Australian Journal of Basic and Applied Sciences, 5(3): 516-520, 2011
ISSN 1991-8178

A Zero-one Programming Model for Partial Digest Problem

[1]Reza Nadimi, [2]Hassan Salehi Fathabadi

[1]Department of Applied Mathematics, Faculty of Mathematics,University of Mazandaran, Babolsar, Mazandaran, Iran
[2]School of Mathematics, Statistics and Computer Science, University of Tehran, Tehran, Iran.

Abstract: The Partial Digest is a well-studied problem with important applications in physical mapping of DNA molecules. In this paper we present a new point of view to the Partial Digest Problem (*PDP*). We present a sufficient condition for solution of *PDP* and formulate the *PDP* as a linear zero-one programming model, such that any optimal solution of this model will be a solution of *PDP*.

Key words: Restriction site mapping, DNA, Zero-one programming.

INTRODUCTION

One of the interesting tasks in computational biology is Restriction Site Mapping. When a particular restriction enzyme is added to a DNA, the DNA is cut at particular restriction sites. The goal of restriction site mapping is to determine the location of every site for a given enzyme. Using gel electrophoresis, one can find the distance between each pair of restriction sites. In the Partial Digest Problem, we are given these distances arising from digestion experiments with one enzyme, and we want to compute the locations of all restriction sites. Let $x = \{x_0, x_1, \cdots, x_n\}$ be the set of restriction site locations on a DNA strand. We denote the "multiset" of all N=n(n+1)/2 pairwise distances between these sites by $\Delta x = \{x_j - x_i \mid x_j > x_i, i, j = 0,1,\cdots,n\}$.

In the partial digest problem, given a multiset $B=\{b_1,b_2,...,b_N\}$ of distances, the goal is to find a set $Y=\{y_0,y_1,...,y_N\}$ of points on a line such that B is the pairwise distance multiset for Y. We denote the minimum and maximum of B respectively by b_m and b_M.

This problem was defined in the 1930's in the area of X-ray crystallography Patterson, (1935). In 1988 P. Lemke and M. Werman, solved this problem in pseudo-polynomial time (the running time of the presented algorithm depended on b_M) Lemke and Werman, (1988). Skiena et al. created a backtracking algorithm to solve this problem where its running time depended only on n Skiena *et al.,* (1990). In 1994 Z. Zhang, by an example, showed that the running time of backtracking algorithm in worst case is exponential Zhang, (1994). T. Dakice in his Ph.D. thesis presented a 0-1 quadratic programming model for PDP and solved it by a heuristic successive semidefinite programming algorithm Dakic, (2000). In 2005, M. Cieliebak et al. proved that Partial Digest is hard to solve for erroneous input data Cieliebak *et al.,* (2005).

In this paper we present a sufficient condition for solution of *PDP* and formulate the *PDP* as a linear zero-one programming model, such that any optimal solution of this model will be a solution of *PDP*.

A Sufficient Condition for Solution of Partial Digest Problem:

In this section we present a new point of view to the *PDP* and obtain a sufficient condition for solution of *PDP*. Suppose that there are $N=n(n+1)/2)$ line segments with lengths of $b_1,b_2,...,b_N$. We want to place them in a line interval $[0,b_M]$ such that the multiset of endpoints of these line segments equals to $B=\{b_1,b_2,...,b_N\}$. In other word, we want to produce a solution of *PDP* with endpoints of line segments. Let a line segment with length b_j, be denoted as "b_j-*segment*". It is obvious that the beginning point of b_M-segment is zero and the endpoint of this line segment is b_M. Let the variables x_j and x_{j+N} show the beginning and end points of the b_j-segment respectively in the interval $[0,b_M]$. Therefore we have $x_{j+N} - x_j = b_j$ for all j=1,2,...,N.

Corresponding Author: Reza Nadimi, Department of Applied Mathematics, Faculty of Mathematics,University of Mazandaran, Babolsar, Mazandaran, Iran
Email: nadimi@umz.ac.ir
Tel: +98-112-5342461; Fax: +98-112-5342460

We design an optimization model with $x=\{x_1, x_2,...,x_{2N}\}$ as the decision variables such that, at optimality, x has exactly (n+1) different values and the multiset of these (n+1) values is equal to B. We create the set x by eliminating the replicated members of x. (The number of different values in x is equal to the cardinality of x, $|x|$).

Each set of values of x_j 's that are between zero and b_M, and satisfy the constraints $x_{j+N} - x_j = b_j \ (j = 1,2,\cdots,N)$, is defined as a "*placement*" of line segments b_1, b_2,...,b_N in interval $[0,b_M]$. It is clear that a placement in which the number of its endpoints is not equal to (n+1) is not desirable to find a solution of PDP. Moreover, in the following example we show that it is possible to place the line segments in interval $[0,b_M]$ with (n+1) endpoints such that the multiset of the endpoints is not equal to B.

Example:

let B={2,2,2,4,4,4,6,6,8,10} be the input data of *PDP*. Then we have N=10, n=4, the target interval is: [0,10], and b_1=2, b_2=2, b_3=2, b_4=4, b_5=4, b_6=4, b_7=6, b_8=6, b_9=8, b_{10}=10.

