Identifying Codes and the Set Cover Problem

Moshe Laifenfeld, Ari Trachtenberg, and Tanya Y. Berger-Wolf[*†]

October 25, 2006

A version of this paper appeared as:

- M. Laifenfeld, A. Trachtenberg, and T.Y. Berger-Wolf, "Identifying codes and the set cover problem", *44th Annual Allerton Conf. on Comm., Ctrl., and Comput.*, October 2006.

Abstract

We consider the problem of finding a minimum identifying code in a graph, *i.e.,* a designated set of vertices whose neighborhoods uniquely overlap at any vertex on the graph. This *identifying code problem* was initially introduced in 1998 and has been since fundamentally connected to a wide range of applications, including fault diagnosis, location detection, environmental monitoring, and connections to information theory, superimposed codes, and tilings. Though this problem is NP-complete, its known reduction is from 3-SAT and does not readily yield an approximation algorithm. In this paper we show that the identifying code problem is computationally equivalent to the set cover problem and present a $\Theta(\log n)$-approximation algorithm based on the greedy approach for set cover; we further show that, subject to reasonable assumptions, no polynomial-time approximation algorithm can do better. Finally, we show that a generalization of the identifying codes problem, for which no complexity results were known thusfar, is NP-hard.

1 INTRODUCTION

An identifying code is a subset of vertices in a graph with the property that the incoming neighborhood of any vertex has a unique intersection with the code. The goal of the *identifying code problem* is to find an identifying code of minimum cardinality for a given graph. Identifying codes have been studied extensively since their introduction in 1998 [1], and they have formed a fundamental aspect of a wide variety of theoretical work and practical applications. They are also intimately linked to *locating-dominating sets*, introduced in 1988 [2] and well studied thereafter.

The initial application for identifying codes was to fault diagnosis in multiprocessor systems [1]. Since then, identifying codes were extended and applied to location detection in hostile environments [3–5], to energy balancing of such systems [6], and to dynamic location detection agents [7]. More recently, these codes were extended for applications to environmental monitoring [8].

From a theoretical perspective, identifying codes are closely linked to error-correcting codes, specifically, super-imposed codes [9], covering codes [1, 7], and locating-dominating codes [10]. The identifying code problem is itself NP-complete [11], although information-theoretic bounds are known [1, 6] and many of the best identifying codes to date are constructed for periodic geometries using known tilings. The source identification problem, a generalization of the identifying codes has been shown to be NP-complete for a time-constrained version [8] but no general results are known.

Despite the extensive amount of work in the area and the fundamental nature of the problem, little has been done towards developing a proven approximation algorithm for the identifying code problem, or its cousins, in the general case. The most general approach we know of so far is the greedy heuristic in [5], which Moncel showed to have no approximation guarantees in [12]. As such, the main contribution of this work is to relate two identifying code variants to well-studied covering problems, primarily to the set cover problem. In one case, our reduction is tight enough that we can carry over set cover hardness results together with greedy approximations, thus providing a concrete and efficient methods for producing provenly good identifying codes in *arbitrary graphs* for a wide variety of theoretical and practical applications.

[*] M. Laifenfeld and A. Trachtenberg are with the Department of Electrical Engineering, Boston University `moshel/trachten@bu.edu`

[†] T. Y. Berger-Wolf is with the Department of Computer Science, University of Illinois. `tanyabw@uic.edu`

The rest of the paper is organized as follows. We give formal definitions of the identifying codes, set covers and the source identification problem in Section 2. In Section 3 we show polynomial equivalence of the set cover problem and the identifying code problem, and we similarly prove the NP-completeness of the source identification problem. In Section 4 we provide an $O(\log n)$-approximation algorithm for the identifying code problem and show that this approximation ratio is tight. We also present simulation results on random graphs.

2 FORMAL DEFINITIONS AND RELATED WORK

2.1 Identifying codes

Given a directed graph $G = (V, E)$, an *incoming ball* $B^+(v)$ consists of edges directed towards $v \in V$, together with v; likewise, an *outgoing ball* $B^-(v)$ consists of edges directed away from v, together with v. For undirected graphs, we shall simply use the notation $B(v) = B^+(v) = B^-(v)$.

As such, an identifying code is a set of vertices in a graph G with the property that any incoming ball in G has a unique intersection with the identifying code. More precisely, a non-empty subset $\mathbb{C} \subseteq V$ is called a *code* and its elements are *codewords.* For a given code $\mathbb{C}$, the *identifying set* $I_{\mathbb{C}}(v)$ of a vertex v is defined to be the codewords directed towards v, *i.e.,* $I_{\mathbb{C}}(v) = B^+(v) \cap \mathbb{C}$ (if $\mathbb{C}$ is not specified, it is assumed to be the set of all vertices V). A code $\mathbb{C}$ is thus an *identifying code* if each identifying set of the code is unique, or in other words $\forall u, v \in V \quad u = v \longleftrightarrow I_{\mathbb{C}}(u) = I_{\mathbb{C}}(v)$.

Identifying codes were initially introduced in [1] and have since developed a prodigious following in the literature. Recently, random graphs were studied in the context of identifying codes [13], where it was shown that any subset of a certain threshold size is almost surely an identifying code (asymptotically in the size of the graph). It was also shown that the threshold is asymptotically sharp, *i.e.,* the probability of finding an identifying code of slightly smaller size asymptotically approaches zero. In contrast to very large random graphs, finding a minimum size identifying code for arbitrary undirected and directed graphs was proven to be NP-complete in [11, 14], based on a reduction from the 3-satisfiability problem. An exception to this result is the specific case of undirected trees, for which there exists a polynomial-time algorithm for finding a minimum radius 1 identifying code.

In spite of all the above work, little has been done towards a polynomial time approximation algorithm for arbitrary graphs. In [3, 5] identifying codes were suggested for use in indoor location detection. A polynomial-time greedy heuristic and its distributed variant were suggested for obtaining an identifying code, and simulations showed it to work well over random graphs. Unfortunately, no guarantees for the quality of the obtained solution were presented, and Moncel later proved in [12] that no such guarantees exist.

