Identifying path covers in graphs

Florent Foucaud, Matjaz Kovse

Identifying path covers in graphs[☆]

Florent Foucaud[a], Matjaž Kovše[b]

[a] *Univ. Bordeaux, LaBRI, UMR5800, F-33400 Talence, France.*
CNRS, LaBRI, UMR5800, F-33400 Talence, France.
[b] *Bioinformatics Group, Department of Computer Science and Interdisciplinary Center for Bioinformatics, Univ. of Leipzig, Härtelstrasse 16-18, D-04107 Leipzig, Germany.*

Abstract

This paper introduces the problem of identifying vertices of a graph using paths. 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 new notion is related to a large number of other identification problems in graphs and hypergraphs. We study the identifying path cover problem under both combinatorial and algorithmic points of view. In particular, we derive the optimal size of an identifying path cover for paths, cycles and hypercubes, and give upper bounds for 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 $\left\lceil \frac{2|V(G)|}{3} \right\rceil + 5$. We then 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 bounded 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.

1.1. 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*

[☆] A shorter version of this paper appeared under the name *On graph identification problems and the special case of identifying vertices using paths* in the Proceedings of the International Workshop on Combinatorial Algorithms, IWOCA 2012 [11]. The present paper contains additional results, mainly Theorems 8, 10 and 20, and a corrected version of Theorems 17 and 18, which contained some mistakes in the short version.

problem [8], *minimum test collection problem* [14], *minimum test set problem* [21]). 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* [14, 16], 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 [21]. On the other hand, both problems are not only NP-hard [4, 14] 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, 13, 18], 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, 12], and to the case where also *sets* of at most ℓ vertices are to be separated and where vertices can identify at some prescribed distance $r \geq 1$ [13]. 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 subgraphs of 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 [15, 23].

1.3. The identifying path cover problem

Given a graph G, a *path* is an ordered set of distinct vertices such that any two consecutive vertices in the ordering are adjacent. A path containing k vertices is a *k-path*; a 1-path is just a single vertex. We will consider the path graph P_k, consisting of a single path on k vertices; similarly, C_k denotes the cycle on k vertices.

In this paper, we study MIN-ID when $\mathcal{I} = V(G)$ and $\mathcal{A}$ is the set of all paths of G. 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 $p^{\mathrm{ID}}(G)$ 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 Figure 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 1-paths of G, that is, $\mathcal{P} = V(G)$.

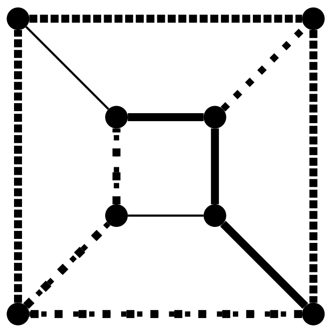


Figure 1: An (optimal) identifying path cover of the hypercube H_3: $p^{\text{ID}}(H_3) = 4$.

Given a fixed 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 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. For fixed k, 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, 21]. When the instance of the problem arises from substructures of a graph, the main applications are routing in networks [20] and the location of threats in facilities or networks using sensors [18]: 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 Section 3: we give exact values for parameter p^{ID} for paths, cycles and upper bounds for trees in general. The latter bound is proved using a procedure for constructing an identifying path cover. We also use it to provide the upper bound $p^{\text{ID}}(G) \leq \lceil \frac{2n}{3} \rceil + 5$ for any connected graph. Finally, we show in Section 4 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 5.

2. Bounds and preliminary observations

The following lower bound was observed in [18] in the context of identifying codes but we refer to [4] for the general statement.

Theorem 2 ([4, 18])**.** *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 inclusion-wise 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 [18], but the proof works in the more general context of MIN-ID-k. We give a general proof for completeness.

Theorem 4 ([18])**.** *Let $k \geq 1$ and $(\mathcal{I}, \mathcal{A})$ be an instance of MIN-ID-k. Then for any solution $\mathcal{C}$, $|\mathcal{C}| \geq \frac{2|\mathcal{I}|}{k+1}$.*

Proof. Let $c = |\mathcal{C}|$. Denote by i_1 the number of individuals covered by a unique attribute of $\mathcal{C}$, and i_2, the number of individuals covered by at least two attributes. We have $i_1 \leq c$ and $i_2 \leq \frac{ck - i_1}{2}$ since each attribute covers at most k individuals. Hence, $|\mathcal{I}| = i_1 + i_2 \leq c + \frac{ck}{2} - \frac{c}{2} = \frac{c(k+1)}{2}$ and we get the claimed bound. □

Applying theorems 2, 3 and 4 to the case of identifying path covers, we get:

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

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, $p^{\mathrm{ID}}(K_n) = \lceil \log_2(n + 1) \rceil$. 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 [15, 23]; 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 ([15, 23])**.** *Let H_d be the hypercube of dimension d with $n = 2^d$ vertices. Then $p^{ID}(H_d) = \lceil \log_2(n + 1) \rceil$.*

In fact, we can describe a construction for all graphs G reaching the bound.

Construction 7. *Let $V(G)$ be a set of n vertices, and $\mathcal{P}$ a set of $\lceil \log_2(n + 1) \rceil$ elements.*

- *Assign a distinct nonempty subset P_v of $\mathcal{P}$ to each vertex v of $V(G)$.*
- *For each element p of $\mathcal{P}$, consider the set V_p of vertices $\{v : p \in P_v\}$.*
- *Add edges such that the vertices of V_p induce a path.*
- *Add any arbitrary set of edges.*

It is easy to check that the following theorem holds:

Theorem 8. *In the bound $p^{ID}(G) \geq \lceil \log_2(n + 1) \rceil$ of Theorem 2 and for the special case of identifying path covers of graphs, equality holds if and only if the graph can be constructed using Construction 7.*

We now consider the lower bound of Theorem 4 and show that it is tight for the MIN-IDPC-k problem.

