The NP-hardness and APX-hardness of the Two-Dimensional Largest Common Substructure Problems *

Syuan-Zong Chiou[a], Chang-Biau Yang[a†] and Yung-Hsing Peng[b]

[a]Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan

[b]Innovative DigiTech-Enabled Applications & Services Institute
Institute for Information Industry, Kaohsiung, Taiwan

Abstract

The similarity of one-dimensional data is usually measured by the longest common subsequence *(LCS) algorithms. However, these algorithms cannot be directly extended to solve the two or higher dimensional data. The* two-dimensional largest common substructure *(TLCS) problem was therefore proposed to compute the similarity of two-dimensional data. In 2016, Chan et al. [6] defined eight different versions of the TLCS problem, and four of them were shown to be valid for pattern matching, while the other four are invalid. In addition, Chan et al. showed that two versions of them are $\mathcal{NP}$-hard, and left a conjecture that the other two are also $\mathcal{NP}$-hard. In this paper, we prove that the remaining two versions of the TLCS problem are $\mathcal{NP}$-hard, showing the correctness of Chan's conjecture. Moreover, we prove that the four valid versions are all $\mathcal{APX}$-hard.*

1 Introduction

The *longest common subsequence* (LCS) problem and its variants have been extensively studied in several decades [3, 12–15, 20]. However, these LCS algorithms cannot be applied directly in two or higher dimensional data (such as picture images). Therefore, finding the similarity of these data by LCS-like definitions becomes another problem of interest.

In 1987, Chang *et al.* [8] represented a two-dimensional picture by 2D strings. In 1989, Chang *et al.* [7] proposed the 2D-string-LCS algorithm to retrieve the largest similar subpicture in an image database. In 1992, Lee and Hsu [18] proposed another similarity retrieval definition with 2D C-strings. In 2000, Guan *et al.* [11] proved that the problems of finding the largest similar subpicture with relations type-0 and type-1 of 2D strings are $\mathcal{NP}$-hard.

With another problem developing way, some researchers [2, 5] directly study the relationship of two matrices. They seemed not to know the above research progress on the 2D strings. In 1998, Baeza-Yates [5] presented a method for computing the edit distance of two matrices. And in 2008, Amir *et al.* [2] gave another similarity definition of two matrices, which is the *two-dimensional largest common substructure* (TLCS) problem. The matching rules of the TLCS problem given by Amir *et al.* are too strict to represent the matrix similarity. In fact, Amir's definition of the TLCS problem is exactly the same as the type-1 relation in 2D strings [8, 11]. However, Amir *et al.* had no idea about the study of 2D strings.

In 2016, Chan *et al.* [6] extended the definition for the TLCS to more general versions. They presented eight different versions in total, and four of them were proved to be valid for pattern matching. Two of the four valid versions have been proved to be $\mathcal{NP}$-hard. In this paper, we prove the $\mathcal{NP}$-hardness of the remaining two versions and the $\mathcal{APX}$-hardness of all four valid versions.

This paper is organized as follows. In Section 2, we give the preliminaries for our proof. In Sec-

*This research work was partially supported by the Ministry of Science and Technology of Taiwan under contract MOST 104-2221-E-110-018-MY3. This research was also partially supported by the "Online and Offline Integrated Smart Commerce Platform (4/4)" of the Institute for Information Industry, which is subsidized by the Ministry of Economy Affairs of Taiwan.

†Corresponding author. E-mail: cbyang@cse.nsysu.edu.tw (Chang-Biau Yang).

tion 3, we prove the hardness for the valid versions of TLCS. Finally, we give some future works and discussions in Section 4.

2 Preliminaries

Suppose that A is a matrix of size $r_A \times c_A$. We use $a_{i,j}$ to denote the entry at the ith row and the jth column of matrix A. Similarly, B is a matrix of size $r_B \times c_B$.

2.1 The Two-dimensional Largest Common Substructure Problem

In 2016, Chan *et al.* [6] proposed eight matching rules for *two-dimensional largest common substructure* (TLCS) problem, and four of them were shown to be valid. A *common substructure* U of two matrices A and B is defined as $U = \{(i,j,p,q) | a_{i,j} = b_{p,q}$, and every two elements in U obey the matching rules (given later)$\}$. A common substructure U is a *largest common substructure* if $|U|$ is maximized.

Definition 1. *[6] Two-dimensional largest common substructure (TLCS) problem*
Input: *Matrix A and matrix B.*
Output: *The largest common substructure U.*

Chan *et al.* partitioned the definitions into three parts as $P(corner, operator, side)$. Figure 1 shows examples for a corner and a side. The gray blocks in matrix B are the valid positions of β. The formal definitions for stipulating the matching rules are given as follows.

Definition 2. *[6] Logical operator for corner*

- And (N): *Both row and column relationships are satisfied.*
- Or (O): *The row relationship or the column relationship is satisfied.*

Definition 3. *[6] Corner: Let two elements a_{i_1,j_1} and a_{i_2,j_2} be in matrix A, where $i_1 \neq i_2$ and $j_1 \neq j_2$. And let two elements b_{p_1,q_1} and b_{p_2,q_2} be in matrix B. A pair of elements $(i_1, j_1, p_1, q_1), (i_2, j_2, p_2, q_2) \in U$ obeys one of the following two properties L and E.*

- Less than (L):
  - row relationship: $p_1 < p_2$ *when* $i_1 < i_2$.
  - column relationship: $q_1 < q_2$ *when* $j_1 < j_2$.

