The d-Identifying Codes Problem for Vertex Identification in Graphs: Probabilistic Analysis and an Approximation Algorithm*

Ying Xiao[1], Christoforos Hadjicostis[2], and Krishnaiyan Thulasiraman[1]

[1] School of Computer Science
University of Oklahoma, Norman, OK 73019, USA
{ying_xiao, thulsi}@ou.edu
[2] University of Illinois at Urbana-Champaign, IL 61801, USA
chadjic@uiuc.edu

Classification of Topics: Algorithms and data structures, Approximation algorithm

Proofs omitted due to space constraints are put into the appendix.

* This work was supported by the National Science Foundation under the ITR grant ECS-0426831

Abstract. Given a graph $G(V, E)$, the identifying codes problem is to find the smallest set of vertices $D \subseteq V$ such that no two vertices in V are adjacent to the same set of vertices in D. The identifying codes problem has been applied to fault diagnosis and sensor based location detection in harsh environments. In this paper, we introduce and study a generalization of this problem, namely, the d-identifying codes problem. We propose a polynomial time approximation algorithm based on ideas from information theory and establish its approximation ratio which is very close to the best possible. Using analysis on random graphs, several fundamental properties of the optimal solution to this problem are also derived.

1 Introduction

Consider an undirected graph G with vertex set V and edge set E. A ball of radius $t \geq 1$ centered at a vertex v is defined as the set of all vertices that are at distance t or less from v. The vertex v is said to cover itself and all the vertices in the ball with v as the center. The identifying codes problem defined by Karpovsky et al. [9] is to find a minimum set D such that every vertex in G belongs to a unique set of balls of radius $t \geq 1$ centered at the vertices in D. The set D may be viewed as a code identifying the vertices and is called an identifying set. Two important applications have triggered considerable research on the identifying codes problem. One of these is the problem of diagnosing faulty processors in a multiprocessor system [9]. Another application is robust location detection in emergency sensor networks [13]. Next we briefly describe the application of identifying codes in fault diagnosis.

Consider a communication network modeled as an undirected graph G. Each vertex in the graph represents a processor and each edge represents the communication link connecting the processors represented by the end vertices. Some of the processors could become faulty. To simplify the presentation let us assume that at most one processor could become faulty at any given time. Assume that a processor, when it becomes faulty, can trigger an alarm placed on an adjacent processor. We would like to place alarms on certain processors that will facilitate unique identification of the faulty processors. We would also like to place alarms on as few processors as possible. If D is a minimum identifying set for the case $t = 1$, then placing alarms on the processors represented by the vertices in the set D will help us to uniquely identify the faulty processor. Notice that we only need to consider $t = 1$ because if $t > 1$ is desired, we can proceed with G^t, the tth power of G.

Karpovsky et al [9] have studied the identifying codes selection problem extensively and have established bounds on the cardinally of the identifying sets. They have shown how to construct the identifying sets for specific topologies such as binary cubes and trees. For arbitrary topology, [2] presents heuristic approaches for a closely related problem that arises in selecting probes for fault localization in communication networks. Several problems closely related to the

identifying codes problem have been studied in the literature. Some of these may be found in [5], [6], [10], [11].

Karpovsky et al. [9] have shown that unique identification of vertices may not always be possible for certain topologies. In other words, triggering of alarms on a set of processors could mean that one of several candidate processors could be faulty. Once such a set of possible faulty processors has been identified then testing each processor in this set will identify the faulty processor. This motivates the generalization of the identifying codes problem to d-identifying codes problem defined below. This generalization is similar to the introduction of t/s diagnosable systems that generalize the t-diagnosable systems introduced by Preparata, Metze and Chien [12]. An introduction to t-diagnosable systems and their generalization may be found in [3], [4].

1.1 Definition of the *d*-Identifying Codes Problem

Consider an undirected graph $G(V, E)$ with each vertex $v \in V$ associated with an integer cost $c(v) > 0$ and an integer weight $w(v) > 0$.

Denote $N[v]$ to be the set of vertices containing v and all its neighbors. For a subset of vertices $S \subseteq V$, define the cost and weight of S as

$$c(S) = \sum_{v \in S} c(v) \text{ and } w(S) = \sum_{v \in S} w(v).$$

Two vertices $u, v \in V$ are distinguished by vertex w iff $|N[w] \cap \{u, v\}| = 1$. A set of vertices $D \subseteq V$ is called an identifying set if (1) every unordered vertex pair (u, v) is distinguished by some vertex in D and (2) D is a dominating set of G, i.e., each vertex in G is adjacent to at least one vertex in D (we will relax this requirement later).

Given $D \subseteq V$, define $I_D(v) = N[v] \cap D$ and an equivalence relation $u \equiv v$ iff $I_D(u) = I_D(v)$. The equivalence relation partition V into equivalence classes $V_D = \{S_1, S_2 \ldots, S_l\}$ such that $u, v \in S_i \iff I_D(u) = I_D(v)$.

For any $D \subseteq V$, we denote V_D to be the equivalence classes induced by D. If D is a dominating set of G and $d \geq \max\{w(S_1), w(S_2) \ldots, w(S_l)\}$, then D is called a d-identifying set of G. The d-identifying codes problem is to find a d-identifying set $D \subseteq V$ with minimum cost.

Note that if $d = 1$ then the d-identifying codes problem reduces to the identifying codes problem if the vertex costs and weights are equal to unity. Also, whereas the cost of the d-identifying set is a measure of the cost of installing alarms, the value of d is a measure of the degree of uncertainty in the identification of faulty processors. Since the value of d is also a measure of the expenses involved in testing each processor in an equivalence class, d has to be set at a small value.

