

Probe Selection Algorithms with Applications in the Analysis of Microbial Communities

J. Borneman* M. Chrobak† G. Della Vedova‡ A. Figueroa§ T. Jiang¶

Abstract

We propose two efficient heuristics for minimizing the number of oligonucleotide probes needed for analyzing populations of ribosomal RNA gene (rDNA) clones by hybridization experiments on DNA microarrays. Such analyses have applications in the study of microbial communities. Unlike in a classical SBH (sequencing by hybridization) procedure, where multiple probes are put on a DNA chip, in our applications we perform a series of experiments, each one consisting of applying a single probe to a DNA microarray containing a large sample of rDNA sequences from the studied population. The overall cost of the analysis is thus roughly proportional to the number of experiments, underscoring the need for minimizing the number of probes. Our algorithms are based on two well-known optimization techniques, *i.e.* simulated annealing and Lagrangian relaxation, and our preliminary tests demonstrate that both algorithms are able to find satisfactory probe sets for real rDNA data.

Keywords: DNA microarray, oligonucleotide probe, hybridization fingerprint, combinatorial optimization, simulated annealing, Lagrangian relaxation, ribosomal RNA gene, microbial community

1 Introduction

Microorganisms are of fundamental importance for agriculture, biotechnology and medicine. However, to fully manage and utilize this resource, a thorough understanding of these organisms and their communities is needed. Current estimates suggest that thousands of different microorganisms inhabit most environments, the vast majority of which have not yet been described because they do not grow on artificial media [1, 29]. Recent studies of microbial communities have been assisted by the development of *ribosomal RNA* (rRNA) gene analyses, which have eliminated the need to culture these organisms and led to the identification of thousands of previously undescribed microorganisms [2, 12, 24]. rRNA genes (rDNAs) are useful taxonomic indicators because they are found in all known organisms, contain both highly conserved and variable regions, and have a slow but relatively constant molecular clock or mutation rate [30].

Analysis of microbial communities using rRNA genes can be done using several simple approaches. The most commonly used methods include *Denaturing Gradient Gel Electrophoresis* (DGGE) [23] and *Terminal Restriction Fragment Length Polymorphisms* (T-RFLP) [20], both of which allow analysis of many samples in a relatively short time period. Unfortunately, they also produce limited data sets as communities that may contain thousands of different species [28] are resolved into approximately 10 to 30 groups. To obtain comprehensive depictions of community structure, investigators can use extensive sequence analysis of rDNA clone libraries. In two such studies, hundreds of bacterial rDNA clones

*Department of Plant Pathology, University of California, Riverside, CA 92521. borneman@ucr.ac1.ucr.edu

†Department of Computer Science, University of California, Riverside, CA 92521. marek@cs.ucr.edu

‡Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, Milano, I-20126, Italy. dellavedova@disco.unimib.it

§Department of Computer Science, University of California, Riverside, CA 92521. andres@cs.ucr.edu

¶Department of Computer Science, University of California, Riverside, CA 92521. jiang@cs.ucr.edu

from several soils were analyzed, no duplicates were found and none had been previously described [3, 4]. Due to this remarkable diversity and to the high cost of DNA sequencing, this approach to comprehensive analysis of microbial communities is not feasible with current technology.

The goal of our research is to develop a high-throughput approach for the examination of microbial communities. To accomplish this, we are adapting an existing strategy termed *oligonucleotide fingerprinting* that permits the identification of thousands of cDNA clones [7, 8, 9, 21, 22]. After the rDNA clone libraries are constructed, the clones are classified by individual hybridization experiments on DNA microarrays with a series of short DNA oligonucleotides into *clone types* or *operational taxonomic units* (OTUs). Once classified, the nucleotide sequence of representative clones from each OTU can then be obtained by DNA sequencing to provide phylogenetic descriptions of the microorganisms. One of the key features of this strategy is that after a comprehensive database, that correlates hybridization patterns with nucleotide sequence data, has been compiled, little additional rDNA clone sequencing will be required, resulting in a significant reduction of cost and effort. The effectiveness of this general strategy has been demonstrated in the biotechnology arena, where it is currently being used to screen and identify millions of cDNA clones [7].

One of the biggest challenges in developing this strategy is the selection of the oligonucleotide probe sets. The optimal probe set will contain as few oligonucleotides as possible. The number of probes used is exactly the number of hybridization experiments, so this will minimize experimental cost and effort. In designing these probes, several key issues need to be addressed, not the least of which is that the nucleotide sequences of most rRNA genes remain unknown. Another complicating factor is that the microbial community in each environment may be different and therefore require a unique probe set. Development of strategies to optimize oligonucleotide probes sets is therefore crucial for this research as well as for other applications of oligonucleotide fingerprinting, such as clone selection for genomic sequencing [27] and gene expression analysis [14, 25, 26].

Our work on probe selection. Recall that an oligonucleotide fingerprinting procedure for a given set of DNA clones consists of performing a series of hybridization experiments on DNA microarrays, each involving the clones and a unique short oligonucleotide probe. As a result of such a procedure, for each clone we obtain a binary vector called the *fingerprint*, which describes which probes occur in this clone. By carefully choosing the probe set, we can reduce the number of hybridization experiments, thus reducing the cost and effort.

Although work based on hybridization experiments typically uses binary membership information of probes in clones, the experiments actually produce more information. The hybridization experiments that we have performed could actually provide linear fluorescence response over a range of 0–4 occurrences of a probe sequence per clone. The results obtained so far have not been consistent enough to provide statistically reliable information. Nevertheless, we believe that with further fine-tuning we will be able to adopt some non-binary model in our strategy. Therefore, in this paper, we will consider two models: binary membership, and frequency of occurrences up to 4.

The basic (binary) probe selection problem can be described as follows. We are given a population \mathcal{C} of m *unknown* rDNA clones. To analyze \mathcal{C} , we need to choose a set S of oligonucleotide probes of a given length l . In our microbial community application, the clones typically have length of approximately 1500 and l is between 5 and 10. We say that a probe p *distinguishes* a pair of clones c and d if p is a substring of *exactly* one of c or d . Our goal is to find a *smallest* set S of length- l probes such that any two distinct clones c and d from \mathcal{C} are distinguished by at least one probe in S .

The paradox that arises here is, of course, that we do not actually know the rDNA sequences in the population. So, how can we compute the minimal probe set? Furthermore, even if we did have complete sequences of these clones, computing optimal probe sets for large data sets is computationally infeasible. To overcome these difficulties, we propose the following two-step approach:

1. Choose a random subset \mathcal{C}' of t rDNA clones from the given population, where t is a parameter chosen by empirical study. Sequence the clones in \mathcal{C}' .
2. Compute an optimal, or near-optimal, probe set S for \mathcal{C}' . Use S for analyzing the whole clone population.

