Adaptive identification in graphs

Identification adaptative dans les graphes

Yael Ben-Haim
Sylvain Gravier
Antoine Lobstein
Julien Moncel

2007D012

Septembre 2007

Département Informatique et Réseaux
Groupe MIC2 : Mathématiques de l'Information,
de la Communication et du Calcul

Identification Adaptative Dans Les Graphes

Adaptive Identification in Graphs

Yael Ben-Haim
Department of Electrical Engineering-Systems
Tel Aviv University
Tel Aviv 69978 - Israel
yael@eng.tau.ac.il

Sylvain Gravier
Institut Fourier
ERTé "maths à modeler"
100 rue des Maths
38402 Saint Martin d'Hères - France
sylvain.gravier@ujf-grenoble.fr

Antoine Lobstein
CNRS - LTCI UMR 5141
ENST - Département INFRES
46 rue Barrault
75634 Paris Cedex 13 - France
lobstein@enst.fr

Julien Moncel
INPG - Laboratoire Leibniz
ERTé "maths à modeler"
Groupe de recherche GéoD
46 avenue Félix Viallet
38031 Grenoble Cedex - France
julien.moncel@imag.fr

Résumé

Considérons un graphe connexe et non orienté $G = (V, E)$, un sous-ensemble de sommets $C \subseteq V$, et un entier $r \geq 1$; pour tout sommet $v \in V$, on dénote par $B_r(v)$ la boule de rayon r centrée sur v, i.e., l'ensemble de tous les sommets à distance au plus r de v. On dit que C est un code r-identifiant de G si et seulement si tous les ensembles $B_r(v) \cap C$ sont non vides et distincts deux à deux. Ces codes peuvent être utilisés pour concevoir des procédés de localisation-détection dans des réseaux. Ils ont été initialement définis pour modéliser un problème de détection de pannes dans des réseaux de multiprocesseurs. En effectuant un test sur tous les processeurs correspondant aux sommets de C, on peut vérifier s'il existe un processeur en panne dans le réseau, et si oui, le localiser. Dans cet article, nous présentons une version adaptative des codes identifiants, qui permet de tester les processeurs du réseau l'un après l'autre. Nous donnons un exemple simple où, dans le pire cas, l'identification adaptative réclame un nombre de tests qui est logarithmique en le nombre minimum de tests dans le cas non adaptatif. Le but de cet article, après avoir défini l'identification adaptative dans les graphes et donné des bornes générales sur les codes r-identifiants adaptatifs, est d'étudier ces codes sur des tores dans les grilles carrée et royale. Nous montrerons que l'identification adaptative peut être rapprochée d'un problème de recherche de type Rényi, étudié par M. Ruszinkó [M. Ruszinkó, *On a 2-dimensional search problem*, Journal of Statistical Planning and Inference **37(3)** (1993), 371-383].

Abstract

Consider a connected undirected graph $G = (V, E)$, a subset of vertices $C \subseteq V$, and an integer $r \geq 1$; for any vertex $v \in V$, let $B_r(v)$ denote the ball of radius r centered at v, i.e., the set of all vertices within distance r from v. We say that C is an r-identifying code of G if and only if all the sets $B_r(v) \cap C$ are nonempty and pairwise distinct. These codes are used for devising location-detection schemes in wireless sensor networks. They were originally introduced to model a fault-detection problem in multiprocessor networks. By running a test procedure on all the processors corresponding to vertices of C, one can check if there is a faulty processor in the network, and locate the faulty processor if there is one. In this paper we introduce an adaptive version of identifying codes, which enables one to run a test procedure on the processors of the network one after the other. We show a simple example where, in the worst case, adaptive identification requires a number of tests which is logarithmic in the minimum number of tests for the non-adaptive case. The purpose of this paper, after introducing adaptive identification in graphs and giving general bounds on adaptive r-identifying codes, is to study these codes in torii in the square and in the king lattices. We show that adaptive identification can be closely related to a Rényi-type search problem studied by M. Ruszinkó [M. Ruszinkó, *On a 2-dimensional search problem,* Journal of Statistical Planning and Inference **37(3)** (1993), 371–383].

Keywords: graph theory, identifying codes, adaptive identification, perfect codes, fault-detection.

Mathematical Subjects Classification: 05C99, 05C70, 94B60, 94C12.

Corresponding Author:

Julien Moncel
INPG – Laboratoire Leibniz
46 Avenue Felix Viallet
38031 Grenoble cedex - France
e-mail: julien.moncel@imag.fr
fax: +33 4 76 57 49 65

1 Introduction and motivations

Given a connected undirected graph $G = (V, E)$ and an integer $r \geq 1$, we define $B_r(v)$, the *ball* of radius r centered at $v \in V$, by

$$B_r(v) = \{x \in V : d(x, v) \leq r\},$$

where $d(x, v)$ denotes the number of edges in any shortest path between x and v. For short, a ball of radius r will be called an *r-ball* in the following. Whenever $d(x, v) \leq r$, we say that x and v *r-cover* each other (or simply *cover* if there is no ambiguity). A set $X \subseteq V$ covers a set $Y \subseteq V$ if every vertex in Y is covered by at least one vertex in X.

A *code* C is a nonempty set of vertices, and its elements are called *codewords*. For each vertex $v \in V$, we denote by

$$K_{C,r}(v) = C \cap B_r(v)$$

the set of codewords which r-cover v. Two vertices v_1 and v_2 with

$$K_{C,r}(v_1) \neq K_{C,r}(v_2)$$

are said to be *r-separated*, or *separated*, by code C.

A code C such that $|K_{C,r}(v)| \geq 1$ for all $v \in V$ is called an *r-covering* code of G (it is sometimes also called an r-dominating set of G). In other words, the set of vertices V is r-covered by C.

A code C such that $|K_{C,r}(v)| \leq 1$ for all $v \in V$ is called an *r-packing* (of r-balls) in G. In other words, the r-balls centered at vertices of C are all pairwise disjoint.

A code being both an r-covering code and an r-packing of G is called an *r-perfect code*.

A code C is called *r-identifying* (or simply *identifying* if there is no ambiguity), if the sets $K_{C,r}(v), v \in V$, are all nonempty and distinct [15]. In other words, all vertices must be r-covered and pairwise r-separated by C.

Remark 1. For given graph $G = (V, E)$ and integer r, there exists an r-identifying code $C \subseteq V$ if and only if

$$\forall v_1, v_2 \in V\ (v_1 \neq v_2),\ B_r(v_1) \neq B_r(v_2).$$

If this holds, we say that G is *r-identifiable*, or *identifiable*. In the following, we consider only identifiable graphs.

The motivations come, for instance, from sensor networks, where identifying codes are used to devise location and detection systems [19]. Identifying codes were originally defined for the purpose of fault diagnosis in multiprocessor systems [15]. Such a system can be modeled as a graph, where vertices are processors, and edges are links between processors. Assume that at most one of the processors is faulty, and that we wish to test the system and locate the faulty processor. For this purpose, some processors (constituting the code) will be selected and assigned the task of testing their neighbourhoods (i.e., the vertices at distance at most r). Whenever a selected processor (i.e., a codeword) detects a fault, it sends an alarm signal, saying that one element in its neighbourhood is faulty. We require that we can uniquely tell the location of the faulty processor based only on knowledge of the set of codewords which gave alarm.

Using r-identifying codes can be seen as follows: given the graph G and the code C, we ask, in one step, $|C|$ queries to the codewords ("is there a faulty vertex in $B_r(c)$?", for all $c \in C$). Thanks to the global answer from the codewords, we can locate the faulty vertex (if there is one) or conclude that all vertices work correctly.

Adaptive identification [18, Sec. 1.2.7] consists in asking the queries *one after the other;* this allows to choose a new query according to the answers we received so far. This can also be seen as a game, where the first player secretly chooses a vertex to be faulty in a graph, or no vertex, and the second player tries to locate it by asking queries of the type

$$\text{"is there a faulty vertex in } B_r(v) \text{ ?"}$$

for $v \in V$. If the graph is identifiable, then the second player will always succeed. In the following, "query" and "ball" will be equivalent.

We denote by $i_r(G)$ the minimum cardinality of an r-identifying code in a graph G; in adaptive identification, we want to minimize the maximum number of queries required for identification, and we denote by $a_r(G)$ this minimum number. Obviously, for all $r \geq 1$ and all r-identifiable graphs G, we have

$$a_r(G) \ \leq \ i_r(G).$$

Sometimes we also consider the *density* of such codes, which is simply the ratio of $a_r(G)$ (or $i_r(G)$) over the number of vertices of G. We mentioned previously that any r-identifying code must r-cover all vertices. In adaptive

r-identification, it may happen that all vertices have to be r-covered by queries (for instance, if there is no faulty vertex in the graph), but this will not always be the case.

We illustrate adaptive identification with the following detailed example, which will be the basis for several subsequent generalizations.

Example 1. Let $T_{5,5}$ be the 5×5 torus in the square lattice (see Figure 1), that is to say the graph having vertex set

$$V = \{(i, j) : 0 \leq i \leq 4, 0 \leq j \leq 4\},$$

and edge set

$$E = \{\{(i, j), (i, j + 1)\}, \{(i, j), (i + 1, j)\} : 0 \leq i \leq 4, 0 \leq j \leq 4\},$$

with all sums carried modulo 5.

