On the Complexity and Approximation of Non-unique Probe Selection Using d-Disjunct Matrix

My T. Thai [*] Taieb Znati [†]

Abstract

In this paper, we studied the MINimum-d-Disjunct Submatrix (MIN-d-DS), which can be used to select the minimum number of non-unique probes for viruses identification. We prove that MIN-d-DS is NP-hard for any fixed d. Using d-disjunct matrix, we present an $O(\log k)$-approximation algorithm where k is an upper bound on the maximum number of targets hybridized to a probe. We also present a $(1+(d+1)\log n)$-approximation algorithm to identify at most d targets in the presence of experimental errors. Our approximation algorithms also yield a linear time complexity for the decoding algorithms.

Keywords: non-unique probe, non-adaptive group testing, pooling designs, d-disjunct matrix

[*] Computer and Information Science and Engineering Department. University of Florida. Gainesville, FL, 32611. Email: mythai@cise.ufl.edu.

[†] Support in part by National Science Foundation under grant CCF-0548895. Department of Computer Science. University of Pittsburgh. Pittsburgh, PA, 15215. Email: znati@cs.pitt.edu

1 Introduction

Non-unique probe selection is a fundamental problem in computational molecular biology for target identification (Moret and Shapiro, 1985; Steinfath et al., 2000; Borneman et al., 2001; Wang and Seed, 2003; Rahmann, 2002, 2003; Gao et al., 2006; Thai et al., 2007; Li et al., 2005; Thai, 2007). A *probe* is a short oligonucleotide of size 8-25, used for identifying targets in a biological sample through hybridizations using DNA microarrays. A probe is called a *unique* probe if it hybridizes to only one specific target; otherwise, called a *non-unique* probe. Since unique probes have a strong separability of targets, identifying the presence of targets in a sample by using unique probes is straightforward. However, finding unique probes for every target is a difficult task due to the strong similarity of closely related targets. Considering a set of n targets and a sample containing at most $d \geq 1$ of these targets, Schilep, Torney, and Rahman (Schliep et al., 2003) introduced a method using non-unique probes with group testing techniques to identify at most d targets in the following three steps:

1. Find a large set of non-unique probes as candidates and let a binary matrix M represent the probe-target hybridizations with rows labeled by probes and columns labeled by targets; that is, $M[i, j] = 1$ if probe p_i hybridizes to target t_j; otherwise, $M[i, j] = 0$.

2. Select a minimum subset of probes obtained in Step 1 so that these probes can identify up to d targets. In other words, find a minimum submatrix H of M with the same number of columns.

3. Decode the presence or absence of targets in a sample from the hybridization results, called *test outcomes* V, where V is a column vector. If $V_i = 1$, then probe p_i hybridizes to at least one target in the sample; otherwise, $V_i = 0$.

In this paper, we study the steps 2 and 3, that is, finding a minimum submatrix H with an efficient decoding algorithm. Unsurprisingly, these two problems are highly related. The design of submatrix H must satisfy the following two conditions: (1) All unions of up to d columns in H must be distinct. Here, the union of a set of columns is equal to the boolean sums of these columns. (2) The time complexity of a decoding algorithm yielding from the design of H must be efficient.

Based on the classical theory of nonadaptive group testing, Schliep *et al.* and Klau *et al.* (Schliep et al., 2003; Klau et al., 2004) proposed a heuristic to construct a $\bar{d}$-separable submatrix H. A binary matrix H is called $\bar{d}$-separable iff all unions of at most d columns are distinct. However, it is hard to decode the test outcomes from a $\bar{d}$-separable matrix (Du and Hwang, 2006). Therefore, in this paper, we consider to use a d-disjunct submatrix instead. A binary matrix H is called d-disjunct iff any union of d columns cannot contain any other column. Decoding the test outcomes from a d-disjunct matrix is very easy with linear time complexity (Du and Hwang, 2006). This introduces the following minimization problem:

MIN-d-DS (MINimum-d-Disjunct Submatrix): Given an $m \times n$ binary matrix M where rows represent the probes and columns represent the targets, find a $h \times n$ submatrix

H with the same number of columns such that H has a minimum number of rows, i.e., h is minimum, and H is a d-disjunct matrix.

