Discrete Applied Mathematics 157 (2009) 3586–3592

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

On the approximability of the Simplified Partial Digest Problem

Jacek Blazewicz [a,b,*], Edmund K. Burke [c], Marta Kasprzak [a,b], Alexandr Kovalev [a], Mikhail Y. Kovalyov [d,e]

[a] Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
[b] Institute of Bioorganic Chemistry, Polish Academy of Sciences, Z. Noskowskiego 12/14, 61-704 Poznan, Poland
[c] School of Computer Science, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, UK
[d] Belarusian State University, Nezavisimosti 4, 220030 Minsk, Belarus
[e] United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova 6, 220012 Minsk, Belarus

ARTICLE INFO

Article history:
Received 21 November 2007
Received in revised form 7 April 2009
Accepted 19 April 2009
Available online 21 May 2009

Keywords:
Genome mapping
Simplified Partial Digest
Computational complexity
Approximation

ABSTRACT

In this paper, we analyse the computational complexity of an optimization version of the Simplified Partial Digest Problem (SPDP), which is a mathematical model for DNA mapping based on the results of a simplified partial digest experiment. We prove that recognizing 46.16% of the elements of the DNA map in the error-free simplified partial digest experiment is NP-hard in the strong sense. This implies that the problem of maximizing the number of correct elements of the DNA map in the error-free simplified partial digest experiment is pseudopolynomially non-approximable with the approximation ratio $\rho = \frac{13}{6}$.

1. Introduction

The *Simplified Partial Digest Problem (SPDP)* is a formal model of a biochemical experiment aimed at recognizing DNA linear structure. A DNA molecule can be thought of as a *double helix* consisting of two strands. Watson and Crick [24] showed that there is a strong relationship between these two strands. Another important point to make is the role of linear structure in a DNA molecule and the open challenge of directly determining it. An important existing indirect method involves an operation called *mapping*. In this procedure, a DNA molecule is exposed to specific chemicals called restriction enzymes. These chemicals cut DNA molecules at particular *restriction sites*. The cutting process leads to the loss of any information about the location of the restriction sites. Experimentation can lead the multiset of lengths of the cut fragments given by the number of nucleotides between the corresponding restriction sites.

Restriction site analysis is concerned with reconstructing the location of restriction sites. See Setubal and Meidanis [20], Waterman [23] or Pevzner [19] for details. Inputs for this analysis are the lengths of the cut fragments and information about the cutting (or digesting) method. Examples of cutting methods include *double digest*, which employs two restriction enzymes (see e.g. [23] or [19]), and *partial digest*, which has one enzyme but where there are different reaction times. The inventor of the partial digest approach, Daniel Nathans (see Danna and Nathans [10] and Danna et al. [11]), received the Nobel Prize in 1978 for his work on restriction enzymes and restriction mapping.

* Corresponding author at: Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland. Tel.: +48 61 6653000; fax: +48 618771 525.
E-mail addresses: jblazewicz@cs.put.poznan.pl, akovalev@cs.put.poznan.pl (J. Blazewicz), ekb@cs.nott.ac.uk (E.K. Burke), marta@cs.put.poznan.pl (M. Kasprzak), akovalev@cs.put.poznan.pl (A. Kovalev), koval@newman.bas-net.by (M.Y. Kovalyov).

0166-218X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2009.04.017

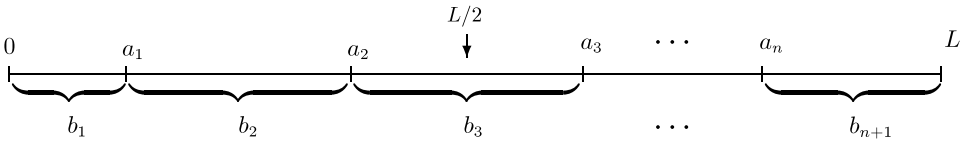

Fig. 1. Graphical interpretation of parameters a_j and b_j.

The *Partial Digest Problem (PDP)* has been studied by several authors see Skiena, Smith, and Lemke [21], Skiena and Sundaram [22], Cieliebak, Eidenbenz and Penna [9], Cieliebak and Eidenbenz [8]. Its main drawback is a lack of experimental data. The output of the partial digestion is a multiset of all *interpoint distances*, where the points are the restriction sites and the ends of the molecule. For n restriction sites, this multiset must consist of $C^2_{n+2} = \frac{(n+1)(n+2)}{2}$ fragment lengths. Pevzner [19] argues that partial digestion has not been widely employed in biological laboratories because of the difficulty in obtaining fragments between every pair of sites. This is confirmed by statistical data where small scale experiments have been reported, see Dudez et al. [13], Keis et al. [17] and Kuwahara et al. [18]. From the point of view of computational complexity theory [14] the status of PDP is yet to be established. The problem was proved to be NP-hard in the cases of measurement errors [8] and noisy data [9]. However, a proof of NP-hardness of the original error-free PDP and a polynomial-time solution algorithm do not currently exist, see Daurat, Gerard and Nivat [12].