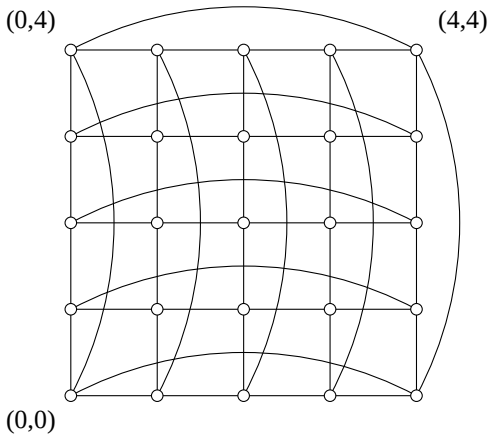

Figure 1: The 5×5 torus, $T_{5,5}$.

It is known that this graph admits a 1-perfect code, that is, a set of five balls of radius one (and cardinality five) which do not intersect and contain all 25 vertices (see Figure 2).

Since the minimum density of a 1-identifying code in the infinite square grid is equal to 0.35 [1], the cardinality of a 1-identifying code in $T_{5,5}$ is at least $\lceil 0.35 \times 25 \rceil = 9$. Indeed, suppose that there exists a 1-identifying code

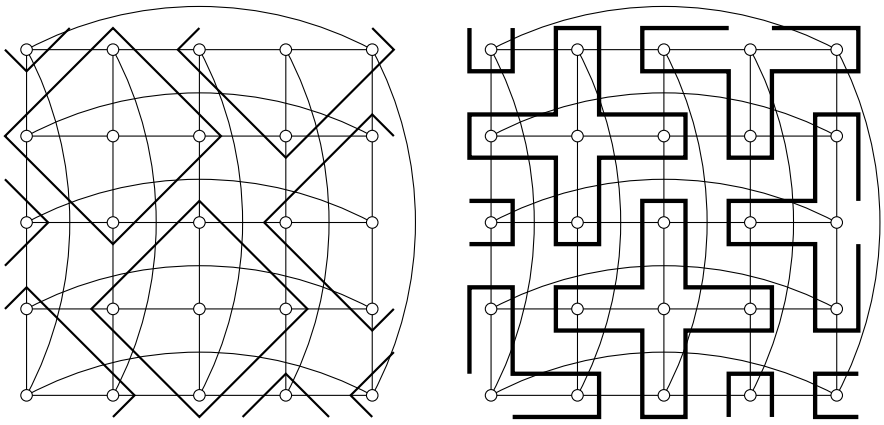

Figure 2: The 5×5 torus and a 1-perfect code, with two different representations of the balls of radius 1.

C in $T_{5,5}$ with smaller cardinality. Then, by tiling torii, one can construct a 1-identifying code of density $\frac{|C|}{25} < 0.35$ in the infinite grid, contradicting [1]. With adaptive 1-identification however, we can locate a faulty vertex, or conclude that there is none, with at most 7 queries, as we now show (we will show later in Theorem 2 that 7 is the best possible, i.e., $a_1(T_{5,5}) = 7$).

First, we consider a ball of radius one, for instance

$$B_1((2,2)) = \{(2,2),(2,1),(2,3),(1,2),(3,2)\},$$

and we assume that it contains at most one faulty vertex, and that there is no faulty vertex outside. How to locate this vertex or know that there is none? This will be the heart of adaptive identification, as we shall see later. Here, it is easy to see that we can answer with at most three queries (see Figure 3):

Q1) is there a faulty vertex in $B_1((1,1))$?
If YES,
    Q2) is there a faulty vertex in $B_1((3,1))$?
    If YES, the faulty vertex is $(2,1)$.
    If NO, the faulty vertex is $(1,2)$.
If NO,
    Q2) is there a faulty vertex in $B_1((3,3))$?
    If YES,

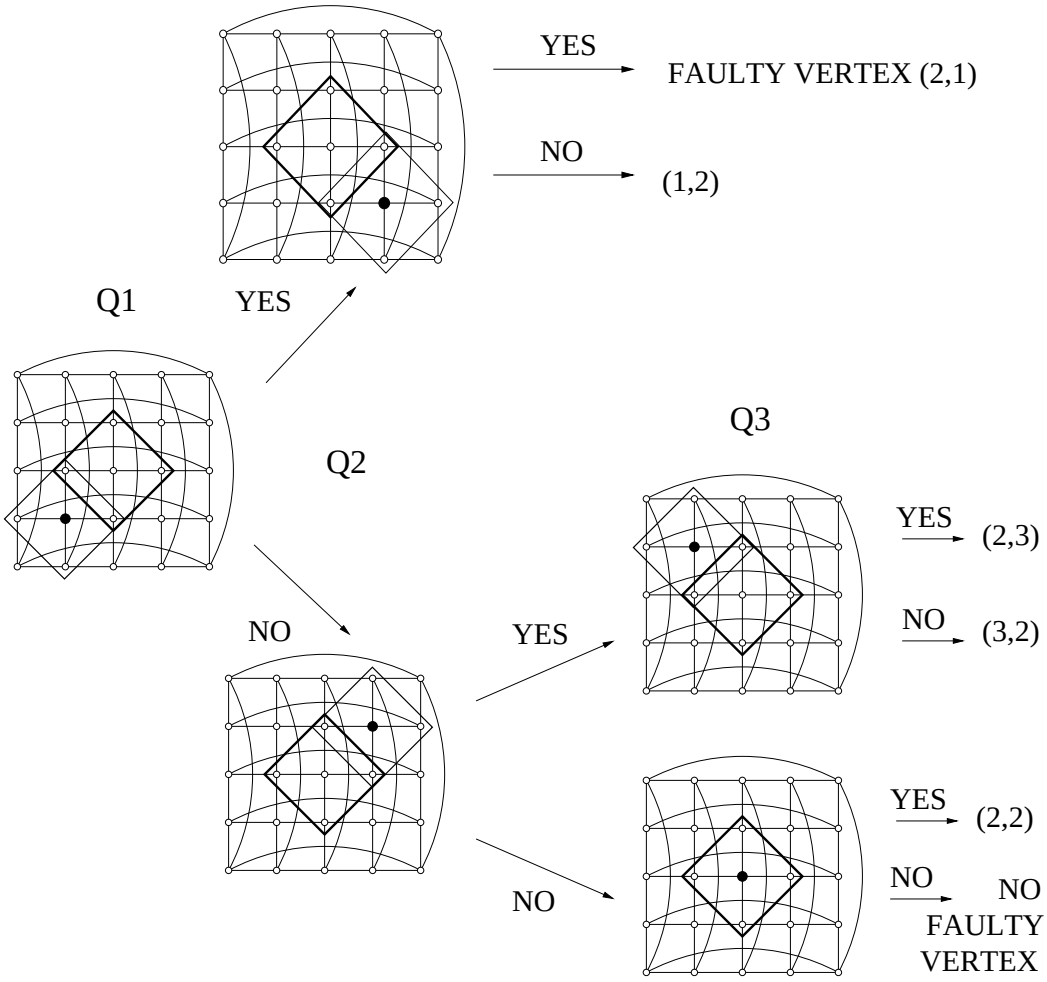

Figure 3: Three queries to $B_1((2,2))$ in the 5×5 torus $T_{5,5}$.

Q3) is there a faulty vertex in $B_1((1,3))$?
If YES, the faulty vertex is $(2,3)$.
If NO, the faulty vertex is $(3,2)$.

If NO,

Q3) is there a faulty vertex in $B_1((2,2))$?
If YES, the faulty vertex is $(2,2)$.
If NO, there is no faulty vertex.

Note that if we already know from the start that there is a faulty vertex in

$B_1((2,2))$, we still need three queries to locate it, because $\lceil \log_2 5 \rceil = 3$.

Now, how to locate a possible faulty vertex in the 5×5 torus? Consider the perfect code given in Figure 2, where the centers of the balls are $(0,0)$, $(2,1)$, $(4,2)$, $(1,3)$ and $(3,4)$. We question each of the first four balls in turn. At the first positive answer, we apply the three-query process described before, which guarantees that we need at most $4 + 3 = 7$ queries. If we get four negative answers, then we do not need to question the last ball, but instead immediately apply the three-query process, which leads also to 7 queries.

The topic of this paper is, after introducing adaptive identification, to generalize Example 1 in several directions (torii of bigger sizes, torii in the king lattice, and adaptive r-identification for $r \geq 1$), and to analyze the performances of adaptive identification with respect to classical identifying codes.

Before that, we give a somewhat extreme example, where adaptive 1-identification requires a number of queries which is only logarithmic with respect to the size of any 1-identifying code.

Example 2. Our construction is recursive (see Figure 4): we consider the

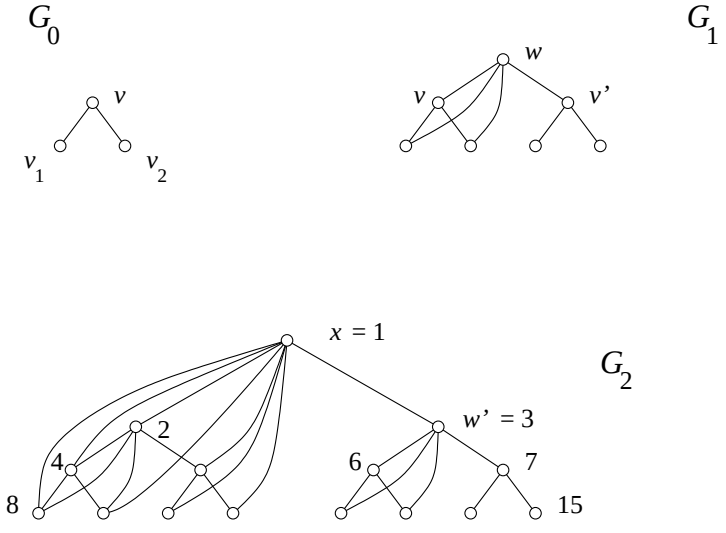

Figure 4: The recursive graphs constructed in Example 2.

graph $G_0 = (V_0, E_0)$, with $V_0 = \{v, v_1, v_2\}$ and $E_0 = \{\{v, v_1\}, \{v, v_2\}\}$, we take a copy G_0' of G_0, we consider a new vertex w, and we link w to v' and to all vertices in V_0, to obtain the graph G_1. We take a copy G_1' of G_1, we

consider a new vertex x, and we link x to w' and to all vertices in V_1, to obtain the graph G_2, and so on, until we obtain the graph G_h, which has $X_h = 2^{h+2} - 1$ vertices, which we number from 1 to X_h, going from left to right and from top to bottom.

