

# The Approximate Period Problem

Vladimir Yu. Popov <sup>\*†</sup>

**Abstract**—We show that the approximate period problem is NP-complete for metric and alphabet cardinality 7.

**Keywords:** *approximate periods, computational complexity*

## 1 Introduction

Most of the current work in music information retrieval has been done with monophonic sources. In monophonic music, no new note begins until the current note has finished sounding. The most basic approach to monophonic feature extraction reduces notes to a single dimension. Pitch is extracted and duration is ignored, or vice versa. Both pitch and duration information may be used in the final retrieval system, but as features they are treated separately. Most music information retrieval researchers favor relative measures because a change in tempo or transposition across keys does not significantly alter the music information expressed [1], [2], [3], [4], [5], [6], [7].

Relative pitch has three standard expressions: exact interval, rough contour, and simple contour. Exact interval is the signed magnitude between two contiguous pitches. Simple contour keeps the sign and discards the magnitude. Rough contour keeps the sign and groups the magnitude into a number of equivalence classes. Collectively, this techniques are known as unigrams. Long sequences, or  $n$ -grams, are constructed from an initial sequence of interval or ratio unigrams. A most sophisticated approach to  $n$ -gram extraction is the detection of repeating patterns [8], [9], [10]. String matching  $n$ -gram extraction techniques use notions such as insertions, deletions, and substitutions.

Regularities in experimentally obtained data often reveal important knowledge about the underlying physical system. Regularities in a biological sequence can be used to identify the sequence among other sequences, or to infer information about the evolution of the sequence. The genomes of eukaryotes, i.e. higher order organisms such humans, contain many regularities. Tandem repeats, or

tandem arrays, which are consecutive occurrences of the same string, are the most frequent. For example, the six nucleotides *TTAGGG* appear at the end of every human chromosome in tandem arrays that contain between one and two thousand copies [11]. Finding occurrences of repeated substrings in string is a widely studied problem. In biological sequence analysis searching for tandem repeats is used to reveal structural and functional information.

DNA and protein sequences can be seen as long texts over specific alphabets. Those sequences represent the genetic code of living beings. Searching specific sequences over those texts appeared as a fundamental operation for problems such as looking for given features in DNA chains, or determining how different two genetic sequences were. This was modeled as searching for given “patterns” in a “text”. However, exact searching was of no use for this application, since the patterns rarely matched the text exactly. The genetic sequences of two members of the same species are not identical, they are just very similar. Moreover, searching for exact tandem repeats can be too restrictive because of experimental errors. This gave a motivation to “search allowing errors”.

## 2 Preliminaries

Traditionally the alignment notation has been used to illustrate a comparison between two or more sequences. Given a set of strings

$$X = \{x_1, x_2, \dots, x_k\}$$

on an alphabet  $\Sigma$ , a multiple alignment of  $X$  is a set of strings

$$A = \{A_1, A_2, \dots, A_k\},$$

$|A_1| = |A_2| = \dots |A_k| = n$ , on augmented alphabet  $\Gamma = \Sigma \cup \{\Delta\}$  such that each string  $A_i$  is a copy of  $x_i$  into which  $n - |x_i|$  copies of special symbol  $\Delta$  have been inserted. Symbol  $\Delta$  is called an indel and represents the insertion or deletion of a particular symbol in one string relative to another [12].

A conventional way to measure the approximate similarity between two sequences  $a_1 \dots a_m$  and  $b_1 \dots b_n$  is to

<sup>\*</sup>Ural State University, Department of Mathematics and Mechanics, 620083 Ekaterinburg, RUSSIA Email: Vladimir.Popov@usu.ru

<sup>†</sup>The work partially supported by Grant of President of the Russian Federation MD-1687.2008.9 and Analytical Departmental Program “Developing the scientific potential of high school” 2.1.1/1775.

calculate local transformations or costs of local transformations. Usually the considered local transformations are the following:

- substitution:  $a_i \rightarrow b_j$ ;
- insertion:  $\Delta \rightarrow b_j$ ;
- deletion:  $a_i \rightarrow \Delta$ .

To define a distance between sequences, one should first fix the set of local transformations and non-negative valued cost function  $\delta$  that gives for each transformation  $a \rightarrow b$  a cost  $\delta(a, b)$ . A penalty matrix specifies the substitution cost for each pair of characters and the insertion/deletion cost for each character. The differences appearing in the considered two sequences can be viewed differently, e.g., one substitution can be viewed as one insertion and one deletion. Therefore, it is natural to observe the minimum number of such differences. The weighted edit distance between  $x$  and  $y$  is the minimum cost to convert  $x$  to  $y$  using a penalty matrix. The general framework of the edit distance and its many variations is given in [13].

