき，Algorithm 1の実行結果 $\mathcal{W}$ と $\mathcal{P}$ から存在が保証されるマクロ DAG $\mathcal{D}$ において，$\vec{t}$ からの深さ優先探索を模倣して，多様解 A を与えるマクロパスの任意の一本 π を見つけることができる．ただし，この際に，重み行列 W_{opt} を与えるマクロパスを同定するために，親ポインタの表 $\mathcal{P}$ を $\vec{t}$ から $\vec{s}$ へ向けて逆向きに辿ることで，正しく W_{opt} を与えるマクロパスを見つけられる．計算時間は，$\mathcal{D}$ の深さで抑えられるので，定理 1に示された時間と領域以内で実行され，補題が示された． □

5 結論

定数 m 個の入力文字列がもつ最長共通部分列集合に対して，k 個の LCS の組合せについて，ハミング距離に関する最小と最大の多様性を最大化する問題 MMD(LCS) と MSD(LCS) について考察した．結果として，k と m が定数である場合に，これらの問題が入力サイズの多項式時間で解けることを示した．今後の課題については，1.4節を参照されたい．

参考文献

[1] J. Baste, M. R. Fellows, L. Jaffke, T. Masaŕık, M. de Oliveira Oliveira, G. Philip, and F. A. Rosamond. Diversity in combinatorial optimization. *CoRR, abs/1903.07410*, 2019.

[2] A. Cevallos, F. Eisenbrand, and R. Zenklusen. An improved analysis of local search for max-sum diversification. *Mathematics of Operations Research*, 44(4):1494–1509, 2019.

[3] A. Conte, R. Grossi, G. Punzi, and T. Uno. A compact dag for storing and searching maximal common subsequences. In *ISAAC 2023*, pages 21:1–21:15, 2023.

[4] M. R. Fellows and F. Rosamond. The diverse x paradigm (open problems). In H. Fernau, P. Golovach, M.-F. Sagot, et al., editors, *Algorithmic enumeration (Dagstuhl Seminar 18421), Vol.8*, 2019.

[5] M. R. Garey and D. S. Johnson. Computers and intractability: A guide to the theory of np-completeness, 1979.

[6] T. Hanaka, Y. Kobayashi, K. Kurita, and Y. Otachi. Finding diverse trees, paths, and more. In *AAAI2021*, pages 3778–3786, 2021.

[7] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Longman, 2006.

[8] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[9] R. W. Irving. Two algorithms for the longest common subsequence of three (or more) strings. In *CPM 1992*, 1992.

[10] T. Uno, G. Punzi, Y. Shida, H. Arimura, and Y. Kobayashi. *Private communication, Seminar, Hokkaido University, Sapporo*, November 2023.