

Adaptive Identification in Torii in the King Lattice

Yael Ben-Haim

IBM Haifa Research Lab in Haifa
Haifa University Campus, Mount Carmel
31905 Haifa (Israel)
yaelbh@il.ibm.com

Sylvain Gravier^{*†}

CNRS - Institut Fourier
100, rue des Maths – BP 74
38402 Saint Martin d’Hères (France)
sylvain.gravier@ujf-grenoble.fr

Antoine Lobstein

CNRS - LTCI UMR 5141
Institut Télécom - Télécom ParisTech
46 rue Barrault
75634 Paris Cedex 13 (France)
lobstein@telecom-paristech.fr

Julien Moncel[†]

CNRS – LAAS
Université de Toulouse;
UPS, INSA, INP, ISAE ; UT1, UTM, LAAS
7 avenue du Colonel Roche
31077 Toulouse Cedex 4 (France)
julien.moncel@iut-rodez.fr

Submitted: Apr 30, 2011; Accepted: May 13, 2011; Published: May 23, 2011
Mathematics Subject Classification: 05C99, 05C70, 94B60, 94C12

Abstract

Given a connected graph $G = (V, E)$, Let $r \geq 1$ be an integer and $B_r(v)$ denote the ball of radius r centered at $v \in V$, i.e., the set of all vertices within distance r from v . A subset of vertices $C \subseteq V$ is an r -identifying code of G (for a given nonzero constant $r \in \mathbb{N}$) if and only if all the sets $B_r(v) \cap C$ are nonempty and pairwise distinct. These codes were introduced in [7] to model a fault-detection problem in multiprocessor systems. They are also used to devise location-detection schemes in the framework of wireless sensor networks. These codes enable one to locate a malfunctioning device in these networks, provided one scans all the vertices of the code. We study here an *adaptive* version of identifying codes, which enables to perform tests dynamically. The main feature of such codes is that they may require significantly fewer tests, compared to usual static identifying codes. In this paper we study adaptive identifying codes in torii in the king lattice. In this framework, adaptive identification can be closely related to a Rényi-type search problem studied by M. Ruszinkó [11].

^{*}Partially supported by ANR/NSC Project GraTel, ANR-09-blanc-0373-01 and NSC99-2923-M-110-001-MY3, 2010–2013

[†]Partially supported by ANR Project IDEA, ANR-08-EMER-007, 2009–2011.

1 Notations

Given a connected undirected graph $G = (V, E)$ and an integer $r \geq 1$, the r -ball centered at $v \in V$, denoted $B_r(v)$, is defined by

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

where $d(x, v)$ denotes the number of edges in any shortest path between x and v . Whenever $x \in B_r(v)$, we say that x and v r -cover each other (or simply *cover* if there is no ambiguity). A set $X \subseteq V$ is said to cover 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 the 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 often also called an r -dominating set of G). In other words, the set of vertices V is r -covered by the set of r -balls centered at vertices of 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. These notations are more or less standard [6].

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 [7]. In other words, all vertices must be r -covered and pairwise r -separated by C .

Notice that, for a given graph $G = (V, E)$ and an integer r , there exists an r -identifying code $C \subseteq V$ if and only if

$$B_r(v_1) \neq B_r(v_2)$$

holds for all $v_1, v_2 \in V, v_1 \neq v_2$. In this case, we say that G is r -identifiable (or simply *identifiable* if there is no ambiguity). Obviously, not all graphs are identifiable. Structural properties of identifiable graphs are studied in [2]. In the following, the graphs we consider are all identifiable graphs.

Let $i_r(G)$ denote the minimum cardinality of an r -identifying code of an r -identifiable graph G . Let us also denote $c_r(G)$ (resp. $\gamma_r(G)$) the maximum cardinality (resp. the minimum cardinality) of an r -packing of G (resp. of an r -covering code in G). Clearly, we always have $c_r(G) \leq \gamma_r(G)$, with equality if and only if there exists an r -perfect code in G .