### 3 Problem Definition

We consider the notion of approximate periods which is an approximate version of periods. This notion is first discussed in [14]. Let  $\delta$  be a distance function which specified by a penalty matrix. Given two strings  $x, p$  and distance function  $\delta$ , we define approximate periods as follows. If there exists a partition of  $x$  into disjoint blocks of substrings, i.e.,  $x = p_1 \dots p_r$ ,  $p_i \neq \epsilon$ , such that  $\delta(p, p_i) \leq t$  for  $1 \leq i < r$ , and  $\delta(p', p_i) \leq t$  where  $p'$  is some prefix of  $p$ , we say that  $p$  is a  $t$ -approximate period of  $x$  (see [14]).

Let us consider the following problem:

THE APPROXIMATE PERIOD PROBLEM (AP)

INSTANCE: A finite alphabet  $\Gamma$ , a string  $x$  from  $\Gamma^*$ , a penalty matrix  $M$ , and a positive integer  $t$ .

QUESTION: Is there a string  $u \in \Gamma^*$  such that  $u$  is a  $t$ -approximate period of  $x$ ?

### 4 The Main Result

If  $|\Gamma| \geq 9$  then there exists  $\delta$  such that  $\delta(a, a) = 0$ ,  $\delta(a, b) = \delta(b, a)$  for all  $a, b \in \Gamma$ , and AP problem is NP-complete [14]. It is shown in [15] that AP problem is NP-complete for  $|\Gamma| \geq 5$  and penalty matrix  $M$  as in figure 1.

**Theorem.** If  $|\Gamma| \geq 7$  then there exists  $\delta$  such that  $\delta(a, a) = 0$ ,  $\delta(a, b) = \delta(b, a)$  for all  $a, b \in \Gamma$ , and AP problem is NP-complete.

	$A$	$G$	$C$	$T$	$\Delta$
$A$	0	$m$	$d$	$t+1$	$t+1$
$G$	$m$	0	$d$	$t+1$	$t+1$
$C$	$d$	$d$	$2d$	$t+1$	$t+1$
$T$	$t+1$	$t+1$	$t+1$	0	$t+1$
$\Delta$	$t+1$	$t+1$	$t+1$	$t+1$	0

Figure 1: The penalty matrix  $M$

**Proof.** It is easy to see that the AP problem is in NP.

Consider the familiar Hamming distance,  $D$ , where the local transformations are of the form  $a \rightarrow b$  with cost  $D(a, a) = 0$  and  $D(a, b) = 1$ , for  $a \neq b$ . Let  $y_1$  and  $y_2$  be finite strings. Let us consider the following problem:

THE CLOSEST STRING PROBLEM (CS)

Instance: A finite alphabet  $\Sigma$ , a set  $S = \{s_1, s_2, \dots, s_n\}$  of strings each of length  $m$ ,  $S \subseteq \Sigma^*$ , and a positive integer  $d$ .

Question: Is there a string  $s$  of length  $m$  such that for every string  $s_i \in S$ ,  $D(s, s_i) \leq d$ ?

The CS problem is NP-complete even for the restriction to a binary alphabet [16], [17]. We will assume that  $\Sigma = \{0, 1\}$ . Now we transform an instance of the CS problem to an instance of the AP problem as follows.

- $\Gamma = \Sigma \cup \{2, 3, 4, T, \Delta\}$
- $x = T2^m T2^3 m T2^2 2^m T2^2 2^m T2^3 m T2^3 m T2^4 m T2^2 s_1 T2^2 s_2 T2^2 s_3 T \dots T s_n T$
- $t = md$ . Note that we can assume that  $d > \frac{m}{2}$ .
- Define the penalty matrix  $M$  as in figure 2.

	0	1	2	3	4	$T$	$\Delta$
0	0	$m$	$d$	$d$	$d$	$t+1$	$t+1$
1	$m$	0	$d$	$d$	$d$	$t+1$	$t+1$
2	$d$	$d$	0	$2d$	$2d$	$t+1$	$t+1$
3	$d$	$d$	$2d$	0	$2d$	$t+1$	$t+1$
4	$d$	$d$	$2d$	$2d$	0	$t+1$	$t+1$
$T$	$t+1$	$t+1$	$t+1$	$t+1$	$t+1$	0	$t+1$
$\Delta$	$t+1$	$t+1$	$t+1$	$t+1$	$t+1$	$t+1$	0

Figure 2: The penalty matrix  $M$

It is easy to see that this transformation can be done in polynomial time.

