

# EFFICIENT GRAPH STRUCTURES FOR VERTEX IDENTIFICATION

**Julien Moncel**

julien.moncel@iut-rodez.fr



maths à modeler



Bordeaux Workshop on Identifying Codes  
21–25 November 2011

# Outline

- 1 Introduction
- 2 Efficient structures
- 3 Superimposed codes
- 4 Constructions of efficient graphs using superimposed codes
- 5 Conclusion
- 6 Bibliography

# And now...

- 1 Introduction
- 2 Efficient structures
- 3 Superimposed codes
- 4 Constructions of efficient graphs using superimposed codes
- 5 Conclusion
- 6 Bibliography

# Identifying codes

## Definition (identifying code)

In a graph  $G$ , an identifying code  $C \subseteq V$  is a set of vertices that :

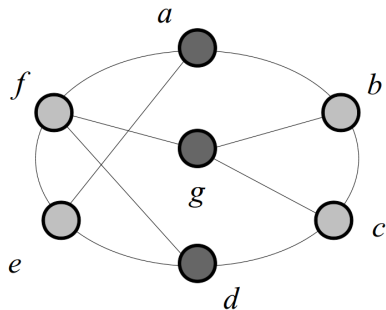
- covers all vertices of  $G$
- *separates* all pairs of distinct vertices of  $G$  :  
 $N[u] \cap C \neq N[v] \cap C$  for all  $u \neq v$

# Identifying codes

## Definition (identifying code)

In a graph  $G$ , an identifying code  $C \subseteq V$  is a set of vertices that :

- covers all vertices of  $G$
- separates all pairs of distinct vertices of  $G$  :  
 $N[u] \cap C \neq N[v] \cap C$  for all  $u \neq v$



- $C = \{a, d, g\}$
- $N[a] \cap C = \{a\}$
- $N[b] \cap C = \{a, g\}$
- $N[c] \cap C = \{d, g\}$
- etc.

## Two variants

Definition (identification at distance  $r > 1$ )

$C$  is an  $r$ -identifying code of  $G$  iff it is an identifying code of  $G^r$   
( $r$ -transitive closure)

## Two variants

### Definition (identification at distance $r > 1$ )

$C$  is an  $r$ -identifying code of  $G$  iff it is an identifying code of  $G^r$  ( $r$ -transitive closure)

### Definition (identification of sets of at most $t$ vertices)

$C$  is an  $(r, t)$ -identifying code of  $G$  iff it is a code of  $G^r$  that can identify sets of at most  $t$  vertices, in the sense that the sets

$$\bigcup_{x \in X} N[x] \cap C$$

(for  $|X| \leq t$ ) are all nonempty and distinct

# Notations

About these codes :

- identifying code  $\equiv$  ID code
- $r$ -identifying code  $\equiv$   $r$ -ID code
- $(r, t)$ -identifying code  $\equiv$   $(r, t)$ -ID code
- $(1, t)$ -identifying code  $\equiv$   $t$ -set-ID code



# Notations

About these codes :

- identifying code  $\equiv$  ID code
- $r$ -identifying code  $\equiv$   $r$ -ID code
- $(r, t)$ -identifying code  $\equiv$   $(r, t)$ -ID code
- $(1, t)$ -identifying code  $\equiv$   $t$ -set-ID code

Minimum cardinality of an ...

- ...identifying code in  $G$ :  $M(G)$
- ... $r$ -identifying code in  $G$ :  $M_r(G)$
- ... $(r, t)$ -identifying code in  $G$ :  $M_r^t(G)$
- ... $(1, t)$ -identifying code in  $G$ :  $M^t(G)$

## A brief survey of the literature

Identifying codes were introduced in [Karpovsky et al (1998)], to model a fault-detection problem in multiprocessor systems.

Since then, more than 200 papers on the topic, plus variants :

- identification of sets of vertices, and at distance  $r > 1$
- identification using cycles, colors ; or using patterns in  $\mathbb{Z}^2$
- identification of edges
- locating-dominating codes
- discriminating codes
- watching systems
- etc.

## A brief survey of the literature

Identifying codes were introduced in [Karpovsky et al (1998)], to model a fault-detection problem in multiprocessor systems.

Since then, more than 200 papers on the topic, plus variants :

- identification of sets of vertices, and at distance  $r > 1$
- identification using cycles, colors ; or using patterns in  $\mathbb{Z}^2$
- identification of edges
- locating-dominating codes
- discriminating codes
- watching systems
- etc.

Online bibliography on *The fabulous pages of Antoine Lobstein* :  
[www.infres.enst.fr/~lobstein/debutBIBidetlocdom.pdf](http://www.infres.enst.fr/~lobstein/debutBIBidetlocdom.pdf)

# Ph.D. thesis on identifying codes

Up to now, 7 Ph.D. thesis defended on the topic.

- J. M. *Identifying codes in graphs*, (dir. S. Gravier), Grenoble – France (in French, 2005)
- Robert Skaggs *Identifying vertices in graphs and digraphs*, (dir. M. Frick and G. Fricke), University of South Africa (2007)
- Sanna Ranto *Identifying and locating-dominating codes in binary Hamming spaces*, (dir. I. Honkala), Turku – Finland (2007)
- Moshe Laifenfeld *Coding for network applications: Robust identification and distributed resource allocation*, (dir. A. Trachtenberg), Boston – USA (2008)

## Ph.D. thesis on identifying codes (cont.)

- David Auger *Combinatorial identification problems and graph powers*, (dir. O. Hudry), Paris – France (in French, 2010)
- Brendon Stanton *On vertex identifying codes for infinite lattices*, (dir. R. Martin), Iowa – USA (2011)
- Ville Junnila *On identifying and locating-dominating codes*, (dir. T. Laihonon), Turku – Finland (2011)

## Ph.D. thesis on identifying codes (cont.)

- David Auger *Combinatorial identification problems and graph powers*, (dir. O. Hudry), Paris – France (in French, 2010)
- Brendon Stanton *On vertex identifying codes for infinite lattices*, (dir. R. Martin), Iowa – USA (2011)
- Ville Junnila *On identifying and locating-dominating codes*, (dir. T. Laihonon), Turku – Finland (2011)

... and still some thesis in progress :

- Marwane Bouznif, Grenoble – France
- Florent Foucaud, Bordeaux – France
- Aline Parreau, Grenoble – France
- Mikko Pelto, Turku – Finland
- (others ?)