2 Motivations

The motivations come, for instance, from sensor networks, where identifying codes are used to devise location and detection schemes [10, 12]. These codes were originally defined for the purpose of fault diagnosis in multiprocessor systems [7]. Let G be a graph modelling a network, and assume that there is at most one “malfunctioning” vertex in G . Let C be an r -identifying code of G , and let us ask to every vertex $c \in C$ the query “is there a malfunctioning vertex in your r -ball $B_r(c)$?”. If they all answer “no”, then one may claim that there is no malfunctioning vertex in G (because C is in particular an r -covering code of G). If at least one vertex of C answers “yes”, then one may claim that there is a malfunctioning vertex in G , and may even find out which vertex is malfunctioning (because the set $\{c \in C \mid c \text{ answered “yes”}\}$ corresponds to a unique $K_{C,r}(v)$).

Adaptive identification was defined in [9, Sec. 1.2.7] and studied in [1], it consists in asking the queries *one after the other*, allowing one to choose the next query according to the answers 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 at all, and the second player tries to locate it by asking queries of the type “is there a faulty vertex in your r -ball $B_r(v)$?” for *any* vertex $v \in V$. If the graph is identifiable, then the second player will always succeed. In the following, “query” and “ball” will be equivalent.

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 always have $a_r(G) \leq i_r(G)$. As we shall see, this inequality can be strict, and $a_r(G)$ can be rather far from $i_r(G)$.

It has been shown in [1, Example 2] that there exists infinite families of graphs for which the cardinality of a minimum 1-identifying code is linear in the size of the graphs, whereas the adaptive identification can be performed in a number of queries which is linear in the size of the graphs.

In [1], adaptive identification was studied in torii in the square lattice. In this paper we derive similar results for the case of torii in the king lattice. The king lattice is a graph with points lying on \mathbb{Z}^2 , such that two distinct points A, B are neighbours if the Euclidean distance from A to B is at most $\sqrt{2}$. This is equivalent to saying that one can reach A from B using a legal move of the King in the game of chess (considering \mathbb{Z}^2 as a chessboard). Static identifying codes were already studied in this lattice [3, 4, 5]. There is a large and fast-growing literature on identifying codes in general, we refer the reader to [13] for a dynamic up-to-date bibliography.

The paper is structured as follows: the next section is dedicated to preliminary results on adaptive identification in torii in the king lattice, and Section 4 is dedicated to deriving bounds and, for many cases, exact values of $a_r(T_{p,q})$. In the sequel we will use some results of M. Ruszinkó on a similar problem [11].

3 Preliminary results

We first recall the following result from [1, Theorem 1], which is valid for any r -identifiable r -regular graph. Let G be an r -regular graph such that all r -balls of G have same cardinality, denoted $v_r(G)$. Let us define $d_r(G)$ as the minimum number of queries to identify an r -ball in G , i.e., the minimum number of queries for identifying any 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).

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

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

3.1 The king lattice

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

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

and edge set

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

with sums on the first coordinate carried modulo p , and sums on the second coordinate carried modulo q . If p and q are both multiples of $2r + 1$, then there exists a perfect r -code in $T_{p,q}$. 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.

Thus, we compute general bounds on $d_r(T_{p,q})$, that are used in Section 4 to derive close bounds on — and, for many values of r , exact values of — $a_r(T_{p,q})$ in the perfect case. Our bounds on $d_r(T_{p,q})$ involve a result on a similar problem studied by M. Ruszinkó [11], that we introduce in Section 3.2.

A ball of radius r in the king lattice can be seen as a square of side $2r + 1$ (see Figure 1). Hence, the problem of identifying an r -ball in $T_{p,q}$ 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

$$1 \leq n_1, n_2 \leq 2r + 1$$

and two symbols

$$s_1, s_2 \in \{\leq, \geq\}$$

are chosen. The query has the form:

“is there a faulty vertex in the rectangle $\{(x, y) \mid x \geq n_1, y \leq n_2\}$?”