2.2 Covering problems

Let $\mathbf{U}$ be a base set of m elements and let $\mathcal{S}$ be a set of subsets of $\mathbf{U}$. We say that a subset $S \in \mathcal{S}$ *covers* its elements in $\mathbf{U}$. A *cover* $C \subseteq \mathcal{S}$ is a collection of subsets whose union is $\mathbf{U}$. The *set cover problem* asks to find a cover C of smallest cardinality. The set cover problem is one of the oldest and most studied NP-hard problem [15]. It admits the following (well-studied) greedy approximation: at each step, and until exhaustion, choose the heretofore unselected set in $\mathcal{S}$ that covers the largest number of uncovered elements in the base set.

The performance ratio of the greedy set cover algorithm has also been well-studied. The classic results of Lovasz and Johnson [16, 17] showed that $\frac{s_{\text{greedy}}}{s_{\min}} = \Theta(\ln m)$, where $s_{\min}, s_{\text{greedy}}$ are the minimum and the greedy covers, and m is the size of the base set. Later Slavik [18] sharpened this ratio further, reaching a difference of less than 1.1 between the lower and upper bounds on the performance ratio. Hardness results [19] show that for any $\epsilon > 0$, no polynomial-time algorithm can approximate the minimum set cover within $(1 - \epsilon) \ln m$ unless $NP \subset TIME\left[m^{O(\log \log m)}\right]$, suggesting that this greedy approach is one of the best polynomial approximations to the problem.

Another closely related problem is the *test cover problem.* This problem asks to find the smallest set $\mathbf{T}$ of tests $T_i \subset \mathbf{U}$ such that any pair $x, y \in \mathbf{U}$ is differentiated by at least one test T_i (*i.e.,* $|\{x, y\} \cap T_i| = 1$). The test covering problem appears naturally in identification problems, with roots in an agricultural study more than 20 years ago, regaining interest recently due to applications in bioinformatics. It is also a general case of the identifying code problem and thus motivates some of our work in Section 4.

Garey and Johnson [20] showed the test cover problem to be NP-hard and later Moret and Shapiro [21] suggested greedy approximations based on a reduction to the set cover problem. More recent work [22, 23] studied different branch-and-bound approximations and established a hardness of approximation by extending the reduction in [21], and using the result of Feige [19]. Berman et. al. [24] also suggested a novel greedy approximation and showed its performance ratio to be within a small constant from the hardness result of [23].

2.3 Source identification

An important related problem is the sensor placement problem [8]. Given a utility distribution network represented as a directed weighted graph, the sensor placement problem is to select the smallest subset of vertices where to place sensors. Various objectives of the sensor placement are discussed in [8]. The *source identification problem* aims to place sensors in a network so that the source of a flow originating from any of the vertices can be uniquely identified by the sensors. We assume that the weights in the utility network graph are defined by the shortest path metric of the flow translocation rates (the time it takes for a unit of material to pass from one vertex to the other).

The *source identification problem* is a generalization of the *identifying codes* problem. An identifying code must uniquely identify a vertex by the set of of the codewords within distance r of the vertex. An identifying sensor placement allows vertices to be identified by any sensor downstream from it and uses the distance ordering of sensors from the vertex as a distinguishing feature.

Let $G = (V, E)$ be a directed graph with positive edge weights representing the utility distribution network. Let $R \subseteq V$, $R = \{r_0, \dots, r_{|R|-1}\}$ be the set of sensors in the graph. For a vertex v, let $(r_0^v, \dots, r_{|R|-1}^v)$ be the sensors ordered by their distance from vertex v (breaking the ties by sensor number). Further, let $(t_1^v, \dots, t_{k-1}^v)$ be the distance difference sequence (*i.e.,* $t_j^v = d(v, r_j^v) - d(v, r_{j-1}^v) = w_{vr_j^v} - w_{vr_{j-1}^v}$ and is possibly infinite). Then we say the set of sensors R *distinguishes* between vertices u and v if the sequences $\langle (r_0^u, 0), (r_1^u, t_1), \dots, (r_{|R|-1}^u, t_{|R|-1}) \rangle$ and $\langle (r_0^v, 0), (r_1^v, t_1), \dots, (r_{|R|-1}^v, t_{|R|-1}) \rangle$ are different. The *source identification* problem is to find the smallest set R of sensors that distinguishes between any two vertices in V.

The time-constrained version of the source identification problem (*i.e.,* the sensors that distinguish between two given vertices must be within a given distance limit from the vertices) is NP-complete [8]. However, no progress has been made on the general source identification problem and no approximation algorithm is known.

3 REDUCTIONS

In this section we establish the computational equivalence between the identifying codes and the set cover problems via reductions to and from the set cover problem and the identifying code problem. In addition, we present a heretofore unknown equivalence between set cover and the source identification problem.

Formally, we connect the following problems:

SET-COVER
INSTANCE: Set $\mathcal{S}$ of subsets of a base set $\mathbf{U}$.
SOLUTION: A collection $C \subseteq \mathcal{S}$ such that $\cup_{s \in C} s = U$.
MEASURE: The size of the cover: $|C|$.

ID-CODE
INSTANCE: Graph $G = (V, E)$.
SOLUTION: A set $\mathbb{C} \subseteq V$ that is an identifying code of G.
MEASURE: The size of the identifying code: $|\mathbb{C}|$.

SOURCE-ID
INSTANCE: Directed weighted graph $G = (V, E)$, $w : E \to \mathcal{R}^+$.
SOLUTION: A set $R \subseteq V$ of source identifying sensors of G.
MEASURE: The size of the source identifying set: $|R|$.

3.1 ID-CODE $\leq_P$ SET-COVER

We first show a reduction from the minimum identifying code problem to the set cover problem. We state the main theorem first and then we provide several properties of identifying codes that are used in its proof.