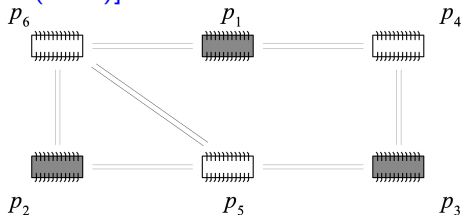
# Applications of these codes

- ▶ **Faut-detection in multiprocessor systems**  
[Karpovsky et al (1998)]

# Applications of these codes

## ► Fault-detection in multiprocessor systems

[Karpovsky et al (1998)]





# Applications of these codes

- ▶ **Faut-detection in multiprocessor systems**

[Karpovsky et al (1998)]

- ▶ **Monitoring of buildings using wireless sensors**

[Ray et al (2004), Ungrangsi et al (2004)]

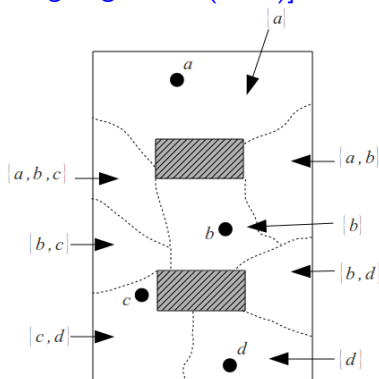
# Applications of these codes

## ► Fault-detection in multiprocessor systems

[Karpovsky et al (1998)]

## ► Monitoring of buildings using wireless sensors

[Ray et al (2004), Ungrangsi et al (2004)]



# Applications of these codes

► **Faut-detection in multiprocessor systems**

[Karpovsky et al (1998)]

► **Monitoring of buildings using wireless sensors**

[Ray et al (2004), Ungrangsi et al (2004)]

- See also David Auger's thesis ("fire detectors" application)

# Applications of these codes

- ▶ **Fault-detection in multiprocessor systems**

[Karpovsky et al (1998)]

- ▶ **Monitoring of buildings using wireless sensors**

[Ray et al (2004), Ungrangsi et al (2004)]

- ▶ **RNA secondary structure analysis** [Haynes et al (2006)]

# Applications of these codes

## ► Faut-detection in multiprocessor systems

[Karpovsky et al (1998)]

## ► Monitoring of buildings using wireless sensors

[Ray et al (2004), Ungrangsi et al (2004)]

## ► RNA secondary structure analysis [Haynes et al (2006)]

- Bioinformatics community: RAG model (“RNA As Graphs”) [Schlick et al (2004)]
- Non-surjective mapping

$$\Phi : \{\text{existing RNA structures}\} \rightarrow \{\text{trees}\}$$

- What are the structural properties of trees belonging to  $\text{Im}(\Phi)$  ?
- Classification heuristics using domination and variants (including ID codes) [Haynes et al (2006)]

# Applications of these codes

- ▶ **Faut-detection in multiprocessor systems**

[Karpovsky et al (1998)]

- ▶ **Monitoring of buildings using wireless sensors**

[Ray et al (2004), Ungrangsi et al (2004)]

- ▶ **RNA secondary structure analysis [Haynes et al (2006)]**

- ▶ **Process monitoring in industrial engineering**

[Galasso & Moncel]

# Applications of these codes

## ▶ **Faut-detection in multiprocessor systems**

[Karpovsky et al (1998)]

## ▶ **Monitoring of buildings using wireless sensors**

[Ray et al (2004), Ungrangsi et al (2004)]

## ▶ **RNA secondary structure analysis [Haynes et al (2006)]**

## ▶ **Process monitoring in industrial engineering**

[Galasso & Moncel]

- Framework: ISO 9000 standards (certification)
- Business processes seen as a graph (causality links)
- Efficient monitoring of the network of process (aggregation of indicators)
- *Work in progress...*

# Applications of these codes

- ▶ **Faut-detection in multiprocessor systems**

[Karpovsky et al (1998)]

- ▶ **Monitoring of buildings using wireless sensors**

[Ray et al (2004), Ungrangsi et al (2004)]

- ▶ **RNA secondary structure analysis [Haynes et al (2006)]**

- ▶ **Process monitoring in industrial engineering**

[Galasso & Moncel]

- ▶ ...