Let us show that if there is a string  $u$  such that  $u$  is a  $t$ -approximate period of  $x$ , then  $u = Tu'T$  where  $u'$  is a string in alphabet  $\{0, 1\}$ .

Let us consider a partition of  $x$  into disjoint blocks of

substrings:  $x = p_1 \dots p_r$ .

First suppose that  $u$  has no  $T$ . Clearly, there exists a partition block of  $x$  which has at least one  $T$ , and the distance between  $u$  and the partition block is greater than  $t$ . Therefore,  $u$  must have at least one  $T$ . Suppose that  $u$  has no more than one  $T$ . In this case,  $u = u'(0, 1, 2, 3, 4, \Delta)Tu''(0, 1, 2, 3, 4, \Delta)$ . Consider the alignment of  $p_1, \dots, p_r$  induced by  $u$ . It is easy to see that  $p_1 = p'_1(\Delta)Tp''_1(2, \Delta)$ . It is clear that either  $\delta(u, p_1) > t$  or

$$\delta(u'(0, 1, 2, 3, 4, \Delta), p'_1(\Delta)) + \delta(u''(0, 1, 2, 3, 4, \Delta), p''_1(2, \Delta)) \leq t. \quad (1)$$

Since  $u$  is a  $t$ -approximate period of  $x$ ,  $\delta(u, p_1) \leq t$ . In view of (1),

$$\delta(u'(0, 1, 2, 3, 4, \Delta), p'_1(\Delta)) \leq t \quad (2)$$

and

$$\delta(u''(0, 1, 2, 3, 4, \Delta), p''_1(2, \Delta)) \leq t.$$

Since  $\delta(\Delta, y) = t + 1$ ,  $y \in \{0, 1, 2, 3, 4\}$ , in view of (2), it is easy to see that

$$u'(0, 1, 2, 3, 4, \Delta) \in \{\Delta\}^*. \quad (3)$$

Since  $p_1 = p'_1(\Delta)Tp''_1(2, \Delta)$ ,  $p_2 = p'_2(2, \Delta)Tp''_2(\Delta)$ . It is clear that either  $\delta(u, p_2) > t$  or

$$\delta(u'(0, 1, 2, 3, 4, \Delta), p'_2(2, \Delta)) + \delta(u''(0, 1, 2, 3, 4, \Delta), p''_2(\Delta)) \leq t. \quad (4)$$

Since  $u$  is a  $t$ -approximate period of  $x$ ,  $\delta(u, p_2) \leq t$ . Therefore, in view of (4),

$$\delta(u'(0, 1, 2, 3, 4, \Delta), p'_2(2, \Delta)) \leq t \quad (5)$$

and

$$\delta(u''(0, 1, 2, 3, 4, \Delta), p''_2(\Delta)) \leq t. \quad (6)$$

Since  $\delta(\Delta, y) = t + 1$ ,  $y \in \{0, 1, 2, 3, 4\}$ , in view of (3) and (5),  $p'_2(2, \Delta) \in \{\Delta\}^*$ . Similarly, in view of (6),

$u''(0, 1, 2, 3, 4, \Delta) \in \{\Delta\}^*$  and  $p''_1(2, \Delta) \in \{\Delta\}^*$ . Since  $p'_1(2, \Delta) \in \{\Delta\}^*$ ,  $p'_2(2, \Delta) \in \{\Delta\}^*$  and  $2^m$  is a subsequence of  $p'_1(2, \Delta)p'_2(2, \Delta)$ ,  $u$  must have at least two  $T$ .

Now suppose that

$$u = u'(0, 1, 2, 3, 4, \Delta)Tu''(0, 1, 2, 3, 4, \Delta)T \dots \dots u^{(k)}(0, 1, 2, 3, 4, \Delta)Tu^{(k+1)}(0, 1, 2, 3, 4, \Delta).$$

It is easy too see that in this case

$$p_1 = p'_1Tp''_1T \dots p_1^{(k)}Tp_1^{(k+1)},$$

where  $p'_1, p''_1, \dots, p_1^{(k)}, p_1^{(k+1)} \in \{0, 1, 2, 3, 4, \Delta\}^*$ . Since  $u$  is a  $t$ -approximate period of  $x$ ,  $\delta(u, p_1) \leq t$ . Therefore,

$$\begin{aligned} & \delta(u'(0, 1, 2, 3, 4, \Delta), p'_1) + \\ & + \delta(u''(0, 1, 2, 3, 4, \Delta), p''_1) + \dots \\ & + \delta(u^{(k)}(0, 1, 2, 3, 4, \Delta), p_1^{(k)}) + \\ & + \delta(u^{(k+1)}(0, 1, 2, 3, 4, \Delta), p_1^{(k+1)}) \leq t. \end{aligned} \quad (7)$$