The intuition behind this approach is quite simple: if the random subset \mathcal{C}' is large enough, the computed probe set S will be, with high probability, close to being optimal for the whole population. Moreover, we may also augment the subset \mathcal{C}' with known rDNA sequences available in databases such as Genbank and the Ribosomal Database. (There are already more than a thousand such sequences.)

As noted earlier, we will also consider a more general definition of distinguishability taking into account the discrepancy between the numbers of occurrences of the probe p in the clones (up to 4 occurrences). We remark here that, in practice, one may only want to distinguish those clones whose similarity is below a certain threshold. Moreover, due to the existence of hybridization noise, it would make sense to consider a more robust version of the problem where we require that every pair of rDNA sequences be distinguished by at least r probes in S , for some small redundancy parameter r .

In this paper, we focus on Step 2 of this approach, that is, on computing an optimal probe set for a given set of rDNA sequences. In Section 2 we introduce two alternative formulations of probe selection, one called Minimum Cost Probe Set (MCPS), and the other called Minimum Indistinguished Probe Set (MIPS). In MCPS, we ask for a minimum number of probes that distinguish all given clones. In MIPS, we ask for a set of k probes, where k is given, that minimizes the number of indistinguished pairs of clones. Both problems are variants of the well-known combinatorial optimization problem SET COVER [11]. Since both MCPS and MDPS are NP-hard, we propose two efficient heuristic algorithms based on well-known optimization techniques, namely simulated annealing, for MDPS, and Lagrangian relaxation, for MCPS. A brief overview of the two techniques is given in Section 2. The actual algorithms based on these techniques and their efficient implementations are presented in Section 3. In Section 4, we report some preliminary test results on these algorithms. These tests demonstrate that both algorithms are able to find satisfactory probe sets for real rDNA data. In particular, the simulated annealing algorithm in general outperforms the greedy heuristic proposed in [15] in terms of the objective functions that we are interested in, and the Lagrangian relaxation algorithm often finds solutions close to being optimal.

Previous work on probe selection. Although the importance of appropriate selection of probes has been discussed in the literature, little systematic study has been done on this problem. One simple approach would be to use random oligonucleotides. However, DNA sequences that appear in nature are not really random, and thus a random probe is not likely to occur in a sufficient number of clones in the population to provide adequate discrimination. Some methods involve choosing probes based on their frequencies in the clones (see, for example, [9]). These methods do not work well for our purpose, since rDNA sequences have highly conserved regions and thus probes selected by these methods tend to occur in too many clones to be useful. Other criteria that have been considered for selecting probes include G + C content (see [6, 10]) and free energy and melting temperature (see [19]). The only research that we are aware of where the probe selection has been formulated as an explicit optimization problem is the recent work by Herwig *et al* [15]. They have presented a simple greedy heuristic based on clustering and entropy and shown empirical results that the algorithm produces probe sets of much higher quality than those chosen by random or according to frequency.

2 Formulations of Probe Selection and Optimization Techniques

We represent clones and probes as sequences over the alphabet $\{A,C,G,T\}$. The set of clones will be denoted by $\mathcal{C} = \{c_1, \dots, c_m\}$ and the set of preselected length- l probes by $\mathcal{P} = \{p_1, \dots, p_n\}$; their

cardinality will be denoted respectively by m and n . Let \mathcal{C}^2 be the set of all *pairs* of different clones from \mathcal{C} , that is, $\mathcal{C}^2 = \{(c, d) : c, d \in \mathcal{C}, c < d\}$, where “ $<$ ” is an arbitrary (say, lexicographic) ordering of \mathcal{C} .

Since the fluorescence response in a hybridization experiment is linear with respect to the number of occurrences of the probe in a clone up to a certain threshold R , different values of R give rise to different versions of the *distinguishability criteria*. In this paper, we will consider two cases: $R = 1$ (called binary) and $R = 4$ (simply called non-binary). By $occ(c, p)$ we denote the number of occurrences of p in c . (To simplify exposition, probe sequences studied here are actually the complementary sequences of real probes used in hybridization experiments.) Given a set S of probes, the S -*fingerprint* of c , denoted by $fingerprint_S(c)$, is the vector of values $\min\{R, occ(p, c)\}$ over all $p \in S$. We will say that a set S of probes *distinguishes* two clones c and d if $fingerprint_S(c) \neq fingerprint_S(d)$. For instance, let $c = AAACCTGA$ and $d = AAACATAAAA$. If $R = 1$, CCT distinguishes c and d , while ACT and AAA do not. On the other hand AAA distinguishes the two clones when $R = 4$. By $\Delta_S \subseteq \mathcal{C}^2$ we denote the set of pairs of clones that are distinguished by S . When S is a singleton, say $S = \{p\}$, we will simplify notation by writing Δ_p instead of $\Delta_{\{p\}}$.

The problem of finding a good set of probes for distinguishing a given set of clones have two sound formulations:

MINIMUM COST PROBE SET (MCPS)

Instance: a set \mathcal{C} of clones and a set \mathcal{P} of probes such that $\Delta_{\mathcal{P}} = \mathcal{C}^2$.

Feasible solutions: a subset $S \subseteq \mathcal{P}$ such that $\Delta_S = \mathcal{C}^2$.

Measure: $|S|$, to be minimized.

MINIMUM INDISTINGUISHED PROBE SET (MIPS)

Instance: a set \mathcal{C} of clones, a set \mathcal{P} of probes, and an integer k .

Feasible solutions: a subset $S \subseteq \mathcal{P}$, with $|S| = k$.

Measure: $\binom{|\mathcal{C}|}{2} - |\Delta_S|$, to be minimized.

Both formulations are useful in the sense that they put stringency condition on different parameters. For example, MCPS is more appropriate when we want to achieve a predetermined level of resolution, while MIPS makes more sense when the number of hybridization experiments (thus the number of probes) has already been decided based on, say, the budget. Note that MCPS is a special case of the well-known SET COVER problem [11], where the universe to be covered is \mathcal{C}^2 and the covering sets are the various Δ_p , where p is a probe. Analogously, the complementary problem of MIPS is a special case of MAXIMUM COVERAGE [16], which can be viewed as a *dual* problem of SET COVER. Observe that if we decide to maximize the *entropy* of the clusters induced by Δ_S instead of minimizing the objective function in MDPS,¹ then we obtain the problem considered in [15]. Other objective functions such as G + C content, free energy, and melting temperature can also be easily incorporated into these formulations. The following result shows that both problems are hard to compute exactly.

