Journal of Bioinformatics and Computational Biology

ALGORITHMIC APPROACHES TO SELECTING CONTROL CLONES IN DNA ARRAY HYBRIDIZATION EXPERIMENTS *

QI FU*, ELIZABETH BENT†, JAMES BORNEMAN†,
MAREK CHROBAK*and NEAL E. YOUNG*

*Department of Computer Science *, Department of Plant Pathology †,*
University of California,
Riverside, CA 92521. U.S.A.
*(qfu, marek, neal)@cs.ucr.edu **
(bente, james.borneman)@ucr.edu †

We study the problem of selecting control clones in DNA array hybridization experiments. The problem arises in the OFRG method for analyzing microbial communities. The OFRG method performs classification of rRNA gene clones using binary fingerprints created from a series of hybridization experiments, where each experiment consists of hybridizing a collection of arrayed clones with a single oligonucleotide probe. This experiment produces analog signals, one for each clone, which then need to be classified, that is, converted into binary values 1 and 0 that represent hybridization and non-hybridization events. In addition to the sample rRNA gene clones, the array contains a number of control clones needed to calibrate the classification procedure of the hybridization signals. These control clones must be selected with care to optimize the classification process. We formulate this as a combinatorial optimization problem called *Balanced Covering*. We prove that the problem is $\mathbb{NP}$-hard, and we show some results on hardness of approximation. We propose approximation algorithms based on randomized rounding and we show that, with high probability, our algorithms approximate well the optimum solution. The experimental results confirm that the algorithms find high quality control clones. The algorithms have been implemented and are publicly available as part of the software package called CloneTools.

Keywords: Control selection; DNA array; balanced covering; linear programming; randomized rounding.

*A preliminary version of this paper appears in the Proceedings of Asia-Pacific Bioinformatics Conference 2007. Electronic version of an article published as Journal of Bioinformatics and Computational Biology 5(4) 937–961, 2007, DOI 10.1142/S0219720007002977, ©World Scientific Publishing Company, http://www.worldscinet.com/jbcb/

1. Introduction

Background. We study the problem of selecting control clones for DNA array hybridization experiments. The specific version of the problem that we address arises in the context of the OFRG (Oligonucleotide Fingerprinting of Ribosomal RNA Genes) method, that we describe below, although our approach is also relevant to other applications of DNA microarray technology.

OFRG ([5], [8], [10], [11], [12]) is a technique for analyzing microbial communities that classifies rRNA gene clones into taxonomic clusters based on binary fingerprints created from hybridizations with a collection of oligonucleotide probes. More specifically, in OFRG, clone libraries from a sample under study (e.g., fungi or bacteria from an environmental sample) are constructed using PCR primers. These cloned rRNA gene fragments are immobilized on nylon membranes and then subjected to a series of hybridization experiments, with each experiment using a single radiolabeled DNA oligonucleotide probe. This experiment produces analog signals, one for each clone, which then need to be classified, that is, converted into binary values 1 and 0 that represent hybridization and non-hybridization events. Overall, this process creates a hybridization fingerprint for each clone, which is a vector of binary values indicating which probes bind with this clone and which do not. The clones are then identified by clustering their hybridization fingerprints with those of known sequences and by nucleotide sequence analysis of representative clones within a cluster.

In addition to sample clones, the array contains a number of *control clones*, with known nucleotide sequences, used to calibrate the classification procedure of hybridization signals. Consider a hybridization experiment with a probe p. Signal intensities from its hybridizations with the control clones produce two distributions: one from control clones that match p (e.g., they contain p or p's reverse complement and thus should hybridize with it) and the other from control clones that do not. This information is used to determine, via appropriate statistical techniques, p's signal intensity threshold t. Once t has been determined, we can classify signal intensities for sample clones as follows: signals above t are interpreted as 1's (hybridization events) while those below are represented by 0's (non-hybridizations).

The quality of information obtained from hybridizations depends critically on the accuracy of the signal classification process. In particular, the control clones should be more or less equally distributed in terms of their ability to bind or not bind with each probe from a given probe set. In prior OFRG work, control clones were selected arbitrarily, often producing control clones with very skewed distribution of binding/non-binding with some probes. As an example, from a set of 100 control clones, only two might bind with a specific probe. The signal classification for this probe would be very unreliable, as it would be based on signal intensities from hybridization with only two control clones.

Problem formulation. Our control-clone selection problem can be then formulated as follows: We are given a collection C of candidate control clones and a set P

of oligonucleotide probes to be used in the hybridization experiments. From among the candidate clones in C, we want to select a set $D \subseteq C$ of s control clones such that each probe in P hybridizes with roughly half of the clones in D.

This gives rise to a combinatorial optimization problem that we call *Balanced Covering*[a]. The instance is given as a pair $\langle G, s\rangle$, where $G = (C, P, E)$ is a bipartite graph and $s \leq |C|$ is an integer. C represents the clone set, P is the probe set, and the edges in E represent potential hybridizations between clones and probes, that is, $(c, p) \in E$ iff c contains p or the reverse complement of p. For $p \in P$ and $D \subseteq C$, let $\deg_D(p)$ be the number of neighbors of p in D (that is, the number of clones in D that hybridize with p). Throughout the paper, unless stated otherwise, by m we will denote the cardinality of C and by n the cardinality of P.

Example. We illustrate the concept with a small example. (The realistic data sets are typically considerably larger.) Let $P = \{p_1, p_2, ..., p_7\}$ be the following probe set:

p_1	p_2	p_3	p_4	p_5	p_6	p_7
CTGGC	TACAT	CGGCG	GCTGG	CGCTA	GCCTA	ATACA

The set of control clones $C = \{c_1, c_2, ..., c_8\}$ and the resulting bipartite graph G are shown below (G is represented by its $C \times P$ adjacency matrix).

		p_1	p_2	p_3	p_4	p_5	p_6	p_7
c_1	ATTGAACG<u>CTGGC</u>GGCAGGCCTAACACATGCAAGTCGGACGGTAG	1	0	0	1	0	1	0
c_2	GACGAACA<u>GCCAG</u>GGCGTGCTTCGGCGATGCAAGTCGAGCGCTAA	1	0	1	0	1	0	0
c_3	ATTTTACG<u>CTGGC</u>GGCAGGCCTAACACATGCAAGTCGAAAAGTAG	1	0	0	1	0	1	0
c_4	ACGCTAGCGGGATGCTTTACACATGCAAGTCGAACGGCAATACAT	0	1	0	0	1	0	1
c_5	ACGAACG<u>CTGGC</u>GGCGTGCCTAATACATGCAAGTCGAACGCTTCT	1	1	1	1	0	1	1
c_6	ACGAACGG<u>CCAG</u>GGCGTGGATTAGGCATGCAACGGCGACGCTGGA	1	0	1	1	0	1	0
c_7	GATGAACGCTAGCGGCAGGCTTAATACATGCAAGTCGAACGGCAG	0	1	0	0	1	0	1
c_8	GACGAACG<u>CTGGC</u>GGCGTGCTTAACACATGCAAGTCGAACGGAAA	1	0	1	1	0	0	0

In G, we have an edge (c_i, p_j) if p_j matches c_i, that is, p_j or its reverse complement appears in c_i. For example, p_1 appears in c_1, c_3, c_5 and c_8, and its reverse complement GCCAG appears in c_2 and c_6. (These occurrences of p_1 are underlined.) There is no edge (c_4, p_1) and (c_7, p_1) since c_4 and c_7 do not contain either p_1 or p_1's reverse complement.

Now suppose that we want to select $s = 6$ control clones from C. The probe degree sequence with respect to $D_1 = \{c_1, c_2, c_3, c_5, c_6, c_8\}$ is $(6, 1, 4, 5, 1, 4, 1)$, while the probe degree sequence with respect to $D_2 = \{c_2, c_4, c_5, c_6, c_7, c_8\}$ is $(4, 3, 4, 3, 3, 2, 3)$. Thus D_2 would be considered a better set of control clones, since more degrees are closer to $s/2 = 3$.

[a]There have been some discussions on the *Balanced Set Cover* (see [1], [7]) problem, however, they are not directly related to *Balanced Covering* problem discussed in this paper.

Generally, as mentioned earlier, our goal is to find a set $D \subseteq C$ of cardinality s such that, for each $p \in P$, $\deg_D(p)$ is close to $s/2$. Several objective functions can be studied. To measure the deviation from the perfectly balanced cover, for a given probe p, we can compute either $\min\{\deg_D(p), s - \deg_D(p)\}$ or $|\deg_D(p) - s/2|$. The objective function can be obtained by considering the average of these values or the worst case over all probes. This gives rise to four objective functions:

$$\begin{aligned}
\text{maximize} \quad & \mathcal{C}_{min}(D) = \min_{p\in P} \min\{\deg_D(p), s - \deg_D(p)\} \\
\text{maximize} \quad & \mathcal{C}_{avg}(D) = \tfrac{1}{n} \sum_{p\in P} \min\{\deg_D(p), s - \deg_D(p)\} \\
\text{minimize} \quad & \mathcal{D}_{max}(D) = \max_{p\in P} |\deg_D(p) - s/2| \\
\text{minimize} \quad & \mathcal{D}_{avg}(D) = \tfrac{1}{n} \sum_{p\in P} |\deg_D(p) - s/2|
\end{aligned}$$

where each function needs to be optimized over all choices of D. There are certain relations among these functions, for an instance, maximizing $\mathcal{C}_{min}(D)$ and $\mathcal{C}_{avg}(D)$ is equivalent to minimizing $\mathcal{D}_{max}(D)$ and $\mathcal{D}_{avg}(D)$, respectively, since $\mathcal{D}_{max}(D) = s/2 - \mathcal{C}_{min}(D)$, and $\mathcal{D}_{avg}(D) = s/2 - \mathcal{C}_{avg}(D)$. Throughout the paper, the four optimization problems corresponding to these functions will be denoted by $\mathsf{BCP}\_\mathcal{C}_{min}$, $\mathsf{BCP}\_\mathcal{C}_{avg}$, $\mathsf{BCP}\_\mathcal{D}_{max}$, and $\mathsf{BCP}\_\mathcal{D}_{avg}$.

Let $\langle G, s\rangle$ be an instance of Balanced Covering. By $\mathcal{C}^*_{min}(G, s) = \max_D \mathcal{C}_{min}(D)$ we denote the optimal value of $\mathcal{C}_{min}(D)$. If $\mathcal{A}$ is an algorithm for $\mathsf{BCP}\_\mathcal{C}_{min}$, then $\mathcal{C}^{\mathcal{A}}_{min}(G, s)$ denotes the value computed by $\mathcal{A}$ on input $\langle G, s\rangle$. We use similar notations, $\mathcal{C}^*_{avg}(G, s)$, $\mathcal{C}^{\mathcal{A}}_{avg}(G, s)$, etc., for all the other objective functions introduced above.

