



**HAL**  
open science

# On graph identification problems and the special case of identifying vertices using paths

Florent Foucaud, Matjaž Kovše

► **To cite this version:**

Florent Foucaud, Matjaž Kovše. On graph identification problems and the special case of identifying vertices using paths. International Workshop on Combinatorial Algorithms (IWOCA), Jul 2012, Krishnankoil, India. pp.32-45, 10.1007/978-3-642-35926-2\_4 . hal-00960561

**HAL Id: hal-00960561**

**<https://hal.science/hal-00960561v1>**

Submitted on 29 Dec 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On graph identification problems and the special case of identifying vertices using paths<sup>\*</sup>

Florent Foucaud<sup>1</sup> and Matjaž Kovše<sup>2,3</sup>

<sup>1</sup> Univ. Bordeaux, LaBRI, UMR5800, F-33400 Talence, France.  
CNRS, LaBRI, UMR5800, F-33400 Talence, France.

`florent.foucaud@gmail.com`

<sup>2</sup> Faculty of Natural Sciences and Mathematics, University of Maribor,  
SI-2000 Maribor, Slovenia.

<sup>3</sup> Bioinformatics Group, Department of Computer Science and Interdisciplinary  
Center for Bioinformatics, Univ. of Leipzig, Härtelstrasse 16-18, D-04107 Leipzig,  
Germany.

`matjaz.kovse@gmail.com`

**Abstract.** In this paper, we introduce the identifying path cover problem: an *identifying path cover* of a graph  $G$  is a set  $\mathcal{P}$  of paths such that each vertex belongs to a path of  $\mathcal{P}$ , and for each pair  $u, v$  of vertices, there is a path of  $\mathcal{P}$  which includes exactly one of  $u, v$ . This problem is related to a large variety of identification problems. We investigate the identifying path cover problem in some families of graphs. In particular, we derive the optimal size of an identifying path cover for paths, cycles, hypercubes and topologically irreducible trees and give an upper bound for all trees. We give lower and upper bounds on the minimum size of an identifying path cover for general graphs, and discuss their tightness. In particular, we show that any connected graph  $G$  has an identifying path cover of size at most  $\left\lceil \frac{2(|V(G)|-1)}{3} \right\rceil$ . We also study the computational complexity of the associated optimization problem, in particular we show that when the length of the paths is asked to be of a fixed value, the problem is APX-complete.

**Keywords:** Test cover, Identification, Paths, Approximation.

## 1 Introduction

This paper aims to study the new optimization problem of identifying the vertices of a graph by means of paths, which we call the *identifying path cover problem*. We first relate this problem to a large number of other problems and review a part of the associated literature, before giving its definition.

---

<sup>\*</sup> This research is supported by the ANR Project IDEA - Identifying coDes in Evolving grAphs, ANR-08-EMER-007, 2009-2012.

### 1.1 On test covers and the identification problem

Identification problems have been addressed many times in the last decades under different denominations and in different contexts. We present two general problems from the literature which have almost the same definition, and which we herein call the minimum test cover problem and the minimum identification problem. Instances of these problems are set systems, i.e. pairs consisting of a set  $\mathcal{I}$  of elements (“individuals”) and a set  $\mathcal{A}$  of subsets of  $\mathcal{I}$  (“attributes”).

Among these two problems, the *minimum test cover problem*, in short MIN-TC, seems to have been studied first and is probably better known. Given a set system of individuals and attributes, the MIN-TC problem asks for a minimum subset  $\mathcal{C}$  of  $\mathcal{A}$  such that for each pair  $I, I'$  of  $\mathcal{I}$ , there is an element  $C$  of  $\mathcal{C}$  such that exactly one of  $I, I'$  is *covered* by  $C$ , that is, belongs to  $C$  (we say that  $C$  *separates*  $I$  from  $I'$ ). The MIN-TC problem appears in a large number of papers under different denominations (*minimum test cover problem* [8], *minimum test collection problem* [13], *minimum test set problem* [18]). In fact, a well-celebrated theorem of J. A. Bondy on *induced subsets* [3] can be seen as the first study of this problem.

In this paper and as in a large portion of the literature dealing with special cases of this kind of problems, we are interested in a slight modification of MIN-TC, where not only each pair of individuals has to be separated, but also, each individual has to be covered. We call this problem the *minimum identification problem*, MIN-ID for short (note that it has been studied under the denomination of *discriminating code problem* in [4], but we use our terminology in order to fit to special cases described later). MIN-TC and MIN-ID are very close to each other, since for any solution to one of them, there is a solution to the other one whose size differs by at most 1: any solution to MIN-ID is also one for MIN-TC, and, given a solution  $\mathcal{C}$  to MIN-TC which is not a valid solution to MIN-ID, at most one individual  $I$  may not be covered by  $\mathcal{C}$ . It is then sufficient to add an arbitrary attribute  $A$  covering  $I$  to  $\mathcal{C}$  to get a valid solution to MIN-ID.

Both MIN-TC and MIN-ID can be seen as special cases of the well-known *minimum set cover problem* [13,15], MIN-SC for short, where, given a base set  $\mathcal{X}$  and a set  $\mathcal{S}$  of subsets of  $\mathcal{X}$ , it is asked to find a minimum subset  $\mathcal{C}$  of  $\mathcal{S}$  covering all elements of  $\mathcal{X}$  [8]. MIN-TC and MIN-ID enjoy the same computational complexity. It is known that both problems are  $O(\ln(|\mathcal{I}|))$ -approximable (where  $\mathcal{I}$  denotes the set of individuals of the input) using a reduction to MIN-SC [18]. On the other hand, both problems are not only NP-hard [4,13] but have also been shown to be NP-hard to approximate within a factor of  $o(\ln(|\mathcal{I}|))$  by reduction from MIN-SC [2,8].

A natural restriction of MIN-ID is, given some integer  $k$ , the one where the sets of  $\mathcal{A}$  all have exactly  $k$  elements. We will call this problem MIN-ID- $k$ .