A *simplified partial digest method* was first proposed by Blazewicz et al. [5] to address the drawbacks of the partial digest approach outlined above. It has been further studied in [3,4,6,7]. We reproduce a description of this problem here for completeness. In this simplified method, one enzyme is used on two sets of clones of the same DNA molecule. The corresponding experiment consists of two parts. In the first part, the time of the chemical reaction is very short and chosen so that target cloned molecules of the first set are cut at one restriction site at most. In the second part, the reaction time span is long enough to cut the cloned molecules of the second set at all restriction sites. This simplified approach reduces the number of reactions and presents an easier choice of the reaction times.

We study a mathematical model for genome mapping based on simplified partial digest. The model was presented in [3] but is reproduced for completeness. From the first and the second parts of this experiment, multisets A and B, respectively, of molecule fragment lengths are obtained. Let L be the length of the target DNA molecule and let $1, \ldots, n$ be the restriction sites to be recognized by the used enzyme in this molecule. We assume that the experiment is error-free such that the multiset A comprises n pairs $\{a_j, L - a_j\}$ of positive numbers, which we call *end distances*, where a_j is the length of the fragment including one specified end of the molecule and restriction site j, and $L - a_j$ is the length of the complementary fragment, $j = 1, \ldots, n$. Furthermore, multiset B comprises $n + 1$ numbers b_j, which we call *interpoint distances*, where b_j is the length of a fragment between two adjacent points, i.e., restriction sites and the ends of the molecule, see the graphical interpretation in Fig. 1.

The error-free *Simplified Partial Digest Problem (SPDP)* can be formulated in terms of number theory as follows. There is an interval $[0, L]$, a positive integer number n and two multisets A and B of positive integer numbers such that

$$A = \{\{a_j, L - a_j\} \mid j = 1, \ldots, n\}, \qquad B = \left\{ b_j \mid j = 1, \ldots, n+1, \sum_{j=1}^{n+1} b_j = L \right\}.$$

Multiset A contains at most two identical pairs $\{a_j, L - a_j\}$. Identical pairs, if they exist, correspond to the restriction sites that are symmetric with respect to the middle of the molecule. Each pair $\{a_j, L - a_j\}$ can be ordered as $(a_j, L - a_j)$ or $(L - a_j, a_j)$. Assume that each such ordered pair (p_j, s_j) is associated with a point $p_j \in [0, L]$ so that p_j and $s_j = L - p_j$ are the distances between points 0 and p_j and between points p_j and L, respectively. The multisets A and B satisfy the property that each pair $\{a_j, L - a_j\}, j = 1, \ldots, n$, can be ordered so that the associated points $p_j, j = 1, \ldots, n$, partition the interval $[0, L]$ into $n + 1$ subintervals with interpoint distances constituting the multiset B, i.e., $\{p_{j+1} - p_j \mid j = 0, 1, \ldots, n\} = B$, where $p_0 = 0$ and $p_{n+1} = L$.

The goal of the error-free SPDP (in its search version) is to find a sequence of points $p^* = (0, p^*_1, \ldots, p^*_n, L)$ such that

$$\{\{p^*_j, L - p^*_j\} \mid j = 1, \ldots, n\} = A \quad \text{and} \quad \{p^*_{j+1} - p^*_j \mid j = 0, 1, \ldots, n\} = B,$$

where $p^*_0 = 0$ and $p^*_{n+1} = L$.

There may exist several solutions of SPDP. One can be interested in finding at least one or all mutually *non-congruent* solutions. Two solutions are called congruent if they are mirror images of each other. At least one solution of SPDP always exists. It corresponds to the map of the original DNA.