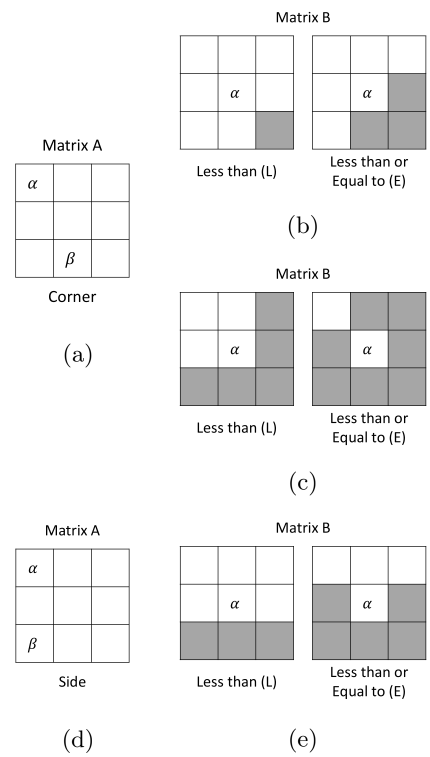


Figure 1: Examples for illustrating the corner and side definitions, where the gray blocks represent the allowed area in matrix B. (a) The corner relation of $a_{i_1,j_1} = \alpha$ and $a_{i_2,j_2} = \beta$. (b) The allowed positions of $b_{p_2,q_2} = \beta$ for the corner relation with logical operator N. (c) The allowed positions of $b_{p_2,q_2} = \beta$ for the corner relation with logical operator O. (d) The side relation of $a_{i_1,j_1} = \alpha$ and $a_{i_2,j_2} = \beta$. (e) The allowed positions of $b_{p_2,q_2} = \beta$ for the side relation.

Table 1: Summary of the various versions of the TLCS problem. Here, it is assumed that (i_1, j_1) and (i_2, j_2) are in matrix A. (i_1, i_2) indicates the relation for (p_1, p_2) in B, while (j_1, j_2) indicates the relation for (q_1, q_2) in B. [6]

Problem	Operator	Corner				Side				Symmetric
		$i_1 < i_2$	$j_1 < j_2$	$i_1 < i_2$	$j_1 > j_2$	$i_1 < i_2$	$j_1 = j_2$	$i_1 = i_2$	$j_1 < j_2$	
LNL	and	$<$	$<$	$<$	$>$	$<$			$<$	N
LNE		$<$	$<$	$<$	$>$	$\leq$			$\leq$	N
ENL		$\leq$	$\leq$	$\leq$	$\geq$	$<$			$<$	Y
ENE		$\leq$	$\leq$	$\leq$	$\geq$	$\leq$			$\leq$	Y
LOL	or	$<$	$<$	$<$	$>$	$<$			$<$	Y
LOE		$<$	$<$	$<$	$>$	$\leq$			$\leq$	Y
EOL		$\leq$	$\leq$	$\leq$	$\geq$	$<$			$<$	N
EOE		$\leq$	$\leq$	$\leq$	$\geq$	$\leq$			$\leq$	N

- Less than or equal to (E):
  - row relationship: $p_1 \leq p_2$ *when* $i_1 < i_2$.
  - column relationship: $q_1 \leq q_2$ *when* $j_1 < j_2$.

Definition 4. *[6] Side: Let two elements* a_{i_1,j_1} *and* a_{i_2,j_2} *be in matrix* A *where* $i_1 = i_2$ *or* $j_1 = j_2$. *And let two elements* b_{p_1,q_1} *and* b_{p_2,q_2} *be in matrix B. A pair of elements* $(i_1, j_1, p_1, q_1), (i_2, j_2, p_2, q_2) \in U$ *obeys one of the two following properties L and E.*

- Less than (L):
  - row relationship: $p_1 < p_2$ *when* $i_1 < i_2$ *and* $j_1 = j_2$.
  - column relationship: $q_1 < q_2$ *when* $i_1 = i_2$ *and* $j_1 < j_2$.
- Less than or equal to (E):
  - row relationship: $p_1 \leq p_2$ *when* $i_1 < i_2$ *and* $j_1 = j_2$.
  - column relationship: $q_1 \leq q_2$ *when* $i_1 = i_2$ *and* $j_1 < j_2$.

Based on the above definitions, $P(corner, operator, side)$ leads to eight different definitions. Note that *operator* is operated on *corner*. Among them, Chan *et al.* showed that $P(ENE)$, $P(ENL)$, $P(LOL)$ and $P(LOE)$ are valid definitions. That is, an optimal solution of two matrices A and B should also be an optimal solution of matrices B and A. Chan *et al.* called those definitions as *symmetric*. Table 1 shows the TLCS problems with various definitions. Note that the other four definitions are invalid.

Chan *et al.* proved that $P(ENL)$ and $P(ENE)$ are $\mathcal{NP}$-hard by reducing from the k-clique problem. Here we omit the detailed proof of Chan *et al.*

In the following, we use CU_{ENL}, CU_{ENE}, CU_{LOL} and CU_{LOE} to represent the solutions of the corresponding problems, respectively.