Construction 9. *Let $k \geq 1$ and $p \geq k$ be a pair of integers such that $p(k - 1)$ is even.*

- *Consider a collection $\mathcal{P}$ of p vertex-disjoint paths of k vertices.*

- *For each path, select a vertex to be the one which will be covered only by this path.*
- *Consider a perfect matching M between all remaining vertices,*[1] *such that:*
  - *No two vertices belonging to the same path are matched.*
  - *For each pair of paths, at most one pair of their vertices is matched.*
  - *Each pair of M belongs to a distinct pair of paths.*
- *Contract each edge of M (identify both vertices).*
- *Add arbitrarily many edges (so as to make the graph connected).*

Theorem 10. *In the bound $p_k^{ID}(G) \geq \frac{2|V(G)|}{k+1}$ of Theorem 4, equality holds for infinitely many connected graphs for any $k \geq 1$ and $p \geq k$ paths in the optimal solution.*

Proof. Let G be a graph constructed using Construction 9, where the identifying k-path cover $\mathcal{P}$ is the set of p paths from the first step of the construction. It is clear that each vertex is identified: for each path P of $\mathcal{P}$, one vertex is covered only by P, and the remaining vertices are all covered by a distinct pair of paths of $\mathcal{P}$. The number of vertices of the resulting graph is $|V(G)| = p + \frac{p(k-1)}{2} = \frac{p(k+1)}{2}$ and hence $p = \frac{2|V(G)|}{k+1}$. Note that connectivity can be achieved without the addition of edges if and only if $k \geq 3$, by choosing a matching which makes the structure connected. $\square$

We remark that in order to identify all vertices covered by two paths, we need $\frac{p(k-1)}{2} \leq \binom{p}{2}$, that is, $p \geq k$ and therefore this condition is necessary in Construction 9 and Theorem 10.

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

The following proposition will be useful. The bound will be shown to be tight for the star (see Theorem 16).

Proposition 12. *If G is a graph having ℓ vertices of degree 1, $p^{ID}(G) \geq \lceil \frac{2\ell}{3} \rceil$.*

Proof. Let $\mathcal{P}$ be an identifying path cover of G. Note that for every vertex v of degree 1, there is a path of $\mathcal{P}$ ending at v (otherwise v is not covered). Moreover, if the two ends x, y of a given path both have degree 1, then these two ends are not identified unless another path starts at x or at y.

We now assign to each degree 1-vertex a weight of value 1: we have total weight ℓ in G. If a vertex of degree 1 is covered by k paths of $\mathcal{P}$, it distributes weight $\frac{1}{k}$ to each of these paths. By the previous paragraph, each path of $\mathcal{P}$ receives at most weight $1 + \frac{1}{2}$, hence $\ell \leq \frac{3|\mathcal{P}|}{2}$. $\square$

3. Identifying path covers for cycles, paths, trees and applications to all connected graphs

In this section, we give bounds for path identification of cycles, paths, trees and apply the bounds on trees to all connected graphs.

[1] One possible way to find M is to consider a $(k-1)$-regular graph H on p vertices: it has $\frac{p(k-1)}{2}$ edges (such graphs always exist, for example circulant graphs with well-chosen chord lengths). Assign a path of $\mathcal{P}$ to each vertex of H: M is built using the edge set of H.

3.1. Paths and cycles

We first investigate identifying path covers in simple graphs such as paths and cycles. We start with a lower bound for these graphs.

Proposition 13. *Let G be a connected graph of maximum degree 2 having m edges and ℓ vertices of degree one. Then $p^{ID}(G) \geq \lceil \frac{m+\ell}{2} \rceil$.*

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. ☐

Theorem 14. *For any $n \geq 1$, $p^{ID}(P_n) = \lceil \frac{n+1}{2} \rceil$.*

Proof. The lower bound comes from Proposition 13. 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 $\lceil \frac{n+1}{2} \rceil$. ☐

Theorem 15. *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 Proposition 13, and from Theorem 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 $\mathcal{P} = \{v_iv_{i+1}v_{i+2} \mid i \text{ even}, i < n-1\}$. 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 $\lceil \frac{n}{2} \rceil$. ☐

3.2. Trees

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 16. *It holds that $p^{ID}(K_{1,3}) = 3$. For any integer $n \geq 3$ and $n \neq 4$, $p^{ID}(K_{1,n-1}) = \left\lceil \frac{2(n-1)}{3} \right\rceil$.*

Proof. It is easily checked that $p^{\mathrm{ID}}(K_{1,2}) = 2$ and $p^{\mathrm{ID}}(K_{1,3}) = 3$. Let $n \geq 5$ be an integer. The lower bound follows from Proposition 12.

For the upper bound, the main idea is to use two paths of the solution to identify three leaves of the star. Let $v_0, \dots, v_{n-2}$ be the leaves of $K_{1,n-1}$ and c, its central vertex. Let $\mathcal{P}$ be the following set of paths: $\mathcal{P} = \{\{v_i, c, v_{i+1 \bmod (n-1)}\} \mid i \neq 2 \bmod 3\}$. The construction is illustrated in Figure 2, where the paths of $\mathcal{P}$ are marked with dotted and dashed lines.

Assume first that $n-1 = 0 \bmod 3$, i.e. $n-1 = 3k$ for some $k \geq 1$. Then, $|\mathcal{P}| = 2k = \frac{2(n-1)}{3}$. Vertex c is the only vertex to belong to all paths of $\mathcal{P}$, whereas each leaf v_i is covered by $\{v_i, c, v_{i+1}\}$ if $i = 0 \bmod 3$, by $\{v_i, c, v_{i+1}\}$ and $\{v_{i-1}, c, v_i\}$ if $i = 1 \bmod 3$, and by $\{v_{i-1}, c, v_i\}$ if $i = 2 \bmod 3$. Since $n \geq 5$, vertex c is covered by at least four paths and $\mathcal{P}$ is an identifying path cover of $K_{1,n-1}$.

If $n-1 = 1 \bmod 3$, i.e. $n-1 = 3k+1$ for some $k \geq 1$, then $|\mathcal{P}| = 2k+1 = \left\lceil \frac{2(n-1)}{3} \right\rceil$. Similarly, if $n-1 = 1 \bmod 3$, i.e. $n-1 = 3k+2$ for some $k \geq 1$, $|\mathcal{P}| = 2k+2 = \left\lceil \frac{2(n-1)}{3} \right\rceil$. For these cases, the argument to show that $\mathcal{P}$ is an identifying path cover of $K_{1,n-1}$ is similar to the first case, except that v_0 is now covered by two paths. ☐

A tree is *topologically irreducible* if it has no vertex of degree 2. We call a *maximal degree-2-path*, a path on at least three vertices where all vertices except the endpoints have degree 2.

Theorem 17. *Let T be a tree with ℓ leaves. Then,*

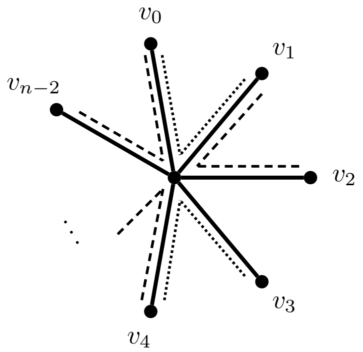


Figure 2: An identifying path cover of $K_{1,n-1}$, with $n \neq 0 \bmod 3$.

