Identifying vertices of a graph using paths

Florent Foucaud (PhD student at LaBRI, Bordeaux, France)
and
Matjaž Kovše (Postdoc at Universität Leipzig, Germany)

Kalasalingam University
Tamil Nadu, India

19-21 July 2012

IWOCA 2012

The test cover problem (TCP)

Definition - Test cover problem (mentioned in Garey, Johnson, 1979)

INPUT: a set system (or hypergraph) $(X, \mathcal{S})$
PROBLEM: find the minimum subset $\mathcal{T} \subseteq \mathcal{S}$ such that each element $x \in X$ belongs to a different set of sets in $\mathcal{T}$.

The test cover problem (TCP)

Definition - Test cover problem (mentioned in Garey, Johnson, 1979)

INPUT: a set system (or hypergraph) $(X, \mathcal{S})$
PROBLEM: find the minimum subset $\mathcal{T} \subseteq \mathcal{S}$ such that each element $x \in X$ belongs to a different set of sets in $\mathcal{T}$.

Remark

Equivalently: for any pair x, y of elements of X, there is a set in $\mathcal{T}$ that contains **exactly** one of x, y.

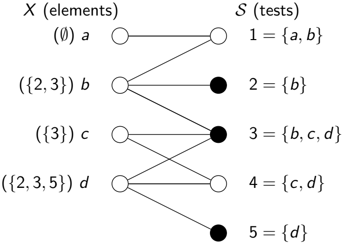
The test cover problem (TCP)

Definition - Test cover problem (mentioned in Garey, Johnson, 1979)

INPUT: a set system (or hypergraph) $(X, \mathcal{S})$
PROBLEM: find the minimum subset $\mathcal{T} \subseteq \mathcal{S}$ such that each element $x \in X$ belongs to a different set of sets in $\mathcal{T}$.

Remark

Equivalently: for any pair x, y of elements of X, there is a set in $\mathcal{T}$ that contains **exactly** one of x, y.



example: $\mathcal{T} = \{2, 3, 5\}$

The identification problem (IDP)

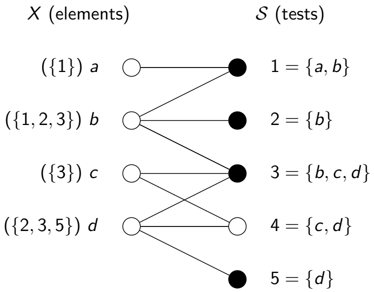
Definition - Identification problem

INPUT: a set system (or hypergraph) $(X, \mathcal{S})$
PROBLEM: find the minimum subset $\mathcal{T} \subseteq \mathcal{S}$ such that each element $x \in X$ belongs to a different **nonempty** set of sets in $\mathcal{T}$.

The identification problem (IDP)

Definition - Identification problem

INPUT: a set system (or hypergraph) $(X, \mathcal{S})$
PROBLEM: find the minimum subset $\mathcal{T} \subseteq \mathcal{S}$ such that each element $x \in X$ belongs to a different **nonempty** set of sets in $\mathcal{T}$.



example: $\mathcal{T} = \{1, 2, 3, 5\}$

Motivation

- Fault analysis: tests are fault-detectors
- medical diagnostics: tests are tests for diseases
- biological identification: tests are attributes

General bounds

Theorem (Folklore)

Given a set system $(X, \mathcal{S})$, a solution to the TCP has size at least $\log_2(|X|)$. A solution to the IDP has size at least $\log_2(|X|+1)$. These bounds are tight.

Proof: Must assign to each element of X, a distinct subset of $\mathcal{T}$. Hence $|X| \leq 2^{|\mathcal{T}|}$ (TCP) and $|X| \leq 2^{|\mathcal{T}|} - 1$ (IDP). □

General bounds

Theorem (Folklore)

Given a set system $(X, \mathcal{S})$, a solution to the TCP has size at least $\log_2(|X|)$. A solution to the IDP has size at least $\log_2(|X|+1)$. These bounds are tight.