### 1.2 Related problems

In this paper, we study a special case of MIN-ID. Just as some particular cases of MIN-SC arising from specific structures have gained a lot of interest (consider for

example all variants of the minimum dominating set problem, or the minimum vertex cover problem), it is of interest to investigate special cases of the MIN-ID problem having a particular structure. In this line of research, many specific cases arising from graph theory are of particular interest since graphs model networks of all kinds and are found in real world applications. For example, in the *identifying code problem* [9,12,16], one wants to identify each vertex  $v$  using vertices at distance at most 1 from  $v$ . This problem can be seen as MIN-ID where  $\mathcal{I} = V(G)$  and  $\mathcal{A}$  is the family of the balls around each vertex. This problem has been generalized to digraphs [6,11], and to the case where also *sets* of at most  $\ell$  vertices are to be separated and where vertices can identify at some prescribed distance  $r \geq 1$  [12]. One may also ask to identify the edges of  $G$  using edges, i.e.  $\mathcal{I} = E(G)$  and  $\mathcal{A}$  is the set of all edge-balls around each edge of  $G$  [10]. Rather than considering full balls, also partial balls may be considered, as in the case of *watching systems* [1], where  $\mathcal{I} = V(G)$ , and  $\mathcal{A}$  is the family of all stars in  $G$ . Finally, the case where  $\mathcal{I} = V(G)$  and  $\mathcal{A}$  is the set of all cycles in  $G$  has been considered in [14,20].

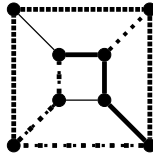
### 1.3 The identifying path cover problem

In this paper, we study MIN-ID when  $\mathcal{I} = V(G)$  and  $\mathcal{A}$  is the set of all paths of  $G$ . This problem was first mentioned in a discussion between the first author, J. L. Sewell and P. J. Slater. We call it *minimum identifying path cover problem*, MIN-IDPC for short and it studies the following notion:

**Definition 1.** *Given a graph  $G$ , a set  $\mathcal{P}$  of paths of  $G$  is an identifying path cover if each vertex of  $G$  belongs to a path of  $\mathcal{P}$  (it is covered) and if for each pair  $u, v$  of vertices, there is a path of  $\mathcal{P}$  which contains exactly one of  $u, v$  ( $u, v$  are separated).*

We point out that the covering condition is not implied by the separation condition, since even when all pairs are separated, one vertex of the graph may remain uncovered. We denote by  $p^{\text{ID}}(G)$  the minimum number of paths required in any identifying path cover of  $G$ . Then, MIN-IDPC is the problem, given a graph  $G$ , of determining the value of  $p^{\text{ID}}(G)$ . An example of an identifying path cover  $\mathcal{P}$  of the cube  $H_3$  is given in Fig. 1, where the four thick paths belong to  $\mathcal{P}$  (the full, the densely dotted, the loosely dotted and the dashed-dotted path). Note that an identifying path cover of  $G$  always exists: consider the set of all 0-paths of  $G$ , that is,  $\mathcal{P} = V(G)$ .

Given an integer  $k \geq 1$ , we will also discuss the natural variant MIN-IDPC- $k$  of MIN-IDPC, where one wants to find a minimum *identifying  $k$ -path cover* of  $G$ , that is, a set of paths of exactly  $k$  vertices forming an identifying path cover of  $G$ . We denote by  $p_k^{\text{ID}}(G)$  the size of a minimum identifying  $k$ -path cover of  $G$ . Unlike for the general MIN-IDPC problem, not all graphs admit an identifying  $k$ -path cover. We call a graph admitting an identifying  $k$ -path cover,  *$k$ -path identifiable*. This is the case if, first of all, each vertex of  $G$  lies on a  $k$ -path, and if for each pair  $u, v$  of vertices, there is a  $k$ -path covering exactly one of  $u, v$ . For example, the



**Fig. 1.** An (optimal) identifying path cover of the hypercube  $H_3$ :  $p^{\text{ID}}(H_3) = 4$ .

path graphs  $P_{k-1}$  and  $P_{2k-2}$  are not  $k$ -path identifiable. Observe that these two conditions are also sufficient: if both are fulfilled, taking all  $k$ -paths of  $G$  gives a valid identifying  $k$ -path cover of  $G$ . Being  $k$ -path identifiable is polynomial-time checkable since there are at most  $\binom{n}{k} = O(n^k)$   $k$ -paths in  $G$ .

#### 1.4 Applications

Problems MIN-TC and MIN-ID have a broad variety of applications, for example in the diagnosis of faults or diseases, biological identification, pattern recognition [8,18]. When the instance of the problem arises from substructures of a graph, the main applications are routing in networks [17] and the location of threats in facilities or networks using sensors [16]: vertices are the “individuals”, sensors are the “attributes”. Sensors may monitor closed neighbourhoods (identifying codes) or sub-neighbourhoods (watching systems). If sensors are capable of monitoring the vertices lying on a path, we have the situation of an identifying path cover. One can for example imagine sensors in the form of laser detectors, or mobile detecting devices patrolling back and forth along their path.

#### 1.5 Outline of the paper

We start by giving some preliminary results in Section 2, in the form of bounds from the literature valid for the general MIN-ID problem (which we apply to MIN-IDPC) and some observations valid only for MIN-IDPC. We continue by studying MIN-IDPC in some basic families of graphs in Sections 3 and 4: we give exact values for parameter  $p^{\text{ID}}$  in paths, cycles, topologically irreducible trees, and an upper bound for trees in general. We use the latter to provide the upper bound  $p^{\text{ID}}(G) \leq \lceil \frac{2n}{3} \rceil$  for any connected graph. Finally, we show in Section 5 that MIN-IDPC- $k$  is APX-complete for any  $k \geq 3$  by means of an L-reduction from the minimum vertex cover problem. We conclude with some open questions in Section 6.

## 2 Preliminary observations

The following lower bound was observed in [16] in the context of identifying codes but we refer to [4] for the general statement.

**Theorem 2 ([4,16]).** *Let  $(\mathcal{I}, \mathcal{A})$  be an instance of MIN-ID, and let  $\mathcal{C}$  be a solution to it. Then  $|\mathcal{C}| \geq \log_2(|\mathcal{I}| + 1)$ .*

The following upper bound can be seen as a direct corollary of Bondy's theorem [3]. We refer to [4] for a formal proof in this context.

**Theorem 3 ([3,4]).** *Let  $(\mathcal{I}, \mathcal{A})$  be an instance of MIN-ID, and let  $\mathcal{C}$  be an inclusionwise minimal solution to it. Then  $|\mathcal{C}| \leq |\mathcal{I}|$ .*

Consider an instance of MIN-ID- $k$ . Then, another lower bound holds. This bound was (to our knowledge) first observed in the context of identifying codes in [16], but the proof works in the more general context of MIN-ID- $k$ .