If we ask whether there is a faulty vertex in the ball of radius one centered at vertex 1, a YES answer tells us that the vertex is either equal to 1 or 3, or belongs to V_{h-1}; whereas a NO answer shows that if it exists, it belongs to $V'_{h-1} \setminus \{3\}$: the query divides the graph into approximately two halves. Asking the second query in the appropriate half again divides the graph into two halves. This informal argument shows that, by successive dichotomies, it is possible to perform adaptive 1-identification in a number of queries which is logarithmic in X_h.

On the other hand, consider, in X_h, two consecutive "leaves" 2ℓ and $2\ell+1$ (where $2^{h+1} \leq 2\ell \leq 2^{h+2} - 2$). The only vertices that can 1-separate these two leaves are themselves, which shows that at least one of them must belong to a 1-identifying code. This implies that any 1-identifying code has a size which is linear in X_h.

The paper is structured as follows: the next section contains general bounds on $a_r(G)$ for a regular graph G, Section 3 is dedicated to torii in the square lattice, Section 4 is dedicated to torii in the king lattice, and we conclude the paper by giving perspectives for further research in the area of adaptive identification in graphs.

2 General bounds

Recall that $a_r(G)$ denotes the minimum number of queries for an adaptive r-identifying code in an r-identifiable graph $G = (V, E)$. We also define the following parameters:

- $c_r(G)$ is the maximum cardinality of an r-packing of G,
- $\gamma_r(G)$ is the minimum cardinality of an r-covering code in G.

Note that $c_r(G) \leq \gamma_r(G)$, with equality if r-perfect codes exist in G.

If G is r-regular, that is, if all r-balls have the same cardinality, then, if we denote by $v_r(G)$ this cardinality, we have

$$c_r(G) \times v_r(G) \leq |V| \leq \gamma_r(G) \times v_r(G),$$

with equalities if r-perfect codes exist in G.

Let also $d_r(G)$ be the minimum number of queries to identify an r-ball in G, i.e., the minimum number of queries for identifying a given r-ball B_r in G, assuming that there is no faulty vertex outside B_r (hence there is one or zero faulty vertex in B_r). In the introduction we have already shown that $d_1(T_{5,5}) = 3$ (see Example 1).

Theorem 1 *Let $r \geq 1$ and let G be an r-regular r-identifiable graph. Then we have*

$$c_r(G) - 1 + \lceil \log_2(v_r(G) + 1) \rceil \ \leq \ a_r(G) \ \leq \gamma_r(G) - 1 + d_r(G).$$

Proof : Let $G = (V, E)$ be an r-regular r-identifiable graph. For the lower bound, it suffices to notice that it is possible that the answer to the first $c_r(G) - 1$ queries is NO. Indeed, by definition, $c_r(G) - 1$ r-balls $B_r(x_1), \ldots, B_r(x_{c_r(G)-1})$ cannot cover the whole vertex set of G, hence there may be a faulty vertex in $V \setminus (B_r(x_1) \cup \ldots \cup B_r(x_{c_r(G)-1}))$, or no faulty vertex at all. Hence after the $(c_r(G) - 1)$-th query, there remain at least $v_r(G)$ uncovered vertices, and there are at least $v_r(G) + 1$ possibilities: either one of these uncovered vertices is the faulty vertex, or there is no faulty vertex at all in the graph. To discriminate between these $v_r(G) + 1$ possibilities, we need at least $\lceil \log_2(v_r(G) + 1) \rceil$ queries.

For the upper bound, let us consider $\{x_1, \ldots, x_{\gamma_r(G)}\}$ an r-covering code of minimum cardinality in G. Let us consider the following strategy: we ask the query "is there a faulty vertex in $B_r(x_i)$?" for $i = 1, \ldots, \gamma_r(G) - 1$, until the answer is YES. If we get a positive answer at the k-th query, $1 \leq k \leq \gamma_r(G) - 1$, then we know that there is a faulty vertex in $B_r(x_k)$ and we find it with, by definition, at most $d_r(G)$ queries. Hence we located the faulty vertex in at most $\gamma_r(G) - 1 + d_r(G)$ queries. If the answer to the first $\gamma_r(G) - 1$ queries is NO, then we know that either there is a faulty vertex in $B_r(x_{\gamma_r(G)})$, or there is no faulty vertex at all. Hence if we identify $B_r(x_{\gamma_r(G)})$, then we are done. By definition, this can be done it at most $d_r(G)$ queries, which leads to a total number of queries of $\gamma_r(G) - 1 + d_r(G)$. □

3 Torii in the square lattice

In Sections 3 and 4, we assume that the dimensions of the torii are "large enough" with respect to r.

Given two integers p and q, the $p \times q$ torus in the square lattice, denoted $T_{p,q}$, is the graph having vertex set

$$V = \{(i,j) : 0 \leq i \leq p-1, 0 \leq j \leq q-1\},$$

$$E = \{\{(i,j),(i,j+1)\},\{(i,j),(i+1,j)\} : 0 \leq i \leq p-1, 0 \leq j \leq q-1\},$$

with sums on the first coordinate carried modulo p, and sums on the second coordinate carried modulo q. The torus $T_{5,5}$ is depicted in Figure 1. Obviously, $v_r(T_{p,q}) = 2r^2 + 2r + 1$.

If p and q are both multiples of $2r^2+2r+1$, then there exists an r-perfect code in $T_{p,q}$ [7, 8]. In this case, we have

$$c_r(T_{p,q}) \;=\; \gamma_r(T_{p,q}),$$

and to get good lower and upper bounds on $a_r(T_{p,q})$, one could derive bounds on $d_r(T_{p,q})$ and use Theorem 1.

Hence, the next section is dedicated to computing general bounds on $d_r(T_{p,q})$, that we use in Section 3.2 to derive close bounds on — and, for many values of r, exact values of — $a_r(T_{p,q})$ in the perfect case. In Section 3.3, we give the asymptotic value of $a_r(T_{p,q})$ in the general case.

3.1 General bounds on $d_r(T_{p,q})$

Lemma 1 *Let $r \geq 1$, $p \geq 2$ and $q \geq 2$, and let $T_{p,q}$ be the $p \times q$ torus in the square lattice. Then we have*

$$\left\lceil \log_2(2r^2+2r+2) \right\rceil \;\leq\; d_r(T_{p,q}) \;\leq\; 2\left\lceil \log_2(r+1) \right\rceil + 1.$$

Proof : The lower bound is the general one for a dichotomic search in a set of cardinality $2r^2+2r+2$ (there can be either a faulty vertex – $2r^2+2r+1$ possibilities – or no faulty vertex at all). For the upper bound, one can see an r-ball in $T_{p,q}$ as the disjoint union of $r+1$ "packets", each of them containing exactly $2r+1$ vertices, except for one which contains only $r+1$ vertices (see Figure 5 for $r = 3$).

Now, to identify an r-ball $B_r((i,j))$ of $T_{p,q}$, one can use the following three-fold strategy: first, look in which packet of $B_r((i,j))$ could the faulty vertex be; then, look in which row of the candidate packet could the faulty vertex be; finally, look if there is a faulty vertex in the candidate row, and

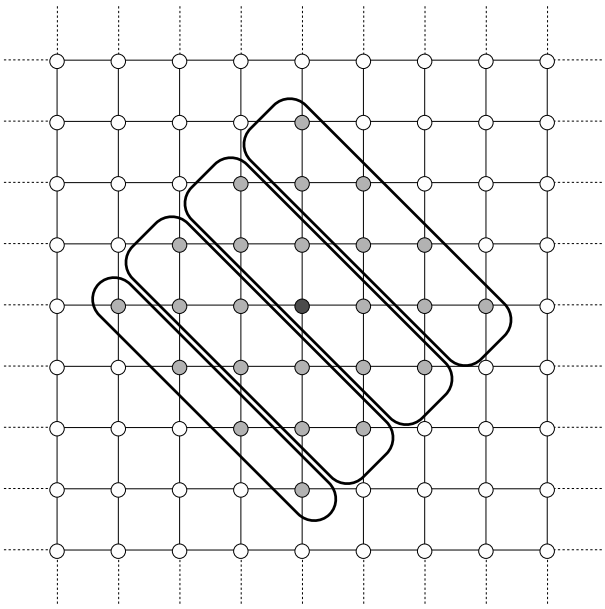

Figure 5: A 3-ball in $T_{p,q}$, seen as the disjoint union of 3 packets of 7 vertices and a packet of 4 vertices.

if yes locate it. We have to be careful with the fact that we do not know *a priori* whether there is a faulty vertex in $B_r((i,j))$ or not.

The first phase in the identification of $B_r((i,j))$ is as follows. We apply a dichotomic search using queries of the form "is there a faulty vertex in $B_r((i-k,j-k))$?", where the values of k are taken among $1,2,\ldots,r$. This provides us with the candidate packet, and uses at most $\lceil \log_2(r+1) \rceil$ queries.

