

On the identification of vertices in graphs using paths

Matjaž Kovše^{1,2,3}

joint work with Florent Foucaud^{1,3}

¹LaBRI - Université Bordeaux 1 - CNRS ²& IMFM, Slovenia

³This research is supported by the ANR Project IDEA, ANR-08-EMER-007, 2009-2011.

Bordeaux Workshop on Identifying Codes, November 21-25, 2011

Outline

- 1 Introduction, definitions, examples
- 2 Identifying vertices in trees using paths
- 3 Lower and upper bounds
- 4 Open problems and ideas

Motivation and comparison of static vs. dynamic version of identification

classical (static) identification of vertices (using r -balls)

simple and static detectors in a network that are able to detect an event in a neighborhood of the sensor

Motivation and comparison of static vs. dynamic version of identification

classical (static) identification of vertices (using r -balls)

simple and static detectors in a network that are able to detect an event in a neighborhood of the sensor

identification of vertices using different subgraphs (dynamic scenario)

moving detectors in a network that are able to detect an event in the area where they are ascribed to move

Identifying vertices with sets

A set $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ of subgraphs of a graph G is said to *identify* the vertices of G if all sets $I(v, \mathcal{C}) = \{i \mid v \in C_i\}$ are nonempty and for any pair of distinct vertices u, v of G , $I(u, \mathcal{C}) \neq I(v, \mathcal{C})$.

Identifying vertices with sets

A set $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ of subgraphs of a graph G is said to *identify* the vertices of G if all sets $I(v, \mathcal{C}) = \{i \mid v \in C_i\}$ are nonempty and for any pair of distinct vertices u, v of G , $I(u, \mathcal{C}) \neq I(v, \mathcal{C})$.

If we restrict \mathcal{C} to be a subset of a given family \mathcal{G} of subgraphs of G , we denote by $ID_{\mathcal{G}}(G)$, the minimum number of subgraphs of G from \mathcal{G} needed to identify all vertices of G .

Identifying vertices with sets

A set $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ of subgraphs of a graph G is said to *identify* the vertices of G if all sets $I(v, \mathcal{C}) = \{i \mid v \in C_i\}$ are nonempty and for any pair of distinct vertices u, v of G , $I(u, \mathcal{C}) \neq I(v, \mathcal{C})$.

If we restrict \mathcal{C} to be a subset of a given family \mathcal{G} of subgraphs of G , we denote by $ID_{\mathcal{G}}(G)$, the minimum number of subgraphs of G from \mathcal{G} needed to identify all vertices of G .

If we take as \mathcal{G} to be the family of all maximal stars of G , we get the definition of identifying codes, introduced by M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin in 1998, and widely studied afterwards.

Identification using cycles

The case when \mathcal{G} equals the family of all cycles or closed walks of G , has been studied for special graph classes (hypercubes, d -dimensional grids, torii) by I. Honkala, M. G. Karpovsky, and S. Litsyn (2003), P. Rosendahl (2003 & 2004).

Identification using cycles

The case when \mathcal{G} equals the family of all cycles or closed walks of G , has been studied for special graph classes (hypercubes, d -dimensional grids, torii) by I. Honkala, M. G. Karpovsky, and S. Litsyn (2003), P. Rosendahl (2003 & 2004).

Theorem (I. Honkala, M. G. Karpovsky, and S. Litsyn (2003) and P. Rosendahl (2003))

Let G be a d -dimensional hypercube (on $n = 2^d$ vertices). Then $ID_{\mathcal{C}}(G) = \lceil \log_2(n + 1) \rceil = d + 1$.

Identification using cycles

The case when \mathcal{G} equals the family of all cycles or closed walks of G , has been studied for special graph classes (hypercubes, d -dimensional grids, torii) by I. Honkala, M. G. Karpovsky, and S. Litsyn (2003), P. Rosendahl (2003 & 2004).

Theorem (I. Honkala, M. G. Karpovsky, and S. Litsyn (2003) and P. Rosendahl (2003))

Let G be a d -dimensional hypercube (on $n = 2^d$ vertices). Then $ID_{\mathcal{C}}(G) = \lceil \log_2(n + 1) \rceil = d + 1$.