Theorem 1 *Given a graph G of n vertices, finding an identifying code requires no more computations than a set cover solution over a base set of $\frac{n(n-1)}{2}$ elements together with $O(n^3)$ operations of length n binary vectors.*

Definition 1 *The* difference set $D_{\mathbb{C}}(u,v)$ *is the symmetric difference between identifying sets of vertices $u, v \in V$:*

$$D_{\mathbb{C}}(u,v) \doteq \mathrm{d}(I_{\mathbb{C}}(u), I_{\mathbb{C}}(v)) \doteq [I_{\mathbb{C}}(u) - I_{\mathbb{C}}(v)] \cup [I_{\mathbb{C}}(v) - I_{\mathbb{C}}(u)],$$

where subtraction denotes set difference. For simplicity of notation, we shall omit the subscript when looking at identifying codes consisting of all graph vertices, i.e., $D(u,z) = D_V(u,z)$. *We shall also use $\mathcal{D}$ to denote the set of difference sets of all vertices pairs,* i.e., $\mathcal{D} = \{D_{\mathbb{C}}(u,z) | (u,z) \in \mathbf{U}\}$ *where* $\mathbf{U} = \{(u,z) | u \neq z \in V\}$.

The following Lemma follows trivially from the definition of an identifying code.

Lemma 1 *A code $\mathbb{C}$ is an identifying code iff $\emptyset \notin \mathcal{D}$.*

Definition 2 *Let* $\mathrm{U} = \{(u,z) | u \neq z, u, z \in V\}$. *Then the* distinguishing set δ_c *is the set of vertex pairs in* U *for which c is a member of their difference set: $\delta_c = \{(u,z) \in \mathrm{U} \,|\, c \in D_{\mathbb{C}}(u,z)\}$.*

Lemma 2 $\mathbb{C}$ *is an identifying code iff the family of all distinguishing sets covers* $\mathrm{U} = \{(u,z) \,|\, u \neq z, \quad u, z \in V\}$.

Proof of Theorem 1: Consider the following construction of an identifying code: compute $\{I(u) | u \in V\}$ and $\Delta = \{\delta_u | u \in V\}$ and then find the minimum set cover of (U, Δ); the resulting identifying code will be $\mathbb{C} = \{u \in V | \delta_u \in \mathbf{C}\}$. Lemma 2 guarantees that $\mathbb{C}$ is an identifying code, and the optimality of the set cover guarantees that no smaller identifying code can be found. To complete the proof we observe that computing the identifying sets $I(u)$ naively requires $\theta(n^2)$ additions of binary vectors, and computing Δ requires n operations for each of the $\frac{n(n-1)}{2}$ elements in $|U|$. ∎

3.2 SET-COVER $\leq_P$ ID-CODE

Theorem 2 *Given a base set $\mathbf{U}$ of m elements and a family of subsets $\mathcal{S}$ of cardinality $s = |\mathcal{S}|$ finding the optimal set cover requires no more computations than finding an identifying code over either a directed graph of $n = 2\max(s, m+1) - 1$ vertices or an undirected graph of $2m + \max(s, m+1) \leq n \leq 3m + \max(s, m+1)$ vertices (together with $O(ms^2)$ operations).*

To prove the theorem we first provide and analyze a construction of a specific directed graph (that will be used in the reduction) from an instance of the set cover problem. The construction and proof for undirected graphs is omitted due to space limitations.

Construction 1 *Let $\mathbf{U} = \{v_1, ..., v_m\}$ be a base set and $\mathcal{S} = \{\mathbf{S}_1, ..., \mathbf{S}_{m+1}\}$ be a family of subsets of $\mathbf{U}$. Then we construct a directed graph $\mathcal{G}([\mathrm{U}], [S])$ with $n = 2m+1$ vertices $V = \{v_1, \ldots, v_n\}$ and edges E such that*

1. *The outgoing ball of each vertex $v_1 \ldots v_{m+1}$ is constructed to be*

$$B^-(v_i) = \begin{cases} \mathbf{S}_i & \text{if } v_i \in \mathbf{S}_i \\ V - \mathbf{S}_i & \text{otherwise.} \end{cases} \qquad (1)$$

2. *The outgoing balls of the remaining vertices $v_{m+2} \ldots v_n$ are constructed to be:*

$$B^-(v_i) = \{v_i, v_{i-m-1}\}.$$

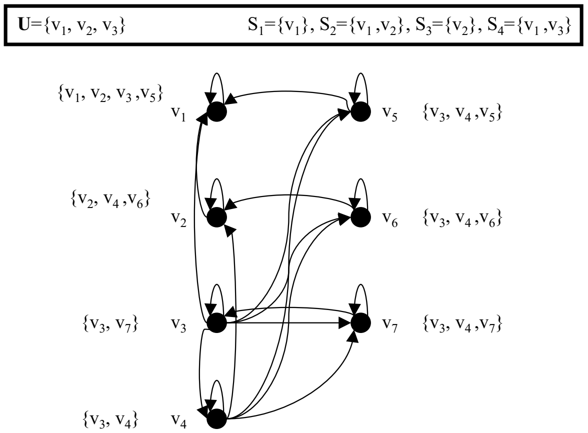


Figure 1: An example of our reduction framework. Incoming balls are noted near their corresponding vertices.

We next provide several properties of an arbitrary identifying code $\mathbb{C}$ for the graph $\mathcal{G}([\mathbf{U}], [S])$; it might be useful to refer to Fig. 1, which demonstrates our construction on a simple example, when reading these properties. Recall that we use the notation $D(v_i, v_j)$ to denote the difference set of the pair $(v_i, v_j) \in \mathcal{G}([\mathbf{U}], [S])$, and the notation δ_{v_i} to denote the distinguishing set of vertex v_i. We introduce as additional notation the set *[U]* $= \{(v_i, v_{i+m+1}) \,|\, i \leq m\}$ and corresponding operator $\overleftrightarrow{\delta} = \delta \cap$ *[U]*. The following properties are provided without proof.

Property 1 *Any identifying code of* $\mathcal{G}([\mathbf{U}], [S])$ *must contain all vertices* $v_{m+2} \ldots v_n$.

Property 2 *For all* $k \geq m+2$, *the distinguishing set* $\overleftrightarrow{\delta_{v_k}}$ *is empty.*

Property 3 $\mathbb{C}$ *is an identifying code iff* $\{v_{m+2} \ldots v_n\} \subseteq \mathbb{C}$ *and* $\{\overleftrightarrow{\delta_{v_i}} \mid i \leq m+1$ *and* $v_i \in \mathbb{C}\}$ *is a cover of* [U].

