RAIRO Operations Research
RAIRO Oper. Res. **39** (2005) 227–241
DOI: 10.1051/ro:2006007

COMBINATORIAL OPTIMIZATION IN DNA MAPPING — A COMPUTATIONAL THREAD OF THE SIMPLIFIED PARTIAL DIGEST PROBLEM *

Jacek Blazewicz [1, 2] and Marta Kasprzak [1,2]

Abstract. In the paper, the problem of the genome mapping of DNA molecules, is presented. In particular, the new approach — the Simplified Partial Digest Problem (SPDP), is analyzed. This approach, although easy in laboratory implementation and robust with respect to measurement errors, when formulated in terms of a combinatorial search problem, is proved to be strongly NP-hard for the general error-free case. For a subproblem of the SPDP, a simple $O(n \log n)$-time algorithm is given, where n is a number of restriction sites.

Keywords. Combinatorial optimization, DNA restriction mapping, partial digest, computational complexity.

1. Introduction

A construction of a DNA physical map is an important step in the genome sequencing process (*cf.* [7] and [15,18,22] as excellent sources of algorithmic ideas used in computational biology). This map of a DNA molecule contains the information about locations of short, specific subsequences called markers and in turn places longer DNA subchains on the chromosome. One approach to create the map is based upon splitting the molecule into many shorter ones and *hybridizing*

Received June 22, 2005. Accepted October 28, 2005.

* *The research has been supported by KBN grant 3T11F00227. The research of the first author was sponsored by the ESPRC Fellowship at the University of Nottingham.*

[1] Institute of Computing Science, Poznan University of Technology, Piotrowo 3A, 60–965 Poznan, Poland; jblazewicz@cs.put.poznan.pl

[2] Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznan, Poland; marta@cs.put.poznan.pl

them with the *markers* (known very short DNA sequences). This approach results in the *interval graph model* used successfully in the past [1].

A more recent approach to the genome map construction relies on a digestion of a DNA molecule with restriction enzymes [22]. These enzymes cut DNA molecules within specific, short patterns of nucleotides called restriction sites. After the digestion, the lengths of obtained fragments are measured and the original ordering of these fragments must be reconstructed, and this is the place where combinatorial optimization (operations research) methods come to the effect. In practice, several variants of this approach are used. Two of the best known are *the double digest* and *the partial digest* (with many its variants).

In the double digest approach two restriction enzymes are used. A target DNA is amplified, *e.g.* using a PCR reaction, and the copies are divided into three sets. Molecules from the first set are digested by one enzyme, molecules from the second set are digested by the other enzyme and molecules from the third set are cut by both enzymes. All digestions are complete for the time span of each reaction is long enough to allow the enzyme to cut the target strand at each occurrence of the restriction site. As the result one obtains three collections of short DNA fragments that correspond to three digestion processes. The lengths of these fragments are measured during a gel electrophoresis process and recorded as three multisets. On the basis of this data locations of restriction sites in the target DNA are reconstructed. Unfortunately, from the combinatorial point of view the *Double Digest Problem* (DDP) is NP-hard even in an ideal case involving no errors [9], thus unlikely to admit a polynomial time algorithm. Another difficulty which must be taken into account when dealing with this approach is an exponential number of possible solutions (as measured with respect to a number of restriction sites — n) [16]. As an alternative *the partial digest approach* has been proposed [20, 21]. Here, one enzyme only is applied to cut the DNA molecule into fragments of different lengths by using the enzyme for different time periods. In the error-free case one gets here all $\binom{n}{2}$ fragment lengths between all pairs of cuts (including two ends of the DNA molecule). The resulting *Partial Digest Problem* (PDP) consists in reconstructing the original positions of the cuts. This problem is known also in discrete geometry [19, 20], where having all interpoint distances, one reconstructs the positions of a set of points on a line. For PDP two backtracking algorithms with an exponential worst case complexity have been proposed in [20] and [21], respectively. In general, the complexity of the PDP remains an open question, although measurement errors and noisy data result in the strong NP-hardness of the problem ([5] and [6], respectively). It is worth stressing that a number of possible solutions in the PDP is bounded from above by a polynomial in a number of restriction sites [20]. Other known approaches in that area include *optical mapping* [11, 17], *probed partial digest mapping* [13], and *labeled partial digest* [14]. Let us especially comment on the latter approach, which although more complicated from the implementation point of view (labeling the ends of a DNA molecule by using radioactive labeling), results in a polynomial algorithm for the related variant of the PDP and produces a unique solution [14].

In the present paper, we study yet another variant of the PDP, called the *Simplified Partial Digest Problem* (SPDP), which is very simple for the laboratory implementation [2]. In this approach only two digestions are performed. We will call them, respectively, a short digestion and a complete (long) digestion. After amplification, the copies of a target strand are splitted into two sets. The goal of a short digestion is to have all molecules from one of the sets cut in at most one occurrence of the restriction site. This is assured by properly chosen time span of the reaction. Molecules from the other set are cut in all occurrences of the restriction site due to the long reaction time span (a complete digestion). Then, as in other methods, the lengths of restriction fragments obtained, are measured during a gel electrophoresis process. Although, the algorihtms proposed for the SPDP [2, 3] are very efficient on the average and robust with respect to measurement errors, the complexity of the problem in the error-free case was open for several years.