2.2 Approximability Classes

Definition 5. *[1, 9, 10] NP optimization (NPO) problem*
An NPO problem F is represented by a 4-tuple $(\oint_F, Sol_F, oj_F, opt_F)$, *explained as follows.*

(i) $\oint_F$ *is the set of the instances of* F, *and* $\oint_F$ *is recognizable in polynomial time.*

(ii) $Sol_F(I)$ *is the set of the feasible solutions with instance* I. *And there exists a polynomial function* p *such that* $\forall sol \in Sol_F(I), |sol| \leq p(|I|)$.

(iii) $oj_F(I, sol)$ *is called the* objective function, *for each instance* I *and each feasible solution* $sol \in Sol_F(I)$.

(iv) $opt_F \in \{max, min\}$ *tells if* F *is a maximization or a minimization problem.*

Solving an NPO problem F with a given instance I means finding a feasible solution $sol \in Sol_F(I)$ which maximizes $oj_F(I, sol)$ over all feasible solutions sol if $opt_F = max$ or minimizes $oj_F(I, sol)$ if $opt_F = min$. Taking the TLCS problem as an example, let $\oint_{TLCS}$ be the set of all possible pairs of matrices. Let I be an instance of matrices A and B, with $|A| = r_A \times c_A$ and $|B| = r_B \times c_B$. For a solution sol of instance I, there is a polynomial function $p(x) = x$, such that $|sol| \leq r_A \times c_A$ and $|sol| \leq r_B \times c_B$. We have $oj_F(I, sol) = |sol|$ for instance I and any solution $sol \in Sol(I)$. Also, we have $opt_{TLCS} = max$, because we want to find the common substructure with maximal size.

Given $I \in \oint_F$, we denote $opt_F(I)$ as the optimal solution set for I. The approximation ratio is defined as $Rat_F = \max\{\frac{|opt_F(I)|}{oj_F(I,sol)}, \frac{oj_F(I,sol)}{|opt_F(I)|}\}$. It is clear that the ratio is always greater than or equal to 1. The closer to 1 the approximation ratio is, the better performance the algorithm has. In the following, we briefly explain the concept of *APX* and *L-reduction*, which will be used for our proof in Sections 3.1 and 3.3.

Definition 6. *[19] Approximability (APX) problem*
An NPO problem F is in APX if F can be approximated within a constant c, that is , there exists a polynomial-time algorithm D such that for all instances $I \in \oint_F$, $D(I) \in Sol_F$ *and* $Rat_F(D(I)) \leq c$, *where* $Rat_F(D(I))$ *is the approximation ratio of* F.

L-reduction was proposed by Papadimitriou and Yannakakis [19] in 1991. In various proofs of APX problems, L-reduction is the easiest and the most restrictive one.

Definition 7. *[19] L-reduction*
Given two NPO problems F and G, and a polynomial-time transformation $f : F \to G$, f is an L-reduction from F to G if there exist positive constants δ and μ such that the following conditions hold for every instance I of F.

(i) $|opt_G(f(I))| \leq \delta \cdot |opt_F(I)|$,

(ii) For every feasible solution sol of $f(I)$ *with objective value* $oj_G(f(I), sol) = oj_2$, *we can find a solution, in polynomial time,* sol' *of* I *with* $oj_F(I, sol') = oj_1$ *such that* $|opt_F(I) - oj_1| \leq \mu|opt_G(f(I)) - oj_2|$.

2.3 The Maximum 3-Satisfiability Problem

In 1999, Ausiello [4] proved that the *maximum 3-satisifiability* (MAX 3SAT-B) problem with $B \geq 3$ is $\mathcal{APX}$-complete. We give the formal definition of MAX 3SAT-3 as follows.

Definition 8. *[4] Maximum 3-satisifiability bounded by 3 (MAX 3SAT-3) problem*
Input: *A set of variables X, and a set of disjunctive clauses C over the variables X, where each clause consists of at most three variables, and each variable occurs in at most $B = 3$ clauses.*
Output: *Truth assignment of X for the maximal number of satisfied clauses.*

For example, suppose that there is a set $X = \{x_1, x_2, x_3\}$ and $C = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$, the number of each variable occurrences in C is bounded by 3. C is satisfied by $\{x_1, \overline{x_2}, \overline{x_3}\}$, and the number of satisfied clauses is 3.

2.4 The Maximum Bounded 3-Dimensional Matching Problem

In 1991, Kann [16] defined the *maximum bounded 3-dimensional matching* (MAX 3DM-B) problem, which has the constraint that each symbol appears at most B times in input. He also proved that the MAX 3DM-B problem is $\mathcal{MAX}$ $\mathcal{SNP}$-complete with $B \geq 3$, which is also $\mathcal{APX}$-hard, since $\mathcal{MAX}$ $\mathcal{SNP}$-complete $\subseteq$ $\mathcal{APX}$-hard. We give the formal definition of MAX 3DM-3 as follows.

Definition 9. *[16] Maximum 3-dimensional matching bounded by 3 (MAX 3DM-3) problem*
Input: *A set $M \subseteq X \times Y \times Z$, where X, Y and Z are disjoint finite sets. Each element of X, Y and Z occurs in M at most $B = 3$ times.*
Output: *The maximal number of 3-dimensional matching.*

For example, suppose that there is a set $M = \{(x_1, y_1, z_1), (x_1, y_3, z_3), (x_2, y_3, z_2), (x_3, y_2, z_4), (x_1, y_2, z_2)\}$. x_1 is the most occurrence with 3 times. Then $M' = \{(x_1, y_1, z_1), (x_2, y_3, z_2), (x_3, y_2, z_4)\}$ is the maximum 3-dimensional matching of M.