In view of (7),

$$\begin{aligned} & \delta(u'(0, 1, 2, 3, 4, \Delta), p'_1) \leq t, \\ & \delta(u''(0, 1, 2, 3, 4, \Delta), p''_1) \leq t, \\ & \dots \\ & \delta(u^{(k)}(0, 1, 2, 3, 4, \Delta), p_1^{(k)}) \leq t, \\ & \delta(u^{(k+1)}(0, 1, 2, 3, 4, \Delta), p_1^{(k+1)}) \leq t. \end{aligned} \quad (8)$$

Suppose that  $k = 2p + 1$  where  $p \geq 1$ . In this case

$$p_2 = p'_2Tp''_2T \dots p_2^{(k)}Tp_2^{(k+1)},$$

where  $p''_2 \in \{\Delta\}^*$ . It is easy to see that  $p'_1 \in \{2, \Delta\}^*$ , and  $2^m$  is a subsequence of  $p'_1$ . Since  $\delta(\Delta, 2) = t + 1$ , In view of (8), it is easy to check that  $u''(0, 1, 2, 3, 4, \Delta) \notin \{\Delta\}^*$ . Since  $\delta(u, p_2) \leq t$ ,

$$\begin{aligned} & \delta(u'(0, 1, 2, 3, 4, \Delta), p'_2) + \\ & + \delta(u''(0, 1, 2, 3, 4, \Delta), p''_2) + \dots \\ & + \delta(u^{(k)}(0, 1, 2, 3, 4, \Delta), p_2^{(k)}) + \\ & + \delta(u^{(k+1)}(0, 1, 2, 3, 4, \Delta), p_2^{(k+1)}) \leq t. \end{aligned} \quad (9)$$

In view of (9),

$$\delta(u''(0, 1, 2, 3, 4, \Delta), p_2'') \leq t.$$

Since  $k = 2p + 1$ , it is easy to check that  $p_2'' \in \{\Delta\}^*$ . In view of  $\delta(\Delta, y) = t + 1$ ,  $y \in \{0, 1, 2, 3\}$ ,  $u''(0, 1, 2, 3, 4, \Delta) \in \{\Delta\}^*$ . Therefore,  $k \neq 2p + 1$ .

Suppose that  $k = 2p$  where  $p \geq 7$ . It is easy to see that in this case

$$p_1 = W_1(\Delta)T(2^m)_\Delta TW_2(\Delta)T(3^m)_\Delta TW_3(\Delta)T(2^m)_\Delta$$

$$TW_4(\Delta)T(2^m)_\Delta TW_5(\Delta)T(3^m)_\Delta TW_6(\Delta)T(3^m)_\Delta$$

$$TW_7(\Delta)T(4^m)_\Delta TW_8(\Delta)T(s_1)_\Delta TW_9(\Delta)T(s_2)_\Delta T \dots$$

$$\dots T(s_{p-7})_\Delta TW_{p+1}(\Delta),$$

where  $W_j(\Delta) \in \{\Delta\}^*$ ,  $1 \leq j \leq p + 1$ ,

$$(Y)_\Delta = \Delta^{q_1} Y_1 \Delta^{q_2} Y_2 \dots \Delta^{q_r} Y_r \Delta^{q_{r+1}},$$

$$Y = Y_1 Y_2 \dots Y_r.$$

Since  $u$  is a  $t$ -approximate period of  $x$ ,  $\delta(u, p_1) \leq t$ . Therefore,

$$\begin{aligned} & \delta(u'(0, 1, 2, 3, 4, \Delta), W_1(\Delta)) + \\ & + \delta(u''(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) + \\ & + \delta(u'''(0, 1, 2, 3, 4, \Delta), W_2(\Delta)) + \\ & + \delta(u^{(4)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) + \\ & + \delta(u^{(5)}(0, 1, 2, 3, 4, \Delta), W_3(\Delta)) + \\ & + \delta(u^{(6)}(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) + \\ & + \delta(u^{(7)}(0, 1, 2, 3, 4, \Delta), W_4(\Delta)) + \\ & + \delta(u^{(8)}(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) + \\ & + \delta(u^{(9)}(0, 1, 2, 3, 4, \Delta), W_5(\Delta)) + \\ & + \delta(u^{(10)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) + \\ & + \delta(u^{(11)}(0, 1, 2, 3, 4, \Delta), W_6(\Delta)) + \\ & + \delta(u^{(12)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) + \\ & + \delta(u^{(13)}(0, 1, 2, 3, 4, \Delta), W_7(\Delta)) + \\ & + \delta(u^{(14)}(0, 1, 2, 3, 4, \Delta), (4^m)_\Delta) + \\ & + \delta(u^{(15)}(0, 1, 2, 3, 4, \Delta), W_8(\Delta)) + \end{aligned}$$