The identifying set must be a dominating set. However we can drop this requirement after a simple transformation of the graph, i.e., adding a new isolated vertex with weight d and a very big cost such that any cost aware algorithm will not include this vertex in the solution set. Thus it will be the only vertex not

adjacent to the identifying set. So we will ignore the dominating set condition for the simplicity of presentation.

We denote $\ln x \equiv \log_e x$, $\lg x \equiv \log_2 x$.

1.2 Main Results

In this paper we introduce and study the d-identifying codes problem. We first propose an approximation algorithm inspired by a heuristic for the minimum probe selection problem [2] based on ideas from information theory. In Theorem 1, we establish the approximation ratio of our algorithm in terms of an entropy function $H(\cdot)$. As a byproduct of the analysis in Theorem 1, we derive in Corollary 1 a lower bound on the cost of the optimal solution. We then study the characteristics of the optimal entropy function that results in the approximation ratio of $1 + \ln d + \ln |V| + \ln(\lg |V|)$ for the d-identifying codes problem and of $1 + \ln |V| + \ln(\lg |V|)$ for the identifying codes problem in Theorem 1. We show that the approximation ratio of our algorithm is very close to the best possible for the d-identifying codes problem if $NP \notin DTIME(n^{\lg \lg n})$. We also derive several fundamental properties of the optimal solution using random graphs.

2 An Approximation Algorithm for the d-Identifying Codes Problem

2.1 A Greedy Algorithm

Our algorithm is presented as Algorithm 1. Following information theoretical terminology, $H(V_S)$ is called the entropy defined on V_S which is the set of equivalence classes induced by S. Similarly, $I(V_S; v) = H(V_S) - H(V_{S+v})$ is called the information content of $v \in V - S$ w.r.t. S. We defer the definition of the entropy until Sect. 2.2. Actually, the framework of our greedy algorithm without specific entropy definition is applicable to a class of identifying codes problems whose detailed specifications can be hidden in the definition of the entropy. Based on the framework of the greedy algorithm, one only needs to focus on the design of entropy for other variations of the identifying codes problem, e.g., the strong identification codes problem [11].

Algorithm 1 Greedy Algorithm

1: Initialize $D = \emptyset$
2: **while** $H(D) > 0$ **do**
3: Select vertex $v^* = \arg\max_{v \in V - D} I(V_D; v)/c(v)$
4: $D \leftarrow D \cup \{v^*\}$.
5: **end while**

The time complexity of the above greedy algorithm is $O(n^2 T_H(n))$, where T_H is the time complexity function of the algorithm computing $H(\cdot)$. The following theorem is the main result on the approximation ratio of the greedy algorithm.

Theorem 1. *Denote V_D as the set of equivalence classes induced by $D \subseteq V$. Suppose an entropy function $H(\cdot)$ satisfies the following conditions:*

(a) $H(V_D) = 0$ for any d-identifying set D,
(a) If $H(V_S) \neq 0$, then $H(V_S) \geq 1$, and
(c) $I(V_S; v) \geq I(V_{S+u}; v)$ for all $u \neq v, S \subseteq V$,

then the greedy algorithm returns a d-identifying set D such that $c(D)/c(D^*) < \ln[H(V_\emptyset)] + 1$, where $D^* = \{v_1^*, v_2^* \ldots, v_{|D^*|}^*\}$ is the minimum d-identifying set.

Proof. Suppose at the rth iteration, the greedy algorithm picks vertex v_r. Let D_r be the partial d-identifying set at the beginning of the rth iteration, $H_r = H(V_{D_r})$, and $D_r^* = D^* - D_r$. Note that $D_1 = \emptyset$ and $H_1 = H(V_\emptyset)$.

Since $D_r \cup D_r^*$ is a d-identifying set, $H(V_{D_r \cup D_r^*}) = 0$ by (a). Define $D_r^*(i) = \{v_1^*, v_2^* \ldots, v_i^*\}$, i.e., the first i values from D_r^*. Note that $D_r^*(0) = \emptyset$. We have

$$\begin{aligned} H(V_{D_r}) &= H(V_{D_r}) - H(V_{D_r \cup D_r^*}) \\ &= \sum_{i=0}^{|D_r^*|-1} [H(V_{D_r \cup D_r^*(i)}) - H(V_{D_r \cup D_r^*(i+1)})] = \sum_{i=0}^{|D_r^*|-1} I(V_{D_r \cup D_r^*(i)}; v_{i+1}^*). \end{aligned}$$

By (c), $I(V_{D_r \cup D_r^*(i)}; v_{i+1}^*) \leq I(V_{D_r \cup D_r^*(i-1)}; v_{i+1}^*) \cdots \leq I(V_{D_r}; v_{i+1}^*)$.
According to the greedy algorithm, $I(V_{D_r}; v_{i+1}^*)/c(v_{i+1}^*) \leq I(V_{D_r}; v_r)/c(v_r)$.
Hence

$$\begin{aligned} H_r = H(V_{D_r}) &= \sum_{i=0}^{|D_r^*|-1} I(V_{D_r \cup D_r^*(i)}; v_{i+1}^*) \\ &\leq \frac{c(D_r^*)}{c(v_r)} I(V_{D_r}; v_r) \leq \frac{c(D^*)}{c(v_r)} I(V_{D_r}; v_r). \end{aligned}$$

Then we know that $H_{r+1} = H_r - I(V_{D_r}; v_r) \leq (1 - \frac{c(v_r)}{c(D^*)}) H_r$.

Let the number of iterations of the greedy algorithm be $t = |D|$, where D is the solution returned by the greedy algorithm. We have

$$1 \leq H_t \leq \prod_{v \in D_t} (1 - \frac{c(v)}{c(D^*)}) H_0 \leq \exp\{-\frac{c(D_t)}{c(D^*)}\} H_0.$$

The first inequality holds because of (b) (note that $D_t = D - v_t$) and the last inequality is true because $1 - x \leq e^{-x}$.