Theorem 1 *The problems MCPS and MIPS are NP-hard, when \mathcal{P} (and thus l , the length of probes) is unbounded.*

There are simple greedy algorithms that approximate SET COVER and MAXIMUM COVERAGE with ratios $\log m$ and $1 - \frac{1}{e} \approx 0.632$, respectively [16]. However, these approximation guarantees are far from being satisfactory for practical purposes. We hence look for efficient heuristics that may provide better solutions in practice.

The heuristic method for SET COVER recently presented in [5] seems to have the best performance among all practical methods. The method is based on linear programming (LP) and *Lagrangian*

¹The entropy of a set of clusters is defined as $-\sum_i f_i \log f_i$, where f_i denotes the fraction of clones that are in cluster i [15].

relaxation, and it can be easily adapted for MCPS. A detailed explanation of the Lagrangian relaxation technique will be given in the next section, but here we will sketch an outline of the technique as applied to MCPS. MCPS can be easily written as a constrained linear integer minimization problem. That is, it can be written in the form: minimize $f(\mathbf{x})$, over all $\mathbf{x} \in \{0, 1\}^n$ such that $g_i(\mathbf{x}) \leq 0$, for $i \in I$, where each g_i is a linear function. We associate with MCPS the so-called Lagrangian function $L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i \in I} \lambda_i g_i(\mathbf{x})$, where each vector $\boldsymbol{\lambda} = \langle \lambda_i | i \in I \rangle$ is called a *Lagrangian multiplier*. For any given $\boldsymbol{\lambda} \geq 0$, the value of $\min_{\mathbf{x}} \{L(\mathbf{x}, \boldsymbol{\lambda})\}$ is a lower bound on the optimum of the objective function f . The computation of an approximate solution to the original problem is split into three steps: computing a good Lagrangian multiplier $\boldsymbol{\lambda}$, computing \mathbf{x} that minimizes $L(\mathbf{x}, \boldsymbol{\lambda})$, and then expanding \mathbf{x} to a feasible solution using a greedy algorithm. Since MCPS is a special case of SET COVER, we believe that this technique would work well for MCPS too.

The above method does not apply to MIPS because the intuitive formulation of this problem as a constrained linear integer optimization problem cannot be usefully exploited. So we turned to simulated annealing [18], which is a widely used framework for solving optimization problems and is an improvement over the *local search* method. A local search algorithm requires that we are able to define a neighborhood relation between feasible solutions. It starts from an initial random solution and iteratively moves to a neighbor solution whose value is better than the one of the present solution, unless no move is possible. Such a simple algorithm is guaranteed to return a local optimal solution, but clearly it does not guarantee a global optimal solution. This approach turns out to be quite ineffective in the case that the function we want to optimize has a large number of local optima that may trap the algorithm. Simulated annealing gives a way to cope with such difficulty: we take into account a parameter called *temperature* and we allow moves to solutions of strictly worse values with a probability that decreases as the temperature approaches zero (*i.e.* as the process cools down) or when the difference of costs approaches infinity. After each iteration the temperature is lowered down, and the algorithm halts when the temperature is reduced to zero.

Note that simulated annealing does not seem to be a suitable framework for MCPS because it is not obvious how to define a simple neighborhood relation between feasible solutions for MCPS that could lead to a simulated annealing algorithm.

3 The Probe Selection Algorithms

Before presenting the algorithms, we need discuss a data structure that is common to both simulated annealing (SA) and Lagrangian relaxation (LR) algorithms. In both algorithms, given a set a set of clones \mathcal{C} , we need first to compute the set of all “interesting” probes \mathcal{P} and generate the matrix containing the fingerprint of each clone in \mathcal{C} induced by \mathcal{P} . Storing such a matrix for all possible clones would be too space consuming, since we want to apply our algorithms to instances of thousands of clones and more than one million probe candidates (in the case of $l = 10$). Since it is unlikely for probes that can distinguish only a small fraction of the pairs of clones to appear in a good solution, we introduce a threshold on $\Delta_p / \binom{m}{2}$ — the fraction of pairs from $\{fingerprint_p(c) : c \in \mathcal{C}\}$ that have to be distinct in order for the probe p to be considered *interesting*. It turns out that setting this threshold to .05 dramatically improves the running time of SA, while the quality of the solutions computed does not suffer noticeably.

3.1 The Simulated Annealing Algorithm for MIPS

We will first describe the simulated annealing algorithm for MIPS. As stated previously this framework requires a suitable notion of neighborhood between solutions. Since in our case each solution is a set of k probes, we define two sets of probes as *neighbors* if they can be obtained from each other by substituting exactly one of the probes.

The bottleneck in the algorithm is the evaluation of $|\Delta_S|$, the value of a solution S . The naive approach would be to compare the S -fingerprints for each pair of distinct clones in $O(m^2k)$ time. Instead, for each non-zero fingerprint value f we can compute the number γ_f of clones with fingerprint f , and then $|\Delta_S| = \frac{1}{2} \sum_f \gamma_f(n - \gamma_f)$. Since each component of $\text{fingerprint}_S(c)$ is an integer no larger than R , it is possible to sort the clones by radix sort using $\text{fingerprint}_S(c)$ as the key. Then we just have to check consecutive clones in order to get the numbers γ_f . Overall, this takes time $O(mk)$.

A high level description of the simulated annealing algorithm is given in Figure 1, where the initial and final temperatures depend on the size of the instance and on the costs of the initial solution and of all its neighbors (the idea is that we do not want to be stuck in the initial solution because the initial temperature is too low). The values of the parameter β has been set to 2000 empirically according to a preliminary experimental analysis.

Algorithm SA($\mathcal{P}, \mathcal{C}, k$)
Initialize S to be a set of k random probes from \mathcal{P} ;
 $t \leftarrow$ initial temperature;
repeat
 Let S' be a random neighbor of S ;
 $S \leftarrow S'$ with probability $\min\{1, \exp(\frac{|\Delta_S| - |\Delta_{S'}|}{t})\}$;
 Set $t \leftarrow \frac{\beta t}{\beta + t}$;
until $t \leq$ final temperature;
return(S);

Figure 1: An outline of the SA algorithm for MIPS.

Observe that the above simulated annealing algorithm also works if different objective functions are used in MIPS. For example, we have considered a more intuitive objective function where we want to maximize $|\Delta_S|$. Unfortunately, our experimental results (not included in this paper) have shown that this variant does not lead to good results. Other possible objective functions considered include the size of the largest cluster and the entropy of the distribution of clusters (these functions are proposed in [14]). One of the goals of our study has been to implement all such cost functions and determine which one gives the best result.