$$\begin{aligned} & + \delta(u^{(16)}(0, 1, 2, 3, 4, \Delta), (s_1)_\Delta) + \\ & + \delta(u^{(17)}(0, 1, 2, 3, 4, \Delta), W_9(\Delta)) + \\ & + \dots + \delta(u^{(k)}(0, 1, 2, 3, 4, \Delta), (s_{p-7})_\Delta) + \\ & + \delta(u^{(k+1)}(0, 1, 2, 3, 4, \Delta), W_{p+1}(\Delta)) \leq t \end{aligned} \quad (10)$$

and

$$\delta(u'(0, 1, 2, 3, 4, \Delta), W_1(\Delta)) \leq t$$

$$\delta(u''(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) \leq t$$

$$\delta(u'''(0, 1, 2, 3, 4, \Delta), W_2(\Delta)) \leq t$$

$$\delta(u^{(4)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) \leq t$$

$$\delta(u^{(5)}(0, 1, 2, 3, 4, \Delta), W_3(\Delta)) \leq t$$

$$\delta(u^{(6)}(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) \leq t$$

$$\delta(u^{(7)}(0, 1, 2, 3, 4, \Delta), W_4(\Delta)) \leq t$$

$$\delta(u^{(8)}(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) \leq t$$

$$\delta(u^{(9)}(0, 1, 2, 3, 4, \Delta), W_5(\Delta)) \leq t$$

$$\delta(u^{(10)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) \leq t$$

$$\delta(u^{(11)}(0, 1, 2, 3, 4, \Delta), W_6(\Delta)) \leq t$$

$$\delta(u^{(12)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) \leq t$$

$$\delta(u^{(13)}(0, 1, 2, 3, 4, \Delta), W_7(\Delta)) \leq t$$

$$\delta(u^{(14)}(0, 1, 2, 3, 4, \Delta), (4^m)_\Delta) \leq t$$

$$\delta(u^{(15)}(0, 1, 2, 3, 4, \Delta), W_8(\Delta)) \leq t$$

$$\delta(u^{(16)}(0, 1, 2, 3, 4, \Delta), (s_1)_\Delta) \leq t$$

$$\delta(u^{(17)}(0, 1, 2, 3, 4, \Delta), W_9(\Delta)) \leq t$$

...

$$\delta(u^{(k)}(0, 1, 2, 3, 4, \Delta), (s_{p-7})_\Delta) \leq t$$

$$\delta(u^{(k+1)}(0, 1, 2, 3, 4, \Delta), W_{p+1}(\Delta)) \leq t \quad (11)$$

Since  $W_j(\Delta) \in \{\Delta\}^*$ ,  $1 \leq j \leq p + 1$ , in view of (11), without loss of generality, we can assume that  $W_j(\Delta)$ ,  $u^{(2^{j-1})}(0, 1, 2, 3, 4, \Delta)$ ,  $1 \leq j \leq p + 1$ , are empty words. In view of (11), it is easy to check that if  $(2^m)_\Delta = Y_1 Y_2 \dots Y_r$  and  $u''(0, 1, 2, 3, 4, \Delta) = Z_1 Z_2 \dots Z_r$ , where  $Y_1, Y_2, \dots, Y_r \in \{2, \Delta\}$ ,  $Z_1, Z_2, \dots, Z_r \in \{0, 1, 2, 3, 4, \Delta\}$ , then

$$Y_i = \Delta \Leftrightarrow Z_i = \Delta, 1 \leq i \leq r.$$

Therefore, without loss of generality, we can assume that  $(2^m)_\Delta \in \{2\}^*$  and  $u''(0, 1, 2, 3, 4, \Delta) \in \{0, 1, 2, 3, 4\}^*$ . Since  $(2^m)_\Delta \in \{2\}^*$ , it is easy to check that  $(2^m)_\Delta = 2^m$ . Similarly, we can assume that  $u^{2^j}(0, 1, 2, 3, 4, \Delta) \in \{0, 1, 2, 3\}^*$ ,  $(3^m)_\Delta = 3^m$ ,  $(4^m)_\Delta = 4^m$ ,  $(s_i)_\Delta = s_i$ , where  $1 < j \leq p$ ,  $1 \leq i \leq p - 2$ .