# And now...

- 1 Introduction
- 2 Efficient structures**
- 3 Superimposed codes
- 4 Constructions of efficient graphs using superimposed codes
- 5 Conclusion
- 6 Bibliography

## General results

Theorem ([Charon et al (2003)])

*Given a graph  $G$  admitting a code, computing  $M(G)$  is **NP-hard**.  
This remains true even if  $G$  is assumed to be bipartite.*

## General results

Theorem ([Charon et al (2003)])

*Given a graph  $G$  admitting a code, computing  $M(G)$  is **NP-hard**.  
This remains true even if  $G$  is assumed to be bipartite.*

Theorem ([Charon et al (2005), Charon et al (2007)])

*For every  $r \geq 1$ , and for every connected graph  $G$  on  $n \geq 3$  vertices admitting an  $r$ -ID code, we have*

$$\lceil \log_2(n+1) \rceil \leq M_r(G) \leq n-1.$$

*In addition, for every  $c \in [\lceil \log_2(n+1) \rceil, n-1]$ , there exists a connected graph on  $n$  vertices such  $G_c$  that*

$$M_r(G_c) = c.$$

# Ideas of proof

- ▶ Logarithmic bound [[Karpovsky et al \(1998\)](#)]

# Ideas of proof

- ▶ Logarithmic bound [Karpovsky et al (1998)]
  - $n + 1$  events:
    - one of the  $n$  vertices is faulty
    - or no faulty vertex at all
  - binary tests to find which event occurs
  - hence at least  $\log_2(n + 1)$  tests are needed

# Ideas of proof

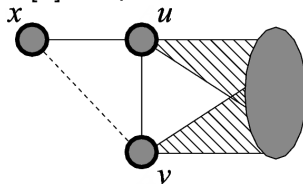
- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]

# Ideas of proof

- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]
  - $V \setminus \{x\}$  is an ID code
    - iff* each vertex of  $V \setminus N[x]$  is separated from each vertex of  $N[x]$

# Ideas of proof

- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]
  - $V \setminus \{x\}$  is an ID code  
*iff* each vertex of  $V \setminus N[x]$  is separated from each vertex of  $N[x]$



- taking  $x$  of **maximum degree**, we have that either  $V \setminus \{x\}$  or  $V \setminus \{v\}$  is an ID code
- (also true for an infinite graph of bounded degree)



# Ideas of proof

- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]
- ▶ Attaining these bounds [Charon et al (2007)]

# Ideas of proof

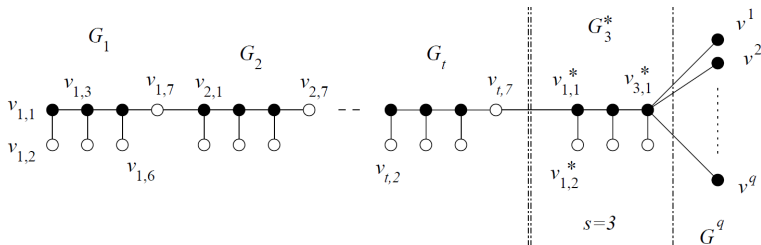
- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]
- ▶ Attaining these bounds [Charon et al (2007)]
  - for  $r = 1$ : “easy” (already treated in [Karpovsky et al (1998)])
  - for any  $r > 1$ : requires more work

# Ideas of proof

- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]
- ▶ Attaining these bounds [Charon et al (2007)]
- ▶ In-between values [Charon et al (2005)]

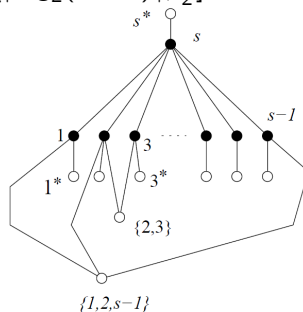
# Ideas of proof

- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]
- ▶ Attaining these bounds [Charon et al (2007)]
- ▶ In-between values [Charon et al (2005)]
  - use of trees for  $\lceil \frac{3(n+1)}{7} \rceil, n - 1$



# Ideas of proof

- ▶ Logarithmic bound [Karpovsky et al (1998)]
- ▶  $n - 1$  bound [Bertrand (2001), Gravier & Moncel (2007)]
- ▶ Attaining these bounds [Charon et al (2007)]
- ▶ In-between values [Charon et al (2005)]
  - use of trees for  $\lceil \frac{3(n+1)}{7} \rceil, n - 1$
  - bipartite graphs for  $\lceil \log_2(n + 1) \rceil, \frac{n}{2}$

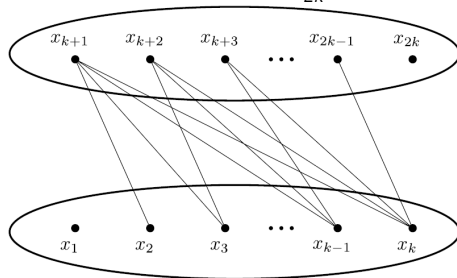


## Related question

- ▶ Finding **all** the extremal graphs
  - $\lceil \log_2(n+1) \rceil$  bound (plus minimization of number of edges):  
[Moncel (2006), Raspaud & Tong]
  - $n - 1$  bound: [Foucaud et al (2011)]

## Related question

- ▶ Finding **all** the extremal graphs
  - $\lceil \log_2(n+1) \rceil$  bound (plus minimization of number of edges):  
[Moncel (2006), Raspaud & Tong]
  - $n - 1$  bound: [Foucaud et al (2011)]
    - $(\{\text{stars}\} \cup \mathcal{A} \cup \mathcal{A} \bowtie K_1) \setminus \overline{K_2}$
    - with  $\bowtie =$  join operation, that is  
 $G_1 \bowtie G_2 = G_1 \uplus G_2 +$  all possible edges between  $G_1$  and  $G_2$
    - and  $\mathcal{A} =$   $\bowtie$ -closure of graphs of the form  $P_{2k}^{k-1}$



## Related question

- ▶ Finding **all** the extremal graphs
  - $\lceil \log_2(n+1) \rceil$  bound (plus minimization of number of edges):  
[Moncel (2006), Raspaud & Tong]
  - $n - 1$  bound: [Foucaud et al (2011)]
- ▶ Infinite graphs
  - classification of infinite graphs admitting only  $V$  as an ID code: [Foucaud et al (2011)]



## Related question

- ▶ Finding **all** the extremal graphs
  - $\lceil \log_2(n+1) \rceil$  bound (plus minimization of number of edges):  
[Moncel (2006), Raspaud & Tong]
  - $n - 1$  bound: [Foucaud et al (2011)]
- ▶ Infinite graphs
  - classification of infinite graphs admitting only  $V$  as an ID code: [Foucaud et al (2011)]
- ▶ Identification at distance  $r$ : ( $r$ -th) **root** of a graph
  - implicit in the above-mentioned papers
  - David Auger's thesis
  - papers [Auger et al, Auger et al (2011)]

## Quite a big gap

The gap between  $\lceil \log_2(n+1) \rceil$  and  $n-1$  is **exponential**.