3.2 The Lagrangian Relaxation Algorithm for MCPS

Any feasible solution S can be represented by its characteristic vector $\mathbf{x} = \langle x_p | p \in \mathcal{P} \rangle$, where $x_p \in \{0, 1\}$ indicates whether probe p is in S or not. We can pose MCPS as a constrained integer linear program as follows:

$$\begin{aligned} \text{Minimize } |\mathbf{x}| &= \sum_{p \in \mathcal{P}} x_p & (1) \\ \text{subject to } \sum_{p \in \mathcal{P}} \delta_{p,c,d} \cdot x_p &\geq 1 \quad \forall (c,d) \in \mathcal{C}^2 \end{aligned}$$

where $\delta_{p,c,d}$ is the characteristic function of set Δ_p , that is, $\delta_{p,c,d} = 1$ iff $(c,d) \in \Delta_p$.

Our algorithm for MCPS is based on the Lagrangian relaxation framework. Given a non-negative vector $\boldsymbol{\lambda} = \langle \lambda_{c,d} | (c,d) \in \mathcal{C}^2 \rangle$ of Lagrangian multipliers, we consider the following integer program:

$$\text{Minimize } L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{p \in \mathcal{P}} x_p + \sum_{(c,d) \in \mathcal{C}^2} \lambda_{c,d} (1 - \sum_{p \in \mathcal{P}} \delta_{p,c,d} \cdot x_p) \quad (2)$$

Solving (2) is easy. Let $C_p(\boldsymbol{\lambda})$, for $p \in \mathcal{P}$, denote the *Lagrangian costs* associated with (2) defined by $C_p(\boldsymbol{\lambda}) = 1 - \sum_{(c,d) \in \mathcal{C}^2} \lambda_{c,d} \delta_{p,c,d}$. The objective function of (2) can then be written as $\sum_{p \in \mathcal{P}} C_p(\boldsymbol{\lambda}) x_p +$

$\sum_{(c,d)} \lambda_{c,d}$. Thus to minimize (2), we need to choose $x_p = 0$ for $C_p(\boldsymbol{\lambda}) > 0$, $x_p = 1$ for $C_p(\boldsymbol{\lambda}) < 0$, and for $C_p(\boldsymbol{\lambda}) = 0$ the value of x_p can be chosen arbitrarily. Then \boldsymbol{x} is an optimal solution of (2), and its value $L(\boldsymbol{\lambda}) = \sum_{p \in \mathcal{P}} C_p(\boldsymbol{\lambda})x_p + \sum_{(c,d)} \lambda_{c,d}$ is a lower bound for the solution of (1).

The vector \boldsymbol{x} computed above is not necessarily feasible for (1), that is, some inequality constraints of (1) may be violated. To obtain a feasible solution, the algorithm now starts from \boldsymbol{x} and greedily extends it to a feasible solution, by changing some coordinates of \boldsymbol{x} from 0 to 1 (that is, we add more probes to S). A naive greedy algorithm would add, at each step, the probe that covers most yet uncovered pairs. In our implementation we chose an alternative method recommended by Caprara *et al* [5]. With each probe p , we associate its score, $score_p(S, \boldsymbol{\lambda})$, defined as follows. Let $\mu_p(S) = |\Delta_p - \Delta_S|$ and $\gamma_p(S) = 1 - \sum_{(c,d) \in \Delta_p - \Delta_S} \lambda_{c,d}$ for each p . If $\mu_p(S) = 0$, set $score_p(S, \boldsymbol{\lambda}) = \infty$. Otherwise, $score_p(S, \boldsymbol{\lambda}) = \gamma_p(S)/\mu_p(S)$ for $\gamma_p(S) > 0$ and $score_p(S, \boldsymbol{\lambda}) = \gamma_p(S)\mu_p(S)$ for $\gamma_p(S) < 0$. At each step of the greedy algorithm we add a probe q that is not yet in \mathcal{P} and has a maximum score.

Procedure LRSOLUTION($\boldsymbol{\lambda}, S$)
 $C_p \leftarrow 1 - \sum_{(c,d) \in \mathcal{C}^2} \lambda_{c,d} \delta_{p,c,d}$ for each $p \in \mathcal{P}$
 $S \leftarrow \{p \in \mathcal{P} : C_p < 0\}$
end

Procedure FEASIBLEEXTENSION($\boldsymbol{\lambda}, S$)
while $\Delta_S \neq \mathcal{C}^2$ **do**
 $q \leftarrow$ a probe in $\mathcal{P} - S$ with maximum $score_q(\boldsymbol{\lambda})$
 $S \leftarrow S \cup \{q\}$
end

Figure 2: The procedures for solving the Lagrangian relaxation and computing a feasible solutions.

A pseudocode for the two steps described above is given in Figure 2. In the pseudocode we use the set notation instead of the vector notation. Procedure LRSOLUTION computes an optimal solution to the Lagrangian relaxation for a given Lagrangian multiplier. Procedure FEASIBLEEXTENSION extends the solution obtained from LRSOLUTION to a feasible solution.

Another task we need to solve is finding a good multiplier vector $\boldsymbol{\lambda}$, that is, one that gives a near-optimal lower bound. This is done by a commonly used heuristic called *subgradient optimization*, introduced in [13] for the TSP problem, which exploits the fact that the gradient of $L(\boldsymbol{x}, \boldsymbol{\lambda})$, for a fixed \boldsymbol{x} , is $\nabla \boldsymbol{\lambda} = \langle \nabla_{c,d} \boldsymbol{\lambda} | (c, d) \in \mathcal{C}^2 \rangle$, where $\nabla_p \boldsymbol{\lambda} = 1 - \sum_{p \in \mathcal{P}} \delta_{p,c,d} x_p$. Starting from $\boldsymbol{\lambda}$, we compute a sequence $\boldsymbol{\lambda}^0 = \boldsymbol{\lambda}, \boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \dots$ according to the formula:

$$\boldsymbol{\lambda}^{i+1} = \max \left\{ \boldsymbol{\lambda}^i + \alpha \frac{|\boldsymbol{x}^*| - L(\boldsymbol{\lambda}^i)}{\|\nabla \boldsymbol{\lambda}^i\|^2} \nabla \boldsymbol{\lambda}^i, \mathbf{0} \right\} \quad (3)$$

where \boldsymbol{x}^* is the best feasible solution found so far and $\alpha > 0$ is a parameter that is chosen empirically and is updated dynamically by the algorithm. A pseudocode for this algorithm is given in Figure 3. Here, the variable **LB** stands for the best lower bound on the size of an optimal solution found so far. The LR algorithm is summarised in Figure 4. Note that, the algorithm proves not only a solution, but also a lower bound which is useful in estimating the performance of the algorithm.

