RAIRO-Oper. Res. 58 (2024) 1633–1651
https://doi.org/10.1051/ro/2024046

RAIRO Operations Research
www.rairo-ro.org

DENSITY OF IDENTIFYING CODES OF HEXAGONAL GRIDS WITH FINITE NUMBER OF ROWS

Rudini M. Sampaio[1,*], Gabriel A.G. Sobral[2] and Yoshiko Wakabayashi[2]

Abstract. In a graph G, a set $C \subseteq V(G)$ is an identifying code if, for all vertices v in G, the sets $N[v] \cap C$ are all nonempty and pairwise distinct, where $N[v]$ denotes the closed neighbourhood of v. We focus on the minimum density of identifying codes of infinite hexagonal grids H_k with k rows, denoted by $d^*(H_k)$, and present optimal solutions for $k \leq 5$. Using the discharging method, we also prove a lower bound in terms of maximum degree for the minimum-density identifying codes of well-behaved infinite graphs. We prove that $d^*(H_2) = 9/20$, $d^*(H_3) = 6/13 \approx 0.4615$, $d^*(H_4) = 7/16 = 0.4375$ and $d^*(H_5) = 11/25 = 0.44$. We also prove that H_2 has a unique periodic identifying code with minimum density.

Mathematics Subject Classification. 94B65, 68R10, 90C27, 05C69.

Received March 2, 2023. Accepted February 15, 2024.

1. Introduction

The concept of *identifying code* (*idcode*, for short), was introduced in 1998 by Karpovsky *et al.* [27] to identify a faulty processor in a multiprocessor system. The vertices of an idcode correspond to special processors (the monitors) that are able to check themselves and their neighbours to identify a faulty processor.

Problems on idcodes have been studied on finite and infinite graphs, being of great interest both from theoretical as well as practical viewpoint. Particular interest has been dedicated to grids as many processor networks have a grid topology (see [34, 35]). Among these, we mention the square grid $\mathcal{G}_S$, the triangular grid $\mathcal{G}_T$ and the king grid $\mathcal{G}_K$, shown in Figure 1.

One fundamental problem on idcodes is that of finding idcodes of minimum density. The density captures the proportion of vertices in the code with respect to the whole graph. For finite graphs, Cohen *et al.* [7] proved that deciding the existence of an idcode of size at most k in a graph is an NP-complete problem. On infinite graphs, studies on minimum-density idcodes have considered grids with infinite or with a finite number of rows (see [1–6, 9, 10, 12–14, 16–21, 24, 25, 27, 28]). For an updated bibliography covering this topic and related ones, the reader is referred to Jean [22].

We denote by $d^*(G)$ the minimum density of an idcode of a graph G. For the infinite grids mentioned previously, it is known that $d^*(\mathcal{G}_S) = 7/20$ [1], $d^*(\mathcal{G}_T) = 1/4$ [27] and $d^*(\mathcal{G}_K) = 2/9$ [4]. When these grids have

Keywords. Identifying code, hexagonal grid, minimum density.

[1] Depto de Computação, Universidade Federal do Ceará (UFC), Fortaleza, CE, Brazil.

[2] Instituto de Matemática e Estatstica, Universidade de São Paulo, São Paulo, SP, Brazil.

*Corresponding author: rudini@dc.ufc.br

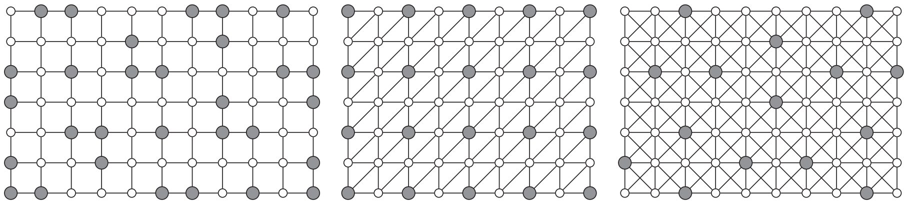

FIGURE 1. Partial representation of infinite square, triangular and king grids, and the corresponding minimum-density idcodes. (A) Square grid $\mathcal{G}_S$. (B) Triangular grid $\mathcal{G}_T$. (C) King grid $\mathcal{G}_K$.

a finite number k of rows, idcodes of minimum density are known for $k \leq 6$, and for larger k only lower and upper bounds have been found.

In this work we focus on infinite graphs, specially the hexagonal grids (see Fig. 2). We denote these grids by $\mathcal{G}_H$ when the number of rows is infinite, and by H_k when the number of rows is a positive integer k. For $\mathcal{G}_H$, new lower and upper bounds have been proved in the last years. Just to mention the more recent ones: in 2009, Cranston and Yu [9] proved a lower bound of $12/29 \approx 0.4138$, and in 2013, Cuckierman and Yu [10] improved the lower bound to $5/12 \approx 0.4166$. In 2014, Stolee [33] presented a computer-assisted framework showing that $d^*(\mathcal{G}_H) \geq 23/55 \approx 0.4181$. As for upper bounds, in 2000, Cohen *et al.* [6] constructed two idcodes of $\mathcal{G}_H$ with density $3/7 \approx 0.4285$. Other idcodes with the same density have also been reported in the literature. Recently, breaking the long-standing bound of 3/7, Salo and Törmä [29] showed that $d^*(\mathcal{G}_H) \leq 53/126 \approx 0.4206$. They found a periodic idcode using a computer-assisted proof that uses automata theory and Karp's minimum mean cycle algorithm. No results on lower or upper bounds have appeared in the literature for $d^*(H_k)$.

We prove that idcodes of well-behaved infinite graphs with maximum degree Δ have density at least $2/(\Delta+2)$. This result and another one on infinite graphs with maximum degree 3 imply that $d^*(H_k) \geq 2/5$ for all $k \geq 2$, and that idcodes of H_k that do not induce trivial components have density at least 3/7. We prove that $d^*(H_2) = 9/20$, and exhibit an idcode with this minimum density, which we show to be unique. We also mention how we proved that $d^*(H_3) = 6/13$, $d^*(H_4) = 7/16$ and $d^*(H_5) = 11/25$, using computer-assisted tools.

In Section 2 we define the concepts used in this paper and establish the notation. We also present a density result on the infinite 3-regular tree, to show that this graph is not so well-behaved as the hexagonal grids, a fact (to be made precise) that has caused an erroneous proof in the literature on a related concept called locating-dominating set (and perhaps on other closely related concepts as well). These preliminary comments help understanding the property (named SG) that we require from the infinite graphs to guarantee that some density proof techniques work. In Sections 3 and 4, we define SG-property and prove results on the discharging method and the mentioned lower bound. In Section 5 we show a minimum-density periodic idcode for H_2, and prove that it is unique. Section 6 contains results on minimum-density idcodes for H_k, $k \in \{3, 4, 5\}$.

A preliminary version of this work (an extended abstract) appeared in [30]. This work contains additional novel results and a simplified and complete proof of Theorem 5.6.

2. DEFINITIONS, NOTATION, AND THE INFINITE 3-REGULAR TREE

The *hexagonal grid*, denoted by $\mathcal{G}_H$, is an infinite graph with vertex set $V = \mathbb{Z} \times \mathbb{Z}$ and edge set $E = \{\{u, v\} : u = (i, j),\ u - v \in \{(\pm 1, 0), (0, (-1)^{i+j+1})\}\}$. See Figure 2. The *hexagonal grid with k rows*, $k \geq 2$, denoted by H_k, is a graph isomorphic to the subgraph of $\mathcal{G}_H$ induced by the vertex set $\mathbb{Z} \times \{1, \ldots, k\}$.

Let G be a connected graph. If v is a vertex of G, and r is a natural number, then $N_r(v)$ denotes the set of vertices of v at distance at most r from v, and $N_r[v] = N_r(v) \cup \{v\}$ denotes the *closed neighbourhood* of v. When $r = 1$, we omit the subscript r and simply write $N(v)$ and $N[v]$. Given $C \subseteq V(G)$, let $C[v] = N[v] \cap C$.

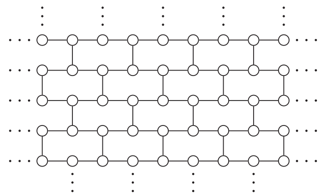

FIGURE 2. Hexagonal grid $\mathcal{G}_H$.

An *idcode* of G is a set $C \subseteq V(G)$ such that $C[v] \neq \emptyset$ for every vertex v of G, and $C[v] \neq C[w]$ for any pair of distinct vertices v, w of G. Thus, if a graph G has two distinct vertices v and w such that $N[v] = N[w]$, then G has no idcode. Such vertices are called *twins*. Clearly, a graph has an idcode if and only if it is twin-free. If C is an idcode, we say that $C[v]$ is the *identifier* of v.

We are interested in minimum-density idcodes of countably infinite connected graphs of bounded degree. For such a graph G, the *density* of a subset $C \subseteq V(G)$, denoted by $d(C,G)$, is defined as follows.

$$d(C,G) = \inf \, \{d_w(C,G) : w \in V(G)\},$$

where

$$d_w(C,G) = \limsup_{r\to\infty} \frac{|C \cap N_r[w]|}{|N_r[w]|}\cdot$$

The *minimum density* of an idcode of a graph G, denoted by $d^*(G)$, is defined as

$$d^*(G) = \inf \, \{d(C,G) : C \text{ is an idcode of } G\}.$$

Notice that we use inf (infimum) in the definition of $d(C,G)$, instead of min (minimum), since the greatest lower bound does not always belong to the set. This definition (with inf) is also given by Jiang [24] to study densities of idcodes of S_k (a topic to be mentioned in Sect. 6). Slater [31] defines density of locating-dominating sets (a notion similar to idcode) with min, but the definition of density $d(C,G)$ makes sense for any set C. In the proof of Lemma 2.1 we show an example of an infinite graph G for which $d_w(C,G) > 0$ for all $w \in V(G)$, but $d(C,G) = 0$.