In this paper, we show that MIN-d-DS is NP-hard for any fixed $d \geq 1$. We then propose an $O(\log k)$-approximation algorithm for the MIN-d-DS problem. Moreover, the presence of errors due to the noise of hybridizations makes the problem become even harder. With the experimental errors, the test outcomes may consist of *false negatives* or *false positives.* In the former, a test i yields a negative outcome, i.e. $V_i = 0$ when a probe p_i *does* hybridize to at least one target in the sample. Likewise, in the latter, a test i yields a positive outcome, i.e. $V_i = 1$ when a probe p_i does *not* hybridize to any targets in the sample. In this case, we present a $(1 + (d + 1)\log n)$-approximation algorithm to correctly identify up to d targets with the presence of at most k errors in experiments. Unfortunately, the decoding algorithm in the case of error tolerance become much more complicated. In this paper, we also present a solution and its hardness to this problem.

2 Main Results

2.1 Complexity

Theorem 1 *MIN-d-DS is NP-hard for any fixed* $d \geq 1$

Proof: It has been proved that MIN-1-DS is NP-hard (Du and Hwang, 2006). We now show how to reduce MIN-1-DS to MIN-d-DS for any fixed $d > 1$ in polynomial time.

Decision Version of MIN-d-DS: Given an $m \times n$ binary matrix M and a positive integer h $(1 \leq h \leq m)$, determine whether M contains an $h \times n$ d-disjunct submatrix H.

Now, we first show how to reduce MIN-1-DS to MIN-2-DS. Consider an $m_1 \times n_1$ matrix M_1 with a 1-disjunct submatrix $h_1 \times n_1$ H_1. We construct an instance of MIN-2-DS as follows:

Set $n = n_1 + 1$, $m = m_1 + (n_1 + 1)$, and $h = h_1 + (n_1 + 1)$. Let us label the columns of M by all columns in M_1 plus one new column N and label the rows of M by all rows in M_1 plus $(n_1 + 1)$ new rows. Now, define the elements in the new cells from those new rows and columns as follows:

1. For $i = 1, \ldots, m_1$, set $M[i, N] = 0$. Note that N is a new column

2. For $j = 1, \ldots, n_1$, set $M[m_1 + 1, j] = 0$ and $M[m_1 + 1, N] = 1$. Let label this row as B.

3. For the last n_1 rows and n_1 columns, enter one at the diagonal and for $j = m_1 + 2, \ldots, m_1 + (n_1 + 1)$, set $M[j, N] = 1$. Let label these n_1 rows as C.

Figure 1 shows an example of constructing an MIN-d-DS instance from MIN-1-DS for $d = 2$ and $d = 3$.

First, suppose that M_1 contains a MIN-1-DS H_1 with size $h_1 \times n_1$. Let a matrix H with size $h \times n$ where $h = h_1 + (n_1 + 1)$ (H is a union of H_1 and all new rows and new columns). Therefore, H is a MIN-2-DS of M.

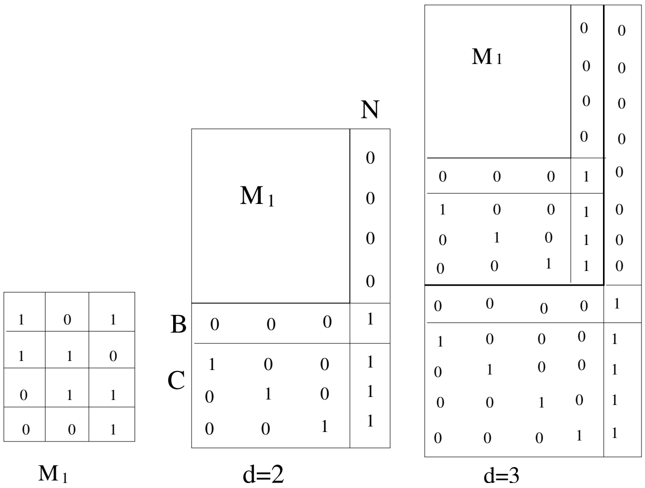

Figure 1: A Construction of an instance of MIN-d-DS from MIN-1-DS where $d = 2$ and $d = 3$

Conversely, suppose M has an $h \times n$ 2-disjunct submatrix H. Consider a collection $\mathcal{P}$ of all probes corresponding to row vectors of H. Since H is a 2-disjunct matrix, H must satisfy this condition (*): for any column t_0 and other 2 columns t_1, t_2 in M, there exist a row p in H such that $H[p, t_0] = 1$ and $H[p, t_i] = 0$ for all $i = \{1, 2\}$.

