

Finding Diverse Strings and Longest Common Subsequences in a Graph

Luca Lombardo

Seminar for the course of Bioinformatics

Structure of the Presentation

- 1 Introduction to the problems
 - Diverse Strings Problems
 - Σ -DAG for LCSs
- 2 Exact Algorithms for Bounded Number of Diverse Strings
 - MAX-MIN DIVERSE STRING SET problem
 - MAX-SUM DIVERSE STRING SET problem
- 3 Approximation Algorithms for Unbounded Number of Diverse Strings
 - Approximation Algorithm for MAX-MIN DIVERSE STRING SET problem
- 4 FPT Algorithms for Bounded Number and Length of Diverse Strings
- 5 Complexity Results for Diverse String Problems

Longest Common Subsequence (LCS)

Definition

Given a set of m strings $S = \{S_1, S_2, \dots, S_m\}$, a **common subsequence** (CS) is a sequence that appears in all m strings. A **longest common subsequence** LCS is a common subsequence of maximum length. We denote the set of all LCSs of S as $LCS(S)$.

Goal

The goal is to find a diverse set of solutions to the LCS problem under the Hamming distance.

Definition

Given two strings $X, Y \in \Sigma^r$, the **Hamming distance** between X and Y , denoted with $d_H(X, Y)$, is the number of positions at which the corresponding symbols differ.

A simple example

Table: Longest common subsequences of two input strings $X = ABABCDDEE$ and $Y = ABCBAEEDD$. over $\Sigma = \{A, B, C, D, E\}$

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

Efficient methods for finding a diverse set of solutions

More formally, let's consider the following two diversity measures for a multiset $\mathcal{X} = \{X_1, X_2, \dots, X_K\} \subseteq \Sigma^r$ of solutions, allowing repetitions:

$$D_{d_H}^{\text{sum}}(\mathcal{X}) = \sum_{1 \leq i < j \leq K} d_H(X_i, X_j) \quad \text{MAX-SUM DIVERSITY} \quad (1)$$

$$D_{d_H}^{\text{min}}(\mathcal{X}) = \min_{i < j} d_H(X_i, X_j) \quad \text{MAX-MIN DIVERSITY} \quad (2)$$

Notation

A subset $\mathcal{X} \subseteq \Sigma^r$ is Δ -diverse w.r.t. $D_{d_H}^\tau$ if $D_{d_H}^\tau(\mathcal{X}) \geq \Delta$ for some $\Delta \geq 0$.

Where for $\tau \in \{\text{sum}, \text{min}\}$, $D_{d_H}^\tau$ denotes one of the two diversity measures.

Two problems

Problem 1: DIVERSE LCSs WITH DIVERSITY MEASURE $D_{d_H}^\tau$

Input: A set $S = \{S_1, S_2, \dots, S_m\}$ of $m \geq 2$ strings over Σ , an integer $K \geq 1$ and $\Delta \geq 0$.

Question: Is there some set $\mathcal{X} \subseteq LCS(S)$ such that $|\mathcal{X}| = K$ and $D_{d_H}^\tau(\mathcal{X}) \geq \Delta$?

Problem 2: DIVERSE STRING SET

Input: $K, r, \Delta \in \mathbb{Z}$ and a Σ -DAG G for a set $L(G) \subseteq \Sigma^r$ of strings.

Question: Decide if there exists some subset $\mathcal{X} \subseteq L(G)$ such that $|\mathcal{X}| = K$ and $D_{d_H}^\tau(\mathcal{X}) \geq \Delta$.

