2023 14th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI) | 979-8-3503-2422-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/IIAI-AAI59060.2023.00088

An Algorithm for Finding the Longest Common Almost Wave Subsequence with the Trend

1st Tsung-Lin Lu
Dept. Computer Science & Engineering
National Sun Yat-sen University
Kaohsiung, Taiwan

2nd Kuo-Si Huang
Dept. Business Computing
NKUST
Kaohsiung, Taiwan

3rd Chang-Biau Yang
Dept. Computer Science & Engineering
National Sun Yat-sen University
Kaohsiung, Taiwan
cbyang@cse.nsysu.edu.tw

***Abstract*—Numeric sequences often exhibit similar behaviors or trends, such as fluctuations in electrocardiograms and stock values that resemble waves. This paper attempts to find the longest common almost wave subsequence with the trend (LCaWSt) between two numeric sequences A and B, given a wave trend T and a certain tolerance constant c. The LCaWSt problem aims to find the longest common subsequence of A and B, consisting of the almost increasing and decreasing segments alternatively, whose trend follows the prefix of T. We propose an algorithm for finding the LCaWSt in O(mnc) time and O(nc) space, where m and n denote the lengths of A and B, respectively, $m \geq n$.**

***Index Terms*—longest increasing subsequence, longest common subsequence, longest common almost wave subsequence, trend sequence, segments**

I. Introduction

Numeric sequences are common in our daily lives and their trends may carry significant meaning and importance. Several algorithms have been developed for finding the longest increasing subsequence (LIS) of a numeric sequence [1], [2], [7], [13], aiming to find the longest subsequence with the increasing order. In addition, the MUMmer system [8] uses the LIS algorithm to align genomes in bioinformatics.

In real applications, the LIS may be relaxed to get a longer subsequence. Based on the trend of sequence, the concept of *almost increasing* (AI) [9] can tolerate a drop of up to a tolerance constant c from the maximum element encountered so far. For example, Figure 1 shows the chart of the stock prices Additionally, in medicine, an electrocardiogram (ECG) that records the electrical activity of the heart [6] is also a wave sequence, as shown in Figure 2.

The *longest common almost increasing subsequence* (LCaIS) problem [3], [12], [14] is a generalized combination of the *longest common subsequence* (LCS) problem [15] and the *longest almost increasing subsequence* (LaIS) problem. The LCS problem aims to find the common subsequence, with the maximal length, of two input sequences. Consequently, the LCaIS problem aims to find the common LaIS of two input sequences. As an example shown in Figure 3, when $A = \langle 1, 3, 7, 5, 4, 8 \rangle$, $B = \langle 1, 7, 5, 3, 4, 8 \rangle$ and a tolerance constant

This research work was partially supported by National Science and Technology Council of Taiwan under contract MOST 111-2221-E-110-034.

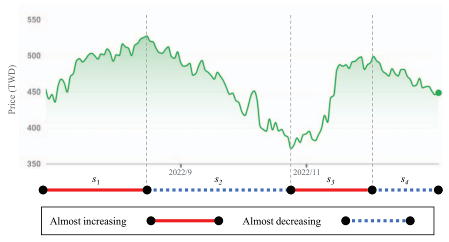

Fig. 1. A price-time chart of the stock of TSMC (Taiwan Semiconductor Manufacturing Co., Ltd.) from 2022/7/1 to 2022/12/31 and its trend. In this figure, segments s_1 and s_3 are AI; segments s_2 and s_4 are AD.

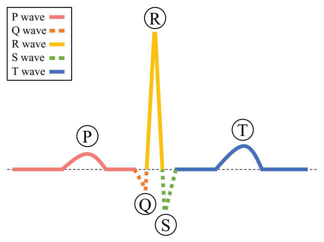

Fig. 2. A normal sinus rhythm, a composite wave starting with an AI segment and consisting of waves P, Q, R, S and T.

$c = 3$; lengths of LIS and LaIS of A are 4 and 5, respectively; the LCaIS is $\langle 1, 7, 5, 8 \rangle$ with length 4.

A stock price sequence and an ECG, as shown in Figures 1 and 2, are both wave sequences, which consist of alternating AI and AD segments. Consequently, the LIS cannot perfectly describe its trend. To get a better trend, Chen and Yang [5] (2020) defined a new generalized LIS problem, called the *longest wave subsequence* (LWS) problem.