### Question

What if we had the **choice** of a network to perform vertex identification ?

## Quite a big gap

The gap between  $\lceil \log_2(n+1) \rceil$  and  $n-1$  is **exponential**.

### Question

What if we had the **choice** of a network to perform vertex identification ?

### Answer

We would surely choose a graph  $G$  on  $n$  vertices for which  $M(G) = O(\log n)$ .

## Quite a big gap

The gap between  $\lceil \log_2(n+1) \rceil$  and  $n-1$  is **exponential**.

### Question

What if we had the **choice** of a network to perform vertex identification ?

### Answer

We would surely choose a graph  $G$  on  $n$  vertices for which  $M(G) = O(\log n)$ .

### Definition (Efficient structure)

An efficient structure to perform vertex identification is a graph  $G$  such that  $M(G)$  is close to  $O(\log n)$ , where  $n$  is the number of vertices of  $G$ .

# Survey of known results

- ▶ ID codes
  - we know (more than) everything thanks to [Moncel (2006), Raspaud & Tong]
- ▶  $r$ -ID codes
  - we know many efficient graphs thanks to [Charon et al (2007)]

# Survey of known results

- ▶ ID codes
  - we know (more than) everything thanks to [Moncel (2006), Raspaud & Tong]
- ▶  $r$ -ID codes
  - we know many efficient graphs thanks to [Charon et al (2007)]
- ▶  $t$ -set-ID codes
  - *we do not know so many things...*

# Identification of sets of vertices

Harder than identification of vertices!!

## MIN $(r, t)$ -ID CODE

- Input: a graph  $G$ , an integer  $k$
- Output: does  $G$  admit an  $(r, t)$ -ID code of cardinality  $\leq k$  ?

*Complexity unknown!*

# Identification of sets of vertices

Harder than identification of vertices!!

## MIN $(r, t)$ -ID CODE

- Input: a graph  $G$ , an integer  $k$
- Output: does  $G$  admit an  $(r, t)$ -ID code of cardinality  $\leq k$  ?

*Complexity unknown!*

Theorem ([Auger et al (2010)])

MIN  $(r, 2)$ -ID CODE is NP-complete, even when restricted to planar graphs of maximum degree  $\leq 3$



# Bounds

We still get a log bound :

Proposition ([Karpovsky et al (1998)])

*Let  $C$  be a  $t$ -set-ID code in a graph on  $n$  vertices. Then*

$$2^{|C|} \geq \sum_{i=0}^t \binom{n}{i}$$

# Bounds

We still get a log bound :

Proposition ([Karpovsky et al (1998)])

Let  $C$  be a  $t$ -set-ID code in a graph on  $n$  vertices. Then

$$2^{|C|} \geq \sum_{i=0}^t \binom{n}{i}$$

This implies that  $|C| \geq \Omega(t \log n)$ , but this can be improved...

# And now...

- 1 Introduction
- 2 Efficient structures
- 3 Superimposed codes**
- 4 Constructions of efficient graphs using superimposed codes
- 5 Conclusion
- 6 Bibliography

# Definition

Definition (Superimposed code [Kautz & Singleton (1964)])

Let  $t \geq 1$  and  $k \geq 1$ . A  $t$ -**superimposed code** of dimension  $k$  is a family  $\mathcal{F} \subseteq 2^{[k]}$  such that

$$\bigcup_{A \in \mathcal{A}} A \neq \bigcup_{B \in \mathcal{B}} B$$

for any  $\mathcal{A}, \mathcal{B} \subseteq \mathcal{F}$ ,  $\mathcal{A} \neq \mathcal{B}$ ,  $|\mathcal{A}| \leq t$ ,  $|\mathcal{B}| \leq t$ .

(also called *union-free* family)

# Definition

Definition (Superimposed code [Kautz & Singleton (1964)])

Let  $t \geq 1$  and  $k \geq 1$ . A  $t$ -**superimposed code** of dimension  $k$  is a family  $\mathcal{F} \subseteq 2^{[k]}$  such that

$$\bigcup_{A \in \mathcal{A}} A \neq \bigcup_{B \in \mathcal{B}} B$$

for any  $\mathcal{A}, \mathcal{B} \subseteq \mathcal{F}$ ,  $\mathcal{A} \neq \mathcal{B}$ ,  $|\mathcal{A}| \leq t$ ,  $|\mathcal{B}| \leq t$ .

(also called *union-free* family)

In other words, the union  $\bigcup_{A \in \mathcal{A}} A$  **identifies** the set  $\mathcal{A}$

# Definition

Definition (Superimposed code [Kautz & Singleton (1964)])

Let  $t \geq 1$  and  $k \geq 1$ . A  $t$ -**superimposed code** of dimension  $k$  is a family  $\mathcal{F} \subseteq 2^{[k]}$  such that

$$\bigcup_{A \in \mathcal{A}} A \neq \bigcup_{B \in \mathcal{B}} B$$

for any  $\mathcal{A}, \mathcal{B} \subseteq \mathcal{F}$ ,  $\mathcal{A} \neq \mathcal{B}$ ,  $|\mathcal{A}| \leq t$ ,  $|\mathcal{B}| \leq t$ .