In this paper, we present the proof that the general error-free variant of the SPDP is strongly NP-hard (of course in its *search version*). On the other hand, if the complete digestion yields a multiset composed of 1s and 2s only, the problem is solvable in $O(n \log n)$ time, where n is a number of digestion sites. An organization of the paper is as follows. Section 2 contains a formal definition of the SPDP and the proof of its strong NP-hardness. Section 3 presents a polynomial time algorithm for the restricted case of the problem. Conclusions in Section 4 summarize the work and indicate further research problems.

2. Computational complexity of the SPDP

In this section we prove, that the combinatorial search version of the Simplified Partial Digest Problem without errors is strongly NP-hard (this proof was presented at the conference in Dagstuhl [12]). What is interesting, we do not do it in a standard way by a transformation from the decision version of a known intractable problem to the decision version of our problem, since the decision counterpart of the SPDP is polynomially solvable. This is because — on the assumption that the instance of our problem has no errors — the restriction fragments from both digestion reactions exactly match those existing in the real restriction map, therefore the solution always exists. However, the process of finding this solution (*i.e.* the map) is not easy from the computational complexity point of view.

A similar situation — where the decision version of a problem is easy but its search version is hard — is encountered in the DNA sequencing problem [4]. In this problem, as well as in SPDP, the additional information about instances — the lack of errors or some type of errors — results in the answer "yes" for every instance of the decision version. Therefore, the process of proving strong NP-hardness of the search problem gone through an artificial decision problem, allowing also instances with the answer "no".

In our reasoning we based on a deduction formed in [10] for the Hamiltonian Circuit Problem (being NP-complete in its decision version). As it has been stated

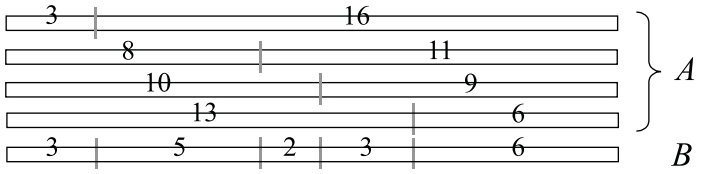

FIGURE 1. An example for the Simplified Partial Digest Problem without errors.

there, even if we knew a graph contains a Hamiltonian circuit (in this case the decision version is, of course, trivially easy), we could not find it in polynomial time unless P = NP. For "if we had such an algorithm A, we could use it to tell in polynomial time whether an arbitrary graph G has a Hamiltonian circuit. Let p be the polynomial that bounds A's running time on graphs with Hamiltonian circuits. Apply A to G. If G has a Hamiltonian circuit, A will find one in time $p(|G|)$. If G does not have such a circuit, then after $p(|G|)$ steps A could not have found one, and we will know that none exists".

2.1. FORMULATIONS OF RELATED PROBLEMS

The innovation of the well-known Partial Digest Problem, studied here, was based on a simplified biochemical experiment and proposed first time in [2]. Its search version without any error in instances can be formulated as below.

Problem 1. *Simplified Partial Digest Problem (Π_{SMs}) — search version* [2].
Instance: *Multiset $A = \{a_1, a_2, ..., a_{2n}\}$ of lengths of restriction fragments coming from short digestion and multiset $B = \{b_1, b_2, ..., b_{n+1}\}$ of lengths of restriction fragments coming from long digestion, A and B containing no errors.*
Answer: *A map of n restriction sites of a DNA chain consistent with multisets A and B, i.e. such an order of elements of B which allows to cover the indicated cuts by some order of elements of A.*

We see that $D_{\Pi_{\mathrm{SMs}}} = Y_{\Pi_{\mathrm{SMs}}}$, where $D_{\Pi_{\mathrm{SMs}}}$ is the set of all instances of Π_{SMs} and $Y_{\Pi_{\mathrm{SMs}}}$ is the set of all instances of Π_{SMs} for which such a solution exists. (This follows from the fact, that both multisets come from a real error-free digestion experiment.) Example 1 shows an instance of this problem together with a possible solution.

Example 1. Let the output of the digestion reactions performed without errors be: $A = \{3, 6, 8, 9, 10, 11, 13, 16\}$ from the short reaction and $B = \{2, 3, 3, 5, 6\}$ from the long one. The feasible solution for the problem is presented in Figure 1 and can be written as the following ordered list of B: $[3, 5, 2, 3, 6]$.

To study the computational complexity of problem Π_{SMs}, we must introduce an additional decision *quasi-mapping* problem Π_{QMd}. In the quasi-mapping problem the multisets contain arbitrary positive integers instead of the ones coming from an errorless experiment, thus the answer to the new problem is not always "yes".