Consider the following three cases:

- Case (i): $t_0 \in N$ and $t_i \in M_1$. Then $\mathcal{P}$ must contain B. Otherwise, the union of 2 columns t_i will contain t_0, contradicting to the 2-disjunctness of H.
- Case (ii): $t_0, t_1, t_2 \in M_1$. Then $\mathcal{P}$ must contain C in order to satisfy (*).
- Case (iii): $t_0 \in M_1$, $t_1 \in M_1$, and $t_2 \in N$. Note that B and C cannot satisfy condition (*) in this case. Therefore, $\mathcal{P} \cap B \cap C$ must satisfy (*). Notice that $\mathcal{P} \cap B \cap C$ contains exactly h_1 rows in M_1. In this case, all entries in column N are zeros. Hence, condition (*) is satisfied iff for any pair of columns t_0 and t_1 in M_1, there exist a row p (of h_1 rows) such that $M[p, t_0] = 1$, $M[p, t_1] = 0$. Thus, these h_1 rows will form a 1-disjunct submatrix.

In general, for any fixed d, set $n = n_1 + (d - 1)$, $m = m_1 + \frac{(2n_1+d)(d-1)}{2}$, and $h = h_1 + \frac{(2n_1+d)(d-1)}{2}$. Using the induction method, we can conclude that MIN-d-DS is NP-hard. □

2.2 An $O(\log k)$-Approximation Algorithm

Since each probe cannot hybridize to too many targets, we set an upper bound k on the maximum number of targets that each probe can hybridize to. In this section, we present an $O(\log k)$-approximation algorithm to construct the submatrix H.

Recall that H is a d-disjunct matrix iff the union of d columns does not contain any other columns. Therefore, we have:

Fact 1: *H is a d-disjunct matrix iff for any $(d+1)$ columns $t_0, t_1, \ldots, t_d$, there exists a row p_i such that an entry $H[i,0] = 1$ and $H[i,j] = 0$ for all $j = 1 \ldots d$.*

Such a row p_i is said to *cover* a pair $(t_0, < t_1, ..., t_d >)$. Hence, to construct matrix H from M, we need to find a set of probes covering all pairs $(t_j, < t_k, ..., t_{k+d-1} >)$ where $j, k, \ldots, (k+d-1) \in \{1, ..., n\}$ and $j \notin \{k, \ldots, (k+d-1)\}$. It is easy to see that this is a special case of set cover problem (Vazirani, 2001) where the collection $\mathcal{S}$ is a set of possible pairs $(t_j, < t_k, ..., t_{k+d-1} >)$.

The proposed algorithm consists of two main steps as follows:

Algorithm 1:

1. Step 1: Find a set cover $\mathcal{C}$ of M where $\mathcal{C}$ consists a set of probes such that these probes cover all the targets. *Note that in this step, we treat M as an instance of the set cover problem. In this context, a probe covers a target if it hybridizes to this target.*

2. Step 2: If $\mathcal{C}$ is a set of corresponding probes in a d-disjunct submatrix, then return $\mathcal{C}$. Otherwise, we will extend $\mathcal{C}$ to cover all other pairs as follows:

   - While there exists at least one pair $(t_0, < t_1, \ldots, t_d >)$ not covered, choose a probe $p \notin \mathcal{C}$ such that p covers at most the non-covered pairs.
   - Add p into $\mathcal{C}$

Lemma 1 *Let $\mathcal{H}^*$ represent the optimal solution of MIN-d-DS problem and $\mathcal{C}^*$ be the optimal solution of corresponding set cover problem. Then $|\mathcal{C}^*| \leq |\mathcal{H}^*|$.*

Theorem 2 *Algorithm 1 will obtain a solution within a factor of $O(\log k)$.*

Proof. Let $\mathcal{H}$ be our obtained solution. Let $\mathcal{C}$ be a set of probes selected in step 1 and $\mathcal{C}'$ be a set of probes selected in step 2. Hence, $\mathcal{H} = \mathcal{C} \cup \mathcal{C}'$.