The *3-dimensional matching* (3DM) problem (without bound B) is the general version, which determines if there exists a 3-dimensional matching with size κ. In 1972, Karp presented some $\mathcal{NP}$-complete problems, including the 3DM problem [17].

3 Hardness for the TLCS Problems

In this section, we first prove that $P(ENL)$ and $P(ENE)$ are $\mathcal{APX}$-hard. After that, we prove that $P(LOL)$ and $P(LOE)$ are both $\mathcal{NP}$-hard. Finally, we show that they are $\mathcal{APX}$-hard.

3.1 $\mathcal{APX}$-hardness for $P(ENL)$ and $P(ENE)$

In this section, we will prove that $P(ENL)$ and $P(ENE)$ are $\mathcal{APX}$-hard.

A picture can be represented by a matrix whose elements are the objects in the picture. Let $a_{i_1,j_1} = \alpha$ and $a_{i_2,j_2} = \beta$ be two elements in matrix A, and $b_{p_1,q_1} = \alpha$ and $b_{p_2,q_2} = \beta$ be two elements in matrix B. Chang *et al.* defined that the relations of type-0, type-1 and type-2 between (a_{i_1,j_1}, a_{i_2,j_2}) and (b_{p_1,q_1}, b_{p_2,q_2}) as follows [8].

- type-0: $(i_2 - i_1) \times (p_2 - p_1) \geq 0$ and $(j_2 - j_1) \times (q_2 - q_1) \geq 0$.
- type-1: $\{(i_2 - i_1) \times (p_2 - p_1) > 0$ or $i_2 - i_1 = p_2 - p_1 = 0\}$ and $\{(j_2 - j_1) \times (q_2 - q_1) > 0$ or $j_2 - j_1 = q_2 - q_1 = 0\}$.
- type-2: $i_2 - i_1 = p_2 - p_1$ and $j_2 - j_1 = q_2 - q_1$.

In 2000, Guan *et al.* [11] proved that the maximum similar subpicture problems of type-0 and type-1 are $\mathcal{NP}$-hard by reducing from the 3-satisfiability (3SAT) problem, in which each clause has exactly three literals. Note that type-0 is exactly the same as $P(ENE)$. In addition, $P(ENL)$ can also be proved to be $\mathcal{NP}$-hard by reducing from 3SAT.

In this paper, we use the same transformation to prove that $P(ENL)$ and $P(ENE)$ are $\mathcal{APX}$-hard. In our proof, we transform from MAX 3SAT-3, instead of 3SAT, to $P(ENL)$ and $P(ENE)$. Even though the transformation sources are slightly different, the transformations are completely same, described as follows. The instance of MAX 3SAT-3 is represented by a set of Boolean variables $X = \{x_1, x_2, \cdots, x_n\}$, and a Boolean formula $C = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where each clause has the form $C_t = (v_{t,1} \vee v_{t,2} \vee v_{t,3})$, $1 \leq t \leq m$, $v_{t,1} = x_{k_{t,1}}$ or $\overline{x_{k_{t,1}}}$, $v_{t,2} = x_{k_{t,2}}$ or $\overline{x_{k_{t,2}}}$, $v_{t,3} = x_{k_{t,3}}$ or $\overline{x_{k_{t,3}}}$, $1 \leq k_{t,1} \neq k_{t,2} \neq k_{t,3} \leq n$, and each variable x_k appears in at most three clauses. Matrices A and B of $P(ENL)$ and $P(ENE)$, where $|A| = |B| = 2n \times 3m$, are constructed as follows [11].

$$a_{i,j} = \begin{cases} l_u^t & \begin{cases} \text{if } i = 2k-1,\ j = 3(t-1)+u, \\ \text{where the } u\text{th variable of } C_t \text{ is } x_k; \\ \text{if } i = 2k,\ j = 3(t-1)+u, \\ \text{where the } u\text{th variable of } C_t \text{ is } \overline{x_k}; \\ u = 1, 2, 3; \end{cases} \\ \alpha & \text{otherwise.} \end{cases} \quad (1)$$

$$b_{i,j} = \begin{cases} l_u^t & \begin{cases} \text{if } i = 2k,\ j = 3t+1-u, \\ \text{where the } u\text{th literal of } C_t \text{ is } x_k; \\ \text{if } i = 2k-1,\ j = 3t+1-u, \\ \text{where the } u\text{th literal of } C_t \text{ is } \overline{x_k}; \\ u = 1, 2, 3; \end{cases} \\ \beta & \text{otherwise.} \end{cases} \quad (2)$$

We denote the above transformation as Γ_{ENL}. Figure 2 shows an example of the above transformation. Every two rows of A or B correspond to one variable of MAX 3SAT, but they are in reverse order in A and B. Every three columns of A or B correspond to one clasue of MAX 3SAT, but they are in reverse order in A and B. Rows 1 and 2 correspond to x_1 and $\overline{x_1}$ in matrix A, but they are for $\overline{x_1}$ and x_1 in matrix B. Columns 1, 2 and 3 correspond to the first, second and third literals in matrix A, but they are for the third, second and first literals in matrix B. For example, l_1^2, l_2^2 and l_3^2 correspond to $C_2 = (x_1 \vee \overline{x_3} \vee \overline{x_2})$.