Problem 2. *Quasi-Mapping Problem (Π_{QMd}) — decision version.*
Instance: *Multisets $A = \{a_1, a_2, ..., a_{2n}\}$ and $B = \{b_1, b_2, ..., b_{n+1}\}$ of positive integers.*
Question: *Does there exist a map of n "restriction sites" consistent with multisets A and B, i.e. such an ordering of elements of B which allows to cover the indicated "cuts" by some order of elements of A?*

The two problems Π_{SMs} and Π_{QMd} have the same sets of instances with the positive answer: $Y_{\Pi_{\mathrm{SMs}}} = Y_{\Pi_{\mathrm{QMd}}}$, because any instance of Π_{QMd} resulting in the "yes" answer belongs also to $D_{\Pi_{\mathrm{SMs}}}$. However, problem Π_{QMd} contains also instances with the negative answer ($D_{\Pi_{\mathrm{SMs}}} \subset D_{\Pi_{\mathrm{QMd}}}$).

In the next subsection we construct a pseudo-polynomial transformation from the problem Numerical Matching With Target Sums, cited below, to the considered SPDP.

Problem 3. *Numerical Matching With Target Sums (Π_{NMd}) — decision version* [8].
Instance: *Disjoint sets X and Y, each containing m elements, sizes $s(x_i)$ and $s(y_i)$ for every $x_i \in X$ and $y_i \in Y$, and a target vector $[z_1, z_2, ..., z_m]$; $s(x_i)$, $s(y_i)$, and z_i being positive integers, $i = 1...m$.*
Question: *Can $X \cup Y$ be partitioned into m disjoint sets $W_1, W_2, ..., W_m$, each containing exactly one element from X and one element from Y, such that $\sum_{w \in W_i} s(w) = z_i$, $i = 1...m$?*

This problem is strongly NP-complete [8].

2.2. The transformation

The first stage of proving strong NP-hardness of the Simplified Partial Digest Problem without errors consists in a simple modification of problem Π_{NMd}. We are interested in a variant with the ranges of variables shifted by some added values. The modified ranges will have several properties, in particular, they will be disjoint. Initially, the ranges of values of variables $s(x_i)$, $s(y_i)$, and z_i, $i = 1...m$, can be written as follows:

$$\begin{aligned} s(x_i) &\in \langle x_{\mathrm{L}}, x_{\mathrm{R}} \rangle, \\ s(y_i) &\in \langle y_{\mathrm{L}}, y_{\mathrm{R}} \rangle, \\ z_i &\in \langle z_{\mathrm{L}}, z_{\mathrm{R}} \rangle, \end{aligned}$$

where the variables with index L mean the smallest values and the ones with index R mean the largest values in the respective collections. These ranges are arbitrary, however, we assume here that they satisfy few obvious conditions:

$$\begin{aligned} z_{\mathrm{R}} &\leq x_{\mathrm{R}} + y_{\mathrm{R}}, \\ z_{\mathrm{L}} &\geq x_{\mathrm{L}} + y_{\mathrm{L}}, \\ z_{\mathrm{R}} &> x_{\mathrm{R}}, \\ z_{\mathrm{R}} &> y_{\mathrm{R}}. \end{aligned}$$

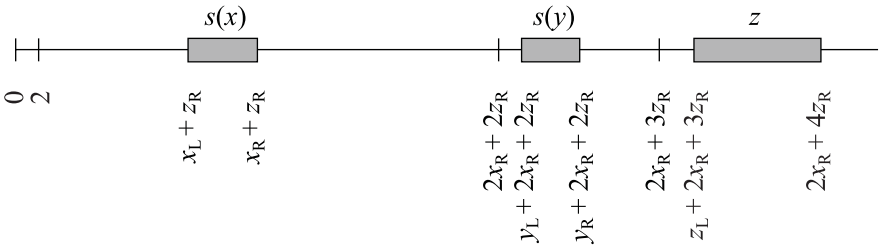

FIGURE 2. The ranges of values of variables $s(x_i)$, $s(y_i)$, and z_i, $i = 1...m$, after the modification.

The above assumptions do not change the computational complexity of problem Π_{NMd} — if one of them is not satisfied, the problem becomes easy because the answer is obviously "no".

We modify the initial ranges of the variables in the following way (where ":=" assigns the right-hand side value to the left-hand side variable):

$$\begin{aligned} s(x_i) &:= s(x_i) + z_{\text{R}}, && i = 1...m \\ s(y_i) &:= s(y_i) + 2x_{\text{R}} + 2z_{\text{R}}, && i = 1...m \\ z_i &:= z_i + 2x_{\text{R}} + 3z_{\text{R}}, && i = 1...m. \end{aligned}$$

The new ranges have the following form ($i = 1...m$):

$$\begin{aligned} s(x_i) &\in \langle x_{\text{L}} + z_{\text{R}}, x_{\text{R}} + z_{\text{R}} \rangle, \\ s(y_i) &\in \langle y_{\text{L}} + 2x_{\text{R}} + 2z_{\text{R}}, y_{\text{R}} + 2x_{\text{R}} + 2z_{\text{R}} \rangle, \\ z_i &\in \langle z_{\text{L}} + 2x_{\text{R}} + 3z_{\text{R}}, 4z_{\text{R}} + 2x_{\text{R}} \rangle. \end{aligned}$$

They are visualized in Figure 2.