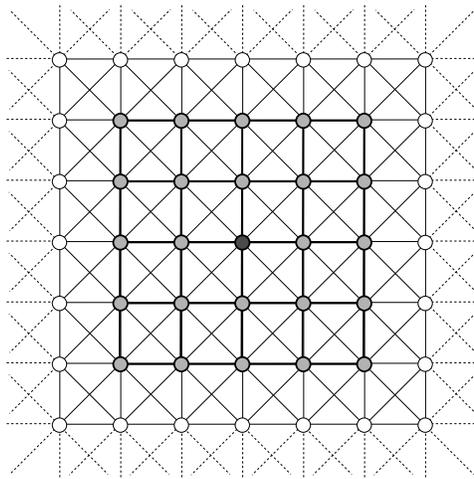


Figure 1: A 2-ball in $T_{p,q}$, seen as a square of side 5.

3.2 The search problem of Ruszinkó

A similar problem has been proposed by G. O. H. Katona [8], and studied by M. Ruszinkó [11]. Let us name our problem the “identification problem” and the problem discussed in [11], 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 $\{(x, y) \mid 1 \leq x \leq a, 1 \leq y \leq b\}$, 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).
- (3) In the search problem, all the queries are of the form: “is the faulty vertex in the rectangle $\{(x, y) \mid x \leq m, y \leq n\}$?”, 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 2 (Ruszinkó [11]) 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:

$$\ell(a) + \ell(b) > 1 \tag{1}$$

$$\ell(a) + \ell(b) \leq 0.8 \tag{2}$$

$$\ell(a) \leq 0.49 \text{ and } \ell(b) \leq 0.49 \tag{3}$$

$$a = b \text{ and } a \leq 180, \tag{4}$$

then there exists an algorithm using at most $\lceil \log_2(ab) \rceil$ type 2 queries which locates the faulty vertex for the search problem in an $a \times b$ rectangle, i.e.,

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

We note that the range of values of a and b for which $\lceil \log_2(ab) \rceil$ queries are sufficient is extended in [11] beyond Theorem 2. This extension is however small and we shall not use it here.

Theorem 2 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.

3.3 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 king lattice. Then we have

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

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

Lemma 2 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 $Q(a, b)$ type 2 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. \square

Proof of Lemma 1: The first inequality

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

is trivial since $v_r(T_{p,q})$, the volume of an r -ball in $T_{p,q}$, is equal to $(2r+1)^2$. 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 $(1, 1), \dots, (2r+1, 2r+1)$, we add a new line of non-faulty vertices

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

above the uppermost line of the square. Let \mathcal{A}_2 be an optimal algorithm for the search problem in this newly created rectangle. Note that \mathcal{A}_2 will output that $(2r+1, 2r+2)$ is the faulty vertex if and only if there is no faulty vertex in the 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

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

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

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

which is equivalent to the query

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

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

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

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 2, 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 2, 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 1, we did not use the fact that in the identification problem we are free to choose the signs s_1 and s_2 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 $r \geq 1$,

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

4 Bounds and exact values for $a_r(T_{p,q})$

4.1 Perfect case

Here we will focus on the case where p and q are both multiples of $2r + 1$. In this case, we have $c_r(T_{p,q}) = \gamma_r(T_{p,q}) = pq/v_r(T_{p,q}) = pq/(2r + 1)^2$. Theorem 2 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 $d_r(T_{p,q})$ by Lemma 1.