Σ -Labeled Directed Acyclic Graphs (Σ -DAGs)

Definitions

- **Alphabet (Σ):** Set of symbols.
- **String Set (Language):** $L = \{X_1, X_2, \dots, X_n\} \subseteq \Sigma^*$, with:
 - ▶ **Total Length:** $||L|| = \sum_{X \in L} |X|$
 - ▶ **Max Length:** $\text{maxLen}(L) = \max_{X \in L} |X|$
- **r-String:** Any string X where $|X| = r$

Σ -DAG Structure

A graph $G = (V, E, s, t)$ with:

- V : vertices, E : labeled edges (v, c, w) with $c \in \Sigma$
- **Source** s and **Sink** t such that paths exist from s to all vertices
- **Size:** $\text{size}(G)$, the number of its labeled edges

Σ -Labeled Directed Acyclic Graphs (Σ -DAGs)

Any path $P = (e_1, e_2, \dots, e_n)$ of outgoing edges *spells out* a string $\text{str}(P) = c_1 c_2 \dots c_n \in \Sigma^n$ where c_i is the label of edge e_i .

Language Representation

A Σ -DAG represents $L(G) \subset \Sigma^*$: all strings spelled from paths $s \rightarrow t$.
Equivalent to an NFA over Σ with initial s , final t , and no ϵ -edges.

Remarks

- For any set L of strings, a Σ -DAG G exists s.t. $L(G) = L$ and $\text{size}(G) \leq ||L||$. Construction time: $O(||L|| \log |\Sigma|)$.
- If G represents a set L of r -strings ($L \subseteq \Sigma^r, r \geq 0$), all paths $s \rightarrow v$ have the same length $d \leq r$.

Σ -DAGs for LCSs and Diverse String Sets

Lemma (Σ -DAG for LCSs)

For any constant $m \geq 1$ and set $S = \{S_1, \dots, S_m\} \subseteq \Sigma^*$ of m strings, there exists a Σ -DAG G of polynomial size in $\ell := \maxlen(S)$ such that $L(G) = \text{LCS}(S)$ and can be computed in polynomial time /

Consequence of the Lemma

- **If** MAX-MIN (or MAX-SUM) DIVERSE STRING SET solvable in $f(M, K, r, \Delta)$,
- **Then** MAX-MIN (or MAX-SUM) DIVERSE LCSs on $S \subseteq \Sigma^r$ solvable in $O(|\Sigma| \cdot \ell^m + f(\ell^m, K, r, \Delta))$ time.

where $\ell = \maxlen(S)$

Exact Algorithms for Bounded Number of Diverse Strings

Algorithm for MAX-MIN DIVERSE STRING SET problem

Dynamic Programming Approach

Pattern of Path Tuple: For each $d \leq r$ and K -tuple of length- d paths $P = (P_1, \dots, P_K)$, define pattern $\text{Pattern}(P) = (\mathbf{w}, \mathbf{Z})$, where:

- $\mathbf{w} = (w_1, \dots, w_K)$: K -tuple of vertices representing endpoints of P_i paths.
- $\mathbf{Z} = (Z_{i,j})$: Upper triangular matrix of Hamming distances, where $Z_{i,j} = \min\{\Delta, d_H(\text{str}(P_i), \text{str}(P_j))\}$.

DP Table of Weights

$$\text{Weights} : V^K \times (\Delta \cup \{0\})^{K \times K} \rightarrow \{0, 1\} \quad (3)$$

Boolean matrix where $\text{Weights}(\mathbf{w}, \mathbf{Z}) = 1$ iff (\mathbf{w}, \mathbf{Z}) matches pattern $\text{Pattern}(P)$ for some K -tuple of paths of length d from s to \mathbf{w} .