In the transformation Γ_{ENL}, if l_u^t is selected into the solution CU_{ENL}, then it means that its corresponding element $v_{t,u}$ in C_t is assigned to be

$$X = \{x_1, x_2, x_3, x_4\}$$

$$C = \left\{ \begin{array}{c} (x_1 \vee x_2 \vee x_3) \wedge \\ (x_1 \vee \overline{x_3} \vee \overline{x_2}) \wedge \\ (\overline{x_1} \vee \overline{x_2} \vee x_4) \end{array} \right\}$$

(a)

Matrix A

	1	2	3	4	5	6	7	8	9
1	l_1^1	α	α	l_1^2	α	α	α	α	α
2	α	α	α	α	α	α	l_1^3	α	α
3	α	l_2^1	α	α	α	α	α	α	α
4	α	α	α	α	α	l_3^2	α	l_2^3	α
5	α	α	l_3^1	α	α	α	α	α	α
6	α	α	α	α	l_2^2	α	α	α	α
7	α	α	α	α	α	α	α	α	l_3^3
8	α	α	α	α	α	α	α	α	α

Matrix B

	1	2	3	4	5	6	7	8	9
1	β	β	β	β	β	β	β	β	l_1^3
2	β	β	l_1^1	β	β	l_1^2	β	β	β
3	β	β	β	l_3^2	β	β	β	l_2^3	β
4	β	l_2^1	β	β	β	β	β	β	β
5	β	β	β	β	l_2^2	β	β	β	β
6	l_3^1	β	β	β	β	β	β	β	β
7	β	β	β	β	β	β	β	β	β
8	β	β	β	β	β	β	l_3^3	β	β

(b)

Figure 2: An example of the transformation Γ_{ENL} for proving the $\mathcal{APX}$-hardness of $P(ENL)$. (a) Instance sets X and C in MAX 3SAT-3. (b) Matrices A and B of $P(ENL)$, constructed from X and C.

true. Any pair of elements l_1^t, l_2^t and l_3^t are never in CU_{ENL} at the same time (proved in Lemma 1). It implies that if one of the literals in C_t is true, then C_t is true for MAX 3SAT-3. Accordingly, $|CU_{ENL}| \leq m$.

Lemma 1. *Suppose that $a_{i_1,j_1} = b_{p_1,q_1} = l_u^t$ and $a_{i_2,j_2} = b_{p_2,q_2} = l_{u'}^t$, where $1 \leq u \neq u' \leq 3$. $(a_{i_1,j_1}, b_{p_1,q_1})$ and $(a_{i_2,j_2}, b_{p_2,q_2})$ cannot be both in CU_{ENL}.*

Proof. If $u < u'$, then $j_1 < j_2$ and $q_1 > q_2$ do not obey the column relationship of corner for $P(ENL)$. Similarly, if $u > u'$, then $j_1 > j_2$ and $q_1 < q_2$ do not obey the column relationship of corner for $P(ENL)$. Thus, the lemma holds. □

Suppose that two elements l_u^t and $l_{u'}^{t'}$ correspond to the same variable x_k, one for x_k and the other for $\overline{x_k}$, where $t \neq t'$ and $1 \leq u, u' \leq 3$. They cannot be both in CU_{ENL}. It is proved in the following lemma.

Lemma 2. *Suppose that C_t and $C_{t'}$ have a common variable, $t \neq t'$, one is x_k and the other is $\overline{x_k}$, $a_{i_1,j_1} = b_{p_1,q_1} = l_u^t$, and $a_{i_2,j_2} = b_{p_2,q_2} = l_{u'}^{t'}$. $(a_{i_1,j_1}, b_{p_1,q_1})$ and $(a_{i_2,j_2}, b_{p_2,q_2})$ cannot be both in CU_{ENL}.*

Proof. If C_t contains x_k and $C_{t'}$ contains $\overline{x_k}$, then $i_1 < i_2$ and $p_1 > p_2$ do not obey the row relationship of corner for $P(ENL)$. Similarly, if C_t

contains $\overline{x_k}$ and $C_{t'}$contains x_k, then $i_1 > i_2$ and $p_1 < p_2$ do not obey the row relationship of corner for $P(ENL)$. □

Except the conflict conditions mentioned in Lemmas 1 and 2, any other pair of matchings can be both in CU_{ENL}.

Theorem 1. *The TLCS problem with $P(ENL)$ is $\mathcal{APX}$-hard.*

Proof. With Lemmas 1 and 2, the transformation Γ_{ENL} is correct. For instance I, let $\kappa = |opt_{MAX\ 3SAT\text{-}3}(I)|$. We have $opt_{ENL}(f(I)) = \kappa = |opt_{MAX\ 3SAT\text{-}3}(I)|$. For each $sol \in Sol_{ENL}(f(I))$ with $|sol| = oj_2$, we can get corresponding $sol' \in Sol_{MAX\ 3SAT\text{-}3}(I)$ with $|sol'| = oj_1 = oj_2$. $|opt_{MAX\ 3SAT\text{-}3}(I) - oj_1| = \kappa - oj_1 = \kappa - oj_2 = |opt_{ENL}(f(I)) - oj_2|$. Hence, Γ_{ENL} is an L-reduction from MAX 3SAT-3 to $P(ENL)$ with $\delta = 1$ and $\mu = 1$. □

The proof for $P(ENL)$ can be applied to $P(ENE)$, thus the following theorem can also be obtained.