Note that Property 3 determines a one-to-one correspondence between identifying codes and set covers using distinguishing sets, so that, in fact, a minimum identifying code produces a minimum set cover. In addition, the family of distinguishing sets $\{\overleftrightarrow{\delta}_{v_i} | i \leq m+1\}$ over support *[U]* is equivalent to the original family of subset $\mathcal{S}$ over the support $\mathbf{U}$. We use this to develop the following Lemma, which is provided without proof.

Lemma 3 $\mathbb{C}$ *is an identifying code of* $\mathcal{G}([\mathbf{U}], [S])$ *if and only if* $\{v_{m+2} \ldots v_n\} \subseteq \mathbb{C}$ *and* $\{\mathbf{S}_i \mid v_i \in \mathbb{C}, i \leq m+1\}$ *is a set cover of* $(\mathbf{U}, \mathbf{S})$.

Proof of Theorem 2: Given a base set $\mathbf{U}$ of size m a family of subsets $\mathcal{S}$ of size s, we trivially produce sets $\mathbf{U}'$ and S' that fit Construction 1 as follows: (i) if $s < m+1$, then $\mathbf{U}'$ is derived from $\mathbf{U}$ by padding it with $\lg(m+1-s)$ new items[1], and S' is derived from S by adding all possible subsets of these new items; (ii) otherwise, $\mathbf{U}'$ is derived from $\mathbf{U}$ by padding it with $s-1-m$ new items, and these items are also added to each set in S to form S'. Lemma 3 then assures that a minimum identifying code of $\mathcal{G}([\mathbf{U}'], [S'])$ corresponds to a minimum set cover of $(\mathbf{U}, S)$. ∎

3.3 SET-COVER $\leq_P$ SOURCE-ID

We now show that the *source identification problem* is NP-complete using a reduction from the set cover problem to the source identification problem.

Theorem 3 *The source identification sensor placement problem is NP-complete.*

[1] As is common, we use the notation $\lg(x)$ to denote $\log_2(x)$.

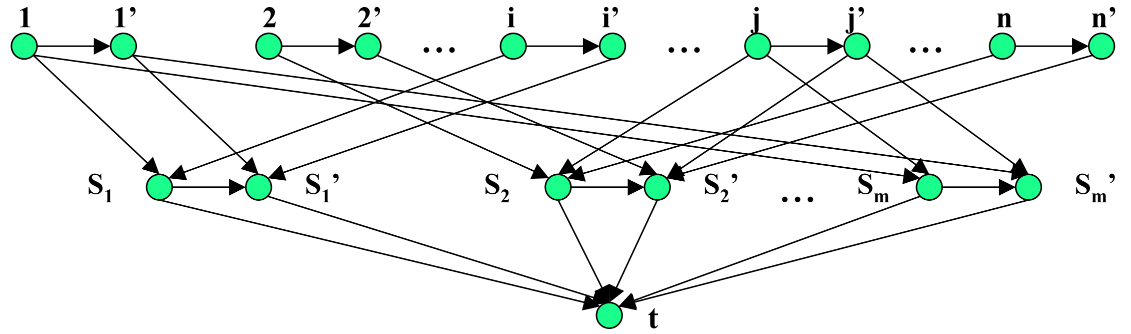


Figure 2: The graph resulting from an instance of a set cover problem.

Proof: Recall that an instance of a set cover problem is defined by a base set of elements $U = \{1, ..., n\}$ and a collection of sets $\mathcal{S} = \{S_1, ..., S_m\}$. We construct a graph $G = (V, E)$ such that there exists a set cover of size at most k $C \subseteq \mathcal{S}$ if and only if there exists a placement of $k + n + m + 1$ sensor for source identification in G.

For each element $i \in U$ we add two vertices $i, i' \in V$ and for each set $S_j \in \mathcal{S}$ we add two vertices $S_j, S'_j \in V$. We also add a vertex $t \in V$: $V = U \cup U' \cup \mathcal{S} \cup \mathcal{S}' \cup \{t\}$. For each element i we add the corresponding edge $(i, i') \in E$. For each set S_j we add an edge (S_j, S'_j) and edges (S_j, t) and (S'_j, t). For each element $i \in S_j$ we add edges (i, S_j) and (i', S'_j). All edges are of weight 1. See Fig. 2.

Lemma 4 *There exists a cover of size at most k in $\mathcal{S}$ if and only if there exists a set of identifying sensors of size $k + n + m + 1$ in G.*

Proof: Let $C \subseteq \mathcal{S}$ be a set cover of size k. We place sensors at all the vertices i', the vertices S_j corresponding to the sets in C, all the vertices S'_p, and the vertex t. Then for every item $i \in U$ there is an edge (i, S_j) such that $i \in S_j$ and $S_j \in C$ and therefore S_j has a sensor at it. Thus, every vertex i is identified by the tuple $\langle (S_j, 0), (i', 0), ... \rangle$. Every vertex i' is identified by $\langle (i', 0), (S'_j, 1), ... \rangle$. Similarly, the vertices S_j are identified by $\langle (S_j, 0), (S'_j, 1), (t, 1) \rangle$ if there is a sensor there, or $\langle (S'_j, 0), (t, 0) \rangle$. The vertices S'_j are identified by $\langle (S'_j, 0), (t, 1) \rangle$. Finally, the vertex t is identified by $\langle (t, 0) \rangle$.

Let $R \subseteq V$ be the set of identifying sensors. Note, that any vertex that has fewer than 2 outgoing edges must have a sensor in it for identification to be possible. Thus, $t \in R$ and all $i', S'_j \in R$. There must be additional vertices in R since $U' \cup S' \cup \{t\}$ does not distinguish between U and U'. Moreover, it is sufficient to have a sensor at either i or S_j such that $(i, S_j) \in E$. We can replace every sensor at a vertex in U with a sensor at a vertex in $\mathcal{S}$ connected to it without increasing the size of R and still have an identifying sensor set. Given this set R we create a set cover C by choosing every set corresponding to a vertex $S_j \in R$. Since every $i \in U$ has an edge to some $S_j \in R$ then, by construction, every $i \in U$ is contained in some set $S_j \in C$. Thus, by definition, C is a set cover of size $|R| - |U'| - |\mathcal{S}'| - |\{t\}| = |R| - n - m - 1$, as claimed. ∎

Polynomial verification of a source identifying solution is immediate, thus completing our NP-completeness proof. ∎

4 APPROXIMATING THE IDENTIFYING CODE

Given a basis set $\mathbf{U}$ of size m and a family of subsets $\mathcal{S}$, a well-known greedy approximation involves repeatedly picking (until exhaustion) an unused set in $\mathcal{S}$ that covers the largest number of uncovered elements of $\mathbf{U}$. This algorithm has performance ratio $\frac{s_{\text{greedy}}}{s_{\min}} = \ln m - \ln \ln m + \Theta(1)$, where $s_{\min}$ and s_{greedy} represent the minimum and greedily produced set covers respectively [18]. The reduction in Theorem 2 thus provides the following construction.