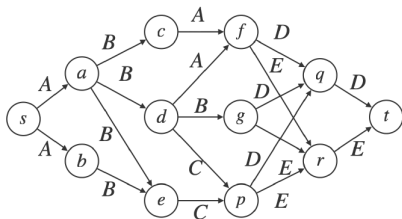
Algorithm for MAX-MIN DIVERSE STRING SET problem

Algorithm 1

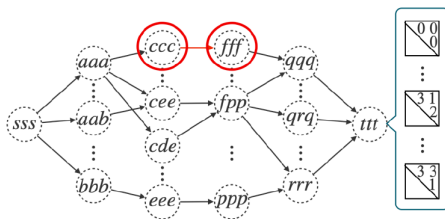
```
1: Set  $\text{Weights}(\mathbf{s}, Z) = 0$  for all  $Z \in (\Delta \cup \{0\})^{K \times K}$  and  $\text{Weights}(\mathbf{s}, \mathbf{0}) \leftarrow 1$ 
2: for  $d \leftarrow 1, \dots, r$  do
3:   for  $\mathbf{v} \leftarrow (v_1, \dots, v_K) \in (V_d)^K$  do
4:     for  $(v_1, c_1, w_1) \in E^+(v_1), \dots, (v_K, c_K, w_K) \in E^+(v_K)$  do
5:       Set  $\mathbf{w} = (w_1, \dots, w_K)$ 
6:       for  $U \in (\Delta \cup \{0\})^{K \times K}$  such that  $\text{Weights}(\mathbf{v}, U) = 1$  do
7:         Set  $Z = (Z_{i,j})_{i < j}$  with  $Z_{i,j} \leftarrow \min\{\Delta, U_{i,j} + \mathbb{1}\{c_i \neq c_j\}\}$   $\forall i, j \in K$ 
8:         Set  $\text{Weights}(\mathbf{w}, Z) \leftarrow 1$  ▷ Update
9:       end for
10:    end for
11:  end for
12: end for
13: Answer Yes if  $\text{Weights}(\mathbf{t}, Z) = 1$  and  $D_{d_H}^{\min}(Z) \geq \Delta$  for some  $Z$ , else No
```

For any $K \geq 1$ and $\Delta \geq 0$, it solves the MAX-MIN DIVERSE STRING SET problem in $O(\Delta^{K^2} K^2 M^K (\log |V| + \log \Delta))$ time and space when an input string set L is represented by a Σ -DAG G with $\text{size}(G) = M$

Example Run of the Algorithm



(a) An input Σ -DAG G_1 for $LCS(X_1, Y_1)$



(b) Example run of Algorithm 1 on G_1

Figure: (a) An input Σ -DAG G_1 over $\Sigma = \{A, B, C, D, E\}$ for the set of all longest common subsequences of two strings $X_1 = ABABCDDEE$ and $Y_1 = ABCBAEEDD$ and (b) an example run of Algorithm 1 based on dynamic programming with $K = 3$ on an input G_1

Algorithm for MAX-SUM DIVERSE STRING SET problem

Modify the Max-Sum Diverse String Set problem

Instead of the entire $K \times K$ weight matrix Z , only the sum $z = \sum_{i < j} d_H(\text{str}(P_i), \text{str}(P_j))$ is needed for computing Max-Sum diversity.

New DP Table Weights

For $\mathbf{w} = (w_1, \dots, w_K)$ of depth $0 \leq d \leq r$ and integer $0 \leq z \leq rK$, define:

$$\text{Weights}(\mathbf{w}, z) = 1$$

if and only if there exists a K -tuple of length- d prefix paths (P_1, \dots, P_K) from s to w_1, \dots, w_K with sum of pairwise Hamming distances z .