Let  $occ(y, v)$  denote the number of occurrences of the letter  $y$  in the word  $v$ . Since  $u''(0, 1, 2, 3, 4, \Delta), u^{(4)}(0, 1, 2, 3, 4, \Delta) \in \{0, 1, 2, 3, 4\}^*$

and  $(2^m)_\Delta = 2^m$ ,  $(3^m)_\Delta = 3^m$ , in view of (10), it is easy to see that

$$\begin{aligned} & d \cdot occ(0, u''(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u''(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(3, u''(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u''(0, 1, 2, 3, 4, \Delta), 2^m), \end{aligned}$$

$$\begin{aligned} & d \cdot occ(0, u^{(4)}(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u^{(4)}(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(2, u^{(4)}(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u^{(4)}(0, 1, 2, 3, 4, \Delta), 3^m), \end{aligned}$$

$$\begin{aligned} & d \cdot occ(0, u^{(6)}(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u^{(6)}(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(3, u^{(6)}(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u^{(6)}(0, 1, 2, 3, 4, \Delta), 2^m), \end{aligned}$$

$$\begin{aligned} & d \cdot occ(0, u^{(8)}(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u^{(8)}(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(3, u^{(8)}(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u^{(8)}(0, 1, 2, 3, 4, \Delta), 2^m), \end{aligned}$$

$$\begin{aligned} & d \cdot occ(0, u^{(10)}(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u^{(10)}(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(2, u^{(10)}(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u^{(10)}(0, 1, 2, 3, 4, \Delta), 3^m), \end{aligned}$$

$$\begin{aligned} & d \cdot occ(0, u^{(12)}(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u^{(12)}(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(2, u^{(12)}(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u^{(12)}(0, 1, 2, 3, 4, \Delta), 3^m), \end{aligned}$$

and

$$\sum_{i=1}^6 occ(2, u^{(2i)}(0, 1, 2, 3, 4, \Delta)) + occ(3, u^{(2i)}(0, 1, 2, 3, 4, \Delta)) \geq 5m. \quad (12)$$

It is not hard to check that

$$p_2 = T s_{p-6} T^2 s_{p-5} T^2 \dots s_{2p-8} T^2 s_{2p-7} T.$$

Since  $\delta(a, b) = d$ ,  $a \in \{0, 1\}$ ,  $b \in \{2, 3\}$ , in view of (12),  $\delta(u, p_2) \geq 5t$ . Therefore,  $p < 7$ . Similarly, it is not hard to check that  $p < 3$ .

Suppose that  $p = 2$ . It is easy to see that in this case

$$p_1 = W_1(\Delta)T(2^m)_\Delta TW_2(\Delta)T(3^m)_\Delta TW_3(\Delta).$$

Since  $u$  is a  $t$ -approximate period of  $x$ ,  $\delta(u, p_1) \leq t$ . Therefore,

$$\begin{aligned} & \delta(u'(0, 1, 2, 3, 4, \Delta), W_1(\Delta)) + \\ & + \delta(u''(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) + \\ & + \delta(u'''(0, 1, 2, 3, 4, \Delta), W_2(\Delta)) + \\ & + \delta(u^{(4)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) + \\ & + \delta(u^{(5)}(0, 1, 2, 3, 4, \Delta), W_3(\Delta)) \leq t \end{aligned} \quad (13)$$

and

$$\begin{aligned} & \delta(u'(0, 1, 2, 3, 4, \Delta), W_1(\Delta)) \leq t \\ & \delta(u''(0, 1, 2, 3, 4, \Delta), (2^m)_\Delta) \leq t \\ & \delta(u'''(0, 1, 2, 3, 4, \Delta), W_2(\Delta)) \leq t \\ & \delta(u^{(4)}(0, 1, 2, 3, 4, \Delta), (3^m)_\Delta) \leq t \\ & \delta(u^{(5)}(0, 1, 2, 3, 4, \Delta), W_3(\Delta)) \leq t \end{aligned} \quad (14)$$

Since  $W_j(\Delta) \in \{\Delta\}^*$ ,  $1 \leq j \leq p + 1$ , in view of (14), without loss of generality, we can assume that  $W_j(\Delta)$ ,  $u^{(2j-1)}(0, 1, 2, 3, 4, \Delta)$ ,  $1 \leq j \leq p + 1$ , are empty words. In view of (14), it is easy to check that if  $(2^m)_\Delta = Y_1 Y_2 \dots Y_r$  and  $u''(0, 1, 2, 3, 4, \Delta) = Z_1 Z_2 \dots Z_r$ , where  $Y_1, Y_2, \dots, Y_r \in \{2, \Delta\}$ ,  $Z_1, Z_2, \dots, Z_r \in \{0, 1, 2, 3, 4, \Delta\}$ , then

