CHAPTER 1

ALGORITHMIC PERSPECTIVES OF THE STRING BARCODING PROBLEMS

Sima Behpour[1], **Bhaskar DasGupta**[1]
[1]Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA.

1.1 INTRODUCTION

Let Σ be a finite alphabet. A string is a concatenation of elements of Σ. The length of a string x, denoted by $|x|$, is the number of the characters that constitute this string. Let $\mathcal{S}$ be a set of strings over Σ. The simplest "binary-valued version" of the string barcoding problem discussed in this chapter is defined as follows [3, 17]:

Problem name: String barcoding problem ($\mathrm{SB}^{\Sigma}(1)$).

Definition of a barcode: for a string s and a set of strings $\mathcal{T} = \{t_0, t_1, \ldots, t_{m-1}\}$,
$\mathsf{barcode}(s, \mathcal{T})$ is the boolean vector (b_0, b_1, b_{m-1}) where $b_i = \begin{cases} 1, & \text{if } t_i \text{ is a } \textit{substring} \text{ of } s \\ 0, & \text{otherwise} \end{cases}$.

	Σ = {A, C, T, G}, $\mathcal{T}$ = {A, CC, TTT, GT}, s =ACC			
	t_0 =A	t_1 =CC	t_2 =TTT	t_3 =GT
s =ACC	(1)	1	0	(0)
	s is a substring of t_0			s is not a substring of t_3

Input: A set of strings $\mathcal{S}$ over Σ.

Valid solutions: A set of strings $\mathcal{T}$ such that (see Fig. 1.1):

$$\forall\, s, s' \in \mathcal{S}\colon\ s \neq s' \ \Leftrightarrow\ \mathsf{barcode}(s, \mathcal{T}) \neq \mathsf{barcode}(s', \mathcal{T})$$

Objective: *minimize* the length of the barcode $|\mathcal{T}|$.

Σ = {A, C, T, G}, $\mathcal{T}$ = {A, CC, TTT, GT}, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$

	t_0 =A	t_1 =CC	t_2 =TTT	t_3 =GT
S_1 =AAC	1	0	0	0
S_2 =ACC	1	1	0	0
S_3 =GGGG	0	0	0	0
S_4 =GTGTGG	0	0	0	1
S_5 =TTTT	0	0	1	0

Figure 1.1 An example of a valid barcode.

The basic string barcoding problem $\mathrm{SB}^{\Sigma}(1)$ was generalized in [3] to a "grouped" string barcoding problem $\mathrm{SB}^{\Sigma}(\kappa)$ in the following manner:

Problem name: grouped string barcoding problem ($\mathrm{SB}^{\Sigma}(\kappa)$).

Definition of a κ-string: a κ-string is a collection of at most κ strings.

Definition of a barcode: for a string s and a set of κ-strings $\mathcal{T} = \{t_0, t_1, \ldots, t_{m-1}\}$, $\mathsf{barcode}(s, \mathcal{T})$ is the boolean vector $(b_0, b_1, \ldots, b_{m-1})$ where:

$$b_i = \begin{cases} 1, & \text{if there exists a } t \in t_i \text{ for some } i \text{ such that } t \text{ is a } \textit{substring} \text{ of } s \\ 0, & \text{otherwise} \end{cases}$$

Input: a set $\mathcal{S}$ of strings over Σ.

Valid solutions: a set of κ-strings $\mathcal{T}$ such that

$$\forall\, s, s' \in \mathcal{S}\colon\ s \neq s' \ \Leftrightarrow\ \mathsf{barcode}(s, \mathcal{T}) \neq \mathsf{barcode}(s', \mathcal{T})$$

Objective: *minimize* the length of the barcode $|\mathcal{T}|$.

Finally, the binary-valued basic version of the string barcoding problem $\mathrm{SB}^{\Sigma}(1)$ is actually a special case of the more general "integral-valued" version defined as follows [4]:

Problem name: Minimum cost probe set with threshold r ($\text{MCP}^{\Sigma}(r)$).

Definition of a r-barcode: for a string s and a set of strings $\mathcal{T} = \{t_0, t_1, \ldots, t_{m-1}\}$, r-barcode$(s, \mathcal{T})$ is the integer vector $(b_0, b_1, \ldots, b_{m-1})$ where

$$b_i = \min\{r, \text{number of occurrences of } t_i \text{ in } s\}$$

Σ = {A, C, T, G}, $\mathcal{T}$ = {A, CC, AC, G}, s =ACCCCA, r = 2

	t_0 =A	t_1 =CC	t_2 =AC	t_3 =G
s =ACCCCA	(2)	(2)	1	0
	$\min\{r, 2\}$	$\min\{r, 3\}$		

Input: Sets $\mathcal{S}$ and $\mathcal{P}$ of strings over Σ and an integer $r > 0$.

Valid solutions: A set of strings $\mathcal{T} \subseteq \mathcal{P}$ such that

$$\forall\, s, s' \in \mathcal{S}\colon\ s \neq s' \ \Leftrightarrow\ r\text{-barcode}(s, \mathcal{T}) \neq r\text{-barcode}(s', \mathcal{T})$$

Objective: *Minimize* the "length" of the barcode $|\mathcal{T}|$.

Note that if $\mathcal{P}$ is the set of *all* substrings of all strings in $\mathcal{S}$ then $\text{MCP}^{\Sigma}(1)$ is precisely $\text{SB}^{\Sigma}(1)$. Inclusion relationships among the various barcoding problems defined above is shown in Fig. 1.2.

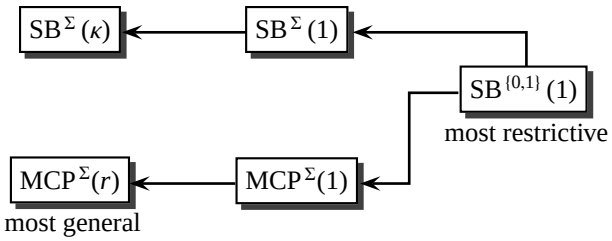


Figure 1.2 Inclusion relationships among the various barcoding problems.