This definition of subset density given above has not always been used. In some papers, such as [10–13, 23], the density $d(C,G)$ was simply defined as $d_w(C,G)$ where w is an "*arbitrary vertex*". This contains an implicit assumption that $d_w(C,G) = d_v(C,G)$ for any two vertices w, v of G, which is not always true as we show in Lemma 2.1. In most of these papers, this problem in the density definition did not lead to erroneous results, since the graphs considered were well-behaved grids, all of them satisfy an important condition (named SG-property in the next section) which guarantees that $d_w(C,G) = d_v(C,G)$ for any two vertices w, v of G (see Lem. 3.2). However, some papers contain erroneous statements, as we will see in Theorem 2.2.

Lemma 2.1. *There are infinite regular connected graphs G with subsets $C \subset V(G)$ for which there are distinct vertices w, v such that $d_w(C,G) \neq d_v(C,G)$.*

Proof. Let us consider the infinite 3-regular tree T, obtained from two infinite binary trees T_1 and T_2 with roots r_1 and r_2, respectively, by adding the edge r_1r_2. We exhibit two examples of sets $C \subset V(T)$ and vertices w, v of $V(T)$ for which $d_w(C,T) \neq d_v(C,T)$.

As a first example, consider $C = V(T_2)$. Let w be a vertex of T_1 that is a neighbour of r_1. Then $d_w(C,T) = 1/6$. (More generally, if w is at distance d from r_1, we have that $d_w(C,T) = 2^{-d}/3$.) Let $v = r_2$. Then, $d_v(C,T) = 2/3$. (Note that here $d(C,T) = 0$.)

As a second example, let C be the set consisting of all vertices of T_2 together with all vertices of T_1 whose distance to r_1 is even (r_1 included). In this case, C is an idcode of T. Let w (resp. v) be a vertex in T_1 (resp. T_2)

that is at distance d from r_1 (resp. r_2). It is not difficult to check that $d_w(C,T)$ converges to $2/3$ and $d_v(C,T)$ converges to 1 when d tends to ∞. □

Even considering the correct definition of subset density $d(C,G)$, when dealing with infinite graphs one has to be careful with arguments which sound plausible, but may be erroneous. One may consider the technique of covering the vertex set of G with a periodic pattern (subgraph) and assume that the density $d(C,G)$ will be the density of C in the pattern. As an example, consider the infinite 3-regular tree T, used in the proof of Lemma 2.1, which is obtained from two infinite binary trees with roots r_1 and r_2 and the edge r_1r_2. Consider that T is rooted at r_1. Let C be the set of vertices in T whose distance to r_1 is even (r_1 included). Then, the vertices of T can be covered by the pattern formed by an edge (a matching) consisting of a vertex and its leftmost child (being one in C and the other not in C), whose density is $1/2$. Also, by ignoring r_2, the vertices of T can be covered by the pattern formed by a vertex in C and its two children not in C (a cherry), whose density is $1/3$. Finally, by ignoring r_1, the vertices of T can also be covered by the pattern that is a cherry formed by a vertex not in C and its two children in C, whose density is $2/3$. Thus, considering three distinct periodic patterns, this technique gives three different values for the density of $d(C,T)$, indicating that it may not be used in arbitrary graphs. For completeness, we mention that $d(T,C) = 2/3$ (the proof is not so short, but the reader may verify this).

Unfortunately, even more elaborate techniques to calculate density of sets on infinite graphs have led to some erroneous results in the literature. We will elaborate more on this in what follows, calling attention to a property that an infinite graph should satisfy for some techniques to work (see Lem. 3.2). The next theorem shows that one of the first results on the density of locating-dominating sets of regular graphs is not correct. We say that a set $C \subseteq V(G)$ is a *locating-dominating set* (lds) of G if $C[v] \neq \emptyset$, for every $v \notin C$, and $C[v] \neq C[w]$, for any two distinct vertices $v, w \notin C$. Notice that every identifying code is also a locating-dominating set (the difference is that a locating-dominating set C only cares about the vertices outside C). In 2002, Slater [31] stated that "the density of any locating-dominating set of a countably infinite d-regular graph is at least $2/(d+3)$". We present an lds of the infinite 3-regular tree whose density is at most $5/16 = 0.3125$ (a value smaller than $2/(3+3)$), which is a counterexample to the result stated by Slater.

Theorem 2.2. *The minimum density of a locating-dominating set of the infinite 3-regular tree is at most* $5/16 = 0.3125$.

Proof. Let T be the infinite 3-regular tree with root R, and let layer L_i be the set of vertices of T at a distance i from the root R. Thus, $V(T) = \bigcup_{i\geq 0} L_i$, $L_0 = \{R\}$, and $|L_i| = 3 \cdot 2^{i-1}$, for $i \geq 1$. Thus, for $i \geq 5$, $|L_i|$ is a multiple of 16, and is composed of 3 groups with 2^{i-1} vertices.

To construct a set $C \subset V(T)$ which we shall prove to be an lds of T, we label first the vertices of T, and then we define which vertices belong to C. The labelling procedure is the following.

(a) We assign label 1 to all vertices in $L_0 \cup L_1 \cup \ldots \cup L_4$.
(b) We label the vertices of L_5 as follows. We consider that L_5 is composed of 3 consecutive groups of 16 vertices (each of these groups consist of the leaves of the subtree of height 4 rooted at one of the children of root R). We label identically these groups of 16 vertices, according to the following pattern:

```
-------------------------------------------------------------------
1   2   3   5     1   2   3   5     2   3   5   5     3   4   5   5
-------------------------------------------------------------------
```

(c) Once the vertices in L_i, $i \geq 5$, have been labelled, we label the vertices in L_{i+1}. For that, we define for each vertex with label j (in L_i) which are the labels k, l of its children (in L_{i+1}), writing $j \longrightarrow \{k,l\}$. We let $1 \longrightarrow \{3,4\}$, $2 \longrightarrow \{3,3\}$, $3 \longrightarrow \{1,5\}$, $4 \longrightarrow \{5,5\}$ and $5 \longrightarrow \{2,5\}$. Representing this in a tree-like structure, we have:

```
  1         2         3         4         5
 / \       / \       / \       / \       / \
3   4     3   3     1   5     5   5     2   5
```

Now that $V(T)$ is labelled, let

$$C := \{v \in V(T) : v \text{ has label 1 or 2}\}.$$

Consider a group, say H, of 16 vertices in L_5, and let x_j be the number of vertices in H with label j. Then, $x_1 = 2$, $x_2 = 3$, $x_3 = 4$, $x_4 = 1$, $x_5 = 6$; or in a condensed form, $x(H) = (2, 3, 4, 1, 6)$.

Now, let $\text{chld}(H)$ be the group (in L_6) formed by the children of the vertices in H. Let now x'_j be the number of vertices with label j in $\text{chld}(H)$. Then, $x'_1 = x_3 = 4 = 2x_1$, $x'_2 = x_5 = 6 = 2x_2$, $x'_3 = x_1 + 2x_2 = 8 = 2x_3$, $x'_4 = x_1 = 2 = 2x_4$, and $x'_5 = x_3 + 2x_4 + x_5 = 12 = 2x_5$. That is, $x'_j = 2x_j$ for $j \in \{1, 2, \ldots, 5\}$, and therefore, $x(\text{chld}(H)) = 2x(H)$. Since, at each layer L_i, $i \geq 5$, there are 3 groups with 2^{i-1} vertices, and each such group G (by the labelling rule) gives rise to a (children) group with $x(\text{chld}(G)) = 2x(G)$, in each new layer the proportion of vertices with labels 1 or 2 (those in C) is exactly the proportion that holds in layer L_5. We have $|C \cap L_5| = 15$ and $|L_5| = 48$. Thus, $|C \cap L_5|/|L_5| = 15/48 = 5/16$. Since $|L_{i+1}| = 2|L_i|$ and $|C \cap L_{i+1}| = 2|C \cap L_i|$, for each layer L_i the ratio $|C \cap L_i|/|L_i| = 5/16$ holds for all $i \geq 5$. Only for the initial layers L_i, $0 \leq i \leq 4$, we have $|C \cap L_i|/|L_i| = 1$. Thus, the density $d_R(C, T)$ is precisely

$$d_R(C, T) = \limsup_{r\to\infty} \frac{|C \cap N_r[R]|}{|N_r[R]|} = \limsup_{h\to\infty} \frac{|C \cap T_h(R)|}{|T_h(R)|} = 5/16,$$

where $T_h(R)$ is the subtree of T with height h rooted at R. Since $d(C, T) = \inf\{d_w(C, T) : w \in V(T)\}$, we conclude that $d(C, T) \leq 5/16 = 0.3125$.

It remains to prove that C is an lds of T. For that, it suffices to check that the vertices with labels 3, 4, 5 have distinct neighbourhood in C. The reader may check that a vertex with label 3 is identified by its parent and one child (with label 1); a vertex with label 4 is identified solely by its parent (which has label 1); and a vertex with label 5, if it belongs to L_5, then is identified by its parent and one child (with label 2), and if it belongs to layer L_i, $i \geq 6$, then it is identified solely by one child (the one with label 2). This concludes our proof that C is an lds of T with $d(C, T) \leq 5/16 = 0.3125$. ☐

We understand that the erroneous proof of Theorem 2 stated in [31] happened because the infinite graph under consideration does not satisfy a property that would allow the application of the method that was used. The author used the *share method*, which is a simple application of the *discharging method* (to be discussed in the next section) to obtain a lower bound proof for the density of a set, say C.