The *longest almost wave subsequence* (LaWS) problem, a combination of the LaIS problem and the LWS problem, was defined by Lin (2023) [10]. An almost wave sequence consists of the AI and AD segments alternatively. Thus, the target of the LaWS problem is similar to the target of the LWS problems, but its increasing and decreasing limitations are relaxed. The *longest common wave subsequence* (LCWS) problem, a version of two input sequences of the LWS problem, was defined by Chang and Yang (2021) [4]. In fact, the LCWS

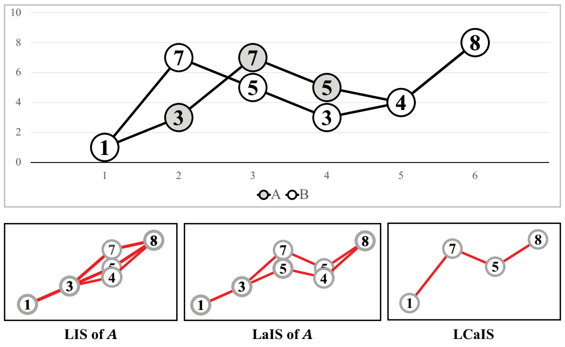

Fig. 3. The visualization of the LIS, LaIS and LCaIS problems with $A = \langle 1, 3, 7, 5, 4, 8\rangle$, $B = \langle 1, 7, 5, 3, 4, 8\rangle$ and a tolerance constant $c = 3$. The LIS answers of A may be $\langle 1, 3, 7, 8\rangle$, $\langle 1, 3, 5, 8\rangle$ or $\langle 1, 3, 4, 8\rangle$, with length 4. The LaIS of A may be $\langle 1, 3, 7, 5, 8\rangle$ or $\langle 1, 3, 5, 4, 8\rangle$ with length 5. The LCaIS of A and B is $\langle 1, 7, 5, 8\rangle$ with length 4.

problem combines the LCS problem and the LWS problem, so the LCWS aims to find the common wave subsequence, with the maximal length, of two input sequences. This paper proposes the longest common almost wave subsequence with the trend (LCaWSt) problem, a variant of the LaWS problem with a trend and two input sequences.

This paper is organized as follows. Section II describes the preliminaries of the LCaWSt problem. Section III proposes the LCaWSt algorithm. Finally, Section IV concludes this paper.

II. Preliminaries

Definition 1: (*almost increasing and almost decreasing [9]*) Given a numeric sequence $A = \langle a_1, a_2, a_3, \cdots, a_n\rangle$ and a tolerance constant c ($c \neq 0$), it tolerates a drop or a rise from the maximum or the minimum element appearing so far, respectively. A is *almost increasing* if $\max\{a_k | 1 \leq k \leq i-1\} - c < a_i$; A is *almost decreasing* if $\min\{a_k | 1 \leq k \leq i-1\} + c > a_i$ for $2 \leq i \leq n$.

Definition 2: (*trend sequence T [5]*) Given a numeric sequence $W = \langle w_1, w_2, \cdots, w_n\rangle$, its *trend sequence* $T = \langle t_1, t_2, \cdots, t_n\rangle$ is defined by *Equation 1*.

$$t_i = \begin{cases} 0 \textit{ if } w_{i-1} > w_i \textit{ for } 2 \leq i \leq n; \\ 1 \textit{ if } w_{i-1} < w_i \textit{ for } 2 \leq i \leq n; \\ \textit{complement of } t_2 \textit{ if } i = 1. \end{cases} \tag{1}$$

Definition 3: (*turning point [5]*) Given a numeric sequence W and its trend sequence T, where $t_1 \neq t_2$, w_i is a turning point if $t_i \neq t_{i+1}$. That is, a *turning point* is a joint element of two consecutive segments with different trends.

Definition 4: (*LaWSt problem [10]*) Given a numeric sequence $A = \langle a_1, a_2, \cdots, a_n\rangle$, a trend sequence $T = \langle t_1, t_2, \cdots, t_n\rangle$, and a tolerance constant c, the *longest almost wave subsequence with trend* (LaWSt) is the longest subsequence of A, consisting of the AI and AD segments alternatively, whose trend follows the prefix of T.

Definition 5: (*LCWSt problem [4]*) Given two numeric sequences $A = \langle a_1, a_2, \cdots, a_m\rangle$ and $B = \langle b_1, b_2, \cdots, b_n\rangle$, and a trend sequence $T = \langle t_1, t_2, \cdots, t_n\rangle$, where $m \geq n$, the *longest common wave subsequence with trend* (LCWSt) is the longest common subsequence of A and B, consisting of the increasing and decreasing segments alternatively, whose trend sequence follows the prefix of T.

By combining the above definitions, in this paper, we define the more generalized problem as follows.

Definition 6: (*LCaWSt problem*) Given two numeric sequences $A = \langle a_1, a_2, \cdots, a_m\rangle$ and $B = \langle b_1, b_2, \cdots, b_n\rangle$, a trend sequence $T = \langle t_1, t_2, \cdots, t_n\rangle$, $m \geq n$, and a tolerance constant c, the *longest common almost wave subsequence with trend* (LCaWSt) is the longest common subsequence of A and B, consisting of alternating AI and AD segments, whose trend sequence follows the prefix of T.