In the rest of this chapter, $\mathcal{OPT}(I)$ (or simply $\mathcal{OPT}$ when I is clear from the context) will denote the optimum value of the objective function for the maximization or minimization problem under consideration. A ε-approximate solution (or simply a ε-approximation) of a minimization (respectively, maximization) problem is a solution with an objective value no larger than (respectively, no smaller than) ε times (respectively, $1/\varepsilon$ times) the value of the optimum; an algorithm of *performance* or *approximation ratio* ε produces an ε-approximate solution. A problem is ε-inapproximable under a certain complexity-theoretic assumption means that the problem does not admit a polynomial-time ε-approximation algorithm assuming that the complexity-theoretic assumption is true. We assume that the reader is familiar

with basic data structures and algorithmic methods found in graduate level algorithms textbooks such as [5].

• **Motivating biological applications**

Applications of barcoding techniques range over a diverse range of applications such as *rapid pathogen identification in epidemic outbreaks*, *database compression*, *point-of-care medical diagnosis* and *monitoring of microbial communities in environmental studies*. A generic high-level description of most of these applications involving identification of microorganisms or similar entities is as follows. The identification is performed by synthesizing the Watson-Crick complements of the barcodes on a DNA microarray and then hybridizing the fluorescent labeled DNA extracted from the unknown microorganism to the microarray. Assuming perfect hybridization, the hybridization pattern can be viewed as a string of zeros and ones, which in our terminology is the barcode of the microorganism. By definition, the barcodes corresponding to a set of microorganisms are distinct and thus the barcodes uniquely identify the organisms. Two specific applications of this nature are discussed below.

Pathogen identification in epidemic outbreaks: In the outbreak of an epidemic, possibly as a result of biological warfare, there is an urgent need to identify the pathogen and the family it belongs to *as early as possible*. Such *First Responder Pathogen Detection Systems* (FRPDS) must be able to recognize pathogens from *minute* amounts of genetic material. To enable reliable detection, one usually first amplifies the collected genetic material using high-efficiency techniques such as the *Multiplex Polymerase Chain Reaction*. Classical approaches to pathogen detection, based on sequencing and direct microarray hybridization [14, 21], are practically applicable only when the number of candidate pathogens is small. In a primer-based FRPDS, once the amplimers have been extracted, barcoding techniques can be used to efficiently generate *short* robust signatures (barcodes) via substrings (distinguishers) that can be detected by DNA or RNA hybridization chips such as DNA tag arrays. The compact size of the barcodes optimizes cost of designing the hybridization array, reduces database size and allows one to perform extremely rapid comparisons against large databases using a significantly small amount of memory. Moreover, robust barcoding can be error tolerant and may work with minute traces of the unknown sample. This was a main motivation for investigating various versions of the barcoding problems in publications such as [3, 6, 7, 17].

Monitoring microbial communities: To minimize the number of oligonucleotide probes needed for analyzing populations of ribosomal RNA gene clones by hybridization experiments on DNA microarrays, the $\mathrm{MCP}^{\Sigma}(r)$ problem was formulated and used in [4]; the probes were selected in [4] from a pre-specified set ($\mathcal{P}$ in our notation).

In real applications, the string barcoding problems are further complicated by factors such as the occurrence of a substring may be approximate due to several reasons such as noise and experimental errors. To address these issues, the *robustness* of de-

signed barcodes are improved by using the grouped string barcoding problem $\mathrm{SB}^{\Sigma}(k)$ integrates the basic barcoding problem with group testing approach for designing probes [18] by allowing groups of distinguishers to differentiate between strings.

1.2 SUMMARY OF ALGORITHMIC COMPLEXITY RESULTS FOR BARCODING PROBLEMS

For the case when the alphabet Σ is allowed to have arbitrarily many symbols, the NP-hardness of $\mathrm{MCP}^{\Sigma}(1)$ follows from a result in Garey and Johnson [11, pp. 71] via a reduction from the 3-dimensional matching problem, and heuristics algorithms for $\mathrm{MCP}^{\Sigma}(1)$ were discussed by Moret and Shapiro in [15].

The (unweighted) *Minimum Set Cover* (MSC) problem is a well-known combinatorial problem that is defined as follows:

Input: A universe of n elements $U = \{u_1, u_2, \ldots, u_n\}$ and a collection of m sets $\Delta_U = \{S_1, S_2, \ldots, S_m\}$ with $\bigcup_{j=1}^{m} S_j \supseteq U$.

Valid Solutions: A subset of indices $I \subseteq \{1, 2, \ldots, m\}$ of selected sets such that:

$$\forall\, u_i \in U\colon \left| j \in I \,:\, u_i \in S_j \right| \geq 1$$

Objective: *Minimize* the number of selected sets $\left| I \right|$.

Let α denote the maximum number of elements in any set in Δ, *i.e.*, let $\alpha = \max_{i \in \{1,2,\ldots,m\}} \left\{ \left| S_i \right| \right\}$. A well-known greedy approach for solving MSC, shown in Fig. 1.3, repeatedly selects a new set that covers a maximum number of "not yet covered" elements. This algorithm is known to have an approximation ratio of $(1 + \ln \alpha)$ [13, 20] and can be easily implemented to run in $O(n + m \log m)$ time.

$I = \emptyset$, uncovered $= U$
while uncovered $\neq \emptyset$ **do**
  select an index $j \in \{1, 2, \ldots, m\} \setminus I$ that maximizes $\left|\text{uncovered} \cap S_j\right|$
  uncovered = uncovered $\setminus S_j$; $I = I \cup \{j\}$
endwhile

Figure 1.3 A greedy algorithm for solving MSC [13].