We present two different placements of these line segments in interval [0,10] with n+1=5 endpoints, such that one of them is a solution of *PDP* but the other one is not. In the presented placement in the table(1) (placement(1)), x is equal to {0,4,6,8,10} and Δx is equal to B. Therefore, x is a solution of *PDP*.In the presented placement in the table(2) (placement(2)), x is equal to {0,4,6,8,10} but Δx=[2,2,2,4,4,6,6,8,810] is not equal to B therefore x is not a solution of *PDP*.W

Table 1: x_i 's values in the placement(1)

i	1	2	3	4	5	6	7	8	9	10
b_i	2	2	2	4	4	4	6	6	8	10
xi	4	6	8	0	4	6	0	4	0	0
x_{i+10}	6	8	10	4	8	10	6	10	8	10

Table 2: x_i 's values in the placement(2)

i	1	2	3	4	5	6	7	8	9	10
b_i	2	2	2	4	4	4	6	6	8	10
xi	8	8	0	0	0	0	4	4	0	0
x_{i+10}	10	10	2	4	4	4	10	10	8	10

Review of differences between placement(1) and placement(2) is useful to provide the rule of correct placing. In placement(1) for each $\overline{x_k}$ from x there are exactly n=4 members of x equal to x_k (see Figure 1), but in placement(2) only x_{13} is equal to 2 and there are more than 4 members of x equal to 4 (see Figure 2). In placement(2) some b_i segments coincide with each other, i.e. they have the same beginning and end points. For example b_4, b_5 and b_6 are coincided together. But in placement(1) there is no coincidence.

In the following lemma and theorem it is proved that each placement of B with (n+1) different endpoints that has no coincidence, is a solution of *PDP*.

Lemma 1:

If a placement has no coincidence, then its set of endpoints consists of at least (n+1) different values.

Proof: If a placement x does not have any coincidence, then each pair of members of x corresponds at most to one member of B. Suppose that x has r members. The number of distinct pairs of the members is greater than or equal to N ($N = |B|$). this means:

$$\binom{r}{2} \geq N = n(n+1)/2 = \binom{n+1}{2}$$

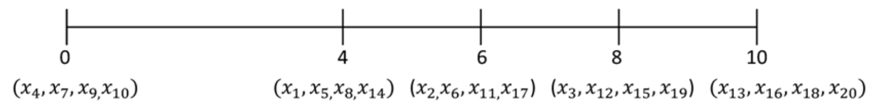
0
4
6
8
10
(x_4, x_7, x_9, x_{10})
(x_1, x_5, x_8, x_{14})
$(x_2, x_6, x_{11}, x_{17})$
$(x_3, x_{12}, x_{15}, x_{19})$
$(x_{13}, x_{16}, x_{18}, x_{20})$

Fig. 1: places of x_j 's in placement(1)

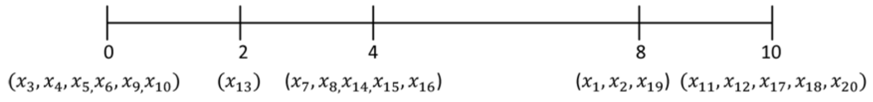
0
2
4
8
10
$(x_3, x_4, x_5, x_6, x_9, x_{10})$
(x_{13})
$(x_7, x_8, x_{14}, x_{15}, x_{16})$
(x_1, x_2, x_{19})
$(x_{11}, x_{12}, x_{17}, x_{18}, x_{20})$

Fig. 2: places of x_j 's in placement(2)

Therefore: $r \geq n+1$□

Theorem 1:

Let $\mathbf{x}$ be a placement with no coincidence and corresponding x has exactly (n+1) members, then x is a solution of *PDP*.

Proof: In a placement with no coincidence each b_j has a unique corresponding pair $(\overline{x}_k, \overline{x}_l)$, so that $\overline{x}_k = x_j$ and $\overline{x}_l = x_{j+N}$. Therefore there are N distinct pair of members of x corresponding to members of B. On the other hand there is only N members in Δx, hence there is no member of Δx that is not in B. In the other word Δx=B.□

In the next section we obtain a solution for *PDP* by looking for a placement of b_i-*segment*'s with (n+1) end points with no coincidence.

Linear zero-one programming model:

In this section we present a linear zero-one programming model which in optimality characterize a solution of *PDP*. As we mentioned in definition of the "*placement*", in any placement we have:

$$\begin{aligned} & x_{j+N} - x_j = b_j \qquad for \quad j = 1,2,\cdots,N \\ & 0 \leq x_j \leq b_M \qquad for \quad j = 1,2,\cdots,2N \\ & x_N = 0 \end{aligned} \tag{1}$$

Now, to avoid coincidence in a placement, we define a set of constraints. A coincidence occurs when two line segments with the same length have equal beginning and end points. Therefore in any placement that satisfies in following constraints we have no coincidence:

$$x_j - x_i \geq b_m \quad if \quad b_j = b_i \quad and \quad j > i \tag{2}$$

Note that b_m is the minimum of B and any solution of *PDP* satisfies in 2.