Results. In this paper we show several analytical and experimental results on Balanced Covering. In Section 2 we prove that all versions of Balanced Covering are $\mathbb{NP}$-hard. In particular, it is $\mathbb{NP}$-complete to decide whether there is a perfectly balanced cover with s clones, as well as to decide whether there is a size-s cover where each probe is covered by at least one but not all clones. These results immediately imply that (unless $\mathbb{P}$=$\mathbb{NP}$), there are no polynomial-time approximation algorithms for $\mathsf{BCP}\_\mathcal{D}_{max}$, $\mathsf{BCP}\_\mathcal{D}_{avg}$, and $\mathsf{BCP}\_\mathcal{C}_{min}$.

Stronger hardness-of-approximation results are shown in Section 3. For example, for $\mathsf{BCP}\_\mathcal{D}_{avg}$, we show that approximating the optimum is hard even if we allow randomization and an additive term in the performance bound. More specifically, we prove that, unless $\mathbb{RP} = \mathbb{NP}$, there is no randomized polynomial-time algorithm $\mathcal{A}$ that for some constants $\alpha > 0$ and $0 < \epsilon < 1$ satisfies $\mathrm{Exp}[\mathcal{D}^{\mathcal{A}}_{avg}] \le \alpha \mathcal{D}^*_{avg} + \beta \frac{1}{n}(mn)^{1-\epsilon}$. ($\mathbb{RP}$ is the class of decision problems that can be solved in randomized polynomial time with one-side error.) For $\mathsf{BCP}\_\mathcal{C}_{min}$, we show that there is no polynomial-time algorithm that computes a solution with the objective value at least $\mathcal{C}^*_{min} - (1-\epsilon)\ln n$ or $\epsilon \mathcal{C}^*_{min}$, unless $\mathbb{NP}$ has slightly superpolynomial-time algorithms. Our results on hardness of approximation are summarized in Ta-

Table 1. Hardness results and algorithms

	Hardness Results	Randomized Polynomial-Time Alg.
$\mathsf{BCP}\_\mathcal{C}_{min}$	No polynomial-time algorithm $\mathcal{A}$ satisfies $\mathcal{C}^{\mathcal{A}}_{min} \geq \mathcal{C}^{*}_{min} - (1-\epsilon)\ln n$ or $\mathcal{C}^{\mathcal{A}}_{min} \geq \epsilon\mathcal{C}^{*}_{min}$, unless $\mathbb{NP} \subseteq \mathbb{DTIME}(n^{O(\log\log n)})$.	Algorithm RCM2 s.t., $\mathcal{C}^{\mathrm{RCM2}}_{min} \geq \mathcal{C}^{*}_{min} - O\left(\sqrt{\mathcal{C}^{*}_{min}\ln n}\right)$, with probability at least $\frac{1}{2}$.
$\mathsf{BCP}\_\mathcal{C}_{avg}$	No randomized polynomial-time algorithm $\mathcal{A}$ satisfies $\mathrm{Exp}[\mathcal{C}^{\mathcal{A}}_{avg}] \geq \mathcal{C}^{*}_{avg} - \beta\frac{1}{n}(mn)^{1-\epsilon}$, unless $\mathbb{RP} = \mathbb{NP}$.	Algorithm RCA2 s.t., $\mathrm{Exp}[\mathcal{C}^{\mathrm{RCA2}}_{avg}] \geq \mathcal{C}^{*}_{avg} - O\left(\sqrt{\mathcal{C}^{*}_{avg}}\right)$.
$\mathsf{BCP}\_\mathcal{D}_{max}$	No polynomial-time algorithm that $\mathcal{A}$ satisfies $\mathcal{D}^{\mathcal{A}}_{max} \leq \mathcal{D}^{*}_{max} + (1-\epsilon)\ln n$, unless $\mathbb{NP} \subseteq \mathbb{DTIME}(n^{O(\log\log n)})$.	Algorithm RDM s.t., $\mathcal{D}^{\mathrm{RDM}}_{max} \leq \mathcal{D}^{*}_{max} + O\left(\sqrt{s\ln n}\right)$, with probability at least $\frac{1}{2}$
$\mathsf{BCP}\_\mathcal{D}_{avg}$	No randomized polynomial-time algorithm $\mathcal{A}$ satisfies $\mathrm{Exp}[\mathcal{D}^{\mathcal{A}}_{avg}] \leq \alpha\mathcal{D}^{*}_{avg} + \beta\frac{1}{n}(mn)^{1-\epsilon}$, unless $\mathbb{RP} = \mathbb{NP}$.	

$\alpha, \beta > 0$ and $0 < \epsilon < 1$ are any constants.

ble 1.

Then, in Section 4, we propose a polynomial-time randomized rounding algorithm RCM for $\mathsf{BCP}\_\mathcal{C}_{min}$. The algorithm solves the linear relaxation of the integer program for $\mathsf{BCP}\_\mathcal{C}_{min}$, and then uses the solution to randomly pick an approximately balanced cover. We show that, with probability at least $\frac{1}{2}$, RCM's solution has objective value at least $\mathcal{C}^{*}_{min} - O(\sqrt{\mathcal{C}^{*}_{min}\ln n} + \sqrt{s})$.

Algorithm RCM performs well for input instances where the optimum is relatively large, but its performance bound can be improved further for instances where the optimum is small compared to s. In Section 4.2, we present another algorithm called RCM2 that, with probability at least $\frac{1}{2}$, computes a solution with objective value at least $\mathcal{C}^{*}_{min} - O(\sqrt{\mathcal{C}^{*}_{min}\ln n})$. (Although the asymptotic approximation bound of RCM is not as good as that of RCM2, we include RCM in the paper because, according to our experiments discussed in Section 5, it outperforms RCM2 in practice.)

We also study problems $\mathsf{BCP}\_\mathcal{D}_{max}$ and $\mathsf{BCP}\_\mathcal{C}_{avg}$, for which we develop some polynomial-time randomized rounding algorithms (RDM for $\mathsf{BCP}\_\mathcal{D}_{max}$, and two algorithms RCA and RCA2 for $\mathsf{BCP}\_\mathcal{C}_{avg}$.) These results are summarized in Table 1.

In Section 5, we present the results of our experimental studies, where we tested algorithms RCM, RCM2 and RDM on both synthetic and real data sets. According to this study, solutions found by these algorithms are very close to the optimal solution of their corresponding linear program, especially on real data sets. For example, in 92.8% of our real data sets, RCM found the solution with value at least 97% of the solution from the linear program.

Algorithm RCM has been implemented and is publicly available at the OFRG website as part of the CloneTools software package, see `http://algorithms.cs.ucr.edu/OFRG/`.

Relation to other work. We are not aware of any other work on the Balanced Covering problem studied in this paper.

Note that OFRG differs from other array-based analysis approaches that, typically, involve a single microarray experiment where one clone of interest is hybridized against a collection of arrayed probes, each targeting a specific sequence. These experiments include control clones as well, but these control clones are used to test whether they bind as predicted to particular microarray probes (see [9,14], for example). In contrast, OFRG uses a small set of probes (roughly 30-50) to coordinately distinguish a much larger set of sequences (for example, all bacterial rRNA genes). Each probe is used in one hybridization experiment, and the unknown DNA clone sequences are immobilized on the array.

2. NP-Completeness

We first show that all four versions of Balanced Covering studied in this paper are $\mathbb{NP}$-hard. In fact, we give two proofs of $\mathbb{NP}$-hardness, as each will lead to different results on hardness of approximation in the next section.

Given a bipartite graph $G = (C, P, E)$ and an even integer s, define a *perfectly balanced cover* in G to be a subset $D \subseteq C$ with $|D| = s$ such that $\deg_D(p) = s/2$ for each $p \in P$. Similarly, we define a *size-s cover* to be a subset $D \subseteq C$ with $|D| = s$ such that $1 \le \deg_D(p) \le s - 1$ for each $p \in P$.

Theorem 1. *The following decision problem is $\mathbb{NP}$-complete: "Given a bipartite graph $G = (C, P, E)$ and an even integer s, is there a perfectly balanced cover in G?" Consequently, $\mathsf{BCP\_C}_{min}$, $\mathsf{BCP\_C}_{avg}$, $\mathsf{BCP\_D}_{max}$ and $\mathsf{BCP\_D}_{avg}$ are $\mathbb{NP}$-hard.*

Proof. The proof is by a polynomial-time reduction from X3C (Exact Cover by 3-Sets), which is known to be $\mathbb{NP}$-complete (see [6], for example). The instance of X3C consists of a finite set X of $3m$ items, and a collection T of n 3-element subsets of X that we refer to as *triples*. We assume that $n \ge m \ge 2$. The objective is to determine whether T contains an *exact cover of X*, that is a sub-collection $T' \subseteq T$ such that every element of X occurs in exactly one triple in T'.

The reduction is defined as follows. Given an instance $\langle X, T\rangle$ of X3C above, we construct an instance $\langle G = (T \cup W, X, E), s\rangle$ of Balanced Covering, where W is a set that contains $m - 2$ new vertices. For $t \in T$ and $x \in X$, we create an edge $(t, x) \in E$ if $x \in t$. Further, we create all edges $(w, x) \in E$ for $x \in X$ and $w \in W$. This defines the bipartite graph G. We let $s = 2m - 2$.

It remains to show that this construction is correct, namely that $\langle X, T\rangle$ has an exact cover iff $\langle G, s\rangle$ has a perfectly balanced cover.

($\Rightarrow$) If $\langle X, T\rangle$ has an exact cover T', we claim that $D = T' \cup W$ is a perfectly balanced cover for $\langle G, s\rangle$. To justify this, note first that $|T'| = m$ and $|W| = m - 2$, and thus $|D| = 2m - 2 = s$. Further, each vertex $x \in X$ has exactly one neighbor in T' and $m - 2$ neighbors in W, so x has $m - 1 = s/2$ neighbors in D, as required.

($\Leftarrow$) Suppose now that $\langle G, s\rangle$ has a perfectly balanced cover $D \subseteq T \cup W$. Denote $W' = D \cap W$, $k = |W'|$, and $T' = D \cap T$. We claim that T' is an exact cover of X.

We first show that D must contain all vertices in W. We count the edges between D and X. There are $3km$ edges between W' and X, since each vertex in W' is connected to all $3m$ vertices in X. There are $3(s-k)$ edges between T' and X, since each vertex in T has degree 3. On the other hand, there must be $3m(m-1)$ edges between X and D, since each vertex in X must be connected to exactly $s/2 = m-1$ vertices in D. Together, this yields $3(2m-2-k) + 3km = 3m(m-1)$. Solving this equation, we get $k = m-2$, which means that $W' = W$.

Since $W' = W$, T' must contain exactly $s - k = m$ vertices. Each vertex $x \in X$ is adjacent to all vertices in W, so it has exactly $s/2 - (m-2) = 1$ neighbor in T'. This means that T' is an exact cover of X, as claimed. □

Next we prove that it is $\mathbb{NP}$-complete to decide whether there is a size-s cover, where each probe in P is covered by at least one but not all clones from the cover.