Figure 4 shows an example of the LCaWSt. Given $A = \langle 3, 2, 7, 4, 5, 2, 8\rangle$, $B = \langle 3, 8, 2, 7, 4, 5, 2\rangle$, a trend sequence $T = \langle 0, 1, 1, 0, 0, 1, 1\rangle$ and a tolerance constant $c = 3$, the LCaWSt is $\langle 3, 2, 7, 5, 2\rangle$ with length 5, whose trend sequence follows $T_{1..5} = \langle 0, 1, 1, 0, 0\rangle$. Furthermore, the LCaWSt answer $\langle 3, 2, 7, 5, 2\rangle$ consists of 2 segments that $s_1 = \langle 3, 2, 7\rangle$ is AI and $s_2 = \langle 7, 5, 2\rangle$ is AD. In addition, Figure 4 shows that we cannot get the LCaWSt from the LaWSt or LCWSt answer directly. Here, we propose an O(mnc)-time algorithm to solve the LCaWSt problem.

III. The LCaWSt Algorithm

Algorithm 1 is the main structure of our LCaWSt algorithm. It contains two parts, the updates of L-table and β-table, formally defined in Definition 7. ExtFindTPt(β, b_j, T) in Algorithm 2 is to update L by extending β to L, finding the best turning point p, and returning L and p. In addition, we use p to update the first element in L.

The functions of Algorithms 3 and 4 are used to update β. Filter(L, a_i, c) in Algorithm 3 filters L by a_i and c, and returns filtered L. That is, Filter(L, a_i, c) keeps 2-tuples, which can get the longer potential solutions in the future from L based on a_i and c. DomMerge(L, β) in Algorithm 4 merges L and β to H by removing the dominated 2-tuples and returns H. In addition, Algorithms 2, 3 and 4 have two versions, superscripted by $+$ and $-$. $+$ is used in an AI segment and $-$ is used in an AD segment.

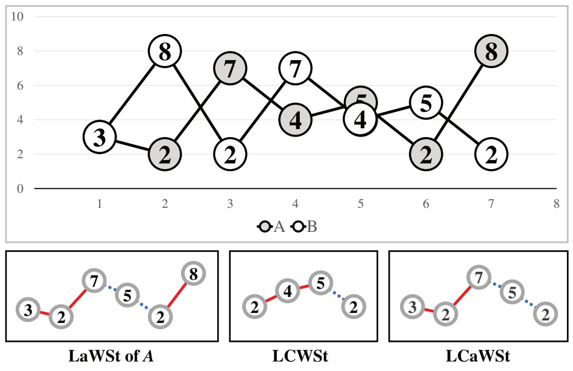

Fig. 4. The visualization of the LaWSt, LCWSt, and LCaWSt problems with $A = \langle 3, 2, 7, 4, 5, 2, 8\rangle$, $B = \langle 3, 8, 2, 7, 4, 5, 2\rangle$, $T = \langle 0, 1, 1, 0, 0, 1, 1\rangle$, and $c = 3$. The LaWSt of A is $\langle 3, 2, 7, 5, 2, 8\rangle$ with length 6. The LCWSt is $\langle 2, 4, 5, 2\rangle$ with length 4. The LCaWSt is $\langle 3, 2, 7, 5, 2\rangle$ with length 5. Here, red solid lines represent AI segments and blue dotted lines represent AD segments.

Algorithm 1 The computation of the LCaWSt length.

```
Require: Two numeric sequences $A = \langle a_1, a_2, \cdots, a_m\rangle$, $B = \langle b_1, b_2, \cdots, b_n\rangle$, $m \geq n$, a trend sequence $T = \langle t_1, t_2, \cdots, t_n, t_{n+1}\rangle$, $t_i \in \{0,1\}$, where $t_1 \neq t_2$, $t_{n+1} = t_n$ (a virtual trend) and a tolerance constant $c$.
Ensure: The LCaWSt length.
 1: $L^+(0,j) \leftarrow \{\langle b_j, 0\rangle\}$ and $L^-(0,j) \leftarrow \{\langle b_j, 0\rangle\}$, for $1 \leq j \leq n$
 2: $\beta^+(i,0) \leftarrow \{\langle a_i, 0\rangle\}$ and $\beta^-(i,0) \leftarrow \{\langle a_i, 0\rangle\}$, for $1 \leq i \leq m$
 3: for $i = 1$ to $m$ do
 4:     for $j = 1$ to $n$ do
 5:         if $a_i = b_j$ then                                                        ▷ Update $L$
 6:             $\langle L^+(i,j), p^-\rangle \leftarrow$ ExtFindTPt($\beta^+(i, j-1)$, $b_j$, $T$)    ▷ $p^-$ is the best turning point in the AD segment
 7:             $\langle L^-(i,j), p^+\rangle \leftarrow$ ExtFindTPt($\beta^-(i, j-1)$, $b_j$, $T$)    ▷ $p^+$ is the best turning point in the AI segment
 8:             $L^+(i,j)$.front().$len \leftarrow \max\{L^+(i,j)$.front().$len, p^+.len\}$
 9:             $L^-(i,j)$.front().$len \leftarrow \max\{L^+(i,j)$.front().$len, p^-.len\}$
10:         else
11:             $L^+(i,j) \leftarrow L^+(i-1,j)$
12:             $L^-(i,j) \leftarrow L^-(i-1,j)$
13:         $\beta^+(i,j) \leftarrow$ DomMerge$^+$(Filter$^+$($L^+(i-1,j)$, $a_i$, $c$), $\beta^+(i,j-1)$)           ▷ Update $\beta$
14:         $\beta^-(i,j) \leftarrow$ DomMerge$^-$(Filter$^-$($L^-(i-1,j)$, $a_i$, $c$), $\beta^-(i,j-1)$)
15: return $\max\{\{p.len | p \in L^+(m,j') \text{ for } 1 \leq j' \leq n\}, \{p.len | p \in L^-(m,j') \text{ for } 1 \leq j' \leq n\}\}$
```