(i) if T is topologically irreducible, then $\lceil \frac{2\ell}{3} \rceil \leq p^{ID}(T) \leq \lceil \frac{2\ell}{3} \rceil + 2$*;*

(ii) if T has t vertices of degree two and k_e maximal degree-2-paths of even order at least 6, then $\lceil \frac{2\ell}{3} \rceil \leq p^{ID}(T) \leq \lceil \frac{2\ell}{3} \rceil + \frac{t+k_e}{2} + 4 \leq \lceil \frac{2\ell}{3} \rceil + \frac{5t}{8} + 4$.

Proof. For both cases, the lower bound follows from Proposition 12. Let T be a tree with ℓ leaves and t vertices of degree two.

<u>Case 1:</u> $t = 0$, i.e. T is topologically irreducible.
First, let C be the *center* of T (that is, the set of vertices of minimum largest distance to any other vertex of T). By Jordan's theorem [17], the center of a tree consists of either a single vertex or a pair of adjacent vertices.

Starting from C, decompose the vertex set of T into *layers* labelled $0, \dots, h$, where h is the largest distance between the center and a vertex of T. The labels correspond to the distance to C. For $i \in \{0, \dots, h\}$, let $T_{\leq i}$ be the sub-tree of T induced by layers $0, \dots, i$, and let ℓ_i be the number of leaves in $T_{\leq i}$.

For $i \in \{1, \dots, h\}$, we will inductively construct an *almost-identifying path cover* of $T_{\leq i}$, i.e. a set $\mathcal{P}_i$ of paths of $T_{\leq i}$ having the following properties.

1. All pairs of vertices of $T_{\leq i}$ are separated by $\mathcal{P}_i$, except possibly (i) the pairs containing a vertex of C, and (ii) a special pair s_1, s_2 of leaves of $T_{\leq i}$ that are not leaves of T, and that are both covered by the same set of two paths.
2. We have $|\mathcal{P}_i| = \lceil \frac{2\ell_i}{3} \rceil$.
3. For each edge $e = xy$ ($x, y \notin C$) of $T_{\leq i}$ with x being the closest vertex to center C, e is contained in a path of $\mathcal{P}_i$, and there is at least one path covering x which does not cover e.[2]
4. The two ends of each path of $\mathcal{P}_i$ are leaves of $T_{\leq i}$.
5. Almost every leaf of $T_{\leq i}$ is covered by at most two paths, and for almost every path $P \in \mathcal{P}_i$, there is a leaf of $T_{\leq i}$ that is covered only by P. The only exceptional cases are the following ones.
   - If $\ell_i = 1 \bmod 3$, there is a unique path P_0 of $\mathcal{P}_i$ such that if $i < h$, one of its ends (called x_i) is a non-leaf of T, and which satisfies:
     (i) the two ends of P_0 are covered by two paths, or
     (ii) $i < h$ and x_i is covered by two or three paths, x_i would be separated from all other vertices even without P_0, and the other end of P_0 is covered only by P_0.
   - If $\ell_i = 2 \bmod 3$, there are two paths P_0, P_1 of $\mathcal{P}_i$ whose two ends are covered by two paths, and P_0, P_1 have an end in common that is not a leaf of T (call it x_i).

[2]This property will be used in Case 2 of the proof.

It is clear that the set of paths $\mathcal{P}_h \cup \{\{c\}, c \in C\}$ would be an identifying path cover of T of size at most $\lceil \frac{2\ell}{3} \rceil + 2$ since $|C| \leq 2$. Next, we show how to construct $\mathcal{P}_h$.

Base step. Let us construct $\mathcal{P}_1$, the identifying path cover of $T_{\leq 1}$. The tree $T_{\leq 1}$ consists of C and the neighbours of vertices of C: if $|C| = 1$, $T_{\leq 1}$ is isomorphic to a star; if $|C| = 2$, $T_{\leq 1}$ has two adjacent vertices of degree at least 3, and all other vertices of degree 1. Let $v_0, \ldots, v_{\ell_1 - 1}$ be the leaves of $T_{\leq 1}$ such that (if $h > 1$) v_0 is not a leaf of T (since $h > 1$ such a vertex exists). Let $\mathcal{P}_1 = \{v_i - v_{(i+1) \bmod \ell_1} \mid i \neq 2 \bmod 3\}$, where $v_i - v_{(i+1) \bmod \ell_1}$ denotes the unique path from v_i to $v_{(i+1) \bmod \ell_1}$. This is the same set of paths as in the proof of Theorem 16, and all desired properties are fulfilled: this is clear for Properties 1 to 4. Property 5 is also clear if $\ell_i = 0 \bmod 3$. If $\ell_i = 1 \bmod 3$, $P_0 = v_0 - v_1$ and its non-leaf end x_1 is v_0 (we are in Case (i) of the first exception in Property 5). If $\ell_i = 2 \bmod 3$, $P_0 = v_{\ell_1 - 1} - v_0$ and $P_1 = v_0 - v_1$, and their non-leaf common end x_1 is v_0.

Inductive step. For $i \in \{2, \ldots, h\}$ we now describe how to extend the almost-identifying path cover $\mathcal{P}_{i-1}$ of $T_{\leq i-1}$ to the one for $T_{\leq i}$. First, let $\mathcal{P}_i = \mathcal{P}_{i-1}$. We will extend the existing paths, and add new ones to the construction.

a. Prolonging existing paths. Let $X, Y = \emptyset$ be two subsets of vertices of layer i that will need to be covered by new paths. More precisely, set X will contain those leaves of $T_{\leq i}$ that will not yet be covered after we prolong the existing paths, and Y will contain at most one leaf of $T_{\leq i}$ that will be already covered, but needs to be covered again in order to be identified. In Step b, we will cover and identify the vertices of $X \cup Y$.

Now, for each vertex v that is a leaf in $T_{\leq i-1}$ but not a leaf of T, we will prolong the paths in $\mathcal{P}_{i-1}$ ending at v. Since T has no degree 2 vertices, v has at least two children in $T_{\leq i}$, say $c_1, \ldots, c_k$ with $k \geq 2$. Moreover we assume that if v has a child that is also a leaf of T, then c_1 is such a child (this choice will be useful later in the proof).