Theorem 3 *We have*

$$d_r(T_{p,q}) = \lceil \log_2((2r + 1)^2 + 1) \rceil$$

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

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

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 2 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 (4) in Theorem 2, and for $91 \leq r \leq 100$ we use (1) in Theorem 2. Since we always trivially have

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

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

$$d_r(T_{p,q}) \leq \lceil 2 \log_2(2r + 2) \rceil.$$

Computation reveals that

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

for all $1 \leq r \leq 100$, except for $r = 2, 5, 22, 90$. The cases $r = 2, 5$ are given in Figures 2 and 3, which show that

$$d_2(T_{p,q}) = 5$$

and

$$d_5(T_{p,q}) = 7,$$

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

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

□

However, similarly to the case of the torus in the square lattice (see [1, Theorem 4]), the general bounds from Lemma 1 differ by at most 1, hence:

Theorem 4 *For all $r \geq 1$ we have*

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

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

Proof: Straightforward from Theorem 1, Lemma 1 and the observation following its proof. □

4.2 General case

Theorem 5 *Let $r \geq 1$ be a integer. For all $p \geq 2$ and $q \geq 2$, let $T_{p,q}$ be the $p \times q$ torus in the king lattice. Then we have*

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

Proof: By Theorem 1, we know that

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

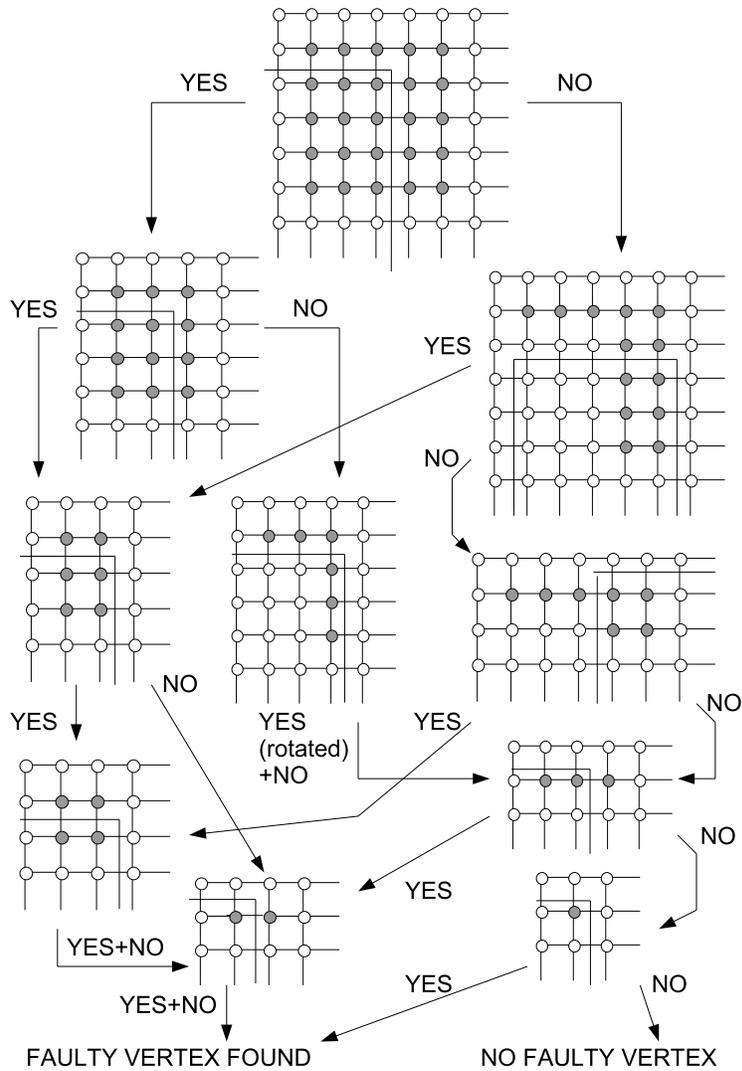


Figure 2: Proof of $d_2(T_{p,q}) = 5$.