On the other hand, by (a), $H(V_D) = 0$,

$$I(V_{D_t}; v_t) = H(V_{D_t}) \leq \frac{c(D^*)}{c(v_t)} I(V_{D_t}; v_t) \Rightarrow c(v_t) \leq c(D^*).$$

So $\frac{c(D)}{c(D^*)} = \frac{c(D_t)+c(v_t)}{c(D^*)} = \frac{c(D_t)}{c(D^*)} + \frac{c(v_t)}{c(D^*)} \leq \ln[H(V_\emptyset)] + 1$. □

Using a similar argument, we can derive a lower bound on the cost of the minimum d-identifying set if the costs of all the vertices are equal.

Corollary 1. *Let $G(V, E)$ be a graph with n vertices with equal cost which are labeled such that $I(V_\emptyset; v_1) \geq I(V_\emptyset; v_2) \cdots \geq I(V_\emptyset; v_n)$. Then the optimal cost of the minimum d-identifying set, $OPT_d(G) \geq K$, where K is the smallest integer such that $\sum_{i=1}^{K} I(V_\emptyset; v_i) \geq H(V_\emptyset)$.*

2.2 Optimal Entropy Function

Let $f_d(\cdot)$ be some non-negative function (to be specified later) and $H_d(V_D) = \sum_{S \in V_D} f_d(w(S))$ and $H_d(\emptyset) = 0$, where $V_D = \{S_1, S_2, ...\}$ is the set of equivalence classes induced by $D \subseteq V$.

We first examine Condition Theorem 1(c), i.e., $I(V_S; v) \geq I(V_{S+u}; v)$ for any $u \neq v$, $S \subseteq V$. In Fig. 1, there are two cases. In Case 1, v is adjacent to all the vertices in T_{uv} and T_v. In Case 2, v is only adjacent to vertices in T_{uv}, where T is an equivalence class in V_S; T_{uv}, T_u, T_v, and T_0 is the set of vertices in T adjacent to both u and v, only u, only v, and none of u, v, respectively. In other words, $T_{uv} \cup T_u$, $T_v \cup T_0 \in V_{S+u}$ and $T_{uv}, T_u, T_v, T_0 \in V_{S+u+v}$. Let $i = w(T_{uv}), j = w(T_u), k = w(T_v)$, and $l = w(T_0)$.

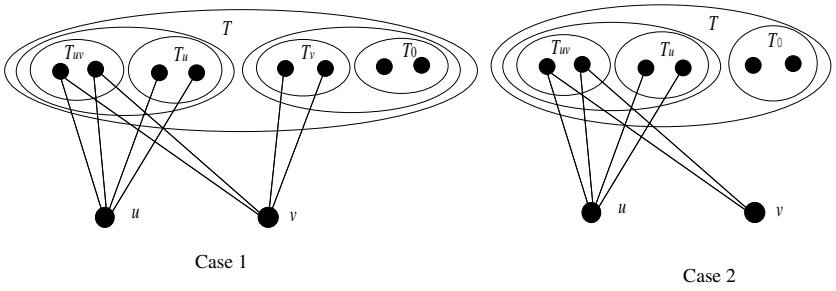

Fig. 1. Two cases. In Case 1, v is adjacent to all the vertices in T_{uv} and T_v. In Case 2, v is only adjacent to vertices in T_{uv}

It is easy to verify that the following conditions are necessary and sufficient for Theorem 1(a)-(c) to be true:

If $i, j, k, l \in \{0, 1, 2 \ldots\}$ and at most one of i, j, k, l is 0, then

$$f_d(i+j+k+l) - f_d(i+k) - f_d(j+l) \geq [f_d(i+j) - f_d(i) - f_d(j)] + [f_d(k+l) - f_d(k) - f_d(l)], \quad (1)$$

$$f_d(t) = 0, 0 \leq t \leq d, \text{ and} \quad (2)$$

$$f_d(t) \geq 1, \forall t \geq d+1. \quad (3)$$

Recall that the approximation ratio given in Theorem 1 is $\ln[H(V_\emptyset)] + 1 = \ln f_d(w(V)) + 1$. An entropy function is called optimal if it is the minimum

function among all functions that satisfy (1)-(3). Because the approximation ratio is $\ln f_d(w(V)) + 1$, we are only interested in the order of the function and ignore the constant coefficients and constant terms in the function. Assume that $w(V)$ is large. We next construct optimal entropy functions. We first consider $d = 1$. For this special case, define

$$f_1(n) = n \lg n. \tag{4}$$

Lemma 1. *$f_1(n)$ satisfies (1)-(3).*

Lemma 2. *Given $d \geq 2$, the function defined below satisfies (1)-(3).*

$$f_d(n) = \begin{cases} n \lg(n/d), n \geq d \\ \\ 0, otherwise. \end{cases} \tag{5}$$

Proof. Since $f_d(n)$ is a nondecreasing function and

$$f_d(d+1) = (d+1)\lg(1+\frac{1}{d}) = \lg((1+\frac{1}{d})^{d+1}) \geq \lg e > 1,$$

Condition (3) is true.

Condition (2) holds by definition of $f_d(n)$.

We next prove that Condition (1) holds.