Definition 7: For calculating the LCaWSt length of $A_{1..i}$ and $B_{1..j}$, the information of some potential solutions is stored in the sets $L^+(i,j)$, $L^-(i,j)$, $\beta^+(i,j)$ and $\beta^-(i,j)$, where each element is a 2-tuple $\langle base, len\rangle$. They are defined as follows.

- $L^+(i,j) = \{\langle base, len\rangle | len$ is the length of a specific LCaWSt solution of $A_{1..i}$ and $B_{1..j}$, that ends at b_j and $b_j \leq base < b_j + c$, where $base$ is the maximal value in the last AI segment.$\}$.
  $L^-(i,j) = \{\langle base, len\rangle | len$ is the length of a specific LCaWSt solution of $A_{1..i}$ and $B_{1..j}$, that ends at b_j and $b_j \geq base > b_j - c$, where $base$ is the minimal value in the last AD segment.$\}$.
- $\beta^+(i,j) = \{\langle a_i, len\rangle | len = \max\{p.len | p \in L^+(i-1,j'), p.base \leq a_i, 1 \leq j' \leq j\}\} \cup \{\langle a', len\rangle | len = \max\{p.len | p \in L^+(i-1,j'), a_i < p.base = a' < a_i + c, 1 \leq j' \leq j\} \neq \varnothing\}$.
  $\beta^-(i,j) = \{\langle a_i, len\rangle | len = \min\{p.len | p \in L^-(i-1,j'), p.base \geq a_i, 1 \leq j' \leq j\}\} \cup \{\langle a', len\rangle | len = \min\{p.len | p \in L^-(i-1,j'), a_i > p.base = a' > a_i - c, 1 \leq j' \leq j\} \neq \varnothing\}$.

In Definition 7, $\beta^+(i,j)$ consists of two parts. One collects the maximal length from $L^+(i-1,j')$ with $base \leq a_i$, for $1 \leq j' \leq j$, and the other collects the dominant 2-tuples from $L^+(i-1,j')$ with $a_i < base < a_i + c$, for $1 \leq j' \leq j$. Thus, the number of elements in $\beta^+(i,j)$ is bounded by c. For each $p \in \beta^+(i,j-1)$, it implies that a_i can be appended to p to become a longer answer ending with the AD segment if $a_i = b_j$. In other words, $\beta^+(i,j-1)$ collects all possible extensions ending with the AD segment for $a_i = b_j$. It is similar for $\beta^-(i,j)$.

Definition 8: (*domination*) For two 2-tuple elements p_1 and p_2, if $p_1.base = p_2.base$ and $p_1.len \geq p_2.len$, we say that p_1 *dominates* p_2.

All 2-tuple elements in each $L^+(i,j)$ are dominant, and

Algorithm 2 ExtFindTPt$^+(\beta,\, b_j,\, T)$ for the LCaWSt problem

```
Require: A linked list $\beta$, a matching value $b_j$, and a trend sequence $T = \langle t_1, t_2, \cdots, t_n \rangle$, $t_i \in \{0, 1\}$.
Ensure: $(L, p)$: an extended $L$ and the best turning point $p$.
1: $L \leftarrow \{\langle b_j, 0\rangle\}$, $p \leftarrow \langle b_j, 0\rangle$
2: for $b$ in $\beta$ do
3:     if (first segment is AI and $b.len = 0$) or ($b.base = b_j$) then        ▷ EFTPt$^-$: first segment is AD
4:         $L$.FRONT().$len = b.len + 1$        ▷ Extend
5:     else if $b.base \neq b_j$ then
6:         $L$.PUSHBACK($\langle b.base, b.len + 1\rangle$)        ▷ Extend
7:     if $b.len \neq 0$ and position $b.len + 1$ is a turning point then
8:         $p.len \leftarrow \max\{p.len, b.len + 1\}$        ▷ Find the best turning point
9: return $(L, p)$
```