We move to the second phase. If the candidate packet is the lowermost one, i.e., consists of $r+1$ vertices, then the faulty vertex, if it exists, can be trivially located in $\lceil \log_2(r+1) \rceil$ queries. Assume then that the candidate packet consists of $2r+1$ vertices. The packet is a disjoint union of $r+1$ rows, each of them containing exactly two vertices, except for one which contains only one vertex. In this packet, denote by (a,b) the unique vertex which is the center of an r-ball that contains the packet. By a dichotomic search using queries of the form "is there a faulty vertex in $B_r((a+k,b+1-k))$?", where the values of k are taken among $1,2,\ldots,r$, one can find the candidate row to contain a faulty vertex in at most $\lceil \log_2(r+1) \rceil$ queries. Note that no query covers the uppermost row, since $k \geq 1$. Hence, the candidate row is

the uppermost one if and only if all the queries in the second phase answer NO.

We move to the third phase. Here we have the candidate row and need to decide whether there is a faulty vertex, and if yes, which of the two vertices is the faulty one. If the candidate row consists of two vertices then necessarily there was a query in the second phase that answered YES, hence we know that there is a faulty vertex, and in order to locate it it suffices to check one of two vertices. If the candidate row consists of a single vertex, then a single query checks if it is faulty. We conclude that the third phase consists of one query.

Summing up the number of queries in the three phases we obtain that the overall number of queries is at most $2\lceil \log_2(r+1) \rceil + 1$. □

Note that it is easy to check that for all $r \geq 1$,

$$(2\lceil \log_2(r+1) \rceil + 1) - \left\lceil \log_2(2r^2+2r+2) \right\rceil \ \in \ \{0,1\}.$$

3.2 Perfect case

We start by the case $r = 1$, where we can directly apply Theorem 1.

Theorem 2 *Let $p \geq 5$ and $q \geq 5$ be both multiples of 5, and let $T_{p,q}$ be the $p \times q$ torus on the square lattice. Then we have*

$$a_1(T_{p,q}) \ = \ \frac{pq}{5} + 2.$$

Proof : We have already seen in the introduction of this paper (see Example 1) that

$$d_1(T_{p,q}) = \lceil \log_2(v_1(T_{p,q}) + 1) \rceil = \lceil \log_2 6 \rceil = 3.$$

This also follows immediately from Lemma 1. If p and q are both multiples of 5, then we know that there exists a 1-perfect code in $T_{p,q}$, i.e.,

$$c_1(T_{p,q}) \ = \ \gamma_1(T_{p,q}) \ = \ \frac{pq}{5}.$$

The conclusion follows from Theorem 1. □

In general, when p and q are both multiples of $2r^2+2r+1$, there exists an r-perfect code in $T_{p,q}$, and we have

$$c_r(T_{p,q}) \ = \ \gamma_r(T_{p,q}) \ = \ \frac{pq}{2r^2+2r+1}.$$

Hence, by Theorem 1 and Lemma 1, we know that

$$\frac{pq}{2r^2+2r+1} - 1 + \left\lceil \log_2(2r^2+2r+2) \right\rceil \;\leq\; a_r(T_{p,q})$$

and

$$a_r(T_{p,q}) \;\leq\; \frac{pq}{2r^2+2r+1} + 2\left\lceil \log_2(r+1) \right\rceil .$$

For infinitely many values of r, we actually have

$$\left\lceil \log_2(2r^2+2r+2) \right\rceil \;=\; 2\left\lceil \log_2(r+1) \right\rceil + 1. \tag{1}$$

For instance, computation shows that this is the case for $r = 1, 3, 6$ and 7, and for $r = 2^m - s$, $1 \leq s \leq 2^{m-2}$, $m \geq 4$. For $r = 2, 4$ and 5, we shall give ad hoc strategies showing that

$$d_r(T_{p,q}) \;=\; \left\lceil \log_2(2r^2+2r+2) \right\rceil .$$

Theorem 3 *For all $r = 1, \ldots, 7$, and for all $r = 2^m - s$, $1 \leq s \leq 2^{m-2}$, $m \geq 4$, we have*

$$a_r(T_{p,q}) \;=\; \frac{pq}{2r^2+2r+1} - 1 + \left\lceil \log_2(2r^2+2r+2) \right\rceil ,$$

for all p and q which are both multiples of $2r^2+2r+1$.

Proof : The cases $r = 1, 3, 6, 7$ and $r = 2^m - s$ follow from Theorem 1, Lemma 1 and (1). For the cases $r = 2, 4, 5$, we exhibit (see Figures 6, 7 and 8) ad hoc strategies showing that

$$d_r(T_{p,q}) \;=\; \left\lceil \log_2(2r^2+2r+2) \right\rceil .$$

□

For other values of r, however, the bounds of Lemma 1 differ by at most 1, hence we have:

Theorem 4 *For all $r \geq 8$, we have*

$$a_r(T_{p,q}) - \left(\frac{pq}{2r^2+2r+1} - 1 + \left\lceil \log_2(2r^2+2r+2) \right\rceil \right) \;\in\; \{0, 1\},$$

provided that p and q are both multiples of $2r^2+2r+1$.

Proof : Straightforward from Lemma 1 and the remark following its proof. □

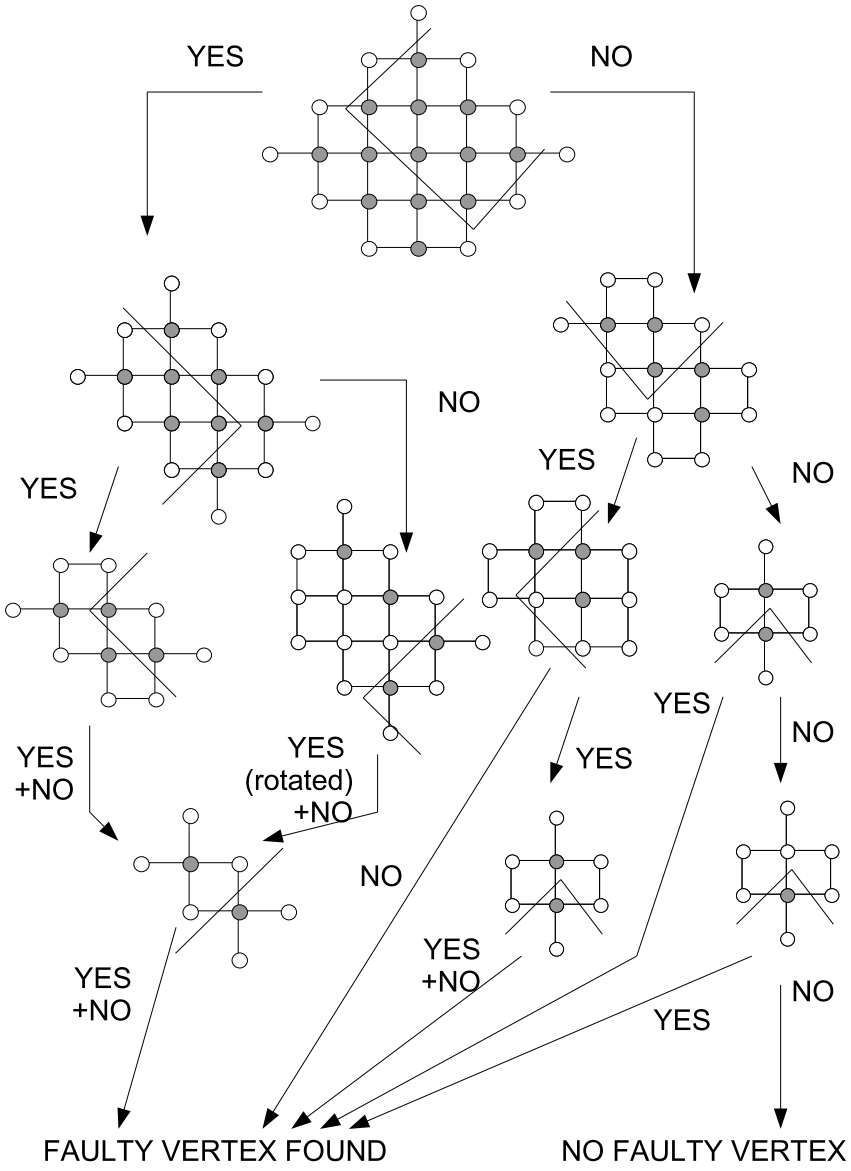

Figure 6: Proof of $d_2(T_{p,q}) = 4$.