Construction 2 *Let G=(V,E) be a given graph, and let* Set_cover_greedy$(\mathbf{U}, \mathcal{S})$ *be the greedy set cover algorithm, then the identifying code greedy algorithm is:*

```
IDgreedy$(G) \rightarrow \mathbb{C}_{greedy}$
    Compute $\{I(u) | u \in V\}$.
    Compute $\Delta = \{\delta_u | u \in V\}$.
    $\mathbf{C} \leftarrow$ Set_cover_greedy$(\mathbf{U}, \Delta)$
    Output $\mathbb{C}_{greedy} \leftarrow \{u \in V \,|\, \delta_u \in \mathbf{C}\}$
```

The remainder of this section is devoted to proving that the reduction in Theorem 1 is tight enough to maintain the approximation guarantees of the set cover solution. This result is formalized with the following theorem.

Theorem 4 *There exist non-negative constants c_1 and c_2 such that, for any graph G on n vertices,*

$$c_1 \ln n \; < \; \frac{c_{greedy}}{c_{min}} \; < \; c_2 \ln n,$$

where c_{min} and c_{greedy} are the sizes of the minimum and greedy identifying codes respectively.

The upper bound of Theorem 4 follows from the fact that transformation in Theorem 1 maps identifying codes on n vertices to set covers over base sets of size $\frac{n(n-1)}{2}$. As such, since greedy set cover algorithm has an approximation guarantee of $\frac{s_{\text{greedy}}}{s_{\min}} < c' \ln m$, we have that $\frac{c_{greedy}}{c_{min}} < c' \ln \frac{n(n-1)}{2} < 2c' \ln n$. We will prove the lower bound of the theorem by providing a specific example that attains it in Section 4.1 and thereafter. As a basis for the lower bound example, we first provide the following lemma without proof.

Lemma 5 *Consider a collection of m non-empty sets, $\mathcal{M} = \{\mathbf{M}_1, ..., \mathbf{M}_m\}$, over a base set $\mathbf{U} = \{u_1, \ldots, u_k\}$ of size $k \geq \lg(m) + 2$. Then there is a collection of $2m$ different sets $\mathcal{I} = \{\mathbf{I}_1, ..., \mathbf{I_{2m}}\}$ such that:*

$$u_i \in \mathbf{I}_i \text{ for all } i \leq k, \quad \text{and} \quad M_i = \mathbf{I}_i \oplus \mathbf{I}_{i+m} \text{ for all } i \leq m.$$

We henceforth assume that the elements of $\mathcal{I}$ are arranged so that $\mathbf{I}_i \oplus \mathbf{I}_{i+m} = \mathbf{M}_i$ for all $i \leq m$.

4.1 Lower bound construction

We now develop the construction that will provide our desired lower bound on approximation. Our construction transforms certain instances of the set cover problem into an identifying code problem. The salient point of the construction is that it provides an explicit link between the cardinalities of the minimum (or greedy) set covers in one problem and the minimum (or greedy) identifying codes in the other problem. We shall then make use of an existing result in the literature to show that the desired set cover instances exist.

Construction 3 *Let $(\mathbf{U} = \{u_i, ...u_m\}, \mathcal{S} = \{\mathbf{S}_1, ...\mathbf{S}_{2m-k}\})$ be a set cover problem. Furthermore, let $\mathcal{S}_{min}$, $\mathcal{S}_{greedy}$, s_{min}, and s_{greedy} be the minimum and greedy set covers and their sizes, and assume that $m = 2^k$ and $s_{min} \geq k + 2$.*

We then generate a graph G from $(\mathbf{U}, \mathcal{S})$ as follows. The graph will have $n = 2m$ vertices, with vertex v_i corresponding to set $\mathbf{S}_i$ for $i \leq m$. To determine the edges of the graph, we shall make use of two collections:

- *$\mathcal{M} = [\mathbf{M}_i]$ is a collection of m sets such that $\mathbf{M}_i = \{v_j \,|\, \mathbf{S}_j \in \mathcal{S}_{min}$ and $u_i \in \mathbf{S}_j\}$.*
- *$\overline{\mathcal{M}} = [\overline{\mathbf{M}}_i]$ is the collection of m sets such that $\overline{\mathbf{M}}_i = \{v_j \,|\, \mathbf{S}_j \notin \mathcal{S}_{min}$ and $u_i \in \mathbf{S}_j\}$.*

Provided that $k > 1$, Lemma 5 implies the existence of the set:

- *$\mathcal{I} = \{\mathbf{I}_i\}$ having $2m$ distinct sets from $\mathcal{S}_{min}$ such that:*
  (i) $\mathbf{I}_i \oplus \mathbf{I}_{i+m} = \mathbf{M}_i$ for all $i \leq m$, and (ii) $v_j \in \mathbf{I}_j$ for all j such that $S_j \in \mathcal{S}_{min}$.

We can also simply generate the following list:

- *$\overline{\mathcal{I}} = [\overline{\mathbf{I}}_i]$ having $2m$ sets from $U - \mathcal{S}_{min}$ such that:*
  (i) $\overline{\mathbf{I}}_i \oplus \overline{\mathbf{I}}_{i+m} = \overline{\mathbf{M}}_i$ for all $i \leq m$, and (ii) $v_j \in \overline{\mathbf{I}}_j$ for all j such that $S_j \notin \mathcal{S}_{min}$.