The reported algorithmic results for SPDP include the following. Blazewicz and Kasprzak [7] proved that SPDP is NP-hard in the strong sense and presented an $O(n \log n)$ time algorithm for the case where $b_j \in \{1, 2\}, j = 1, \ldots, n + 1$. Abrams and Chen [1] claim that SPDP is APX-complete by presenting a reduction from the Tripartite-matching problem. However, the proof of their claim does not appear in their short paper. In this paper, unaware of their results, we provide our own proof of the inapproximability of SPDP. Enumerative algorithms for SPDP were proposed by Blazewicz et al. [5], Blazewicz and Jaroszewski [6] and Blazewicz et al. [3]. Blazewicz et al. [3] also presented a dynamic programming algorithm with $O(n^{2q})$ running time for SPDP with q distinct interpoint distances. They gave an example of the problem, in which the maximum

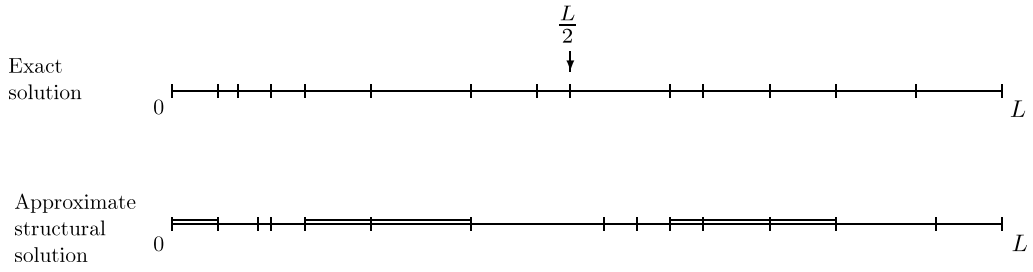

Fig. 2. Exact and approximate solutions to SPDP.

number of mutually non-congruent solutions is equal to $2^{\frac{n+2}{3}-1}$ and provided a suite of computational experiments with 3710 real instances of SPDP taken from GenBank [15] to establish the number of non-congruent solutions. It was equal to 1, 2 and 4 for 3641 (98.14%), 64 (1.725%) and 5 (0.135%) instances, respectively, and it was never equal to 3 or exceeded 4. Blazewicz et al. [4] considered optimization versions of SPDP, denoted as SPDP-Min and SPDP-Max. The goal of SPDP-Min (SPDP-Max) is to find the orderings of the pairs from the multiset A such that the multiset of the corresponding interpoint distances B^0 (being a result of this ordering) contains the minimum (maximum) number of interpoint distances not from the multiset B (from the multiset B). It was proved that SPDP-Min cannot be approximated in pseudo-polynomial time with any finite *approximation ratio* ρ, while SPDP-Max cannot be approximated in pseudo-polynomial time with $\rho < 1 + \frac{1}{n}$, unless $\mathcal{P} = \mathcal{NP}$. Here $\max\left\{\frac{f^0}{n+1}, \frac{n+1}{f^0}\right\} \leq \rho$ for any problem instance, where f^0 is the value of the solution delivered by the approximation algorithm. An $O(n \log n)$ time algorithm with $\rho = 1 + \frac{n}{n+2}$ and several heuristic algorithms based on a graph-theoretic model were presented for SPDP-Max. In the experiments provided, two of the proposed heuristic algorithms always delivered an exact solution for real instances of SPDP from GenBank [15] with $20 \leq n \leq 50$, and for randomly generated instances with $n = 50$.

Since SPDP is NP-hard in the strong sense, we are interested in finding its approximate solution which contains correct parts of the DNA map. Note that an approximate solution of SPDP-Min or SPDP-Max does not guarantee that any of its interpoint fragments is located properly in the corresponding DNA map. In this paper, we show that determining more than $\frac{6}{13}$, i.e., 0.4616 part of the DNA map in the simplified partial digest experiment is computationally difficult.

We call an *interval* a part of the DNA between two adjacent restriction sites including these restriction sites. An interval is specified by the positions of its ends. In this paper, we assume that an approximate solution we wish to construct, specifies a certain number of (correct) intervals which are all present in some exact solution of the considered instance of SPDP. We call such a solution a *structural* solution. It presents a partial structure of a DNA. An example of an approximate structural solution and corresponding exact solution are given in Fig. 2. Correct intervals are indicated by double lines.

Note that an approximate structural solution may contain more correct intervals than it indicates but only the specified correct intervals are guaranteed to be present in some exact solution of SPDP.

We evaluate the quality of an approximate solution by the ratio $\Delta = \frac{n_0}{n+1}$, where n_0 is the number of intervals specified as correct and $n + 1$ is the total number of intervals (the cardinality of the multiset B). For the solution in Fig. 2, $n = 13$ and $\Delta = \frac{6}{14}$. We will prove that there exists no pseudo-polynomial time algorithm which delivers an approximate structural solution to any instance of SPDP with the ratio $\Delta > \frac{6}{13}$, unless $\mathcal{P} = \mathcal{NP}$. We have $\frac{6}{13} \times 100 < 46.16$. Therefore, recognizing 46.16% of the DNA map in the error-free simplified partial digest experiment is NP-hard in the strong sense. Using the standard terminology of computational complexity for optimization problems (see Ausiello et al. [2]), we can see that the problem of maximizing the number of correct intervals in the error-free simplified partial digest experiment is pseudopolynomially non-approximable with the approximation ratio $\rho = \frac{13}{6}$. This further denies the existence of a *polynomial-time approximation scheme* (PTAS) to the problem.