For three strings x, y and z, we say x "distinguishes" y from z if and only if x is a substring of exactly one of the two strings y and z. It is not very difficult to translate an instance of either $\mathrm{SB}^{\Sigma}(1)$ or $\mathrm{MCP}^{\Sigma}(r)$ to an instance of MSC as follows:

- For a given instance $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of $\mathrm{SB}^{\Sigma}(1)$, the corresponding instance of MSC is defined as:

$$U = \left\{ u_{i,j} \,\middle|\, i, j \in \{1, 2, \ldots, n\} \;\&\; i < j \right\}$$

$$\Delta_U = \left\{ S_x = \bigcup_{i,j} \left\{ u_{i,j} \right\} \,\middle|\, \exists\, \ell\colon \left(x \text{ is a substring of } s_\ell \right) \& \left(x \text{ distinguishes } s_i \text{ from } s_j \right) \right\}$$

- For a given instance $\mathcal{S} = \{ s_1, s_2, \ldots, s_n \}$ and $\mathcal{P}$ of MCP$^{\Sigma}(r)$, the corresponding instance of MSC is defined as:

$$U = \left\{ u_{i,j} \,\middle|\, i, j \in \{ 1, 2, \ldots, n \} \;\&\; i < j \right\}$$
$$\Delta_U = \left\{ S_x = \bigcup_{i,j} \left\{ u_{i,j} \right\} \,\middle|\, \left(x \in \mathcal{P} \right) \& \left(x \text{ distinguishes } s_i \text{ from } s_j \right) \right\}$$

Thus, we can use the greedy algorithm for MSC to approximate both SB$^{\Sigma}(1)$ and MCP$^{\Sigma}(r)$. For SB$^{\Sigma}(1)$, $|U| = \binom{n}{2} = O\left(n^2\right)$, $|\Delta_U| \leq \sum_{\ell=1}^{n} \binom{|s_\ell|}{2} = O\left(\sum_{\ell=1}^{n} |s_\ell|^2\right)$ and $\max_{S_x \in \Delta_U} \left\{ \left| S_x \right| \right\} \leq \frac{|U|}{2} = \frac{n(n-1)}{4}$, giving an approximation algorithm that runs in $O\left(\left(\sum_{\ell=1}^{n} |s_\ell|^2 \right) \log \left(\sum_{\ell=1}^{n} |s_\ell|^2 \right) + n^2 \right)$ time and has an approximation ratio of $1 + \ln \frac{n(n-1)}{4} \approx 2 \ln n - 0.4$. For MCP$^{\Sigma}(r)$, $|U| = \binom{n}{2} = O\left(n^2\right)$, $|\Delta_U| \leq |\mathcal{P}|$ and $\max_{S_x \in \Delta_U} \left\{ \left| S_x \right| \right\} \leq \frac{|U|}{2} = \frac{n(n-1)}{4}$, giving an approximation algorithm that runs in $O\left(|\mathcal{P}| \log |\mathcal{P}| + n^2 \right)$ time and has an approximation ratio of $1 + \ln \frac{n(n-1)}{4} \approx 2 \ln n - 0.4$.

The above-mentioned NP-hardness results and approximation algorithms were further improved by the authors in [3] and shown in Table 1.1. In the next two sections, we will discuss some of the methodologies used to prove these improved results.

$$L = \max_{\ell \in \{1,2,\ldots,n\}} \left\{ |s_\ell| \right\}, \quad \mathcal{L} = \sum_{\ell=1}^{n} |s_\ell|, \quad \varepsilon \text{ and } \delta \text{ are constants}$$

Problem name	Approximation hardness		Approximation algorithm	
	ρ-inapproximable for $\rho =$	minimal assumptions necessary	running time	ρ-approximation for $\rho =$
SB$^{\Sigma}(1)$	$(1-\varepsilon)\ln n$	NP $\not\subset$ DTIME$\left(n^{\log\log n}\right)$ $\lvert\Sigma\rvert > 1$, $0 < \varepsilon < 1$	$O\left(n^3 L^2\right)$	$1 + \ln n$
MCP$^{\Sigma}(r)$	$(1-\varepsilon)\ln n$	NP $\not\subset$ DTIME$\left(n^{\log\log n}\right)$ $\lvert\Sigma\rvert > 1$, $0 < \varepsilon < 1$	$O\left(\left(n^2 \lvert + \mathcal{L} \right) \lvert\mathcal{P}\rvert \right)$	$1 + \ln n$ $+ \ln\left(\log_2 \left(\min\{r, n\} + 1 \right) \right)$
SB$^{\Sigma}(\kappa)$	$\Omega(n^{\varepsilon})$	NP $\neq$ co-RP $\lvert\Sigma\rvert > 1$, $\kappa = n^{\delta}$ $0 < \varepsilon < \delta < 1/2$		

Table 1.1 List of of a subset of approximability results proved in [3]. DTIME$(n^{\log\log n})$ denotes the class of problems that can be solved in deterministic quasi-polynomial time and co-RP denotes the class of decision problems that admits a randomized polynomial-time algorithm $\mathcal{A}$ with the property that if the answer to the problem is YES then $\mathcal{A}$ always outputs YES but if the answer to the problem is NO then $\mathcal{A}$ outputs NO with probability at least 1/2 (see [2] for further details).

1.2.1 Average length of optimal barcodes

Via simple information-theoretic argument it follows that for $\mathrm{SB}^{\Sigma}(1)$ we must have $\mathcal{OPT}(I) \geq \log_2 |\mathcal{S}|$ for any instance I of the problem. On the other hand, it is trivial to see that $\mathcal{OPT}(I) \leq |\mathcal{S}| - 1$. Thus, it behooves to investigate the average value of $\mathcal{OPT}$ when the input strings are generated based on some probability distribution. Unfortunately, tight bounds for the average value of $\mathcal{OPT}$ is not currently known. However, the authors in [7] provide a partial answer via the following theorem. The proof of the theorem uses some asymptotic bounds by Odlyzko on average occurrences of substrings in random strings via generating function [16, Examples 6.4, 6.7, 6.8, 9.3 and 10.11],

Theorem 1 [7] *Consider a randomly generated instance of the* $SB^{\Sigma}(1)$ *of n strings over a fixed finite alphabet* Σ *in which each string in* $s_i = s_{i,0}\, s_{i,1} \ldots s_{i,\ell-1} \in \mathcal{S}$ *is of length exactly* ℓ *and is generated independently randomly with* $\Pr\left[s_{i,j} = a\right] = \dfrac{1}{|\Sigma|}$ *for any* $j \in \{1, 2, \ldots, \ell\}$ *and any* $a \in \Sigma$*. Also assume that* ℓ *is sufficiently large compared to n. Then, for a random string x over* Σ *of length* $O(\log \ell)$*, the expected number of the strings in* $\mathcal{S}$ *which contain x as a substring is* $p\,n$ *for some constant* $0 < p < 1$*.*