(also called *union-free* family)

In other words, the union  $\bigcup_{A \in \mathcal{A}} A$  **identifies** the set  $\mathcal{A}$

Given  $t$  and  $k$ , the aim is to find  $\mathcal{F}$  of **maximum cardinality**

# Applications of superimposed codes

- ▶ CDMA : code division multiple access channel
  - at most  $t$  users on a given communication channel
  - each of them is identified by a code  $A \in 2^{[k]}$
  - it is enough to know  $\cup A$  to know exactly who is using the channel

# Applications of superimposed codes

- ▶ CDMA : code division multiple access channel
  - at most  $t$  users on a given communication channel
  - each of them is identified by a code  $A \in 2^{[k]}$
  - it is enough to know  $\cup A$  to know exactly who is using the channel
- ▶ Group testing [Du & Hwang (1993)]
  - at most  $t$  positive (e.g. blood) samples among a large set of samples
  - possibility to *pool* the tests
  - a pooled test is positive *iff* there is at least one positive sample in the pool
  - group testing is used for instance in genomics



## Connection with ID codes

Quote from [Karpovsky et al (1998)]

*“Near-optimal constructions [of the graph] for sets of vertices can be obtained using superimposed codes.”*

Intuitively: YES, but **formally??**

# Connection with ID codes

Quote from [Karpovsky et al (1998)]

*“Near-optimal constructions [of the graph] for sets of vertices can be obtained using superimposed codes.”*

Intuitively: YES, but **formally??**

What is clear :

**Proposition**

*Let  $C$  be a  $t$ -set-ID code of a graph  $G$  on  $n$  vertices. Then the family  $\{N[v] \cap C \mid v \in V(G)\}$  is a  $t$ -superimposed code of dimension  $|C|$ .*

$\Rightarrow$  *superimposed codes can be obtained using identifying codes*

# Bounds on $t$ -set-ID codes

## Theorem

([D'yachkov & Rykov (1983), Füredi (1996), Ruszinkó (1994)])

*There exists two constants  $c_1 > 0$  and  $c_2 > 0$  such that for any optimal  $t$ -superimposed code  $\mathcal{F}^*$  of dimension  $k$  we have*

$$c_1 \frac{k}{t^2} \leq \log_2 |\mathcal{F}^*| \leq c_2 \frac{k \log t}{t^2}$$

## Corollary

There exists a constant  $c_2$  such that any  $t$ -set-ID code  $C$  in a graph on  $n$  vertices satisfies

$$|C| \geq c_2 \frac{t^2}{\log t} \log n$$

# Bounds on $t$ -set-ID codes

## Theorem

([D'yachkov & Rykov (1983), Füredi (1996), Ruszinkó (1994)])

*There exists two constants  $c_1 > 0$  and  $c_2 > 0$  such that for any optimal  $t$ -superimposed code  $\mathcal{F}^*$  of dimension  $k$  we have*

$$c_1 \frac{k}{t^2} \leq \log_2 |\mathcal{F}^*| \leq c_2 \frac{k \log t}{t^2}$$

## Corollary

There exists a constant  $c_2$  such that any  $t$ -set-ID code  $C$  in a graph on  $n$  vertices satisfies

$$|C| \geq c_2 \frac{t^2}{\log t} \log n$$

Improves the  $O(t \log n)$  bound of [Karpovsky et al (1998)]

# Reverse connection

How to get  $t$ -set-ID codes from  $t$ -superimposed codes ?

Harder !!

Main problem :

- no **structure** in superimposed codes (sets)
- heavy structure in ID codes (graph)

## Example – a 2-superimposed code of dimension 6

$v_1$	0	1	0	1	0	0
$v_2$	1	0	1	0	0	0
$v_3$	0	0	0	1	0	1
$v_4$	0	1	0	0	1	0
$v_5$	1	0	0	0	0	1
$v_6$	1	0	0	0	1	0

## Example – a 2-superimposed code of dimension 6

$v_1$	0	1	0	1	0	0	$v_1 \vee v_2$	1	1	1	1	0	0
$v_2$	1	0	1	0	0	0	$v_1 \vee v_3$	0	1	1	1	0	1
$v_3$	0	0	0	1	0	1	$v_1 \vee v_4$	0	1	0	1	1	0
$v_4$	0	1	0	0	1	0	$v_1 \vee v_5$	1	1	0	1	0	1
$v_5$	1	0	0	0	0	1	$v_1 \vee v_6$	1	1	0	1	1	0
$v_6$	1	0	0	0	1	0	$v_2 \vee v_3$	1	0	1	1	0	1
							$v_2 \vee v_4$	1	1	1	0	1	0
							$v_2 \vee v_5$	1	0	1	0	0	1
							$v_2 \vee v_6$	1	0	1	0	1	0
							$v_3 \vee v_4$	0	1	0	1	1	1
							$v_3 \vee v_5$	1	0	0	1	0	1
							$v_3 \vee v_6$	1	0	0	1	1	1
							$v_4 \vee v_5$	1	1	0	0	1	1
							$v_4 \vee v_6$	1	1	0	0	1	0
							$v_5 \vee v_6$	1	0	0	0	1	1

## Example – a 2-superimposed code of dimension 6

$v_1$	0	1	0	1	0	0
$v_2$	1	0	1	0	0	0
$v_3$	0	0	0	1	0	1
$v_4$	0	1	0	0	1	0
$v_5$	1	0	0	0	0	1
$v_6$	1	0	0	0	1	0