2. Idea of the proof

The basic idea behind our proof can be outlined as follows. We will construct a special case of SPDP, in which there are $2K$ *symmetric* points each of which has its mirror image with respect to the middle of the molecule (including points 0 and L) and the point $L/2$, see Fig. 3. Here, K is part of the problem instance (not a constant). The distance between any two adjacent symmetric points is equal to the same number D. Thus, $L = 2DK$.

We call a set of intervals between two adjacent symmetric points and intervals between mirror images of these points a *block* and we number the blocks $1, 2, \ldots, K$ from the ends of the molecule to the middle. We set the number of intervals in any block to be equal to 13. Thus, $n + 1 = 13K$. We call this special case *SPDP-Blocks* and assume that it always has a solution like the general SPDP. In Section 3, we will prove that SPDP-Blocks (in its search version) is NP-hard in the strong sense. This implies that the general error-free SPDP is NP-hard in the strong sense. Our proof is easier than the one in [7].

The problem SPDP-Blocks has the following property. Assume that there is an algorithm which gives us all 13 correct intervals in some block i. Construct an *i-modified* problem SPDP-Blocks by removing lengths of the above mentioned correct

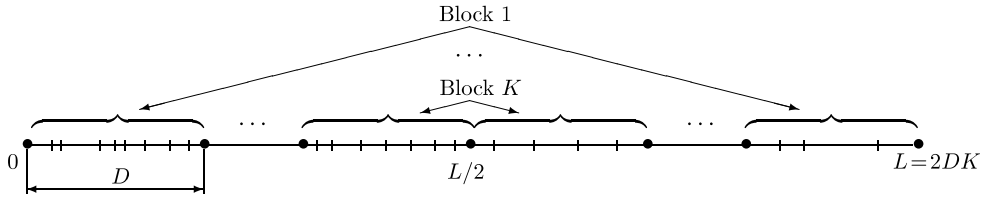

Fig. 3. Special case with K blocks and 13 intervals in each block. Symmetric points are bold circles.

intervals from the multiset B and adjusting end distances in the multiset A as follows. Consider end distances $a \in A$ and $L - a \in A$ such that $a \leq L/2$. Let e_1 and e_2 be the endpoints of block i such that $e_1 < e_2 \leq L/2$, $e_2 - e_1 = D$. If $a \leq e_1$, then replace end distance $L - a$ by $L - a - 2D$. If $e_1 < a \leq e_2$, then remove a and $L - a$ from A. If $e_2 < a \leq L/2$, then replace end distance a by $a - D$ and end distance $L - a$ by $L - a - D$. Finally, in the i-modified problem we re-set $L := L - 2D$.

Proposition 1. *The i-modified problem SPDP-Blocks has an exact solution and any such solution can be extended to an exact solution of the original problem SPDP-Blocks by inserting into appropriate positions intervals of block i.*

Proof. The first part of the above statement is satisfied because an exact solution of the i-modified problem SPDP-Blocks can be obtained by removing intervals of block i from any exact solution of the original problem SPDP-Blocks and filling the obtained gaps by shifting the remaining points to the left. The correctness of the second part can easily be verified by inserting intervals of block i so that the left part of this block follows the left part of block $i - 1$ (or starts at point 0 if $i = 1$) and the right part of this block precedes the right part of block $i + 1$ (or ends at point L if $i = K$). □

Due to Proposition 1, if there exists an algorithm which determines all 13 intervals in at least one block correctly, then at most K iterative applications of this algorithm are needed to find an exact solution of the original search problem SPDP-Blocks.

Now assume that there exists a pseudo-polynomial time algorithm, denoted as A_0, which always provides $\Delta > \frac{6}{13}$ for SPDP-Blocks. Then A_0 outputs more than $13K \times \frac{6}{13} = 6K$ correct intervals. If A_0 outputs at most 6 correct intervals for each block, then the total number of correct intervals does not exceed $6K$. Therefore, it should output at least 7 correct intervals for at least one block. In Section 3, we will show that for any block, if any 7 of its intervals are determined correctly, then the remaining 5 of them can be determined correctly in constant time. Therefore, all 13 intervals for at least one block can be determined in pseudo-polynomial time. This fact and Proposition 1 show that the search problem SPDP-Blocks could be solved in pseudo-polynomial time, what would contradict its NP-hardness in the strong sense. We deduce that recognizing more than $\frac{6}{13}$-th part of the DNA map in the simplified partial digest experiment is NP-hard in the strong sense.