This is done by setting $\overline{\mathcal{I}}_i = \emptyset$ and $\overline{\mathcal{I}}_{i+m} = \overline{\mathbf{M}}_i$ for $i \leq m$, and then toggling the existence of u_i and u_{i+m} in sets $\overline{\mathcal{I}}_i$ and $\overline{\mathcal{I}}_{i+m}$ so as to satisfy the stated properties. The edges of G are then defined as

$$B^+(v_i) = \mathbf{I}_i \cup \overline{\mathbf{I}}_i \cup p_i \pmod{m} \tag{2}$$

where p_j is the $j + 1$-th set in the power set $\mathcal{P}(\{v_{2m-k+1} \ldots v_{2m}\})$, where the power set elements are ordered so that the $m - i$-th set contains u_i (thus ensuring an all-one diagonal in the adjacency matrix of G).

$$M(\mathbf{U},\mathcal{S}) = \left[\begin{array}{c||ccc|ccccc|cccccccc}
 & v_{14} & v_{15} & v_{16} & v_1 & v_3 & v_5 & v_6 & v_7 & v_4 & v_2 & v_8 & v_9 & v_{10} & v_{11} & v_{12} & v_{13} \\
\hline\hline
v_{14} & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
v_{15} & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{16} & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
v_1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
v_3 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
v_5 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
v_6 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_7 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
v_4 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
v_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
v_8 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
v_9 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
v_{10} & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
v_{11} & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
v_{12} & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
v_{13} & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}\right]$$

Figure 3: The adjacency matrix of a graph $G(\mathbf{U},\mathcal{S})$ from Construction 3.

4.2 Example of Construction 3

Consider the base set $\mathbf{U} = \{u_1, u_2, u_3, \ldots u_8\}$ and the collection of sets

$$\begin{array}{rl} \mathcal{S} = \{ & \{u_1,u_2\},\{u_2,u_3\},\{u_4,u_3\},\{u_4,u_5\},\{u_5,u_6\}, \\ & \{u_6\},\{u_1,u_7\},\{u_1,u_8\},\{u_1\}, \\ & \{u_2\},\{u_3\},\{u_4\},\{u_5\} \qquad \} \end{array}.$$

In the terminology of Construction 3, we have $k = 3$, $m = 8$, and it is clear that the smallest set covering for $(\mathbf{U},\mathcal{S})$ is of size $s_{\min} = 5$. We then generate a graph $G = (V,E)$ corresponding to this set cover problem, with vertices $v_1 \ldots v_{16}$. Following the construction, we compute:

$$\begin{array}{rl} \mathcal{M} = [& \{v_1,v_7,v_8\},\{v_1\},\{v_3\},\{v_3\}, \\ & \{v_5\},\{v_5\},\{v_7\},\{v_8\} \qquad] \end{array}$$

$$\begin{array}{rl} \overline{\mathcal{M}} = [& \{v_8\},\{v_2,v_9\},\{v_2,v_{10}\}, \\ & \{v_4,v_{11}\},\{v_4,v_{12}\},\{v_{13}\},\emptyset,\emptyset \quad] \end{array}.$$

Intuitively, the i-th set in $\mathcal{M}$ represents the sets that cover basis element u_i in the minimum set cover, whereas the i-th set in $\overline{\mathcal{M}}$ represents the the sets that cover u_i but are not in the minimum set cover. Utilizing Lemma 5, we also construct two collections:

$$\begin{array}{rl} \mathcal{I} = \{ & \{v_1\},\{v_7\},\{v_3,v_6\},\{v_3,v_5\},\{v_3,v_5,v_7\}, \\ & \{v_3,v_6,v_5,v_7\},\{v_1,v_5,v_7\},\{v_1,v_3\}, \\ & \{v_6,v_7\},\ldots \qquad \} \end{array}$$

$$\begin{array}{rl} \overline{\mathcal{I}} = [& \{v_1,v_9\},\{v_2,v_{10}\},\{v_3,v_{11}\},\{v_4,v_{12}\}, \\ & \{v_5,v_{13}\},\{v_6,v_{14}\},\{v_7,v_{15}\},\{v_8,v_{16}\}, \\ & \{v_1,v_8,v_9\},\{v_9,v_{10}\},\ldots \qquad] \end{array}$$

Finally, applying (2) provides the edges of the graph in terms of incoming balls of vertices, the first of which are: $B^+(v_1) = \{v_1,v_9,v_{16}\}$, $B^+(v_2) = \{v_2,v_7,v_{10},v_{15}\}$, $B^+(v_3) = \{v_3,v_6,v_{11},v_{15},v_{16}\}$... It is easier to understand the graph in terms of its adjacency matrix, depicted in Fig. 3 with each row representing an incoming ball.

4.3 Lower bound

We next provide some properties Construction 3 that will be crucial in completing the proof of the lower approximation bound of Theorem 4. We omit their proof due to space considerations.

Property 4 *Given a set cover problem* $(\mathbf{U},\mathcal{S})$ *with* $|\mathbf{U}| = 2^k = m$, $|\mathcal{S}| = 2m - k$, *and* $s_{min} \geq k + 2$, *Construction 3 produces a graph* G *with the following properties:*

1. *The vertices V_{min} associated with $\mathcal{S}_{min}$ form an identifying code of $G = (V, E)$. More precisely, V_{min} contains only vertices v_i, where i is such that $S_i \in \mathcal{S}_{min}$.*

2. *The distinguishing sets of $\hat{V} = \{v_{2m-k+1}, ..., v_{2m}\} \subseteq V$ cover all pairs except* [U]. *In other words*

$$\begin{aligned}&\textstyle\bigcup_{i \in \{2m-k+1 \ldots 2m\}} S_i = \\ &\quad \{(v_u, v_z) \,|\, z \notin \{u, u+m\} \textit{ and } 1 \leq u, z \leq 2m\}.\end{aligned}$$

3. *The modified set cover problem* $([\mathrm{U}], \{\overleftrightarrow{\delta_{v_1}} \ldots \overleftrightarrow{\delta_{v_{2m}}}\})$ *is equivalent to the original problem* $(\mathbf{U}, \mathcal{S})$. *As such, the function $f : [\mathrm{U}] \longrightarrow U$ where $f(v_i, v_{i+m}) = u_i$ has the property that $f(\overleftrightarrow{\delta_{v_i}}) = S_i$, with the usual understanding that $f(S) = \cup_{s \in S} f(s)$. Note that this also provides an equivalence between covers in the modified problem and covers in the original problem.*