Proof: Must assign to each element of X, a distinct subset of $\mathcal{T}$.
Hence $|X| \leq 2^{|\mathcal{T}|}$ (TCP) and $|X| \leq 2^{|\mathcal{T}|} - 1$ (IDP). □

Theorem (Bondy's theorem, 1972)

Given a set system $(X, \mathcal{S})$, a minimal solution to the TCP has size at most $|X| - 1$. A minimal solution to the IDP has size at most $|X|$. These bounds are tight.

Proof: TCP: nice graph-theoretic argument.
IDP: sizes of solutions to TCP and IDP differ by at most 1! □

TCP-k and IDP-k

Definition - k-bounded Test Cover Problem and Identification Problem

INPUT: a set system $(X, \mathcal{S})$ such that
each test has size at most k
PROBLEM: find the minimum subset $\mathcal{T} \subseteq \mathcal{S}$ such that each element $x \in X$ belongs to a different (nonempty) set of sets in $\mathcal{T}$.

TCP-k and IDP-k

Definition - k-bounded Test Cover Problem and Identification Problem

INPUT: a set system $(X, \mathcal{S})$ such that
each test has size at most k
PROBLEM: find the minimum subset $\mathcal{T} \subseteq \mathcal{S}$ such that each element $x \in X$ belongs to a different (nonempty) set of sets in $\mathcal{T}$.

Theorem (Moret and Shapiro, 1985)

Given a k-bounded set system $(X, \mathcal{S})$, a solution to the TCP or IDP has size at least $\frac{2|X|}{k+1}$. This bound is tight.

Proof: i_1: elements belonging to 1 test of $\mathcal{T}$; i_2: elements in at least 2 tests

$i_1 \leq |\mathcal{T}|$, $i_2 \leq \frac{|\mathcal{T}|k - i_1}{2}$

$|X| = i_1 + i_2 \leq |\mathcal{T}| + \frac{|\mathcal{T}|k - i_1}{2} = \frac{|\mathcal{T}|(k+1)}{2}$ □

Complexity results

Theorem (Garey, Johnson, 1979)

TCP is NP-complete.

Theorem (Charon, Cohen, Hudry, Lobstein, 2008)

IDP is NP-complete (even in "planar" set systems).

Complexity results

Theorem (Garey, Johnson, 1979)

TCP is NP-complete.

Theorem (Charon, Cohen, Hudry, Lobstein, 2008)

IDP is NP-complete (even in “planar” set systems).

Theorem (De Bontridder, Haldorsson, Haldorsson, Hurkens, Lenstra, Ravi, Stougie, 2003)

TCP is $O(\log(|X|))$-approximable, but NP-hard to approximate within $o(log(|X|))$. TCP-k is $O(\log(k))$-approximable.

Proof: Reductions from and to SET-COVER and k-BOUNDED SET COVER. □

Complexity results

Theorem (Garey, Johnson, 1979)

TCP is NP-complete.

Theorem (Charon, Cohen, Hudry, Lobstein, 2008)

IDP is NP-complete (even in “planar” set systems).

Theorem (De Bontridder, Haldorsson, Haldorsson, Hurkens, Lenstra, Ravi, Stougie, 2003)

TCP is $O(\log(|X|))$-approximable, but NP-hard to approximate within $o(log(|X|))$. TCP-k is $O(\log(k))$-approximable.

Proof: Reductions from and to SET-COVER and k-BOUNDED SET COVER. □

Remark: The same holds for IDP and IDP-k.

Special cases of IDP

Rich literature (250+ publications) on variants arising from **graph theory**:

Definition - Identifying codes (Karpovsky, Chakrabarty, Levitin, 1998)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **closed neighbourhoods** in G (a vertex identifies its neighbours and itself).

Special cases of IDP

Rich literature (250+ publications) on variants arising from **graph theory**:

Definition - Identifying codes (Karpovsky, Chakrabarty, Levitin, 1998)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **closed neighbourhoods** in G (a vertex identifies its neighbours and itself).

Definition - Watching systems (Auger, Charon, Hudry, Lobstein, 2010+)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **stars** in G (a vertex identifies a part of its neighbourhood).