If $i+j+k+l \leq d$, the proof is trivial. Without loss of generality, assume $i+j+k+l \geq d+1$. Consider 5 cases:

Case 1: $i, j, k, l \geq d$.

$$\begin{aligned} & f_d(i+j+k+l) + f_d(i) + f_d(j) + f_d(k) + f_d(l) \\ &= (i+j+k+l)\lg((i+k+j+l)/d) + i\lg(i/d) \\ &\quad + j\lg(j/d) + k\lg(k/d) + l\lg(l/d) \\ &= (i+j+k+l)\lg(i+j+k+l) + i\lg i + j\lg j \\ &\quad + k\lg k + l\lg l - 2(i+j+k+l)\lg d \\ &\geq (i+k)\lg(i+k) + (j+l)\lg(j+l) + (i+j)\lg(i+j) \\ &\quad + (k+l)\lg(k+l) - ((i+k)+(j+l)+(i+j)+(k+l))\lg d \\ &= f_d(i+k) + f_d(j+l) + f_d(i+j) + f_d(k+l) \end{aligned}$$

Case 2: Precisely one of $i+k, j+l, i+j$, and $k+l$ is $\leq d$.

Due to the symmetry of i, j, k, l in the function, assume that $i+k \leq d$. We have $i \leq d$ and $k \leq d$. Let $g(n) = n\lg(n/d)$.

Therefore

$$\begin{aligned} & f_d(i+j+k+l) + f_d(i) + f_d(j) + f_d(k) + f_d(l) \\ &\geq g(i+j+k+l) + g(i) + g(j) + g(k) + g(l) - g(i) - g(k) \\ &\geq g(j+l) + g(i+j) + g(k+l) + (g(i+k) - g(i) - g(k)) \\ &= f_d(j+l) + f_d(i+j) + f_d(k+l) + ((i+k)\lg\frac{i+k}{d} - i\lg\frac{i}{d} - k\lg\frac{k}{d}) \\ &\geq f_d(j+l) + f_d(i+j) + f_d(k+l) \end{aligned}$$

Case 3: Precisely $i + k \leq d$ and $j + l \leq d$ or $i + j \leq d$ and $k + l \leq d$.
Assume $i + k \leq d$ and $j + l \leq d$.
In this case, $i, j, k, l \leq d$. It suffices to show that

$$f_d(i + j + k + l) \geq f_d(i + j) + f_d(k + l).$$

This is obviously true.
Case 4: Precisely $i + k \leq d$ and $i + j \leq d$ (ignore those equivalent cases).
We have $i, j, k \leq d$. Hence

$$\begin{aligned}
&f_d(i + j + k + l) + f_d(i) + f_d(j) + f_d(k) + f_d(l) \\
&= f_d(i + j + k + l) + f_d(l) \\
&\geq (i + j + k + l) \lg((i + j + k + l)/d) + l \lg(l/d) \\
&= [(j + k + l) \lg(i + j + k + l) + l \lg l] + \\
&\quad [i \lg(i + j + k + l) - (i + j + k + l) \lg d - l \lg d] \\
&\geq [(j + l) \lg(j + l) + (k + l) \lg(k + l)] - [((j + l) \lg d + (k + l) \lg d)] \\
&= f_d(j + l) + f_d(k + l)
\end{aligned}$$

Case 5: All or 3 of the 4 terms, $i + k$,$j + l$, $i + j$, and $k + l$ are $\leq d$.
The proof for this case is trivial. □

Finally, we get the main results of this paper.

Theorem 2. *Using the entropy* $H_d(V_D) = \sum_{S \in V_D} f_d(w(S))$ *with* $f_d(\cdot)$ *as defined in (5), the greedy algorithm guarantees the approximation ratio of* $1 + \ln d + \ln(|V| \lg |V|)$.

Proof. Without loss of generality, assume $w(V) > d \geq max_{v \in V} w(v)$. We have $H_d(V_\emptyset) = f_d(w(V)) = w(V) \lg(w(V)/d) \leq d|V| \lg |V|$.
The rest of the proof follows from Theorem 1. □

Corollary 2. *For the identifying codes problem, our algorithm guarantees the approximation ratio of* $1 + \ln |V| + \ln(\lg |V|)$.

We next show that the function defined in (5) is optimal in asymptotic sense, i.e., the approximation ratio based on Theorem 1 cannot be improved by finding better entropy.

Lemma 3. $f(n) \geq \Theta(n \lg n)$.

2.3 Hardness of the *d*-Identifying Codes Problem

To study the hardness, i.e., the approximability of the d-identifying codes problem, we consider a subclass of d-identifying codes problem where the vertex costs and weights are 1. Since this subclass includes the identifying codes problem, d-identifying codes problem is at least as hard as identifying codes problem (here d is treated as a variable). On the other hand, an interesting question is whether

the approximability of the d-identifying codes problem changes with some fixed d. For example, if the best approximation ratio for the identifying codes problem is ϕ, one may ask whether the 2-identifying codes problem is $\phi/2$ approximable. The next lemma shows that the approximability will not change if d is a constant.

Lemma 4. *Assume that the identifying codes problem is feasible. For any fixed $d \geq 2$, if there exists a ϕ-approximation algorithm for the d-identifying codes problem, there also exists a ϕ-approximation algorithm for the identifying codes problem.*

Proof. We first give a ϕ-approximation algorithm for the identifying codes problem on $G(V, E)$ staring from the ϕ-approximation algorithm for the d-identifying codes problem:

1. Construct a graph $G'(V', E')$ defined as follows: Split each vertex $v \in V$ into d copies denoted as $v^d = \{v_1, v_2 \ldots, v_d\}$. For all $(u, v) \in E$, add edges to connect all vertices in u^d to all vertices in v^d and for all $v \in V$, add edges to join each pair of vertices in v^d(See Fig. 2). Formally,

$$V' = \bigcup_{v \in V} \{v_1, v_2 \ldots v_d\}, \text{ and}$$
$$E' = \{(u_i, v_j), i, j = 1, 2 \ldots, d | (u, v) \in E\} \bigcup \{(v_i, v_j), i, j = 1, 2 \ldots, d) | v \in V\}.$$

2. Apply the ϕ-approximation algorithm to get a d-identifying set D_d on G'.
3. Return $D = \{v \in V | v^d \cap D_d \neq \emptyset\}$ as an identifying set on G.