Corollary 1 *The directed graph G generated by Construction 3 has the following two properties: (1) $c_{min} = s_{min}$, and (2) $c_{greedy} = s_{greedy} + k$.*

Proof of Lower bound of Theorem 4: Slavik [18] demonstrated that there exist set cover problems (U, S) with greedy covers of any size s_{greedy} and minimum covers of size $s_{\text{min}} \geq 2$ as long as the size $|U| = m$ is at least as large as the *greedy number* $N(s_{\text{greedy}}, s_{\text{min}})$. He further showed that, for any $k' \geq l \geq 2$ the greedy number satisfies

$$\begin{aligned}&\ln N(k', l) \leq \\ &\quad \ln l + \tfrac{2l-1}{2l(l-1)} \left[(k' - l) + (l - 2)\left(1 - (\tfrac{l-1}{l})^{k'-l}\right)\right],\end{aligned} \tag{3}$$

which can be (weakened and) simplified to $\ln l + \frac{k'}{l-1}$. As such, we can see that, for $s_{\text{greedy}} \geq s_{\text{min}} \geq 2$, if

$$\ln m \geq \ln s_{\text{min}} + \frac{s_{\text{greedy}}}{s_{\text{min}} - 1}, \tag{4}$$

then $m > N(s_{\text{greedy}}, s_{\text{min}})$, and a corresponding set cover problem exists.

In order to apply Construction 3 to a set cover problem (U, S) produced by Slavik's construction, we need to ensure that the construction's assumptions are satisfied, namely that (i) $m = 2^k$, (ii) $s_{\text{min}} = k + 2$, and that (iii) $|S| = 2m - k$. In addition, we need that the constructed graph to have the property that $\frac{c_{\text{greedy}}}{c_{\text{min}}} \geq c \ln n = c \ln m + c \ln 2$ for some constant c in order to have our performance bound; under Corollary 1, this corresponds to a condition (iv) that $\frac{s_{\text{greedy}}+k}{s_{\text{min}}} = c \ln m + c \ln 2$, which reduces to

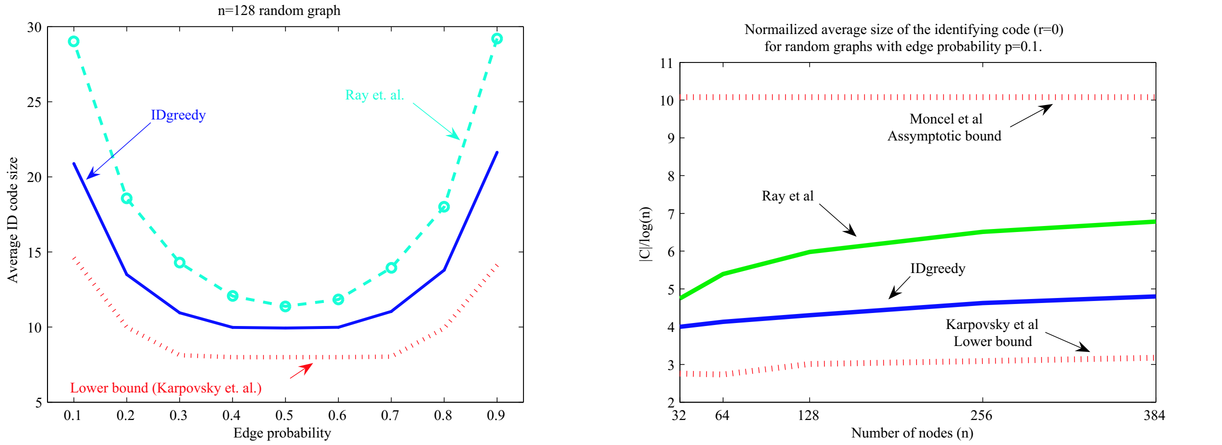
$$s_{\text{greedy}} = ck^2 + 2ck + 2c \ln 2. \tag{5}$$

Thus, if we have set cover problems satisfying conditions (i)-(iv), then we can create an identifying code instance satisfying the lower bound of Theorem 4.

Slavik's construction trivially satisfies conditions (i) and (ii) since (4) holds for any given $m, s_{\text{min}} \geq 2$. Reconciling (4) with (5), we see that condition (iv) is satisfied when $ck^2 + 2ck + 2c \ln 2 \leq k^2 + (\ln 2 - \ln(k+2))k - \ln(k+2)$, which will clearly hold for any $0 \leq c < 1$ when k is sufficiently large. Finally, we can transform a set cover problem (U, S) satisfying conditions (i),(ii), and (iv) into a problem (U, S') satisfying (i)-(iv) as follows:

If $|S| < 2m - k$, then we can pad S with empty sets to get S' of cardinality $2m - k$ without breaking Construction 3.

If $|S| > 2m - k$, then we can take $S' = \mathcal{S}_{\text{min}} \cup S_{\text{greedy}}$ without violating any conditions. This new set will have cardinality $|S'| \leq s_{\text{min}} + s_{\text{greedy}} \leq (k + 2) + ck^2 + 2ck + 2c \ln 2$, which is clearly $\leq m$ for $c = 1$ and large k. ∎



(a) Average ID code size as a function of edge probability.

(b) Normalized average code size as a function of the number of vertices.

Figure 4: Simulation results for random graphs.

4.4 Hardness of approximation

We next manipulate the reductions of Theorem 2 to carry set cover hardness results over to the identifying code problem. Our work is based on the work of Feige [19] proving that (for any $\epsilon > 0$) no polynomial-time approximation of set cover can attain a performance ratio of $(1-\epsilon)\ln m$ unless NP$\subset$DTIME($m^{\lg\lg m}$). Our proof differs from the more general hardness result of [23] for the test cover problem because of the constraints imposed by the (undirected) graph structure on which identifying codes are defined (rather than the arbitrary "tests" permitted in the test cover problem). The proof is quite involved, and we omit it due to space considerations.