**Theorem 4 ([16]).** *Let  $k \geq 1$  and  $(\mathcal{I}, \mathcal{A})$  be an instance of MIN-ID- $k$ . Then for any solution  $\mathcal{C}$ ,  $|\mathcal{C}| \geq \frac{2|\mathcal{I}|}{k+1}$ .*

Applying theorems 2, 3 and 4 to MIN-IDPC and MIN-IDPC- $k$ , we get:

**Theorem 5.** *Let  $G$  be a graph on  $n$  vertices and  $k \geq 1$  an integer. Then  $\log_2(n+1) \leq p^{\text{ID}}(G) \leq n$  and  $\max\{\log_2(n+1), \frac{2n}{k+1}\} \leq p_k^{\text{ID}}(G) \leq n$ .*

It is easily observed that in the complete graph  $K_n$ , since we have full freedom to choose the paths in the identifying path cover,  $p^{\text{ID}}(K_n) = \lceil \log_2(n+1) \rceil$ . In fact, much sparser graphs also fulfill this bound, such as the hypercubes: one can easily come up with a solution with  $\lceil \log_2(n+1) \rceil$  paths. A similar problem of identification using cycles is addressed in [14,20]; we refer to these papers for the construction. Since removing an edge from a cycle yields a path, their construction is also valid in our case:

**Theorem 6 ([14,20]).** *Let  $H_d$  be the hypercube of dimension  $d$  with  $n = 2^d$  vertices. Then  $p^{\text{ID}}(H_d) = \lceil \log_2(n+1) \rceil$ .*

One can easily see that the bound  $p_k^{\text{ID}}(G) \geq \frac{2n}{k+1}$  is tight; given two integers  $k \geq 1$  and  $p \geq k$ , one can construct a graph  $G$  with  $p_k^{\text{ID}}(G) = p$ . To do so, one has to take care that for each of the  $p$  paths of the solution, there is a vertex that belongs only to this path. All other vertices must belong to distinct sets of exactly two paths of the solution.

Since the set of paths of a graph  $G$  is a superset of the set of paths of a subgraph  $H$  of  $G$ , if  $H$  is spanning the vertices of  $G$ , any identifying path cover of  $H$  will also be one for  $G$ . We get the following proposition:

**Proposition 7.** *Let  $G$  be a graph and  $H$  a spanning subgraph of  $G$ . Then  $p^{\text{ID}}(G) \leq p^{\text{ID}}(H)$ .*

The following proposition will be useful. The bound will be shown to be tight for the star (see Thm. 12).

**Proposition 8.** *If  $G$  is a graph having  $l$  vertices of degree 1,  $p^{\text{ID}}(G) \geq \lceil \frac{2l}{3} \rceil$ .*

*Proof.* A vertex  $v$  of degree 1 can only be covered by a path  $P$  if  $v$  is an endpoint of  $P$ , and two vertices of degree 1 cannot be covered only by the same path (otherwise they are not separated from each other). If a degree 1 vertex is the endpoint of  $k$  paths, then these  $k$  paths can cover at most  $k+1$  degree 1 vertices. Hence, the minimum is reached when  $k = 2$  when two degree 1 vertices are identified with three paths.  $\square$

### 3 Identifying path covers of paths and cycles

We first investigate identifying path covers in simple graphs such as paths and cycles. The path and the cycle on  $n$  vertices are denoted  $P_n$  and  $C_n$ , respectively. We start with a lower bounds for these graphs.

**Proposition 9.** *Let  $G$  be a connected graph of maximum degree 2 having  $m$  edges and  $l$  vertices of degree one. Then  $p^{ID}(G) \geq \lceil \frac{m+l}{2} \rceil$ .*

*Proof.* Let  $u, v$  be two adjacent vertices of  $G$ . In any identifying path cover  $\mathcal{P}$  of  $G$ , there must be a path  $P$  that either ends in  $u$  and does not contain  $v$ , or ends in  $v$  and does not contain  $u$  (let us say that  $P$  cuts the edge  $uv$ ). Moreover, for any vertex of degree 1, there is a path of  $\mathcal{P}$  that ends in it. Since one single path can at most cut or cover two edges/degree 1 vertices, the result follows.  $\square$

**Theorem 10.** *For any  $n \geq 1$ ,  $p^{ID}(P_n) = \lceil \frac{n+1}{2} \rceil$ .*

*Proof.* The lower bound comes from Prop. 9. For the upper bound, let  $V(P_n) = \{v_0, \dots, v_{n-1}\}$  and  $\mathcal{P} = \{v_i \dots v_{i+\lceil \frac{n}{2} \rceil} \mid i \in \{0, \dots, \lceil \frac{n}{2} \rceil - 1\}\}$  be a set of  $\lceil \frac{n}{2} \rceil$  paths. If  $n$  is odd,  $\mathcal{P}$  is an identifying path cover of cardinality  $\lceil \frac{n}{2} \rceil = \lceil \frac{n+1}{2} \rceil$ . If  $n$  is even,  $\mathcal{P}$  separates all pairs of vertices, and covers all vertices but  $v_{n-1}$ . Hence,  $\mathcal{P} \cup \{v_0, \dots, v_{n-1}\}$  is an identifying path cover of cardinality  $\lceil \frac{n+1}{2} \rceil$ .  $\square$

**Theorem 11.** *It holds that  $p^{ID}(C_3) = 2$ ,  $p^{ID}(C_4) = 3$  and for any  $n \geq 5$ ,  $p^{ID}(C_n) = \lceil \frac{n}{2} \rceil$ .*

*Proof.* For  $n \neq 4$ , the lower bounds come from Prop. 9, and from Thm. 2 for  $n = 4$ . We give constructions for the upper bounds. Let  $V(C_n) = \{v_0, \dots, v_{n-1}\}$ . One can check that  $\{v_0v_1, v_1v_2\}$  and  $\{v_0v_1, v_1v_2, v_2v_3\}$  are valid identifying path covers of  $C_3$  and  $C_4$ . For  $n \geq 5$ , let  $\mathcal{P} = \{v_i v_{i+1} v_{i+2} \mid i \text{ even}, i < n-1\}$ . If  $n$  is even,  $\mathcal{P}$  is a identifying path cover of  $C_n$  of cardinality  $\lceil \frac{n}{2} \rceil$ . Otherwise, the pairs  $v_0, v_1$  and  $v_{n-2}, v_{n-1}$  are covered but not separated. Then  $\mathcal{P} \cup \{v_{n-1}v_0\}$  is an identifying path cover of  $C_n$  of cardinality  $\lceil \frac{n}{2} \rceil$ .  $\square$

### 4 The case of trees with an application to all graphs

We start by giving the value of parameter  $p^{ID}$  for the star on  $n$  vertices, denoted  $K_{1, n-1}$ . The provided construction and bound will prove useful in what follows.