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

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

Now, since $p' \geq p - 2r$ and $q' \geq q - 2r$, we have

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

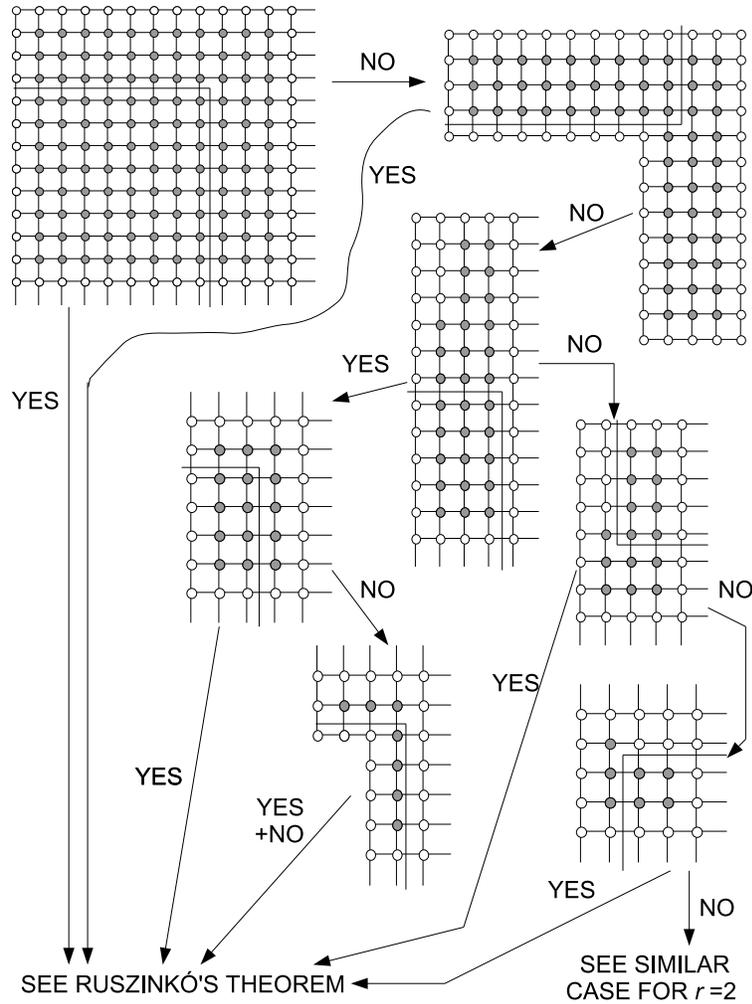


Figure 3: Proof of $d_5(T_{p,q}) = 7$.

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

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

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

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

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

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

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

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

which concludes the proof. □

Hence, the density of an optimal adaptive r -identifying code in $T_{p,q}$ tends to

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

as p and q tend to infinity, where the density is defined as the ratio between a_r and the number of vertices. One can compare this density to 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. Here again we can see that there is a significant gap (in terms of density, hence efficiency) between adaptive and non-adaptive identification.

Acknowledgments

The third author would like to thank Irène Charon and Olivier Hudry for stimulating discussions. The fourth author would like thank Simon Litsyn for his hospitality, since some part of this work was done during a stay at the Department of Electrical Engineering of the University of Tel-Aviv.

References

- [1] Y. Ben-Haim, S. Gravier, A. Lobstein, J. Moncel, *Adaptive Identification in Graphs*, Journal of Combinatorial Theory Series A **115(7)** (2008), 1114–1126.
- [2] I. Charon, I. Honkala, O. Hudry, A. Lobstein, *Structural Properties of Twin-Free Graphs*, Electronic Journal of Combinatorics **14(1)** R16 (2007).
- [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] 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.
- [8] G. O. H. Katona, personal communication, 1991.
- [9] J. Moncel, *Codes identifiants dans les graphes*, Ph.D. dissertation (French), Université Joseph Fourier, Grenoble (2005).
- [10] 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.
- [11] M. Ruszinkó, *On a 2-dimensional search problem*, Journal of Statistical Planning and Inference **37(3)** (1993), 371–383.
- [12] 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.
- [13] Online bibliography on Antoine Lobstein’s webpage : <http://www.infres.enst.fr/~lobstein/bibLOCDOmetID.html>