First, assume v is one of the special vertices s_1, s_2 from Property 1 (without loss of generality, $v = s_1$). Vertex $v = s_1$ is covered by two paths P, P', has some children, and is not separated from s_2. We distinguish two cases. First, assume there is a non-leaf vertex $s \notin \{s_1, s_2\}$ of layer $i - 1$. Let c'_1, c'_2 be the children of s, and extend P to c_1, delete P' and instead, add a path $c_2 - c'_2$. Also add a path $c_1 - c'_2$. It is clear that c_1, c_2, c'_2, s_1, s_2 are identified. We will identify c'_1 in another step. Otherwise, the only non-leaf vertices of layer $i - 1$ are s_1, s_2. In that case we remove paths P, P' and instead we compute a solution similar to the base case. While doing this, we make sure that the path $c_1 - c_2$ belongs to the cover (in order to separate s_1 from s_2), and that we have a path going from a child of s_1 to a child of s_2 (in order to cover all edges as required by Property 3). This is possible by a good choice of ordering of the leaves. All vertices, including s_1, s_2, are now identified, which completes this case.

If v is neither the special vertex x_{i-1} nor s_1, prolong the path(s) covering v up to c_1. Note that v now needs to be separated from c_1. Moreover, all other children of v still need to be covered and identified (except if v is s, the vertex of the previous paragraph: then c_2 is already identified and covered). Add $c_2, \ldots, c_k$ ($c_3, \ldots, c_k$ if $v = s$) to X.

Now, we consider the case $v = x_{i-1}$ (which occurs if $\ell_{i-1} \neq 0 \bmod 3$).

If $\ell_{i-1} = 1 \bmod 3$ and we are in Case (i) of the first exception of Property 5, prolong P_0 up to child c_1, and prolong the other path P that cover v up to child c_2. Now, v, c_1, c_2 are separated from each other and c_1 is the unique vertex covered only by P_0. However, child c_2 needs to be covered by an additional path in order to be separated from the other end of P: add c_2 to Y. Finally, if they exist, add all vertices $c_3, \ldots, c_k$ to set X.

If $\ell_{i-1} = 1 \bmod 3$ and we are in Case (ii) of the first exception of Property 5, prolong P_0 up to child c_2, and prolong the other path(s) P (or P, P') that cover v up to child c_1. Again, v, c_1, c_2 are separated from each other, but now c_2 is not the only vertex covered only by P_0: add c_2 to Y, and if they exist, add $c_3, \ldots, c_k$ to X.

If $\ell_{i-1} = 2 \bmod 3$, prolong path P_0 to c_1 and path P_1 to c_2. Observe that v, c_1, c_2 are separated from each other. Moreover, c_1, c_2 are the unique vertices covered only by P_0 and P_1, respectively, and v is still the unique vertex covered only by both P_0, P_1. Again, if they exist, vertices $c_3, \ldots, c_k$ need to be covered:

add them to the set X.

b. Covering the vertices of X and Y. Now, we need to cover and identify all vertices of X (note that $X \neq \emptyset$, indeed, there are at least two non-leaves in each layer $0 < i < h$; in particular, in layer $i-1$, and by our construction, for at least one of them, at least one child belongs to X). If it exists, the vertex of Y needs to be covered by a new path. Furthermore, the leaves where existing paths have been prolonged need to be separated from their parent.

Assume $X = \{v_0, \ldots, v_{|X|-1}\}$, and, if $i < h$, v_0 is not a leaf of T (v_0 exists by our choice of ordering the children of each non-leaf vertex of layer $i-1$ in part a of the inductive step). Assume $|X| \geq 3$. If $Y = \emptyset$, as in the base step, we add the set of paths $\{v_j - v_{(j+1) \bmod |X|} \mid j \neq 2 \bmod 3\}$ to $\mathcal{P}_i$, where $v_j - v_{(j+1) \bmod |X|}$ denotes the unique path from v_j to $v_{(j+1) \bmod |X|}$. If $Y \neq \emptyset$, we have $Y = \{y\}$: add the set of paths $\{v_0 - y\} \cup \{v_j - v_{(j+1) \bmod |X|} \mid j \neq 0 \bmod 3\}$.

Assume now that $1 \leq |X| \leq 2$: $X = \{v_i, 0 \leq i \leq |X|-1\}$. Assume first $Y = \{y\} \neq \emptyset$. If $|X| = 1$, add path $y - v_0$ to the solution. If $|X| = 2$, add paths $y - v_0$ and $v_0 - v_1$ to the solution.

If $Y = \emptyset$ and $i = h$, we are in the last step; we add $|X|$ paths to cover the elements of X. If $i < h$ and $|X| = 1$, let z be another vertex of layer i that is a non-leaf of T: z exists since there are at least two non-leaves in this layer. Add path $v_0 - z$ to the solution: z is the special vertex x_i of Property 5, and $v_0 - z$ is path P_0 in the first exception of Property 5.

If $|X| = 2$, add two copies of path $v_0 - v_1$ to the solution, and mark v_0, v_1 as the special vertices s_1, s_2 of Property 1.

It is clear that all leaves of $(X \cup Y) \setminus \{s_1, s_2\}$ are now covered and identified. All vertices of $T_{\leq i-1}$ remain identified. Moreover, each non-leaf v of $T_{\leq i-1}$ is now separated from all its children $c_1, \ldots, c_k$: there is at least one path going through v but not c_1, and the paths of $\mathcal{P}_{i-1}$ going through v cover c_1 but none of $c_2, \ldots, c_k$. This proves Property 1. Moreover, Properties 2 to 4 are clear from the construction. Finally, for Property 5, the proof is the same as for the base step.

Case 2: $t > 0$**.** First, contract all maximal degree-2-paths into an edge as to obtain the topologically irreducible tree T'. Denote by C' the center of T'. Now, find a solution $\mathcal{P}'$ for T' as in Case 1 (recall that it is obtained from an almost-identifying path cover of T by adding a 0-path $\{c\}$ for each vertex c of C'). Expand T' and the paths of $\mathcal{P}'$ to obtain T and $\mathcal{P}$.

Let $S_1, \ldots, S_k$ ($k \leq n - t - 1$) be the vertex sets of all k maximal degree-2-paths in T. In other words, each set S_i induces a path $s^i_1, \ldots, s^i_{|S_i|}$, where we let s^i_1 be the vertex of S_i that is closest to the center C' of T', and $s^i_1, s^i_{|S_i|}$ are of degree different from 2. Note that $|S_i| \geq 3$.

By Property 1 of $\mathcal{P}'$ stated in the proof of Case 1, all vertices of degree different from 2 are identified by $\mathcal{P}$, except possibly those from the center C'; each vertex c of C' is identified by path $\{c\}$. Moreover, by Property 3 and by the existence of paths $\{c\}$ ($c \in C'$) in $\mathcal{P}'$, for each path S_i, all vertices of S_i are covered and separated from s^i_1. Hence, it remains to identify the vertices of degree 2 from S_i and, possibly, vertex $s^i_{|S_i|}$.