Algorithm for MAX-SUM DIVERSE STRING SET problem

Algorithm 2

```
1: Set  $\text{Weights}(\mathbf{s}, Z) = 0$  for all  $Z \in (\Delta \cup \{0\})^{K \times K}$  and  $\text{Weights}(\mathbf{s}, \mathbf{0}) \leftarrow 1$ 
2: for  $d \leftarrow 1, \dots, r$  do
3:   for  $\mathbf{v} \leftarrow (v_1, \dots, v_K) \in (V_d)^K$  do
4:     for  $(v_1, c_1, w_1) \in E^+(v_1), \dots, (v_K, c_K, w_K) \in E^+(v_K)$  do
5:       Set  $\mathbf{w} = (w_1, \dots, w_K)$ 
6:       for  $u \leftarrow (0, \dots, rK)$  such that  $\text{Weights}(\mathbf{v}, U) = 1$  do
7:         Set  $Z = (Z_{i,j})_{i < j}$  with  $Z_{i,j} \leftarrow \min\{\Delta, u + \sum_{i < j} \mathbb{1}\{c_i \neq c_j\}\}$ 
8:         Set  $\text{Weights}(\mathbf{w}, Z) \leftarrow 1$  ▷ Update
9:       end for
10:    end for
11:  end for
12: end for
13: Answer Yes if  $\text{Weights}(\mathbf{t}, Z) = 1$  and  $D_{d_H}^{\min}(Z) \geq \Delta$  for some  $Z$ , else No
```

For any $K \geq 1$, 15 solves the MAX-SUM DIVERSE STRING SET under Hamming Distance in $O(\Delta K^2 M^K (\log |V| + \log \Delta))$ time and space, where M is the size of the input Σ -DAG G

Approximation Algorithms for Unbounded Number of Diverse Strings

MAX-SUM DIVERSE STRING SET problem

Use a local search algorithm for computing approximate solutions $\mathcal{X} \subseteq \mathcal{L}$ with $|\mathcal{X}| = K$ on a finite metric space (\mathcal{L}, d) , where $d : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}_{\geq 0}$.

Algorithm 3 LocalSearch(\mathcal{L}, K, d)

```
1:  $\mathcal{X} \leftarrow$  arbitrary set of  $K$  solutions in  $\mathcal{L}$ 
2: for  $i \leftarrow 1, \dots, \lceil \frac{K(K-1)}{K+1} \log \frac{(K+2)(K-1)^2}{4} \rceil$  do
3:   for  $X \in \mathcal{X}$  s.t.  $\mathcal{L} \setminus \{X\} \neq \emptyset$  do
4:      $Y \leftarrow \operatorname{argmax}_{Y \in \mathcal{L} \setminus \{X\}} \sum_{X' \in \mathcal{X} \setminus \{X\}} d(X', Y)$ 
5:      $\mathcal{X} \leftarrow \mathcal{X} \setminus \{X\} \cup \{Y\}$ 
6:   end for
7: end for
8: return  $\mathcal{X}$ 
```

Theorem

When the distance d is a semi-metric of negative type over \mathcal{X} , then LOCALSEARCH has improved approximation ratio $(1 - \frac{2}{K})$ for any $K \geq 2$. The Hamming distance d_H over the set of r -strings is a semi-metric of negative type.

MAX-SUM FARTHEST r-STRING problem

How do we solve efficiently the subproblem at line 4 of the LOCALSEARCH algorithm?

$$Y \leftarrow \operatorname{argmax}_{Y \in \mathcal{L} \setminus \{X\}} \sum_{X' \in \mathcal{X} \setminus \{X\}} d(X', Y)$$

The following algorithm can be modified to compute such Y by reordering the parent pair (v, y) of each (w, z) and then tracking back

Algorithm 4 Decisional MAX-SUM FARTHEST r-STRING

- 1: Set $\text{Weights}(s, z) := 0$ for all $z \in [\Delta]_+$, and $\text{Weights}(s, 0) := 1$
 - 2: **for** $d := 1, \dots, r$ **do**
 - 3: **for** $0 \leq u \leq \Delta$ such that $\text{Weights}(v, u) := 1$ **do**
 - 4: Set $\text{Weights}(w, z) := 1$ for $z := u + \sum_{i \in [K]} \mathbb{1}\{c \neq X_i[d]\}$ ▷ Update
 - 5: **end for**
 - 6: **end for**
 - 7: Answer YES if $\text{Weights}(t, \Delta) = 1$, and NO otherwise ▷ Decide
-

MAX-SUM FARTHEST r -STRING problem

Lemma (Max-Sum Farthest r -String)

For any $K \geq 1$ and $\Delta \geq 0$, ALGORITHM 3 computes the farthest r -string $Y \in L(G)$ that maximizes $D_{d_H}^{sum}(Y \cup \mathcal{X})$ over all r -strings in $L(G)$ in $O(K\Delta M)$ time and space, where M is the size of the input Σ -DAG G .