Note that in step 1, each probe can cover k $\binom{n-k}{d}$ pairs $(c_0, < c_1, \ldots, c_d >)$. Hence $\mathcal{C}$ can cover at least k $\binom{n-k}{d}$ pairs. In step 2, for each probe we pick, it can cover at most k $\binom{k-1}{d}$ pairs. Therefore, the instance of this set cover problem in Step 2 has at most $k^{(d+1)}$ elements. Hence $|\mathcal{C}'| \leq 1 + \log(k^{d+1})|\mathcal{C}^*| \leq 1 + (d+1)\log k|\mathcal{H}^*|$. It is well-known that the greedy algorithm for the set cover at the first step has an approximation ratio of $O(\log k)$.

Combining both steps, we obtain an approximation ratio of $O(\log k)$ where d is fixed. □

2.3 Error Tolerance

In this section, we study a more general problem of non-unique probe selection. In particular, we consider the problem where there exists at most k experimental errors and we do not set an upper bound on the number of targets that each probe can hybridize to. Let us first introduce the following definitions:

Definition 1 Hamming distance*: The Hamming distance of two column vectors is defined as the number of different components between them.*

Definition 2 k**-error-correcting***: A matrix H is said to be k-error-correcting if the Hamming distance of any two unions of d columns must be at least $2k+1$*

Definition 3 (d,k)***-disjunct****: H is called (d,k)-disjunct if for any column t_j, t_j must have at least $k+1$ 1-entries not contained in the union of other d columns.*

In order to identify all the targets with at most k experimental errors in hybridizations, H must have the k-error-correcting property. Note that by the definition, it is enough to see that the (d,k)-disjunct matrix is a k-error-correcting $d-$separable matrix. Therefore, in the case of error tolerance, we study the following two problems:

MIN-(d,k)**-DS (Minimum** (d,k)**-Disjunct Submatrix)**: Given an $m \times n$ binary matrix M, find a minimum (d,k)-disjunct $h \times n$ submatrix H where $h \leq m$.

Decoding Algorithm: Given a (d,k)-disjunct matrix H, a sample s, and the test outcomes vector V, find an algorithm to identify all the targets t_j present in s where there exists at most k experimental errors.

Since the decoding algorithm is complicated in the error tolerance, we consider two cases of the sample space: (1) Let $S(d,n)$ be a sample space such that for a given sample $s \in S$, s will have *exactly* d targets from a set of n targets. (2) Let $S(\bar{d},n)$ be a sample space such that for a given sample $s \in S$, s will contain *at most* d targets.

In this section, we present solutions on finding the (d,k)-disjunct submatrix H and decoding the hybridization results for both $S(d,n)$ and $S(\bar{d},n)$ sample spaces.

2.3.1 $S(d,n)$ Sample Space

Let a_{ij} denote an entry at cell $M[i,j]$. Let $k' = k+1$. The problem of finding a minimum (d,k)-disjunct submatrix is equivalent to the following integer programming:

$$\begin{array}{ll}
\text{min} & \sum_{i=1}^{m} x_i \\
\text{subject to} & \sum_{i=1}^{m} a_{ij} x_i \geq k' \text{ for } j = 1,2,...,n \\
 & \sum_{i=1}^{m} (|a_{ij} - (a_{ik} + ... + a_{i(k+d-1)})|) x_i \geq k' \text{ for } j,k,...,k+d-1 \in \{1,...,n\} \\
 & \qquad \text{and } j \notin \{k,...,k+d-1\} \\
 & x_i \in \{0,1\} \text{ for all } i = 1,2,...,m
\end{array} \tag{1}$$

where $x_i = 1$ if probe p_i is selected; otherwise, $x_i = 0$.
Remarks:

- The first constraint is to make sure that each target is covered by at least k' probes
- The second constraint is to make sure that each pair $(t_j, < t_k, ..., t_{k+d-1} >)$ can be covered by at least k' probes

Now, let E be a set of all pairs $(t_j, < t_k, ..., t_{k+d-1} >)$. Let matrix B be a binary matrix where rows represent m probes p_i and columns represent the pairs. Let b_{ij} be each entry at cell $B[i, j]$. $b_{ij} = 1$ iff probe p_i covers the pair at column j.