Roughly speaking, the share method works as follows: each vertex of C starts with charge $q > 0$ and each vertex outside C starts with charge 0. For any vertex $c \in C$ and $u \in N[c]$, the vertex c sends charge $1/|C[u]|$ to u (this includes the case in which $u = c$). At the end of this procedure, all vertices outside C will have charge exactly 1 and every vertex $c \in C$ will have charge $q + 1 - sh(c)$, where $sh(c) = \sum_{u\in N[c]} 1/|C[u]|$ is the total charge sent by c. The idea is that, if $sh(c) \leq q$ for every $c \in C$, all vertices in G will have charge at least 1. Then, if G is finite,

$$1 \cdot |V(G)| \leq \sum_{c\in C} sh(c) \leq q \cdot |C|, \quad \text{and hence} \quad d(C, G) = \frac{|C|}{|V(G)|} \geq \frac{1}{q}\cdot$$

Now, let G be an infinite connected graph and let v be a vertex of G. To guarantee charge at least 1 at every vertex in $N_{r-1}[v]$, it suffices to consider the vertices in $C \cap N_r[v]$. Thus,

$$1 \cdot |N_{r-1}[v]| \leq \sum_{c\in C\cap N_r[v]} sh(c) \leq q \cdot |C \cap N_r[v]|,$$

which implies that

$$d_v(C,G) = \limsup_{r\to\infty} \frac{|C\cap N_r[v]|}{|N_r[v]|} \geq \frac{1}{q}\cdot \limsup_{r\to\infty} \frac{|N_{r-1}[v]|}{|N_r[v]|}\cdot$$

As we can see, the *share method* of [31] will work if $\limsup_{r\to\infty} |N_{r-1}[v]|/|N_r[v]| = 1$, which is a consequence (Lem. 3.2(a) with $t=-1$) of our SG-property, defined in the next section.

3. The use of discharging method to prove lower bounds for the density of idcodes

The discharging method is a proof technique in combinatorics, first used in graph theory, that has now been used in many different contexts, such as in graph colouring, decomposition, embedding, geometric and structural problems. For a guide on the use of the this method to prove results on colouring and other structural properties of graphs see [8].

To prove results on a graph G, this method involves two phases: *charging* and *discharging*. In the charging phase, we assign charges (a rational number) to certain structures of G using a *charging rule*, which describes the value of the charge and the structures of G which will receive the charge. These structures may be vertices, edges, faces (if G is planar), etc. In the discharging phase, we re-assign the charges using the *discharging rules*, which describe the structures that will send and/or receive charge from other vertices. The discharging must preserve the total charge that was assigned in the charging phase.

Both the charging and discharging rules are designed to guarantee that, after these phases some information on the charges of certain vertices/edges will help us prove some property of the graph. In some applications, the initial charges or the discharging rules may take into consideration the degree of the vertices.

The discharging method has been one of the main tools to prove lower bounds for density of idcodes. Theorem 3.3, proved in this section, tells how this method can be used to obtain density results in infinite graphs, once these graphs satisfy certain properties. Before that, we define *SG-property* and present a general result (Lem. 3.2) that is related to this property and is used in Theorem 3.3 and Lemma 3.4. (Here, the mnemonic SG stands for "slow growth", the concept we want to emphasize.)

Definition 3.1. We say that a graph G satisfies the SG-property if G is connected and has a vertex s such that $\lim_{r\to\infty} \frac{|N_{r+1}[s]|}{|N_r[s]|} = 1$.

Notice that, since $N_r[s] \subseteq N_{r+1}[s]$, then $\lim_{r\to\infty} \frac{|N_{r+1}[s]|}{|N_r[s]|} = 1$ if and only if $\limsup_{r\to\infty} \frac{|N_{r+1}[s]|}{|N_r[s]|} = 1$. Also notice that the integer t in the item (a) of the following lemma may be negative.

Lemma 3.2. *Let G be an infinite connected graph satisfying the SG-property, and let $s \in V(G)$ be such that $\lim_{r\to\infty} \frac{|N_{r+1}[s]|}{|N_r[s]|} = 1$. Then the following hold.*

(a) *For every vertex v and integer t, we have $\lim_{r\to\infty} \frac{|N_{r+t}[v]|}{|N_r[v]|} = 1$.*
(b) *For every vertex v and subset $C \subseteq V(G)$, we have $d_v(C,G) = d_s(C,G)$. Thus the density of C is $d(C,G) = d_w(C,G)$, where w is an arbitrary vertex of G.*

Proof. To simplify notation, let $n_k[w] = |N_k[w]|$ for any positive integer k and vertex w. For the vertex s stated in the lemma, and any integer $t > 0$, we have

$$\lim_{r\to\infty} \frac{n_{r+t}[s]}{n_r[s]} = \lim_{r\to\infty} \left(\frac{n_{r+t}[s]}{n_{r+t-1}[s]} \cdot \frac{n_{r+t-1}[s]}{n_{r+t-2}[s]} \cdots \cdot \frac{n_{r+2}[s]}{n_{r+1}[s]} \cdot \frac{n_{r+1}[s]}{n_r[s]} \right) = 1. \tag{1}$$

It is immediate that $\lim_{r\to\infty} \frac{n_{r+t}[s]}{n_r[s]} = 1$ also holds when t is negative (as long as $r+t \geq 0$). Now, to prove (a), consider a vertex v and let $d := \mathrm{dist}(v,s)$. First, we prove that (for $r \geq d$)

$$N_{r-d}[s] \subseteq N_r[v] \subseteq N_{r+d}[s]. \tag{2}$$

To prove the first inclusion, take a vertex y in $N_{r-d}[s]$. Thus, $\text{dist}(y,s) \le r-d$. Since $\text{dist}(y,v) \le \text{dist}(y,s) + \text{dist}(s,v)$, it follows that $\text{dist}(y,v) \le r$, and therefore, $y \in N_r[v]$. The proof of the second inclusion is analogous: take $y \in N_r[v]$, which means that $\text{dist}(y,v) \le r$. Since $\text{dist}(y,s) \le \text{dist}(y,v) + \text{dist}(v,s)$, we have that $\text{dist}(y,s) \le r+d$, and therefore, $y \in N_{r+d}[s]$. From (2), we have that

$$N_{r+1-d}[s] \subseteq N_{r+1}[v] \subseteq N_{r+1+d}[s]. \tag{3}$$

Combining (3) and (2), we have

$$\frac{n_{r+1-d}[s]}{n_{r+d}[s]} \le \frac{n_{r+1}[v]}{n_r[v]} \le \frac{n_{r+1+d}[s]}{n_{r-d}[s]}. \tag{4}$$

Since (1) holds for every integer t (see the observation in the paragraph following (1)), it follows that the limit of the fraction on the left (resp. right) side of (4) when r tends to ∞ is 1, and therefore,

$$\lim_{r\to\infty} \frac{n_{r+1}[v]}{n_r[v]} = 1. \tag{5}$$

From (5), we may conclude that (1) holds when s is replaced by v, and this completes the proof of statement (a).

Now, let us prove (b). For that, we first note that, from (2) we have that

$$\frac{n_{r-d}[s]}{n_r[s]} \le \frac{n_r[v]}{n_r[s]} \le \frac{n_{r+d}[s]}{n_r[s]}. \tag{6}$$

Since the limit of the fraction on the left (resp. right) when r tends to ∞ is 1, it follows that

$$\lim_{r\to\infty} \frac{n_r[v]}{n_r[s]} = 1. \tag{7}$$

By definition, we have that

$$d_v(C,G) = \limsup_{r\to\infty} \frac{|C \cap N_r[v]|}{n_r[v]}. \tag{8}$$

From (2), we obtain

$$C \cap N_{r-d}[s] \subseteq C \cap N_r[v] \subseteq C \cap N_{r+d}[s].$$

Thus,

$$\limsup_{r\to\infty} \frac{|C \cap N_{r-d}[s]|}{n_r[v]} \le d_v(C,G) \le \limsup_{r\to\infty} \frac{|C \cap N_{r+d}[s]|}{n_r[v]}. \tag{9}$$

The lower (resp. upper) bound of $d_v(C,G)$ given by (9) is precisely $d_s(C,G)$. Indeed, for the lower bound, using (8), (1) and (7), we have

$$\limsup_{r\to\infty} \frac{|C \cap N_{r-d}[s]|}{n_r[v]} = \limsup_{r\to\infty} \left(\frac{|C \cap N_{r-d}[s]|}{n_{r-d}[s]} \cdot \frac{n_{r-d}[s]}{n_r[s]} \cdot \frac{n_r[s]}{n_r[v]} \right) = d_s(C,G).$$

For the upper bound, the proof follows similarly. Thus, $d_v(C,G) = d_s(C,G)$, and hence $d(C,G) = d_w(C,G)$, where w is an arbitrary vertex in G. □

The SG-property is very important for the forthcoming proofs on the minimum density based on the discharging method. Lemma 3.2 guarantees that if a connected graph G has this property, then the density of a vertex set C in G may be calculated by considering $d_v(C,G)$ for an arbitrary vertex v.

It is not difficult to see that the infinite hexagonal grids ($\mathcal{G}_H$ and H_k), as well as the grids mentioned in the introduction (square, triangular, king), and many others have the SG-property. In particular, for the grid $\mathcal{G}_H$, it is known that $n_{r+1}[s] = (3(r+2)(r+1))/2 + 1$ for any vertex s, from which we conclude that it has the SG-property. (For more information on $n_r[s]$, see any reference on the rth centered triangular number.) For the grid H_k, as k is fixed, it is easier to conclude that it has the SG-property. As an example of an infinite graph that does not have this property, consider an infinite complete binary tree. Recall that we have shown (see Lem. 2.1) that the infinite 3-regular tree does not have this property.

Theorem 3.3 (Discharging Method)**.** *Let G be an infinite graph with bounded maximum degree which satisfies the SG-property. Let C be a vertex set in G. Suppose that the discharging method is applied to G in the following way. In the charging phase, charge* 1 *is assigned to each vertex in C and charge* 0 *is assigned to the remaining vertices. In the discharging phase, among other rules, the following one is respected: no vertex sends charge from it to a vertex at a distance greater than d, for a fixed integer d. If, at the end, every vertex v of G has final charge* $\mathrm{chg}(v)$ *such that $q \leq \mathrm{chg}(v) \leq q'$, where q and q' are rational numbers, then $q \leq d(C,G) \leq q'$.*

Proof. Given a set $W \subseteq V(G)$, let $\mathrm{chg}(W) = \sum_{w\in W} \mathrm{chg}(w)$. Let q, q' and d be as in the hypothesis of the lemma, and let s be an arbitrary vertex in G. As in the proof of Lemma 3.2, to simplify notation, we let $n_r[s] = |N_r[s]|$. Note that $q \cdot n_r[s] \leq \mathrm{chg}(N_r[s]) \leq q' \cdot n_r[s]$.