3. Proving NP-hardness

A problem to be proved NP-hard is often formulated as a question asking about an existence of its solution which satisfies a given property. In the case of a *decision version* of SPDP-Blocks, a question could be whether there exists a DNA map with the given multisets A and B. The answer to this question is always "yes" (it follows from the assumption that the digestion experiment was error-free). Hence, the corresponding decision problem cannot be NP-hard. However, SPDP-Blocks is an example of a *search problem*, which is *to find* a solution that satisfies a given property while knowing that such a solution exists. Johnson [16] gives an explanation that a (strong) NP-hardness of a decision problem, for which the existence of solution satisfying a given property is not known, implies a (strong) NP-hardness of its search version, for which the existence of such solution is known *a priori*.

For our purposes, we introduce a decision problem *Quasi-SPDP-Blocks*, which differs from the decision version of the original problem SPDP-Blocks in that the multisets A and B do not necessarily contain numbers coming from the error-free digesting experiment, i.e., the answer to Quasi-SPDP-Blocks is not always "yes". Following Johnson [16], strong NP-hardness of Quasi-SPDP-Blocks implies strong NP-hardness of the original search problem SPDP-Blocks.

Theorem 2. *Quasi-SPDP-Blocks is NP-hard in the strong sense.*

Proof. We construct a pseudo-polynomial transformation to Quasi-SPDP-Blocks of the strongly NP-complete problem Exact Cover by 3-sets (X3C) (see Garey and Johnson [14]).

Exact Cover by 3-Sets (X3C) can be defined as follows: Given a family $C = \{C_1, \ldots, C_k\}$ of 3-element subsets of the set $M = \{1, \ldots, 3m\}$, does C contain an exact cover of M, i.e., a subfamily $Y \subseteq C$ such that each $j \in M$ belongs to exactly one 3-element set in Y?

Given an instance of X3C, we construct the following instance of Quasi-SPDP-Blocks. Let r_j be the number of times that element $j \in M$ appears in the 3-element subsets of the family C. Firstly, calculate the numbers r_j, $j \in M$. It can be done in $O(mk)$ time. Calculate number $E = (3m + 1)^2$.

We generate an instance of Quasi-SPDP-Blocks as follows. Set $D = 12E$, $K = k$, $n = 13K - 1$ and $L = 2DK$. For each $C_i \in C$, $i < k$, generate two end distances iD and two end distances $L - iD$, which correspond to the pair of symmetric points

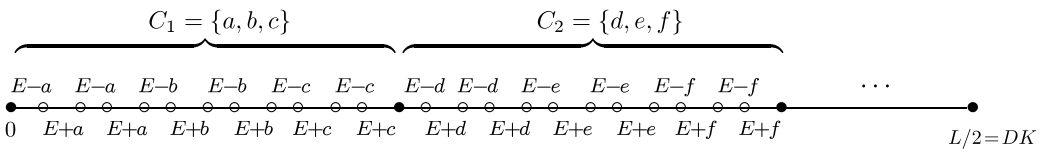

Fig. 4. Graphical interpretation of an instance of Quasi-SPDP-Blocks.

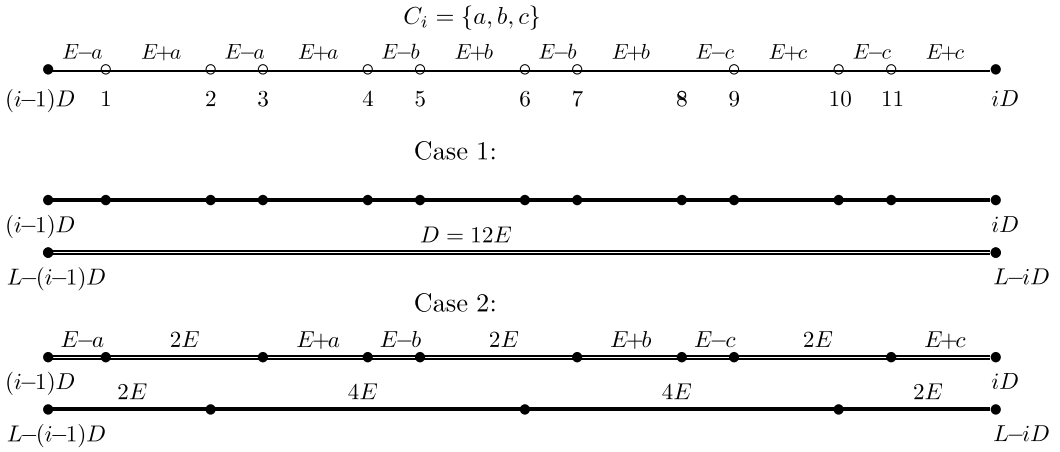

Fig. 5. Two cases for block *i*. In each case, upper and lower pictures correspond to the left and right parts of the block, respectively.