All distinct : OK

$v_1 \vee v_2$	1	1	1	1	0	0
$v_1 \vee v_3$	0	1	1	1	0	1
$v_1 \vee v_4$	0	1	0	1	1	0
$v_1 \vee v_5$	1	1	0	1	0	1
$v_1 \vee v_6$	1	1	0	1	1	0
$v_2 \vee v_3$	1	0	1	1	0	1
$v_2 \vee v_4$	1	1	1	0	1	0
$v_2 \vee v_5$	1	0	1	0	0	1
$v_2 \vee v_6$	1	0	1	0	1	0
$v_3 \vee v_4$	0	1	0	1	1	1
$v_3 \vee v_5$	1	0	0	1	0	1
$v_3 \vee v_6$	1	0	0	1	1	1
$v_4 \vee v_5$	1	1	0	0	1	1
$v_4 \vee v_6$	1	1	0	0	1	0
$v_5 \vee v_6$	1	0	0	0	1	1



## Example – a 2-superimposed code of dimension 6

$v_1$	0	1	0	1	0	0	$c_1$
$v_2$	1	0	1	0	0	0	$c_2$
$v_3$	0	0	0	1	0	1	$c_3$
$v_4$	0	1	0	0	1	0	$c_4$
$v_5$	1	0	0	0	0	1	$c_5$
$v_6$	1	0	0	0	1	0	$c_6$

$c_1$     $c_2$     $c_3$     $c_4$     $c_5$     $c_6$

# Example – a 2-superimposed code of dimension 6

$v_1$	0	1	0	1	0	0	$c_1$	1							
$v_2$	1	0	1	0	0	0	$c_2$		1						
$v_3$	0	0	0	1	0	1	$c_3$			1					
$v_4$	0	1	0	0	1	0	$c_4$				1				
$v_5$	1	0	0	0	0	1	$c_5$					1			
$v_6$	1	0	0	0	1	0	$c_6$							1	
								$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$		

# Example – a 2-superimposed code of dimension 6

$v_1$	0	1	0	1	0	0	$c_1$	1	a	b					
$v_2$	1	0	1	0	0	0	$c_2$	a	1		...				
$v_3$	0	0	0	1	0	1	$c_3$	b		1					
$v_4$	0	1	0	0	1	0	$c_4$		...		1				
$v_5$	1	0	0	0	0	1	$c_5$					1			
$v_6$	1	0	0	0	1	0	$c_6$							1	
								$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$		

## Example – a 2-superimposed code of dimension 6

$v_1$	0	1	0	1	0	0	$c_1$	1	a	b					
$v_2$	1	0	1	0	0	0	$c_2$	a	1		...				
$v_3$	0	0	0	1	0	1	$c_3$	b		1					
$v_4$	0	1	0	0	1	0	$c_4$		...		1				
$v_5$	1	0	0	0	0	1	$c_5$					1			
$v_6$	1	0	0	0	1	0	$c_6$							1	
								$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$		

Looking for a **specific structure** into the matrix so that it becomes the  $C - V$ -incidence matrix of a graph equipped with an ID code  $C$ !

- permutations of rows and columns to get this structure
- not always possible to do it

# And now...

- 1 Introduction
- 2 Efficient structures
- 3 Superimposed codes
- 4 Constructions of efficient graphs using superimposed codes**
- 5 Conclusion
- 6 Bibliography

## Oriented case

Theorem ([Frieze et al (2007)])

*Let  $\mathcal{F}$  be a maximal superimposed code of dimension  $n$ . Then it is possible to find a set of permutations of rows and columns of the 0 – 1-matrix associated to  $\mathcal{F}$  such that it can be seen as a  $C - V$ -incidence matrix of an **oriented** graph equipped with an ID code  $C$  such that  $|C| = |\mathcal{F}|$ .*

In other words, we can get the 1-diagonal, but not the symmetry.

We do not have any result of this type for the **non-oriented case**.

## A Lemma using projective planes

Still, we get *something*:

Lemma ([Gravier & Moncel (2005)])

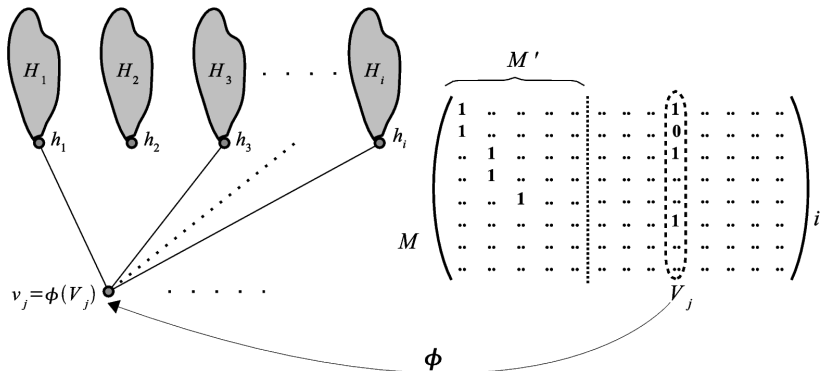
*Let  $q$  be a prime power, and let  $A$  be the incidence matrix of a projective plane of dimension  $q$ . Then the graph having*

$$B = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$$

*as adjacency matrix is a connected graph on  $2(q^2 + q + 1)$  vertices, admitting a  $q$ -set-ID code.*

# Use of previous Lemma (from [Gravier & Moncel (2005)])

$H_i =$  copy of the graph of the Lemma



Since the  $H_i$ 's are "small", then we do not lose too much:

$$|C| = O(t^4 \log n)$$



# Random construction

Theorem ([Frieze et al (2007)])

Let  $t \geq 2$ . There exists an infinite family of graphs  $(G_i)_{i \in \mathbb{N}}$  such that

$$M^t(G_i) = O(t^2 \log n_i) \quad \forall i \in \mathbb{N}$$

where  $n_i$  is the number of vertices of  $G_i$  for all  $i$ .