Implementation and Improvements. The implementation of the LR algorithm has presented a number of challenges to us that are different from the ones tackled in [5], where a LR heuristic for SET COVER is presented. In [5], the authors had to deal with sparse instances where the constraint matrix contains up to 5500 rows and 1100000 columns, whereas in our case we had to solve instances of up to 1200000 rows (*i.e.* pairs of clones) and 5000 columns (*i.e.* probes). Our constraint matrices are not

```

Procedure SUBGRADOPTIMIZE( $\lambda, S^*, \mathbf{LB}$ )
 $\alpha \leftarrow \text{InitialAlpha}$ 
while  $\alpha > \text{MinAlpha}$  do
   $\text{PrevLB} \leftarrow \mathbf{LB}$ 
  repeat  $\text{LBIncreaseTrials}$  times
     $\text{LRSolution}(\lambda, S)$ 
     $L \leftarrow L(x, \lambda)$ 
     $\mathbf{LB} \leftarrow \max\{L, \mathbf{LB}\}$ 
     $\text{FEASIBLESOLUTION}(\lambda, S)$ 
    if  $|S| < |S^*|$  then  $S^* \leftarrow S$ 
     $\lambda \leftarrow \max\left\{\lambda + \alpha \frac{|S^*| - L}{\|\nabla \lambda\|^2} \nabla \lambda, \mathbf{0}\right\}$ 
  if  $\mathbf{LB} = \text{PrevLB}$  then  $\alpha \leftarrow \alpha/2$ 
end

```

Figure 3: The procedure for subgradient optimization.

```

Algorithm LR( $\mathcal{P}, \mathcal{C}$ )
 $S^* \leftarrow \emptyset$ 
 $\lambda_{c,d} \leftarrow \min_{\Delta_p \ni (c,d)} \left\{ \frac{1}{|\Delta_p|} \right\}$  for all  $(c, d) \in \mathcal{C}$ 
 $\mathbf{LB} \leftarrow 0$ 
SUBGRADOPTIMIZE( $\lambda, S^*, \mathbf{LB}$ )
 $\text{Accuracy} \leftarrow |S^*|/\mathbf{LB}$ 
Print( $S^*, \text{Accuracy}$ )
end

```

Figure 4: The LR Algorithm.

as sparse as the ones dealt in [5], and they could not be stored in the main memory (we estimate that the matrix for our dataset 1 requires about 4GB of memory). We solved this problem by computing each “block” of the constraint matrix and store it in the main memory only when it is needed.

The iterative nature of the LR algorithm suggests a natural way of improving the running time: first solve MCPS on a smaller instance and then use the solution computed as an initial solution to the larger instance. Clearly this idea, in order to be effective, requires that the smaller instance is computed carefully, so that the smaller instance is representative of the larger one. One method is to form hierarchical clusters of clones and pick a representative from each cluster. We have implemented a variant of the algorithm in [17] to cluster the clones according to the Hamming distance between the fingerprints of clones with respect to the probes in \mathcal{P} .

Our initial tests of the LR algorithm on real rDNA data suggested that the algorithm might not be very effective when the constraint matrix is not sufficiently sparse. So, we have considered the following *sparsification* technique: use a fast (*e.g.* greedy) algorithm to find a small set of probes distinguishing a large fraction of the pairs of clones and remove these probes and pairs of clones from further consideration. This would often result in a very sparse and much smaller constraint matrix.

4 Experimental Results

We have implemented all algorithms described in the previous section, and performed testing on several real rDNA datasets. The data used includes:

- dataset 1 contains 1158 small-subunit ribosomal genes from GenBank (NCBI). For this dataset, the nucleotide sequence of each gene was edited such that it contains the sequence between two

highly conserved primers (27F, AGAGTTTGATCMTGGCTCAG; 1492R TACGGYTACCTGTTACGACTT) but not the primer sequences themselves.

- dataset 2 contains 131 large-subunit ribosomal genes from the Ribosomal Database Project II. These sequences have not been edited.
- dataset 3 contains a random set of 5000 eubacteria rDNA sequences from GenBank.
- dataset 4 contains a different random set of 2000 eubacteria rDNA sequences from GenBank.

For each of the datasets, we have tested the SA algorithm for MIPS using various combinations of values of k (the expected number of probes, up to 20), values of l (the length of probes, between 5 and 10), the two distinguishability criteria (binary or non-binary, *i.e.* frequency of occurrence up to 4), as well as various objective functions including the original objection in MIPS to minimize the total number of indistinguished pairs of clones (SA+Pairs), the objective function to minimize the size of the largest resulting cluster (SA+Largest), and the objective function to maximize the entropy of the distribution of clusters (SA+entropy). In particular, the entropy function was considered here because, as observed in [15], entropy might give a better indication than frequency on the performance of a probe set in the presence of experimental errors. For each combination of these parameters, we ran the algorithm with 20 random starts and report the best solution found. The running time of these tests ranged from 30 to 90 minutes on a PC with Pentium III/500 Mhz cpu and 128MB memory. Due to the page limit, here we report only some of the test results. The complete test results will be reported in the full version of the paper.

Figure 5 shows a comparison of the three variants of the SA algorithm and the greedy heuristic proposed in [15, 21]. In the figure, we consider the sequences in dataset 3 (the largest dataset), probe length 6 (the most useful length for our application), both distinguishability criteria, and the number of expected probes ranging from 8 to 20. The comparison is made in terms of the total number of indistinguished pairs of clones, the entropy, the size of the largest cluster, and the number of clusters. A similar summary of test results for datasets 1 and 4 are shown in Figures 6 and 7 given in the appendix. We observe that in general, the SA algorithm that attempts to minimize the total number of indistinguished pairs of clones (*i.e.* SA+Pairs) has the best overall performance. Moreover, it outperforms the greedy algorithm in almost all categories, except with respect to the number of resulting clusters where its performance is comparable to that of the greedy algorithm. Its results are highly satisfactory. For example, in the case of binary distinguishability, the algorithm SA+Pairs was able to find 12 probes of length 6 that distinguish $1 - 10000/\binom{5000}{2} \approx 99.92\%$ of the pairs of clones in dataset 3. These probes induce about 2000 clusters (which is almost a half of the maximum number of clusters that can be induced by 12 probes) with its largest cluster having size 30 and its entropy being 10.5 (the maximum entropy is $\log 2^{12} = 12$). The results are even better in the case of non-binary distinguishability. It is interesting to observe that the performance of these algorithms does not improve too much once the number of probes is more than 12. Our tests also suggest that probes of length 6 provide the best solutions among all probe lengths between 5 and 10 (not all data is included here).