Theorem 2. *The following decision problem is $\mathbb{NP}$-complete: "Given a bipartite graph $G = (C, P, E)$ and an integer s, is there a size-s cover in G?"*

Proof. The proof is by a polynomial-time reduction from the $\mathbb{NP}$-complete problem Set Cover (see [6]). Given an instance of Set Cover $\langle Q, X, b\rangle$, where Q is a collection of subsets over universe X, the query is whether there is a set cover of size b for X, that is a sub-collection $Q' \subseteq Q$ with $|Q'| = b$ such that $\bigcup Q' = X$.

The reduction is defined as follows. Given an instance $\langle Q, X, b\rangle$ of Set Cover, we construct an instance $\langle G = (Q \cup \{q_0\}, X \cup \{x_0\}, E), s\rangle$, where q_0 and x_0 are two new vertices. For $q \in Q$ and $x \in X$, we create an edge $(q, x) \in E$ if $x \in q$. We also create all edges $(q, x_0) \in E$ for $q \in Q$. This defines the bipartite graph G. We let $s = b + 1$.

We now justify the correctness of the construction by showing that $\langle G, s\rangle$ has a size-s cover iff $\langle Q, X, b\rangle$ has a set cover with size b.

($\Rightarrow$) If $\langle Q, X, b\rangle$ has a set cover Q' of size b, it is clear that $D = Q' \cup \{q_0\}$ is a size-s cover for $\langle G, s\rangle$ since each vertex in $X \cup \{x_0\}$ is adjacent to at least one element from Q', and not adjacent to $q_0 \in D$.

($\Leftarrow$) Suppose now that $\langle G, s\rangle$ has a size-s cover D. We denote $Q' = D \cap Q$. Every $x \in X$ must be adjacent to at least one vertex in Q' since there is no x adjacent to q_0. Thus Q' is a set cover of X of size b. □

3. Hardness of Approximation

Approximation of BCP_$\mathcal{D}_{avg}$ and BCP_$\mathcal{C}_{avg}$. Now we prove that approximating BCP_$\mathcal{D}_{avg}$ and BCP_$\mathcal{C}_{avg}$ is hard. Theorem 1 immediately implies that BCP_$\mathcal{D}_{avg}$ (as well as BCP_$\mathcal{D}_{max}$) cannot be efficiently approximated with any finite ratio. We show that even if we allow an additive term in the approximation bound and randomization, achieving finite ratio for BCP_$\mathcal{D}_{avg}$ is still $\mathbb{NP}$-hard. For BCP_$\mathcal{C}_{avg}$ we show

that it is hard to be approximated with the bound $\mathcal{C}^*_{avg} - \beta\frac{1}{n}(nm)^{1-\epsilon}$, where $\beta > 0$, $0 < \epsilon < 1$ and $m = |C|$, $n = |P|$.

Let $\langle G, s\rangle$ be an instance of Balanced Covering. Given an algorithm $\mathcal{A}$ for $\mathsf{BCP}\_\mathcal{D}_{avg}$, recall that by $\mathcal{D}^{\mathcal{A}}_{avg}(G, s)$ we denote the value of the objective function computed by $\mathcal{A}$, that is

$$\mathcal{D}^{\mathcal{A}}_{avg}(G, s) \;=\; \tfrac{1}{n}\sum_{p\in P} |\deg_D(p) - s/2|,$$

where $D \subseteq C$ is the set computed by $\mathcal{A}$. Similarly, given an algorithm $\mathcal{A}$ for $\mathsf{BCP}\_\mathcal{C}_{avg}$, $\mathcal{C}^{\mathcal{A}}_{avg}(G, s)$ is the value of the objective function computed by $\mathcal{A}$ for $\mathsf{BCP}\_\mathcal{C}_{avg}$, that is

$$\mathcal{C}^{\mathcal{A}}_{avg}(G, s) \;=\; \tfrac{1}{n}\sum_{p\in P} \min\{\deg_D(p), s - \deg_D(p)\}.$$

Recall that by $\mathcal{D}^*_{avg}(G, s)$ and $\mathcal{C}^*_{avg}(G, s)$ we denote the optimal value for $\mathsf{BCP}\_\mathcal{D}_{avg}$ and $\mathsf{BCP}\_\mathcal{C}_{avg}$, respectively.

Recall that the class $\mathbb{RP}$ (randomized polynomial time) is the complexity class of decision problems $\mathcal{P}$ which have polynomial-time probabilistic Turing machines M such that, for each input I, (i) if $I \in \mathcal{P}$ then M accepts I with probability at least $\frac{1}{2}$, and (ii) if $I \notin \mathcal{P}$ then M rejects I with probability 1. It is still open whether $\mathbb{RP} = \mathbb{NP}$.

Theorem 3. *Let $\alpha, \beta > 0$ and $0 < \epsilon < 1$ be any constants. If $\mathbb{RP} \neq \mathbb{NP}$ then there is no randomized polynomial-time algorithm $\mathcal{A}$ that*

(a) for any instance $\langle G, s\rangle$ of $\mathsf{BCP}\_\mathcal{D}_{avg}$ satisfies

$$Exp[\mathcal{D}^{\mathcal{A}}_{avg}(G, s)] \;\le\; \alpha \cdot \mathcal{D}^*_{avg}(G, s) + \beta\tfrac{1}{n}(nm)^{1-\epsilon}, \quad \textit{or} \tag{3}$$

(b) for any instance $\langle G, s\rangle$ of $\mathsf{BCP}\_\mathcal{C}_{avg}$ satisfies

$$Exp[\mathcal{C}^{\mathcal{A}}_{avg}(G, s)] \;\ge\; \mathcal{C}^*_{avg}(G, s) - \beta\tfrac{1}{n}(nm)^{1-\epsilon}. \tag{4}$$

Proof. We first prove part (a) of the theorem. Suppose, towards contradiction, that for some α, β and ϵ there exists a randomized polynomial-time algorithm $\mathcal{A}$ that satisfies (3). We show that this would imply the existence of a randomized polynomial-time algorithm that decides if there is a perfectly balanced covering, contradicting Theorem 1.

Given an instance $\langle G, s\rangle$ of $\mathsf{BCP}\_\mathcal{D}_{avg}$, where $G = (C, P, E)$, convert it into another instance $\langle G^r, s\rangle$ of $\mathsf{BCP}\_\mathcal{D}_{avg}$, where $G^r = (C, P', E')$ is obtained by creating r copies of each probe $p \in P$ (that is, with the same neighbors in C). Thus $|P'| = rn$. We choose $r = \lceil (2\beta m^{1-\epsilon}n^{1-\epsilon})^{\frac{1}{\epsilon}} \rceil + 1$. For this r, we have $2\beta m^{1-\epsilon}(nr)^{-\epsilon} < \frac{1}{n}$. Therefore the new instance $\langle G^r, s\rangle$ has the following properties:

- If $\langle G, s\rangle$ has a perfectly balanced cover (that is, $\mathcal{D}^*_{avg}(G, s) = 0$) then $\mathcal{D}^*_{avg}(G^r, s) = 0$, and therefore $2 \cdot \mathrm{Exp}[\mathcal{D}^{\mathcal{A}}_{avg}(G^r, s)] \le 2\beta m^{1-\epsilon}(nr)^{-\epsilon} < \frac{1}{n}$. Using Markov's inequality, this implies that $\Pr[\mathcal{D}^{\mathcal{A}}_{avg}(G^r, s) < \frac{1}{n}] \ge \frac{1}{2}$.

- if $\langle G, s\rangle$ does not have a perfectly balanced cover (that is, $\mathcal{D}^*_{avg}(G,s) \geq \frac{1}{n}$) then $\mathcal{D}^{\mathcal{A}}_{avg}(G^r, s) \geq \mathcal{D}^*_{avg}(G^r, s) = \mathcal{D}^*_{avg}(G, s) \geq \frac{1}{n}$, with probability 1.

Since G^r can be computed from G in polynomial time, from $\mathcal{A}$ we could obtain a randomized polynomial-time algorithm that determines the existence of a perfectly balanced cover – a problem that is $\mathbb{NP}$-complete, according to Theorem 1. The part (a) of the theorem follows.

Part (b) follows directly from part (a) of the theorem and the fact that $\mathcal{C}^*_{avg}(G,s) = s/2 - \mathcal{D}^*_{avg}(G,s)$ and $\mathcal{C}_{avg}(H) = s/2 - \mathcal{D}_{avg}(H)$ for any solution H for instance $\langle G, s\rangle$ of $\mathsf{BCP}\_\mathcal{C}_{avg}$ and $\mathsf{BCP}\_\mathcal{D}_{avg}$. □

Using an argument very similar to the proof of Theorem 3, one can show that, unless $\mathbb{P} = \mathbb{NP}$, there is no deterministic polynomial-time algorithm that satisfies bounds analogous to those in Theorem 3.

Approximation of $\mathsf{BCP}\_\mathcal{C}_{min}$ and $\mathsf{BCP}\_\mathcal{D}_{max}$. Next we show that $\mathsf{BCP}\_\mathcal{C}_{min}$ cannot be approximated efficiently with the objective value at least $\epsilon\mathcal{C}^*_{min}(G,s)$ or $\mathcal{C}^*_{min}(G,s) - O(\ln n)$, unless $\mathbb{NP}$ has slightly superpolynomial time algorithms. As a result, $\mathsf{BCP}\_\mathcal{D}_{max}$ cannot be approximated efficiently with the objective value at most $\mathcal{D}^*_{max}(G,s) + O(\ln n)$. Recall that for a given instance $\langle G, s\rangle$, we denote by $\mathcal{C}^*_{min}(G,s)$ and $\mathcal{D}^*_{max}(G,s)$ the optimal value of $\mathcal{C}_{min}(G,s)$ and $\mathcal{D}_{max}(G,s)$, respectively. Similarly, $\mathcal{C}^{\mathcal{A}}_{min}(G,s)$ and $\mathcal{D}^{\mathcal{A}}_{max}(G,s)$ are the values of the objective function computed by an algorithm $\mathcal{A}$ for $\mathsf{BCP}\_\mathcal{C}_{min}$ or $\mathsf{BCP}\_\mathcal{D}_{max}$, respectively, on an instance $\langle G, s\rangle$.

Theorem 4. *Unless* $\mathbb{NP} \subseteq \mathbb{DTIME}(n^{O(\log\log n)})$*, then*

(a) there is no polynomial-time algorithm $\mathcal{A}$ *for* $\mathsf{BCP}\_\mathcal{C}_{min}$ *that, for some* $0 < \epsilon < 1$*, for any instance* $\langle G, s\rangle$*, satisfies*

$$\mathcal{C}^{\mathcal{A}}_{min}(G,s) \geq \epsilon\,\mathcal{C}^*_{min}(G,s), \quad \textit{and} \tag{5}$$

(b) there is no polynomial-time algorithm $\mathcal{A}$ *for* $\mathsf{BCP}\_\mathcal{C}_{min}$ *that, for some* $0 < \epsilon < 1$*, for any instance* $\langle G, s\rangle$*, satisfies*

$$\mathcal{C}^{\mathcal{A}}_{min}(G,s) \geq \mathcal{C}^*_{min}(G,s) - (1-\epsilon)\ln n. \tag{6}$$

Proof. We first prove part (a) of the theorem. Suppose, towards contradiction, that there exists a polynomial-time algorithm $\mathcal{A}$ that satisfies (5). We show that this would imply the existence of a polynomial-time $((1-\Omega(\epsilon))\ln n)$-approximation algorithm $\mathcal{B}_1$ for the Set Cover problem, which would imply in turn that problems in $\mathbb{NP}$ have $n^{O(\log\log n)}$-time deterministic algorithms [4].