3.3 General case

We consider the general case, of a torus which does not necessarily admit an r-perfect code. We have the following asymptotic result:

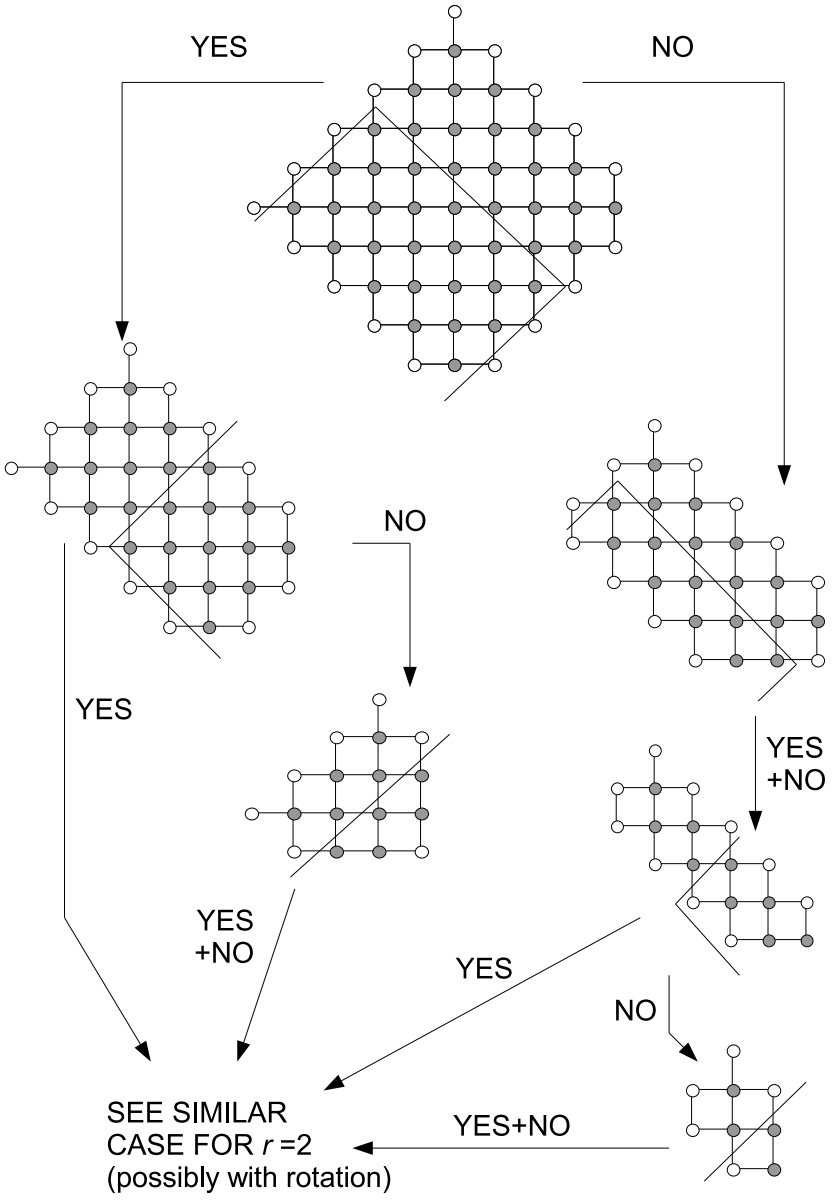

Figure 7: Proof of $d_4(T_{p,q}) = 6$.

Theorem 5 *Let $r \geq 1$, $p \geq 2$ and $q \geq 2$, and let $T_{p,q}$ be the $p \times q$ torus in*

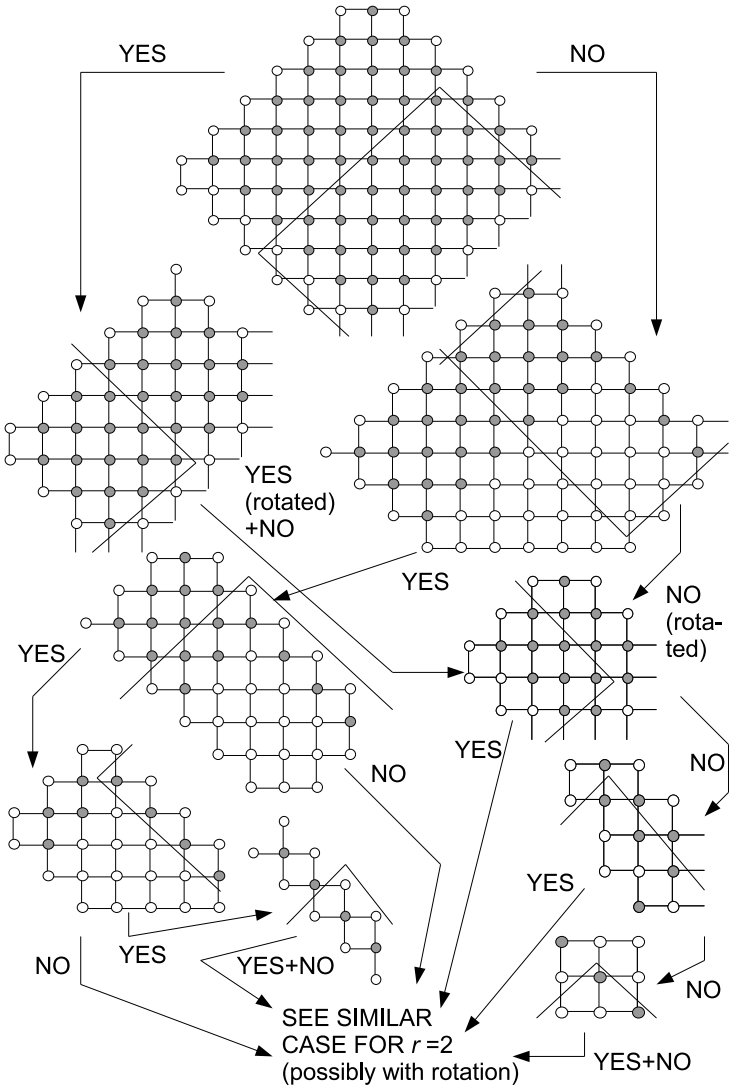

Figure 8: Proof of $d_5(T_{p,q}) = 6$.

the square lattice. Then we have

$$a_r(T_{p,q}) \;=\; \frac{pq}{2r^2+2r+1} + \Theta(p+q).$$

Proof : By Theorem 1, we know that

$$a_r(G) \;\geq\; c_r(G) - 1 + \lceil \log_2(v_r(G)+1) \rceil$$

for any r-regular r-identifiable graph G. Let us define p' as the greatest multiple of $2r^2+2r+1$ which is smaller than or equal to p. Similarly, let us define q' as the greatest multiple of $2r^2+2r+1$ which is smaller than or equal to q. Clearly, we have

$$c_r(T_{p,q}) \ \geq \ c_r(T_{p',q'}) \ = \ \frac{p'q'}{2r^2+2r+1}.$$

Now, since $p' \geq p-(2r^2+2r+1)$ and $q' \geq q-(2r^2+2r+1)$, we have

$$\frac{p'q'}{2r^2+2r+1} \ = \ \frac{pq}{2r^2+2r+1}+\Omega(p+q).$$

Since $\lceil \log_2(v_r(G)+1) \rceil = O(1)$, we have

$$a_r(T_{p,q}) \ = \ \frac{pq}{2r^2+2r+1}+\Omega(p+q).$$

Still by Theorem 1, we know that

$$a_r(G) \ \leq \ \gamma_r(G)-1+d_r(G)$$

for any r-regular r-identifiable graph G. Let us define p'' as the smallest multiple of $2r^2+2r+1$ which is greater than or equal to p. Similarly, let us define q'' as the smallest multiple of $2r^2+2r+1$ which is greater than or equal to q. Clearly, we have

$$\gamma_r(T_{p,q}) \ \leq \ \gamma_r(T_{p'',q''}) \ = \ \frac{p''q''}{2r^2+2r+1}.$$

Now, since $p'' \leq p+(2r^2+2r+1)$ and $q'' \leq q+(2r^2+2r+1)$, we have

$$\frac{p''q''}{2r^2+2r+1} \ = \ \frac{pq}{2r^2+2r+1}+O(p+q).$$

Since $d_r(T_{p,q})$ is clearly independent of p and q, $d_r(T_{p,q})=O(1)$, and we have

$$a_r(T_{p,q}) \ = \ \frac{pq}{2r^2+2r+1}+O(p+q),$$

which concludes the proof. □

This confirms the intuitive idea that an optimal strategy consists in, first, r-covering the graph, and then locating the faulty vertex with a dichotomization of the first r-ball answering YES. Asymptotically, the cost of the optimal strategy is then approximately equal to the cardinality of an optimal r-covering code of the graph, that is to say

$$\frac{pq}{2r^2+2r+1} + \Theta(p+q),$$

which has density tending to

$$\frac{1}{2r^2+2r+1}$$

as p and q tend to infinity. One can compare this density with the non-adaptive case, where we know that any minimum r-identifying code has density greater than or equal to

$$\frac{3}{8r+4}$$

as p and q tend to infinity, for all $r \geq 1$ [3]. When $r = 1$, it is known that the best possible density is equal to 0.35 [1], vs 0.2 for adaptive 1-identification.

4 Torii in the king lattice

Given two integers p and q, the $p \times q$ torus in the king lattice, denoted $T^{\mathrm{k}}_{p,q}$, is the graph having vertex set

$$V = \{(i,j) : 0 \leq i \leq p-1, 0 \leq j \leq q-1\},$$

and edge set

$$\begin{aligned} E &= \{\{(i,j),(i,j+1)\},\{(i,j),(i+1,j)\} : 0 \leq i \leq p-1, 0 \leq j \leq q-1\} \\ &\cup \{\{(i,j),(i+1,j+1)\} : 0 \leq i \leq p-1, 0 \leq j \leq q-1\} \\ &\cup \{\{(i,j),(i+1,j-1)\} : 0 \leq i \leq p-1, 0 \leq j \leq q-1\} \end{aligned}$$

with sums on the first coordinate carried modulo p, and sums on the second coordinate carried modulo q.

We present results similar to those for torii in the square lattice. If p and q are both multiples of $2r+1$, then there exists a perfect r-code in $T^{\mathrm{k}}_{p,q}$. In this case, we have

$$c_r(T^{\mathrm{k}}_{p,q}) \;=\; \gamma_r(T^{\mathrm{k}}_{p,q}),$$

and to get good lower and upper bounds on $a_r(T^{\mathrm{k}}_{p,q})$, one could derive bounds on $d_r(T^{\mathrm{k}}_{p,q})$ and use Theorem 1.