To do this, we will pair the sets $S_1, \ldots, S_k$ and, roughly speaking, for two sets S_i, S_j, add about $\frac{|S_i|+|S_j|}{2}$ paths to $\mathcal{P}$. We also do an additional special pairing for degree 2 paths of four vertices. Let us describe the process.

For every $i = 1 \bmod 2$, $1 \leq i \leq k-1$, we pair S_i with S_{i+1}. If k is even, all sets S_i are paired, otherwise, the last set is not. For a set S_j, we let $p(S_j)$ denote its paired path.

Moreover, let $\{T_1, \ldots, T_{k_2}\} \subseteq \{S_1, \ldots, S_k\}$ be the set of $k_2 \leq k$ paths having exactly four vertices (i.e. two degree 2-vertices vertices and two non-degree 2-vertices). We assume that sets $\{T_1, \ldots, T_{k_2}\}$ are ordered following the same order as $\{S_1, \ldots, S_k\}$. Each set T_j is also paired with another set $q(T_j)$ in such a way that, if possible, pairings p and q are different. This can be done as follows: for every $i = 1 \bmod 2$, $1 \leq i \leq k_2 - 3$, we pair T_i with T_{i+3}; if k_2 is even, T_2 is paired with T_{k_2-1}, otherwise T_2 is paired with T_{k_2} but T_{k_2-2} is not paired at all. Note that unless $k_2 \leq 2$ and $k \leq 3$, each T_i has $p(T_i) \neq q(T_i)$.

Now, for each pair of paired sets $S_i, p(S_i)$ and $T_j, q(T_j)$, we will construct the new paths of $\mathcal{P}$ depending on the parity of each $|S_i|$ and the relative position of $S_i, p(S_i)$ and $T_j, q(T_j)$ in T. For each S_i, there will be a set P_i of paths within S_i. For each pair $S_i, p(S_i)$, there will be a single path going from a specific vertex x_i of S_i to a vertex $x_{i'}$ of $p(S_i) = S_{i'}$. Similarly, for each pair $T_j, q(T_j)$, there will be a single path going from a specific vertex y_j of T_j to a vertex $y_{j'}$ of $q(T_j) = T_{j'}$. We describe how to determine P_i, x_i (and y_i if $|S_i| = 4$) for each S_i:

(a) If $|S_i|$ is odd and the unique path from S_i to $p(S_i)$ goes through s^i_1, then $P_i = \{s^i_j - s^i_{j+\lceil |S_i|/2\rceil-1} \mid 2 \leq j \leq \lceil |S_i|/2\rceil - 1\}$, and $x_i = s^i_{\lceil |S_i|/2\rceil}$.

(b) If $|S_i|$ is odd and the unique path from S_i to $p(S_i)$ goes through $s^i_{|S_i|}$, as in the previous case $P_i = \{s^i_j - s^i_{j+\lceil |S_i|/2\rceil-1} \mid 2 \leq j \leq \lceil |S_i|/2\rceil - 1\}$, but $x_i = s^i_{\lceil |S_i|/2\rceil-1}$.

(c) If $|S_i|$ is even but $|S_i| \neq 4$ and the unique path from S_i to $p(S_i)$ goes through s^i_1, then $P_i = \{s^i_j - s^i_{j+(|S_i|-2)/2} \mid 2 \leq j \leq |S_i|/2\}$, and $x_i = s^i_{|S_i|-1}$.

(d) If $|S_i|$ is even but $|S_i| \neq 4$ and the unique path from S_i to $p(S_i)$ goes through $s^i_{|S_i|}$, as in the previous case $P_i = \{s^i_j - s^i_{j+(|S_i|-2)/2} \mid 2 \leq j \leq |S_i|/2\}$, but $x_i = s^i_1$.

(e) If $|S_i| = 4$ and the paths from S_i to $p(S_i)$ and $q(S_i)$ go through s^i_1 and s^i_4 (or vice-versa), then $x_i = s^i_2$ and $y_i = s^i_4$.

(f) If $|S_i| = 4$ and and the paths from S_i to $p(S_i)$ and $q(S_i)$ both go through s^i_1, then $x_i = s^i_2$ and $y_i = s^i_3$.

(g) If $|S_i| = 4$ and and the paths from S_i to $p(S_i)$ and $q(S_i)$ both go through s^i_4, then $x_i = s^i_3$ and $y_i = s^i_4$.

The construction is illustrated in Figure 3.

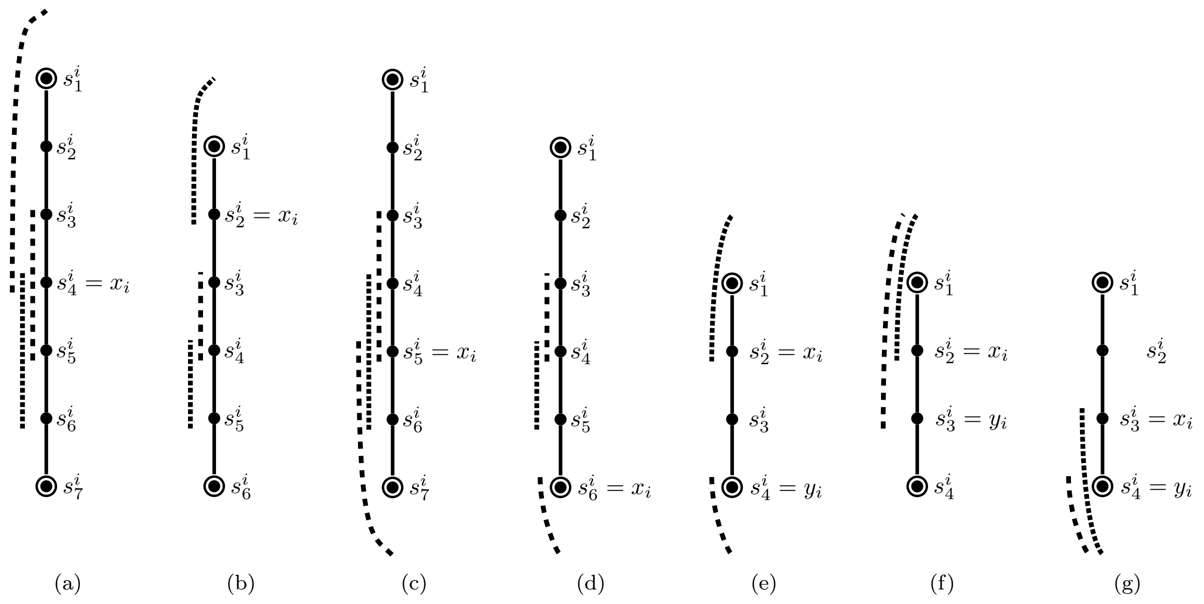


Figure 3: Illustration of the four cases for the additional paths in Case 2 of the proof of Theorem 17, for $|S_i| \in \{4, 6, 7\}$. Circled vertices have degree different from 2. The paths of P_i are marked with dotted and dashed lines.