Proof.

Just take hamiltonian cycles in precisely one member in each pair of halfcubes (subhypercubes of dimension $d - 1$) and recall that intersection of any collection of k halfcubes of d -dimensional hypercube is always a $(d - k)$ -dimensional hypercube. □

Identification using paths

In this talk, we present some results when \mathcal{G} equals the family \mathcal{P} of all paths of a graph G .

(as proposed by P. Slater and J.L. Sewell at CID 2009)

Identification using paths

In this talk, we present some results when \mathcal{G} equals the family \mathcal{P} of all paths of a graph G .

(as proposed by P. Slater and J.L. Sewell at CID 2009)

Vertices of any graph can be identified this way (just take "0-length" paths (vertices) or for a graph without isolated vertices and no K_2 components take edges and longer paths). In other words there are no "path"-twins.

Identification using paths

In this talk, we present some results when \mathcal{G} equals the family \mathcal{P} of all paths of a graph G .

(as proposed by P. Slater and J.L. Sewell at CID 2009)

Vertices of any graph can be identified this way (just take "0-length" paths (vertices) or for a graph without isolated vertices and no K_2 components take edges and longer paths). In other words there are no "path"-twins.

We denote by $ID_{\mathcal{P}}(G)$ the minimum number of paths needed to identify all vertices of a graph G .

Identification using paths

In this talk, we present some results when \mathcal{G} equals the family \mathcal{P} of all paths of a graph G .

(as proposed by P. Slater and J.L. Sewell at CID 2009)

Vertices of any graph can be identified this way (just take "0-length" paths (vertices) or for a graph without isolated vertices and no K_2 components take edges and longer paths). In other words there are no "path"-twins.

We denote by $ID_{\mathcal{P}}(G)$ the minimum number of paths needed to identify all vertices of a graph G .

Graphs of maximum degree 2

Proposition

Let G be a connected graph of maximum degree 2 having m edges and l vertices of degree one. Then $ID_{\mathcal{P}}(G) \geq \lceil \frac{m+l}{2} \rceil$.

Proposition

Let $n \geq 1$ be an integer and P_n , the path on n vertices.
Then $ID_{\mathcal{P}}(P_n) = \lceil \frac{n+1}{2} \rceil$.

Graphs of maximum degree 2

Proposition

Let G be a connected graph of maximum degree 2 having m edges and l vertices of degree one. Then $ID_{\mathcal{P}}(G) \geq \lceil \frac{m+l}{2} \rceil$.

Proposition

Let $n \geq 1$ be an integer and P_n , the path on n vertices.
Then $ID_{\mathcal{P}}(P_n) = \lceil \frac{n+1}{2} \rceil$.

Proposition

Let $n \geq 3$ be an integer and C_n the cycle on n vertices.

- $ID_{\mathcal{P}}(C_3) = 2$
- $ID_{\mathcal{P}}(C_4) = 3$
- For $n \geq 5$, $ID_{\mathcal{P}}(C_n) = \lceil \frac{n}{2} \rceil$

Stars

Proposition

Let $n \geq 4$ be an integer and $K_{1,n-1}$, a star on n vertices.

Then $ID_{\mathcal{P}}(K_{1,n-1}) = \lceil \frac{2(n-1)}{3} \rceil$.

Topologically irreducible trees

Topologically irreducible trees are trees with no vertices of degree 2.

Topologically irreducible trees

Topologically irreducible trees are trees with no vertices of degree 2.

Theorem

Let T be a topologically irreducible tree on at least 5 vertices, and let ℓ denote the number of leaves of T . Then $ID_{\mathcal{P}}(T) = \lceil \frac{2\ell}{3} \rceil$.

Topologically irreducible trees

Topologically irreducible trees are trees with no vertices of degree 2.

Theorem

Let T be a topologically irreducible tree on at least 5 vertices, and let ℓ denote the number of leaves of T . Then $ID_{\mathcal{P}}(T) = \lceil \frac{2\ell}{3} \rceil$.

Sketch of a proof.