1.3 ENTROPY BASED INFORMATION CONTENT TECHNIQUE FOR DESIGNING APPROXIMATION ALGORITHMS FOR STRING BARCODING PROBLEMS

This technique, introduced in [3], is a *greedy* technique based on *information content* (entropy) of a partial solution; the notion of information content is directly related to the Shannon information complexity [1, 19]. In this approach we seek to select an augmenting step for a partial solution of our optimization problem that optimizes the information content of the augmented partial solution as compared to the original partial solution. A key non-trivial step for applicability of this technique is to define a suitable efficiently computable measure of the information content of a partial solution such that the monotonicity of this measure is ensured with respect to any subset of an optimal solution. For the case of $\mathrm{SB}^{\Sigma}(1)$, a high level overview of the approach is shown below:

Input: A set of strings $\mathcal{S}$ over Σ.

Output: A set of strings $\mathcal{T}$ such that $\forall\, s, s' \in \mathcal{S}\colon\ s \neq s' \ \Leftrightarrow\ \mathsf{barcode}(s, \mathcal{T}) \neq \mathsf{barcode}(s', \mathcal{T})$.

Notation for the information content (entropy) of a partial solution:

$\mathcal{H}_{\mathcal{T}}$ for an arbitrary set $\mathcal{T}$ of strings (partial solution) over Σ

Algorithm:

compute $\Gamma(\mathcal{S}) = \{\, s \mid s$ is a substring of some string in $\mathcal{S}\,\}$

$\mathcal{T} = \emptyset$

while $\mathcal{H}_{\mathcal{T}} \neq 0$ **do**

select $x \in \Gamma(\mathcal{S}) \setminus \mathcal{T}$ that *maximizes* $IC(x, \mathcal{T}) = \mathcal{H}_{\mathcal{T}} - \mathcal{H}_{\mathcal{T} \cup \{x\}}$

$\mathcal{T} = \mathcal{T} \cup \{x\}$

endwhile

Of course, a key non-trivial step is to figure out a suitable value of the entropy $\mathcal{H}_{\mathcal{T}}$ such that an execution of the above algorithm produces a valid solution with the desired approximation ratio. The authors in [3] define it in the following manner. For an arbitrary set $\mathcal{D}$ of strings over Σ:

- Define an *equivalence relation* $\overset{\mathcal{D}}{\equiv}$ on $\mathcal{S}$ as (for any two $s, s' \in \mathcal{S}$):

  $s \overset{\mathcal{D}}{\equiv} s'$ if and only if $\forall\, x \in \mathcal{D}$: x is a substring of $s \equiv x$ is a substring of s'

- If the equivalence relation $\overset{\mathcal{D}}{\equiv}$ has ℓ *equivalence classes* of size $p_1, p_2, \ldots, p_\ell > 0$, then $\mathcal{H}_{\mathcal{D}} = \log_2 \left(\Pi_{i=1}^{\ell} (\, p_i! \,) \right)$.

The above definition of entropy is somewhat similar (but not the same) to the one suggested in [15], namely $\frac{1}{|\mathcal{S}|} \log_2 \left(\Pi_{i=1}^{\ell} p_i^{p_i} \right)$, for empirical evaluation purposes.

Note that $\mathcal{H}_{\mathcal{D}} = 0$ implies the equivalence classes of $\overset{\mathcal{D}}{\equiv}$ are $|\mathcal{S}|$ singleton sets each containing one distinct string from $\mathcal{S}$, and if $\mathcal{H}_{\mathcal{D}} \neq 0$ then there exists a $x \in \mathcal{S} - \mathcal{D}$ with $IC(x, \mathcal{D}) > 0$; thus the algorithm terminates in polynomial time with a valid solution. To prove the desired approximation ratio via an amortized analysis, [3] first proves the following combinatorial properties of the function $IC(x, \mathcal{D}) > 0$:

- $\forall\, x$: $\mathcal{D} \subset \mathcal{D}' \;\Rightarrow\; IC(x, \mathcal{D}) \geq IC(x, \mathcal{D}')$, and
- $\forall\, x$: $IC(x, \emptyset) = h < |\mathcal{S}|$.

Thus, if the algorithm selected strings $x_1, x_2, \ldots, x_q$ in this order in $\mathcal{T}$, then $\sum_{i=1}^{q} IC(x_i, \{x_1, x_2, \ldots, x_{i-1}\}) = \mathcal{H}_{\emptyset} = h < |\mathcal{S}|$. The proof shows how to carefully distribute the cost 1 of adding each extra set in $\mathcal{T}$ to the strings in an optimal solution of $\mathrm{SB}^{\Sigma}(1)$ using the $IC(x_i, \{x_1, x_2, \ldots, x_{i-1}\})$ quantities such that each element in this optimal solution receives a total cost of at most $1 + \int_1^{\sum_{i=1}^{q} IC(x_i, \{x_1, x_2, \ldots, x_{i-1}\})} \mathrm{d}x/x < 1 + \ln h < 1 + \ln |\mathcal{S}|$

A very similar proof with appropriate modifications work for $\mathrm{MCP}^{\Sigma}(r)$ as well. In this case, $IC(x, \emptyset) = h < |\mathcal{S}| \log_2 \left(\min \{ r+1, |\mathcal{S}| \} \right)$ and thus each element in this optimal solution receives a total cost of at most $1 + \ln h < 1 + \ln |\mathcal{S}| + \ln \log_2 \left(\min \{ r+1, |\mathcal{S}| \} \right)$.

1.4 TECHNIQUES FOR PROVING INAPPROXIMABILITY RESULTS FOR STRING BARCODING PROBLEMS

In this section, we review a few techniques from structural complexity theory that were used to prove inapproximability results for various string barcoding problems.