Now, for each pair $S_i, p(S_i) = S_{i'}$, we add the path $x_i - x_{i'}$, and for each pair $T_j, q(T_{j'})$, we add the path $y_j - y_{j'}$. Observe that if $p(S_i) = q(S_i)$, we need to separate two vertices from $S_i, p(S_i)$: we add one path to do so. Furthermore, if k is odd, S_k is not paired, and we let P_k be a set of at most $\frac{|S_k|+1}{2}$ paths within S_k that can easily constructed using similar techniques. If k_2 is odd, T_{k_2-2} is not paired (note that $S_k \neq T_{k_2-2}$) two vertices of T_{k_2-2} need to be separated, and we add one additional path to do this.

Observe that the construction is similar to the construction for paths in Theorem 14, and all vertices are now identified.

Note that whenever S_i was paired and $|S_i| \neq 4$ is even, S_i has contributed to exactly $\frac{|S_1|-1}{2}$ additional paths in the solution. Similarly, if $|S_i| = 4$ and S_i was paired twice using p and q and $p(S_i) \neq (S_i)$, S_i has contributed to two half-paths, that is, a total of $1 = \frac{|S_i|-2}{2}$. If $|S_i| = 4$ but $p(S_i) = q(S_i)$, S_i has contributed to three half-paths, that is, a total of $2 = \frac{|S_i|}{2}$ (however by our previous remarks this can happen only once in the entire construction). If S_i was paired and $|S_i|$ is odd, S_i has contributed to exactly $\frac{|S_1|-2}{2}$ additional paths. If k is odd and S_k is not paired, S_k has contributed to at most $\frac{|S_k|-1}{2}$ paths. If k_2 is odd and T_{k_2-2} was not paired using pairing q, T_{k_2-2} has contributed to $\frac{3}{2} = \frac{|T_{k_2-2}|-2+1}{2}$ paths.

Let k_e be the number of sets S_i of even size, at least 6. Then, we have added at most $\frac{t+k_e+4}{2}$ paths to $\mathcal{P}'$ to obtain $\mathcal{P}$, proving the second inequality of part (ii) of the statement.

Observing that a set S_i of even size (at least 6) has at least four degree 2-vertices, $k_e \leq \frac{t}{4}$, and we obtain the last inequality of part (ii) of the statement. $\square$

Though there are small trees which do not match the lower bounds of Theorem 17 (such as $K_{1,3}$), we do not know whether the upper bounds are tight. We suspect that they both could be improved.

3.3. An application to all connected graphs

We get the following improvement of Theorem 3 for identifying path covers of connected graphs.

Theorem 18. *For any connected graph G on n vertices, $p^{ID}(G) \leq \lceil \frac{2n}{3} \rceil + 5$.*

Proof. Consider a spanning tree T of G. By Proposition 11, an identifying path cover of T is also one for G. Assume that T has ℓ leaves and t vertices of degree 2. Since $\ell + t \leq n$ and $\frac{5x}{8} \leq \frac{2x}{3}$, we have $\lceil \frac{2\ell}{3} \rceil + \frac{5t}{8} \leq \lceil \frac{2n}{3} \rceil + 1$.

By Theorem 17(ii), T has an identifying path cover of size at most $\lceil \frac{2\ell}{3} \rceil + \frac{5t}{8} + 4 \leq \lceil \frac{2n}{3} \rceil + 5$. $\square$

Unlike for many other variants of identification problems (such as identifying codes, see [9]), Theorem 18 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 $w(G) \leq \frac{2n}{3}$ holds for the size $w(G)$ of a smallest watching system (i.e. an "identifying *star* cover") in any connected graph G on n vertices [1].

3.4. Bounds using domination parameters

The bound of Theorem 18 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 Proposition 11 and Theorem 17 we get the following upper bound.

Theorem 19. *For any connected graph G on n vertices, it holds that $p^{ID}(G) \leq \left\lceil \frac{2(n-\gamma_C(G))}{3} \right\rceil + \frac{5\gamma_C(G)}{8} + 4$.*

For any value k, one can construct a tree T on n vertices with $\gamma_C(T) = k$ and $p^{\mathrm{ID}}(G) \geq \left\lceil \frac{2(n-k)}{3} \right\rceil + \lceil \frac{k-1}{2} \rceil$: build T from a star with $n-k$ leaves by subdividing one edge of this star $k-1$ times. We now have $\gamma_C(T) = k$ and the only minimum connected dominating set includes all vertices but the leaves of T. One can now check that $p^{\mathrm{ID}}(G) \geq \left\lceil \frac{2(n-k)}{3} \right\rceil + \lceil \frac{k-1}{2} \rceil$.

Using the (usual) domination number of a graph G, $\gamma(G)$, we can obtain a new bound as follows:

Theorem 20. *Let G be a graph on n vertices with maximum degree $\Delta \geq 4$. Then $p^{ID}(G) \leq \gamma(G) \lceil \frac{2\Delta}{3} \rceil$.*

Proof. Consider an optimal dominating set D of size $\gamma(G)$. For each vertex v of D, build a set of paths that identifies all vertices of $N(v)$, such as done for the star $K_{1,n-1}$ ($n \geq 5$) in the proof of Theorem 16. This solution has size $\sum_{v \in D} \left\lceil \frac{2N(v)}{3} \right\rceil \leq \gamma(G) \left\lceil \frac{2\Delta}{3} \right\rceil$. $\square$

This bound is tight e.g. for stars, which have domination number 1. Moreover, consider a caterpillar graph C_k^{Δ} built from a path on k vertices with $\Delta - 2$ pendant vertices attached to each vertex of the path; C_k^{Δ} has domination number $\gamma(C_k^{\Delta}) = k$. By Proposition 12, $p^{\mathrm{ID}}(C_k^{\Delta}) \geq \left\lceil \frac{k \cdot 2(\Delta-2)}{3} \right\rceil$, and the bound of Theorem 20 is almost tight for any pair of values of γ and Δ.