Lemma 1. *Problem Π_{NMd} and its version with the ranges of variable values shifted as above, are equivalent.*

Proof. Both problems have the same sets of instances (at the beginning). The only difference is in the equation from the question, where, instead of $s(x_i) + s(y_j) = z_k$ in problem Π_{NMd}, we have $s(x_i) + z_{\text{R}} + s(y_j) + 2x_{\text{R}} + 2z_{\text{R}} = z_k + 2x_{\text{R}} + 3z_{\text{R}}$ in the new problem with the shifted ranges. We see, that the two problems are the same. □

The variables from problem Π_{NMd} increased by the proposed values have several useful properties.

Lemma 2. *None of the modified z_i, $i = 1...m$, can be equal to some $s(x_j)$ or to a sum of any $s(x_j)$ and $s(x_k)$.*

Proof. The first statement is obvious, see Figure 2. As to the second one, the sum of two largest possible values of $s(x_j)$ and $s(x_k)$, *i.e.* $2x_{\mathrm{R}} + 2z_{\mathrm{R}}$, is still smaller than $z_{\mathrm{L}} + 2x_{\mathrm{R}} + 3z_{\mathrm{R}}$ being the smallest z_i. □

Lemma 3. *None of the modified z_i, $i = 1...m$, can be equal to some $s(y_j)$ or to a sum of any $s(y_j)$ and $s(y_k)$.*

Proof. Also here the first statement is obvious, see Figure 2. On the other hand, the sum of two smallest possible values of $s(y_j)$ and $s(y_k)$, *i.e.* $2y_{\mathrm{L}} + 4x_{\mathrm{R}} + 4z_{\mathrm{R}}$, is larger than $2x_{\mathrm{R}} + 4z_{\mathrm{R}}$ being the largest possible z_i. □

Lemma 4. *None of the modified z_i, $i = 1...m$, can be equal to a sum of any $s(y_j)$, $s(x_k)$, and $s(x_l)$.*

Proof. The sum of one smallest $s(y_j)$ and two smallest $s(x_k)$ and $s(x_l)$, *i.e.* $y_{\mathrm{L}} + 2x_{\mathrm{L}} + 2x_{\mathrm{R}} + 4z_{\mathrm{R}}$, is larger than the largest $z_i = 2x_{\mathrm{R}} + 4z_{\mathrm{R}}$. □

Now, we can define a transformation from problem Π_{NMd} to problem Π_{QMd}, which is given below.

The transformation

Given an instance of problem Π_{NMd}, the corresponding instance of Π_{QMd} is constructed as follows.

(1) Shift the ranges of numbers in the problem Π_{NMd} as specified above, *i.e.*

$$\begin{aligned} s(x_i) &:= s(x_i) + z_{\mathrm{R}}, & i &= 1...m \\ s(y_i) &:= s(y_i) + 2x_{\mathrm{R}} + 2z_{\mathrm{R}}, & i &= 1...m \\ z_i &:= z_i + 2x_{\mathrm{R}} + 3z_{\mathrm{R}}, & i &= 1...m. \end{aligned}$$

From now on all the variables have these modified values, if not stated otherwise.

(2) Add values $s(x_i)$ and $s(y_i)$, $i = 1...m$, to (initially empty) multiset B. Also add to B $(\sum_{i=1...m} z_i - 2m)$ times value 1.

(3) Set L to $2\sum_{i=1...m} z_i - 2m$ and n to $\sum_{i=1...m} z_i - 1$.

(4) For all $i = 1...\frac{L}{2}$ add values i and $L-i$ to (initially empty) multiset A. Also add to A values $\sum_{j=1...i} z_j - i$ and $L - \sum_{j=1...i} z_j - i$ for all $i = 1...m-1$.

Lemma 5. *The proposed transformation can be computed in time bounded by a polynomial in the length of the instance of* Π_{NMd} *(*$\mathrm{Len}_{\mathrm{NMd}}$*) and the maximal number appearing in this instance (*$\mathrm{Max}_{\mathrm{NMd}}$*).*

Proof. $\mathrm{Len}_{\mathrm{NMd}}$ is $\mathrm{O}(m\lceil\log\mathrm{Max}_{\mathrm{NMd}}\rceil)$. In the first step of the transformation we make $\mathrm{O}(\mathrm{Len}_{\mathrm{NMd}})$ operations. New values of the variables do not change $\mathrm{Len}_{\mathrm{NMd}}$ and $\mathrm{Max}_{\mathrm{NMd}}$ substantially: m is not changed, new $\mathrm{Max}_{\mathrm{NMd}}$ is up to 6 times larger than previously.

Filling multiset B requires $\mathrm{O}(\mathrm{Len}_{\mathrm{NMd}}\mathrm{Max}_{\mathrm{NMd}})$ operations. Step (3) is $\mathrm{O}(\mathrm{Len}_{\mathrm{NMd}})$ and filling A takes $\mathrm{O}(\mathrm{Len}_{\mathrm{NMd}}\lceil\log\mathrm{Len}_{\mathrm{NMd}}\rceil\mathrm{Max}_{\mathrm{NMd}})$ operations. Taking the above

functions together, we have $O(\mathrm{Len}_{\mathrm{NMd}}\lceil\log\mathrm{Len}_{\mathrm{NMd}}\rceil\mathrm{Max}_{\mathrm{NMd}})$ as the complexity of the proposed transformation. Thus, it is pseudo-polynomial in time. □

We can now prove the following main theorem.