Moreover, notice that $\mathrm{chg}(N_r[s])$ is at most $|C \cap N_r[s]|$ plus the charge received from vertices outside $N_r[s]$, which are contained in $N_{r+d}[s]$. Then, $q \cdot n_r[s] \leq \mathrm{chg}(N_r[s]) \leq |C \cap N_r[s]| + n_{r+d}[s] - n_r[s]$. Therefore,

$$d_s(C,G) = \limsup_{r\to\infty} \frac{|C\cap N_r[s]|}{n_r[s]} \geq q - \limsup_{r\to\infty} \frac{n_{r+d}[s] - n_r[s]}{n_r[s]} = q.$$

The last equality holds because $\lim_{r\to\infty} n_{r+d}[s]/n_r[s] = 1$, by Lemma 3.2(a).

Moreover, for $r > d$, $\mathrm{chg}(N_r[s])$ is at least $|C \cap N_r[s]|$ minus the charge sent to vertices outside $N_r[s]$, which comes from vertices in $N_r[s] \setminus N_{r-d}[s]$. Then, $q' \cdot n_r[s] \geq \mathrm{chg}(N_r[s]) \geq |C \cap N_r[s]| - (n_r[s] - n_{r-d}[s])$. Therefore,

$$d_s(C,G) = \limsup_{r\to\infty} \frac{|C\cap N_r[s]|}{n_r[s]} \leq q' + \limsup_{r\to\infty} \frac{n_r[s] - n_{r-d}[s]}{n_r[s]} = q'.$$

Thus, from Lemma 3.2(b) we conclude that $q \leq d(C,G) \leq q'$. □

The next lemma shows that the method of determining the density of a set from periodic patterns, which we showed that may not work on arbitrary graphs, it works on graphs satisfying the SG-property.

Lemma 3.4. *Let G be an infinite connected graph with bounded maximum degree that satisfies the SG-property. Let ℓ, c, c', d be positive integers, and let C be a subset of $V(G)$. Suppose that $V(G)$ can be partitioned into subsets $V_1, V_2, \ldots$ of size ℓ such that, $c \leq |V_i \cap C| \leq c'$ for each $i \geq 1$, and the distance between any two vertices of V_i is at most d. Then $c/\ell \leq d(C,G) \leq c'/\ell$.*

Proof. We use the discharging method as stated in Lemma 3.3 with $q = c/\ell$ and $q' = c'/\ell$. Recall that every vertex of C starts with charge 1 and the vertices outside C starts with charge 0. In the discharging phase, for every part V_i of $V(G)$, the set of vertices in $C \cap V_i$ can guarantee charge at least $q = c/\ell$ and at most $q' = c'/\ell$ for every vertex of V_i. Since the distance between any two vertices of V_i is at most d, no vertex sends charge to a vertex at a distance greater than d. From Lemma 3.3, we conclude that $c/\ell \leq d(C,G) \leq c'/\ell$. □

In particular, for H_k, the above result indicates that to prove a lower bound for the density of an idcode C, one can show that if H_k can be covered with a periodic pattern H, then H is a pattern (subgraph of H_k containing vertices of C) for which the ratio $|C \cap V(H)|/|V(H)|$ is minimum possible (a result that might not be so easy to prove). This would lead us to the conclusion that this ratio gives a lower bound for $d(C, H_k)$. In Section 4, we prove a lower bound for $d^*(H_2)$ using the discharging method, as stated in Theorem 3.3, and we also give another proof based on this idea of a pattern H with best possible ratio. The latter idea also yields a uniqueness proof of the minimum-density periodic idcode of H_2.

4. Lower bounds for the density of some idcodes of H_k

Karpovsky *et al.* [27] proved that for $d \geq 2$, every finite twin-free d-regular graph G satisfies $d^*(G) \geq 2/(d+2)$. This was done using a double counting argument on the set of possible idcodes. The next theorem shows that the same bound holds for infinite connected graphs with maximum degree bounded by a constant d, if the graph has the SG-property. To prove this result, we use the discharging method, in a similar way that Cranston and Yu [9] proved the lower bound 2/5 for the minimum density $d^*(\mathcal{G}_H)$ of the hexagonal grid.

Theorem 4.1. *Let $\Delta \geq 2$ be a fixed integer and G be a connected infinite twin-free graph with maximum degree Δ. If G has the SG-property, then $d^*(G) \geq 2/(\Delta+2)$. In particular, $d^*(H_k) \geq 2/5$ for every $k \geq 2$.*

Proof. Let C be an idcode of G, and let $q = 2/(\Delta+2)$. We apply the discharging method with charging rules as stated in Lemma 3.3, and with the following discharging rule:

(R) If $v \notin C$ and $|C[v]| = p$, then v receives a charge of q/p from each vertex in $C[v]$.

We note that only neighbouring vertices exchange charges (thus we may apply Lem. 3.3 with $d = 1$). We prove now that $\text{chg}(v) \geq q$ for every vertex v in G. Clearly, if $v \notin C$, then $\text{chg}(v) = q$; so assume that $v \in C$. If v has no neighbours in C, then for all $w \in N(v)$ we have $|C[w]| \geq 2$, otherwise $C[v] = C[w]$. Thus, vertex v sends a charge of at most $q/2$ to each vertex in $N(v)$. As a vertex in G has degree at most Δ, it follows that $\text{chg}(v) \geq 1 - \Delta(q/2) = q$.

Suppose now that v has a neighbour in C. Then for at most one vertex, say w, that is a neighbour of v outside C, we have that $C[w] = \{v\}$; and for all the remaining neighbours x of v outside C, we have that $|C[x]| \geq 2$. Thus v sends a charge of at most q to w and at most $q/2$ to the remaining neighbours x in $N(v) \setminus C$. Since the degree of v is at most Δ, it follows that $\text{chg}(v) \geq 1 - q - (\Delta - 2)(q/2) = q$.

As $\text{chg}(v) \geq q$ for every vertex v in G, by Lemma 3.3 we have that $d(C, G) \geq q$. As this holds for an arbitrary idcode C, it follows that $d^*(G) \geq q = 2/(\Delta+2)$. When G is the hexagonal grid H_k with k rows, the result we have shown implies that $d^*(H_k) \geq 2/5$ for every $k \geq 2$. □

If C is an idcode of a graph G, then a component of $G[C]$, the subgraph induced by C, is called a *cluster* of G (w.r.t. C). If a cluster has precisely (resp. at least) t vertices, then it is called a *t-cluster* (resp. *t^+-cluster*). The unique vertex of a 1-cluster is also called a 1-cluster. Note that $G[C]$ has no 2-clusters, otherwise, the 2 vertices in such a cluster would have the same identifier. The idcodes shown in Figures 1B and 1C induce only 1-clusters.

In what follows, we show that if C is an idcode of a graph G such that $G[C]$ has no 1-clusters, and G satisfies certain conditions, then $d(C, G) \geq 3/7$.

Theorem 4.2. *Let G be a connected infinite twin-free graph with maximum degree* 3*, and with the SG-property. If C is an idcode of G such that $G[C]$ has no* 1*-clusters, then $d(C, G) \geq 3/7$. In particular, $d(C, \mathcal{G}_H) \geq 3/7$ and $d(C, H_k) \geq 3/7$ for every $k \geq 2$.*

Proof. We use the discharging method with charging rules as stated in Lemma 3.3. We take $q = 3/7$, and consider the following discharging rules:

(R1) If $v \notin C$ and $|C[v]| = p$, then v receives a charge of $3/(7p)$ from each vertex in $C[v]$.
(R2) If $c \in C$ and $|N[c] \cap C| \geq 2$, then c sends a charge of $1/14$ to each neighbour in $N(c) \cap C$.

Let us prove now that $\text{chg}(v) \geq 3/7$ for every vertex v. Clearly, $\text{chg}(v) = 3/7$ if $v \notin C$. Consider now a vertex $c \in C$. By hypothesis, we have that c has at least one neighbour in C. If c has exactly one neighbour c' in C, then c' must have another neighbour in C. Since c has at most 2 neighbours outside C, then c sends a charge of at most $3/7$ to one of them, at most $3/14$ to the other, and receives $1/14$ from c'. (Note that, if these two neighbours exist, then one of them must have another neighbour in C, distinct from c). Hence, $\text{chg}(c) \geq 1 - 3/7 - 3/14 + 1/14 = 3/7$. If c has exactly two neighbours in C, then c sends a charge of at most $3/7$ to some neighbour $w \notin C$ and exactly $1/14$ to each one of the two neighbours in C. Thus, $\text{chg}(c) \geq 1 - 3/7 - 2(1/14) = 3/7$. If c has exactly three neighbours in C, then c sends exactly $1/14$ of charge to each of them. Hence, $\text{chg}(c) \geq 1 - 3(1/14) = 11/14 > 3/7$. The results follow from Lemma 3.3. □

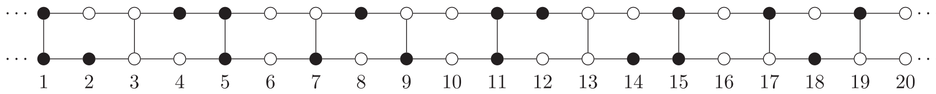

FIGURE 3. An idcode of $T \subset H_2$, which gives an idcode of H_2.

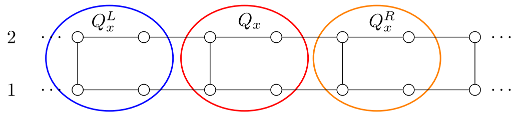

FIGURE 4. Quartets Q_x^L, Q_x and Q_x^R.

5. An identifying code of H_2 with minimum density

In this section we prove that $d^*(H_2) = 9/20$. For that, we prove first the following result.