For the lower bound, observe that identifying set of paths for a tree T , will also identify all vertices of a star $K_{1,\ell}$ obtained by contracting all non-leaf vertices into a single vertex.

Topologically irreducible trees

Topologically irreducible trees are trees with no vertices of degree 2.

Theorem

Let T be a topologically irreducible tree on at least 5 vertices, and let ℓ denote the number of leaves of T . Then $ID_{\mathcal{P}}(T) = \lceil \frac{2\ell}{3} \rceil$.

Sketch of a proof.

For the lower bound, observe that identifying set of paths for a tree T , will also identify all vertices of a star $K_{1,\ell}$ obtained by contracting all non-leaf vertices into a single vertex.

To prove that $\lceil \frac{2\ell}{3} \rceil$ paths is enough we give a constructive procedure how to obtain identifying set of paths of appropriate size.

Topologically irreducible trees

Topologically irreducible trees are trees with no vertices of degree 2.

Theorem

Let T be a topologically irreducible tree on at least 5 vertices, and let ℓ denote the number of leaves of T . Then $ID_{\mathcal{P}}(T) = \lceil \frac{2\ell}{3} \rceil$.

Sketch of a proof.

For the lower bound, observe that identifying set of paths for a tree T , will also identify all vertices of a star $K_{1,\ell}$ obtained by contracting all non-leaf vertices into a single vertex.

To prove that $\lceil \frac{2\ell}{3} \rceil$ paths is enough we give a constructive procedure how to obtain identifying set of paths of appropriate size. We start from the center of a tree (which is always either a vertex or an edge (by Jordan's theorem) and we extend the set of identifying paths level by level).

Topologically irreducible trees

Topologically irreducible trees are trees with no vertices of degree 2.

Theorem

Let T be a topologically irreducible tree on at least 5 vertices, and let ℓ denote the number of leaves of T . Then $ID_{\mathcal{P}}(T) = \lceil \frac{2\ell}{3} \rceil$.

Sketch of a proof.

For the lower bound, observe that identifying set of paths for a tree T , will also identify all vertices of a star $K_{1,\ell}$ obtained by contracting all non-leaf vertices into a single vertex.

To prove that $\lceil \frac{2\ell}{3} \rceil$ paths is enough we give a constructive procedure how to obtain identifying set of paths of appropriate size. We start from the center of a tree (which is always either a vertex or an edge (by Jordan's theorem) and we extend the set of identifying paths level by level). The crucial observation is that a vertex in a new level is either a leaf or a branching and that once a vertex is identified it will remain so during our procedure. \square

General trees

Theorem

Let T be a tree on n vertices, and let ℓ denote the number of leaves of T and d the number of vertices of degree 2. Then $ID_{\mathcal{P}}(T) = \lceil \frac{2\ell}{3} \rceil + f(\ell, d)$. Moreover $f(\ell, d) \approx \lceil \frac{d}{2} \rceil$ and $f(\ell, d)$ and therefore also $ID_{\mathcal{P}}(T)$ can be computed in $O(n)$ time.

General trees

Theorem

Let T be a tree on n vertices, and let ℓ denote the number of leaves of T and d the number of vertices of degree 2. Then $ID_{\mathcal{P}}(T) = \lceil \frac{2\ell}{3} \rceil + f(\ell, d)$. Moreover $f(\ell, d) \approx \lceil \frac{d}{2} \rceil$ and $f(\ell, d)$ and therefore also $ID_{\mathcal{P}}(T)$ can be computed in $O(n)$ time.

The main ideas of the proof

First we solve a problem for the corresponding topologically irreducible tree. Just further solving a problem separately for all subdivided edges would give us an upper bound for $ID_{\mathcal{P}}(T)$. In cases when $3 \nmid \ell$ and there are edges that are subdivided even number of times we can lower the number of paths we need to identify all vertices of T .

Upper and lower bound for $ID_{\mathcal{P}}(G)$ in terms of n

Theorem

Let G be a graph on n vertices. Then $\lceil \log_2(n+1) \rceil \leq ID_{\mathcal{P}}(G) \leq \lceil \frac{2(n-1)}{3} \rceil$.