Algorithm $\mathcal{B}_1$ works as follows. Given an instance $\langle Q, X\rangle$ of Set Cover, where $|X| = n$ and Q is a collection of sets over X, the algorithm $\mathcal{B}_1$ first reduces $\langle Q, X\rangle$ to an instance $\langle G = (T \cup W, P, E), s\rangle$ of $\mathsf{BCP}\_\mathcal{C}_{min}$, where $P = X \cup \{x_0\}$, T contains $k = \lfloor \frac{\ln n}{2} \rfloor$ vertices $q_1, q_2, ..., q_k$ for each set $q \in Q$, and W is a set containing k new vertices. For each $q \in Q$ and $i = 1, 2, ..., k$, we create an edge $(q_i, x_0) \in E$

and edges $(q_i, x) \in E$ for each $x \in q$. This defines the bipartite graph G. Let b represent the size of the minimum set cover of X. We now assume that, without loss of generality, algorithm $\mathcal{B}_1$ knows the value of b. Otherwise, $\mathcal{B}_1$ can simply try each $b \in \{1, 2, ..., n\}$, and choose the smallest set cover. We now let $s = kb + k$.

Next $\mathcal{B}_1$ calls algorithm $\mathcal{A}$ on input $\langle G, s\rangle$ to get a balanced cover H for $\langle G, s\rangle$, and outputs the collection of sets $H' = \{q : (\exists i) q_i \in H\}$ as a set cover of X.

To prove that $\mathcal{B}_1$ is a $((1 - \Omega(\epsilon)) \ln n)$-approximation algorithm for Set Cover, we now show that H' is a set cover of X and $|H'| \leq (1 - \frac{\epsilon}{2}) \ln(n) b$.

Assuming that $\langle Q, X\rangle$ has a set cover Q' of size b, we first claim that $\mathcal{C}^*_{min}(G, s) \geq k$. To justify this, from Q', we build the balanced cover $D = \{q_i : q \in Q'\} \cup W$. Obviously, $|D| = kb + k = s$. For each $x \in X$, the k copies of Q' ensure that $\deg_D(x) \geq k$, while the k vertices in W ensure that $\deg_D(x) \leq s - k$. Our claim implies that algorithm $\mathcal{A}$ on input $\langle G, s\rangle$ will find a balanced cover H with objective function value at least $\epsilon \mathcal{C}^*_{min}(G, s) \geq \epsilon k$. We have $|H \cap W| \geq \epsilon k$, because $x_0 \in P$ is adjacent to every vertex in T and H has at least ϵk vertices not adjacent to x_0. Therefore $|H \cap T| \leq s - \epsilon k = kb + (1 - \epsilon)k$. Thus, since each $x \in P$ is adjacent to at least one vertex in H (in fact, at least ϵk), H' forms a set cover of X of size at most $kb + (1 - \epsilon)k \leq (2 - \epsilon)kb \leq (1 - \frac{\epsilon}{2}) \ln(n) b$, as claimed.

The algorithm $\mathcal{B}_1$ clearly runs in polynomial time, and is a $((1 - \Omega(\epsilon)) \ln n)$-approximation algorithm for the Set Cover problem. Thus the part (a) of the theorem follows.

Next we prove the part (b) of the theorem. Suppose, towards contradiction, that there exists a polynomial-time algorithm $\mathcal{A}$ that satisfies (6). As in part (a), we will prove that this would imply the existence of a polynomial-time $((1 - \Omega(\epsilon)) \ln n)$-approximation algorithm $\mathcal{B}_2$ for the Set Cover problem.

$\mathcal{B}_2$ works like algorithm $\mathcal{B}_1$ described previously except we let $k = \lceil (1 - \epsilon) \ln n + 1 \rceil$ this time.

Assuming that $\langle Q, X\rangle$ has a set cover Q' of size b, an argument similar to the proof of part (a) shows that algorithm $\mathcal{A}$ on input $\langle G, s\rangle$ will find a balanced cover H with objective function value at least

$$\mathcal{C}^*_{min}(G, s) - (1 - \epsilon) \ln n \geq k - (1 - \epsilon) \ln n \geq 1.$$

Also, H' forms a set cover of X of size at most s. We now assume that, without loss of generality, $b \geq \frac{2}{\epsilon}$, because otherwise the Set Cover problem can be solved in polynomial time $O(|X|^2 \cdot |Q|^{\frac{2}{\epsilon}})$. Thus we get $|H'| \leq kb + k \leq kb + \frac{\epsilon}{2} kb \leq (1 + \frac{\epsilon}{2})(1 - \epsilon)(\ln(n) + \frac{2}{1-\epsilon}) b \leq (1 - \Omega(\epsilon)) \ln(n) b$, as claimed.

The algorithm $\mathcal{B}_2$ clearly runs in polynomial time, and is a $((1 - \Omega(\epsilon)) \ln n)$-approximation algorithm for the Set Cover problem. Thus the theorem follows. □

As a corollary, we also get an approximation hardness result for $\mathsf{BCP\_}\mathcal{D}_{max}$.

Corollary 1. *Unless* $\mathbb{NP} \subseteq \mathbb{DTIME}(n^{O(\log \log n)})$*, there is no polynomial-time algorithm* $\mathcal{A}$ *for* $\mathsf{BCP\_}\mathcal{D}_{max}$ *that, for some* $0 < \epsilon < 1$ *and for any instance* $\langle G, s\rangle$*,*

satisfies

$$\mathcal{D}^{\mathcal{A}}_{max}(G,s) \leq \mathcal{D}^{*}_{max}(G,s) + (1-\epsilon)\ln n. \tag{7}$$

Proof. The corollary follows directly from part (b) of Theorem 4 and the fact that $\mathcal{D}^{*}_{max}(G,s) = s/2 - \mathcal{C}^{*}_{min}(G,s)$ and $\mathcal{D}_{max}(H) = s/2 - \mathcal{C}_{min}(H)$ for any solution H for instance $\langle G, s\rangle$ of $\mathsf{BCP}\_\mathcal{D}_{max}$ and $\mathsf{BCP}\_\mathcal{C}_{min}$. □

4. Approximation Algorithms and Analysis

In this section we present several randomized algorithms for different versions of Balanced Covering. We give two algorithms RCM and RCM2 for $\mathsf{BCP}\_\mathcal{C}_{min}$, algorithm RDM for $\mathsf{BCP}\_\mathcal{D}_{max}$, and two algorithms RCA and RCA2 for $\mathsf{BCP}\_\mathcal{C}_{avg}$.

All algorithms are based on randomized rounding. We first solve a linear relaxation LP of the integer program ILP for Balanced Covering, and then use the fractional solution as probabilities to randomly choose the integral solutions.

Let $x^*_1, ..., x^*_n$, where $0 \leq x^*_i \leq 1$ for each i, be the optimum solution of LP and z^* the corresponding optimum value of the objective function. We choose $X_i = 1$ with probability x^*_i and 0 otherwise, which gives us a "provisional" integral solution $X_1, ..., X_n$ with objective value Z. Since the expectation of Z is equal to z^* and the random variables X_i are independent, we can apply the Chernoff bound to show that with high probability the value of Z is close to z^* (and thus also approximates well the optimum of ILP). If Z is not feasible, we adjust the values of a sufficient number L of the variables X_i obtaining a final feasible solution whose value $\tilde{Z}$ differs from Z by at most L. Applying the Chernoff bound again, we get an estimate on L, and combining it with the bound on Z we obtain a bound on $\tilde{Z}$.

For some objective functions we refine this approach further, by adjusting the probability of setting X_i to 1, in order to reduce the violation L of the constraints. This modification improves asymptotic performance bounds but – as we show later in Section 5 – it tends to degrade the experimental performance on both random and real data sets.

4.1. *Algorithm RCM for* $\mathsf{BCP}\_\mathcal{C}_{min}$

Given $G = (C, P, E)$, let $C = \{c_1, c_2, ..., c_m\}$, $P = \{p_1, p_2, ..., p_n\}$. And let $A = [a_{ij}]$ be the Boolean $m \times n$ adjacency matrix of G, that is $a_{ij} = 1$ iff $(c_i, p_j) \in E$; otherwise $a_{ij} = 0$. Then $\mathsf{BCP}\_\mathcal{C}_{min}$ is equivalent to the following integer linear program MinIP:

$$\begin{aligned}
\text{maximize:} \quad & z \\
\text{subject to:} \quad & z \leq \textstyle\sum_{i=1}^{m} a_{ij} x_i \quad \forall j = 1, ..., n \\
& z \leq \textstyle\sum_{i=1}^{m} (1 - a_{ij}) x_i \quad \forall j = 1, ..., n \\
& \textstyle\sum_{i=1}^{m} x_i \leq s \\
& x_i \in \{0, 1\} \quad \forall i = 1, ..., m
\end{aligned}$$

The Boolean variables x_i indicate whether the corresponding $c_i \in C$ are selected or not.

Algorithm RCM. The algorithm first relaxes the last constraint to $0 \le x_i \le 1$ to obtain the linear program MinLP, and then computes an optimal solution x_i^*, $i = 1, 2, ..., m$, of MinLP. Next, applying randomized rounding, RCM computes an integral solution $X_1, ..., X_m$ by choosing $X_i = 1$ with probability x_i^* and 0 otherwise. Note that this solution may not be feasible since $\sum_{i=1}^m X_i$ may exceed s. Let $L = \max\{\sum_{i=1}^m X_i - s, 0\}$. RCM changes L arbitrary variables $X_i = 1$ to 0, obtaining a feasible solution $\tilde{X}_1, ..., \tilde{X}_m$.

Analysis. We denote by $\mathcal{C}_{min}^{\text{RCM}}(G, s)$ or $\tilde{Z}$ the value of the objective function computed by RCM on input $\langle G, s\rangle$, that is $\mathcal{C}_{min}^{\text{RCM}}(G, s) = \tilde{Z} = \min_{j=1}^n\{\sum_{i=1}^m a_{ij}\tilde{X}_i, \sum_{i=1}^m (1 - a_{ij})\tilde{X}_i\}$.

Lemma 1. *For any instance $\langle G, s\rangle$ of BCP_$\mathcal{C}_{min}$, with probability at least $\frac{1}{2}$,*

$$\mathcal{C}_{min}^{RCM}(G, s) \;\ge\; \mathcal{C}_{min}^*(G, s) - O\left(\sqrt{\mathcal{C}_{min}^*(G, s)\ln n} + \sqrt{s}\right). \tag{8}$$

Proof. Let $z^* = \min_{j=1}^n\{\sum_{i=1}^m a_{ij}x_i^*, \sum_{i=1}^m (1-a_{ij})x_i^*\}$ be the optimum solution of MinLP. Let also $Z = \min_{j=1}^n\{\sum_{i=1}^m a_{ij}X_i, \sum_{i=1}^m (1 - a_{ij})X_i\}$.