Theorem 2. *The TLCS problem with $P(ENE)$ is $\mathcal{APX}$-hard.*

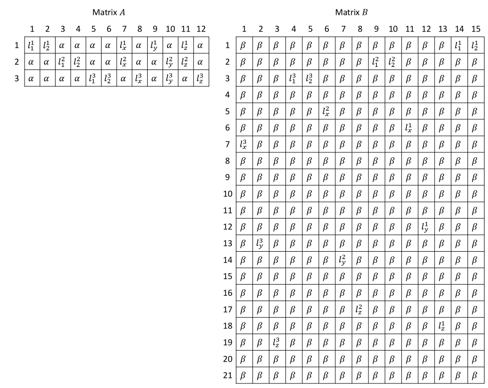
3.2 $\mathcal{NP}$-hardness for $P(LOL)$ and $P(LOE)$

In this section, we prove that $P(LOL)$ and $P(LOE)$ are $\mathcal{NP}$-hard by reducing from the 3DM problem. The transformation Γ_{LOL} for $P(LOL)$ is described as follows. The input instance of 3DM is represented by a set $M = \{M_1, M_2, \cdots, M_m\} \subseteq X \times Y \times Z$, where X, Y and Z are disjoint finite sets, $|M| = m$, and $|X| + |Y| + |Z| = n$. Let $M_t = (x_k,\ y_{k'},\ z_{k''}) \in M$, where $1 \le t \le m$, $x_k \in X$, $y_{k'} \in Y$, $z_{k''} \in Z$, $1 \le k \le |X|$, $1 \le k' \le |Y|$ and $1 \le k'' \le |Z|$. The matrices A and B of $P(LOL)$ are constructed as follows, where $|A| = m \times (2m + n)$ and $|B| = (m + nm) \times 5m$.

$$a_{i,j} = \begin{cases} l_u^t & \text{if } i = t, \text{ and } j = 2(t-1) + u,\ u = 1, 2; \\ l_x^t & \text{if } i = t, \text{ and } j = 2m + k; \\ l_y^t & \text{if } i = t, \text{ and } j = 2m + |X| + k'; \\ l_z^t & \text{if } i = t, \text{ and } j = 2m + |X| + |Y| + k''; \\ \alpha & \text{otherwise.} \end{cases} \tag{3}$$

$$M = \left\{ \begin{array}{c} (x_1, y_1, z_1), \\ (x_1, y_2, z_1), \\ (x_2, y_2, z_2) \end{array} \right\}$$

(a)



(b)

Figure 3: An example of Γ_{LOL} for proving the $\mathcal{NP}$-hardness of $P(LOL)$. (a) An input set M of the 3DM problem. (b) Matrices A and B of $P(LOL)$, constructed from M.

$$b_{i,j} = \begin{cases} l_u^t & \text{if } i = t, \\ & \text{and } j = 5(m-t) + 3 + u,\ u = 1, 2; \\ l_x^t & \text{if } i = m(1+k) - (t-1), \\ & \text{and } j = 5(m-t) + 1; \\ l_y^t & \text{if } i = m(1 + |X| + k') - (t-1), \\ & \text{and } j = 5(m-t) + 2; \\ l_z^t & \text{if } i = m(1 + |X| + |Y| + k'') \\ & -(t-1), \\ & \text{and } j = 5(m-t) + 3; \\ \beta & \text{otherwise.} \end{cases} \tag{4}$$

Figure 3 shows an example of Γ_{LOL}, where $m = 3$ and $n = 6$. Each row in matrix A and every submatrix of size $(m + nm) \times 5 = 21 \times 5$ in matrix B correspond to one element M_t of M. l_x^t, l_y^t and l_z^t correspond to elements x_k, $y_{k'}$ and $z_{k''}$ of M_t. l_1^t and l_2^t are the expanded symbols generated from M_t.

In matrix A of Figure 3, row t corresponds to M_t, columns 1 to 6 are for l_1^t and l_2^t, columns 7 and 8 are for x_1 and x_2, respectively, columns 9 and 10 are for y_1 and y_2, respectively. In matrix B, columns 11 to 15 are for M_1, columns 6 to 10 are for M_2 (reverse order). Rows 1 to 3 correspond to l_1^t and l_2^t. Rows 4 to 6 correspond to x_1, in which rows 4, 5 and 6 are for M_3, M_2 and M_1 (reverse order), respectively. Rows 7 to 9 correspond to x_2, in which rows 7, 8 and 9 are for M_3, M_2 and M_1, respectively. Rows 10 to 15 are for Y, and rows 16 to 21 are for Z.

Each symbol, except α and β, appears exactly once in matrix A and once in matrix B. Obviously, $|CU_{LOL}| \leq 5m$ (including all symbols, but excluding α and β). To obtain a tighter bound, for one element $M_t \in M$, two possible matches (l_1^t, l_2^t) or (l_x^t, l_y^t, l_z^t) can be made between A and B. Thus, $2m \leq |CU_{LOL}| \leq 3m$.