Upper and lower bound for $ID_{\mathcal{P}}(G)$ in terms of n

Theorem

Let G be a graph on n vertices. Then $\lceil \log_2(n+1) \rceil \leq ID_{\mathcal{P}}(G) \leq \lceil \frac{2(n-1)}{3} \rceil$.

Proof.

Lower bound - use usual theoretic information argument (the cardinality of the power set of a set with n elements is 2^n , where one of the sets is an empty set)

Upper and lower bound for $ID_{\mathcal{P}}(G)$ in terms of n

Theorem

Let G be a graph on n vertices. Then $\lceil \log_2(n+1) \rceil \leq ID_{\mathcal{P}}(G) \leq \lceil \frac{2(n-1)}{3} \rceil$.

Proof.

Lower bound - use usual theoretic information argument (the cardinality of the power set of a set with n elements is 2^n , where one of the sets is an empty set)

Upper bound - observe that for a spanning subgraph H of G , we have $ID_{\mathcal{P}}(G) \leq ID_{\mathcal{P}}(H)$ and the bound follows from results about trees. \square

Upper and lower bound for $ID_{\mathcal{P}}(G)$ in terms of n

Theorem

Let G be a graph on n vertices. Then $\lceil \log_2(n+1) \rceil \leq ID_{\mathcal{P}}(G) \leq \lceil \frac{2(n-1)}{3} \rceil$.

Proof.

Lower bound - use usual theoretic information argument (the cardinality of the power set of a set with n elements is 2^n , where one of the sets is an empty set)

Upper bound - observe that for a spanning subgraph H of G , we have $ID_{\mathcal{P}}(G) \leq ID_{\mathcal{P}}(H)$ and the bound follows from results about trees. \square

There are graphs attaining lower bound (hypercubes, complete graphs,...) and also graphs attaining the upper bound (stars on at least 5 vertices).

Upper and lower bound for $ID_{\mathcal{P}}(G)$ in terms of n

Theorem

Let G be a graph on n vertices. Then $\lceil \log_2(n+1) \rceil \leq ID_{\mathcal{P}}(G) \leq \lceil \frac{2(n-1)}{3} \rceil$.

Proof.

Lower bound - use usual theoretic information argument (the cardinality of the power set of a set with n elements is 2^n , where one of the sets is an empty set)

Upper bound - observe that for a spanning subgraph H of G , we have $ID_{\mathcal{P}}(G) \leq ID_{\mathcal{P}}(H)$ and the bound follows from results about trees. \square

There are graphs attaining lower bound (hypercubes, complete graphs,...) and also graphs attaining the upper bound (stars on at least 5 vertices).

One can also characterize graphs attaining both bounds.

Open and related problems, ideas

- The complexity of determining $ID_{\mathcal{P}}(G)$ for an arbitrary graph.

Open and related problems, ideas

- The complexity of determining $ID_{\mathcal{P}}(G)$ for an arbitrary graph.
- Bounding the sizes of paths used for identification.

Open and related problems, ideas

- The complexity of determining $ID_{\mathcal{P}}(G)$ for an arbitrary graph.
- Bounding the sizes of paths used for identification.
- Similar approach for cactus graphs, block graphs, other graph families

Open and related problems, ideas

- The complexity of determining $ID_{\mathcal{P}}(G)$ for an arbitrary graph.
- Bounding the sizes of paths used for identification.
- Similar approach for cactus graphs, block graphs, other graph families
- Use other subgraph families for identification

Open and related problems, ideas

- The complexity of determining $ID_{\mathcal{P}}(G)$ for an arbitrary graph.
- Bounding the sizes of paths used for identification.
- Similar approach for cactus graphs, block graphs, other graph families
- Use other subgraph families for identification
- Characterizing twins for different types of identification with subgraphs of a given graph.

Open and related problems, ideas

- The complexity of determining $ID_{\mathcal{P}}(G)$ for an arbitrary graph.
- Bounding the sizes of paths used for identification.
- Similar approach for cactus graphs, block graphs, other graph families
- Use other subgraph families for identification
- Characterizing twins for different types of identification with subgraphs of a given graph.