**Theorem 12.** *It holds that  $p^{ID}(K_{1, n-1}) = \lceil \frac{2(n-1)}{3} \rceil$ .*

*Proof.* The lower bound follows from Prop. 8. Let  $v_0, \dots, v_{n-1}$  be the leaves of  $K_{1, n-1}$  and  $c$  its central vertex. Let  $\mathcal{P}$  be the set  $\mathcal{P} = \{v_i c v_{i+1} \mid i \neq 2 \bmod 3\}$  of  $2\lfloor \frac{n}{3} \rfloor$  paths. If  $n = 0 \bmod 3$ ,  $\mathcal{P}$  is an identifying path cover of  $K_{1, n-1}$ . If  $n = 1 \bmod 3$ ,  $\mathcal{P} \cup \{c v_{n-1}\}$  is, and if  $n = 2 \bmod 3$ ,  $\mathcal{P} \cup \{c v_{n-2}, c v_{n-1}\}$  is.  $\square$

We call the procedure used in the proof of Thm. 12 "covering three leaves with two paths". A tree is *topologically irreducible* if it has no vertex of degree 2.

**Theorem 13.** *Let  $T$  be a tree with  $l$  leaves. Then we have*

- (i) *if  $T$  is topologically irreducible, then  $p^{ID}(T) = \lceil \frac{2l}{3} \rceil$ ,*
- (ii) *if  $T$  has  $t$  vertices of degree two, then  $\lceil \frac{2l}{3} \rceil \leq p^{ID}(T) \leq \lceil \frac{2l}{3} \rceil + \lceil \frac{t}{2} \rceil$ .*

*Proof.* The lower bound in both cases follows from Prop. 8.

If  $T$  is topologically irreducible, we show how to construct an identifying path cover which size is meeting the lower bound.

First, determine the *center* of  $T$  (that is, the set of vertices of minimum largest distance to any other vertex of  $T$ ). By Jordan's theorem, the center of a tree consists of either a single vertex or a pair of adjacent vertices.

Starting from the center of  $T$ , decompose the vertex set of  $T$  into *layers* labelled  $0, \dots, h$ , where  $h$  is the radius of  $T$  (the minimum largest distance among pairs of vertices of  $T$ ). The labels correspond to the distance to the center. For  $\ell \in \{0, \dots, h\}$ , let  $T_{\leq \ell}$  be the sub-tree of  $T$  induced by layers  $0, \dots, \ell$ .

In the case when the center of  $T$  consists of a single vertex  $T_{\leq 1}$  is isomorphic to a star, and in the case when the center consists of two adjacent vertices, it is isomorphic to a tree with two adjacent vertices of degree at least 3, and all other vertices of degree 1. In both cases, it is straightforward to find a solution of size  $\lceil \frac{2l}{3} \rceil$ .

For  $i \in \{2, \dots, h\}$  we now describe how to extend the valid solution of  $T_{\leq i-1}$  to the solution of  $T_{\leq i}$ . For any vertex  $v$  from layer  $i-1$ , we choose an arbitrary neighbour from layer  $i$  and extend the corresponding paths (by our construction always exactly one or two paths are ending in a leaf) from the identifying path cover to the neighbour. Now we have two vertices covered by exactly the same set of paths, and still many uncovered vertices, all of them from layer  $i$  (at least one more uncovered neighbour for any vertex from layer  $i-1$ ). Now we contract all covered vertices into a single vertex, obtaining a star. Then we order (arbitrarily) all other remaining vertices and do the procedure "covering three leaves with two paths", as done in the case of the proof of Thm. 12. Doing this we separate all uncovered vertices from layer  $i$ . We then expand the obtained star together with the chosen paths into the original tree, where a path between two leaves expands through the unique shortest path in  $T$  between these two leaves. Doing this we also separate the two vertices, one from layer  $i-1$  and the other from layer  $i$ , which were sharing the same paths. Vertices from layers  $0, 1, \dots, i-2$  still remain separated and covered.

Doing this procedure until we reach layer  $h$  and tree  $T_{\leq h} = T$ , we obtain at each layer  $i$  a solution of size  $\lceil \frac{2(l_i-1)}{3} \rceil$  for a tree  $T_{\leq i}$ , where  $l_i$  denotes the number of vertices from layer  $i$ . At the end we obtain a solution of size  $\lceil \frac{2l}{3} \rceil$  for  $T_{\leq h} = T$ . This concludes the proof of the first part of the theorem.

If  $T$  is arbitrary, we first contract all vertices of degree 2 to obtain the topologically irreducible tree  $T'$  and find a solution for  $T'$  as described above. Then we subdivide edges of  $T'$  to obtain tree  $T$ , keeping (expanding) the same identifying path cover. Now, observe that we may get pairs of vertices which are not separated from each other. Each such pair contains a vertex of degree 2. Using



a similar procedure as in the proof of Thm. 10, we can use  $\lceil \frac{t}{2} \rceil$  additional paths to get a solution reaching the upper bound from (ii).  $\square$

By Prop. 7, we get an identifying path cover  $\mathcal{P}$  of a connected graph  $G$  by choosing a spanning tree  $T$  of  $G$  and constructing  $\mathcal{P}$  using Thm. 13 on  $T$ . We get the following improvement of Thm. 3 for the case of identifying path covers of connected graphs. Note that by Thm. 12, this bound is tight for stars.

**Theorem 14.** *For any connected graph  $G$  on  $n$  vertices,  $p^{ID}(G) \leq \lceil \frac{2(n-1)}{3} \rceil$ .*

Unlike for many other variants of identification problems (such as identifying codes, see [9]), Thm. 14 shows that one needs much less sensors than  $n$  in order to identify connected graphs, which may prove useful in practice. We remark that the similar upper bound  $\frac{2n}{3}$  holds for the size of a watching system (i.e. an “identifying *star* cover”) in any connected graph on  $n$  vertices [1].