Algorithm 3 Filter$^+(L,\, a_i,\, c)$

```
Require: A linked list $L$, the $i$th element $a_i$ in $A$ and a tolerance constant $c$.
Ensure: A linked list $F$, consisting of 2-tuples, which can get the longer potential solution in the future from $L$ based on $a_i$ and $c$.
1: $F \leftarrow \{\langle a_i, 0\rangle\}$        ▷ A filtered $L$
2: for $l$ in $L$ do
3:     if $l.base \leq a_i$ and position $l.len$ is not a turning point then        ▷ Filter$^-$: $l.base \geq a_i$
4:         $F$.LAST().$len \leftarrow \max\{F.\mathrm{LAST}().len, l.len\}$
5:     else if $a_i < l.base < a_i + c$ then        ▷ Filter$^-$: $a_i > l.base > a_i - c$
6:         $F$.PUSHBACK($l$)
7: return $F$
```

Algorithm 4 DomMerge$^+(L, \beta)$

```
Require: Two linked lists $L$ and $\beta$.
Ensure: A linked list $H$, which is merged from $L$ and $\beta$ by removing the dominated 2-tuples.
1: $H \leftarrow \varnothing$
2: Set pointers $\check{L} \leftarrow L$.BEGIN() and $\check{\beta} \leftarrow \beta$.BEGIN()        ▷ $\check{L}$ and $\check{\beta}$ are pointers
3: while $\check{L} \neq L$.END() or $\check{\beta} \neq \beta$.END() do
4:     if $\check{\beta} = \beta$.end() or $\check{L}.base < \check{\beta}.base$ then        ▷ DM$^-$: $\check{L}.base > \check{\beta}.base$
5:         $H$.PUSHBACK($\check{L}$)
6:         $\check{L} \leftarrow \check{L}$.NEXT()
7:     else if $\check{L} = L$.END() or $\check{L}.base > \check{\beta}.base$ then        ▷ DM$^-$: $\check{L}.base < \check{\beta}.base$
8:         $H$.PUSHBACK($\check{\beta}$)
9:         $\check{\beta} \leftarrow \check{\beta}$.NEXT()
10:    else
11:        $H$.PUSHBACK($\langle \check{L}.base, \max\{\check{L}.len, \check{\beta}.len\}\rangle$)        ▷ Domination
12:        $\check{L} \leftarrow \check{L}$.NEXT(), $\check{\beta} \leftarrow \check{\beta}$.NEXT()
13: return $H$
```

each 2-tuple represents a potential LCaWSt solution for $a_i = b_j$ when the last segment is AI. Each potential solution may be the final LCaWSt solution of A and B when the last segment is AI. Since we cannot guarantee which potential solution will be the best, we preserve all potential solutions by the 2-tuple elements in $L^+(i,j)$. It is similar for $L^-(i,j)$ when the last segment is AD.

Algorithm 1 finds the LCaWSt by using the L-table and β-table. Examples for constructing the L-table and β-table are shown in Tables I and II, respectively. Table III shows the potential LCaWSt solutions of Table I.

For row $i = 1$ ($a_i = a_1 = 3$), $a_1 = b_1 = 3$ gets the answer $\langle 3\rangle$ with length 1. Thus, $L^+(i,1) = L^+(1,1) = \{\langle 3,1\rangle\}$ means that 3 is the maximal value in the last AI segment with length 1. However, $\langle 3,1\rangle \notin L^-(1,1)$ because $t_2 = 1$ means that it should be in an AI segment.

For row i = 2 ($a_2 = 2$), we have $\beta^+(i,1) = \beta^+(2,1) = \{\langle 3,1\rangle\}$, where $\langle 3,1\rangle \in L^+(1,1)$, and $a_2 = 2 < 3 < 2 + c$. It implies that $a_2 = 2$ can be appended to 3 to become a longer answer in row 2 when $a_2 = b_j$ in the future. For $b_3 = 2 = a_2$, 2 extends $\langle 3,1\rangle \in \beta^+(2,2)$ to become $\langle 3, 1+1\rangle$ (adding 1 to the length). Thus, $\langle 3,2\rangle \in L^+(2,3)$. In addition,

TABLE I

THE L-TABLE WITH $A = \langle 3, 2, 7, 4, 5, 2, 8\rangle$, $B = \langle 3, 8, 2, 7, 4, 5, 2\rangle$, $T = \langle 0, 1, 1, 0, 0, 1, 1\rangle$ AND $c = 3$, WHOSE LCAWST ANSWER IS $\langle 3, 2, 7, 5, 2\rangle$ WITH LENGTH 5. SYMBOLS $+$ AND $-$ CORRESPOND TO AN AI SEGMENT AND AN AD SEGMENT, RESPECTIVELY. A BLANK CELL MEANS THAT ALL ELEMENTS OF THIS CELL ARE IDENTICAL TO ITS UPPER CELL.