Special cases of IDP

Rich literature (250+ publications) on variants arising from **graph theory**:

Definition - Identifying codes (Karpovsky, Chakrabarty, Levitin, 1998)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **closed neighbourhoods** in G (a vertex identifies its neighbours and itself).

Definition - Watching systems (Auger, Charon, Hudry, Lobstein, 2010+)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **stars** in G (a vertex identifies a part of its neighbourhood).

Motivation: fault-detection in computer networks or location of threats in facilities

The path identifying cover problem

Definition - Identifying path cover (F., Kovše)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **paths** in G: we look for a set $\mathcal{P}$ of paths such that:

- each vertex belongs to some path in $\mathcal{P}$, and
- for each pair x, y of vertices, we have some path $P_{x,y}$ in $\mathcal{P}$ that includes **exactly one** of x, y.

The path identifying cover problem

Definition - Identifying path cover (F., Kovše)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **paths** in G: we look for a set $\mathcal{P}$ of paths such that:

- each vertex belongs to some path in $\mathcal{P}$, and
- for each pair x, y of vertices, we have some path $P_{x,y}$ in $\mathcal{P}$ that includes **exactly one** of x, y.

Possible motivations: laser-like sensor systems or patrolling robots in facilities/networks

The path identifying cover problem

Definition - Identifying path cover (F., Kovše)

Given a graph G, it is the IDP problem where $X = V(G)$ and $\mathcal{S}$ is the set of **paths** in G: we look for a set $\mathcal{P}$ of paths such that:

- each vertex belongs to some path in $\mathcal{P}$, and
- for each pair x, y of vertices, we have some path $P_{x,y}$ in $\mathcal{P}$ that includes **exactly one** of x, y.

Possible motivations: laser-like sensor systems or patrolling robots in facilities/networks

Notation - $p^{\mathrm{ID}}(G)$

minimum number of paths needed in an identifying path cover.

Example: paths and cycles

To distinguish two adjacent vertices: need a path stopping at one of them.

P_n, C_n: the path and cycle on n vertices.

Theorem (F., Kovše)

We have:

- $p^{\mathrm{ID}}(P_n) = \lceil \frac{n+1}{2} \rceil$
- $p^{\mathrm{ID}}(C_3) = 2$, $p^{\mathrm{ID}}(C_4) = 3$
- for $n \geq 5$, $p^{\mathrm{ID}}(C_n) = \lceil \frac{n}{2} \rceil$

Example: stars

The optimum is roughly to cover three leaves using two paths.

$K_{1,n-1}$: star on n vertices.

Theorem (F., Kovše)

We have $p^{\text{ID}}(K_{1,n-1}) = \lceil \frac{2(n-1)}{3} \rceil$.

Trees

Same idea can be used for trees.
T: tree with ℓ leaves, t degree 2 vertices.

Identify all vertices that are not of degree 2 using $\lceil \frac{2(\ell-1)}{3} \rceil$ paths.
Intuition: first contract the tree to a star, then de-contract it level by level; at each step, re-route paths accordingly.

Degree 2 vertices can be identified using at most $\lceil \frac{t}{2} \rceil$ additional paths.

Theorem (F., Kovše)

$p^{\mathrm{ID}}(T) \leq \lceil \frac{2\ell}{3} \rceil + \lceil \frac{t}{2} \rceil$

Trees

Same idea can be used for trees.
T: tree with ℓ leaves, t degree 2 vertices.

Identify all vertices that are not of degree 2 using $\lceil \frac{2(\ell-1)}{3} \rceil$ paths.
Intuition: first contract the tree to a star, then de-contract it level by level; at each step, re-route paths accordingly.

Degree 2 vertices can be identified using at most $\lceil \frac{t}{2} \rceil$ additional paths.

Theorem (F., Kovše)

$p^{\text{ID}}(T) \leq \lceil \frac{2\ell}{3} \rceil + \lceil \frac{t}{2} \rceil$

Corollary

Let G be a **connected** graph. Then $p^{\text{ID}}(G) \leq \lceil \frac{2n}{3} \rceil$.

Proof: Take spanning tree T of G. An identifying path cover of T is also one for G! In the worst case, T has many leaves (up to $n-1$).