$$Y_i = \Delta \Leftrightarrow Z_i = \Delta, 1 \leq i \leq r.$$

Therefore, without loss of generality, we can assume that  $(2^m)_\Delta \in \{2\}^*$  and  $u''(0, 1, 2, 3, 4, \Delta) \in \{0, 1, 2, 3, 4\}^*$ . Since  $(2^m)_\Delta \in \{2\}^*$ , it is easy to check that  $(2^m)_\Delta = 2^m$ . Similarly, we can assume that  $u^{2j}(0, 1, 2, 3, 4, \Delta) \in \{0, 1, 2, 3, 4\}^*$ ,  $(3^m)_\Delta = 3^m$ , where  $1 < j \leq p$ .

Since

$$u''(0, 1, 2, 3, 4, \Delta), u^{(4)}(0, 1, 2, 3, 4, \Delta) \in \{0, 1, 2, 3, 4\}^*$$

and  $(2^m)_\Delta = 2^m$ ,  $(3^m)_\Delta = 3^m$ , in view of (13), it is easy to see that

$$\begin{aligned} & d \cdot occ(0, u''(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u''(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(3, u''(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u''(0, 1, 2, 3, 4, \Delta), 2^m), \end{aligned}$$

$$\begin{aligned} & d \cdot occ(0, u^{(4)}(0, 1, 2, 3, 4, \Delta)) + \\ & + d \cdot occ(1, u^{(4)}(0, 1, 2, 3, 4, \Delta)) + \\ & + 2d \cdot occ(2, u^{(4)}(0, 1, 2, 3, 4, \Delta)) = \\ & = \delta(u^{(4)}(0, 1, 2, 3, 4, \Delta), 3^m), \end{aligned}$$

and

$$\begin{aligned} & \sum_{i=1}^2 occ(2, u^{(2i)}(0, 1, 2, 3, 4, \Delta)) + \\ & + occ(3, u^{(2i)}(0, 1, 2, 3, 4, \Delta)) \geq m. \end{aligned}$$

It is not hard to check that

$$p_2 = T2^mT^22^mT, p_3 = T3^mT^23^mT.$$

Therefore,

$$\begin{aligned} & occ(2, u^{(2)}(0, 1, 2, 3, 4, \Delta)) + \\ & + occ(2, u^{(4)}(0, 1, 2, 3, 4, \Delta)) \geq m, \end{aligned}$$

$$\begin{aligned} & occ(3, u^{(2)}(0, 1, 2, 3, 4, \Delta)) + \\ & + occ(3, u^{(4)}(0, 1, 2, 3, 4, \Delta)) \geq m, \end{aligned}$$

and

$$\begin{aligned} & \sum_{i=1}^2 occ(2, u^{(2i)}(0, 1, 2, 3, 4, \Delta)) + \\ & + occ(3, u^{(2i)}(0, 1, 2, 3, 4, \Delta)) \geq 2m. \end{aligned} \quad (15)$$

It is not hard to check that

$$p_5 = Ts_2T^2s_3T.$$

Since  $\delta(a, b) = d$ ,  $a \in \{0, 1\}$ ,  $b \in \{2, 3\}$ , in view of (15),  $\delta(u, p_5) \geq 2t$ . Therefore,  $p = 1$ .

Since  $p = 1$ , it is easy to check that  $u = Tu'T$ , where  $u'$  is a string in alphabet  $\{0, 1, 2, 3, 4\}$ , and  $p_1 = T2^mT$ ,  $p_2 = T3^mT$ ,  $p_7 = T4^mT$ . Clearly,

$$\delta(u, p_1) \leq t, \delta(u, p_2) \leq t, \delta(u, p_7) \leq t.$$

Therefore,

$$\begin{aligned} & d \cdot occ(0, u) + d \cdot occ(1, u) + \\ & + 2d \cdot occ(3, u) + 2d \cdot occ(4, u) \leq t, \\ & d \cdot occ(0, u) + d \cdot occ(1, u) + \\ & + 2d \cdot occ(2, u) + 2d \cdot occ(4, u) \leq t, \\ & d \cdot occ(0, u) + d \cdot occ(1, u) + \\ & + 2d \cdot occ(2, u) + 2d \cdot occ(3, u) \leq t. \end{aligned} \quad (16)$$

In view of (16),

$$\begin{aligned} & occ(0, u) + occ(1, u) + \\ & + 2occ(3, u) + 2occ(4, u) \leq m, \\ & occ(0, u) + occ(1, u) + \\ & + 2occ(2, u) + 2occ(4, u) \leq m, \\ & occ(0, u) + occ(1, u) + \\ & + 2occ(2, u) + 2occ(3, u) \leq m. \end{aligned} \quad (17)$$