We have tested the LR algorithm for MCPS on datasets 1 and 2. The results are summarized in Table 1, where we considered all combinations of various distinguishability criteria and probe length values. For example, in the case of probe length 5 and non-binary distinguishability, the algorithm is able to find a solution of 23 probes distinguishing all pairs of the 1158 clones in dataset 1. The numbers in the parentheses indicate (meaningful) lower bounds obtained by the LR algorithm combined with the sparsification technique described in the previous section, and demonstrate that the algorithm was able to obtain optimal or close-to-optimal probe sets for dataset 2. Unfortunately, as of the time of submission, we were not able to use the technique to obtain nontrivial lower bounds for dataset 1. We suspect that this could be due to the fact that the density of the constraint matrix for dataset 1 is far greater than that for dataset 2. We are working on the refinement of the sparsification

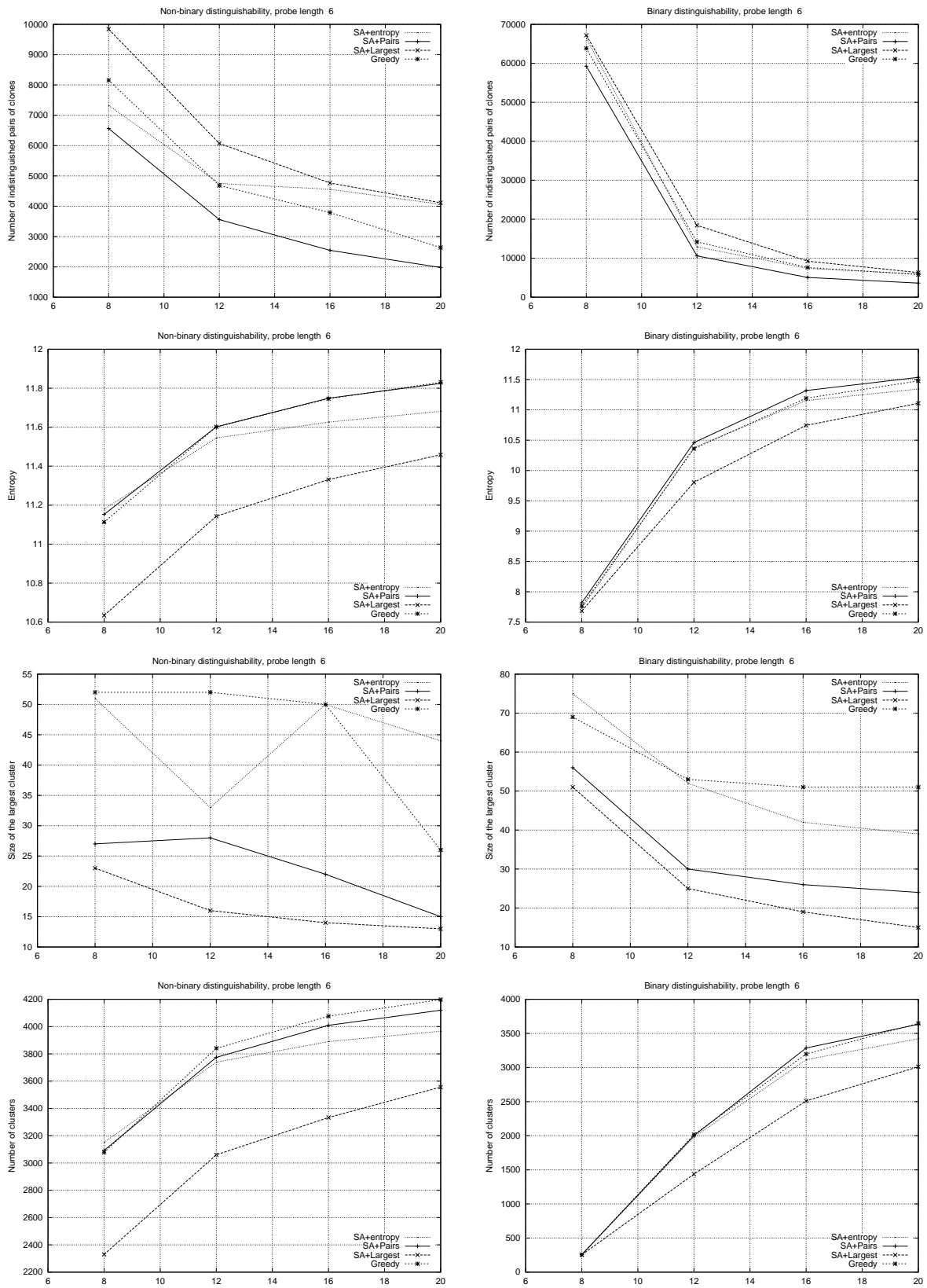


Figure 5: Some comparative test results on dataset 3

technique and hope to report general lower bound results in the final version of the paper. Again, it is observed that non-binary distinguishability greatly improves the performance and for both datasets short probes (of length 5 or 6) seem to provide smaller solutions than long probes (of length 8) do.

Dataset 1			Dataset 2	
Distinguishability		Length of probes	Distinguishability	
binary	non-binary		binary	non-binary
46	23	5	18 (18)	11 (8)
48	29	6	17 (14)	14 (10)
57	46	8	23 (23)	21 (18)

Table 1: Results of the LR algorithm on datasets 1 and 2.

5 Concluding Remarks

In this paper, we have presented two efficient heuristics for selecting minimal probe sets to be used in the oligonucleotide fingerprinting of rDNA clones. The algorithms are being used in the analysis of microbial communities. We have tested the algorithms on four sets of rDNA sequences collected from public databases, and the results are very promising. We next plan to test the algorithms on appropriate groups of cDNA sequences (there are many more cDNA sequences available than rDNA sequences in the public domain). A key to the utility of these algorithms is their efficiency. At the moment, the speed of the LR algorithm is far from being satisfactory mainly because of its requirement of large memory space. We plan to consider some efficient external memory data structures and find ways to speed up the algorithm.