The $\{X_i\}$ are independent Bernoulli random variables with $\text{Exp}[X_i] = x_i^*$. So, for each j, $\text{Exp}[\sum_{i=1}^m a_{ij}X_i] = \sum_{i=1}^m a_{ij}x_i^* \ge z^*$, By a standard Chernoff bound, we get

$$\Pr[\textstyle\sum_{i=1}^m a_{ij}X_i \le (1 - \lambda)z^*] \;\le\; e^{-\lambda^2 z^*/2},$$

where $0 < \lambda \le 1$. Similarly, for all j,

$$\Pr[\textstyle\sum_{i=1}^m (1 - a_{ij})X_i \le (1 - \lambda)z^*] \;\le\; e^{-\lambda^2 z^*/2}.$$

By the naive union bound, the probability that any of the $2n$ above events happens is at most $2ne^{-\lambda^2 z^*/2}$. Hence we have

$$\Pr[Z \le (1 - \lambda)z^*] \;\le\; 2ne^{-\lambda^2 z^*/2}. \tag{9}$$

Likewise, $\text{Exp}[\sum_{i=1}^m X_i] = \sum_{i=1}^m x_i^* \le s$. Thus by the Chernoff bound, $\Pr[\sum_{i=1}^m X_i \ge (1 + \epsilon)s] \le e^{-\epsilon^2 s/4}$, where $0 < \epsilon \le 2e - 1$. Recalling $L = \max\{\sum_{i=1}^m X_i - s, 0\}$, we have

$$\Pr[L \ge \delta\sqrt{s}] \;\le\; e^{-\delta^2/4}, \tag{10}$$

where $0 < \delta \le (2e - 1)\sqrt{s}$.

Since $\tilde{Z} \ge Z - L$, we get $\Pr[\tilde{Z} \le (1 - \lambda)z^* - \delta\sqrt{s}] \le \Pr[Z \le (1 - \lambda)z^*] + \Pr[L \ge \delta\sqrt{s}]$. Combining this with (9) and (10), we have

$$\Pr[\tilde{Z} \le (1 - \lambda)z^* - \delta\sqrt{s}] \;\le\; 2ne^{-\lambda^2 z^*/2} + e^{-\delta^2/4}. \tag{11}$$

Suppose $\mathcal{C}_{min}^*(G, s) \ge 2\ln(8n)$. Then $z^* \ge 2\ln(8n)$ as well, because $\mathcal{C}_{min}^*(G, s) \le z^*$. Choosing $\lambda = \sqrt{2\ln(8n)/z^*}$ and $\delta = \sqrt{4\ln 4}$, from (11), we get

$$\Pr[\tilde{Z} \le z^* - \sqrt{2\ln(8n)z^*} - \sqrt{4\ln(4)s}] \;\le\; \tfrac{1}{2}.$$

Since $z^* \geq \mathcal{C}^*_{min}(G,s) \geq 2\ln(8n)$, with probability at least $\frac{1}{2}$, we have

$$\tilde{Z} \geq \mathcal{C}^*_{min}(G,s) - \sqrt{2\ln(8n)\mathcal{C}^*_{min}(G,s)} - \sqrt{4\ln(4)s}. \tag{12}$$

Inequality (12) is also trivially true for $\mathcal{C}^*_{min}(G,s) < 2\ln(8n)$. Thus the lemma follows. □

4.2. *An Alternative Algorithm RCM2 for* BCP_$\mathcal{C}_{min}$

The performance bound for RCM given in Section 4.1 can be improved for instances where the optimum is small compared to s. We now provide an alternative algorithm RCM2, which is identical to RCM in all steps except for the rounding scheme: choose $X_i = 1$ with probability $(1-\epsilon)x^*_i$, and 0 otherwise, where $\epsilon = \min\left\{2\sqrt{\ln(4n+2)/z^*}, 1\right\}$.

Analysis. All notations are defined similarly to those in Section 4.1.

Lemma 2. *For any instance $\langle G,s\rangle$ of BCP_$\mathcal{C}_{min}$, with probability at least $\frac{1}{2}$,*

$$\mathcal{C}^{RCM2}_{min}(G,s) \geq \mathcal{C}^*_{min}(G,s) - O\left(\sqrt{\mathcal{C}^*_{min}(G,s)\ln n}\right). \tag{13}$$

Proof. The $\{X_i\}$ are independent random variables with $\mathrm{Exp}[X_i] = (1-\epsilon)x^*_i$. By linearity of expectation, $\mathrm{Exp}[\sum_{i=1}^m X_i] \leq \sum_{i=1}^m (1-\epsilon)x^*_i \leq (1-\epsilon)s$. Thus, by the Chernoff bound,

$$\Pr[\textstyle\sum_{i=1}^m X_i \geq s] \leq \Pr[\sum_{i=1}^m X_i \geq (1+\epsilon)(1-\epsilon)s] \leq e^{-\epsilon^2(1-\epsilon)s/4}.$$

As $z^* \leq s/2$, we have $s/4 \geq z^*/2$. The above bound implies

$$\Pr[\textstyle\sum_{i=1}^m X_i \geq s] \leq e^{-\epsilon^2(1-\epsilon)z^*/2}. \tag{14}$$

Likewise, for each j, $\sum_{i=1}^m a_{ij}x^*_i \geq z^*$, so $\mathrm{Exp}[\sum_{i=1}^m a_{ij}X_i] \geq (1-\epsilon)z^*$. By the Chernoff bound,

$$\Pr[\textstyle\sum_{i=1}^m a_{ij}X_i \leq (1-\epsilon)^2 z^*] \leq e^{-\epsilon^2(1-\epsilon)z^*/2}. \tag{15}$$

Similarly, for all j,

$$\Pr[\textstyle\sum_{i=1}^m (1-a_{ij})X_i \leq (1-\epsilon)^2 z^*] \leq e^{-\epsilon^2(1-\epsilon)z^*/2}. \tag{16}$$

Letting $L = \max\{\sum_{i=1}^m X_i - s, 0\}$, since $\tilde{Z} \geq Z - L$, we get $\Pr[\tilde{Z} \leq (1-\epsilon)^2 z^* - L] \leq \Pr[Z \leq (1-\epsilon)^2 z^*] + \Pr[\sum_{i=1}^m X_i \geq s]$. Combining this with (14), (15) and (16), we have

$$\Pr[\tilde{Z} \leq (1-\epsilon)^2 z^*] \leq (2n+1)e^{-\epsilon^2(1-\epsilon)z^*/2}.$$

Since $(1-\epsilon)^2 \geq 1-2\epsilon$, for $\epsilon < \frac{1}{2}$, we get

$$\Pr[\tilde{Z} \leq z^* - 4\sqrt{\ln(4n+2)z^*}] \leq \tfrac{1}{2}. \tag{17}$$

The above bound is also trivially true for $\epsilon \geq \frac{1}{2}$ (that is, $z^* \leq 16\ln(4n+2)$). Finally, suppose $\mathcal{C}^*_{min}(G,s) \geq 16\ln(4n+2)$. Since also $\mathcal{C}^*_{min}(G,s) \leq z^*$, inequality (17) implies that with probability at least $\frac{1}{2}$,

$$\tilde{Z} \geq \mathcal{C}^*_{min}(G,s) - 4\sqrt{\ln(4n+2)\mathcal{C}^*_{min}(G,s)}, \tag{18}$$

Inequality (18) is also trivially true for $\mathcal{C}^*_{min}(G,s) \leq 16\ln(4n+2)$. Thus the lemma follows. □

We will show later in Section 5 that RCM2 does not outperform RCM in experimental analysis. Therefore RCM cannot be completely substituted by RCM2.

4.3. *Algorithm RDM for* BCP_$\mathcal{D}_{max}$

In this section we present our randomized algorithm RDM for BCP_$\mathcal{D}_{max}$. Given $G = (C, P, E)$, let A be the Boolean $m \times n$ adjacency matrix of G, as in Section 4.1. Then BCP_$\mathcal{D}_{max}$ is equivalent to the following integer linear program MaxIP:

$$\begin{array}{rrcl} \text{minimize:} & z & & \\ \text{subject to:} & z & \geq & \sum_{i=1}^m a_{ij}x_i - s/2 \quad \forall j = 1,...,n \\ & z & \geq & s/2 - \sum_{i=1}^m a_{ij}x_i \quad \forall j = 1,...,n \\ & \sum_{i=1}^m x_i & = & s \\ & x_i & \in & \{0,1\} \quad \forall i = 1,...,m \end{array}$$

The Boolean variables x_i indicate whether the corresponding $c_i \in C$ are selected or not.

Algorithm RDM. The algorithm first relaxes the last constraint to $0 \leq x_i \leq 1$ to obtain the linear program MaxLP, and then computes an optimal solution x^*_i, $i = 1, 2, ..., m$, of MaxLP. Next, applying randomized rounding like in RCM, RDM computes an integral solution $X_1, ..., X_m$ by choosing $X_i = 1$ with probability x^*_i and 0 otherwise. Note that this solution may not be feasible since $\sum_{i=1}^m X_i$ may not be exactly s. Let $L = \sum_{i=1}^m X_i - s$. RDM changes $|L|$ arbitrary variables $X_i = 1$ to 0 if $L > 0$, and does the contrary if $L < 0$, obtaining a feasible solution $\tilde{X}_1, ..., \tilde{X}_m$.

Analysis. We denote by z^* and $\mathcal{D}^{\text{RDM}}_{max}(G,s)$ (or $\tilde{Z}$) the value of the objective function computed by MaxLP and RDM for BCP_$\mathcal{D}_{max}$, respectively. Namely, $z^* = \max_{j=1}^n |\sum_{i=1}^m a_{ij}x^*_i - s/2|$, and $\mathcal{D}^{\text{RDM}}_{max}(G,s) = \tilde{Z} = \max_{j=1}^n |\sum_{i=1}^m a_{ij}\tilde{X}_i - s/2|$.

Lemma 3. *For any instance $\langle G, s\rangle$ of BCP_$\mathcal{D}_{max}$, with probability at least $\frac{1}{2}$,*

$$\mathcal{D}^{RDM}_{max}(G,s) \;\leq\; \mathcal{D}^*_{max}(G,s) + O\left(\sqrt{s\ln n}\right). \tag{19}$$

Proof. We now assume that, without loss of generality, $s \geq 9$, because otherwise s is a constant then (19) will be trivially true.

Let $Z = \max_{j=1}^{n} |\sum_{i=1}^{m} a_{ij}X_i - s/2|$. For each j, define $\bar{z}_j = \sum_{i=1}^{m} a_{ij}x_i^*$, and $z_j^* = |\bar{z}_j - s/2|$. Similarly, define random variables $\bar{Z}_j = \sum_{i=1}^{m} a_{ij}X_i$ and $Z_j = |\bar{Z}_j - s/2|$. Thus $z^* = \max_{j=1}^{n} z_j^*$ and $Z = \max_{j=1}^{n} Z_j$.

