



HAL
open science

Some Results About a Conjecture on Identifying Codes in Complete Suns

Olivier Hudry, Antoine Lobstein

► **To cite this version:**

Olivier Hudry, Antoine Lobstein. Some Results About a Conjecture on Identifying Codes in Complete Suns. *International Transactions in Operational Research*, 2019, 26 (2), pp.732-746. hal-01613344

HAL Id: hal-01613344

<https://hal.science/hal-01613344>

Submitted on 30 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Some Results About a Conjecture on Identifying Codes in Complete Suns

Olivier Hudry

CNRS - LTCI & Institut Télécom - Télécom ParisTech
46, rue Barrault, 75634 Paris Cedex 13 - France
hudry@telecom-paristech.fr

Antoine Lobstein

CNRS - LTCI & Institut Télécom - Télécom ParisTech
46, rue Barrault, 75634 Paris Cedex 13 - France
lobstein@telecom-paristech.fr

March 17, 2016

Abstract

Consider a graph $G = (V, E)$ and, for every vertex $v \in V$, denote by $B(v)$ the set $\{v\} \cup \{u : uv \in E\}$. A subset $C \subseteq V$ is an identifying code if the sets $B(v) \cap C$, $v \in V$, are all nonempty and distinct. It is a locating-dominating code if the sets $B(v) \cap C$, $v \in V \setminus C$, are all nonempty and distinct.

Let S_n be the graph whose vertex set can be partitioned into two sets U_n and V_n , where $U_n = \{u_1, u_2, \dots, u_n\}$ induces a clique, and $V_n = \{v_{1,2}, v_{2,3}, \dots, v_{n-1,n}, v_{n,1}\}$ induces an independent set, with edges $v_{i,i+1}u_i$ and $v_{i,i+1}u_{i+1}$, $1 \leq i \leq n$; computations are carried modulo n . This graph is called a complete sun. We prove the conjecture, stated by Argiroffo, Bianchi and Wagler in 2014, that the smallest identifying code in S_n has size equal to n . We also characterize and count all the identifying codes with size n in S_n . Finally, we determine the sizes of the smallest locating-dominating codes in S_n .

Key Words: Graph Theory, Identifying Codes, Locating-Dominating Codes

1 Introduction

This paper was first motivated by a conjecture, stated by Argiroffo, Bianchi and Wagler in [1], about the size of optimal identifying codes in a special class of graphs, the complete suns. We prove below that this conjecture is true, but we also investigate these graphs further: we characterize and count all their optimal identifying codes, and we determine the size of their optimal locating-dominating codes.

Consider a graph $G = (V, E)$; the *order* of G is its number of vertices. For every vertex $v \in V$, denote by $B(v)$ the ball of radius one and centre v , i.e., the set $\{v\} \cup \{u : uv \in E\}$. Two vertices $u \in V$, $v \in V$ are said to *dominate* each other if $u \in B(v)$, which is equivalent to $v \in B(u)$. A vertex $z \in V$ is said to *separate* u and v , or u from v , if it dominates u but not v (note that in this case $z = u$ is possible), or the other way round. A set of vertices $Y \subseteq V$ is said to *dominate* a vertex $u \in V$ (respectively, to *separate* two vertices $u \in V$ and $v \in V$) if Y contains at least one vertex which dominates u (respectively, which separates u and v).

A subset $C \subseteq V$ is an *identifying code* if the sets $B(v) \cap C$, $v \in V$, are all nonempty and distinct. In other words, every vertex is dominated by C , and every pair of vertices is separated by C . The vertices in C are called *codewords* and those in $V \setminus C$ *non-codewords*. With a slight abuse of language, when the context is clear, we shall often say that “a vertex is dominated”, or “a pair of vertices is separated”, omitting to add “...by the code” or “...by such codeword”.

When G admits an identifying code, we denote by $\gamma^{ID}(G)$ the minimum cardinality of an identifying code in G . Any identifying code C with $|C| = \gamma^{ID}(G)$ is said to be *optimal*.

Closely related are the locating-dominating codes, which are less demanding: a subset $C \subseteq V$ is a *locating-dominating code* (LD code for short) if the sets $B(v) \cap C$, $v \in V \setminus C$, are all nonempty and distinct. In other words, every non-codeword is dominated by C , and every pair of non-codewords is separated by C . For LD codes, we shall often say that we separate two vertices also when we put one of them in the code. We denote by $\gamma^{LD}(G)$ the minimum cardinality of a LD code in G . Any LD code C with $|C| = \gamma^{LD}(G)$ is said to be *optimal*.

The motivations for identifying and locating-dominating codes may come, for instance, from fault diagnosis in multiprocessor systems. Identifying codes were introduced in the seminal paper [6], which establishes general bounds on γ^{ID} , then studies particular classes of graphs, such as hypercubes, trees, or the hexagonal and triangular infinite grids. Locating-dominating codes were introduced in [9] and further developed in [3], with a focus on complexity and algorithms. Both families of codes now constitute a topic of their own, as shows the bibliography at [7], which contains approximately 340 references.

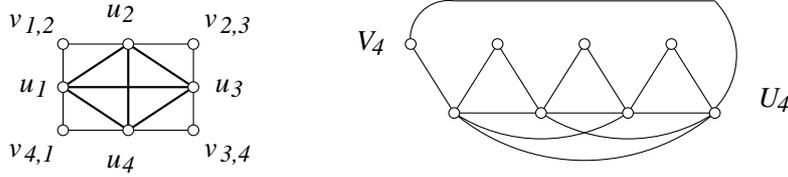


Figure 1: The complete sun S_4 .