In the following, we will prove that $|CU_{LOL}| = 2m + \kappa$ if and only if there exists a 3-dimensional matching with size κ in M. Moreover, if $|CU_{LOL}| = 2m + \kappa$ and (l_x^t, l_y^t, l_z^t) are three of the common elements in CU_{LOL}, then M_t will be picked as one match in the optimal solution of M. The formal proof is accomplished by the following lemmas.

Lemma 3. *Suppose that $a_{i_1,j_1} = b_{p_1,q_1} = l_u^t$, where $u = 1, 2$, and $a_{i_2,j_2} = b_{p_2,q_2} = l_v^t$, where $v = x, y, z$. $(a_{i_1,j_1}, b_{p_1,q_1})$ and $(a_{i_2,j_2}, b_{p_2,q_2})$ cannot be both in CU_{LOL}.*

Proof. By the definitions in (3) and (4), it is clear that $i_1 = i_2$, $j_1 < j_2$, $p_1 < p_2$ and $q_1 > q_2$. We have that $j_1 < j_2$ and $q_1 > q_2$ do not obey the column relationship of side $(i_1 = i_2)$ in $P(LOL)$. Thus, $(a_{i_1,j_1}, b_{p_1,q_1})$ and $(a_{i_2,j_2}, b_{p_2,q_2})$ cannot be both in CU_{LOL}. □

Lemma 4. *Suppose that M_t and $M_{t'}$ have a common element, $a_{i_1,j_1} = b_{p_1,q_1} = l_v^t$ and $a_{i_2,j_2} = b_{p_2,q_2} = l_v^{t'}$, where $v = x, y, z$ and $t \neq t'$. $(a_{i_1,j_1}, b_{p_1,q_1})$ and $(a_{i_2,j_2}, b_{p_2,q_2})$ cannot be both in CU_{LOL}.*

Proof. Assume that $t < t'$. By the definitions in (3) and (4), it is clear that $i_1 < i_2$, $j_1 = j_2$, $p_1 > p_2$ and $q_1 > q_2$. We have that $i_1 < i_2$ and $p_1 > p_2$ do not obey the row relationship of side $(j_1 = j_2)$ in $P(LOL)$. A similar result can be obtained when $t > t'$. Therefore, $(a_{i_1,j_1}, b_{p_1,q_1})$ and $(a_{i_2,j_2}, b_{p_2,q_2})$ cannot be both in CU_{LOL}. □

Except the conflict conditions in Lemmas 3 and 4, any other pair of matchings can be both in CU_{LOL}.

Lemma 5. *$|CU_{LOL}| = 2m+\kappa$ if and only if there exists a 3-dimensional matching with size κ.*

Proof. If there exists a 3-dimensional matching with size κ, it is obvious that in Γ_{LOL}, $2(m - \kappa) + 3\kappa = 2m + \kappa$ elements in matrix A can be matched with elements in B.

With Lemma 3, we pick up either 2 or 3 elements from each row in A. If 3 elements are picked up in one row of A, then l_x^t, l_y^t and l_z^t are the targets. It means that M_t is picked up in the solution of 3DM. If $|CU_{LOL}| = 2m+\kappa$, it means that κ rows of A are picked up with 3 elements. With Lemma 4, the picked elements are all distinct in X, Y or Z. Therefore, by Γ_{LOL}, the matches we pick up in matrices A and B of $P(LOL)$ correspond to a 3-dimensional matching with size κ. □

With Lemma 5, 3DM reduces to $P(LOL)$, and thus we have the following result.

Theorem 3. *The TLCS problem with $P(LOL)$ is $\mathcal{NP}$-hard.*

Similarly, the reduction and Lemma 5 can also be applied to $P(LOE)$, thus we have the following result.

Theorem 4. *The TLCS problem with $P(LOE)$ is $\mathcal{NP}$-hard.*

3.3 $\mathcal{APX}$-hardness for $P(LOL)$ and $P(LOE)$

In this section, we prove that $P(LOL)$ and $P(LOE)$ are $\mathcal{APX}$-hard. We use the same transformation Γ_{LOL} from the MAX 3DM-3 problem, instead of the 3DM problem, to $P(LOL)$ and $P(LOE)$.

If there is a matching $(x, y, z) \in opt_{MAX\ 3DM\text{-}3}$, then at most 6 matches are not in $opt_{MAX\ 3DM\text{-}3}$. For example, if $(x, y, z) \in opt_{MAX\ 3DM\text{-}3}$, then (x, y_1, z_1), (x, y_2, z_2), (x_1, y, z_1), (x_2, y, z_2), (x_1, y_1, z), (x_2, y_2, z) cannot be in $opt_{MAX\ 3DM\text{-}3}$. It is clear that $m \leq (6 + 1)\kappa$.

Theorem 5. *The TLCS problem with $P(LOL)$ is $\mathcal{APX}$-hard.*

Proof. For instance I, let $\kappa = |opt_{MAX\ 3DM\text{-}3}(I)|$. We have $|opt_{LOL}(f(I))| = 2m+\kappa \leq 14\kappa+\kappa = 15\kappa$. That is, $|opt_{LOL}(f(I))| \leq 15|opt_{MAX\ 3DM\text{-}3}(I)|$. Thus, $\delta = 15$. For each $sol \in Sol_{LOL}(f(I))$ with $|sol| = oj_2$, we can get corresponding sol' $\in Sol_{MAX\ 3DM\text{-}3}(I)$ with $|sol'| = oj_1 \leq \frac{1}{3}oj_2$, since three elements picked up in one row of A of $P(LOL)$ corresponds to one picked M_t of MAX 3DM-3. $oj_2 - oj_1 = \frac{2}{3}oj_2 \leq 2m = (2m + \kappa) - \kappa$ implies $|opt_{MAX\ 3DM\text{-}3}(I) - oj_1| = k - oj_1 \leq (2m + \kappa) - oj_2 = |opt_{LOL}(f(I)) - oj_2|$. Hence, Γ_{LOL} is an L-reduction from MAX 3DM-3 to $P(LOL)$ with $\delta = 15$ and $\mu = 1$. □

Similarly, the same reduction and proof can also be applied to $P(LOE)$, thus we have the following result.