The $\{X_i\}$ are independent Bernoulli random variables with $\mathrm{Exp}[X_i] = x_i^*$. So $\mathrm{Exp}[\bar{Z}_j] = \bar{z}_j$ for each j. Applying a standard Chernoff bound, we get $\Pr[|\bar{Z}_j - \bar{z}_j| \geq \epsilon\bar{z}_j] \leq 2e^{-\epsilon^2\bar{z}_j/4}$, for $0 < \epsilon \leq 1$. This and the triangle inequality imply

$$\Pr[Z_j \geq z^* + \lambda\sqrt{s}] \;\leq\; \Pr[Z_j \geq z_j^* + \lambda\sqrt{\bar{z}_j}] \;\leq\; \Pr[|\bar{Z}_j - \bar{z}_j| \geq \lambda\sqrt{\bar{z}_j}] \;\leq\; 2e^{-\lambda^2/4}, \quad (20)$$

where $0 < \lambda \leq \sqrt{\bar{z}_j}$. Since $|\bar{z}_j - s/2| \leq z^*$, $\bar{z}_j \geq s/2 - z^*$ for all j. Hence (20) also holds when $0 < \lambda \leq \sqrt{s/2 - z^*}$.

By the naive union bound,

$$\Pr[Z \geq z^* + \lambda\sqrt{s}] \;\leq\; 2ne^{-\lambda^2/4}. \quad (21)$$

Likewise, $\mathrm{Exp}[\sum_{i=1}^{m} X_i] = \sum_{i=1}^{m} x_i^* = s$. By the Chernoff bound, $\Pr[|\sum_{i=1}^{m} X_i - s| \geq \epsilon s] \leq 2e^{-\epsilon^2 s/4}$, for $0 < \epsilon \leq 1$. Thus we have

$$\Pr[|\textstyle\sum_{i=1}^{m} X_i - s| \geq \delta\sqrt{s}] \;\leq\; 2e^{-\delta^2/4}, \quad (22)$$

where $0 < \delta \leq \sqrt{s}$.

Since $\tilde{Z} \leq Z + |\sum_{i=1}^{m} X_i - s|$, we get $\Pr[\tilde{Z} \geq z^* + \lambda\sqrt{s} + \delta\sqrt{s}] \leq \Pr[Z \geq z^* + \lambda\sqrt{s}] + \Pr[|\sum_{i=1}^{m} X_i - s| \geq \delta\sqrt{s}]$. Combining this with (21) and (22), we have

$$\Pr[\tilde{Z} \geq z^* + (\lambda + \delta)\sqrt{s}] \;\leq\; 2ne^{-\lambda^2/4} + 2e^{-\delta^2/4}. \quad (23)$$

Choose $\lambda = \sqrt{4\ln(8n)}$ and $\delta = \sqrt{4\ln 8}$. (Note that $\delta \leq \sqrt{s}$, since $s \geq 9$.) When $\lambda \leq \sqrt{s/2 - z^*}$, from (23), with probability at least $\frac{1}{2}$, we get

$$\tilde{Z} \leq z^* + (\sqrt{4\ln(8n)} + \sqrt{4\ln 8})\sqrt{s}. \quad (24)$$

If $s < 4\sqrt{\ln(8n)}$, then $\sqrt{4s\ln(8n)} > s/2$, inequality (24) will be trivially true. Suppose $s \geq 4\sqrt{\ln(8n)}$ and $z^* > s/2 - \sqrt{4\ln(8n)}$ (i.e., $\lambda > \sqrt{s/2 - z^*}$). Then $z^* + \sqrt{4s\ln(8n)} \geq s/2$, and inequality (24) is also trivially true. Thus by (24) together with the bound $\mathcal{D}_{max}^*(G, s) \geq z^*$, we obtain the lemma. □

4.4. *Algorithm RCA for* BCP_$\mathcal{C}_{avg}$

In this section we present our randomized algorithm RCA for BCP_$\mathcal{C}_{avg}$. Given $G = (C, P, E)$, again let A be the Boolean $m \times n$ adjacency matrix of G. Then BCP_$\mathcal{C}_{avg}$ is equivalent to the following integer linear program AvgIP:

$$\begin{aligned}
\text{maximize:} \quad & \frac{1}{n}\sum_{j=1}^{n} z_j \\
\text{subject to:} \quad & z_j \;\leq\; \textstyle\sum_{i=1}^{m} a_{ij}x_i \quad \forall j = 1, ..., n \\
& z_j \;\leq\; \textstyle\sum_{i=1}^{m} (1 - a_{ij})x_i \quad \forall j = 1, ..., n \\
& \textstyle\sum_{i=1}^{m} x_i \;\leq\; s \\
& x_i \;\in\; \{0, 1\} \quad \forall i = 1, ..., m
\end{aligned}$$

The Boolean variables x_i indicate whether the corresponding $c_i \in C$ are selected or not.

Algorithm RCA. The algorithm first relaxes the last constraint to $0 \le x_i \le 1$ to obtain the linear program AvgLP, and then computes an optimal solution x_i^*, $i = 1, 2, ..., m$, of AvgLP. Next, applying randomized rounding, RCA computes an integral solution $X_1, ..., X_m$ by choosing $X_i = 1$ with probability x_i^* and 0 otherwise. Note that this solution may not be feasible since $\sum_{i=1}^m X_i$ may exceed s. Let $L = \max\{\sum_{i=1}^m X_i - s, 0\}$. RCA changes L arbitrary variables $X_i = 1$ to 0, obtaining a feasible solution $\tilde{X}_1, ..., \tilde{X}_m$.

One can show that, in expectation, for any instance $\langle G, s\rangle$, RCA finds a solution with objective value at least $\mathcal{C}^*_{avg}(G, s) - O(\sqrt{s})$. We omit the proof because in the next section we provide an algorithm with a better asymptotic bound.

4.5. *An Alternative Algorithm RCA2 for* BCP_$\mathcal{C}_{avg}$

We now modify Algorithm RCA, to improve its approximation bound. Let z^* be the optimum solution of AvgLP, that is $z^* = \frac{1}{n}\sum_{j=1}^n \min\{\sum_{i=1}^m a_{ij}x_i^*, \sum_{i=1}^m (1-a_{ij})x_i^*\}$. Our new Algorithm RCA2 is identical to RCA in all steps except for the rounding scheme: choose $X_i = 1$ with probability $\frac{x_i^*}{1+\lambda}$ and 0 otherwise, where $\lambda = \frac{1}{\sqrt{z^*}}$ (without loss of generality, assuming $z^* > 0$).

Before we start RCA2's analysis, we state and prove a variant of the Chernoff bound needed to estimate the error introduced by changing L variables X_i at the end of the algorithm.

Lemma 4. *Let $Y_1, Y_2, ..., Y_n$ be n independent Bernoulli trials, where $\Pr[Y_i = 1] = p_i$. Then if $Y = \sum_{i=1}^n Y_i$ and if $Exp[Y] = \sum_i p_i \le \mu$, for any $0 < \epsilon \le 1$:*

$$Exp[\max\{0, Y - (1+\epsilon)\mu\}] \;\le\; \frac{2e^{-\mu\epsilon^2/4}}{\ln(1+\epsilon)}. \tag{25}$$

Proof. See Appendix Appendix A. □

Analysis. We denote by $\mathcal{C}^{\text{RCA2}}_{avg}(G, s)$ or $\tilde{Z}$ the value of the objective function computed by RCA2 for BCP_$\mathcal{C}_{avg}$, that is $\mathcal{C}^{\text{RCA2}}_{avg}(G, s) = \tilde{Z} = \frac{1}{n}\sum_{j=1}^n \min\{\sum_{i=1}^m a_{ij}\tilde{X}_i, \sum_{i=1}^m (1-a_{ij})\tilde{X}_i\}$. Recall $\mathcal{C}^*_{avg}(G, s)$ is the optimal value of $\mathcal{C}_{avg}(G, s)$ of BCP_$\mathcal{C}_{avg}$.

Lemma 5. *For any instance $\langle G, s\rangle$ of BCP_$\mathcal{C}_{avg}$,*

$$Exp[\mathcal{C}^{RCA2}_{avg}(G, s)] \;\ge\; \mathcal{C}^*_{avg}(G, s) - O\left(\sqrt{\mathcal{C}^*_{avg}(G, s)}\right). \tag{26}$$

Proof. We can assume that $\mathcal{C}^*_{avg}(G, s) \ge 1$, because otherwise (26) is trivially true. Thus $z^* \ge 1$ as well, since $z^* \ge \mathcal{C}^*_{avg}(G, s)$.

For all j define constants $z_j^* = \min\{\sum_{i=1}^m a_{ij}x_i^*, \sum_{i=1}^m (1-a_{ij})x_i^*\}$ and variables $\bar{Z}_j = \sum_{i=1}^m a_{ij}X_i$, $\hat{Z}_j = \sum_{i=1}^m (1-a_{ij})X_i$ and $Z_j = \min\{\bar{Z}_j, \hat{Z}_j\}$. Thus we have $z^* = \frac{1}{n}\sum_{j=1}^n z_j^*$ and $Z = \frac{1}{n}\sum_{j=1}^n Z_j$.

The $\{X_i\}$ are independent Bernoulli random variables with $\mathrm{Exp}[X_i] = \frac{x_i^*}{1+\lambda}$. So $\mathrm{Exp}[\bar{Z}_j] \geq \frac{z_j^*}{1+\lambda}$ and $\mathrm{Exp}[\hat{Z}_j] \geq \frac{z_j^*}{1+\lambda}$, for each j. Applying the Chernoff-Wald bound [13], we get

$$\begin{aligned}&\mathrm{Exp}[(1-\epsilon)\tfrac{z_j^*}{1+\lambda} - (1+\epsilon)Z_j] \;\leq\\ &\mathrm{Exp}\left[\max\left\{(1-\epsilon)\frac{z_j^*}{1+\lambda} - (1+\epsilon)\bar{Z}_j, (1-\epsilon)\frac{z_j^*}{1+\lambda} - (1+\epsilon)\hat{Z}_j\right\}\right] \;\leq\; \frac{\ln 2}{\epsilon},\end{aligned}$$

where $0 < \epsilon \leq \frac{1}{2}$. Since $\frac{1-\epsilon}{1+\epsilon} \geq 1-2\epsilon$, $\mathrm{Exp}[Z_j] \;\geq\; \frac{z_j^*}{1+\lambda} - 2\left(\frac{\epsilon z_j^*}{1+\lambda} + \frac{1}{\epsilon}\right)$, and thus we have

$$\mathrm{Exp}[Z] \;\geq\; \frac{z^*}{1+\lambda} - 2\left(\frac{\epsilon z^*}{1+\lambda} + \frac{1}{\epsilon}\right).$$