Theorem 5 *For any $\epsilon > 0$ the identifying code problem has no polynomial time approximation with performance ratio $(1-\epsilon)\ln n$ (for directed and undirected graphs) unless NP$\subset$DTIME($n^{\lg\lg n}$).*

The identifying codes problem is actually a special case of the test cover problem, and thus test cover approximations can be applied to produce "good" identifying codes. One such greedy approximation was recently devised by Berman et. al. [24] using a modified notion of entropy as the optimization measure. This greedy approximation was proven to have a performance ratio of $1 + \ln n$, where n is the number of elements in the basis set. Applying this algorithm to graphs of size n guarantees identifying codes with the same performance ratio, closing the gap (up to a small constant) from the lower bound of Theorem 5.

5 SIMULATIONS

We have simulated the approximations based on the greedy set cover heuristic and the entropy based approximation of [24], and applied them to random graphs with different edge probabilities. We use average identifying code size as our performance measurements, and our simulation results are compared to the algorithm of Ray et. al. [5]. For the sake of comparison, we also show the combinatorial lower bound of Karpovsky et al. [1] and the convergence bound of Moncel et al. [13], who provided a threshold number of vertices that, with high probability, will form an identifying code when the size of the graph tends to infinity.

Fig. 4 shows the theoretic lower bound and the results of our greedy algorithm (IDgreedy). Our algorithm clearly improves over that of Ray et al., although the curves should converge (apparently at a very slow rate) to the result of Moncel et al. for large n. The slow convergence rate suggests that one can gain significantly from more sophisticated identifying code algorithms, even for reasonably large graphs.

6 ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewer for referring them to the closely related problem of test cover, which allowed to significantly extend the main results of this paper.

References

[1] M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin, “A new class of codes for identification of vertices in graphs,” *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 599–611, March 1998.

[2] P. Slater, “Fault-tolerant locating-dominating sets,” *Discrete Mathematics*, vol. 249, pp. 179–189, 2002.

[3] S. Ray, R. Ungrangsi, F. D. Pellegrinin, A. Trachtenberg, and D. Starobinski, “Robust location detection in emergency sensor networks,” *Proc. INFOCOM*, April 2003.

[4] R. Ungrangsi, A. Trachtenberg, and D. Starobinski, “An implementation of indoor location detection systems based on identifying codes,” *Proc. of the IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM 04)*, 2004.

[5] S. Ray, D. Starobinski, A. Trachtenberg, and R. Ungrangsi, “Robust location detection with sensor networks,” *IEEE Journal on Selected Areas in Comm. (Special Issue on Fundamental Performance Limits of Wireless Sensor Networks)*, vol. 22, no. 6, August 2004.

[6] M. Laifenfeld and A. Trachtenberg, “Disjoint identifying codes for arbitrary graphs,” *IEEE International Symposium on Information Theory, Adelaide Australia*, 4-9 Sept 2005.

[7] I. Honkala, M. Karpovsky, and L. Levitin, “On robust and dynamic identifying codes,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 2, pp. 599–612, February 2006.

[8] T. Y. Berger-Wolf, W. E. Hart, and J. Saia, “Discrete sensor placement problems in distribution networks,” *Journal of Mathematical and Computer Modelling*, vol. 42, no. 13, pp. 1385–1396, Dec 2005.

[9] S. Gravier and J. Moncel, “Construction of codes identifying sets of vertices,” *The Electronic Journal of Comb.*, vol. 12, no. 1, 2005.

[10] I. Honkala and A. Lobstein, “On identifying codes in binary hamming spaces,” *Journal of Comb. Theory, Series A*, vol. 99, no. 2, pp. 232–243, 2002.

[11] I. Charon, O. Hudry, and A. Lobstein, “Identifying and locating-dominating codes: Np-completeness results for directed graphs,” pp. 2192–2200, August 2002.

[12] J. Moncel, “Optimal graphs for identification of vertices in networks,” preprint at www-leibniz.imag.fr/NEWLEIBNIZ/LesCahiers/2005/Cahier138/CLLeib138.pdf.

[13] J. Moncel, A. Frieze, R. Martin, M. Ruszink, and C. Smyth, “Identifying codes in random networks,” *IEEE International Symposium in Information Theory, Adelaide, 4-9 Sept.*, 2005.

[14] I. Charon, O. Hudry, and A. Lobstein, “Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard,” *Theoretical Computer Science*, vol. 290, no. 3, pp. 2109–2120.

[15] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms.* MIT Press, 2001.

[16] L. Lovasz, “On the ratio of optimal integral and fractional covers,” *Discrete Mathematics*, vol. 13, pp. 383–390, 1975.

[17] D. S. Johnson, “Approximation algorithms for combinatorial problems,” *Journal of Comp. and System Sciences*, vol. 9, pp. 256–278, 1974.

[18] P. Slavik, “A tight analysis of the greedy algorithm for set cover,” *In Proceedings of the 28th Annual ACM Symposium on Theory of Computing (Philadelphia, Pa., May 2224)*, pp. 435–439, 1996.

[19] U. Feige, “A threshold of $\ln n$ for approximating set cover,” *In. Proc. ACM Symp. on Theory of NP-Completeness*, 1996.

[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman & Co., 1979.

[21] B. M. E. Moret and H. D. Shapiro, “On minimizing a set of tests,” *SIAM J. Scientific & Statistical Comput.*, vol. 6, pp. 983–1003, 1985.

[22] K. M. J. D. Bontridder, B. J. Lageweg, J. K. Lenstra, J. B. Orlin, and L. Stougie, “Branch-and-bound algorithms for the test cover problem,” in *ESA ’02: Proceedings of the 10th Annual European Symposium on Algorithms.* London, UK: Springer-Verlag, 2002, pp. 223–233.

[23] K. D. Bontridder, B. Halldrsson, M. Halldrsson, C. Hurkens, J. Lenstra, R. Ravi, and L. Stougie, “Approximation algorithms for the test cover problem,” *Math. Programming-B 98*, no. 1-3, pp. 477–491, 2003.

[24] P. Berman, B. DasGupta, and M.-Y. Kao, “Tight approximability results for test set problems in bioinformatics,” *J. Comput. Syst. Sci.*, vol. 71, no. 2, pp. 145–162, 2005.