B / A	L	$b_1 = 3$	$b_2 = 8$	$b_3 = 2$	$b_4 = 7$	$b_5 = 4$	$b_6 = 5$	$b_7 = 2$
$a_1 = 3$	$+$	**$\langle 3, 1\rangle$**						
	$-$							
$a_2 = 2$	$+$			$\langle 2, 1\rangle$ **$\langle 3, 2\rangle$**				$\langle 2, 1\rangle$ $\langle 3, 2\rangle$
	$-$							
$a_3 = 7$	$+$				**$\langle 7, 3\rangle$**			
	$-$				**$\langle 7, 3\rangle$**			
$a_4 = 4$	$+$					$\langle 4, 3\rangle$		
	$-$					**$\langle 4, 4\rangle$**		
$a_5 = 5$	$+$						$\langle 5, 5\rangle$	
	$-$						$\langle 5, 4\rangle$ $\langle 4, 5\rangle$	
$a_6 = 2$	$+$			$\langle 2, 1\rangle$ $\langle 3, 2\rangle$				**$\langle 2, 5\rangle$** $\langle 3, 3\rangle$
	$-$							**$\langle 2, 5\rangle$**
$a_7 = 8$	$+$		$\langle 8, 2\rangle$					
	$-$							

TABLE II

THE β-TABLE WITH $A = \langle 3, 2, 7, 4, 5, 2, 8\rangle$, $B = \langle 3, 8, 2, 7, 4, 5, 2\rangle$, $T = \langle 0, 1, 1, 0, 0, 1, 1\rangle$ AND $c = 3$. SYMBOLS $+$ AND $-$ CORRESPOND TO AN AI SEGMENT AND AN AD SEGMENT, RESPECTIVELY. A BLANK CELL MEANS THAT ALL ELEMENTS OF THIS CELL ARE IDENTICAL TO ITS LEFT CELL.

B / A	β	$b_1 = 3$	$b_2 = 8$	$b_3 = 2$	$b_4 = 7$	$b_5 = 4$	$b_6 = 5$	$b_7 = 2$
$a_1 = 3$	$+$							
	$-$							
$a_2 = 2$	$+$	$\langle 3, 1\rangle$						
	$-$							
$a_3 = 7$	$+$	$\langle 7, 1\rangle$		$\langle 7, 2\rangle$				
	$-$							
$a_4 = 4$	$+$	$\langle 4, 1\rangle$		$\langle 4, 2\rangle$				
	$-$				$\langle 4, 3\rangle$			
$a_5 = 5$	$+$	$\langle 5, 1\rangle$		$\langle 5, 2\rangle$				
	$-$				$\langle 5, 3\rangle$	$\langle 5, 3\rangle$ $\langle 4, 4\rangle$		
$a_6 = 2$	$+$	$\langle 3, 1\rangle$		$\langle 2, 1\rangle$ $\langle 3, 2\rangle$				
	$-$				$\langle 2, 3\rangle$	$\langle 2, 4\rangle$		
$a_7 = 8$	$+$	$\langle 8, 1\rangle$		$\langle 8, 2\rangle$			$\langle 8, 5\rangle$	
	$-$				$\langle 7, 3\rangle$			

we get the answer $\langle 2\rangle$ with length 1, so $L^+(2,3) = \{\langle 2,1\rangle, \langle 3,2\rangle\}$. Similarly, $L^+(2,7) = \{\langle 2,1\rangle, \langle 3,2\rangle\}$, extended from $\beta^+(2,6) = \{\langle 3,1\rangle\}$, for $b_7 = 2 = a_2$.

For row $i = 3$ ($a_3 = 7$), we have $\beta^+(i,1) = \beta^+(3,1) = \{\langle 7,1\rangle\}$, which comes from $\langle 3,1\rangle \in L^+(2,1)$, and $3 \leq a_3 = 7$. Then, $\beta^+(3,2) = \beta^+(3,1) = \{\langle 7,1\rangle\}$. Next, the sources for $\beta^+(3,3)$ are $L^+(2,3)$ and $\beta^+(3,2)$. From $\langle 3,2\rangle \in L^+(2,3)$, we get $\langle 7,2\rangle$ because $3 \leq 7$. Furthermore, $\langle 7,2\rangle$ dominates $\langle 7,1\rangle \in \beta^+(3,2)$, meaning that $\langle 7,2\rangle$ with length 2 has better extensity than $\langle 7,1\rangle$ with length 1. Thus, $\beta^+(3,3) = \{\langle 7,2\rangle\}$. For $b_4 = 7 = a_3$, $L^+(3,4) = \{\langle 7,3\rangle\}$, extended from $\langle 7,2\rangle \in \beta^+(3,3)$. $\langle 7,2\rangle \in \beta^+(3,3)$ could be extended by $a_3 = 7$ to get a longer answer subsequence $\langle 3,2,7\rangle$. Meanwhile, the third position is a turning point since $t_3 = 1$ and $t_4 = 0$. When $a_3 = 7$ is appended at the third position, it is viewed as both the ending element of the current AI segments and the starting element of the next AD segment. Thus, we get $L^+(3,4) = \{\langle 7,3\rangle\}$, and $L^-(3,4) = \{\langle 7,3\rangle\}$.