1.4.1 Reductions from set covering problem

An usual starting point for this technique is the following well-known inapproximability result for MSC.

Theorem 2 [9] *Assuming* NP $\not\subset DTIME\left(n^{\log\log n}\right)$*, instances of the MSC problem whose solution requires at least* $(\log_2 n)^2$ *sets cannot be approximated to within an approximation ratio of* $(1-\varepsilon)\ln n$ *for any constant* $\varepsilon > 0$ *in polynomial time.*

It seems difficult to transform the above inapproximability result for MSC to an inapproximability bound for $\text{SB}^{\{0,1\}}(1)$ of a similar quality because of the special restrictive nature of $\text{SB}^{\{0,1\}}$, and thus the techniques used by the authors in [8, 12] does not seem to apply. To overcome this issue, the authors in [3] introduced an intermediate problem, called *test set with order with parameter m problem* and denoted by TSO^m, which could be transformed to $\text{SB}^{\{0,1\}}$. TSO^m is a non-trivial generalization of the well-known NP-hard *minimum test collection problem* in diagnostic testing [11, pp. 71] and is defined as follows:

Problem name: Test set with order with parameter m (TSO^m).

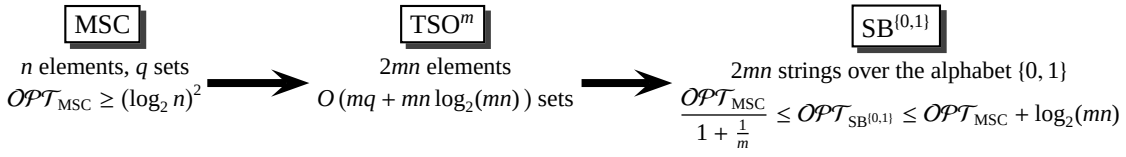
Input: A universe $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ of n elements, a collection $\mathcal{S}$ of subsets of $\mathcal{U}$ (tests) that includes the $2n-1$ special sets $S_1 = \{u_1\}, S_2 = \{u_1, u_2\}, S_3 = \{u_1, u_2, u_3\}, \ldots, S_n = \{u_1, u_2, u_3, \ldots, u_n\}$, $S_{n+1} = \{u_2\}, S_{n+2} = \{u_3\}, \ldots, S_{2n-1} = \{u_n\}$, and a positive integer m.

Valid solutions: A collection $\mathcal{T} \subseteq \mathcal{S}$ of subsets from $\mathcal{S}$ such that

$$\forall\, u_i, u_j \in \mathcal{U}\colon i \neq j \;\Rightarrow\; \exists T \in \mathcal{T} \text{ such that } \left|\{u_i, u_j\} \cap T\right| = 1$$

Objective: *Minimize* $\text{cost}(\mathcal{T}) = \left|\mathcal{T} \setminus \{S_1, S_2, \ldots, S_{2n-1}\}\right| + \frac{1}{m}\left|\mathcal{T} \cap \{S_1, S_2, \ldots, S_{2n-1}\}\right|$.

The proof is completed by having a transformation between these problems such that the corresponding optimal solutions are closely related as shown schematically below where $\mathcal{OPT}_{\text{MSC}}$ and $\mathcal{OPT}_{\text{SB}^{\{0,1\}}}$ denote the objective values of an optimal solution of the generated instances of MSC and $\text{SB}^{\{0,1\}}$, respectively. This provides an $(1-\varepsilon)$-inapproximability result for $\text{SB}^{\{0,1\}}$.



A formal description of the transformations of the input instances among the problems are complicated, so here we just illustrate the transformation with the following simple example for $m = 1$:

Input instance of MSC:

$$U = \{u_1, u_2, u_3, u_4\}, \Delta_U = \{S_1, S_2, S_3\}, S_1 = \{u_1, u_2, u_4\}, S_2 = \{u_2, u_3\}, S_1 = \{u_2, u_3, u_4\}$$

Tranformed input instance of TSO^1 from input instance of MSC:

$\mathcal{U} = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$

$\mathcal{S}$ contains the following sets:

$\boldsymbol{B_1} = \{u_1, u_3, u_7\}$, $\boldsymbol{B_2} = \{u_3, u_5\}$, $\boldsymbol{B_3} = \{u_3, u_5, u_7\}$ (corresponding to S_1, S_2, S_3 in MSC)

$\boldsymbol{B_4} = \{u_3, u_4, u_7, u_8\}$, $\boldsymbol{B_5} = \{u_5, u_6, u_7, u_8\}$ (additional sets)

$S_1 = \{u_1\}$, $S_2 = \{u_1, u_2\}$, $S_3 = \{u_1, u_2, u_3\}$, $S_4 = \{u_1, u_2, u_3, u_4\}$

$S_5 = \{u_1, u_2, u_3, u_4, u_5\}$, $S_6 = \{u_1, u_2, u_3, u_4, u_5, u_6\}$

$S_7 = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$, $S_8 = \boldsymbol{B_6} = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$

$S_9 = \{u_2\}$, $S_{10} = \{u_3\}$, $S_{11} = \{u_4\}$, $S_{12} = \{u_5\}$, $S_{13} = \{u_6\}$, $S_{14} = \{u_7\}$, $S_{15} = \{u_8\}$ (special sets)

Tranformed input instance of SB$^{\{0,1\}}$ from input instance of TSO^1: the 8 strings over $\Sigma = \{0, 1\}$ are as follows where 0^i and 1^i indicates a string of i zeroes or ones, respectively (*e.g.*, $0^3 = 000$):

$S_{\mathbf{1}} = \mathbf{0}\ 1^{\mathbf{1}}\ \mathbf{0}\ 1^{\mathbf{6}}\ \mathbf{0} = \mathbf{0\ 1\ 0\ 111111\ 0}$ (since $u_1 \in B_1, B_6$)

$S_{\mathbf{2}} = \mathbf{0}^{\mathbf{2}}\ 1^{\mathbf{6}}\ \mathbf{0}^{\mathbf{2}} = \mathbf{00\ 111111\ 00}$ (since $u_2 \in B_6$)