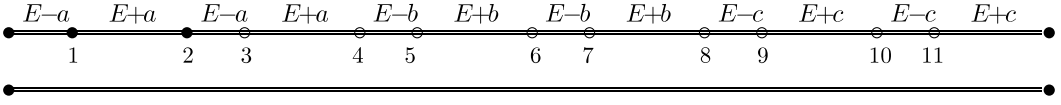

Fig. 6. Point 2 is in the upper part and block *i* does not contain interval of length *D*.

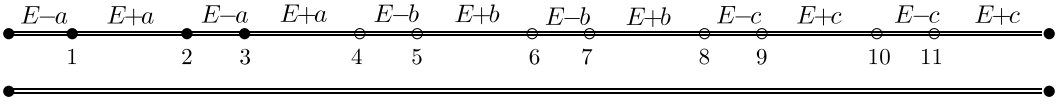

Fig. 7. Point 3 is in the upper part and block *i* does not contain interval of length *D*.

iD and $L - iD$. For C_k, generate two end distances *kD* (this pair of distances corresponds to the point $L/2$). Furthermore, assume that the order of elements in each triple $C_i \in C$ is fixed. For each $C_i \in C$, $i = 1, \ldots, K$, generate 11 pairs of end distances according to Fig. 4.

If elements t_1, t_2 and t_3 appear in $C_i \in C$ in this order, then we generate 11 end distances of the set $F_i := \{(i-1)D + E - t_1, (i-1)D + 2E, (i-1)D + 3E - t_1, (i-1)D + 4E, (i-1)D + 5E - t_2, (i-1)D + 6E, (i-1)D + 7E - t_2, (i-1)D + 8E, (i-1)D + 9E - t_3, (i-1)D + 10E, (i-1)D + 11E - t_3\}$ and 11 their complements $L - a$, $a \in F_i$. Thus, multiset *A* is fully specified.

We generate multiset *B* to consist of $r_j + 1$ numbers $E - j, j = 1, \ldots, 3m$, $r_j + 1$ numbers $E + j, j = 1, \ldots, 3m$, $5(K - m)$ numbers $2E$, $2(K - m)$ numbers $4E$ and m numbers $12E$.

The described instance of Quasi-SPDP-Blocks can be constructed in time polynomial in k and m, and all the numbers of this instance are upper bounded by a polynomial of m. Therefore, our transformation is polynomial and pseudo-polynomial at the same time. We now show that for the considered instance, a solution to X3C exists *if and only if* there exists a solution to Quasi-SPDP-Blocks.

Assume that there exists a solution to Quasi-SPDP-Blocks. We first show that only four cases are possible for the (correct) intervals in each block. Two of these cases for block *i* are given in Fig. 5. The other two cases are mirror images of the given two cases.

We will analyze possible positions of the points $1, 2, \ldots, 11$ in a solution of Quasi-SPDP-Blocks. Each of these points can either be present in the left part (upper part in Fig. 5) or right part (lower part in Fig. 5) of block *i*. We can assume without loss of generality that point 1 is present in the upper part. With regard to point 2 there are two cases to consider. First, assume that point 2 is located in the upper part and block *i* does not contain interval of length $D = 12E$, see Fig. 6.

Point 3 cannot be located in the lower part because it would produce interpoint distance $3E - a \notin B$. Assume that point 3 is located in the upper part and block *i* does not contain interval of length *D*, see Fig. 7.

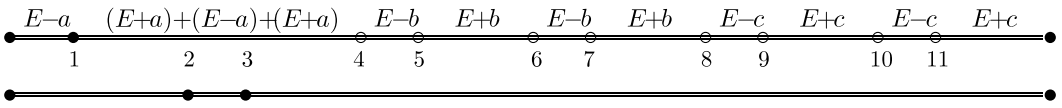

Fig. 8. Points 2 and 3 are in the lower part.

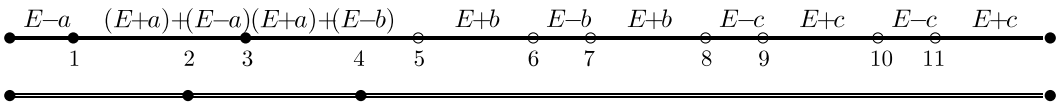

Fig. 9. Point 4 is in the lower part.