Lemma 5.1. *The minimum density of an idcode of H_2 is at most $9/20$.*

Proof. Consider the subgraph, say T, indicated in Figure 3, which is a subgraph of H_2 induced by the vertices from columns 1 to 20. Let C the set of 18 black vertices indicated in T.

Note that, the pattern defined by C in the first 10 columns of T is a reflected form of the pattern defined by C in the next 10 columns. We claim that if we concatenate infinite copies of T (side by side), the set of black vertices obtained is an idcode of H_2 (with period 20). We leave to the reader to check this fact (it is enough to check the first 11 columns, and the columns 20 and 21). By Lemma 3.4 we conclude that $d^*(H_2) \leq 9/20$. □

To show that $d^*(H_2) \geq 9/20$, we present two different proofs, which are closely related. Both are based on the patterns defined by an idcode C in the graph H_2. To study these patterns, we consider that the graph H_2 is an infinite strip that can be "split" into "sequential" 4-vertex sets, defined formally in what follows.

For an integer x, we say that a vertex of column x of H_2 is *cubic* if it has degree 3 in H_2. We adopt the convention that when x is odd then the vertices in column x are cubic. For an odd integer x, we denote by Q_x the set of vertices $\{(x,1),(x+1,1),(x,2),(x+1,2)\}$, and call it a *quartet*.

Note that $H_2[Q_x]$ is a $\sqsubset$-shaped path in H_2 with 4 vertices, and $V(H_2)$ is the disjoint union of quartets Q_x such that x is an odd integer. Given a quartet Q_x, we also refer to Q_{x-2} (resp. Q_{x+2}), its left (resp. right) quartet, as Q_x^L (resp. Q_x^R), see Figure 4.

For a given idcode C, we say that Q_x is *type* i (resp. *type* i^+) if $|Q_x \cap C| = i$ (resp. $|Q_x \cap C| \geq i$). Type 1 quartets Q_x play an important role in the proofs. If the single vertex in the idcode that belongs to Q_x is cubic (resp. not cubic) in H_2, we say that Q_x is *type 1-cubic* (resp. *type 1-noncubic*). See Figure 6. All references to types assume that an idcode is clear from the context.

The next lemmas tell us, for each quartet Q_x of type i ($1 \leq i \leq 3$), which are the possible (or forbidden) types of its neighbouring quartets Q_x^L and/or Q_x^R. Once we have these results, we can either use the discharging method or an idea based on the average density of patterns defined by consecutive quartets.

We denote by (H_2, C, x) a triple consisting of the grid H_2, an idcode C of H_2, and an odd integer x. In the figures, vertices coloured black belong to C, vertices coloured gray may belong to C.

Lemma 5.2 (Q_x is type 0)**.** *Consider a triple (H_2, C, x). If Q_x is type 0, then Q_x^L is type 4; moreover, Q_x^R is type 3^+ and $C \cap Q_x^R$ contains two cubic vertices.*

Proof. If Q_x is type 0, it is immediate that all vertices in columns $x-1$ and $x+2$ must be in C, since all vertices of Q_x must have a nonempty identifier. As C is an idcode, the vertices of column $x-2$ must belong to C; thus, Q_x^L is type 4. See Figure 5. Since $H_2[C]$ has no 2-clusters, Q_x^R is type 3^+. □

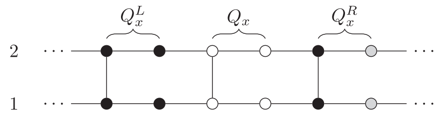

FIGURE 5. Quartet Q_x is type 0 implies quartet Q_x^L is type 4.

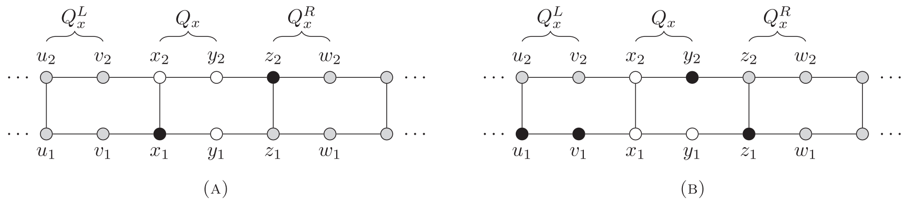

FIGURE 6. Quartet Q_x is type 1. (A) Q_x is type 1-cubic. (B) Q_x is type 1-noncubic.

Lemma 5.3 (Q_x is type 1)**.** *Consider a triple (H_2, C, x). If Q_x is type 1, then the following holds.*

(a) *If Q_x is type 1-cubic, then Q_x^L is type 2^+ and Q_x^R is type 3^+.*
(b) *If Q_x is type 1-noncubic, then Q_x^L is type 3^+ and Q_x^R is type 2^+.*

Proof. For simplicity, rename the vertices of $Q_x^L \cup Q_x \cup Q_x^R$ as shown in Figure 6.

To prove (a), let Q_x be type 1-cubic, and assume without loss of generality that $Q_x \cap C = \{x_1\}$. See Figure 6A.

– If $v_1 \in C$, then $u_1 \in C$, otherwise $\{v_1, x_1\}$ would induce a 2-cluster in H_2, a contradiction. Thus, $Q_x^L \cap C \supseteq \{v_1, u_1\}$ and therefore Q_x^L is type 2^+. If $v_1 \notin C$, then $v_2 \in C$, otherwise $C[x_1] = C[x_2]$, a contradiction. Moreover, $u_1 \in C$, otherwise $C[v_1] = C[x_1]$. Thus, $Q_x^L \cap C \supseteq \{v_2, u_1\}$ and therefore Q_x^L is type 2^+.
– Clearly, $z_2 \in C$, otherwise $C[y_2] = \emptyset$. If $z_1 \in C$, then $|Q_x^R \cap C| \geq 3$, otherwise $\{z_1, z_2\}$ would induce a 2-cluster in H_2. Hence, Q_x^R is type 3^+. If $z_1 \notin C$, then $w_1 \in C$, otherwise $C[z_1] = C[y_2]$. Moreover, $w_2 \in C$, otherwise $C[y_2] = C[z_2]$. Thus, $Q_x^R \cap C = \{z_2, w_1, w_2\}$, and Q_x^R is type 3.

To prove (b), let Q_x be type 1-noncubic, and assume without loss of generality that $Q_x \cap C = \{y_2\}$. See Figure 6B.

– Clearly, $v_1 \in C$, otherwise $C[x_1] = \emptyset$. Moreover, $u_1 \in C$, otherwise $C[x_1] = C[v_1]$. If $u_2 \notin C$, then $v_2 \in C$ (because $C[v_2] \neq \emptyset$). Thus, Q_x^L is type 3^+.
– Clearly, $z_1 \in C$ (because $C[y_1] \neq \emptyset$). If $z_2 \in C$, then Q_x^R is type 2^+. If $z_2 \notin C$, then $w_1 \in C$, otherwise $C[z_1] = C[y_1]$, a contradiction. Hence, Q_x^R is type 2^+.

□

Lemma 5.4 (Q_x is type 2)**.** *Consider a triple (H_2, C, x). If Q_x is type 2, then Q_x^L and Q_x^R may not be both type 1.*

Proof. Suppose, by contradiction, that both Q_x^L and Q_x^R are type 1. By Lemma 5.3, if Q_x^L (resp. Q_x^R) is type 1-cubic (resp. 1-noncubic), then Q_x is type 3^+. Thus, let us suppose now that Q_x^L is type 1-noncubic, $Q_x^L \cap C = \{u\}$; and Q_x^R is type 1-cubic, $Q_x^R \cap C = \{w\}$.

First, assume that u and w are in the same row, say 2. See Figure 7A. Then $(x, 1) \in C$, because $C[(x-1, 1)] \neq \emptyset$. Note that one of the vertices $(x+1, 1)$ or $(x, 2)$ belongs to C, because $C[(x-1, 1)] \neq C[(x, 1)]$. If $(x+1, 1) \in C$, then $(x, 1)$ and $(x+1, 1)$ would induce a 2-cluster in H_2, a contradiction. If $(x, 2) \in C$, then $C[(x+2, 2)] = \{w\} = C[(x+2, 1)]$, a contradiction.

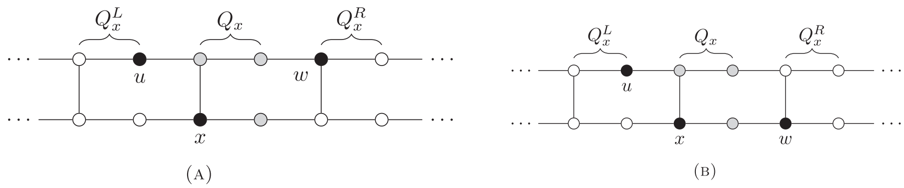

FIGURE 7. Quartet Q_x is type 2. (A) Vertices u and w are in the same row. (B) Vertices u and w are in different rows.

If u and w are in different rows, assume without loss of generality that u is in row 2 and w is in row 1. See Figure 7B. Then $(x, 1) \in C$, because $C[(x-1, 1)] \neq \emptyset$. If both $(x+1, 1)$ and $(x+1, 2)$ do not belong to C, then $C[(x+2, 2)] = C[(x+2, 1)]$, a contradiction. Thus, exactly one of them belongs to C. If $(x+1, 1) \in C$, then $C[(x+1, 2)] = \emptyset$, a contradiction. Hence, $(x+1, 2) \in C$. But in this case, $C[(x-1, 1)] = C[(x, 1)]$, a contradiction. This concludes the proof of the lemma. □

We state now a lemma that will be helpful to simplify the proof of the next theorem.

Lemma 5.5. *The grid H_2 has idcodes of minimum density without type 0 quartets.*

Proof. Let C be an idcode of H_2, and Q_x be a quartet of type 0. By Lemma 5.2, Q_x^L is type 4. It is simple to verify that $C' = C \setminus \{(x-1, 1)\} \cup \{(x, 1)\}$ is an idcode of H_2 such that Q_x is type 1 and Q_x^L is type 3. Thus, $d(C', H_2) = d(C, H_2)$. This means that If C is an idcode of minimum density containing type 0 quartets, then H_2 has also an idcode of the same density without type 0 quartets. □

Remark. The previous lemma does not guarantee anything about the elimination of type 4 quartets. We note that by doing a local change (more involved than the above one) we may also eliminate type 4 quartets and obtain an idcode of equal or possibly smaller density. We do not prove this statement as we do not use it here. Moreover, later we present arguments showing that type 4 quartets do not occur in minimum density idcodes of H_2.