In the section 2 we saw that any placement with no coincidence that has exactly (n+1) different values is a solution for *PDP*. To obtain a placement with exactly (n+1) different values, we divide the x_i 's to (n+1) groups, such that each group contains n members with same values. We define the variables d_{ij} to specify equality or inequality of x_i and x_i. d_{ij} is a zero-one variable that is equal to 1 iff x_1=x_j and is equal to 0 iff $x_1 \neq x_j$. To avoid duplication we define d_{ij} only for $j \geq i (i, j = 1,2,\cdots,2N)$. We denote the set of d_{ij} 's by ***d***.

With respect to definition of d_{ij} we have:

$$d_{ij} = \begin{cases} 1 & if \quad x_i = x_j, \\ \\ 0 & if \quad x_i \neq x_j. \end{cases} \tag{3}$$

When we divide the x_i 's to (n+1) groups (each group with n equal values), For each i there are (n+1) members of $\mathbf{x}$ equal to x_i. In the other word we have:

$$\sum_{j>i} d_{ij} + \sum_{j<i} d_{ji} = n-1 \quad for \quad i = 1,2,\cdots,2N \tag{4}$$

With respect to theorem 2, If (x,d) satisfies in equations (1), (2), (3) and (4), then x is a solution of *PDP*. Our aim is to present the *PDP* as a linear zero-one programming model. To linearize the definition of d_{ij} consider the following optimization model:

$$\begin{aligned} &maximum: \quad f = \sum_{i<j} d_{ij} \\ &subject \quad to: \; b_M - b_M.d_{ij} + x_j - x_i \geq 0 \\ &b_M - b_M.d_{ij} - x_j + x_i \geq 0 \\ &d_{ij} = 0,1 \end{aligned} \tag{5}$$

In the next theorem we prove that any optimal solution of 5 satisfies in the (3)

Theorem 2:

If (x,d) be an optimal solution of (5), then (x,d) satisfies in the (3).

Proo: Suppose that (x,d) be an optimal solution of (5). When x_i=x_j, d_{ij}=1 satisfies in the constraints of (5), and with respect to optimality we have d_{ij}=1. If $x_i \neq x_j$ with respect to constraints of (5) we have b_M-b_M.d_{ij} >0 and d_{ij} = 0. When d_{ij} = 1, constraints of (5) imply -x_j+x_i≥0 and x_j-x_i≥0 therefore -x_i=x_i. If d_{ij}=0 with respect to optimality we have $x_i \neq x_j$ because $x_i = x_j$ and d_{ij}=1 satisfy in (5) and have better objective function. □

Now we present the *PDP* as a linear zero-one programming model. The final model is as follow:

$$\begin{aligned} &maximum: \quad f = \textstyle\sum_{i<j} d_{ij} \\ &\text{subject to:} \\ &b_M - b_M.d_{ij} + x_j - x_i \geq 0 \quad for \quad j>i, \quad i,j = 1,2,\cdots,2N \\ &b_M - b_M.d_{ij} - x_j + x_i \geq 0 \quad for \quad j>i, \quad i,j = 1,2,\cdots,2N \\ &x_{j+N} - x_j = b_j \quad for \quad j = 1,2,\cdots,N \\ &\textstyle\sum_{j>i} d_{ij} + \sum_{j<i} d_{ji} = n-1 for \quad i = 1,2,\cdots,2N \\ &d_{ij} = 0,1 \; for \quad j>i, \quad i,j = 1,2,\cdots,2N \\ &0 \leq x_j \leq b_M \quad for \quad j = 1,2,\cdots,2N \\ &x_N = 0 \end{aligned}$$

This model can be solved by any zero-one programming or integer programming algorithms.

Conclusion:

In this paper we presented a sufficient condition for the solution of *PDP* and then we formulated the *PDP* az a zero-one programming model, that can be solved by any zero-one programming or integer programming algorithms. The computational class of partial digest problem is an open problem. Neither a proof of NP-hardness nor a polynomial time algorithm is known for this problem.

REFERENCES

Cieliebak, M., S. Eidenbenz and P. Penna, 2005. *Partial Digest is hard to solve for erroneous input data*. Theor. Comput. Sci., 349(3): 361-381.

Dakic, T., 2000. *On the Turnpike Problem*. PhD thesis, Simon Fraser University.

Lemke, P., M. Werman, 1988. *On the complexity of inverting the autocorrelation function of a finite integer sequence, and the problem of locating n points on a line, given the unlabelled distances between them*. Preprint 453, Institute for Mathematics and its Application IMA.

Patterson, A.L., 1935. A direct method for the determination of the components of interatomic distances in crystals, Zeitschr. Krist., 90: 517-542.

Skiena, S.S., W. Smith and P. Lemke, 1990. *Reconstructing sets from interpoint distances*. In Proc. of the 6th ACM Symposium on Computational Geometry (SoCG 1990): 332-339.

Zhang, Z., 1994. *An exponential example for a partial digest mapping algorithm*. Journal of Computational Biology, 1(3): 235-239.