Point 4 cannot be located in the lower part because the upper part will contain one of the interpoint distances $2E + a - b$, $3E + a$, $4E + a - b$ or the one greater than or equal to $5E + a$, none of which is present in the multiset B. Therefore, point 4 should be located in the upper part. We have shown that in the case when point 2 is in the upper part and block i does not contain an interval of length D, points 1, 2, 3 and 4 are all in the upper part. However, the lower part should contain at least one point and this point will produce an interpoint distance greater than or equal to $5E - b$ which is not in the multiset B. Therefore, if point 2 is in the upper part, then the structure of block i is in accordance with Case 1 in Fig. 5.

Now assume that points 2 and 3 are located in the lower part, see Fig. 8.

In this case, the upper part should contain interpoint distance $3E + a$, $4E + a - b$ or the one greater than or equal to $5E + a$, none of which is present in the multiset B. Therefore, point 3 should be located in the upper part and positions of points 1, 2 and 3 are in accordance with Case 2. Assume that point 4 is in the lower part, see Fig. 9.

In this case, the upper part will contain one of the interpoint distances $2E + a - b$, $3E + a$, $4E + a - b$ or the one greater than or equal to $5E + a$, none of which is present in the multiset B. Therefore, point 4 should be located in the upper part. Now positions of the points 1, 2, 3 and 4 are in accordance with Case 2. Point 5 cannot be in the lower part because then it will produce an interpoint distance $3E - b \notin B$. If points 1, 2, 3, 4 and 5 are in accordance with Case 2, then point 6 cannot be in the upper part because in the lower part there will be an interpoint distance greater than $4E$, which is not in the multiset B.

Continuing in the same fashion we can prove that in any solution of Quasi-SPDP-Blocks the structure of each block is in accordance with Case 1, Case 2 or their mirror images.

Consider a solution of Quasi-SPDP-Blocks. Let Y' be the family of 3-element subsets $C_i \in C$ corresponding to the blocks of Case 1 in Fig. 5 or its mirror image. We will now show that $\{j \in C_i | C_i \in Y'\} = \{1, \ldots, 3m\} = M$. Assume that some $j \in M$ does not appear in Y'. Then interpoint distance $E - j$ appears only in blocks of Case 2 and the number of its appearances is equal to r_j which is less than the required number $r_j + 1$. Now assume that some $j \in M$ appears x times in Y', $2 \leq x \leq r_j$. Then the number of appearances of interpoint distance $E - j$ in blocks of Case 1 and Case 2 is equal to $2x + r_j - x = r_j + x \geq r_j + 2 > r_j + 1$. Hence, each $j \in M$ appears exactly once in Y', which means that Y' is a solution to X3C. Thus, the first part of the equivalence statement is proved.

Establishing the converse is easy. If Y is a solution to X3C, then construct a solution to Quasi-SPDP-Blocks in which block i is of Case 1 if $C_i \in Y$ and it is of Case 2 otherwise. □

Due to the discussion in Johnson [16], the following corollary takes place.

Corollary 3. *SPDP-Blocks (search version) is NP-hard in the strong sense.*

Our proof and the proof in [7] both consider the cases where the number of symmetric points is a part of the instance. It is worth noting that the computational complexity of SPDP containing a constant number of symmetric points or no such points is open.

We now prove that recognizing correctly more than 6 intervals of a block implies having recognized all 13 intervals in a constant time.

Theorem 4. *For an instance of SPDP-Blocks (search version) defined as it is shown in Theorem 2, if any* 7 *intervals of any block are determined correctly, then all* 13 *of them can be determined correctly in a constant time.*

Proof. Assume that we are given 7 correct intervals for some block i in an instance of SPDP-Blocks defined as it is shown in Theorem 2. Recall that this instance has a solution and the structure of block i in any solution coincides with either Case 1 or Case 2 in Fig. 5. The recognition of Case 1 or 2 can easily be carried out if we are given a correct interval different from the 6 intervals of lengths $E \pm a$, $E \pm b$ and $E \pm c$ in Case 2. For example, if there is a correct interval of length $2E$, then we know that block i is in accordance with Case 2 and if interval $[(i-1)D + E - a, (i-1)D + 2E]$ is given as correct, then block i is in accordance with Case 1. Since we are given 7 intervals, one of them is different from the above mentioned 6 intervals and it can be used to determine all 13 correct intervals of block i. Clearly, this determination can be done in a constant time. □

4. Conclusion

We have proved that recognizing more than $\frac{6}{13}$-th part of the DNA map in the simplified partial digest experiment is NP-hard in the strong sense. This result should stimulate discoveries and studies of algorithms which behave well for practical cases of SPDP.