The next section is dedicated to computing general bounds on $d_r(T^{\mathrm{k}}_{p,q})$, that are used in Section 4.2 to derive close bounds on — and, for many values of r, exact values of — $a_r(T^{\mathrm{k}}_{p,q})$ in the perfect case. Our computation of $d_r(T^{\mathrm{k}}_{p,q})$ involves a result on a similar problem studied by M. Ruszinkó [20]. In Section 4.3, we give the asymptotic value of $a_r(T^{\mathrm{k}}_{p,q})$ in the general case.

4.1 General bounds on $d_r(T^{\mathbf{k}}_{p,q})$

A ball of radius r in the king lattice can be seen as a square of side $2r+1$ (see Figure 9). Hence, our problem is equivalent to finding out if a given square, containing at most one faulty vertex, indeed contains one, and if yes, then locate it, using queries of the following kind. In each query, two numbers

$$1 \le m, n \le 2r+1$$

and two symbols

$$s(x), s(y) \in \{\le, \ge\}$$

are chosen. The query has the form:

"is there a faulty vertex in the rectangle $\{(x,y) :\ x\ s(x)\ m,\ y\ s(y)\ n\}$?"

A similar problem has been proposed by G. O. H. Katona [16], and studied by M. Ruszinkó [20]. Let us name our problem the "identification problem" and the problem discussed in [20], the "search problem". The search problem is different from the identification problem in three ways:

1. The search problem considers the dichotomization problem in a rectangle $\{(x,y) :\ 1 \le x \le a,\ 1 \le y \le b\}$, which is not necessarily a square.

2. The search problem assumes that there is *exactly* one faulty vertex in the rectangle (rather than *at most one* faulty vertex in the identification problem).

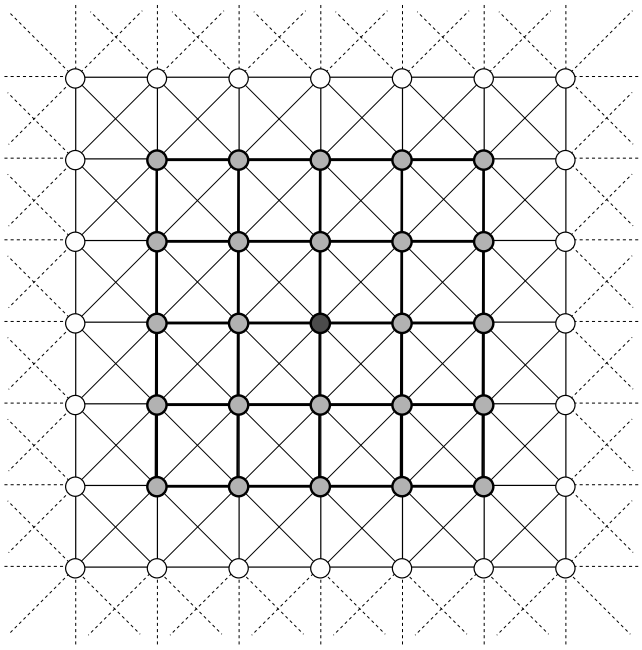

Figure 9: A 2-ball in $T^{\mathrm{k}}_{p,q}$, seen as a square of side 5.

3. In the search problem, all the queries are of the form: "is the faulty vertex in the rectangle $\{(x, y) : \ x \leq m, \ y \leq n\}$?", where $1 \leq m \leq a$, $1 \leq n \leq b$. In the following, we shall refer to these queries as "type 2 queries", while queries corresponding to the identification problem will be called "type 1 queries".

Denote by $Q(a, b)$ the minimum number of type 2 queries required to locate the faulty vertex in an $a \times b$ rectangle for the search problem. We shall use the following results.

Theorem 6 (Ruszinkó [20]) *For a natural number x, denote $\ell(x)$ as the fractional part of $\log_2 x$, that is to say*

$$\ell(x) \ := \ \log_2(x) - \lfloor \log_2(x) \rfloor \, .$$

If the natural numbers a and b satisfy at least one of the following conditions:

$$\ell(a) + \ell(b) > 1 \tag{2}$$
$$\ell(a) + \ell(b) \leq 0.8 \tag{3}$$
$$\ell(a) \leq 0.49 \text{ and } \ell(b) \leq 0.49 \tag{4}$$
$$a = b \text{ and } a \leq 180, \tag{5}$$

then there exists an algorithm using at most $\lceil \log_2(ab) \rceil$ queries which locates the faulty vertex for the search problem in an $a \times b$ rectangle, i.e.,

$$Q(a,b) \;=\; \lceil \log_2(ab) \rceil \,.$$

We note that the range of values of a and b for which $\lceil \log_2(ab) \rceil$ queries are sufficient is extended in [20] beyond Theorem 6. This extension is however small and we shall not use it here.

Theorem 6 could be easily used for the identification problem if we were only guaranteed that a faulty vertex exists. One way to overcome this is to add a query that questions the entire ball. However, in many cases this extra query is unnecessary. The following lemma is the analogue of Lemma 1 for the king lattice.

Lemma 2 *Let $r \geq 1$, $p \geq 2$ and $q \geq 2$, and let $T^k_{p,q}$ be the $p \times q$ torus in the king lattice. Then we have*

$$\begin{aligned} \left\lceil \log_2((2r+1)^2+1) \right\rceil \;\leq\; d_r(T^k_{p,q}) \;&\leq\; Q(2r+1, 2r+2) \\ &\leq\; \lceil \log_2(2r+1) \rceil + \lceil \log_2(2r+2) \rceil \,. \end{aligned}$$

In order to prove it, we first need the following lemma.

Lemma 3 *Let $\mathcal{A}$ be an optimal algorithm for the search problem, i.e., an algorithm that locates the faulty vertex in an $a \times b$ rectangle using at most $Q(a,b)$ queries. Denote by $x = (a,b)$ the rightmost and uppermost vertex of the rectangle. Then no query covers x. Furthermore, x is the faulty vertex if and only if all the queries answer NO.*

Proof : We first show that since $\mathcal{A}$ is optimal, then no query covers x. Indeed, the only query that contains x is the query that questions the entire rectangle, and it is known *a priori* that the answer to this query is YES. This

query is thus unnecessary and, if posed, can be removed from any optimal algorithm.

Since no query contains x, all the queries answer NO if x is the faulty vertex. For the other direction, assume that all the queries answer NO. In each step of the algorithm, the candidate set of faulty vertices is the set of vertices that have not been covered by any of the preceding queries. Since x is not covered by any of the queries, x belongs to this set all along the execution of $\mathcal{A}$. By the end of the algorithm, the candidate set of faulty vertices contains exactly one vertex, which is the faulty vertex. Therefore x is the faulty vertex. □

Proof of Lemma 2 : The first inequality

$$\left\lceil \log_2((2r+1)^2+1) \right\rceil \;\leq\; d_r(T^{\mathrm{k}}_{p,q})$$

is trivial since $v_r(T^{\mathrm{k}}_{p,q}) = (2r+1)^2$. To prove the second inequality, we exhibit an algorithm $\mathcal{A}_1$ which solves the identification problem with $Q(2r+1, 2r+2)$ type 1 queries. Given a square with side $2r+1$ and vertices $(1,1), \ldots, (2r+1, 2r+1)$, we add a new line of non-faulty vertices

$$\{(1, 2r+2), (2, 2r+2), \ldots, (2r+1, 2r+2)\}$$

above the uppermost line of the square. Let $\mathcal{A}_2$ be an optimal algorithm for the search problem in this newly created rectangle. Using type 1 queries on the square, we simulate the execution of $\mathcal{A}_2$ on the rectangle in the following way. For each type 2 query

$$\text{“is there a faulty vertex in } \{(x,y) :\; x \leq m,\; y \leq n\}?\text{”},$$

where $1 \leq m \leq 2r+1$, $1 \leq n \leq 2r+2$, algorithm $\mathcal{A}_1$ produces the type 1 query

$$\text{“is there a faulty vertex in } \{(x,y) :\; x \leq m,\; y \leq \min\{2r+1, n\}\}?\text{”}$$

which is equivalent to the query

$$\text{“is there a faulty vertex in } B_r((m-r, \min\{2r+1, n\} - r))?\text{”}$$

If the square contains a faulty vertex, then it will be located by $\mathcal{A}_2$ and hence also by $\mathcal{A}_1$, since we know in advance that the line

$$\{(1, 2r+2), (2, 2r+2), \ldots, (2r+1, 2r+2)\}$$

contains no faulty vertices. If the square does not contain a faulty vertex, then the vertex $x = (2r+1, 2r+2)$ will be declared as the faulty vertex by $\mathcal{A}_2$. Indeed, by Lemma 3, no query of $\mathcal{A}_2$ covers x, therefore $\mathcal{A}_2$ will be properly executed without being bothered by the fact that the rectangle does not contain a faulty vertex: the possibility that x is the faulty vertex always remains valid. Furthermore, since the rectangle does not contain a faulty vertex, all the queries answer NO, hence, by Lemma 3, x will be declared as the faulty vertex.

The third inequality is easy: in general, to locate the faulty vertex in an $a \times b$ rectangle, one can still use a dichotomic search to find the row containing the faulty vertex (at most $\lceil \log_2 a \rceil$ queries), and then find the column containing the faulty vertex (at most $\lceil \log_2 b \rceil$ queries), hence

$$Q(a,b) \ \leq \ \lceil \log_2 a \rceil + \lceil \log_2 b \rceil$$

for all natural numbers a and b. □

We remark that in the proof of Lemma 2, we did not use the fact that in the identification problem we are free to choose the signs $s(x)$ and $s(y)$ in each query, i.e., we used only r-balls with centers (x,y) such that $x \leq r+1$, $y \leq r+1$.