References

- [1] R. I. Amann, W. Ludwig, and K.-H. Schleifer. Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiological Reviews*, 59:143–169, 1995.
- [2] S. M. Barns, R. E. Fundyga, M. W. Jeffries, and N. R. Pace. Remarkable archaeal diversity detected in a yellowstone national park hot spring environment. *Proceedings of the National Academy of Sciences USA*, 91:1609–1613, 1994.
- [3] J. Borneman, P. W. Skroch, K. M. O’Sullivan, J. A. Palus, J. A. Rumjanek, J. L. Jansen, J. Nienhuis, and E. W. Triplett. Molecular microbial diversity of an agricultural soil in wisconsin. *Applied and Environmental Microbiology*, 62:1935–1943, 1996.
- [4] J. Borneman and E. W. Triplett. Molecular microbial diversity in soils from eastern amazonia: Evidence for unusual microorganisms and microbial population shifts associated with deforestation. *Applied and Environmental Microbiology*, 63:2647–2653, 1997.
- [5] A. Caprara, M. Fischetti, and P. Toth. A heuristic method for the set covering problem. *Operations Research*, 47(5):730–743, 1999.
- [6] A. Cutichia, J. Arnold, and W. Timberlake. Pcap: probe choice and analysis package – a set of programs to aid in choosing synthetic oligomers for contig mapping. *CABIOS*, 9:201–203, 1993.
- [7] R. Drmanac. cDNA screening by array hybridization. *Methods in Enzymology*, 303:165–178, 1999.
- [8] S. Drmanac and R. Drmanac. Processing of cDNA and genomic kilobase-size clones for massive screening mapping and sequencing by hybridization. *Biotechniques*, 17:328–336, 1994.
- [9] S. Drmanac, N. A. Stavropoulos, I. Labat, J. Vonau, B. Hauser, M. B. Soares, and R. Drmanac. Gene representing cDNA clusters defined by hybridization of 57,419 clones from infant brain libraries with short oligonucleotide probes. *Genomics*, 37:29–40, 1996.

- [10] Y. Fu, W. Timberlake, and J. Arnold. On the design of genome mapping experiments using short synthetic oligonucleotides. *Biometrics*, 48:337–359, 1992.
- [11] M. Garey and D. Johnson. *Computer and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, 1979.
- [12] S. J. Giovannoni, T. B. Britschgi, C. L. Moyer, and K. G. Field. Genetic diversity in sargasso sea bacterioplankton. *Nature*, 345:60–63, 1990.
- [13] M. Held and R. M. Karp. The traveling salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1:6–25, 1971.
- [14] S. Hennig, R. Herwig, M. Clark, P. Aanstad, A. Musa, J. O’Brien, C. Bull, U. Radelof, G. Panopoulou, A. J. Poustka, and H. Lehrach. A data-analysis pipeline for large-scale gene expression analysis. In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB2000)*, pages 165–173, 2000.
- [15] R. Herwig, A. Schmidt, M. Steinfath, J. O’Brien, H. Seidel, S. Meier-Ewert, H. Lehrach, and U. Radelof. Information theoretical probe selection for hybridisation experiments. *Bioinformatics*, 2000. to appear.
- [16] D. S. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS publisher, 1997.
- [17] D. S. Hochbaum and D. B. Shmoys. A unified approach to approximate algorithms for bottleneck problems. *Journal of ACM*, 33(3), 1986.
- [18] S. Kirkpatrick, C. D. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 4598(220):671–680, 1983.
- [19] F. Li and G. D. Stormo. Selecting optimum DNA oligos for microarrays. In *Proc. IEEE International Symposium on Bio-Informatics and Biomedical Engineering, Arlington, VA.*, 2000.
- [20] W. T. Liu, T. L. Marsh, H. Cheng, and L. J. Forney. Characterization of microbial diversity by determining terminal restriction fragment length polymorphisms of genes encoding 16s rna. *Applied and Environmental Microbiology*, 63:4516–4522, 1997.
- [21] E. Maier, S. Meier-Ewert, A. R. Ahmadi, J. Curtis, and H. Lehrach. Application of robotic technology to automated sequence fingerprint analysis by oligonucleotide hybridisation. *Journal of Biotechnology*, 35:191–203, 1994.
- [22] S. Meier-Ewert, J. Lange, H. Gerst, R. Herwig, A. Schmitt, J. Freund, T. Elge, R. Mott, B. Hermann, and L. H. Comparative gene expression profiling by oligonucleotide fingerprinting. *Nucleic Acids Research*, 26:2216–2223, 1998.
- [23] G. Muyzer, E. C. D. Waal, and A. G. Uitterlinden. Profiling of complex microbial populations by denaturing gradient gel electrophoresis analysis of polymerase chain reaction amplified genes coding for 16s rna. *Applied and Environmental Microbiology*, 59:695–700, 1993.
- [24] N. R. Pace. A molecular view of microbial diversity and the biosphere. *Science*, 276:734–740, 1997.
- [25] G. D. Panopoulou, M. D. Clark, H. Gerst, R. Herwig, L. Z. Holland, N. D. Holland, and H. Lehrach. Large-scale identification of amphioxus genes from different developmental stages using oligonucleotide fingerprinting. *Developmental Biology*, 198:200, 1998.
- [26] A. Poustka, R. Herwig, A. Krause, S. Hennig, S. Meier-Ewert, and H. Lehrach. Toward the gene catalogue of sea urchin egg cDNA library highly normalized by oligonucleotide fingerprinting. *Genomics*, 59:122–133, 1999.
- [27] U. Radelof, S. Hennig, P. Seranski, M. Steinfath, J. Ramser, R. Reinhardt, A. Poustka, F. Francis, and H. Lehrach. Preselection of shotgun clones by oligonucleotide fingerprinting: an efficient and high throughput strategy to reduce redundancy in large-scale sequencing projects. *Nucleic Acids Research*, 26:5358–5364, 1998.
- [28] V. Torsvik, J. Goksoyr, and F. L. Daae. High diversity in DNA of soil bacteria. *Appl. Environ. Microbiol.*, 56:782–787, 1990.
- [29] D. Ward, M. M. Bateson, R. Weller, and A. L. Ruff-Roberts. Ribosomal analysis of microorganisms as they occur in nature. *Advances in Microbial Ecology*, 12:219–286, 1992.
- [30] C. R. Woese. Bacterial evolution. *Microbiological Reviews*, (51):221–271, 1987.