Theorem 1. *Quasi-mapping problem* Π_{QMd} *is strongly NP-complete.*

Proof. The proof uses the proposed transformation, which is pseudo-polynomial one (see Lem. 5). Here it is proven that the transformation is correct. For all $I \in D_{\Pi_{\mathrm{NMd}}}$, $I \in Y_{\Pi_{\mathrm{NMd}}}$ if and only if $t(I) \in Y_{\Pi_{\mathrm{QMd}}}$, where t means the transformation.

The first step of the transformation slightly modifies problem Π_{NMd}, but both versions are equivalent (see Lem. 1). Thus, in the following the shifted ranges of values of the variables are used.

Let us assume, that there exists a solution for problem Π_{NMd}. It means, that there is a partition of $X \cup Y$ such that every disjoint subset W_i, $i = 1...m$, contains one x_j and one y_k and $s(x_j)+s(y_k) = z_i$, for some $j, k \in \langle 1, m\rangle$. Then, the solution of problem Π_{QMd} can be constructed by ordering elements of B in the following way.

```
for i := 1 to m with step 1
begin
   take s(x_j) : x_j ∈ W_i;
   for j := 1 to z_i − s(x_j) − 1 with step 1
      take 1;
end
for i := m to 1 with step -1
begin
   take s(y_k) : y_k ∈ W_i;
   for j := 1 to z_i − s(y_k) − 1 with step 1
      take 1;
end
```

We take to the solution m times some $s(x_j)$, m times some $s(y_k)$, and $\sum_{i=1...m}$ $(z_i - s(x_j) - 1 + z_i - s(y_k) - 1)$ times element 1, $x_j, y_k \in W_i$. All x_j and y_k, $j, k = 1...m$, are in the partitioning and the sums of sizes of these pairs cover the whole vector $[z_1, z_2, ..., z_m]$. Thus, all the elements of B are used after finishing the above procedure.

Moreover, we can cover n indicated "cuts" by an order of pairs of all elements of A. The pairs can be easily determined by summing the elements up to L (which is the length of the solution of Π_{QMd}). The procedure of ordering them (placing an element on the left or on the right side of the solution, *i.e.* calculating the distance from the left or from the right end of the restriction map, respectively), is shown below.

```
for i := 1 to m with step 1
   for j := 1 to z_i − 1 with step 1
   begin
      if j < s(x_k) : x_k ∈ W_i then
```

```
      begin
         place ∑_{l=1...i-1}(z_l − 1) + j on the right side;
         place L − ∑_{l=1...i-1}(z_l − 1) + j on the left side;
      end
      else
      begin
         place ∑_{l=1...i-1}(z_l − 1) + j on the left side;
         place L − ∑_{l=1...i-1}(z_l − 1) + j on the right side;
      end
   end
for i := 1 to m − 1 with step 1
begin
   place ∑_{j=1...i}(z_j − 1) on the right side;
   place L − ∑_{j=1...i}(z_j − 1) on the left side;
end
```

We use in the procedure all elements of A: values increasing from 1 to $\sum_{i=1...m}(z_i - 1)$ with step 1, as well as the elements $\sum_{j=1...i}(z_j - 1)$, i = 1...m-1, together with their complements. Also their ordering agrees with the ordering within B in the sense of "cuts", what follows the procedures. Therefore, a feasible solution for problem Π_{QMd} exists.

Now, let us assume, that there exists a solution for problem Π_{QMd}. Thus, there is an order of $n+1$ elements of B which allows to cover the indicated "cuts" by some order of n pairs of elements of A. The solution for the corresponding instance of problem Π_{NMd} can be then constructed as follows.

Having the solution for problem Π_{QMd}, we know that several "cuts" appear for sure. Those are the ones corresponding to duplicated pairs of complements: if we have in A value d twice (together with $L-d$ twice, of course), then in any solution to problem Π_{QMd} one d will be placed on the left and the second d will be placed on the right. (Two "cuts" cannot appear on the same side, because we assume the lack of any errors in the data of the problem. Duplication of a "cut" would be such an error.) The guaranteed "cuts" are determined by pairs $\sum_{j=1...i} z_j - i$ and $L - \sum_{j=1...i} z_j - i$ for all $i = 1...m-1$, together with their mirrors. Also the "cut" in the middle of L is guaranteed by the pair of two $\frac{L}{2}$. These "cuts" become ends of fragments, which correspond to disjoint subsets from a solution of problem Π_{NMd}. Every pair of mirroring fragments F_{iL} and F_{iR} of length $z_i - 1$ corresponds to set W_i of size z_i.

How are elements of B arranged in the solution? Except for many 1s, B contains also all $s(x_i)$ and $s(y_i)$, $i = 1...m$. These sizes are significant, especially for Y. Summing up Lemmae 2, 3, and 4, which are still true if values z_i will be decreased by 1, we deduce that every pair F_{iL}, F_{iR} can contain at most one $s(y_j)$ of any value. Because B includes m sizes of elements of Y and there are m pairs of fragments, all $s(y_i)$, $i = 1...m$, must be placed in separate pairs. In addition, none two (or more) sizes of elements of X can be added to a pair of fragments (already

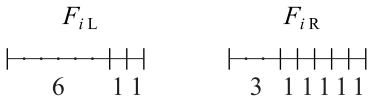

FIGURE 3. Two mirroring fragments F_{iL} and F_{iR} with some components.

containing some $s(y_j)$), so all $s(x_i)$, $i = 1...m$, must be also placed in separate pairs.