The bound of Thm. 14 can be refined in the following way. Let  $\gamma_C(G)$  denote the connected domination number of a graph  $G$  (that is, the minimum size of a dominating set of  $G$  inducing a connected subgraph) and let  $L(G)$  denote the maximum number of leaves in a spanning tree of  $G$ . One can observe that for a connected graph  $G$  on  $n$  vertices, we have  $n = \gamma_C(G) + L(G)$ . Hence using Prop. 7 and Thm. 13 we get the following upper bound.

**Theorem 15.** *For any connected graph  $G$  on  $n$  vertices, it holds that  $p^{ID}(G) \leq \lceil \frac{2(n-\gamma_C(G))}{3} \rceil + \lceil \frac{\gamma_C(G)}{2} \rceil$ .*

## 5 On the complexity of MIN-IDPC- $k$

In this section, we discuss the computational complexity of MIN-IDPC- $k$ . It is shown in [8] that MIN-IDPC- $k$  is approximable within a factor of  $O(\ln(k))$  for any  $k \geq 1$ . In fact, when  $k = 1$ , we are allowed only paths of length 0 (that is, vertices) and MIN-IDPC-1 is trivial: the only solution consists of the whole set of vertices. When  $k = 2$ , we want to identify the vertices using paths of two vertices, i.e. edges. This problem is equivalent to MIN-ID-2, where each attribute is common to exactly two individuals. Indeed, an edge can precisely be seen as such an attribute. This case has already been studied in [8], where a strong link between MIN-ID-2 and the maximum  $P_3$ -packing problem was established; the authors give a  $\frac{7}{6}$ -approximation for MIN-ID-2 and show that it is APX-hard by reduction from the maximum 3-dimensional matching problem.

We next prove that MIN-IDPC- $k$  is APX-hard for all  $k \geq 3$ , i.e. that there exists a constant  $c$  (depending on  $k$ ) for which MIN-IDPC- $k$  is not  $c$ -approximable. We use the framework of L-reductions. We recall the definition of an L-reduction between two optimization problems  $P$  and  $Q$  in Definition 16. It is known that if such a reduction exists and  $P$  is APX-hard, then  $Q$  is APX-hard as well. For more details, see [19]. Given an optimization problem  $P$  and a solution  $s$  to an instance  $x$  of  $P$ , we denote by  $cost_P(x, s)$ , the value of  $s$ , and by  $opt_P(x)$ , the value of an optimal solution to  $x$ .

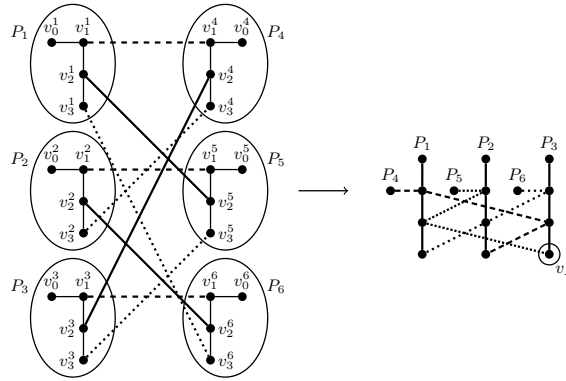
**Definition 16.** Let  $P$  and  $Q$  be two optimization problems. An L-reduction from  $P$  to  $Q$  is a four-tuple  $(f, g, \alpha, \beta)$  where  $f$  and  $g$  are polynomial time computable functions and  $\alpha, \beta$  are positive constants with the following properties:

1. Function  $f$  maps instances of  $P$  to instances of  $Q$  and for every instance  $x$  of  $P$ ,  $\text{opt}_Q(f(x)) \leq \alpha \cdot \text{opt}_P(x)$ .
2. For every instance  $x$  of  $P$  and every solution  $y$  of  $f(x)$ ,  $g$  maps the pair  $(f(x), y)$  to a solution  $y'$  of  $x$  such that  $|\text{opt}_P(x) - \text{cost}_P(x, g(f(x), y'))| \leq \beta \cdot |\text{opt}_Q(f(x)) - \text{cost}_Q(f(x), y')|$ .

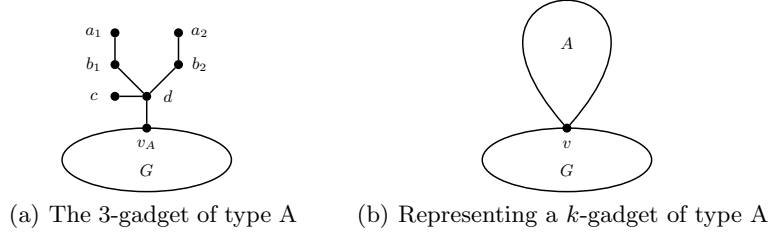
The problem *minimum vertex cover in cubic graphs*, MIN-VC-3 for short, given a cubic graph, is to find a minimum set of vertices such that each edge is covered by a vertex of the set. MIN-VC-3 is APX-hard [7].

**Theorem 17.** Let  $k \geq 3$ . There is an L-reduction (with parameters  $\alpha = 40k^2 - 116k + 47$  and  $\beta = 1$ ) from MIN-VC-3 to MIN-IDPC- $k$  in graphs of maximum degree 4. Hence MIN-IDPC- $k$  is APX-complete, even in this class of graphs.

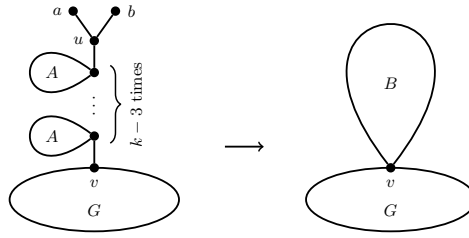
Before giving the proof of Thm. 17, we first provide two useful gadgets and exhibit some of their properties. Given  $k \geq 3$ , we call these gadgets *k-gadget of type A* (see Fig. 3 for an example when  $k = 3$  and Fig. 2 when  $k = 4$ ) and *k-gadget of type B* (see Fig. 4). Both gadgets include an *attachment vertex* which will be identified with a vertex of the rest of the graph in our constructions. The *k-gadget of type A* is the basis for the construction of the *k-gadget of type B*, which includes  $k - 3$  copies of the *k-gadget of type A*. The *k-gadget of type B* is described in Fig. 4, but we define the *k-gadget of type A* more formally. Using these two gadgets, we construct a *vertex k-gadget* and an *edge k-gadget* which will be needed in the reduction (see Fig. 5). The idea of the *k-gadget of type A* is to attach it at a vertex and make sure that this vertex can be easily covered and identified by a locally optimal solution; the idea of the *k-gadget of type B* is to force a path from outside the gadget to go through the attached vertex.