Appendix: Some More Experimental Results

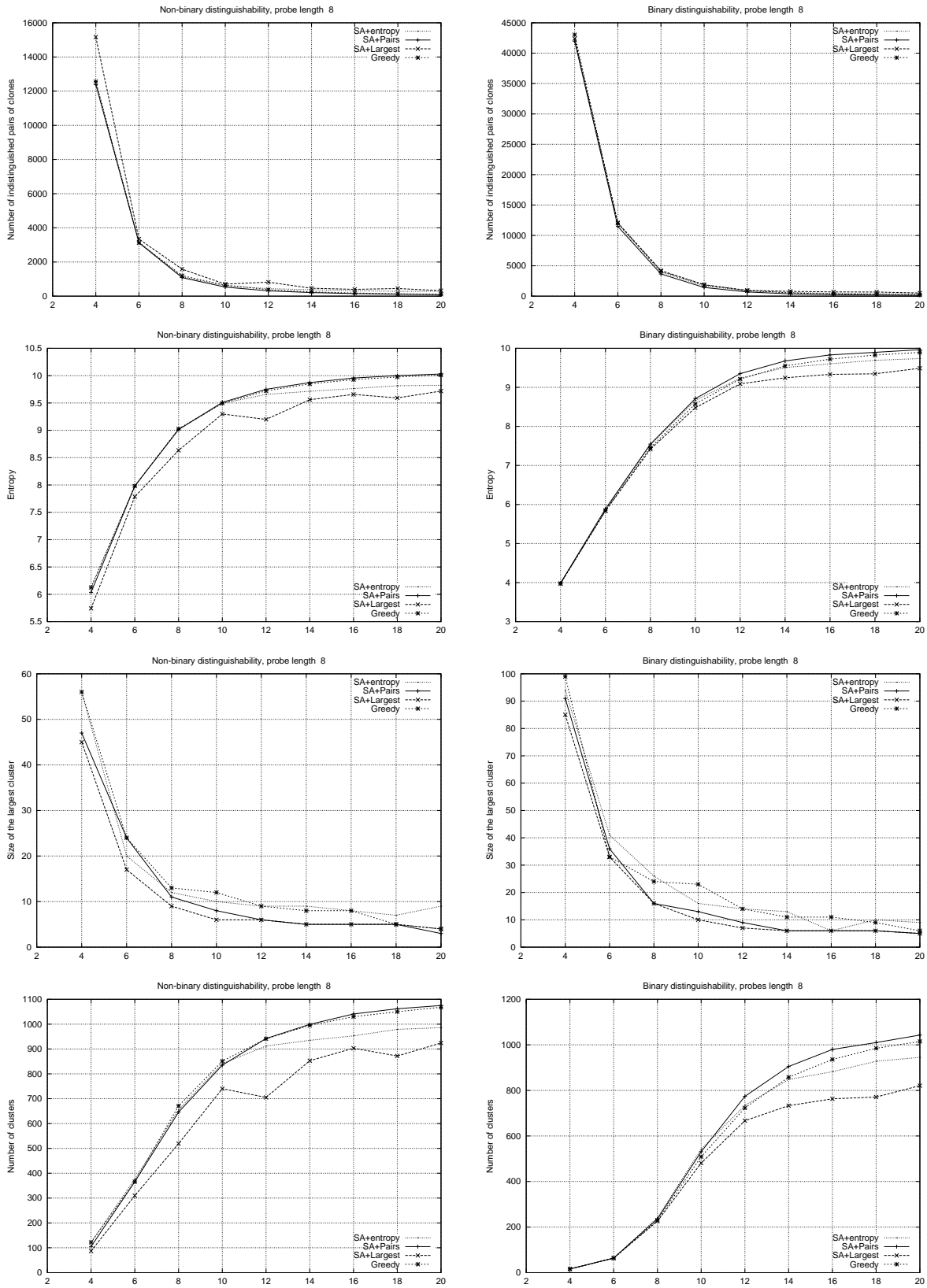


Figure 6: Some comparative test results on dataset 1

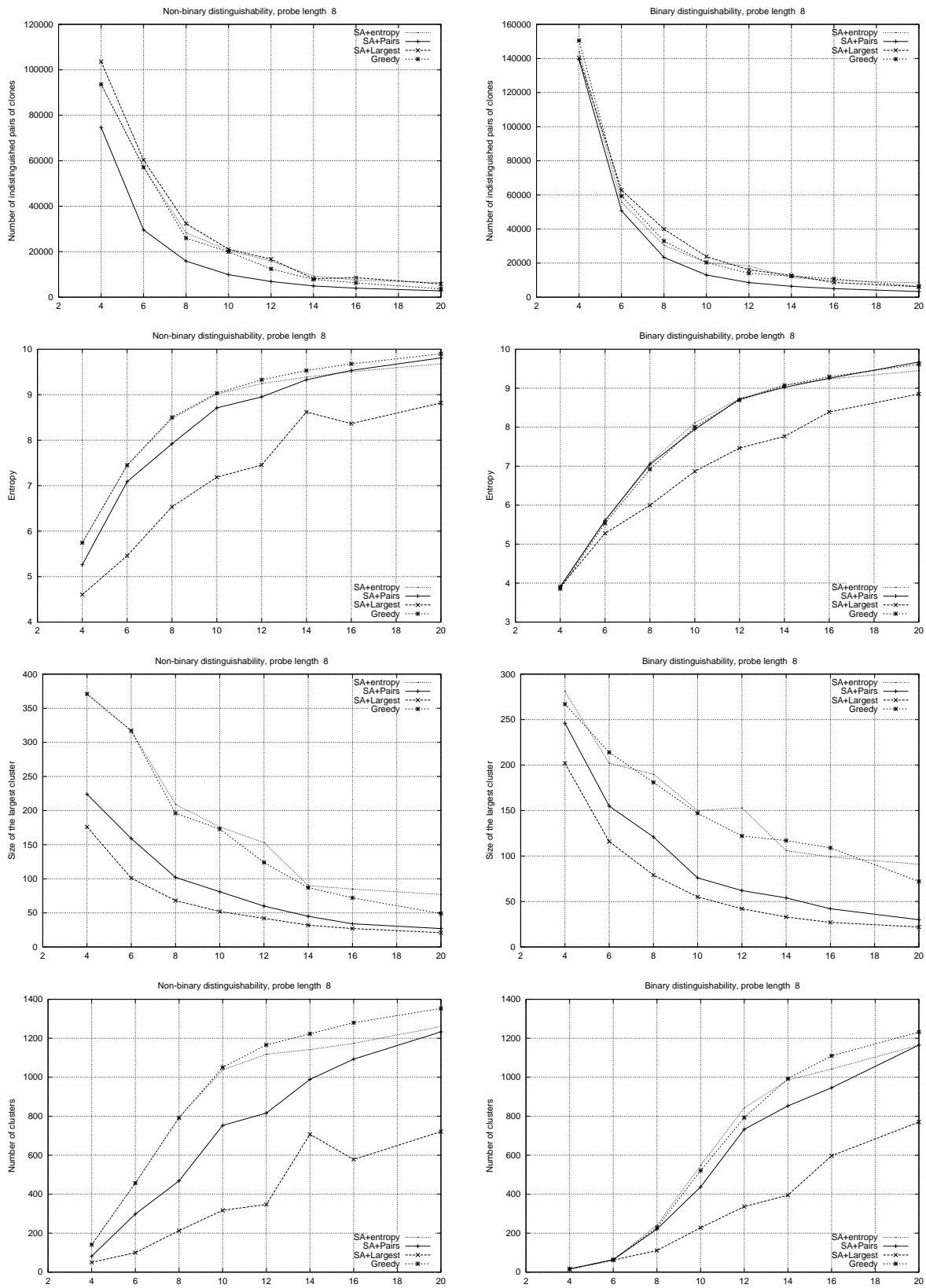


Figure 7: Some comparative test results on dataset 4.