In a (d, k)-disjunct matrix, for any pair $(t_j, < t_k, ..., t_{k+d-1} >)$, we can find at least $k + 1$ rows such that the intersection entries of these rows at column t_j are 1 whereas the intersection entries of these rows at d columns $< t_k, ..., t_{k+d-1} >$ are all 0. Therefore, the integer programming (1) should be equivalent to the following:

$$\begin{array}{ll} \text{min} & \sum_{i=1}^{m} x_i \\ \text{subject to} & \sum_{i=1}^{m} b_{ij} x_i \geq k' \text{ for } j = 1, 2, ..., |E| \\ & x_i \in \{0, 1\} \text{ for all } i = 1, 2, ..., m \end{array} \tag{2}$$

To solve (2), at each iteration, we select a probe covering the most *unsatisfied* pairs. A pair is unsatisfied if it has not been covered by at least k' probes yet. The details of this algorithm are described in Algorithm 2.

Algorithm 2: Constructing a (d, k)-disjunct Submatrix

```
$I \leftarrow \{1, 2, ..., m\}$
$J \leftarrow \{1, 2, ..., |E|\}$
$H \leftarrow 0; P \leftarrow \emptyset$
while $J \neq \emptyset$ do
   Find $i_0 \in I$ such that
   $\sum_{j \in J} b_{i_0 j} = \max_{i \in I} \sum_{j \in J} b_{ij}$
   $P \leftarrow P \cup \{i_0\}$
   $I \leftarrow I - \{i_0\}$
   $J \leftarrow J - \{j \mid \sum_{i \notin I} b_{ij} \geq k'\}$
end while
for $j = 1...|P|$ do
   for $i = 1...m$ do
      if $p_i \in P$ then
         $H[j, :] = M[i, :]$
      end if
   end for
end for
Return $H$
```

Theorem 3 *Algorithm 2 produces an approximation solution within a factor of* $1 + (d + 1)\log n$

Proof. Since the proof is similar to that of the set cover problem (Vazirani, 2001), we omit it here.

□

We now consider the decoding algorithm.

Lemma 2 *(Du and Hwang, 2006) Suppose testing is based on a* (d, k)*-disjunct matrix. If the number of error tests is no more than k, then the number of negative results containing a target is always smaller than that of the number of negative results containing a non-target.*

Proof. For the convenience of readers, we present the proof of Lemma 2 here. Let i be a target and j be other items in a hybridization (but not a target). Suppose the number of negative hybridization containing i is l. Then these l hybridization must receive error tests. Therefore, there are at most $k-l$ error tests turning negative outcomes to positive outcomes. Moreover, we note that if no error exists, the number of negative results containing j is at least $k + 1$ by the definition of (d, k)-disjunctness. Hence, the number of negative results containing j is at least $(k + 1) - (k - l) = l + 1 > l$

□

From the above lemma, we see that to decode the targets from testing based on (d, k)-disjunct matrix for $S(d, n)$ sample space, we only need to compute the number of negative results containing each item and select d smallest ones. This decoding algorithm has an $O(hn)$-time complexity.

2.3.2 $S(\bar{d}, n)$ Sample Space

In the $S(\bar{d}, n)$ sample space, the decoding algorithm is much more complicated. Although Lemma 2 still holds in $S(\bar{d}, n)$, we do not know how many smallest one we should select. Fortunately, Du and Hwang (Du and Hwang, 2006) have proven the following lemma:

Lemma 3 *(Du and Hwang, 2006) There exists a decoding algorithm for a k-error-correcting d-disjunct matrix H, running in time* $O((n + h)h^k)$ *where h is the number of rows (selected probes) in H.*

where a k-error-correcting d-disjunct matrix is defined as follows:

Definition 4 **k-error-correcting d-disjunct matrix***: A matrix H is called k-error-correcting d-disjunct matrix if H is d-disjunct and the Hamming distance between two union of* at most *d columns is at least* $2k + 1$.

Thus, we need to find a minimum k-error-correcting d-disjunct submatrix H in order to find a possible decoding algorithm in $S(\bar{d}, n)$. Once we construct a k-error-correcting

d-disjunct submatrix H, we can use a decoding algorithm mentioned in (Du and Hwang, 2006), of which the time complexity is $O((n+h)h^k)$. This time complexity is quite high.

Interestingly, the following lemma gives an efficient way to construct such a k-error-correcting d-disjunct submatrix H with a linear decoding algorithm.