Before going to the next proof, the reader may highlight in Figure 3 the 1-cubic and 1-noncubic quartets, and check the statements of Lemmas 5.3 and 5.4 with respect to the quartets of this figure. This will help the understanding of the discharging rule (resp. the idea based on the average density) used in the next two proofs.

Theorem 5.6. *The minimum density of an idcode of H_2 is precisely $9/20$.*

Proof. We use the discharging method to prove that $d^*(H_2) \geq 9/20$. For that, let C be a minimum identifying code of H_2 that has no quartets of type 0 (cf. Lem. 5.5). In the charging phase, we proceed as stated in Lemma 3.3: we set $\text{chg}(v) = 1$ if $v \in C$, and $\text{chg}(v) = 0$, otherwise. We shall prove that after the discharging phase (to be defined), we have $\text{chg}(Q_x) \geq 9/5$ for each quartet Q_x. If this happens, then the total charge of each Q_x can be distributed among its 4 vertices, and we get $\text{chg}(v) \geq 9/20$ for each vertex v in Q_x. Thus, we say that a quartet Q_x is *satisfied* if $\text{chg}(Q_x) \geq 9/5$, otherwise, it is *unsatisfied*.

After the charging phase, only type 1 quartets are unsatisfied. Apply the following discharging rule.

(R) As long as there are type 1 quartets Q_x that are unsatisfied,
  (a) if Q_x is 1-cubic, then it receives $1/5$ from Q_x^L, and $3/5$ from Q_x^R;
  (b) if Q_x is 1-noncubic, then it receives $3/5$ from Q_x^L, and $1/5$ from Q_x^R.

We prove now that each quartet Q_x is satisfied after the discharging phase.

Case 1. Q_x is type 1.

If Q_x is type 1, then by Lemma 5.3, both Q_x^L and Q_x^R have charge at least 2. Thus, they have sufficient charge to send to Q_x. If Q_x is type 1-cubic, it received 1/5 from Q_x^L and 3/5 from Q_x^R. If Q_x is type 1-noncubic, then it received 3/5 from Q_x^L, and 1/5 from Q_x^R. Hence, in both cases, $\text{chg}(Q_x) = 1 + 1/5 + 3/5 = 9/5$, and therefore Q_x is satisfied.

Case 2. Q_x is type 2.

If Q_x is type 2, then by Lemma 5.4, Q_x^L and Q_x^R are not both type 1. If Q_x^L is type 1, then by Lemma 5.3, it is type 1-noncubic (because Q_x is type 2). Thus, according to rule (R)(b), Q_x^L received 1/5 from Q_x. Since Q_x did not send charge to Q_x^R (because Q_x^R is not type 1) we have that $\text{chg}(Q_x) = 2 - 1/5 = 9/5$. Analogously, if Q_x^R is type 1, then by Lemma 5.3, it is type 1-cubic (because Q_x is type 2). Thus, according to rule (R)(a), Q_x^R received 1/5 from Q_x. Since Q_x did not send charge to Q_x^L (because Q_x^L is not type 1), we have that $\text{chg}(Q_x) = 2 - 1/5 = 9/5$.

Case 3. Q_x is type 3^+.

The only possibility for Q_x to decrease its initial charge is when it has type 1 neighbours. In the worst case, when both Q_x^L and Q_x^R are type 1, Q_x sends at most 3/5 to each of them. Thus, $\text{chg}(Q_x) \geq 3-3/5-3/5 = 9/5$.

Since every quartet Q_x is satisfied, by Lemma 3.3, we have that $d(C, H_2) \geq 9/20$. Using Lemma 5.1, we conclude that $d^*(H_2) = 9/20$. □

From the previous result and the fact that the idcode shown in Lemma 5.1 has density at most 9/20, we conclude the following result.

Corollary 5.7. *The idcode shown in Lemma 5.1 is a periodic idcode of H_2 with minimum density.*

In what follows we present a second proof of Theorem 5.6 which is based on the idea of finding a periodic pattern that covers H_2 and has the minimum possible density. This proof also uses Lemmas 5.2–5.5, and it is based on the fact (mentioned in Sect. 6) that H_2 has a periodic idcode of minimum density. As we will see, the information provided by this proof, combined with further tests, will lead us to conclude that the periodic idcode that we have found is unique.

Proof 2 of Theorem 5.6. Let C be an idcode of minimum density in H_2 that has no quartets of type 0. If C has no quartets of type 1, then all quartets in C are of type 2^+, and in this case, $d(C, H_2) \geq 1/2$, contradicting Lemma 5.1. Thus, C has a quartet of type 1, and by Lemma 5.3 we conclude that C has a quartet of type 3^+.

Now let us consider that H_2 (seen as a concatenation of quartets) can be split into subgraphs corresponding to special sequences of consecutive quartets. We are interested in sequences, which we call $S(3)$-*sequences*, defined as those *starting with a quartet of type* 3^+ *and containing exactly one quartet of type* 3^+. The $S(3)$-sequences whose second quartet is of type 1 (resp. type 2) are called $S(3,1)$-*sequences* (resp. $S(3,2)$-*sequences*). (We remark that not allowing the presence of another quartet of type 3^+ is not a restriction to the size of the periods of the patterns we want to study. We may have different $S(3)$-sequences, and later we allow them to be concatenated, so that periods with many occurrences of quartets of type 3^+ are made possible.)

For an $S(3)$-sequence S, let $I(S) = (i_1, i_2, \ldots)$ be the sequence where each $i_j \in \{1, 2, 3, 4\}$ indicates the type of each of the jth quartet in S. In this proof, $i_j = i^+$ means that $i_j \in \{i, i+1\}$. A simplified notation such as $I(S) = (3^+, 1, 2, 2, 1^+)$ stands for $I(S) \in \{(3, 1, 2, 2, 1), (3, 1, 2, 2, 2), (4, 1, 2, 2, 1), (4, 1, 2, 2, 2)\}$. We denote by $H[S]$ the subgraph of H_2 induced by the quartets in S, and denote by $C(S)$ the restriction of C to $H[S]$. We are interested in $d(C(S), H[S])$, the density of $C(S)$ with respect to $H[S]$.

Note that $I(S)$ may not contain subsequences of the form $(1, 2, 1)$, $(2, 1, 2)$ or $(1, 1)$ because of Lemmas 5.3 and 5.4. If S is an infinite $S(3)$-sequence, then $I(S) = (3^+, 1, 2, 2, \ldots)$ or $I(S) = (3^+, 2, 2, \ldots)$, and therefore $d(C(S), H[S]) \geq 1/2$. If S is a finite $S(3,1)$-sequence, then $I(S)$ contains at most two (non-consecutive) 1's.

Let S_t be a finite $S(3,1)$-sequence of length t, let $I_t = I(S_t)$, and let C_t be the restriction of C to S_t. The possibilities for I_t are: $I_1 = (3^+)$, $I_2 = (3^+, 1)$, $I_3 = (3^+, 1, 2)$, $I_4 = (3^+, 1, 2, 2)$, $I_5 = (3^+, 1, 2, 2, 1^+)$, and $I_t = (3^+, 1, 2, \ldots, 2, 1^+)$ if $t > 5$. Thus $d(C_t, H[S_t]) \geq 1/2$, for $1 \leq t \leq 4$, $d(C_5, H[S_5]) \geq 9/20$ and $d(C_t, H[S_t]) \geq (3 + 1 + 2(t-3) + 1)/4t = (2t-1)/4t > 9/20$ if $t > 5$. Thus the minimum density $9/20$ may possibly occur for $S(3)$-sequences of length 5 with sequence of types $(3, 1, 2, 2, 1)$.

It is easy to see that if S is a finite $S(3,2)$-sequence, then $d(C(S), H[S]) \geq 1/2$ (because $I(S)$ contains at most one 1). This ends the proof that all $S(3)$-sequences of H_2 have density at least $9/20$. Thus, $d(C, H_2) \geq 9/20$ (as H_2 has a minimum-density periodic idcode). Combining this result with Lemma 5.1, we conclude that $d^*(H_2) = 9/20$. □

Remark on the uniqueness of a periodic minimum-density idcode for H_2. By Corollary 5.7, the idcode shown in Figure 3 is a periodic idcode of H_2 with minimum density. An interesting question is whether this idcode is unique, among the periodic ones. The meaning of uniqueness will be clear in what follows.

The second proof of Theorem 5.6 suggests that to construct a periodic minimum-density idcode for H_2 we should look for idcodes that define $S(3,1)$-sequences of length 5 of type $(3, 1, 2, 2, 1)$, and try to concatenate them to see whether they yield a periodic idcode.

As the reader may check, the $S(3,1)$-sequence, say S, corresponding to the 5 initial quartets (first 10 columns) shown in Figure 3 is of type $(3, 1, 2, 2, 1)$. However, the concatenation SS does not define an idcode of H_2 restricted to these sequences. But, as one can see in Figure 3, after S, the next sequence of 5 quartets, say S', which is a reflected form of S is also an $S(3)$-sequence of type $(3, 1, 2, 2, 1)$. As we mentioned before, this is an idcode of H_2 with period 20. This is not the way we obtained this idcode. In fact, this idcode was obtained by an ad hoc method, and we used it as an inspiration to derive the properties (Lems. 5.2–5.5) that we proved. These lemmas, in turn, helped us in the lower bound proof. If a sequence such as S could not be found, one should look for $S(3)$-sequences of lengths $t = 6, 7, \ldots$, as they would be the next candidates (if we did not know an idcode with density $9/20$).