In view of (17),

$$\begin{aligned} & 3occ(0, u) + 3occ(1, u) + 4occ(2, u) + \\ & + 4occ(3, u) + 4occ(4, u) \leq 3m. \end{aligned}$$

Since  $u = Tu'T$ , where  $u'$  is a string in alphabet  $\{0, 1, 2, 3, 4\}$ ,

$$\begin{aligned} & occ(0, u) + occ(1, u) + occ(2, u) + \\ & + occ(3, u) + occ(4, u) = m. \end{aligned}$$

Therefore,

$$occ(2, u) + occ(3, u) + occ(4, u) = 0,$$

and  $u = Tu'T$ , where  $u'$  is a string in alphabet  $\{0, 1\}$ .

Since  $u = Tu'T$ , where  $u'$  is a string in alphabet  $\{0, 1\}$ , for every string  $s_i \in S$ ,  $\delta(u', s_i) \leq t$ . In view of  $\delta(0, 1) = \delta(1, 0) = m$ ,  $D(u', s_i) \leq d$ . Therefore,  $s = u'$ .

Now suppose that there is a string  $s$  of length  $m$  such that for every string  $s_i \in S$ ,  $D(s, s_i) \leq d$ . It is easy to see that  $u = TsT$  is  $t$ -approximate period of  $x$ .

## References

- [1] Blackburn, S., DeRoure, D., "A tool for content-based navigation of music," *Proceedings of ACM International Multimedia Conference (ACMMM)*, Bristol, UK, pp. 361-368 9/98.
- [2] Dillon, M., Hunter, M., "Automated identification of melodic variants in folk music," *Computers and the Humanities*, V16, N2 pp. 107-117 6/82.
- [3] Ghias, A., Logan, J., Chamberlin, D., Smith, B., "Query by humming-musical information retrieval in an audio database," *Proceedings of ACM International Multimedia Conference (ACMMM)*, San Francisco, CA, pp. 231-236 11/95.
- [4] Kornstädt, A., "Themefinder: A web-based melodic search tool," *Melodic Comparison: Concepts, Procedure, and Applications, Computing in Musicology*, V11, pp. 231-236 98.
- [5] Lemström, K., Laine, P., Perttu, S., "Using relative interval slope in music information retrieval," *Proceedings of the International Computer Music Conference (ICMC)*, Beijing, China, pp. 317-320 9/99.
- [6] Lindsay, A.T., *Using contour as mid-level representation of melody*, Master's thesis, MIT Media Lab, 1996.
- [7] Selfridge-Field, E., "Conceptual and representational issues in melodie comparison," *Melodic Comparison: Concepts, Procedure, and Applications, Computing in Musicology*, V11, pp. 3-64 98.
- [8] Hsu, J.-L., Liu, C.-C., Chen, A.L.P., "Efficient repeating pattern finding in music databases," *Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management*, Bethesda, Maryland, USA, pp. 281-288 11/98.
- [9] Liu, C.-C., Hsu, J.-L., Chen, A.L.P., "Efficient theme and non-trivial repeating pattern discovering in music databases," *Proceedings 15th International Conference on Data Engineering*, Sydney, Australia, IEEE Computer Society, pp. 14-21 3/99.
- [10] Tseng, Y.-H., "Content-based retrieval for music collections," *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, Berkeley, CA, pp. 176-182 8/99.
- [11] Moyzis, R.K., "The human telomere," *Scientific American*, pp. 48-55 8/91.
- [12] Pevzner, P.A., "Multiple alignment, communication cost, and graph matching," *SIAM Journal on Applied Mathematics*, V52, N6, pp. 1763-1779 12/92.
- [13] Ukkonen, E., "Algorithms for approximate string matching" *Information and Control*, V64, N1-3 pp. 100-118 1-3/85.
- [14] Sim, J.S., Iliopoulos, C.S., Park, K., Smyth, W.F., "Approximate periods of strings," *Theoretical Computer science*, V262, N1-2, pp. 557-568 7/01.
- [15] Popov, V.Yu., "The approximate period problem for DNA alphabet," *Theoretical Computer Science*, V304, N1-3, pp. 443-447 7/03.
- [16] Frances, M., Litman, A., "On covering problems of codes," *Theory of Computer Systems*, V30, N2, pp. 113-119 3/97.
- [17] Lanctot, J.K., Li, M., Ma, B., Wang, S., Zhang, L., "Distinguishing string selection problems," *Proceedings of 10th ACM-SIAM Symposium on Discrete Algorithms*, pp. 633-642 1/99.