**Fig. 2.** Construction of the 4-gadget of type A



**Fig. 3.** The  $k$ -gadgets of type A.



**Fig. 4.** The  $k$ -gadget of type B and its representation.

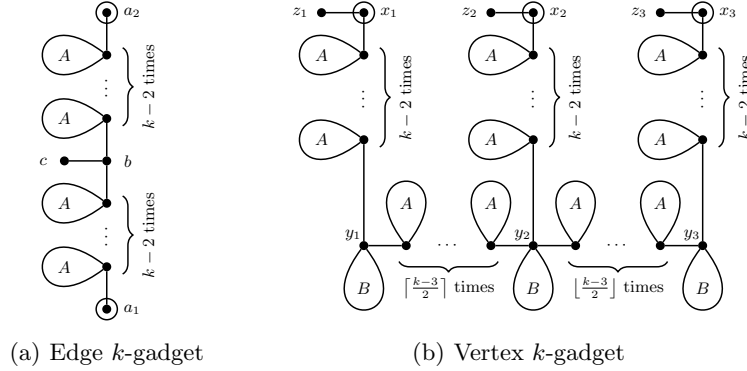
In order to construct the  $k$ -gadget of type A, we use a construction of an extremal graph for the lower bound  $\frac{2n}{k+1} \leq p_k^{\text{ID}}(G)$  (Theorem 5). This construction is detailed in Definition 18.

**Definition 18.** Let  $k \geq 3$  be an integer. If  $k = 3$ , the  $k$ -gadget of type A with attachment vertex  $v_A$  is the graph of Fig. 3. When  $k \geq 4$ , the  $k$ -gadget of type A is constructed as follows (see Fig. 2):

- Let  $P_1, \dots, P_{2(k-1)}$  be  $2(k-1)$  vertex-disjoint paths, where for  $i \in \{1, \dots, 2(k-1)\}$ ,  $P_i = \{v_0^i, \dots, v_{k-1}^i\}$ .
- Consider the complete bipartite graph  $B$  whose vertices are  $P_1, \dots, P_{2(k-1)}$ . It is  $(k-1)$ -regular.
- Select a  $(k-1)$ -edge-colouring of  $B$  with colours from  $\{1, \dots, k-1\}$  (or equivalently, find a partition of the edges of  $B$  into  $k-1$  perfect matchings).
- If  $k$  is even or  $c \notin \{\lfloor \frac{k}{2} \rfloor, \lceil \frac{k}{2} \rceil\}$ , for each edge  $\{P_i, P_j\}$  (assume  $i \leq j$ ) of  $B$  coloured with colour  $c$ , identify vertices  $v_c^i$  and  $v_c^j$ . If  $k$  is odd and  $c = \lfloor k/2 \rfloor$  (resp.  $c = \lceil k/2 \rceil$ ), identify vertices  $v_{\lfloor k/2 \rfloor}^i$  and  $v_{\lceil k/2 \rceil}^j$  (resp.  $v_{\lceil k/2 \rceil}^i$  and  $v_{\lfloor k/2 \rfloor}^j$ ).
- We let the attachment vertex  $v_A$  be vertex  $v_{k-1}^i$  for some arbitrary  $i \in \{1, \dots, 2(k-1)\}$ .

We let  $x_A = 2(k-1)$ . Note that when  $k \geq 4$ ,  $x_A$  is the number of degree 1 vertices in a  $k$ -gadget of type A.

Let  $G$  be a cubic graph on  $n$  vertices and  $m = \frac{3n}{2}$  edges. We construct the graph  $f(G)$  by replacing every vertex  $v$  by a copy of vertex gadget  $G_v$  and each edge  $e$  by a copy of edge gadget  $G_e$  (see Fig. 5). Given a vertex  $v$  incident to edges  $e_1, e_2, e_3$  in  $G$ , the vertices  $x_1, x_2, x_3$  of  $G_v$  are identified each with either one of the vertices  $a_1, a_2$  of  $G_{e_1}, G_{e_2}, G_{e_3}$  in  $f(G)$ . It is easily noticed that since  $G$  is cubic,  $f(G)$  has maximum degree 4.



**Fig. 5.** Reduction  $k$ -gadgets for vertices and edges.

The first main idea of the reduction is to simulate the covering of an edge  $e$  by the separation of vertices  $b$  and  $c$  in  $G_e$  by a path going through  $b$ . The second main idea is, given a vertex  $v$ , to encode the fact that  $v$  is part of a vertex cover of  $G$ , by having path  $y_1 \dots y_3$  inside the path cover of  $f(G)$  (which enables us to “cover” the three edge-gadgets corresponding to the three edges incident to  $v$  in  $G$ ). The proof of the validity of the reduction uses the following Claims 19, 20 and 21 about the gadgets of type A and B.

**Claim 19.** *Let  $A$  be  $k$ -gadget of type A ( $k \geq 4$ ). Then, for each pair  $v, v'$  of the  $x_A$  vertices of degree 1 in  $A$ , there is no path of length  $k - 1$  between  $v$  and  $v'$ . Moreover, vertex  $v_A$  is at distance at least  $k - 1$  of any degree 1-vertex in  $A$ .*

*Proof.* If  $k$  is even, then  $A$  is bipartite with all degree 1 vertices in the same part. Hence all paths between two degree 1 vertices have even length, but  $k - 1$  is odd. If  $k$  is odd, by contradiction consider a path  $P$  between two degree 1 vertices of  $A$ . If for any  $i$ , there is no vertex of the form  $v_{\lfloor k/2 \rfloor}^i$  or  $v_{\lceil k/2 \rceil}^i$  in  $P$ , then  $P$  cannot be of length  $k - 1$  because there is no such path of length more than  $k - 3$ . Hence  $P$  contains some vertex  $v_{\lfloor k/2 \rfloor}^i$  or  $v_{\lceil k/2 \rceil}^i$ . But in either case,  $P$  must have at least  $k + 1$  vertices, a contradiction. The second part of the statement follows immediately from the choice of vertex  $v_A$  in the construction of  $A$ .  $\square$

In what follows, we let  $G$  be a graph and  $\mathcal{P}$ , an identifying path cover of  $G$ .