For row $i = 4$ ($a_4 = 4$), we have $\beta^+(i,1) = \beta^+(4,1) = \{\langle 4,1\rangle\}$, coming from $\langle 3,1\rangle \in L^+(3,1)$, and $3 \leq a_4 = 4$. Then, $\beta^+(4,2) = \beta^+(4,1) = \{\langle 4,1\rangle\}$. For calculating $\beta^+(4,3)$, $\langle 4,1\rangle$ come from $\beta^+(4,2)$, and $\langle 4,2\rangle$ comes from $\langle 3,2\rangle \in L^+(3,3)$ because $3 \leq 4$. However, $\langle 4,1\rangle$ is dominated by $\langle 4,2\rangle$. Thus, $\beta^+(4,3) = \{\langle 4,2\rangle\}$. Next, $\beta^-(4,4) = \{\langle 4,3\rangle\}$, coming from $\langle 7,3\rangle \in L^-(3,4)$, and $7 \geq 4$. Note that the base $4 = \min\{7, a_4 = 4\}$. For $b_5 = 4 = a_4$, $L^+(4,5) = \{\langle 4,3\rangle\}$, extended from $\langle 4,2\rangle \in \beta^+(4,4)$, and $L^-(4,5) = \{\langle 4,4\rangle\}$, extended from $\langle 4,3\rangle \in \beta^-(4,4)$.

TABLE III
THE POTENTIAL LCAWST SOLUTIONS OF TABLE I WITH $A = \langle 3, 2, 7, 4, 5, 2, 8\rangle$, $B = \langle 3, 8, 2, 7, 4, 5, 2\rangle$, $T = \langle 0, 1, 1, 0, 0, 1, 1\rangle$ AND $c = 3$, WHOSE LCAWST ANSWER IS $\langle 3, 2, 7, 5, 2\rangle$ WITH LENGTH 5. SYMBOLS $+$ AND $-$ CORRESPOND TO AN AI SEGMENT AND AN AD SEGMENT, RESPECTIVELY. A BLANK CELL MEANS THAT ALL ELEMENTS OF THIS CELL ARE IDENTICAL TO ITS UPPER CELL. HERE, RED VALUES CORRESPOND TO $p.base$ AND VALUES WITH UNDERLINES EXIST IN THE LAST SEGMENT.

A \ B		$b_1 = 3$	$b_2 = 8$	$b_3 = 2$	$b_4 = 7$	$b_5 = 4$	$b_6 = 5$	$b_7 = 2$
$a_1 = 3$	$+$	$\langle \underline{\mathbf{3}} \rangle$						
	$-$							
$a_2 = 2$	$+$			$\langle \underline{\mathbf{2}} \rangle$ $\langle \underline{\mathbf{3}, 2} \rangle$				$\langle \underline{\mathbf{2}} \rangle$ $\langle \underline{\mathbf{3}, 2} \rangle$
	$-$							
$a_3 = 7$	$+$				$\langle \underline{3, 2, \mathbf{7}} \rangle$			
	$-$				$\langle 3, 2, \underline{\mathbf{7}} \rangle$			
$a_4 = 4$	$+$					$\langle \underline{3, 2, \mathbf{4}} \rangle$		
	$-$					$\langle 3, 2, \underline{7, \mathbf{4}} \rangle$		
$a_5 = 5$	$+$						$\langle 3, 2, 7, 4, \underline{\mathbf{5}} \rangle$	
	$-$						$\langle 3, 2, \underline{7, \mathbf{5}} \rangle$ $\langle 3, 2, \underline{7, \mathbf{4}, 5} \rangle$	
$a_6 = 2$	$+$			$\langle \underline{\mathbf{2}} \rangle$ $\langle \underline{\mathbf{3}, 2} \rangle$				$\langle 3, 2, 7, 5, \underline{\mathbf{2}} \rangle$ $\langle \underline{\mathbf{3}, 2, 2} \rangle$
	$-$							$\langle 3, 2, \underline{7, 5, \mathbf{2}} \rangle$
$a_7 = 8$	$+$		$\langle \underline{3, \mathbf{8}} \rangle$					
	$-$							

For $5 \le i \le 7$, $\beta^+(i, j)$, $\beta^-(i, j)$, $L^+(i, j)$ and $L^-(i, j)$ are updated similarly. Note that $\langle 4, 3\rangle \in L^+(4, 5)$, but $\langle 4, 3\rangle$ is not inserted into $\beta^+(5, 5)$ because the third position is the turning point. That is, the trend of third to fourth position is AD. However, $L^+(5, j)$ cannot store the potential answer, whose last segment is AD, so $\langle 4, 3\rangle$ is not inserted into $\beta^+(5, 5)$. Therefore, $\beta^+(5, 5) = \{\langle 5, 2\rangle\}$. For $\langle 2, 5\rangle \in L^+(6, 7)$, the fifth position is the turning point, where $a_6 = 2$ is appended as the starting point in the third AI segment.