At first glance, it is not obvious why two mirroring fragments F_{iL} and F_{iR} cannot include two sizes of elements of Y. It is displayed in Figure 3. If some size is assigned to one fragment, the corresponding place in the second fragment must be filled by "cuts". Because sizes of elements of Y are greater than halfs of lengths of fragments plus 1, none two such sizes can be inserted into a mirroring pair.

If we have exactly one $s(x_j)$ and exactly one $s(y_k)$ assigned to a pair F_{iL}, F_{iR}, for some $i, j, k \in \langle 1, m\rangle$, plus in addition only 1s, there is only one possible placement. One of the longer sizes is shifted to the left (or right) end of F_{iL} and the second one is shifted to the left (or right, respectively) end of F_{iR}. The sum of both sizes is equal to z_i. The remaining places are filled by 1s (see Fig. 3). Another placement would result in the occurrence of an additional element of a value greater than 1, what is in contradiction to previous proofs.

Having the solution of problem Π_{QMd} we can immediately read the solution of problem Π_{NMd}: those elements x_j and y_k, which sizes appear in pair F_{iL}, F_{iR}, will compose set W_i. All the requirements for a solution of problem Π_{NMd} are satisfied. This ends the proof. □

The proposed transformation of an instance of problem Π_{NMd} to an instance of problem Π_{QMd} is illustrated by the following example.

Example 2. Let the example instance of problem Π_{NMd} be:

$$\begin{gathered} m = 3, \\ X = \{x_1, x_2, x_3\}, \\ Y = \{y_1, y_2, y_3\}, \\ s(x_1) = 2, s(x_2) = 3, s(x_3) = 5, \\ s(y_1) = 4, s(y_2) = 5, s(y_3) = 6, \\ z_1 = 7, z_2 = 9, z_3 = 9. \end{gathered}$$

After shifting the ranges of the variables we get the values:

$$\begin{gathered} s(x_1) = 11, s(x_2) = 12, s(x_3) = 14, \\ s(y_1) = 32, s(y_2) = 33, s(y_3) = 34, \\ z_1 = 44, z_2 = 46, z_3 = 46. \end{gathered}$$

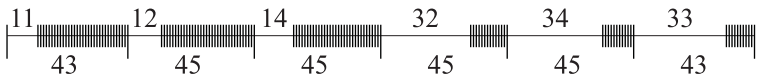

FIGURE 4. An example solution of problem Π_{QMd}.

The construction of the instance of problem Π_{QMd} ends with the following result:

$$
\begin{aligned}
&n = 135, L = 266,\\
&A = \bigcup_{i=1\ldots133}\{i, 266 - i\} \cup \{43, 223, 88, 178\},\\
&B = \bigcup_{i=1\ldots130}\{1\} \cup \{11, 12, 14, 32, 33, 34\}.
\end{aligned}
$$

The feasible solution for the instance after the transformation (*i.e.* of problem Π_{QMd}) is shown in Figure 4.

This solution can be easily translated to a feasible solution of problem Π_{NMd}:

$$
\begin{aligned}
&W_1 = \{x_1 + y_2\} \quad (11 + 33 = 44 \;\rightarrow\; 2 + 5 = 7),\\
&W_2 = \{x_2 + y_3\} \quad (12 + 34 = 46 \;\rightarrow\; 3 + 6 = 9),\\
&W_3 = \{x_3 + y_1\} \quad (14 + 32 = 46 \;\rightarrow\; 5 + 4 = 9).
\end{aligned}
$$

Finally, we prove the computational hardness of Simplified Partial Digest Problem without errors in the search version Π_{SMs}.

Theorem 2. *The Simplified Partial Digest Problem without errors* Π_{SMs} *(search version) is strongly NP-hard.*

Proof. Proving strong NP-completeness of quasi-mapping problem Π_{QMd} (Th. 1) directly leads to proving strong NP-hardness of the corresponding problem Π_{SMs}. For, if we had an algorithm solving Π_{SMs} in polynomial time, we could use it to solve problem Π_{QMd} in polynomial time in the following way. We could apply the algorithm to the instance of Π_{QMd}, and after a number of steps bounded by the polynomial function known for the algorithm we could have the answer for Π_{QMd}. Either the algorithm would find the solution and the answer would be "yes", or the algorithm would not find one and the answer would be "no". As pointed out earlier, a similar reasoning has been used in [10], where the problem of looking for a Hamiltonian circuit in a graph has been considered. □

3. A polynomially solvable subproblem of SPDP

On the other hand, a special version of the Simplified Partial Digest Problem without errors (a combinatorial search problem), in which multiset B is composed of only the elements of values 1 and 2, is polynomially solvable. In this section we present the algorithm solving the subproblem, defined below.