The construction of G' takes $O(d^2|E|)$ time with d as a constant.

We next show that the above procedure is a ϕ-approximation algorithm for the identifying codes problem.

Let D^* be an optimal solution to the identifying codes problem on G. It is easy to verify that $D'_d = \{v_1 | v \in D^*\}$ is a d-identifying set of G'. Denote D^*_d as an optimal d-identifying set of G'. We have

$$c(D^*_d) = |D^*_d| \leq |D'_d| = |D^*| = c(D^*).$$

Because in G', $\forall v \in V, v_1, v_2 \ldots, v_d \in V'$ has the same set of neighbors in G', there is no way to distinguish them in G'. Hence the set of equivalence classes of V' induced by D_d is simply $V'_{D_d} = \{v^d | v \in V\}$.

Observe that condensing all the vertices in v^d for all $v \in V$ into a single vertex v transforms G' back to G. So an identifying set of G can be formed by picking all the vertices whose corresponding set of vertices in G' contain at least one vertex in D_d. Hence the set D returned by the above procedure is an identifying set of G. Since the v^d's are pairwise disjoint,

$$c(D) = |D| \leq |D_d| = c(D_d) \leq \phi \cdot c(D^*_d) \leq \phi \cdot c(D^*).$$

□

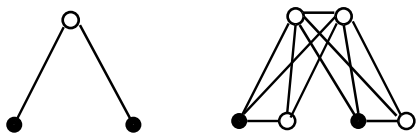

Fig. 2. Transformation from identifying codes problem to 2-identifying codes problem: The identifying set and d-identifying set consists of the solid vertices

Lemma 4 means that for any fixed d, the d-identifying codes problem is at least as hard as the identifying codes problem in term of approximability. Thus, with an application of the results in [6], we have the following theorem.

Theorem 3. *For any given $d \geq 1$, the d-identifying codes problem with unit vertex costs and weights is not approximable within $(1-\epsilon)\ln|V|$ unless $NP \in DTIME(n^{\lg\lg n})$.*

In view of Corollary 2, we can see that the approximation ratio of our algorithm is quite tight for the d-identifying codes problem where the vertex costs and weights are 1. Furthermore, we can expect that our approximation ratio is also very tight for general d-identifying cods problem as in the special case.

3 A Special Case with Unit Vertex Costs and Weights

In Sect. 2, we established the approximation ratio, i.e., the ratio of the cost of the approximation solution and the optimal cost of the d-identifying set. In this section we shall investigate the characteristics of the optimal solution itself.

Since it is difficult to study the d-identifying codes problem with arbitrary vertex costs and weights, we shall only consider a special class of d-identifying codes problem in which the cost and weight of each vertex is 1. In this setting, the cost and weight of a set of vertices is just the cardinality of the set.

We shall next investigate the impact of d on the cardinality of the resultant solution. Let $OPT_1(G)$ and $OPT_d(G)$ be the cardinality of the minimum identifying set and the minimum d-identifying set, respectively. We shall show that the value of $OPT_1(G)$ / $OPT_d(G)$ is unbounded.

Lemma 5. *Given $d \geq 2$ and $M > 0$, there exists a graph G such that*

$$OPT_1(G)/OPT_d(G) \geq M.$$

The graph constructed in Lemma 5 looks rather artificial. So let's consider the size of d-identifying sets on average basis. To study the average characteristics, assumptions are needed on the distribution of graphs. Given the vertices of G, the cardinality of the d-identifying set is totally decided by the edges. So we assume that for any unordered pair of vertices there is an edge with probability

p which is a constant. Notice that this is exactly the definition of a class of random graphs [1], [7].

For the sake of completeness, we first present Suen's inequality proved in [8], [14]. Let $A_1, A_2 \ldots, A_n$ be a set of events, and $X = \sum_{i=1}^{n} X_i$, where X_i is the indicator variable of event A_i($X_i = 1$ if event A_i occurs and $X_i = 0$ otherwise). We use $i \sim j$ to indicate that events A_i and A_j are dependent. Denote $\mu = \sum_{i=1}^{n} E[X_i]$, $\Delta = \sum_{i,j:i\sim j} E[X_i X_j]$, and $\delta = \max_{1\le i\le n} \sum_{j:j\sim i} E[X_j]$. Then $\Pr(X = 0) \le \exp\{-\mu + \Delta e^{2\delta}\}$.

Let $P \equiv p^{d+1} + (1-p)^{d+1}$, $Q(i) \equiv p^{2d+2-i} + (1-p)^{2d+2-i}$, $R^+(p,d) \equiv \ln(1 + (\frac{1-p}{p})^{d+1})/\ln(\frac{1}{p})$, and $R^-(p,d) \equiv R^+(1-p,d)$.

The following lemma is easy to prove.

Lemma 6. *$R^+(p,d)$ ($R^-(p,d)$) is a strictly decreasing (increasing) function of p and a decreasing (decreasing) function of d for $p \in [1/2, 1)$ ($p \in (0, 1/2]$).*

Given $\epsilon \in (0,1)$, denote $p_\epsilon^- \in (0, 1/2]$, $p_\epsilon^+ \in [1/2, 1)$ as two values such that $R^-(d, p_\epsilon^-) = R^+(d, p_\epsilon^+) = \epsilon$. Since $R^-(1/2, d) = R^+(1/2, d) = 1$, $p_\epsilon^- < 1/2 < p_\epsilon^+$. It can be shown that $p_\epsilon^- + p_\epsilon^+ = 1$ and $p_\epsilon^- \to 0$ ($p_\epsilon^+ \to 1$) if $\epsilon \to 0$.