**Claim 20.** *Let  $A$  be a  $k$ -gadget of type  $A$  ( $k \geq 3$ ) attached at vertex  $v_A$  in  $G$ . Then, there is a set of at least  $x_A$  paths of  $\mathcal{P}$  having an endpoint in  $A$ , and none of these paths can reach a vertex outside of  $A$ . Moreover, there is a set of  $x_A$  paths in  $A$  which can be used to cover and identify all vertices of  $A$ .*

*Proof.* When  $k = 3$ , we note that we need at least three 3-paths in order to cover the three degree 1 vertices of  $A$ . Doing so, we need an additional path to separate either  $a_1$  from  $b_1$  or  $a_2$  from  $b_2$ . Finally, the four paths  $a_1 \dots d$ ,  $a_2 \dots d$ ,  $c \dots b_1$ ,  $b_2 \dots v_A$  fulfill the last part of the statement.

If  $k \geq 4$ , the first part of the statement follows easily from Claim 19: no path can be used to cover two degree 1 vertices in  $A$ , and no path starting outside of  $A$  can cover a degree 1 vertex of  $A$ . Considering the  $x_A$  paths of the construction of  $A$  proves the second part of the statement.  $\square$

**Claim 21.** *Let  $B$  be a  $k$ -gadget of type  $B$  attached at some vertex  $v$  in  $G$ . Then, at least  $x_B = x_A(k - 3) + 2$  paths of  $\mathcal{P}$  are entirely contained in  $B$ . Moreover, if exactly  $x_B$  paths of  $\mathcal{P}$  are entirely contained in  $B$ , then there is an additional path of  $\mathcal{P}$  containing vertex  $v$ . Finally, there exists such a set of  $x_B$  paths.*

*Proof.* Following Claim 20, we need at least  $x_A$  paths in each of the  $k - 3$  copies of the  $k$ -gadget of type  $A$  in  $B$ . In order to dominate vertices  $a$  and  $b$ , we need two additional paths  $P_a$  and  $P_b$  starting in  $a$  and  $b$ , which completes the first part of the claim. For the second part, by Claim 20, among these paths, only paths of type  $P_a$  or  $P_b$  can contain vertex  $v$ . If  $P_a$  or  $P_b$  or both  $P_a, P_b$  dominate  $v$ , then  $v$  is not separated from either  $a, b$  or  $u$ , proving the second part. Taking the solution from Claim 20 for each copy of the  $k$ -gadget of type  $A$  together with the paths  $a \dots v$  and  $b \dots v$ , we get the last part of the claim.  $\square$

We are now ready to prove Thm. 17.

*Proof (Proof of Thm. 17).* We first prove that the first part of Def. 16 holds. Let  $C^*$  be a minimum vertex cover of  $G$ . We construct an identifying  $k$ -path cover  $\mathcal{P}$  of  $f(G)$  as follows. For each copy of a gadget of type  $A$  (resp. type  $B$ ), take the solution of size  $x_A$  described in the proof of Claim 20 (resp. of size  $x_B$  of Claim 21) into  $\mathcal{P}$ . Now, for each edge  $e$  of  $G$ , add an arbitrary path starting in vertex  $c$  of  $G_e$ . For each vertex  $v$  of  $G$ , add three arbitrary paths starting in vertices  $z_1, z_2$  and  $z_3$ , respectively. Let  $e_1, e_2, e_3$  be the three edges incident to  $v$  in  $G$ , and  $b_1, b_2, b_3$ , the three vertices labelled  $b$  in  $G_{e_1}, G_{e_2}, G_{e_3}$ . If  $v \in C^*$ , add path  $y_1 \dots y_3$  of  $G_v$ , as well as paths  $x_2 \dots b_1, x_3 \dots b_2$  and  $x_1 \dots b_3$  to  $\mathcal{P}$ . If  $v \notin C^*$ , add paths  $x_1 \dots y_1, x_2 \dots y_2$  and  $x_3 \dots y_3$  to  $\mathcal{P}$ .

The reader can check that  $\mathcal{P}$  is an identifying  $k$ -path cover of  $f(G)$ , and that:

$$|\mathcal{P}| \leq |C^*| + (2(k - 2)x_A + 1)m + ((3(k - 2) + k - 3)x_A + 3x_B + 6)n \quad (1)$$

Since  $G$  is cubic, each vertex of  $C^*$  can cover at most three edges and we have  $|C^*| \geq \frac{m}{3}$  and hence  $m \leq 3|C^*|$  and  $n \leq 2|C^*|$ . We get:  $p_k^{\text{ID}}(G) \leq |\mathcal{P}| \leq (16 + (14k - 30)x_A + 6x_B)|C^*|$  and hence (1) of Def. 16 is fulfilled with  $\alpha = 16 + (14k - 30)x_A + 6x_B \leq 40k^2 - 116k + 47$ .