Let us now investigate whether the idcode shown in Figure 3 is the unique periodic idcode of H_2 with density $9/20$. We note that S and S' are the unique $S(3)$-sequences of type $(3, 1, 2, 2, 1)$ (we have verified this by running a program). We also note that the concatenation $S'S'$ does not define an idcode. So, for the moment we may say that the answer to this question is "yes", if we consider minimum idcodes without type 0 quartets (as we proved).

The question now is whether there are minimum-density idcodes containing type 0 quartets. We will not go into details, but we can prove that carrying out analogous arguments as those we used for $S(3,1)$- and $S(3,2)$-sequences, the answer is "no". By Lemma 5.2, a type 0 quartet is preceded by a type 4 quartet, and is succeeded by a type 3^+ quartet. Using this fact, we can show that any $S(3)$-sequence that is of subtype $S(4,0)$ has density greater than $9/20$. Thus, we conclude that the idcode shown in Figure 3 is the unique periodic idcode of H_2 with minimum density. This idcode was also obtained by running a computer program, about which we report in the next section.

We note that, the idea we mentioned after Lemma 3.4 to prove lower bound for the density of idcodes of H_k – based on periodic patterns with minimum density – is basically the idea behind the study we have carried out on the types of sequences of H_2. This study led us to conclude that the periodic pattern H defined by the concatenation SS' is the shortest periodic pattern that has the minimum density $9/20$. Of course, we may say that $S'S$ is also such a shortest periodic pattern, but here we consider that they are equivalent.

6. Minimum-density identifying codes of H_3, H_4 and H_5

In this section we present minimum-density idcodes for H_3, H_4 and H_5 that we found with an algorithm implemented in C++. We describe briefly the algorithm, then exhibit some of these idcodes and the values $d^*(H_3)$, $d^*(H_4)$ and $d^*(H_5)$.

The algorithm that we implemented searches for a periodic idcode for these grids, and uses an idea that was already proposed in 2018 by Jiang [24], to find minimum-density idcodes for square grids S_k with finite number k of rows. We were not aware of his algorithm, although we knew about his results on S_k. Jiang [24] proved that such grids have idcodes with minimum density that are periodic, and described an algorithm to find them. His work presents in detail an algorithm that constructs a weighted directed graph (associated with S_k) in which a minimum mean cycle corresponds to a periodic minimum-density idcode of S_k. Unfortunately, the size of this graph is exponential in k. With his implementation in C, in 2018 Jiang was able to obtain optimum idcodes for S_4 and S_5. We used basically the same idea for H_k. For completeness, we describe briefly the construction of this graph, using the terminology introduced by Jiang.

We do not prove here that H_k has finite periodic idcodes that have minimum density, but this result holds. A proof similar to the one presented by Jiang [24] for S_k can be done for H_k, using the idea based on the concept of bars, which is central here, and is defined in what follows.

For $\ell \geq 1$ and $k \geq 2$, any subgraph of H_k induced by $\{j_1, \ldots, j_\ell\} \times [k]$, where $j_1 \leq j_2 \leq \ldots \leq j_\ell$ are ℓ consecutive columns of H_k, is called an *ℓ-bar* (see Fig. 9). Let R be any ℓ-bar with $\ell \geq 3$ in H_k, and let R' be the $(\ell-2)$-bar consisting of the middle columns of R (obtained by excluding the first and the last columns of R). We say that a subset C of vertices of R is a *barcode* of R if $C[v] \neq \emptyset$ and $C[u] \neq C[v]$ for every distinct $u, v \in R'$. We adopt the convention that the first column of each 4-bar of H_k is indexed by an odd number.

6.1. Construction of the arc-weighted directed graph $G_{k,4,j}$

For $k \geq 2$ and $5 \leq j \leq 8$, let $G_{k,4,j} = (V, A)$ denote the *j-configuration graph of the idcodes of H_k* defined as follows. The vertex set V of this graph consists of barcodes C of any 4-bar of H_k. There is an arc from C to C' if there is a barcode Q of a j-bar B of H_k such that C (resp. C') is the restriction of Q to the first (resp. last) 4 columns of B. In this case, the arc from C to C' gets weight $|Q| - |C|$. Note that, $|V| \leq 2^{4k}$ and $|A| \leq 2^{jk}$. In our implementation, we used $j = 6$ and $j = 8$ (as in this case we have to deal only with 4-bars whose first column is indexed by an odd number).

Jiang [24] considered, for the grid S_k, the graph $G_{k,4,5}$, described above for H_k (for S_k, the 4-bars correspond to subgraphs of S_k). He showed that in this graph, each 4-bar pattern of a periodic idcode for S_k corresponds to a directed cycle and *vice-versa*. We defined $G_{k,4,j}$ for $5 \leq j \leq 8$. It is not difficult to see that an equivalent statement also holds for $j = 6, 7, 8$, and for the grid H_k. Thus, in this case, the density of a minimum periodic idcode in $G_{k,4,j}$ is $w(Z)/pk$, where $w(Z)$ is the weight of a minimum mean cycle Z in the configuration graph $G_{k,4,j}$ and p is the period. (If Z is a cycle, then the *mean weight* of Z is the ratio between the total weight $w(Z)$ of the arcs in Z and the number of arcs in Z.)

In Figure 9 we show a minimum density periodic idcode (with period 8) for H_4 that was found in the 8-configuration graph $G_{4,4,8}$. The two curly braces indicate two consecutive 4-bars (corresponding to two barcodes, say C and C', which are adjacent vertices in this graph). In this case, Q is the barcode of the 8-bar (formed by the indicated 4-bars), and the weight of the arc from C to C' is $|Q| - |C| = 14 - 7 = 7$. This solution corresponds to the weighted directed cycle $Z = (C, C')$ that has length $|Z| = 2$ and weight $w(Z) = 14$ (with mean weight $w(Z)/2 = 14/2 = 7$). In this case, the period is $p = 8$. Thus, the density of this solution is $w(Z)/(8 \cdot 4) = 14/32 = 7/16$. We observe that when $j = 8$ the period is $|Z| \cdot 4$ (but the period is $|Z| \cdot 2$ if $j = 6$, as in this there is an overlap of 2 columns for each two adjacent barcodes).

It is well known that the *minimum mean cycle problem* on a graph with n vertices and m arcs can be solved in $O(nm)$ time by Karp's algorithm [26]. This is the algorithm that Jiang [24] used in his implementation for S_k. For H_k, we use Hartmann–Orlin's algorithm [15], which is an improved version of Karp's algorithm, to find a minimum mean cycle. We implemented a program in C++, using *lemon*[1] library for graphs: it builds the graph $G_{k,4,j}$, finds a minimum mean cycle and outputs an idcode with minimum density for H_k. This implementation can be found in [32].

We run this program to find minimum-density idcodes for H_3, H_4 and H_5. This program constructed $G_{3,4,6}$, $G_{4,4,8}$, $G_{5,4,6}$, and obtained $d^*(H_3) = 6/13$, $d^*(H_4) = 7/16$ and $d^*(H_5) = 11/25$. The corresponding idcodes for these grids are depicted in Figures 8–10. In Table 1, we indicate the size of these configuration graphs and the total running time the program needed to find an optimal solution. The running times for $j = 8$ are included

[1] https://lemon.cs.elte.hu/trac/lemon

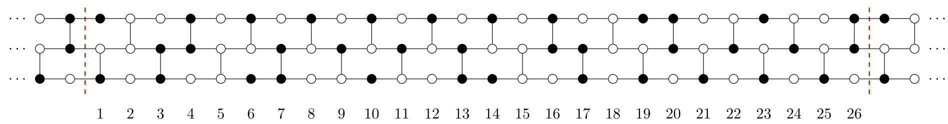

FIGURE 8. A minimum-density idcode of H_3 found in the graph $G_{3,4,6}$ (density $6/13 \approx 0.46153$, period 26).

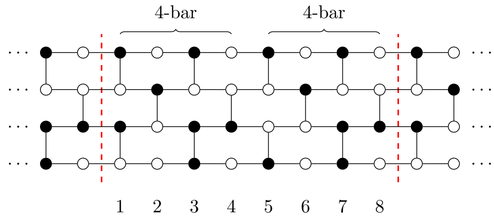

FIGURE 9. A minimum-density idcode of H_4 found in the graph $G_{4,4,8}$ (density $7/16 = 0.4375$, period 8).

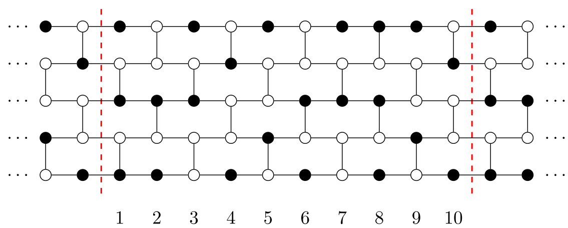

FIGURE 10. A minimum-density idcode of H_5 found in the graph $G_{5,4,6}$ (density $11/25 = 0.44$, period 10).

to show the difference when compared with $j = 6$. The code was compiled with g++ 11.4.0 and option −O3, and executed in a computer with Intel(R) Xeon(R) CPU E7-2870 @ 2.40 GHz processor with 512 GB of RAM.

Theorem 6.1. *For $k = 3, 4, 5$, the idcodes for H_k shown in Figures 8–10 have minimum density. The corresponding densities of these idcodes are $d^*(H_3) = 6/13$, $d^*(H_4) = 7/16$ and $d^*(H_5) = 11/25$.*

As a side remark, we observe that if instead of considering 4-bars, we consider 3-bars (to define the vertices of the graph), and define adjacency of vertices in an analogous way, the corresponding graphs $G_{k,3,5}$ or $G_{k,3,6}$ for S_k or H_k do not have the desired property (as some arcs would indicate a wrong adjacency). We leave to the reader finding examples to verify this statement. But such incorrect adjacencies occur rarely. Since it is much faster to work with 3-bars, one possibility is to work with 3-bars, and check whether the solution found does not have wrong adjacencies, as in this case, an optimum solution may be found more quickly.

We conclude this section mentioning that with our implementation we were not able to find a minimum-density idcode for H_6 using the computer resources available to us.