4. On the complexity of MIN-IDPC-k

In this section, we discuss the computational complexity of MIN-IDPC-k, that is, we deal with k-paths only (recall that a k-path has k vertices). 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. Moreover, we note that results of [8] combined with a later $\frac{4}{3}$-approximation for max-P_3-P in subcubic graphs without degree 1-vertices [19] implies the existence of a $\frac{9}{8}$-approximation algorithm for MIN-IDPC-2 in this class of graphs.

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 21. 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 [22]. 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 21. *Let P and Q be two optimization problems. An* L-reduction *from P to Q is a four-tuple (f, g, α, β) where f and g are polynomial time computable functions and α, β 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, $opt_Q(f(x)) \leq \alpha \cdot opt_P(x)$.*
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 $|opt_P(x) - cost_P(x, g(f(x), y'))| \leq \beta \cdot |opt_Q(f(x)) - cost_Q(f(x), y')|$.*

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 22. *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 Theorem 22, 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 Figure 4 for an example when $k = 3$ and Figure 5 when $k = 4$, and Figure 6 for their succinct representation) and *k-gadget of type B* (see Figure 7). Both gadgets include an *attachment vertex* which will be merged 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 Figure 7, 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 Figure 8). 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 k-path from outside the gadget to go through the attached vertex.

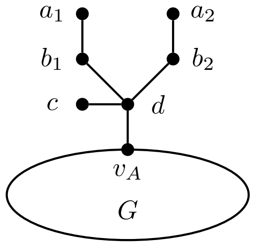


Figure 4: The 3-gadget of type A.

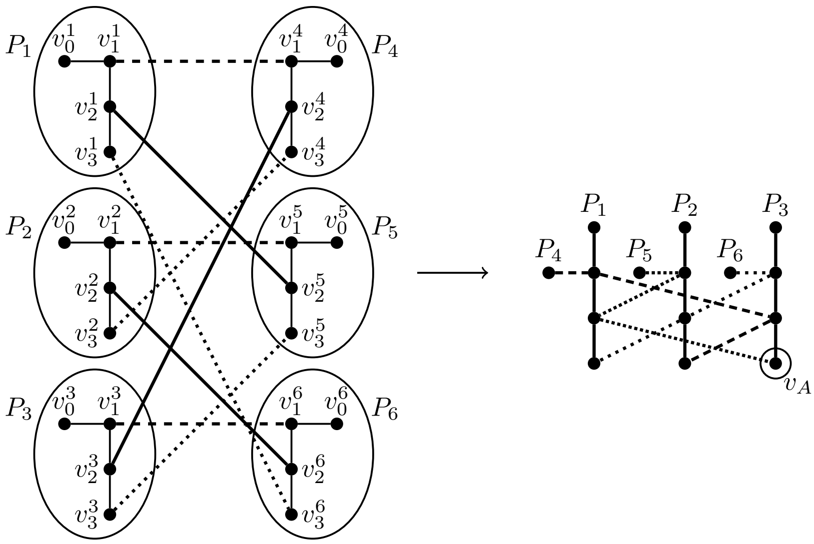


Figure 5: Construction of the 4-gadget of type A.

In order to construct the k-gadget of type A, we use a construction of an extremal graph for the lower bound $\frac{2n}{k+1} \leq p_k^{\mathrm{ID}}(G)$ (Theorem 5). This construction is detailed in Definition 23.

Definition 23. *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 Figure 4. When $k \geq 4$, the k-gadget of type A is constructed as follows (see Figure 5):*

- *Let $P_1, \ldots, P_{2(k-1)}$ be $2(k-1)$ vertex-disjoint k-paths, where for $i \in \{1, \ldots, 2(k-1)\}$, $P_i = \{v_0^i, \ldots, v_{k-1}^i\}$.*
- *Consider the complete bipartite graph B whose vertices are $P_1, \ldots, P_{2(k-1)}$. It is $(k-1)$-regular.*
- *Select a $(k-1)$-edge-colouring of B with colours from $\{1, \ldots, k-1\}$ (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 $v_{\lfloor k/2 \rfloor}^j$).*
- *Let the attachment vertex v_A be vertex v_{k-1}^i for some arbitrary $i \in \{1, \ldots, 2(k-1)\}$.*

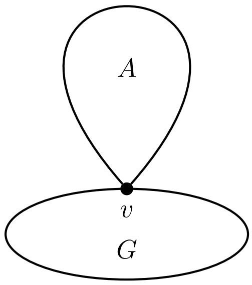


Figure 6: Representing a k-gadget of type A.

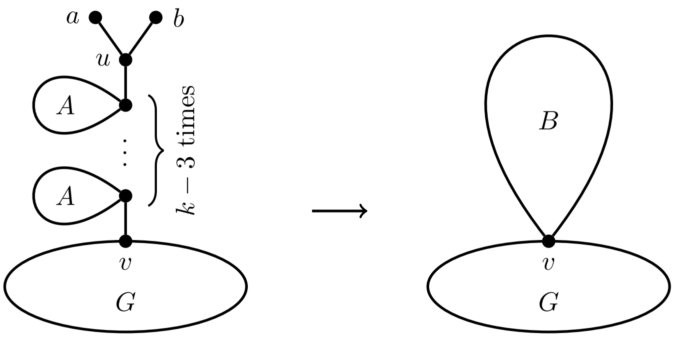
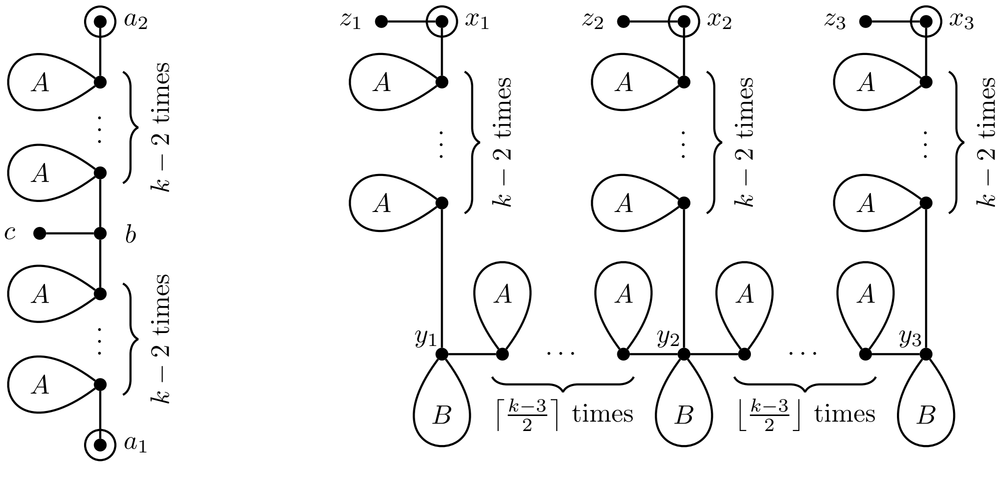


Figure 7: The k-gadget of type B and its succinct representation.