In the above inequality we substitute $\lambda = \frac{1}{\sqrt{z^*}}$ and choose $\epsilon = \frac{1}{2\sqrt{z^*}}$, which, by simple algebra, yields

$$\mathrm{Exp}[Z] \;\geq\; z^* - 6\sqrt{z^*}. \tag{28}$$

Likewise, $\mathrm{Exp}[\sum_{i=1}^m X_i] = \frac{1}{1+\lambda}\sum_{i=1}^m x_i^* \leq \frac{s}{1+\lambda}$. By Lemma 4, we have

$$\mathrm{Exp}[\max\{0, \textstyle\sum_{i=1}^m X_i - (1+\delta)\frac{s}{1+\lambda}\}] \;\leq\; \frac{2e^{-\frac{s}{1+\lambda}\delta^2/4}}{\ln(1+\delta)},$$

where $0 < \delta \leq 1$. Letting $\delta = \frac{1}{\sqrt{z^*}}$ and substituting $\lambda = \frac{1}{\sqrt{z^*}}$, the above inequality implies

$$\mathrm{Exp}[L] \;\leq\; \frac{2e^{-\frac{s}{4(z^*+\sqrt{z^*})}}}{\ln(1+\frac{1}{\sqrt{z^*}})} \;\leq\; \frac{2}{\ln(1+\frac{1}{\sqrt{z^*}})} \;\leq\; 4\sqrt{z^*}, \tag{30}$$

where the last inequality follows from $\ln(1+\epsilon) \geq \epsilon/2$ for $0 < \epsilon \leq 1$. Combining (28), (30), and $\tilde{Z} \geq Z - L$, we get

$$\mathrm{Exp}[\tilde{Z}] \;\geq\; \mathrm{Exp}[Z-L] \;\geq\; z^* - 10\sqrt{z^*}.$$

Since $1 \leq \mathcal{C}_{avg}^*(G,s) \leq z^*$, the above bound implies (26). □

Note that performance bounds for RCA2 and RCA are weaker than those for the algorithms in the previous sections, as it holds only in expectation. Algorithm RCA's approximation error is slightly worse than that of RCA2. Nevertheless, our experimental analysis (not included) show that on synthetic and real data sets, RCA2 does not outperform RCA.

Table 2. Performance of RCM on synthetic data with $m = 100$ and $n = 30$

s	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90
MinLP	10	12.5	15	17.5	20	22.5	25	27.5	29.82	31.89	33.88	35.76	37.54	39.11	40
RCM	7	10	13	15	18	19	23	25	28	30	32	34	35	38	40

5. Experimental Analysis

We implemented Algorithms RCM, RCM2 and RDM using *LP_SOLVE* solver [2], and tested their performance on both synthetic and real data.

Synthetic data. We tested these three algorithms on random data sets represented by adjacency matrices of four sizes $(m, n) = (100, 30)$, $(100, 100)$, $(200, 60)$, $(200, 200)$, where each element of the matrix is chosen to be 1 or 0 with probability $\frac{1}{2}$. We ran these programs for $s = 20, 21, ..., 90$, and compared the solution to the optimal solution of the linear relaxation (that is, RCM and RCM2 were compared to MinLP, RDM was compared to MaxLP).

Table 2 shows results of the comparison of RCM's solution and the MinLP solution of BCP_$\mathcal{C}_{min}$ from the experiment in which $m = 100$ and $n = 30$. This table presents only the performance of a single run of RCM, so the results are likely to be even better if we run RCM several times and choose the best solution.

We also repeated our simulation test 10 times for each of the above settings and took the average of them. Figure 1 and Figure 2 illustrate results of these experiments.

It is worth observing that (on average) RCM was always able to find the solution close to that of MinLP. Furthermore, since the true optimum (integral solution) for BCP_$\mathcal{C}_{min}$ could be smaller than the solution for MinLP, our approximation of RCM could be even closer to the true optimum than it appears. These observations apply to RDM as well.

We have repeated these experiments for sparser random matrices, where the values in the matrix were chosen to be 1 with probability $\frac{1}{4}$ or $\frac{1}{8}$. In all these experiments the results were very similar to those for the distribution with probability $\frac{1}{2}$.

Real data. To test the performance of these three algorithms on real data, we used four clone-probe adjacency matrices. The first two matrices represent hybridization of 500 bacterial clones extracted from rRNA genes analyzed in [12] with two sets of 30 and 40 probes designed with the algorithm in [3]. The other two matrices (with similar parameters) represent hybridization of rRNA genes from fungal clones analyzed in [11] with their corresponding sets of probes. For each of these four data sets (and for each $s = 200, 210, ..., 400$) we tested RCM 10 times, and took the average of them. We observe that RCM found the solution with value at least 97% of MinLP's solution in 92.8% cases. We also repeated our tests for RCM2 and RDM using the

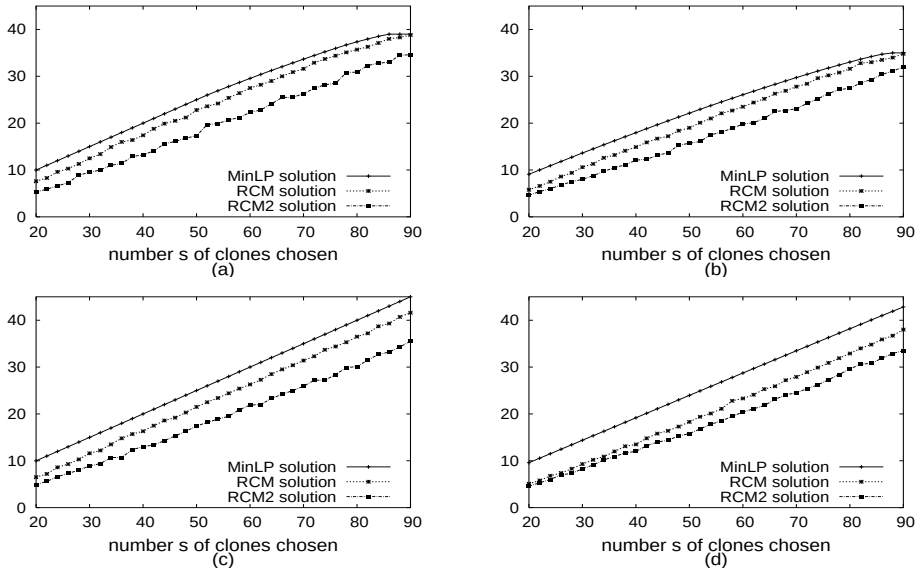

Fig. 1. RCM's and RCM2's performance for BCP_$\mathcal{C}_{min}$ on synthetic data for four matrices: (a) $(m, n) = (100, 30)$; (b) $(m, n) = (100, 100)$; (c) $(m, n) = (200, 60)$; (d) $(m, n) = (200, 200)$. The y-axis in the graph represents the objective value.

above data sets with the same settings. The results are summarized in Figure 3 and Figure 4.

Figure 3 and Figure 4 suggest that solutions found by RCM and RDM on real data are even closer to MinLP and MaxLP solutions, respectively, than those for synthetic data, sometimes even coinciding with MinLP and MaxLP solutions; i.e., RCM and RDM achieved optimum in some cases.

We also performed experimental analysis for RCA and RCA2 using the same synthetic and real data sets. The study shows that RCA and RCA2 approximate well the optimum solutions. (The results are similar to those of RCM and RCM2 and are omitted.)

Our experimental results indicate that RCM performs better than RCM2 in practice even though, according to our analysis in Section 4.1 and 4.2, RCM2 has a better asymptotic bound. This is likely to be caused by a combination of several factors. First, the constants in the asymptotic bounds for RCM2 appear to be larger than those for RCM, and our data sets may not be large enough for the asymptotic trends to show. Second, the bound for RCM2 is better than that for RCM only if the optimum is sufficiently small compared to $s/\log n$. As the parameters of this range depend on the hidden asymptotic constants, it is not clear whether our data sets are within this range. Finally, if the number of clones initially selected by the algorithm is less than s, our implementation of both algorithms adds some arbitrary clones to increase their number to s. Since RCM2 uses slightly smaller probabilities

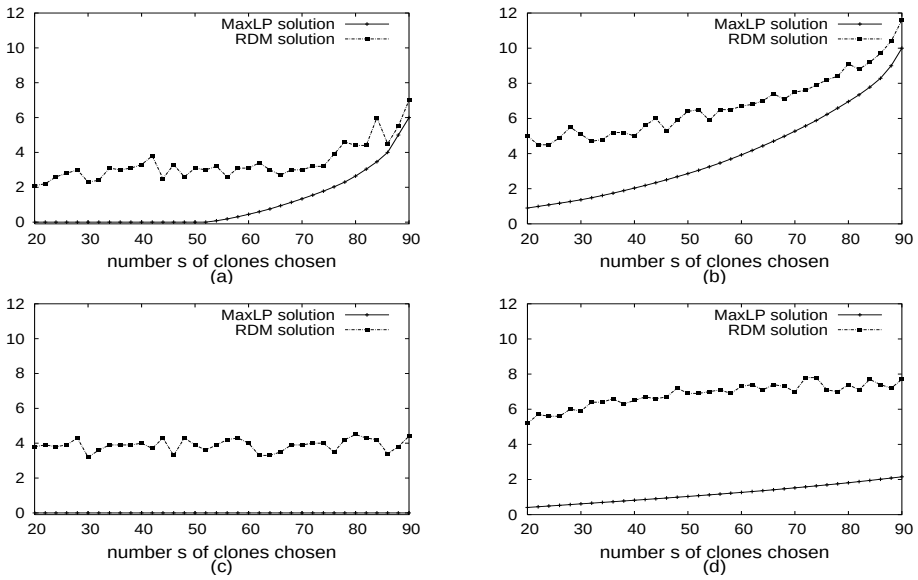

Fig. 2. RDM's performance for BCP_$\mathcal{D}_{max}$ on synthetic data for four matrices: (a) $(m, n) = (100, 30)$; (b) $(m, n) = (100, 100)$; (c) $(m, n) = (200, 60)$; (d) $(m, n) = (200, 200)$. The y-axis in the graph represents the objective value.

in the rounding scheme, it tends to choose initially fewer clones, and thus it is also likely to add more of these arbitrary clones. The performance of RCM2 relative to RCM would probably be improved with additional random sampling. (The same arguments apply to RCA and RCA2 as well.)

Our experiments were performed on a machine with Intel Pentium 4 2.4GHz CPU and 1GB RAM. The total running time for each single run of RCM or RDM on these synthetic and real data sets was in the range of 20-80 seconds, which is practically acceptable.

Example. To complement the above statistics with a more concrete example, we now describe the results of RCM on a typical data set. Here we used RCM to compute a set D of $s = 100$ control clones for $m = 500$ bacterial clones with a set of $n = 30$ probes. The distribution of the degrees of the probes with respect to D is given in the table below:

degree	0 –30	31 – 40	41 – 50	51 – 60	61 – 70	71 – 100
n. of probes	0	14	7	4	5	0

The minimum and maximum degrees of probes in D were 37 and 68, respectively, thus producing the objective value $\mathcal{C}_{min}(D) = 32$ for this instance. Thus this D is a high quality control clone set.