Problem 4. *Simplified Partial Digest Problem with long-digestion fragments of lengths only 1 or 2 (*Π_{SM12s}*) — search version.*

Instance: *Multiset* $A = \{a_1, a_2, ..., a_{2n}\}$ *of lengths of restriction fragments coming from short digestion and multiset* $B = \{b_1, b_2, ..., b_{n+1}\}$ *of lengths of restriction fragments coming from long digestion,* A *and* B *containing no errors and* $b_i \in \langle 1, 2\rangle$, $i = 1...n+1$.
Answer: *A map of* n *restriction sites of a DNA chain consistent with multisets* A *and* B, *i.e. such an order of elements of* B *which allows to cover the indicated cuts by some order of the elements of* A.

The algorithm solving Π_{SM12s}

(1) Calculate $L = \sum_{i=1...n+1} b_i$. Initially the solution is one fragment of length L without cuts.

(2) Match complementary pairs of lengths in A, *i.e.* the ones summing up to L. Let the pairs be denoted by A_i, $i = 1...n$, and satisfying $\bigcup_{i=1...n} A_i = A$ and $\bigcap_{i=1...n} A_i = \emptyset$.

(3) Select all identical pairs A_i and A_j, $i, j = 1...n$, and mark the cuts corresponding to them in the solution symmetrically (from both sides). Remove all selected A_i and A_j from A.

(4) If A is empty, go to step (7). Otherwise, select such A_i in A, that contains the shortest length l in current A. Mark the cut placed l units from the left end of the solution and remove A_i from A.

(5) If A is empty, go to step (7). Otherwise, increase l by 1. If some cut already exists in the solution l units from the right end, go to step (4). Otherwise, mark the cut placed l units from the right end of the solution and remove A_i which contais l from A.

(6) If A is empty, go to step (7). Otherwise, increase l by 1. If some cut already exists in the solution l units from the left end, go to step (4). Otherwise, mark the cut placed l units from the left end of the solution, remove A_i which contais l from A, and go to step (5).

(7) The current solution contains all cuts, *i.e.* it corresponds to an order of elements of multiset B.

Let us note, that the algorithm uses only information coming from multiset A and does not check it with multiset B. This is because the applied moves are the only possible ones, and in the case where we have no errors they have to be correct. This is demonstrated in the following proof.

Theorem 3. *Problem* Π_{SM12s} *is solvable in time* $O(n \log n)$.

Proof. Let us analyze the consecutive steps of the algorithm.

The first two steps are obvious. The third one consists in placing the symmetric cuts indicated by pairs from A. Of course, if we have in A two identical subsets $\{x, L-x\}$, then in the problem without errors both cuts must appear. Thus, they must be placed in opposite parts of the solution, *i.e.* one cut x units from the left and the second one x units from the right end.

All remaining cuts are assymetric. Because the current solution is symmetric, no matter what side we place the next cut. So, the left side can be chosen, as well

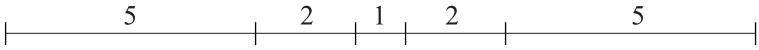

FIGURE 5. A symmetric part of an example solution of problem Π_{SM12s}.

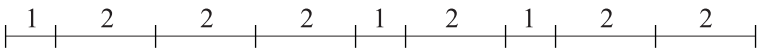

FIGURE 6. An example solution of problem Π_{SM12s} with the first slot filled by cuts.

as the strategy of taking the cut nearest the left end first. Therefore, the fourth step is correct. It also concerns further calling of step (4), when we fill a part between symmetric cuts and go to the next *slot* (*i.e.* a fragment between two pairs of symmetric cuts from step (3)) — then there is no difference from what side we start to place next cuts.

After placing the previous cut in step (4) we must look at the opposite side of the solution. In the symmetric place we cannot have a cut, since all cuts added after step (3) are assymetric. And because in multiset B there are only lengths equal to 1 or 2, we must place a cut to get a fragment of length at most 2 after one unit on the right side without a cut. So, step (5) is correct as well as step (6) with a similar reasoning. The only exception, when we reach an existing cut, concerns the situation when we fill the whole current slot and jump to the next one (the slot closer to the middle of the solution).

Summing up the above reasoning, we may conclude that the algorithm is correct. As to its complexity, if the elements of A are sorted in step (2), the steps can be done in time, respectively, $\mathrm{O}(n \log n)$, $\mathrm{O}(n \log n)$, $\mathrm{O}(n \log n)$, $\mathrm{O}(\log n)$, $\mathrm{O}(\log n)$, and $\mathrm{O}(\log n)$. Steps (4)–(6) are performed up to n times, so the whole algorithm is of the complexity $\mathrm{O}(n \log n)$. □

Example 3 presents in detail the proposed algorithm.

Example 3. Let our instance of problem Π_{SM12s} be: $A = \{1, 2, 3, 4, 5, 5, 6, 7, 7, 8, 8, 9, 10, 10, 11, 12, 13, 14\}$ and $B = \{1, 1, 1, 1, 1, 2, 2, 2, 2, 2\}$. Thus, we have 9 restriction places within the solution of length 15. The complementary pairs of lengths in A are: $\{1, 14\}$, $\{2, 13\}$, $\{3, 12\}$, $\{4, 11\}$, $\{5, 10\}$, $\{5, 10\}$, $\{6, 9\}$, $\{7, 8\}$, and $\{7, 8\}$. To the initially empty solution we add all symmetric cuts determined by the instance and we get the sequence of fragments as in Figure 5.