In general

In general, for connected graphs G, we have the tight bounds:

$$\lceil \log_2(n+1) \rceil \leq p^{\text{ID}}(G) \leq \lceil \frac{2n}{3} \rceil$$

In general

In general, for connected graphs G, we have the tight bounds:

$$\lceil \log_2(n+1) \rceil \leq p^{\text{ID}}(G) \leq \lceil \frac{2n}{3} \rceil$$

This an improvement over IDP, where for a connected instance:

$$\lceil \log_2(n+1) \rceil \leq p^{\text{ID}}(G) \leq n$$

(see e.g. Foucaud, Naserasr, Parreau 2012 for the special case of identifying codes in digraphs).

Computational complexity

Definition - ID. PATH COVER

INPUT: a graph G
PROBLEM: find the minimum-size identifying path cover of G.

Computational complexity

Definition - ID. PATH COVER

INPUT: a graph G
PROBLEM: find the minimum-size identifying path cover of G.

Definition - ID. PATH COVER-k

INPUT: a graph G
PROBLEM: find the minimum-size identifying path cover of G such that each path has at most k vertices.

Computational complexity

Definition - ID. PATH COVER

INPUT: a graph G
PROBLEM: find the minimum-size identifying path cover of G.

Definition - ID. PATH COVER-k

INPUT: a graph G
PROBLEM: find the minimum-size identifying path cover of G such that each path has at most k vertices.

We know that ID. PATH COVER is $O(\log(n))$-approximable and that ID. PATH COVER-k is $O(\log(k))$-approximable.

Computational complexity

Definition - ID. PATH COVER

INPUT: a graph G
PROBLEM: find the minimum-size identifying path cover of G.

Definition - ID. PATH COVER-k

INPUT: a graph G
PROBLEM: find the minimum-size identifying path cover of G such that each path has at most k vertices.

We know that ID. PATH COVER is $O(\log(n))$-approximable and that ID. PATH COVER-k is $O(\log(k))$-approximable.

Theorem (F., Kovše)

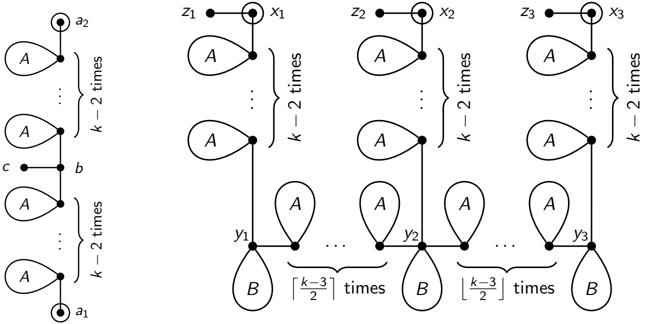
ID. PATH COVER-k is APX-hard, i.e., there exists some constant $c > 1$ such that it is NP-hard to approximate ID. PATH COVER-k within ratio c.

Computational complexity

Theorem (F., Kovše)

ID. PATH COVER-k is APX-hard, i.e., there exists some constant $c > 1$ such that it is NP-hard to approximate ID. PATH COVER-k within ratio c.

Proof: L-reduction from VERTEX COVER in cubic graphs. A and B are local sub-gadgets.



Open problems

- Give good **upper** bounds for TCP-k, IDP-k, ID. PATH COVER-k (this already seems to be a difficult question for identifying codes).

- Is there a polynomial-time algorithm for ID. PATH COVER in trees?

- What is the complexity of the general ID. PATH COVER problem?

- What are the graphs that admit a k-path identifying cover (i.e. all paths have **exactly** k vertices)?

Open problems

- Give good **upper** bounds for TCP-k, IDP-k, ID. PATH COVER-k (this already seems to be a difficult question for identifying codes).
- Is there a polynomial-time algorithm for ID. PATH COVER in trees?
- What is the complexity of the general ID. PATH COVER problem?
- What are the graphs that admit a k-path identifying cover (i.e. all paths have **exactly** k vertices)?

Thank you / Nandri / Shukriya / Merci!