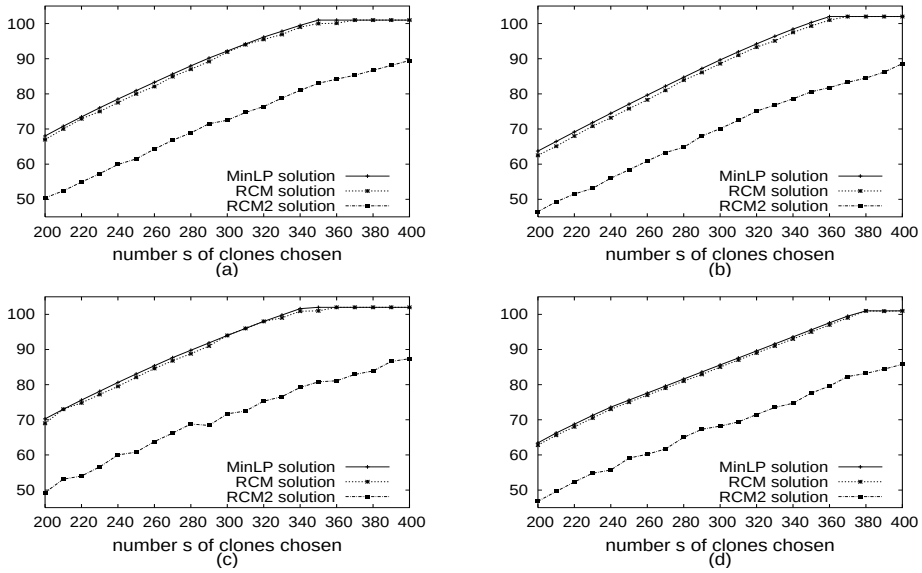

Fig. 3. RCM's and RCM2's performance on real data: (a) 500 bacterial clones and 30 probes; (b) 500 bacterial clones and 40 probes; (c) 500 fungal clones and 30 probes; (d) 500 fungal clones and 40 probes. The y-axis in the graph represents the objective value.

6. Concluding Remarks

We performed similar experiments for other algorithms provided in this paper, and the results were equally promising. Overall, our work demonstrates that randomized rounding is a very effective method for solving all versions of Balanced Covering, especially on real data sets. In the actual implementation available at `http://algorithms.cs.ucr.edu/OFRG/`, the solution of RCM is fed as an initial solution into a simulated-annealing algorithm. We found out that the simulated annealing rarely produces any improvement of this initial solution, which provides further evidence for the effectiveness of randomized rounding in this case. (In contrast, when we run simulated annealing from a random initial solution, in a typical run, it takes approximately 10 minutes to find a solution that is about 80% as good as that of RCM.)

We remark that (by creating two copies of the matrix and inverting the bits in the second copy) $\mathsf{BCP}\_\mathcal{C}_{min}$ can be reduced to a more general problem where we want to cover all columns with the maximum number of 1's. Our algorithms and their analyses apply to this problem as well.

Acknowledgments

This work is supported by NSF Grant BD&I-0133265. Work of Qi Fu and Marek Chrobak is partially supported by NSF Grant CCR-0208856.

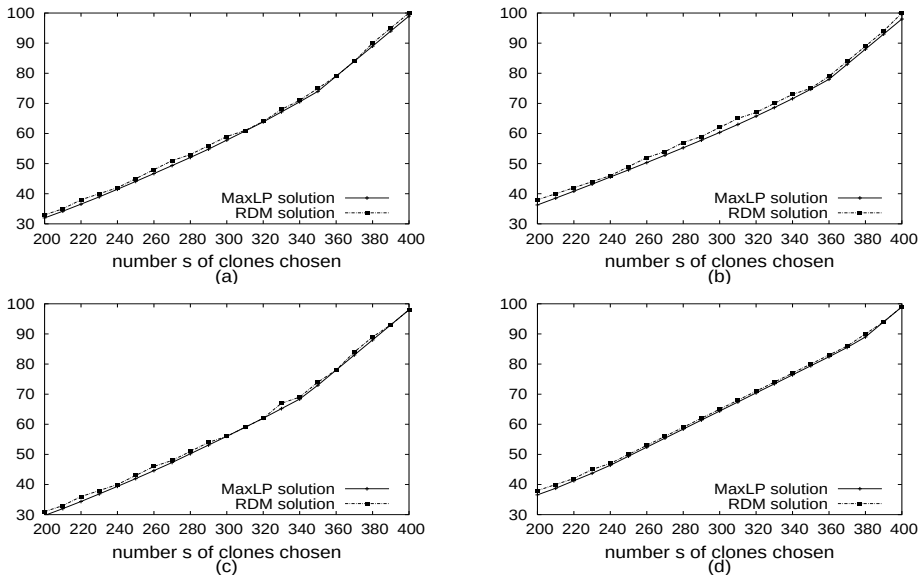

Fig. 4. RDM's performance on real data: (a) 500 bacterial clones and 30 probes; (b) 500 bacterial clones and 40 probes; (c) 500 fungal clones and 30 probes; (d) 500 fungal clones and 40 probes. The y-axis in the graph represents the objective value.

We would like to thank the anonymous referees for invaluable suggestions that helped us improve the presentation of this work.

Appendix A. Appendix: Proof of Lemma 4

Proof. Let $c(\epsilon) = e^{\epsilon}/(1+\epsilon)^{1+\epsilon}$ and $f(x) = -\ln(c(x))/x$. We will first show

$$\int_{x=\epsilon}^{+\infty} c(x)^{\mu}\, dx \;\leq\; \frac{2c(\epsilon)^{\mu}}{\mu \ln(1+\epsilon)}. \tag{A.1}$$

Integrating both sides of the inequality $1 + \ln(1+x) \leq 1 + x$, for $x \geq 0$, we get $(1+x)\ln(1+x) \leq x(1+x/2)$. By simple algebra, we then get $f'(x) \geq (\ln(1+x)/2)'$, and thus $f(x)$ is an increasing function and $f(x) \geq \ln(1+x)/2$. We can now verify (A.1) as follows,

$$\begin{aligned}
\int_{x=\epsilon}^{+\infty} c(x)^{\mu}\, dx &= \int_{x=\epsilon}^{+\infty} e^{-\mu x f(x)}\, dx \\
&\leq \int_{x=\epsilon}^{+\infty} e^{-\mu x f(\epsilon)}\, dx \\
&= \frac{e^{-\mu\epsilon f(\epsilon)}}{\mu f(\epsilon)} \\
&= \frac{c(\epsilon)^{\mu}}{\mu f(\epsilon)}
\end{aligned}$$

$$\leq \frac{2c(\epsilon)^{\mu}}{\mu \ln(1+\epsilon)}.$$

We have

$$\mathrm{Exp}[\max\{0, Y-(1+\epsilon)\mu\}] = \int_{y=0}^{+\infty} \Pr[Y-(1+\epsilon)\mu \geq y]\, dy.$$

Choose δ so that $(1+\delta)\mu = (1+\epsilon)\mu + y$. Changing variables from y to $(\delta-\epsilon)\mu$, and applying a standard Chernoff bound, the expected value above becomes

$$\begin{aligned}\mathrm{Exp}[\max\{0, Y-(1+\epsilon)\mu\}] &= \mu \int_{\delta=\epsilon}^{+\infty} \Pr[Y \geq (1+\delta)\mu]\, d\delta \\ &\leq \mu \int_{\delta=\epsilon}^{+\infty} c(\delta)^{\mu}\, d\delta.\end{aligned}$$

Combining this, inequality (A.1) and the fact that $c(\epsilon)^{\mu} \leq e^{-\mu\epsilon^2/4}$ for $0 < \epsilon \leq 1$, we get

$$\mathrm{Exp}[\max\{0, Y-(1+\epsilon)\mu\}] \;\leq\; \frac{2c(\epsilon)^{\mu}}{\ln(1+\epsilon)} \;\leq\; \frac{2e^{-\mu\epsilon^2/4}}{\ln(1+\epsilon)},$$

and the lemma follows. □

References

1. B. Berger, J. Rompel, and P. Shor. Efficient NC algorithms for set cover with applications to learning and geometry. *30th Annual Symposium on the Foundations of Computer Science*, pages 54–59, 1989.
2. M. Berkelaar, K. Eikland, and P. Notebaert. *Lp_solve mixed integer linear programming solver 5.5*, 2004. Available at `http://lpsolve.sourceforge.net/5.5`.
3. J. Borneman, M. Chrobak, G.D. Vedova, A. Figueroa, and T. Jiang. Probe selection algorithms with applications in the analysis of microbial communities. *Bioinformatics*, 17(1):S39–S48, 2001.
4. U. Feige. A threshold of ln n for approximating Set Cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
5. A. Figueroa, J. Borneman, and T. Jiang. Clustering binary fingerprint vectors with missing values for DNA array data analysis. *Journal of Computational Biology*, 11(5):887–901, 2004.
6. M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W.H.Freeman, New York, 1979.
7. L. Gargano, AA. Rescigno, and U. Vaccaro. Multicasting to groups in optical networks and related combinatorial optimization problems. *International Parallel and Distributed Processing Symposium*, page 223, 2003.
8. K. Jampachaisri, L. Valinsky, J. Borneman, and S. J. Press. Classification of oligonucleotide fingerprints: application for microbial community and gene expression analyses. *Bioinformatics*, 21(14):3122–3130, 2005.
9. J. Schuchhardt, D. Beule, A. Malik, E. Wolski, H. Eickhoff, H. Lehrach, and H. Herzel. Normalization strategies for cDNA microarrays. *Nucleic Acids Res.*, 28(10):e47, 2000.
10. L. Valinsky, A. Scupham, G.D. Vedova, Z. Liu, A. Figueroa, K. Jampachaisri, B. Yin, E. Bent, R. Mancini-Jones, J. Press, T. Jiang, and J. Borneman. Oligonucleotide

fingerprinting of ribosomal RNA genes (OFRG). In G.A. Kowalchuk, F.J. de Bruijn, I.M. Head, A.D. Akkermans, and J.D. Van Elsas, editors, *Molecular Microbial Ecology Manual*, pages 569–585. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2nd edition, 2004.

11. L. Valinsky, G. Della Vedova, T. Jiang, and J. Borneman. Oligonucleotide fingerprinting of ribosomal RNA genes for analysis of fungal community composition. *Applied and Environmental Microbiology*, 68(12):5999–6004, 2002.
12. L. Valinsky, G. Della Vedova, A. Scupham, S. Alvey, A. Figueroa, B. Yin, R. Hartin, M. Chrobak, D. Crowley, T. Jiang, and J. Borneman. Analysis of bacterial community composition by oligonucleotide fingerprinting of rRNA genes. *Applied and Environmental Microbiology*, 68(7):3243–3250, 2002.
13. N. E. Young. K-medians, facility location, and the Chernoff-Wald bound. *ACM-SIAM Symposium on Discrete Algorithms*, pages 86–95, 2000.
14. W. Yu, B.C. Ballif, C.D. Kashork, H.A. Heilstedt, L.A. Howard, W.W. Cai, L.D. White, W. Liu, A.L. Beaudet, B.A. Bejjani, C.A. Shaw, and L.G. Shaffer. Development of a comparative genomic hybridization microarray and demonstration of its utility with 25 well-characterized 1p36 deletions. *Human Molecular Genetics*, 12(17):2145–2152, 2003.