*Non-constructive* result, better than [Gravier & Moncel (2005)], based on probabilistic arguments:

- drop edges with a suitable probability  $p$
- estimate the probability that a given subset of vertices  $C$  is a code
- tune  $p$  so that this probability is positive for a  $C$  of small size

## Another construction (from [Moncel (2006)])

Greedy algorithm (sort of “derandomization” of the random construction of [Frieze et al (2007)], inspired from the greedy algorithm of [Hwang & Sós (1987)]):

- 1 set  $G_N$  a graph on  $N$  vertices  $v_1, \dots, v_N$  obtained with the projective plane Lemma
- 2 set  $I(i) = \{j \mid v_j \in N(v_i)\}$  for all  $i$
- 3 set  $\mathcal{F}$  as  $\{A \mid A \subseteq 2^{[M]}, |A| = \lceil \frac{N}{3t} \rceil\}$
- 4 set  $\mathcal{F} = \mathcal{F} \setminus \bigcup_i \{A \mid |A \cap I(i)| > \frac{1}{t} |I(i)|\}$
- 5 while  $\mathcal{F} \neq \emptyset$  do
  - pick  $A_0 \in \mathcal{F}$
  - add a new vertex  $v_{A_0}$  to  $G_N$ , such that  $N(v_{A_0}) = \{v_i \mid i \in A_0\}$
  - set  $\mathcal{F} = \mathcal{F} \setminus \{B \mid |B \cap A_0| \geq \frac{1}{t} |A_0|\}$

# Main arguments

Why this algorithm works (general ideas):

- elements of  $\mathcal{F}$  are of size  $\frac{N}{3t}$ , hence  $\mathcal{F}$  is large
- elements of  $\mathcal{F}$  do not intersect the  $I(i)$ 's too much, which helps for separation property
- elements added to  $G_N$  do not intersect themselves too much, which helps for separation property
- parameters are tuned such that we do not throw out too many elements of  $\mathcal{F}$

# Main arguments

Why this algorithm works (general ideas):

- elements of  $\mathcal{F}$  are of size  $\frac{N}{3t}$ , hence  $\mathcal{F}$  is large
- elements of  $\mathcal{F}$  do not intersect the  $I(i)$ 's too much, which helps for separation property
- elements added to  $G_N$  do not intersect themselves too much, which helps for separation property
- parameters are tuned such that we do not throw out too many elements of  $\mathcal{F}$

And there are useful notions behind this...

# Cover-free families

Useful notion:

Definition (Cover-free family [Kautz & Singleton (1964)])

Let  $t \geq 1$  and  $k \geq 1$ . A  $t$ -cover-free family of dimension  $k$  is a family  $\mathcal{F} \subseteq 2^{[k]}$  such that

$$A \not\subseteq \bigcup_{B \in \mathcal{B}} B$$

for any  $A \in \mathcal{F}, \mathcal{B} \subseteq \mathcal{F}, A \notin \mathcal{B}, |\mathcal{B}| \leq t$ .

# Cover-free families

Useful notion:

**Definition (Cover-free family [Kautz & Singleton (1964)])**

Let  $t \geq 1$  and  $k \geq 1$ . A  $t$ -**cover-free** family of dimension  $k$  is a family  $\mathcal{F} \subseteq 2^{[k]}$  such that

$$A \not\subseteq \bigcup_{B \in \mathcal{B}} B$$

for any  $A \in \mathcal{F}, \mathcal{B} \subseteq \mathcal{F}, A \notin \mathcal{B}, |\mathcal{B}| \leq t$ .

**Proposition**

$\mathcal{F}$  is  $t$ -cover-free  $\Rightarrow \mathcal{F}$  is a  $t$ -superimposed code  $\Rightarrow \mathcal{F}$  is  $(t - 1)$ -cover-free

# Part-intersecting families

Another useful notion:

Definition (Part-intersecting family [Kautz & Singleton (1964)])

Let  $t \geq 1$  and  $k \geq 1$ . A  **$t$ -part-intersecting** family of dimension  $k$  is a family  $\mathcal{F} \subseteq 2^{[k]}$  such that

$$|A \cap B| < \frac{1}{t} \min(|A|, |B|)$$

for any  $A, B \in \mathcal{F}, A \neq B$ .

# Part-intersecting families

Another useful notion:

**Definition** (Part-intersecting family [Kautz & Singleton (1964)])

Let  $t \geq 1$  and  $k \geq 1$ . A  **$< t$ -part-intersecting** family of dimension  $k$  is a family  $\mathcal{F} \subseteq 2^{[k]}$  such that

$$|A \cap B| < \frac{1}{t} \min(|A|, |B|)$$

for any  $A, B \in \mathcal{F}, A \neq B$ .

**Proposition**

$\mathcal{F}$  is  $< t$ -part-intersecting  $\Rightarrow \mathcal{F}$  is  $t$ -cover-free  $\Rightarrow \mathcal{F}$  is a  $t$ -superimposed code



# And now...

- 1 Introduction
- 2 Efficient structures
- 3 Superimposed codes
- 4 Constructions of efficient graphs using superimposed codes
- 5 Conclusion**
- 6 Bibliography

# Summary

Overview of discussed results:

- Graphs for which  $M(G) = O(\log n)$  are of specific interest, in particular with respect to applications
- We know **all** of them for ID codes
- We know **some** of them for  $r$ -ID codes
- As for  $t$ -set-ID codes:
  - we do not even know the tight lower bound!
  - we have a lower bound of  $\Omega\left(\frac{t^2}{\log t} \log n\right)$
  - we have random and explicit constructions for  $O(t^2 \log n)$
  - we do not know what happens in between
  - everything happens as for the case of superimposed codes
  - although we do not have an **explicit** correspondence between  $t$ -set-ID codes and superimposed codes

# Perspectives

Related problems:

- **upper bound** for  $t$ -set-ID codes ?
- **characterization** of all extremal graphs (analog to [Foucaud et al (2011)]) ?
- better lower bounds (for ID codes) and upper bounds (for superimposed codes) ?
- establish a **formal link** between ID codes and superimposed codes ?