Theorem (Polynomial Time Approximation Scheme for unbounded K)

When K is part of an input, MAX-SUM DIVERSE STRING SET problem on a Σ -DAG admits a PTAS

The corresponding result for the MAX-SUM DIVERSE LCSs follows immediately.

Fixed-Parameter Tractable (FPT) Algorithms for Bounded Number and Length of Diverse Strings

FPT Algorithms for MAX-MIN and MAX-SUM DIVERSE STRING SET problems

Definition (Fixed-Parameter Tractable (FPT) Algorithm)

A problem parametrized with κ is said to be *fixed-parameter tractable* (FPT) if there exists an algorithm for the problem running on an input x in time $f(\kappa(x)) \cdot |x|^c$, where f is a computable function and $c > 0$ is a constant.

Proposed FPT Algorithm

We use the *color-coding* technique with dynamic programming to solve these problems efficiently. Assign a random color to the edges of the Σ -DAG G , creating a colored graph called C-DAG.

Computation of Reduced C-DAG

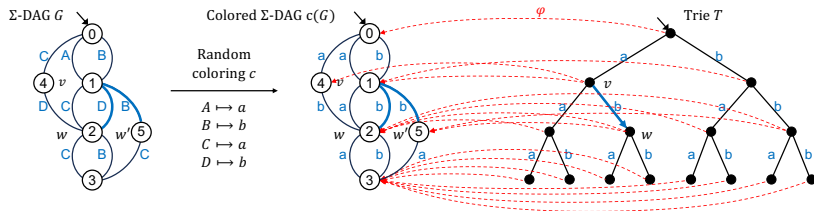


Figure: Computation of reduced C-DAG H from an input Σ -DAG G over alphabet $\Sigma = \{A, B, C, D\}$, which shows G (left), a random coloring c on $C = \{a, b\}$, a colored C-DAG $c(G)$ (middle), and a reduced C-DAG H in the form of trie T (right)

An analogous FPT algorithm can be designed for the MAX-SUM DIVERSE STRING SET problem with a small modification in the research phase, that, in this case, is $O(\Delta K^2 M^K)$

Complexity Results for Diverse String Problems

Negative Results

- NP-hard for unbounded K (MAX-MIN, MAX-SUM) in Σ -graphs for r -strings, $r \geq 3$.
- W[1]-hard parameterized with K for MAX-MIN and MAX-SUM in Σ -DAGs.

Reduction to Diverse LCSs

MAX-MIN and MAX-SUM problems are FPT-reducible to DIVERSE LCSs for $m = 2$ strings.

Corollaries

NP-hard and W[1]-hard results extend to DIVERSE LCSs for two r -strings.

Conclusion

- **Polynomial-Time Solutions:** When K is bounded, both the MAX-SUM and MAX-MIN versions of DIVERSE STRING SET and DIVERSE LCSS can be solved in polynomial time using dynamic programming (DP).
- **PTAS for Input-Based K :** For input-dependent K , the MAX-SUM versions of both DIVERSE STRING SET and DIVERSE LCSS admit a PTAS using local search due to the Hamming distance being a metric of negative type.
- **Fixed-Parameter Tractability (FPT):** Both versions are FPT when parameterized by K and r , combining the color coding technique and DP.
- **NP-Hardness for Constant $r \geq 3$:** When K is part of the input, both the MAX-SUM and MAX-MIN versions are NP-hard for any constant $r \geq 3$.
- **W[1]-Hard for Parameterized K :** Parameterized by K , both versions are W[1]-hard.