Lemma 4 *Every $(d, 2k)$-disjunct matrix is k-error-correcting d-disjunct matrix*

Proof. Given a matrix H as a $(d, 2k)$-disjunct matrix. Since H is $(d, 2k)$-disjunct, H is d-disjunct. Therefore, for any two different subsets of at most d columns in H, there must be one not contained by the other. By the definition of $(d, 2k)$-disjunct matrix, the union of the former contains at least $2k + 1$ 1-entries not appearing in the union of the latter. This implies that the Hamming distance between these two unions is at least $2k + 1$. Hence H is also an k-error-correcting matrix.

□

From Lemma 4, instead of directly constructing a k-error-correcting d-disjunct submatrix H, we find a $(d, 2k)$-disjunct submatrix H by using Algorithm 2.

Now, we consider the decoding algorithm using a $(d, 2k)$-disjunct submatrix H.

Lemma 5 *Suppose testing done on a $(d, 2k)$-disjunct matrix H with at most k errors, an item is a target iff it appears in at most k negative results.*

Proof. Since there are at most k errors, a target can appear in at most k negative results (due to errors). However, a non-target item appears in at least $2k + 1 - k = k + 1 > k$ negative results. It implies that an item is a target iff it appears in at most k negative results.

□

Based on the above Lemma 5, the decoding algorithm becomes quite simple. For each item, we just need to count the number of negative results containing it. If this number is less than k, then this item must be a target. Hence, the time complexity of this decoding algorithm is $O(hn)$, which is linear.

References

Garey MR, Johnson DS (1979) *Computers and Intractability - A Guide to the Theory of NP-completeness,* W.H. Freeman & Co.

Moret BME, Shapiro HD (1985) *On Minimizing a set of tests*, SIAM Journal on Sceientific and Statistical Computing, vol. 6, pp. 983–1003

Schliep A, Torney DC, Rahmann S (2003) *Group Testing with DNA Chips: Generating Designs and Decoding Experiments*, Proceedings of the Computational Systems Bioinformatics (CSB'03)

Li Y, Thai MT, Liu Z, Wu W (2005) *Protein-Protein Interaction and Group Testing in Bipartite Graphs*, International Journal of Bioinformatics Research and Applications (IJBRA), vol. 1, no. 4, pp. 414–419

Gao H, Hwang FK, Thai MT, Wu W, Znati T (2006) *Construction of d(H)-Disjunt Matrix for Group Testing in Hypergraphs*, J. of Combinatorial Optimization, vol. 12, no. 3, pp. 297–301

Klau GW, Rahmann S, Schliep A, Vingron M, Reinert K (2004)*Optimal Robust Non-unique Probe Selection Using Integer Linear Programming*, J. of Bioinformatics, vol. 20, pp. 186–193

Du D-Z, Hwang FK (2006) *Pooling Designs: Group Testing in Molecular Biology*, World Scientific, Singapore

Steinfath, OB́rien J, Seidel H, Meier-Ewert S, Lehrach H, Radelof U (2000) *Information theoretical probe selection for hybridization experimetnts*, Bioinformatics, 16(10):890–898

Borneman J, Chrobak M, Vedova GD, Figueroa A, Jiang T (2001) *Probe selection algorithms with applications in the analysis of microbial communities*, Bioinformatics, 17 Suppl 1:339–48

Wang X, Seed B (2003) *Selection of oligonucleotide probes for protein coding sequences*, Bioinformatics, 19, 796–802

Rahmann MS (2002) *Rapid large-scale oligonucleotide selection for microarrays*, Proceedings of the First IEEE Computer Society Bioinformatics conference(CSB). Stanford, pp. 54–63

Rahmann S (2003) *Fast and sensitive probe selection for DNA chips using jumps in matching statistics*, Proceedings of the 2nd IEEE Computational Systems Bioinformatics Conference(CSB), Stanford, pp. 57–64

Thai MT, MacCallum D, Deng P, Wu W (2007) *Decoding Algorithms in Pooling Designs with Inhibitors and Fault Tolerance*, International Journal of Bioinformatics Research and Applications (IJBRA), vol. 3, no. 2, pp. 145–152

Thai MT, Deng P, Wu W, Znati T (2007) *Approximation Algorithms of Non-unique Probes Selection for Biological Target Identification*, in Proceedings of Conference on Data Mining, Systems Analysis and Optimization in Biomedicine

Vazirani VV (2001) *Approximation Algorithms*, Springer-Verlag