Finally, we get the LCaWSt with length 5 because $\langle 2, 5\rangle$ has the maximal length in $L^+(7, j)$. We can get the answer sequence by tracing the L-table back, where $\langle 2, 5\rangle$, $\langle 4, 4\rangle$, $\langle 7, 3\rangle$, $\langle 3, 2\rangle$ and $\langle 3, 1\rangle$ are key matching 2-tuple elements. Therefore, one of LCaWSt is $\langle 3,\ 2,\ 7,\ 5,\ 2\rangle$, where $s_1 = \langle 3,\ 2,\ 7\rangle$ is an AI segment and $s_2 = \langle 7,\ 5,\ 2\rangle$ is an AD segment.

IV. CONCLUSION

This paper discusses the longest common almost wave subsequence with the trend (LCaWSt) problem, a variant of the longest common wave subsequence (LCWS) problem. An algorithm is proposed for finding the LCaWSt in O(mnc) time and O(nc) space, where c denotes the constant of tolerance, m and n denote the lengths of the two input numeric sequences, respectively. The LCaWSt considers a certain tolerance to follow a general trend or direction, which deserves to find more applications in our lives.

For detailed analysis of time and space complexities, correctness proof, and possible extension, the readers can refer to [11].

REFERENCES

[1] M. R. Alam and M. S. Rahman, "A divide and conquer approach and a work-optimal parallel algorithm for the LIS problem," *Information Processing Letters*, vol. 113, no. 13, pp. 470–476, 2013.

[2] S. Bespamyatnikh and M. Segal, "Enumerating longest increasing subsequences and patience sorting," *Information Processing Letters*, vol. 76, no. 1-2, pp. 7–11, 2000.

[3] M. T. H. Bhuiyan, M. R. Alam, and M. S. Rahman, "Computing the longest common almost-increasing subsequence," *Theoretical Computer Science*, vol. 930, pp. 157–178, 2022.

[4] Y.-C. Chang and C.-B. Yang, "The longest common wave subsequence problem," in *The 38th Workshop on Combinatorial Mathematics and Computation Theory*, Taipei, Taiwan, 2021, pp. 9–17.

[5] G.-Z. Chen and C.-B. Yang, "The longest wave subsequence problem: Generalizations of the longest increasing subsequence problem," in *The 37th Workshop on Combinatorial Mathematics and Computation Theory*, Kaohsiung, Taiwan, 2020, pp. 28–33.

[6] Y.-A. Chiou, J.-Y. Syu, S.-Y. Wu, L.-Y. Lin, L. T. Yi, T.-T. Lin, and S.-F. Lin, "Electrocardiogram lead selection for intelligent screening of patients with systolic heart failure," *Scientific Reports*, vol. 11, no. 1:1948, 2021.

[7] M. Crochemore and E. Porat, "Fast computation of a longest increasing subsequence and application," *Information and Computation*, vol. 208, no. 9, pp. 1054–1059, 2010.

[8] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg, "Alignment of whole genomes," *Nucleic Acids Research*, vol. 27, no. 11, pp. 2369–2376, 1999.

[9] A. Elmasry, "The longest almost-increasing subsequence," *Information Processing Letters*, vol. 110, no. 16, pp. 655–658, 2010.

[10] S.-C. Lin, "The longest almost wave subsequence problem," in *The 40th Workshop on Combinatorial Mathematics and Computation Theory*, Taoyuan, Taiwan, 2023.

[11] T.-L. Lu, "The longest common almost wave subsequence problem," in *Master's Thesis, Department of Computer Science and Engineering, National Sun Yat-sen University*, Kaohsiung, Taiwan, 2023.

[12] J. M. Moosa, M. S. Rahman, and F. T. Zohora, "Computing a longest common subsequence that is almost increasing on sequences having no repeated elements," *Journal of Discrete Algorithms*, vol. 20, pp. 12–20, 2013.

[13] C. Schensted, "Longest increasing and decreasing subsequences," *Canadian Journal of Mathematics*, vol. 13, pp. 179–191, 1961.

[14] T. T. Ta, Y.-K. Shieh, and C. L. Lu, "Computing a longest common almost-increasing subsequence of two sequences," *Theoretical Computer Science*, vol. 854, pp. 44–51, 2021.

[15] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM*, vol. 21, no. 1, pp. 168–173, 1974.