Acknowledgments

This work was partially supported by the Polish Ministry of Science grant N519 314635 and the Marie Curie BIOPTRAIN fellowship of Mr. A. Kovalev.

References

[1] Z. Abrams, H.L. Chen, The Simplified Partial Digest Problem: Hardness and a probabilistic analysis, in: RECOMB Satellite Meeting on DNA Sequencing Technologies and Computation, 2004.
[2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties, Springer Verlag, Berlin, 1999.
[3] J. Blazewicz, E.K. Burke, M. Kasprzak, A. Kovalev, M.Y. Kovalyov, The simplified partial digest problem: Enumerative and dynamic programming algorithms, IEEE/ACM Transactions on Computational Biology and Bioinformatics 4 (2007) 668–680.
[4] J. Blazewicz, E.K. Burke, M. Kasprzak, A. Kovalev, M.Y. Kovalyov, The Simplified Partial Digest Problem: Approximation and a graph-theoretic model (submitted for publication).
[5] J. Blazewicz, P. Formanowicz, M. Kasprzak, M. Jaroszewski, W.T. Markiewicz, Construction of DNA restriction maps based on a simplified experiment, Bioinformatics 17 (2001) 398–404.
[6] J. Blazewicz, M. Jaroszewski, New algorithm for the simplified partial digest problem, Lecture Notes in Bioinformatics 2812 (2003) 95–110.
[7] J. Blazewicz, M. Kasprzak, Combinatorial optimization in DNA mapping—A computational thread of the Simplified Partial Digest Problem, RAIRO Operations Research 39 (2005) 227–241.
[8] M. Cieliebak, S. Eidenbenz, Measurement errors make the Partial Digest Problem NP-hard, LATIN (2004) 379–390.
[9] M. Cieliebak, S. Eidenbenz, P. Penna, Noisy data make the partial digest problem NP-hard, Lecture Notes in Bioinformatics 2812 (2003) 111–123.
[10] K. Danna, D. Nathans, Specific cleavage of simian virus 40 DNA by restriction endonuclease of Hemophilus Influenzae, Proceedings of the National Academy of Sciences of the United States of America 68 (1971) 2913–2917.
[11] K.J. Danna, G.H. Sack, D. Nathans, Studies of simian virus 40 DNA. VII. A cleavage map of the SV40 genome, Journal of Molecular Biology 78 (1973) 363–376.
[12] A. Daurat, Y. Gerard, M. Nivat, Some necessary clarifications about the Chords' Problem and the Partial Digest Problem, Theoretical Computer Science 347 (1–2) (2005) 432–436.
[13] A. Dudez, S. Chaillou, L. Hissler, R. Stentz, M. Champomier-Verges, C. Alpert, M. Zagorec, Physical and genetic map of the Lactobacillus sakei 23K chromosome, Microbiology 148 (2002) 421–431.
[14] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[15] GenBank genetic sequence database, http://www.ncbi.nlm.nih.gov/Genbank/index.html.
[16] D.S. Johnson, The NP-completeness column: An ongoing guide, Journal of Algorithms 6 (1985) 291–305.
[17] S. Keis, J.T. Sullivan, D.T. Jones, Physical and genetic map of the Clostridium saccharobutylicum (formerly Clostridium acetobutylicum) NCP 262 chromosome, Microbiology 147 (2001) 1909–1922.
[18] T. Kuwahara, M.R. Sarker, H. Ugai, S. Akimoto, S.M. Shaheduzzaman, H. Nakayama, T. Miki, Y. Ohnishi, Physical and genetic map of the Bacteroides fragilis YCH46 chromosome, FEMS Microbiology Letters 207 (2002) 193–197.
[19] P.A. Pevzner, Computational Molecular Biology. An Algorithmic Approach, MIT Press, Cambridge, MA, 2000.
[20] J. Setubal, J. Meidanis, Introduction to Computational Molecular Biology, PWS Publishing Company, Boston, 1997.
[21] S.S. Skiena, W.D. Smith, P. Lemke, Reconstructing sets from interpoint distances, in: Proceedings of the 6th ACM Symposium on Computational Geometry, 1990, pp. 332–339.
[22] S.S. Skiena, G. Sundaram, A partial digest approach to restriction site mapping, Bulletin of Molecular Biology 56 (1994) 275–294.
[23] M.S. Waterman, Introduction to Computational Biology. Maps, Sequences and Genomes, Chapman & Hall, London, 1995.
[24] J.D. Watson, F.H.C. Crick, Genetic implications of the structure of deoxyrinucleic acid, Nature 171 (1953) 964–967.