After that $A = \{\{1, 14\}, \{2, 13\}, \{3, 12\}, \{4, 11\}, \{6, 9\}\}$. We add the remaining cuts starting from the one nearest the left end of the solution, the result is shown in Figure 6.

Finally, we jump to the next slot and place the only remaining cut $\{6, 9\}$. The whole solution is in Figure 7.

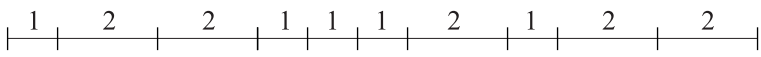

FIGURE 7. A complete solution for the example problem Π_{SM12s}.

4. Conclusions

In the paper, the DNA Simplified Partial Digest Problem was analyzed from the viewpoint of combinatorial optimization (OR) approaches. The general error-free case was proved to be strongly NP-hard. On the other hand, a special form of the multiset resulting from the long digestion and composed of only 1s and 2s results in a simple, polynomial-time algorithm. The question remains open, whether or not, a similar approach can be used for the multisets composed of other restricted values (*e.g.* 1 and 3). An interesting problem is a question of approximability, *i.e.* an existence of polynomial-time algorithm constructing solutions being not far from the optimum one with respect to a certain optimality criterion.

References

[1] S. Benzer, On the topology of the genetic fine structure, in *Proceedings of the National Academy of Sciences of the USA* **45** (1959) 1607–1620.

[2] J. Blazewicz, P. Formanowicz, M. Kasprzak, M. Jaroszewski and W.T. Markiewicz, Construction of DNA restriction maps based on a simplified experiment. *Bioinformatics* **17** (2001) 398–404.

[3] J. Blazewicz and M. Jaroszewski, New algorithm for the Simplified Partial Digest Problem. *Lect. Notes Bioinformatics* **2812** (2003) 95–110.

[4] J. Blazewicz and M. Kasprzak, Complexity of DNA sequencing by hybridization. *Theoret. Comput. Sci.* **290** (2003) 1459–1473.

[5] M. Cieliebak and S. Eidenbenz, Measurement errors make the partial digest problem NP-hard. *Lect. Notes Comput. Sci.* **2976** (2004) 379–390.

[6] M. Cieliebak, S. Eidenbenz and P. Penna, Noisy data make the partial digest problem NP-hard. *Lect. Notes Bioinformatics* **2812** (2003) 111–123.

[7] G.B. Fogel and D.W. Corne, eds. *Evolutionary Computations in Bioinformatics*. Morgan Kaufman, Boston (2003).

[8] M.R. Garey and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco (1979).

[9] L. Goldstein and M.S. Waterman, Mapping DNA by stochastic relaxation. *Adv. Appl. Math.* **8** (1987) 194–207.

[10] D.S. Johnson, The NP-completeness column: an ongoing guide. *J. Algorithms* (1985) **6** 291–305.

[11] R.M. Karp and R. Shamir, Algorithms for optical mapping, in *Proceedings of the Second International Conference on Computational Biology* (RECOMB), New York, (1998) 117–124.

[12] M. Kasprzak, Computational complexity of the Simplified Partial Digest Problem, in *05441 Abstracts Collection — Managing and Mining Genome Information: Frontiers in Bioinformatics*, edited by J. Blazewicz, J.Ch. Freytag and M. Vingron. IBFI, Dagstuhl (2005).

[13] L. Newberg and D. Naor, A lower bound on the number of solutions of the probed partial digest problem. *Adv. Appl. Math.* **14** (1993) 172–183.

[14] G. Pandurangan and H. Ramesh, The restriction mapping problem revisited. *J. Comput. Syst. Sci.* **65** (2002) 526–544.

[15] P.A. Pevzner, *Computational Molecular Biology: An Algorithmic Approach*. MIT Press, Cambridge (2000).
[16] W. Schmitt and M.S. Waterman, Multiple solutions of DNA restriction mapping problem. *Adv. Appl. Math.* **12** (1991) 412–427.
[17] D.C. Schwartz, X. Li, L.I. Hernandez, S.P. Ramnarain, E.J. Huff and Y.K. Wang, Ordered restriction maps of *Saccharomyces cerevisiac* chromosomes constructed by optical mapping. *Science* **262** (1993) 110–114.
[18] J. Setubal and J. Meidanis, *Introduction to Computational Biology*. PWS Publishing Company, Boston (1997).
[19] S.S. Skiena, Geometric reconstruction problems, in *Handbook of Discrete and Computational Geometry* edited by J.E. Goodman and J. O'Rourke. CRC Press, Boca Raton (1997), 481–490.
[20] S.S. Skiena, W.D. Smith and P. Lemke, Reconstructing sets from interpoint distances in *Proceedings of Sixth ACM Symposium on Computational Geometry* (1990) 332–339.
[21] S.S. Skiena and G. Sundaram, A partial digest approach to restriction site mapping. *Bull. Math. Biology* **56** (1994) 275–294.
[22] M.S. Waterman, *Introduction to Computational Biology. Maps, Sequences and Genomes*, Chapman and Hall: London (1995).