# EFFICIENT GRAPH STRUCTURES FOR VERTEX IDENTIFICATION

**Julien Moncel**

julien.moncel@iut-rodez.fr



maths à modeler



Bordeaux Workshop on Identifying Codes  
21–25 November 2011






# And now...

- 1 Introduction
- 2 Efficient structures
- 3 Superimposed codes
- 4 Constructions of efficient graphs using superimposed codes
- 5 Conclusion
- 6 Bibliography**






## Cited papers (1)

-  D. Auger, I. Charon, O. Hudry, A. Lobstein, *On the square roots of graphs*, submitted.
-  D. Auger, I. Charon, O. Hudry, A. Lobstein, *Complexity results for identifying codes in planar graphs*, International Transactions in Operational Research **17(6)** (2010), 691–710.
-  D. Auger, I. Charon, O. Hudry, A. Lobstein, *On the sizes of graphs and their powers: The undirected case*, Discrete Applied Mathematics **159(16)** (2011), 1666–1675.
-  N. Bertrand, *Identifying and locating-dominating codes on some classes of graphs*, Master's thesis (in French, 2001).
-  I. Charon, O. Hudry, A. Lobstein, *Minimizing the Size of an Identifying or Locating-Dominating Code in a Graph is NP-hard*, Theoretical Computer Science **290(3)** (2003), 2109–2120.

## Cited papers (2)





-  I. Charon, O. Hudry, A. Lobstein, *Possible cardinalities for identifying codes in graphs*, Australasian Journal of Combinatorics **32** (2005), 177–195.
-  I. Charon, O. Hudry, A. Lobstein, *Extremal cardinalities for identifying and locating-dominating codes in graphs*, Discrete Mathematics **307** (2007), 356–366.
-  A. G. D'yachkov, V. V. Rykov, *Bounds on the length of disjunctive codes*, Problems of Information Transmission **18** (1983), 166–171.
-  D.-Z. Du, F. K. Hwang, *Combinatorial group testing and its applications*, World scientific (1993).
-  F. Foucaud, E. Guerrini, M. Kovše, R. Naserasr, A. Parreau, P. Valicov, *Extremal graphs for the identifying code problem*, European Journal of Combinatorics **32(4)** (2011), 628–638.

## Cited papers (3)

-  A. Frieze, R. Martin, J. Moncel, M. Ruszinkó, C. Smyth, *Codes identifying sets of vertices in random networks*, Discrete Mathematics **307(9-10)** (2007), 1094–1107.
-  Z. Füredi, *On  $r$ -cover free families*, Journal of Combinatorial Theory Series A **73(1)** (1996), 172–173.
-  F. Galasso, J. Moncel, *Business process monitoring using identifying codes*, in preparation.
-  S. Gravier, J. Moncel, *Construction of codes identifying sets of vertices*, Electronic Journal of Combinatorics **12(1)** (2005), R13.
-  F. Hwang, V. Sós, *Non adaptive hypergeometric group testing*, Studia Scientiarum Mathematicarum Hungaricae, **22(1-4)** (1987), 257–263.



## Cited papers (4)

-  H. Gan, D. Fera, J. Zorn, N. Shiffeldrim, U. Laserson, N. Kim, T. Schlick, *RAG: RNA-As-Graphs database – concepts, analysis, and features*, *Bioinformatics* **20** (2004), 1285–1291.
-  S. Gravier, J. Moncel, *On graphs having a  $V \setminus \{x\}$  set as an identifying code*, *Discrete Mathematics* **307(3-5)** (2007), 432–434.
-  T. Haynes, D. Knisley, E. Seier, Y. Zou, *A quantitative analysis of secondary RNA structure using domination based parameters on trees*, *BMC Bioinformatics* **7** (2006), 108.
-  M. Karpovsky, K. Chakrabarty, L. Levitin, *On a New Class of Codes for Identifying Vertices in Graphs*, *IEEE Transactions on Information Theory* **44(2)** (1998), 599–611.

## Cited papers (5)



W. Kautz, R. Singleton, *Nonrandom binary superimposed codes*, IEEE Transactions on Information Theory **10(4)**, (1964), 363–377.



J. Moncel, *On graphs on  $n$  vertices having an identifying code of cardinality  $\lceil \log_2(n+1) \rceil$* , Discrete Applied Mathematics **154(14)** (2006), 2032–2039.



J. Moncel, *Constructing codes identifying sets of vertices*, Designs, Codes and Cryptography **41(1)** (2006), 23–31.



S. Ray, D. Starobinski, A. Trachtenberg, R. Ungrangsi, *Robust location detection with sensor networks*, IEEE Journal on Selected Areas in Communications **22(6)** (2004), 1016–1025.

## Cited papers (6)



L.-D. Tong, A. Raspaud, *The minimum identifying code graphs*, submitted.



M. Ruszinkó, *On the upper bound of the size of the  $r$ -cover-free families*, *Journal of Combinatorial Theory Series A* **66(2)** (1994), 302–310.



R. Ungrangsi, A. Trachtenberg, D. Starobinski, *An Implementation of Indoor Location Detection Systems Based on Identifying Codes*, *Lecture Notes in Computer Science* **3283** (2004), 175–189.