Lemma 7. *If $1 \le i \le d, 0 < \epsilon < 1$, and $p \in [p_\epsilon^-, p_\epsilon^+]$, then*

$$(\frac{d+1-\epsilon}{d+1-i})(\frac{\ln Q(i)}{\ln P} - 1) > 1.$$

Theorem 4. *Given $0 < \epsilon < 1$, $\forall p \in [p_\epsilon^-, p_\epsilon^+]$, with high probability, there exists no d-identifying set of cardinality of $\frac{(d+1-\epsilon)\ln n}{\ln(1/P)}$ in $G(n,p)$ if n is sufficiently large.*

Proof. Let $c = \frac{(d+1-\epsilon)\ln n}{\ln(1/P)}$. it suffices to show that

$$\Pr(\text{There exists a } d\text{-identifying set of cardinality } c) = o(1) \to 0.$$

Consider a given set C of cardinality c. Let $S \subset V$ be a set of $d+1$ vertices, define event A_S: $\forall u, v \in S, I_C(u) = I_C(v)$. We can see that C is a d-identifying set iff no such event occurs for all S with $|S| = d+1$. Denote X_S to be the indicator variable for event A_S.

It can be seen that two events A_S and $A_{S'}$ are dependent iff $S \cap S' \neq \emptyset$.

Let $X = \sum_{S\subset V-C, |S|=d+1} X_S$.

Evidently, $\Pr(C \text{ is a } d\text{-identifying set}) \le \Pr(X = 0)$.

Assume $n - c - d - 1 \ge n/k$ for some small k (recall d is a constant). Then

$$\begin{aligned}
\mu &= \binom{n-c}{d+1}(p^{d+1} + (1-p)^{d+1})^c \ge (n-c-d-1)^{d+1} P^c \\
&\ge (n/k)^{d+1} P^c = \exp\{(d+1)\ln n - (d+1)\ln k + c\ln P\} \\
&= \exp\{(d+1)\ln n - (d+1)\ln k - (d+1-\epsilon)\ln n\} \\
&= \exp\{\epsilon \ln n - (d+1)\ln k\} = \Theta(n^\epsilon),
\end{aligned}$$

$$\Delta = \sum_{i=1}^{d} \binom{n-c}{2d+2-i} \binom{2d+2-i}{d+1} (p^{2d+2-i} + (1-p)^{2d+2-i})^c,$$

and

$$\delta = \sum_{i=1}^{d} \binom{d+1}{i} \binom{n-c-d-1}{d+1-i} P^c$$

Denote $\theta = \min_{1 \le i \le d}\{(\frac{d+1-\epsilon}{d+1-i})(\frac{\ln Q(i)}{\ln P} - 1)\} - 1$. By Lemma 7, $\theta > 0$. We have

$$\begin{aligned}
\frac{\Delta}{\mu} &\le \sum_{i=1}^{d} \frac{\binom{n-c}{2d+2-i}\binom{2d+2-i}{d+1}}{\binom{n-c}{d+1}} (\frac{Q(i)}{P})^c = \sum_{i=1}^{d} \binom{n-c-d-1}{d+1-i} (\frac{Q(i)}{P})^c \\
&\le \sum_{i=1}^{d} (\frac{ne}{d+1-i})^{d+1-i} (\frac{Q(i)}{P})^c \\
&\le \sum_{i=1}^{d} \exp\{(d+1-i)(1 - \frac{(d+1-\epsilon)}{(d+1-i)}(\frac{\ln Q(i)}{\ln P} - 1)) \ln n + d\} \\
&\le \sum_{i=1}^{d} \exp\{-\theta(d+1-i)\ln n + d\} \le d \exp\{-\theta \ln n + d\} = \Theta(n^{-\theta}) = o(1).
\end{aligned}$$

Similarly, we can show that $e^{2\delta} = O(\exp\{n^{\epsilon-1}\}) \to 1$.
Hence $-\mu + \Delta e^{2\delta} = -\mu(1 - \frac{\Delta}{\mu} e^{2\delta}) \le \Theta(-n^{\theta})$ and
Pr(There exists a d-identifying code of cardinality c) $\le \binom{n}{c} \exp\{\Theta(-n^{\theta})\}$
Since $\binom{n}{c} \exp\{\Theta(-n^{\theta})\} = O(\exp\{\Theta(\ln^2 n - n^{\theta})\}$,

$$\Pr(\text{There exists a } d\text{-identifying set of cardinality } c) = o(1) \to 0.$$

□

Theorem 5. *For a set of vertices $C \subseteq V$ and $|C| = \frac{(d+1+\epsilon)\ln n}{\ln(1/P)}$,*

$$\lim_{n\to\infty} \Pr(C \textit{ is a } d\textit{-identifying set }) = 1.$$

By Theorem 4 and Theorem 5, with high probability, the cardinality of minimum d-identifying set is approximately $(d+1)\ln n / \ln(1/P)$ when n is sufficiently large.

4 summary

In this paper we introduced and studied the d-identifying codes problem that generalizes the identifying codes problem studied in [9]. This problem is of great

theoretical and practical interest in several applications, in particular, fault diagnosis in multiprocessor systems and placement of alarms for robust identification of faulty components in sensor networks. The value of d associated with the identifying set is a measure of the degree of uncertainty in the identification of faulty processors. We presented an approximation algorithm and established its approximation ratio. This algorithm is a generalization of the heuristic presented in [2] but without analysis of the approximation ratio. Our analysis also provides a way to compute a lower bound on the cost of the optimum solution. We also established certain hardness results in terms of approximability of the d-identifying codes problem.