Theorem 6. *The TLCS problem with $P(LOE)$ is $\mathcal{APX}$-hard.*

4 Conclusion

In this paper, we prove that $P(LOL)$ and $P(LOE)$ are $\mathcal{NP}$-hard, showing the correctness of Chan's conjecture. In addition, we prove that the four valid definitions $P(ENL)$, $P(ENE)$, $P(LOL)$ and $P(LOE)$ are all $\mathcal{APX}$-hard. In the future, it is worthy to design approximation algorithms for various TLCS problems. It is also interesting to discover whether these problems are $\mathcal{APX}$-complete.

References

[1] E. Amaldi and V. Kann, "The complexity and approximability of finding maximum feasible subsystems of linear relations," *Theoretical computer science*, Vol. 147, No. 1-2, pp. 181–210, 1995.

[2] A. Amir, T. Hartmana, O. Kapaha, B. R. Shaloma, and D. Tsur, "Generalized LCS," *Theoretical Computer Science*, Vol. 409, pp. 438–449, 2008.

[3] H.-Y. Ann, C.-B. Yang, C.-T. Tseng, and C.-Y. Hor, "A fast and simple algorithm for computing the longest common subsequence of run-length encoded strings," *Information Processing Letters*, Vol. 108, pp. 360–364, 2008.

[4] G. Ausiello, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties.* Springer Science & Business Media, 1999.

[5] R. A. Baeza-Yates, "Similarity in two-dimensional strings," *Proc. of the 4th Annual International Conference on Computing and Combinatorics(COCOON '98)*, pp. 319–328, London, UK, 1998.

[6] H.-T. Chan, C.-B. Yang, and Y.-H. Peng, "The generalized denitions of the two-dimensional largest common substructure problems," *Proc. of the 33rd Workshop on Combinatorial Mathematics and Computation Theory*, Taipei, Taiwan, pp. 1–12, 2016.

[7] S.-K. Chang, E. Jungert, and Y. Li, "Representation and retrieval of symbolic pictures using generalized 2D strings," *SPIE Proc. Visual Communications and Image Processing*, pp. 1360–1372, Philadelphia, 1989.

[8] S.-K. Chang, Q.-Y. Shi, and C.-W. Yan, "Iconic indexing by 2-D strings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 3, pp. 413–428, 1987.

[9] P. Crescenzi and A. Panconesi, "Completeness in approximation classes," *Information and Computation*, Vol. 93, No. 2, pp. 241–262, 1991.

[10] G. Galbiati, F. Maffioli, and A. Morzenti, "A short note on the approximability of the maximum leaves spanning tree problem," *Information Processing Letters*, Vol. 52, No. 1, pp. 45–49, 1994.

[11] D. Guan, C.-Y. Chou, and C.-W. Chen, "Computational complexity of similarity retrieval in a pictorial database," *Information Processing Letters*, Vol. 75, No. 3, pp. 113 – 117, 2000.

[12] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Communications of the ACM*, Vol. 18, pp. 341–343, 1975.

[13] K.-S. Huang, C.-B. Yang, K.-T. Tseng, Y.-H. Peng, and H.-Y. Ann, "Dynamic programming algorithms for the mosaic longest common subsequence problem," *Information Processing Letters*, Vol. 102, pp. 99–103, 2007.

[14] J. W. Hunt and T. G. Szymanski, "A fast algorithm for computing longest common subsequences," *Communications of the ACM*, Vol. 20, pp. 350–353, 1977.

[15] C. S. Iliopoulos and M. S. Rahman, "New efficient algorithms for the LCS and constrained LCS problems," *Information Processing Letters*, Vol. 106, pp. 13–18, 2008.

[16] V. Kann, "Maximum bounded 3-dimensional matching is MAX SNP-complete," *Information Processing Letters*, Vol. 37, No. 1, pp. 27–35, 1991.

[17] R. M. Karp, "Reducibility among combinatorial problems," *Proceedings of a symposium on the Complexity of Computer Computations*, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, pp. 85–103, 1972.

[18] S.-Y. Lee and F.-J. Hsu, "Spatial reasoning and similarity retrieval of images using 2D C-string knowledge representation," *Pattern Recognition*, Vol. 25, No. 3, pp. 305 – 318, 1992.

[19] C. H. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes," *Journal of computer and system sciences*, Vol. 43, No. 3, pp. 425–440, 1991.

[20] Y.-H. Peng, C.-B. Yang, K.-S. Huang, C.-T. Tseng, and C.-Y. Hor, "Efficient sparse dynamic programming for the merged LCS problem with block constraints," *International Journal of Innovative Computing, Information and Control*, Vol. 6, pp. 1935–1947, 2010.