$S_{\mathbf{3}} = \mathbf{0}^{\mathbf{3}}\ 1^{\mathbf{1}}\ \mathbf{0}^{\mathbf{3}}\ 1^{\mathbf{2}}\ \mathbf{0}^{\mathbf{3}}\ 1^{\mathbf{3}}\ \mathbf{0}^{\mathbf{3}}\ 1^{\mathbf{4}}\ \mathbf{0}^{\mathbf{3}}\ 1^{\mathbf{6}}\ \mathbf{0}^{\mathbf{3}}$
$= \mathbf{000\ 1\ 000\ 11\ 000\ 111\ 000\ 1111\ 000\ 111111\ 000}$ (since $u_3 \in B_1, B_2, B_3, B_4, B_6$)

$S_{\mathbf{4}} = \mathbf{0}^{\mathbf{4}}\ 1^{\mathbf{4}}\ \mathbf{0}^{\mathbf{4}}\ 1^{\mathbf{6}}\ \mathbf{0}^{\mathbf{4}} = \mathbf{0000\ 1111\ 0000\ 111111\ 0000}$ (since $u_4 \in B_4, B_6$)

$S_{\mathbf{5}} = \mathbf{0}^{\mathbf{5}}\ 1^{\mathbf{2}}\ \mathbf{0}^{\mathbf{5}}\ 1^{\mathbf{3}}\ \mathbf{0}^{\mathbf{5}}\ 1^{\mathbf{5}}\ \mathbf{0}^{\mathbf{5}}\ 1^{\mathbf{6}}\ \mathbf{0}^{\mathbf{5}}$
$= \mathbf{00000\ 11\ 00000\ 111\ 00000\ 11111\ 00000\ 111111\ 000000}$ (since $u_5 \in B_2, B_3, B_5, B_6$)

$S_{\mathbf{6}} = \mathbf{0}^{\mathbf{6}}\ 1^{\mathbf{5}}\ \mathbf{0}^{\mathbf{6}}\ 1^{\mathbf{6}}\ \mathbf{0}^{\mathbf{6}} = \mathbf{000000\ 11111\ 000000\ 111111\ 000000}$ (since $u_6 \in B_5, B_6$)

$S_{\mathbf{7}} = \mathbf{0}^{\mathbf{7}}\ 1^{\mathbf{1}}\ \mathbf{0}^{\mathbf{7}}\ 1^{\mathbf{3}}\ \mathbf{0}^{\mathbf{7}}\ 1^{\mathbf{4}}\ \mathbf{0}^{\mathbf{7}}\ 1^{\mathbf{5}}\ \mathbf{0}^{\mathbf{7}}\ 1^{\mathbf{6}}\ \mathbf{0}^{\mathbf{7}}$
$= \mathbf{00000000\ 1\ 00000000\ 111\ 00000000\ 1111\ 00000000\ 11111\ 00000000\ 1111111\ 00000000}$
(since $u_7 \in B_1, B_3, B_4, B_5, B_6$)

$S_{\mathbf{8}} = \mathbf{0}^{\mathbf{8}}\ 1^{\mathbf{4}}\ \mathbf{0}^{\mathbf{8}}\ 1^{\mathbf{5}}\ \mathbf{0}^{\mathbf{8}}\ 1^{\mathbf{6}}\ \mathbf{0}^{\mathbf{8}}$
$= \mathbf{00000000\ 1111\ 00000000\ 11111\ 00000000\ 111111\ 00000000}$ (since $u_8 \in B_4, B_5, B_6$)

For further details see [3].

1.4.2 Reduction from graph coloring problem

The $\Omega(n^{\varepsilon})$-inapproximability result for $\mathrm{SB}^{\{0,1\}}\left(n^{\delta}\right)$ under the assumption of $0 < \varepsilon < \delta < 1/2$ and NP $\neq$ co-RP is proved in [3] by providing an approximation-preserving reduction from a strong inapproximability result for the graph coloring problem. The graph coloring problem is a well-known combinatorial optimization problem defined as follows [11]:

Problem name: Graph coloring.

Input: An undirected unweighted graph $G = (V, E)$.

Valid solutions: An assignment of colors to nodes such that no two adjacent nodes have the same color.

Objective: *Minimize* the number of colors used.

Let $\mathcal{OPT}_{\mathrm{color}}(G)$ denote the minimum number of colors used in a valid coloring of G. A set of nodes in G are said to be *independent* if no two of them are connected by an edge. Let $\mathcal{OPT}_{\mathrm{ind}}(G)$ denote the *maximum* number of independent nodes in G. The following strong inapproximability result for computing $\mathcal{OPT}_{\mathrm{color}}(G)$ can be found in [10].

Theorem 3 [10] *Assuming* NP *$\neq$co-RP, there is no polynomial-time algorithm that computes $\mathcal{OPT}_{\mathrm{color}}(G)$ with an approximation ratio of $|V|^{\rho}$ even if $\mathcal{OPT}_{\mathrm{ind}}(G) \leq |V|^{\delta}$ for any two constants $0 < \rho < \delta < 1$,*

As in the case of reduction from the set covering problem in the previous section, the authors in [3] used an intermediate problem, namely the grouped test set (TS^{κ}) problem, that helps in the reduction from graph coloring to $\mathrm{SB}^{\{0,1\}}$. The TS^{κ} problem can be thought as a generalization of the TSO^{m} problem defined in the previous section without the order property and the parameter m; formally the problem is as follows.

Problem name: Grouped test set (TS^{κ}).

Input: A universe $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ of n elements, a collection $\mathcal{S}$ of subsets of $\mathcal{U}$ (tests).

Definition of a κ-test: A *κ-test* is a union of at most k sets from $\mathcal{S}$.

Valid solutions: A collection $\mathcal{T} \subseteq \mathcal{S}$ of κ-tests such that

$$\forall\, u_i, u_j \in \mathcal{U}\colon\ i \neq j \ \Rightarrow\ \exists T \in \mathcal{T} \text{ such that } \left|\{u_i, u_j\} \cap T\right| = 1$$

Objective: *Minimize* $\mathrm{cost}(\mathcal{T}) = |\mathcal{T}|$.