We performed a probabilistic analysis on random graphs assuming that vertex costs and weights are all equal. We established that a d-identifying set of certain cardinality exists with very high probability. We also showed that a d-identifying set of cardinality smaller than this number does not exist with a high probability.

Further investigation of the identifying codes problem on special topologies such as hypercubes is in progress.

References

1. Béla Bollobás. *Random graphs.* Academic Press, Inc, London, 1985.
2. M. Brodie, I. Rish, and S. Ma. Optimizing probe selection for fault localization. In *DSOP*, 2001.
3. A. Das, K. Thulasiraman, and V. Agarwal. Diagnosis of t/s-diagnosable systems. *Journal of Circuits, Systems and Computers*, 1:353–371, 1991.
4. A. Das, K. Thulasiraman, and V. Agarwal. Diagnosis of t/(t+1)-diagnosable systems. *SIAM J. Comput.*, 23(5):895–905, 1994.
5. A. Frieze, R. Martin, J. Moncel, and K. Ruszinkóand C. Smyth. Codes identifying sets of vertices in random networks. submitted for publication, 2005.
6. Bjarni V. Halldórsson, Magnús M. Halldórsson, and R. Ravi. On the approximability of the minimum test collection problem. In *ESA*, pages 158–169, 2001.
7. S. Janson, T. Luczak, and A. Rucinski. *Random graphs.* Wiley, New York, 2000.
8. Svante Janson. New versions of suen's correlation inequality. *Random Struct. Algorithms*, 13(3-4):467–483, 1998.
9. M. Karpovsky, K. Chakrabarty, and L. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE Trans. on Information Theory*, 44(2):599–611, 1998.
10. M. Laifenfeld and A. Trachtenberg. Disjoint identifying-codes for arbitrary graphs. submitted to IEEE Symposium on Information Theory, 2005.
11. Tero Laihonen. Optimal codes for strong identification. *Eur. J. Comb.*, 23(3):307–313, 2002.
12. F. Preparata, G. Metze, and R. Chien. On the connection assignment problem of diagnosiable systems". *IEEE Trans. on Electronic Computers*, 16:848–854, 1967.
13. S. Ray, R. Ungrangsi, F. Pellegrini, A. Trachtenberg, and D. Starobinski. Robust location detection in emergency sensor networks. In *INFOCOM*, 2003.
14. Stephen Suen. A correlation inequality and a poisson limit theorem for nonoverlapping balanced subgraphs of a random graph. *Random Struct. Algorithms*, 1(2):231–242, 1990.

Appendix: Omitted Proofs

Lemma 1. $f_1(n)$ satisfies (1)-(3).

Proof. Conditions (2)-(3) are trivial. We only show the proof to Condition (1).
Let $i, j, k, l \geq 0$ and at most one of them be 0. Consider 2 cases.
Case 1: $i, j, k, l > 0$. It suffices to show that

$$\begin{aligned}&(i+j+k+l)\lg(i+j+k+l) - (i+k)\lg(i+k) - (j+l)\lg(j+l)\\ &\quad \geq [(i+j)\lg(i+j) - i\lg i - j\lg j] + [(k+l)\lg(k+l) - k\lg k - l\lg l].\end{aligned}$$

Equivalently, we will prove that

$$\lg(\frac{(i+j+k+l)^{(i+j+k+l)} i^i j^j k^k l^l}{(i+k)^{(i+k)}(j+l)^{(j+l)}(i+j)^{(i+j)}(k+l)^{(k+l)}}) \geq 0.$$

Define function

$$g(x) = \ln(\frac{(x+j+k+l)^{x+j+k+l} x^x j^j k^k l^l}{(x+k)^{x+k}(j+l)^{j+l}(x+j)^{x+j}(k+l)^{k+l}}).$$

It suffices to show that $\forall x > 0$, $g(x) \geq 0$. We have

$$\begin{aligned}&g'(x_0) = \ln\frac{x_0(x_0+j+k+l)}{(x_0+k)(x_0+j)} = 0 \Leftrightarrow x_0 = kj/l > 0.\\ &g''(x_0) = l/[x_0(x_0+j+k+l)] > 0.\end{aligned}$$

Since

$$\begin{aligned}&\frac{(x+j+k+l)^{x+j+k+l} x^x j^j k^k l^l}{(x+k)^{x+k}(j+l)^{j+l}(x+j)^{x+j}(k+l)^{k+l}}\\ &= (1 + \frac{xl-jk}{(x+k)(x+j)})^x (1 - \frac{xl-jk}{(j+l)(x+j)})^j\\ &\times (1 - \frac{xl-jk}{(x+k)(k+l)})^k (1 + \frac{xl-jk}{(j+l)(k+l)})^l,\end{aligned}$$

$g(x_0) = \ln 1 = 0$ and hence $\forall x > 0, g(x) \geq g(x_0) = 0$.
Case 2: Precisely one of i, j, k, l is 0.
Without loss of generality, assume $l = 0$. It suffices to show that

$$\forall x \geq 0, h(x) = \ln(\frac{(i+j+x)^{i+j+x} i^i}{(i+x)^{i+x}(i+j)^{i+j}}) \geq 0.$$

Since $h'(x) = \ln(\frac{i+j+x}{i+x}) \geq 0$ and $h(0) = 0$, $\forall x \geq 0, h(x) \geq h(0) = 0$. □

Lemma 3. $f(n) \geq \Theta(n \lg n)$.