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 Figure 8). 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.



(a) Edge k-gadget (b) Vertex k-gadget

Figure 8: 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 k-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 the k-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 24, 25 and 26 about the gadgets of type A and B.

Claim 24. *Let A be a 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 k-path 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 $v^i_{\lfloor k/2 \rfloor}$ or $v^i_{\lceil k/2 \rceil}$ 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 $v^i_{\lfloor k/2 \rfloor}$ or $v^i_{\lceil k/2 \rceil}$. 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. □

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

Claim 25. *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 k-paths of $\mathcal{P}$ having an endpoint in A, and none of these k-paths can reach a vertex outside of A. Moreover, there is a set of x_A k-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 3-path to separate either a_1 from b_1 or a_2 from b_2. Finally, the four 3-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 24: no k-path can be used to cover two degree 1 vertices in A, and no k-path starting outside of A can cover a degree 1 vertex of A. Considering the x_A k-paths of the construction of A proves the second part of the statement. ◻

Claim 26. *Let B be a k-gadget of type B attached at some vertex v in G. Then, at least $x_B = x_A(k-3)+2$ k-paths of $\mathcal{P}$ are entirely contained in B. Moreover, if exactly x_B k-paths of $\mathcal{P}$ are entirely contained in B, then there is an additional k-path of $\mathcal{P}$ containing vertex v. Finally, there exists such a set of x_B k-paths.*

Proof. Following Claim 25, we need at least x_A k-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 k-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 25, among these k-paths, only k-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 25 for each copy of the k-gadget of type A together with the k-paths $a \ldots v$ and $b \ldots v$, we get the last part of the claim. ◻

We are now ready to prove Theorem 22.

Proof of Theorem 22. We first prove that the first part of Definition 21 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 25 (resp. of size x_B of Claim 26) into $\mathcal{P}$. Now, for each edge e of G, add an arbitrary k-path starting in vertex c of G_e. For each vertex v of G, add three arbitrary k-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 k-path $y_1 \ldots y_3$ of G_v, as well as k-paths $x_2 \ldots b_1$, $x_3 \ldots b_2$ and $x_1 \ldots b_3$ to $\mathcal{P}$. If $v \notin C^*$, add k-paths $x_1 \ldots y_1$, $x_2 \ldots y_2$ and $x_3 \ldots y_3$ to $\mathcal{P}$.

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

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

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: $p_k^{\mathrm{ID}}(G) \leq |\mathcal{P}| \leq (16 + (14k-30)x_A + 6x_B)|C^*|$ and hence (1) of Definition 21 is fulfilled with $\alpha = 16 + (14k-30)x_A + 6x_B \leq 40k^2 - 116k + 47$.

It remains to prove the second part of Definition 21. 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 25, each gadget of type A contains at least x_A k-paths of $\mathcal{P}$, and by Claim 26, each gadget of type B contains at least $x_B = x_A(k-3)+2$ k-paths of $\mathcal{P}$. Moreover, in each vertex gadget, at least three k-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 26, if there are exactly that many k-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 k-paths of $\mathcal{P}$ that we already considered, in any vertex-gadget, at least three k-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 $x_1 \ldots y_1$, $x_2 \ldots y_2$, $x_3 \ldots y_3$. We construct C as follows: for each vertex v of G, if there are at least four such additional k-paths in G_v, we add v to C. Set C is a vertex cover of G: indeed, when $v \in C$, the k-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 25 and 26. Moreover, the (at least) four additional k-paths of $\mathcal{P}$ in G_v could be replaced by $y_1 \ldots y_3$ and three k-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 k-path cover $\mathcal{P}'$ with $\mathcal{P}' \leq \mathcal{P}$, and since all vertices labelled b are covered by a k-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 $|C^*| \leq |C| \leq p_k^{\mathrm{ID}}(G) - (2(k-2)x_A+1)m - ((3(k-2)+k-3)x_A + 3x_B + 6)\, n$. Together with Equation (1), this implies:

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

From Equations (1) and (2), we get $|C|-|C^*| \leq |\mathcal{P}|-p_k^{\mathrm{ID}}(G)$, which implies $||C^*|-|C|| \leq |p_k^{\mathrm{ID}}(G)-|\mathcal{P}||$; hence (2) of Definition 21 is fulfilled with $\beta = 1$. □

5. Conclusion and open problems

We conclude with some open problems.

We gave a formula for an almost-tight upper bound on p^{ID} for trees, and how to construct in polynomial time a solution of this size. It would be interesting to determine the exact value and/or an exact algorithm for trees.

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.

Moreover, when k is not fixed but part of the input, it is not even clear whether one can check in polynomial time whether a given graph is k-path identifiable. When $k = n$ for example, such graph needs to be Hamiltonian. This suggests that it might be NP-hard to decide if a given graph G is k-path identifiable for some input value $k = k(G)$.

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.

Finally, it would be a natural model to measure the quality of the solution using the *sum* of the lengths of the paths of the identifying path cover, instead of their number.

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 19, and the anonymous referee for pointing out some mistakes and making useful comments.

[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 and M. Kovše. On graph identification problems and the special case of identifying vertices using paths. To appear in *Proceedings of the International Workshop on Combinatorial Algorithms, IWOCA 2012.*

[12] 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.

[13] 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.

[14] M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[15] 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.

[16] D. S. Johnson, Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.* 9, 256–278, 1974.

[17] C. Jordan. Sur les assemblages de lignes (in French). *Journal für die reine und angewandte Mathematik* 70(2):185–190, 1869.

[18] 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.

[19] A. Kosowski, M. Małafiejski and P. Zyliński. An approximation algorithm for maximum P3-packing in subcubic graphs. *Inf. Proc. Letters* 99:230–233, 2006.
[20] 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.
[21] B. M. E. Moret, H. D. Shapiro. On minimizing a set of tests. *SIAM J. Sci. Stat. Comp.* 6(4):983–1003, 1985.
[22] C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
[23] P. Rosendahl. On the identification of vertices using cycles. *Electron. J. Comb.* 10:P1, 2003.