7. Concluding remarks

We note that for H_3 we have found only the minimum-density idcode shown in Figure 8. But we are not claiming that it is unique. For H_4 and H_5, we have found other minimum-density idcodes with different periods. For H_5 we note that the minimum-density idcode shown in Figure 11 is different from the idcode shown in

TABLE 1. Sizes of the configuration graphs generated by our implementation and total running times.

Configuration graph	#vertices	#edges	Total running time
(A) Data for $j = 6$			
$G_{2,4,6}$ (H_2)	144	1359	8 ms
$G_{3,4,6}$ (H_3)	1896	57 723	253 ms
$G_{4,4,6}$ (H_4)	5870	63 095	8 s
$G_{5,4,6}$ (H_5)	63 751	1 650 188	87 m
(B) Data for $j = 8$			
$G_{2,4,8}$ (H_2)	144	12 894	46 ms
$G_{3,4,8}$ (H_3)	1896	1 784 401	9 s
$G_{4,4,8}$ (H_4)	5870	3 291 346	820 s
$G_{5,4,8}$ (H_5)	63 751	248 161 004	928 m

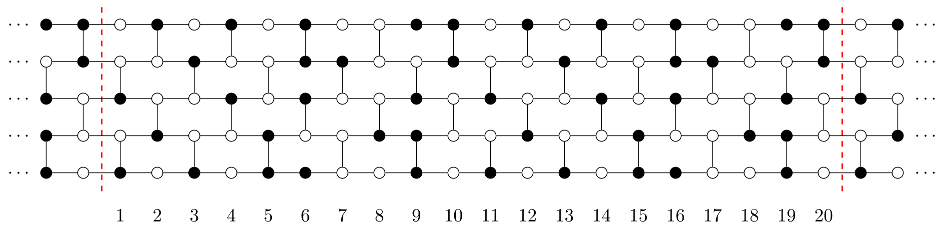

FIGURE 11. A minimum-density idcode of H_5 found in the graph $G_{5,4,8}$ (density $11/25 = 0.44$).

Figure 10, but both have period 10. By considering the graph $G_{5,4,8}$, the corresponding program output the solution of Figure 11 indicating that the period is 20. We noted that the columns from 1 to 10 of this idcode is equal to the columns from 11 to 20. Thus, we may say that the period of this idcode is 10. This does not indicate that the program is incorrect. Clearly, when $j = 8$, the program outputs a solution whose period is always a multiple of 4, while when $j = 6$ the program outputs a solution whose period is a multiple of 2.

With this respect, we note that if H_k has a minimum-density idcode with period p, even when p is odd, an idcode with the same density and possibly different period can be found in the graph $G_{k,4,6}$ and $G_{k,4,8}$. This is true because there is a (smallest) multiple of p which is always a multiple of 2 or of 4, and therefore such a solution will be present in the corresponding graphs. We observe that our program finds one optimal solution (a minimum mean cycle) but not all optimal solutions.

Our implementation may possibly be improved if we can eliminate from the graph $G_{k,4,j}$ some vertices and arcs which we are sure will not occur in an optimal solution. For example, barcodes corresponding to the set of all vertices in a 4-bar, or possibly barcodes whose densities are much larger than some known upper bound for the minimum-density idcode. But to implement such steps safely, some proofs are needed. We also believe that a more substantial improvement is needed to be able to solve for larger k. We are working on this topic and hope that in a forthcoming paper we will be able to present good upper bounds for $d^*(H_k)$, for all $k \geq 6$.

Acknowledgements

The authors would like to thank the referees for the careful reading of the paper, for pointing out some references, and for the valuable suggestions that improved the content and presentation of this paper. The authors are also thankful to Antoine Lobstein for the updated bibliography on identifying codes and related topics that he maintained for so many years, and to Devin Jean [22] for continuing his work. The authors extend their gratitude to the Brazilian research agencies for the financial support:

CAPES (Sobral, Finance Code 01), CNPq (Wakabayashi, Proc. 311892/2021-3), FAPESP (Wakabayashi, Proc. 2015/11937-9), and FUNCAP (Sampaio, Proc. 186-155.01.00/2021).

References

[1] Y. Ben-Haim and S. Litsyn, Exact minimum density of codes identifying vertices in the square grid. *SIAM J. Discrete Math.* **19** (2005) 69–82.
[2] M. Bouznif, F. Havet and M. Preissmann, Minimum-density identifying codes in square grids. *Lecture Notes Comput. Sci.* **11** (2016) 77–88.
[3] I. Charon, I. Honkala, O. Hudry and A. Lobstein, General bounds for identifying codes in some infinite regular graphs. *Electron. J. Comb.* **8** (2001) 21.
[4] I. Charon, O. Hudry and A. Lobstein, Identifying codes with small radius in some infinite regular graphs. *Electron. J. Comb.* **9** (2002) 25.
[5] I. Charon, I. Honkala, O. Hudry and A. Lobstein, The minimum density of an identifying code in the king lattice. *Discrete Math.* **276** (2004) 95–109.
[6] G.D. Cohen, I. Honkala, A. Lobstein and G. Zémor, Bounds for codes identifying vertices in the hexagonal grid. *SIAM J. Discrete Math.* **13** (2000) 492–504.
[7] G.D. Cohen, I. Honkala, A. Lobstein and G. Zémor, On identifying codes, in Codes and Association Schemes (Piscataway, NJ, 1999). *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.* Vol. 56. Amer. Math. Soc., Providence, RI (2001) 97–109.
[8] D.W. Cranston and D.B. West, An introduction to the discharging method via graph coloring. *Discrete Math.* **340** (2017) 766–793.
[9] D.W. Cranston and G. Yu, A new lower bound on the density of vertex identifying codes for the infinite hexagonal grid. *Electron. J. Comb.* **16** (2009) 16.
[10] A. Cukierman and G. Yu, New bounds on the minimum density of an identifying code for the infinite hexagonal grid. *Discrete Appl. Math.* **161** (2013) 2910–2924.
[11] M. Daniel, S. Gravier and J. Moncel, Identifying codes in some subgraphs of the square lattice. *Theor. Comput. Sci.* **319** (2004) 411–421.
[12] R. Dantas, F. Havet and R.M. Sampaio, Identifying codes for infinite triangular grids with a finite number of rows. *Discrete Math.* **340** (2017) 1584–1597.
[13] R. Dantas, F. Havet and R.M. Sampaio, Minimum density of identifying codes of king grids. *Discrete Math.* **341** (2018) 2708–2719.
[14] F. Foucaud, *Combinatorial and algorithmic aspects of identifying codes in graphs.* Ph.D. thesis, Université Sciences et Technologies (2012).
[15] M. Hartmann and J.B. Orlin, Finding minimum cost to time ratio cycles with small integral transit times. *Networks* **23** (1993) 567–574.
[16] I. Honkala and T. Laihonen, Codes for identification in the king lattice. *Graphs Comb.* **19** (2003) 505–516.
[17] I. Honkala and T. Laihonen, On the identification of sets of points in the square lattice. *Discrete Comput. Geom.* **29** (2003) 139–152.
[18] I. Honkala and T. Laihonen, On identification in the triangular grid. *J. Comb. Theory Ser. B* **91** (2004) 67–86.
[19] I. Honkala and T. Laihonen, On identifying codes in the hexagonal mesh. *Inf. Process. Lett.* **89** (2004) 9–14.
[20] I. Honkala and T. Laihonen, On identifying codes in the triangular and square grids. *SIAM J. Comput.* **33** (2004) 304–312.
[21] I. Honkala and A. Lobstein, On the density of identifying codes in the square lattice. *J. Comb. Theory Ser. B* **85** (2002) 297–306.
[22] D. Jean, Watching systems, identifying, locating-dominating and discriminating codes in graphs. Accessed on July 2023 https://dragazo.github.io/bibdom/main.pdf (2023).
[23] D. Jean and S. Seo, Fault-tolerant locating-dominating sets with error-correction, Preprint arXiv:2212.08193v1 (2022).
[24] M. Jiang, Periodicity of identifying codes in strips. *Inf. Process. Lett.* **135** (2018) 77–84.
[25] V. Junnila and T. Laihonen, Optimal lower bound for 2-identifying codes in the hexagonal grid. *Electron. J. Comb.* **19** (2012) 16.
[26] R.M. Karp, A characterization of the minimum cycle mean in a digraph. *Discrete Math.* **23** (1978) 309–311.
[27] M.G. Karpovsky, K. Chakrabarty and L.B. Levitin, On a new class of codes for identifying vertices in graphs. *IEEE Trans. Inf. Theory* **44** (1998) 599–611.
[28] R. Martin and B. Stanton, Lower bounds for identifying codes in some infinite grids. *Electron. J. Comb.* **17** (2010) 16.
[29] V. Salo and I. Törmä, Finding codes on infinite grids automatically. Preprint arXiv.2303.00557 (2023).
[30] R. Sampaio, G. Sobral and Y. Wakabayashi, Minimum density of identifying codes of hexagonal grids with a finite number of rows, in Anais do VII Encontro de Teoria da Computação (Porto Alegre, RS, Brasil). SBC (2022) 145–148.
[31] P.J. Slater, Fault-tolerant locating-dominating sets. *Discrete Math.* **249** (2002) 179–189.
[32] G. Sobral, R. Sampaio and Y. Wakabayashi, Identifying codes. https://gitlab.uspdigital.usp.br/gagsobral/identifying-codes (2023).
[33] D. Stolee, Automated discharging arguments for density problems in grids. Preprint arXiv.1409.5922 (2014).

[34] Z. Xiao and B. Baas, A hexagonal processor and interconnect topology for many-core architecture with dense on-chip networks, in VLSI-SoC: From Algorithms to Circuits and System-on-Chip Design. Springer Berlin Heidelberg (2013) 125–143.

[35] J. Zerovnik and I. Sau, An optimal permutation routing algorithm on full-duplex hexagonal networks. *Discrete Math. Theor. Comput. Sci.* **10** (2008) 1–15.