Proof. Set $i = j = k = l = 2^i d$ in (1), we get

$$f_d(2^{i+2}d) - 2f_d(2^{i+1}d) \geq 2(f_d(2^{i+1}d) - 2f_d(2^i d)). \tag{6}$$

Solving the recurrence inequalities on i, we get

$$f_d(2^{i+2}d) \geq 2f_d(2^{i+1}d) + 2^i f(2d). \tag{7}$$

Hence, $f_d(2^i d) \geq i2^{i-1}f(2d)$. Letting $n = 2^i d$ completes the proof. □

Lemma 5. Given $d \geq 2$ and $M > 0$, there exists a graph G such that

$$OPT_1(G)/OPT_d(G) \geq M.$$

Proof. Given $d \geq 2$ and $M > 1$, we construct a graph G using vertices in Fig. 3.

In Fig. 3, there are k equal sized sets of disjoint vertices $L_1, L_2 \ldots, L_k$, with each set being called a super node. Let $L = L_1 \cup L_2 \cdots \cup L_k$, $|L_1| = |L_2| \cdots = |L_k| = d$, $S = \{s_1, s_2 \ldots, s_l\}$, where $l = \lceil \lg(k+1) \rceil$, i.e., $|S| = \lceil \lg(k+1) \rceil$.

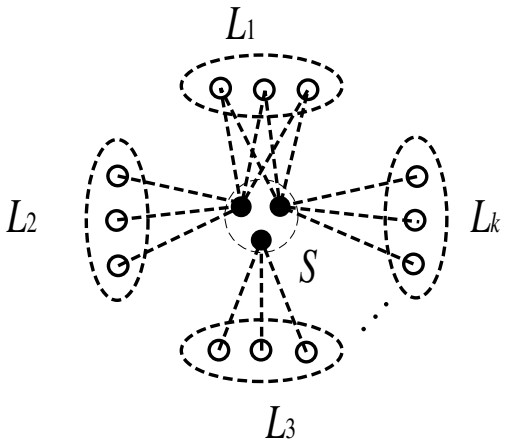

Fig. 3. The vertex set of the component B constructed in the proof of Lemma 5: $L_i's$ are equal sized sets with each containing d vertices. S is a set of $\lceil \lg(k+1) \rceil$ vertices which distinguish vertices from different $L_i's$.

Add edges between vertices in L and S such that:

a) For $i = 1, 2 \ldots, k, j = 1, 2 \ldots, l, s_j$ is either adjacent to all the vertices in L_i or none of the vertices in L_i.

b) $\forall i \neq j$, vertices in L_i and L_j are distinguished by vertices in S (This is possible because $|S| = \lceil \lg(k+1) \rceil$).

Since all vertices contained in a super node have the same neighbors except for themselves, at least $d-1$ of them should be included in any identifying set to distinguish the d vertices in the same super node. So $OPT_1(G) \geq (d-1)k$. Obviously, $OPT_d(G) = |S| = \lceil \lg(k+1) \rceil$.

So $OPT_1(G)/OPT_d(G) \geq M$ if k is large enough. □

Lemma 7. If $1 \leq i \leq d, 0 < \epsilon < 1$, and $p \in [p_\epsilon^-, p_\epsilon^+]$, then

$$(\frac{d+1-\epsilon}{d+1-i})(\frac{\ln Q(i)}{\ln P} - 1) > 1.$$

Proof. If $p = 1/2$, the proof is trivial. We now consider 2 cases.

Case 1: $1/2 < p \leq p_\epsilon$.

By Lemma 6, $\epsilon \leq R^+(p,d) \leq 1$ and $\forall 1 \leq i \leq d, R^+(p,d) > R^+(p, 2d+1-i)$. So

$$\begin{aligned}
\frac{\ln Q(i)}{\ln P} - 1 &= \frac{\ln(p^{2d+2-i} + (1-p)^{2d+2-i})}{\ln(p^{d+1} + (1-p)^{d+1})} - 1 \\
&= \frac{(2d+2-i) - R^+(p, 2d+1-i)}{(d+1) - R^+(p,d)} - 1 \\
&= \frac{(d+1-i) + R^+(p,d) - R^+(p, 2d+1-i)}{(d+1) - R^+(p,d)} \\
&\geq \frac{d+1-i}{d+1-\epsilon} + \frac{R^+(p,d) - R^+(p, 2d+1-i)}{d+1-\epsilon}.
\end{aligned}$$

Therefore,

$$(\frac{d+1-\epsilon}{d+1-i})(\frac{\ln Q(i)}{\ln P} - 1) \geq 1 + \frac{R^+(p,d) - R^+(p,2d+1-i)}{d+1-i} > 1.$$

Case 2: $p_\epsilon \leq p < 1/2$.

$$\frac{\ln Q(i)}{\ln P} - 1 = \frac{(d+1-i) + R^-(p,d) - R^-(p, 2d+1-i)}{(d+1) - R^-(p,d)}.$$

The rest of the proof is the same as in Case 1. □

Theorem 5. For a set of vertices $C \subseteq V$ and $|C| = \frac{(d+1+\epsilon)\ln n}{\ln(1/P)}$,

$$\lim_{n \to \infty} \Pr(C \text{ is a } d\text{-identifying set }) = 1.$$

Proof. Let $X = \sum_{S \subset V, |S|=d+1} X_S$, where X_S is defined as in the proof of Theorem 4. By Markov's inequality, we have,

$$\Pr(C \text{ is a } d\text{-identifying set}) = \Pr(X = 0) = 1 - \Pr(X \geq 1) \geq 1 - E(X).$$

$$\begin{aligned} E(X) &= \sum_{S} E(X_S) = \sum_{i=0}^{d+1} \sum_{|S \cap C| = i} E(X_S) \\ &= \sum_{i=0}^{d+1} \binom{|C|}{i} \binom{n - |C|}{d+1-i} P^{|C|-i} (p^{d+1-i})^i p^{i(i-1)/2} \\ &\leq n^{d+1} P^{|C|-d-1} \leq \exp\{-\epsilon \ln n + (d+1) \ln(1/P)\} \to 0. \end{aligned}$$

□