Observe that it is easy to check that for all $r \geq 1$,

$$(\lceil \log_2(2r+1) \rceil + \lceil \log_2(2r+2) \rceil) - \left\lceil \log_2((2r+1)^2+1) \right\rceil \ \in \ \{0,1\}.$$

4.2 Perfect case

Theorem 6 enables us to find many values of r for which $\lceil \log_2((2r+1)^2+1) \rceil$ coincides with $Q(2r+1, 2r+2)$, which directly gives the value of $d_r(T^{\mathrm{k}}_{p,q})$ by Lemma 2.

Theorem 7 *We have*

$$d_r(T^k_{p,q}) \ = \ \left\lceil \log_2((2r+1)^2+1) \right\rceil$$

for all $1 \leq r \leq 100$, *except maybe for* $r = 22$ *and* $r = 90$, *as well as for* $r = 2^m - s$, $1 \leq s \leq 2^{m-2}$, $m \geq 7$. *Consequently, we have*

$$a_r(T^k_{p,q}) \ = \ \frac{pq}{(2r+1)^2} - 1 + \left\lceil \log_2((2r+1)^2+1) \right\rceil$$

for all $r \leq 100$, $r \neq 22$, $r \neq 90$, *and for all* $r = 2^m - s$, $1 \leq s \leq 2^{m-2}$, $m \geq 7$, *provided that* p, q *are both multiples of* $2r+1$.

Proof : For all the values of $r \leq 100$, Theorem 6 guarantees that

$$Q(2r+2, 2r+2) \;=\; \lceil 2\log_2(2r+2) \rceil .$$

Indeed, for $1 \leq r \leq 89$, we can directly apply (5) in Theorem 6, and for $91 \leq r \leq 100$ we use (2) in Theorem 6. Since we always trivially have

$$Q(2r+1, 2r+2) \leq Q(2r+2, 2r+2)$$

for all $r \geq 1$, then by Lemma 2 we have

$$d_r(T^{\mathrm{k}}_{p,q}) \;\leq\; \lceil 2\log_2(2r+2) \rceil .$$

Computation reveals that

$$\left\lceil \log_2((2r+1)^2+1) \right\rceil \;=\; \lceil 2\log_2(2r+2) \rceil$$

for all $1 \leq r \leq 100$, except for $r = 2, 5, 22, 90$. The cases $r = 2, 5$ are given in Figures 10 and 11, which show that

$$d_2(T^{\mathrm{k}}_{p,q}) \;=\; 5$$

and

$$d_5(T^{\mathrm{k}}_{p,q}) \;=\; 7,$$

and we conclude by Theorem 1 and Lemma 2. The case $r = 2^m - s$ comes from the fact that the bounds of Lemma 2 coincide for these values of r:

$$\left\lceil \log_2((2r+1)^2+1) \right\rceil \;=\; d_r(T^{\mathrm{k}}_{p,q}) \;=\; \lceil \log_2(2r+1) \rceil + \lceil \log_2(2r+2) \rceil .$$

□

However, as for the torus in the square lattice, the general bounds from Lemma 2 differ by at most 1, hence:

Theorem 8 *For all $r \geq 1$ we have*

$$a_r(T^k_{p,q}) - \left(\frac{pq}{(2r+1)^2} - 1 + \left\lceil \log_2((2r+1)^2+1) \right\rceil \right) \;\in\; \{0, 1\},$$

provided that p and q are both multiples of $2r+1$.

Proof : Straightforward from Lemma 2 and the observation following its proof. □

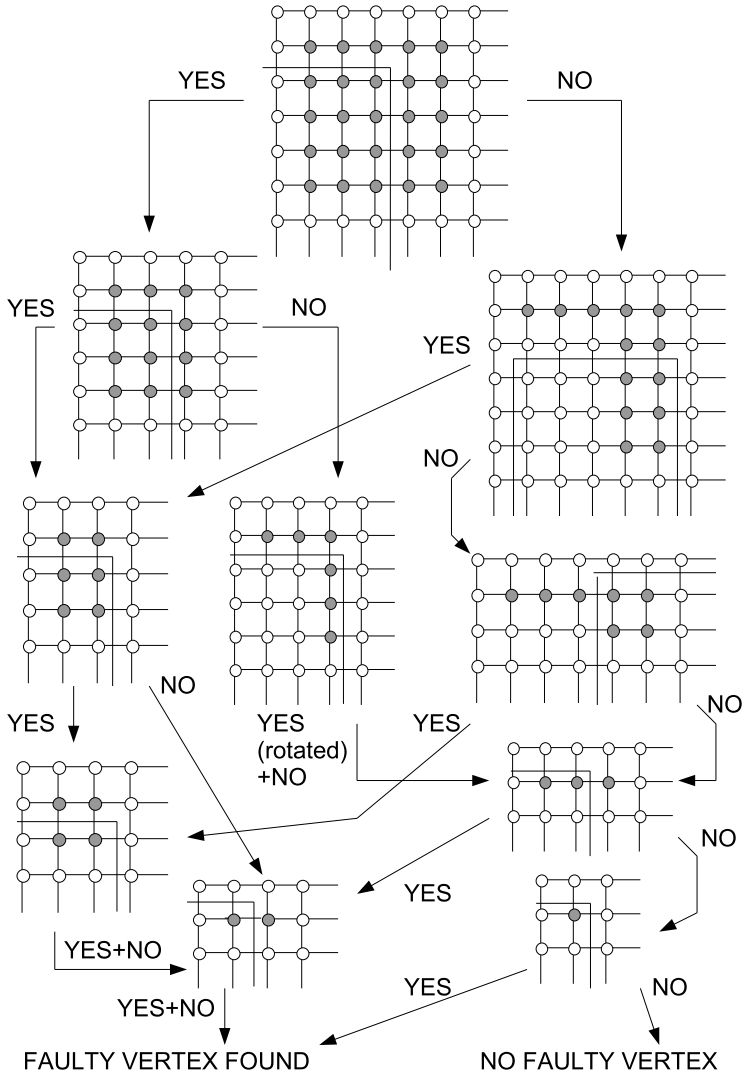

Figure 10: Proof of $d_2(T^{\mathrm{k}}_{p,q}) = 5$.

4.3 General case

Theorem 9 *Let $r \geq 1$, $p \geq 2$ and $q \geq 2$, and let $T^k_{p,q}$ be the $p \times q$ torus in the king lattice. Then we have*

$$a_r(T^k_{p,q}) \;=\; \frac{pq}{(2r+1)^2} + \Theta(p+q).$$

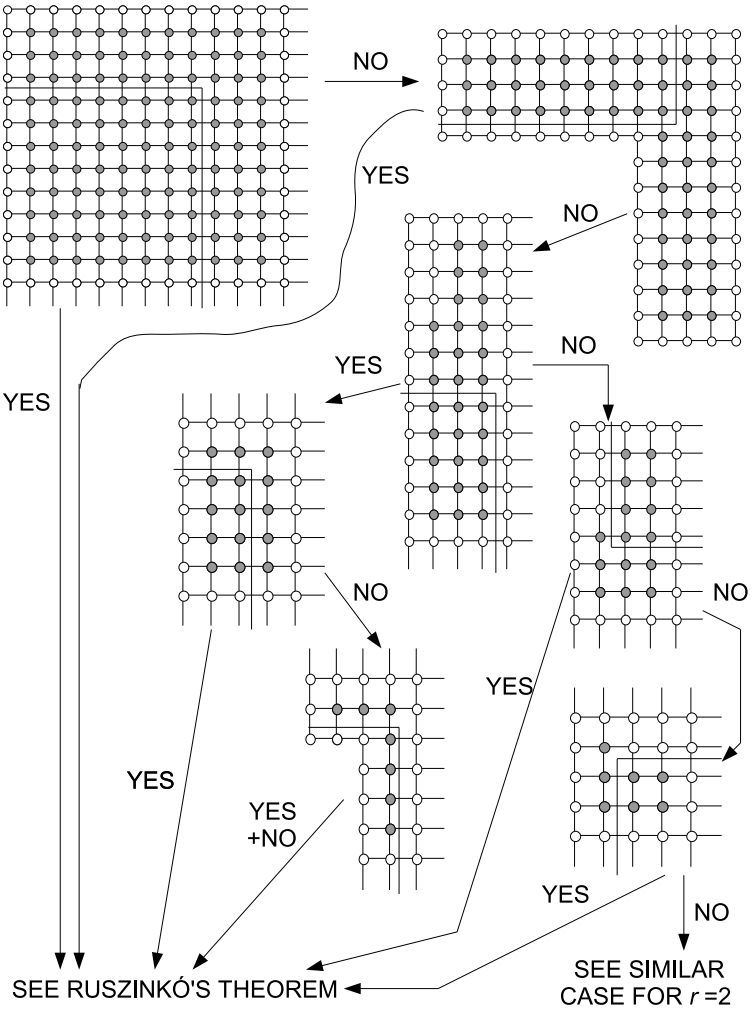

Figure 11: Proof of $d_5(T^{\mathrm{k}}_{p,q}) = 7$.

Proof : The proof is similar to that of Theorem 5. By Theorem 1, we know that

$$a_r(G) \;\geq\; c_r(G) - 1 + \lceil \log_2(v_r(G) + 1) \rceil$$

for any r-regular r-identifiable graph G. Let us define p' as the greatest multiple of $2r+1$ which is smaller than or equal to p. Similarly, let us define q' as the greatest multiple of $2r+1$ which is smaller than or equal to q.