As before, one can defined a version of TS^{κ} "with order" in the following manner:

Problem name: Grouped test set with order (TS^{κ} with order).

Input: A universe $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ of n elements, a collection $\mathcal{S}$ of subsets of $\mathcal{U}$ (tests) such that

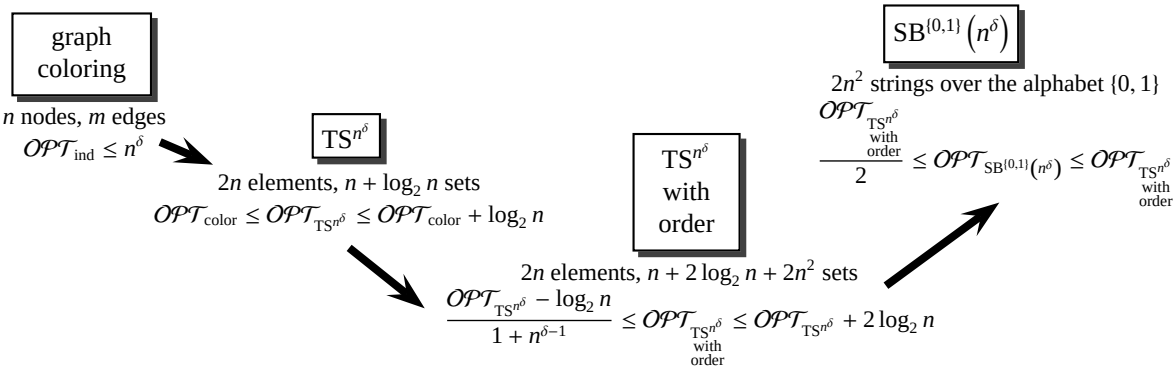
$$\left\{\{u_1\}, \{u_1, u_2\}, \{u_1, u_2, u_3\}, \ldots, \{u_1, u_2, u_3, \ldots, u_n\}, \{u_2\}, \{u_3\}, \ldots, \{u_n\}\right\} \subseteq \mathcal{S}$$

Valid solutions: A collection $\mathcal{T} \subseteq \mathcal{S}$ of κ-tests such that

$$\forall\, u_i, u_j \in \mathcal{U}\colon i \neq j \;\Rightarrow\; \exists T \in \mathcal{T} \text{ such that } \big|\{u_i, u_j\} \cap T\big| = 1$$

Objective: *Minimize* $\text{cost}(\mathcal{T}) = |\mathcal{T}|$.

The proof is completed by having a transformation between these problems such that the corresponding optimal solutions are closely related as shown schematically below where $\mathcal{OPT}_{\mathcal{P}}$, for a problem $\mathcal{P}$, denotes the objective values of an optimal solution of the generated instances of the problem $\mathcal{P}$. This provides an $\Omega(n^{\varepsilon})$-inapproximability result for $\text{SB}^{\{0,1\}}\left(n^{\delta}\right)$; for further details, see [3].



1.5 HEURISTIC ALGORITHMS FOR STRING BARCODING PROBLEMS

In addition to designing efficient algorithms with provable bounds on approximation ratio, one can also consider designing heuristic algorithms for barcoding problems that may not admit a proof of their approximation bounds but nonetheless work well in practice. For the basic binary-valued string barcoding problem $\text{SB}^{\Sigma}(1)$, we outline a few possible heuristic approaches below.

Entropy based method with different measure for information content

The greedy approach described in Section 1.3 used a very specific definition of the measure of information content (entropy), namely $\mathcal{H}_{\mathcal{D}} = \log_2\left(\Pi_{i=1}^{\ell}(\,p_i!\,)\right)$. In principle, the approach can be used with other entropy measures that decrease monotonically as partial solutions progress towards a complete solution. An appealing candidate is $\mathcal{H}_{\mathcal{D}} = \frac{1}{|\mathcal{S}|}\log_2\left(\Pi_{i=1}^{\ell} p_i^{p_i}\right)$ suggested in [15] as this version of the measure follows the standard entropy definition more closely.

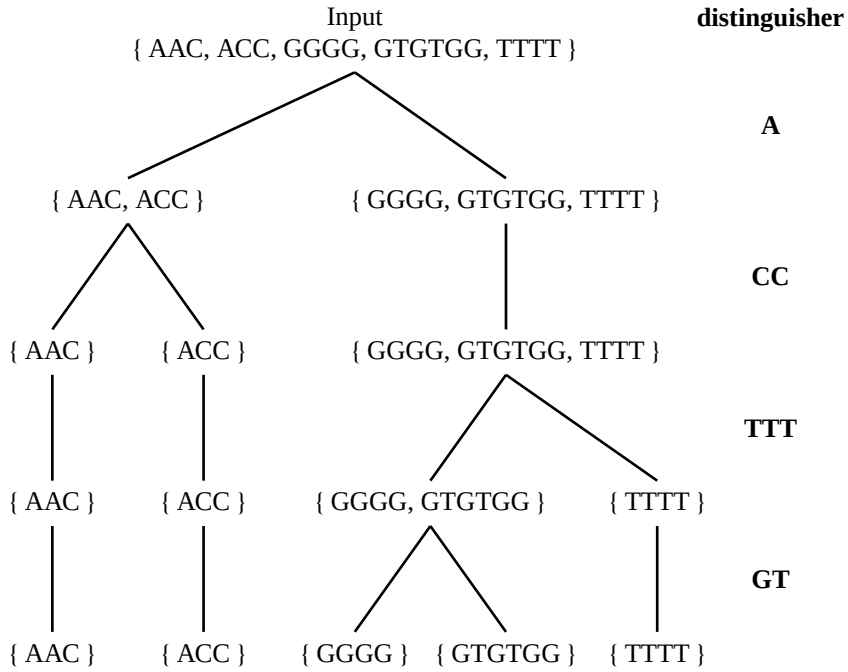


Figure 1.4 Greedy selection of strings in our solution generates a tree partition of the set of input sequences such that each leaf node has exactly one sequence.