It remains to prove the second part of Def. 16. Let  $\mathcal{P}$  be an identifying  $k$ -path cover of  $f(G)$ . We construct a vertex cover  $C$  using  $\mathcal{P}$ . First of all, by Claim 20, each gadget of type  $A$  contains at least  $x_A$  paths of  $\mathcal{P}$ , and by Claim 21, each gadget of type  $B$  contains at least  $x_B = x_A(k - 3) + 2$  paths of  $\mathcal{P}$ . Moreover, in each vertex gadget, at least three paths belong to  $\mathcal{P}$  in order to cover vertices  $z_1, z_2, z_3$ . However, using the structure of the vertex- and edge-gadgets together with Claim 21, if there are exactly that many paths, in each vertex-gadget, we are not able to separate vertices  $y_1, y_2, y_3$  from their respective neighbours, as well as vertices  $x_1, z_1, x_2, z_2$ , and  $x_3, z_3$  (similarly, vertices  $b$  and  $c$  in each edge-gadget). Besides the paths of  $\mathcal{P}$  that we already considered, in any vertex-gadget, at least three paths are required in order to cover vertices  $x_1, y_1, x_2, y_2, x_3, y_3$ . If there are *exactly* three, then they must be  $x_1 \dots y_1, x_2 \dots y_2, x_3 \dots y_3$ . We construct  $C$  as follows: for each vertex  $v$  of  $G$ , if there are at least four such additional paths in  $G_v$ , we add  $v$  to  $C$ . Set  $C$  is a vertex cover of  $G$ : indeed, when  $v \in C$ , the paths in the copies of the gadgets of type  $A$  and  $B$  in  $G_v$  could be replaced by the “standard” solution given in Claims 20 and 21. Moreover, the (at least) four additional paths of  $\mathcal{P}$  in  $G_v$  could be replaced by  $y_1 \dots y_3$  and three paths starting at  $x_1, x_2, x_3$  covering the three vertices labelled  $b$  in the three edge-gadgets corresponding to the three edges incident to  $v$  in  $G$ . Hence these edges would be covered. This procedure would give an easy constructable identifying path cover  $\mathcal{P}'$  with  $\mathcal{P}' \leq \mathcal{P}$ , and since all vertices labelled  $b$  are covered by a path of  $\mathcal{P}'$ ,  $C$  is a vertex cover of  $G$ . Furthermore, we have  $|C| \leq |\mathcal{P}| - (2(k - 2)x_A + 1)m - ((3(k - 2) + k - 3)x_A + 3x_B + 6)n$ . Applying the construction to a minimum identifying  $k$ -path cover, we get  $|C^*| \leq |C| \leq p_k^{\text{ID}}(G) - (2(k - 2)x_A + 1)m - ((3(k - 2) + k - 3)x_A + 3x_B + 6)n$ . Together with Equation (1), this implies:

$$|C^*| = p_k^{\text{ID}}(G) - (2(k - 2)x_A + 1)m - ((3(k - 2) + k - 3)x_A + 3x_B + 6)n \quad (2)$$

From Equations (1) and (2), we get  $|C| - |C^*| \leq |\mathcal{P}| - p_k^{\text{ID}}(G)$ , which implies  $||C^*| - |C|| \leq |p_k^{\text{ID}}(G) - |\mathcal{P}||$ ; hence (2) of Def. 16 is fulfilled with  $\beta = 1$ .  $\square$

## 6 Conclusion and open problems

We conclude with some open problems. We gave a procedure to compute the exact value of parameter  $p^{\text{ID}}$  for topologically irreducible trees, but only gave an upper bound for general trees. It seems not easy to extend the algorithm to the latter case, but it would be interesting to design an(other) algorithm to solve it. Regarding MIN-IDPC- $k$ , we mentioned that not all graphs admit an identifying  $k$ -path cover. Identifiable graphs have been studied for some other identification problems [5]; it would be interesting to do so in our context, i.e. studying  $k$ -path identifiable graphs. Finally, we have settled the complexity of MIN-IDPC- $k$  by showing that it is APX-complete. However, the question of the complexity of the general MIN-IDPC problem remains open.

**Acknowledgements** We would like to thank P. J. Slater and J. L. Sewell for motivating us to study this problem. We also thank O. Schaudt for giving the idea of Theorem 15.

## References

1. D. Auger, I. Charon, O. Hudry, A. Lobstein. Watching systems in graphs: an extension of identifying codes. *Discrete Appl. Math.*, in press.
2. T. Y. Berger-Wolf, M. Laifenfeld, A. Trachtenberg. Identifying codes and the set cover problem. *Proc. 44th Annual Allerton Conference on Communication, Control and Computing*, September 2006.
3. J. A. Bondy. Induced subsets. *J. Comb. Theory B* 12(2):201–202, 1972.
4. I. Charon, G. Cohen, O. Hudry, A. Lobstein. Discriminating codes in bipartite graphs. *Adv. Math. Commun.* 4(2):403–420, 2008.
5. I. Charon, I. Honkala, O. Hudry, A. Lobstein. Structural properties of twin-free graphs. *Electron. J. Comb.* 14:R16, 2007.
6. I. Charon, O. Hudry, A. Lobstein. Identifying and locating-dominating codes: NP-completeness results for directed graphs. *IEEE T. Inform. Theory* 48(8):2192–2200, 2002.
7. M. Chlebík, J. Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theor. Comput. Sci.* 354:320–338, 2006.
8. K. M. J. De Bontridder, B. V. Halldórsson, M. M. Halldórsson, C. A. J. Hurkens, J. K. Lenstra, R. Ravi, L. Stougie. Approximation algorithms for the test cover problem. *Math. Program. B*, 98:477–491, 2003.
9. F. Foucaud, E. Guerrini, M. Kovše, R. Naserasr, A. Parreau, P. Valicov. Extremal graphs for the identifying code problem. *Eur. J. Combin.* 32(4):628–638, 2011.
10. F. Foucaud, S. Gravier, R. Naserasr, A. Parreau, P. Valicov. Identifying codes in line graphs. To appear in *J. Graph Theor.*
11. F. Foucaud, R. Naserasr, A. Parreau. Characterizing extremal digraphs for identifying codes and extremal cases of Bondy’s theorem on induced subsets. *Graphs Comb.*, in press.
12. A. Frieze, R. Martin, J. Moncel, M. Ruszinkó, C. Smyth. Codes identifying sets of vertices in random networks. *Discrete Math.* 307(9-10):1094–1107, 2007.
13. M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
14. I. Honkala, M. Karpovsky, S. Litsyn. Cycles identifying vertices and edges in binary hypercubes and 2-dimensional tori. *Discrete Appl. Math.* 129(2-3):409–419, 2003.
15. D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.* 9, 256–278, 1974.
16. M. Karpovsky, K. Chakrabarty, L. B. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE T. Inform. Theory*, 44:599–611, 1998.
17. M. Laifenfeld, A. Trachtenberg, R. Cohen, D. Starobinski. Joint monitoring and routing in wireless sensor networks using robust identifying codes. *Proc. IEEE Broadnets 2007*, pages 197–206, September 2007.
18. B. M. E. Moret, H. D. Shapiro. On minimizing a set of tests. *SIAM J. Sci. Stat. Comp.* 6(4):983–1003, 1985.
19. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
20. P. Rosendahl. On the identification of vertices using cycles. *Electron. J. Comb.* 10:P1, 2003.