Clearly, we have

$$c_r(T^{\mathrm{k}}_{p,q}) \;\geq\; c_r(T^{\mathrm{k}}_{p',q'}) \;=\; \frac{p'q'}{(2r+1)^2}.$$

Now, since $p' \geq p-(2r+1)$ and $q' \geq q-(2r+1)$, we have

$$\frac{p'q'}{(2r+1)^2} \;=\; \frac{pq}{(2r+1)^2} + \Omega(p+q).$$

Since $\lceil \log_2(v_r(G)+1) \rceil = O(1)$, we have

$$a_r(T^{\mathrm{k}}_{p,q}) \;=\; \frac{pq}{(2r+1)^2} + \Omega(p+q).$$

Still by Theorem 1, we know that

$$a_r(G) \;\leq\; \gamma_r(G) - 1 + d_r(G)$$

for any r-regular r-identifiable graph G. Let us define p'' as the smallest multiple of $2r+1$ which is greater than or equal to p. Similarly, let us define q'' as the smallest multiple of $2r+1$ which is greater than or equal to q. Clearly, we have

$$\gamma_r(T^{\mathrm{k}}_{p,q}) \;\leq\; \gamma_r(T^{\mathrm{k}}_{p'',q''}) \;=\; \frac{p''q''}{(2r+1)^2}.$$

Now, since $p'' \leq p+(2r+1)$ and $q'' \leq q+(2r+1)$, we have

$$\frac{p''q''}{(2r+1)^2} \;=\; \frac{pq}{(2r+1)^2} + O(p+q).$$

Since $d_r(T^{\mathrm{k}}_{p,q})$ is clearly independent of p and q, we have $d_r(T^{\mathrm{k}}_{p,q}) = O(1)$, and

$$a_r(T^{\mathrm{k}}_{p,q}) \;=\; \frac{pq}{(2r+1)^2} + O(p+q),$$

which concludes the proof. □

Hence, the density of an optimal adaptive r-identifying code in $T^{\mathrm{k}}_{p,q}$ tends to

$$\frac{1}{(2r+1)^2}$$

as p and q tend to infinity. One can compare this density with the non-adaptive case, where we know that a minimum r-identifying code has density tending to

$$\frac{2}{9}$$

if $r = 1$ [5] and

$$\frac{1}{4r}$$

if $r > 1$ [4], as p and q tend to infinity.

5 Conclusions and perspectives

In this paper we introduced adaptive identification in graphs, that we studied on torii in the square and in the king lattices. For some values of r and p, q, we gave the exact values of $a_r(T_{p,q})$ and $a_r(T^{\mathrm{k}}_{p,q})$. In general, if G is an r-regular graph, the value of $a_r(G) - \gamma_r(G)$ strongly depends on $d_r(G)$, which is the minimum number of queries required to identify an r-ball B_r in G, assuming that there is no faulty processor outside B_r. The computation of $d_r(G)$ is closely related to a Rényi-type search problem, proposed by G. O. H. Katona and studied by M. Ruszinkó [20] and M. Ruszinkó and G. Tardos [21]. In particular, we used results of [20] to compute the exact value of $d_r(T^{\mathrm{k}}_{p,q})$.

We do not know the exact value of $d_r(G)$ for all values of r when G is a torus in the square or in the king lattice, and we leave here as an open problem the computation of $d_r(T_{p,q})$ and $d_r(T^{\mathrm{k}}_{p,q})$ for all $r \geq 1$. One can also consider other types of grids, like the triangular or the hexagonal grids, which were already studied in the non-adaptive case [3, 5, 10]. The study of the hypercube could also be of interest, since it is maybe the most studied topology for identification [2, 12, 13, 15] (and for many other coding problems too). In the case of the hypercube, we predict that things do not behave so nicely as in the 2-dimensional square or king lattice. Indeed, like multidimensional square grids, hypercubes are $(0-2)$-graphs, that is to say that the intersection of two 1-balls is of cardinality either 0 or 2. This means that it is not so easy to identify a 1-ball with 1-balls in the hypercube, since a 1-ball cannot split another 1-ball into two halves of roughly even cardinalities like in the case of the square or the king lattice.

We suggest also two possible generalizations of adaptive identification. The first one consists in locating not only one, but at most a fixed number

$t \geq 1$ of faulty vertices, that is to say to study adaptive $(r, \leq t)$-identifying codes. This problem has already been studied for the non-adaptive version of these codes [9, 10, 11, 12, 17]. The second perspective for further research is to consider that vertices have probabilities of defection, which would lead to minimizing the expected number of queries and not only the greatest possible number of queries (worst-case analysis). We show in Figure 12 a simple example, taken from [18, Sec. 1.2.7], where the expected number of queries is less than the greatest number of queries in the worst case. In this example, $p \in [0, 1]$ is the probability that no vertex is faulty in the path on three vertices P_3, and $\frac{1-p}{3}$ is the probability that any vertex of P_3 is faulty. It is easy to see that $a_1(P_3)$ is 2, and that an optimal adaptive strategy is to ask first "is there a faulty vertex in $B_1(x_1)$?", and then "is there a faulty vertex in $B_1(x_3)$?". However, we let the reader check that, if we take into account the probabilities, it is cleverer to ask first "is there a faulty vertex in $B_1(x_2)$?" as soon as $p > \frac{2}{5}$.

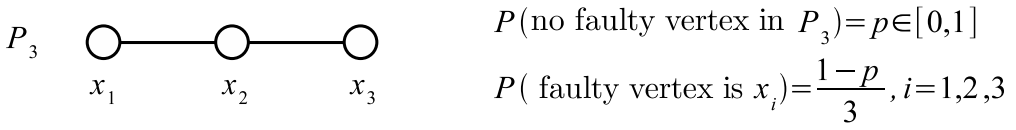

Figure 12: Assignment of defection probabilities to vertices. For $p > \frac{2}{5}$, the expected number of queries is less than in the worst case.

Observe that the path P_3 is such that $a_1(P_3) = i_1(P_3)$. An open problem is to characterize all graphs G for which $a_r(G) = i_r(G)$.

Acknowledgments

The third author would like to thank Irène Charon and Olivier Hudry for stimulating discussions.

References

[1] Y. Ben-Haim and S. Litsyn, *Exact minimum density of codes identifying vertices in the square grid*, SIAM Journal on Discrete Mathematics **19(1)** (2005), 69–82.

[2] U. Blass, I. Honkala, S. Litsyn, *On Binary Codes for Identification*, Journal of Combinatorial Designs **8** (2000), 151–156.

[3] I. Charon, I. Honkala, O. Hudry, A. Lobstein, *General Bounds for Identifying Codes in some Infinite Regular Graphs*, Electronic Journal of Combinatorics **8(1)** R39 (2001).

[4] I. Charon, I. Honkala, O. Hudry, A. Lobstein, *The Minimum Density of an Identifying Code in the King Lattice*, Discrete Mathematics **276(1-3)** (2004), 95–109.

[5] I. Charon, O. Hudry, A. Lobstein, *Identifying Codes with Small Radius in some Infinite Regular Graphs*, Electronic Journal of Combinatorics **9(1)** R11 (2002).

[6] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering codes*, Elsevier (1997).

[7] S. W. Golomb, L. R. Welch, *Algebraic coding and the Lee metric*, Proceedings of the 1968 Madison Symposium on Error Correcting Codes (1969), 175–194.

[8] S. W. Golomb, L. R. Welch, *Perfect codes in the Lee metric and the packing of polyominoes*, SIAM Journal on Applied Mathematics, **18** (1970), 302–317.

[9] S. Gravier, J. Moncel, *Construction of Codes Identifying Sets of Vertices*, Electronic Journal of Combinatorics **12(1)** R13 (2005).

[10] I. Honkala, T. Laihonen, *On identification in the triangular grid*, Journal of Combinatorial Theory Series B **91(1)** (2004), 67–86.

[11] I. Honkala, T. Laihonen, *On the Identification of Sets of Points in the Square Lattice*, Discrete and Computational Geometry **29** (2003), 139–152.

[12] I. Honkala, T. Laihonen, S. Ranto, *On Codes Identifying Sets of Vertices in Hamming Spaces*, Designs, Codes and Cryptography **24(2)** (2001), 193–204.

[13] I. Honkala, A. Lobstein, *On Identifying Codes in Binary Hamming Spaces*, Journal of Combinatorial Theory Series A **99** (2002), 232–243.

[14] I. Honkala, A. Lobstein, *On the density of identifying codes in the square lattice*, Journal of Combinatorial Theory Series B **85(2)** (2002), 297–306.

[15] M. G. Karpovsky, K. Chakrabarty, L. B. Levitin, *On a New Class of Codes for Identifying Vertices in Graphs*, IEEE Transactions on Information Theory **44(2)** (1998), 599–611.

[16] G. O. H. Katona, personal communication, 1991.

[17] T. Laihonen, S. Ranto, *Codes Identifying Sets of Vertices*, Lecture Notes in Computer Science **2227** (2001), 82–91.

[18] J. Moncel, *Codes identifiants dans les graphes*, Ph.D. dissertation (French), Université Joseph Fourier, Grenoble (2005).

[19] 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.

[20] M. Ruszinkó, *On a 2-dimensional search problem,* Journal of Statistical Planning and Inference **37(3)** (1993), 371–383.

[21] M. Ruszinkó, G. Tardos, *On a search problem in multidimensional grids,* Journal of Statistical Planning and Inference **59(1)** (1997), 101–109.

Dépôt légal : 2007 – 3ème[r] trimestre
Imprimé à l'Ecole Nationale Supérieure des Télécommunications – Paris
ISSN 0751-1345 ENST D (Paris) (France 1983-9999)

Ecole Nationale Supérieure des Télécommunications
Groupe des Ecoles des Télécommunications - membre de ParisTech
46, rue Barrault - 75634 Paris Cedex 13 - Tél. + 33 (0)1 45 81 77 77 - www.enst.fr
Département INFRES