Balanced partitioning approach

The $\overset{\mathcal{T}}{\equiv}$ equivalence relation used in Section 1.3 suggests an alternate way of looking at greedy selection of strings to form a barcoding. At every step, the selected string (distinguisher) in the solution affects each equivalence set either by keeping it same or by partitioning the set into two parts. Equivalently, one can view the successive selection of strings in the solution as generating a tree partitioning the given set of input sequences $\mathcal{S}$ (see Fig. 1.4). Note that the height of the tree is precisely the number of strings in our solution. Thus, one possible greedy strategy is to select a distinguisher greedily at each step that increases the height of the current partition tree by the *least* amount.

1.6 CONCLUSION

In this chapter, we have described a few versions of the string barcoding problems and have reviewed some algorithmic and inapproximability reduction tools to analyze algorithmic complexity questions about these problems. There are other aspects of these problems, such as robustness of barcodes against noises, that arise in practical applications of barcoding in pathogen detections, that we have not discussed here;

the reader can find further information about them in the cited references. A software incorporating some of the algorithmic methodologies discussed in this review was reported in [6] and can be found at the website `http://dna.engr.uconn.edu/?page_id=23`.

From an algorithmic and computational complexity perspective, the following research questions may be worthy of further investigation:

- If the set of strings are generated via a biologically relevant (non-uniform) distribution over the alphabet, what is the expected length of an optimal barcode and what is the computational complexity of finding such a barcode ?
- Is there an efficient approximation algorithm for $\mathrm{SB}^{\Sigma}(\kappa)$ when $\kappa > 1$ grows slowly with n (*e.g.*, $\kappa = O(\log n)$) ?

Acknowledgments

The authors were supported in part by NSF grants IIS-1160995 and DBI-1062328. DasGupta would also like to thank all their collaborators in their joint barcoding research articles.

References

1. Y. S. Abu-Mostafa (editor). *Complexity in Information Theory*, Springer Verlag, 1986.
2. J. L. Balcázar, J. Díaz and J. Gabarro. *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science, Springer, Berlin, New York, 1988.
3. P. Berman, B. DasGupta and M.-Y. Kao. *Tight Approximability Results for Test Set Problems in Bioinformatics*, Journal of Computer & System Sciences, 71 (2), 145-162, 2005.
4. J. Borneman, M. Chrobak, G. D. Vedova, A. Figueora and T. Jiang. *Probe Selection Algorithms with Applications in the Analysis of Microbial Communities*, Bioinformatics, 1, 1-9, 2001.
5. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*, The MIT Press, 2001.
6. B. DasGupta, K. Konwar, I. Mandoiu and A. Shvartsman. *DNA-BAR: Distinguisher Selection for DNA Barcoding*, Bioinformatics, 21 (16), 3424-2426, 2005.
7. B. DasGupta, K. Konwar, I. Mandoiu and A. Shvartsman. *Highly Scalable Algorithms for Robust String Barcoding*, International Journal of Bioinformatics Research & Applications, 1 (2), 145-161, 2005.
8. K. M. J. De Bontridder, B. V. Halldórsson, M. M. Halldórsson, C. A. J. Hurkens, J. K. Lenstra, R. Ravi and L. Stougie. *Approximation algorithms for the test cover problem*, Mathematical Programming, series B, 98 (1-3), 477-491, 2003.
9. U. Feige. *A threshold for approximating set cover*, Journal of the ACM, 45, 634-652, 1998.

10. U. Feige and J. Kilian. *Zero knowledge and the chromatic number*, Journal of Computer and System Sciences, 57 (2), 187-199, 1998.

11. M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1979.

12. B. V. Halldórsson, M. M. Halldórsson and R. Ravi. *On the Approximability of the Minimum Test Collection Problem*, Proceedings of the Ninth Annual European Symposium on Algorithms, Lecture Notes in Computer Science 2161, 158-169, Springer Verlag, 2001.

13. D. S. Johnson. *Approximation Algorithms for Combinatorial Problems*, Journal of Computer and Systems Sciences, 9, 256-278, 1974.

14. T. G. Ksiazek, D. Erdman, C. S. Goldsmith, and S. R. Zaki, T. Peret, S. Emery, S. Tong, C. Urbani, J. A. Comer, W. Lim, P. E. Rollin, S. F. Dowell, A.-E. Ling, C. D. Humphrey, W.-J. Shieh, J. Guarner, C. D. Paddock, P. Rota, B. Fields, J. DeRisi, J.-Y. Yang, N. Cox, J. M. Hughes, J. W. LeDuc, W. J. Bellini and L. J. Anderson. *A novel coronavirus associated with severe acute respiratory syndrome*, *New England Journal of Medicine*, 348 (20), 1953-1966, 2003.

15. B. M. E. Moret and H. D. Shapiro. *On minimizing a set of tests*, SIAM Journal on Scientific and Statistical Computing, 6, 983-1003, 1985.

16. A. M. Odlyzko. *Asymptotic enumeration methods*, in Handbook of Combinatorics, Vol. II, R. L. Graham, M. Grötschel and L. Lovász (editors), The MIT Press, 1063-1230, 1995.

17. S. Rash and D. Gusfield. *String Barcoding: Uncovering Optimal Virus Signatures*, Sixth Annual International Conference on Computational Biology, 54-261, 2002.

18. A. Schliep, D. C. Torney and S. Rahmann. *Group Testing with DNA Chips: Generating Designs and Decoding Experiments*, Proceedings of the IEEE Computational Systems Conference, 2, 84-91, 2003.

19. C. E. Shannon. *Mathematical Theory of Communication*, Bell Systems Technical Journal, 27, 379-423, 623-658, 1948.

20. V. Vazirani. *Approximation Algorithms*, Springer-Verlag, 2001.

21. D. Wang, L. Coscoy, M. Zylberberg, P. C. Avila, H. A. Boushey, D. Ganem and J. L. DeRisi. *Microarray-based detection and genotyping of viral pathogens*, Proceedings of the National Academy of Sciences, 99 (24), 15687-15692, 2002.