A *complete sun* S_n is a graph whose vertex set can be partitioned into two sets U_n and V_n , where $U_n = \{u_1, u_2, \dots, u_n\}$ induces a clique (or complete graph), and $V_n = \{v_{1,2}, v_{2,3}, \dots, v_{n-1,n}, v_{n,1}\}$ induces an independent (or stable) set, with edges $v_{i,i+1}u_i$ and $v_{i,i+1}u_{i+1}$, $1 \leq i \leq n$; computations are carried modulo n . See Figure 1 for $n = 4$.

Complete suns belong to a larger class of graphs, the *split graphs*, whose vertex set can be partitioned into a clique and an independent set. Complexity results for the identifying code problem in split graphs can be found in [4].

In [1], the authors investigate the minimum cardinality of an identifying code for different families of suns, and in particular they conjecture that for $n \geq 3$, $\gamma^{ID}(S_n) = n$ (note that V_n is obviously an identifying code in S_n). In Section 2, we prove that the conjecture is true. We then widen our study of complete suns: in Section 3, we characterize all the optimal identifying codes in S_n , and we count them in Section 4; in Section 5, we prove that $\gamma^{LD}(S_3) = 3$ and $\gamma^{LD}(S_n) = \lfloor \frac{4n+2}{5} \rfloor$ for $n \geq 4$.

NB: In [1, Fig. 2], the graphs (b) and (c) are erroneously inverted.

2 Proof of the Conjecture

In the subsequent figures, black circles will represent codewords, white circles will represent non-codewords, and when necessary there will be squares for the vertices with unknown or immaterial status; we do not represent all the edges in the clique, except in Figure 4.

Since V_n is an identifying code, all we have to prove is that $\gamma^{ID}(S_n) \geq n$. Let C_n be an identifying code in S_n , optimal or not. When we speak of consecutive vertices in U_n or V_n , “consecutive” refers to the subscripts. A vertex $v_{i,i+1}$ such that $v_{i-1,i} \notin C_n$ and $v_{i+1,i+2} \notin C_n$ is called a *single*. A sequence of two consecutive non-codewords $v_{i-1,i}, v_{i,i+1}$ is called a *duet*; two duets with no vertices in common form a *quartet*.

Lemma 1 *Let C_n be an identifying code in S_n . Then:*

- (a) *There is no single in V_n ;*
- (b) *There is at most one duet in V_n ; equivalently, there is no quartet in V_n .*

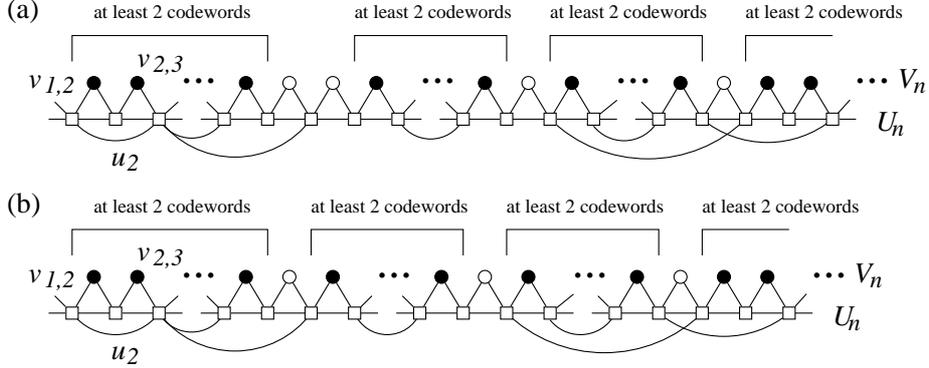


Figure 2: Codewords and non-codewords in V_n .

Proof. (a) If $v_{i,i+1}$ is a single, then u_i and u_{i+1} are not separated by the code, since $v_{i-1,i}$ and $v_{i+1,i+2}$ are the only vertices separating u_i and u_{i+1} .

(b) If $v_{i-1,i}$, $v_{i,i+1}$ and $v_{j-1,j}$, $v_{j,j+1}$ form a quartet, then u_i and u_j are not separated, with $j \in \{1, \dots, n\} \setminus \{i-1, i, i+1\}$. \diamond

Proposition 2 For all integers $n \geq 3$, we have: $\gamma^{ID}(S_n) \geq n$.

Proof. Using Lemma 1, we study the possible sequences of codewords and non-codewords in V_n . In particular, no sequence of three consecutive non-codewords exists in V_n .

There is at least one codeword in V_n , because the vertices in U_n are separated, and without loss of generality (wlog), we assume that $v_{1,2} \in C_n$. Then by Lemma 1(a) $v_{n,1} \in C_n$ or $v_{2,3} \in C_n$: wlog, we take $v_{2,3} \in C_n$. We can then alternate the codewords and the non-codewords in V_n as in Figure 2; in Figure 2(a), we have:

- Sequences of consecutive codewords of various lengths, with at least one sequence, and the lengths at least two by Lemma 1(a).
- One duet; by Lemma 1(b), there is at most one duet.
- Sequences of isolated non-codewords; their number can range between zero and $\lfloor \frac{n-4}{3} \rfloor$.

In Figure 2(b), we have only isolated non-codewords in V_n ; their number can range between zero and $\lfloor \frac{n}{3} \rfloor$ (but as we shall see later on, exactly one [isolated] non-codeword in V_n is impossible).

We denote by q the number of codewords in V_n ; then there are $n - q$ non-codewords in V_n , distributed according to Figure 2(a) or (b). These non-codewords must be dominated and separated by codewords in U_n . It is straightforward to observe that each isolated non-codeword requires (at least) one codeword in U_n (in order to be dominated), that the duet requires (at least) two codewords in U_n (so that its two vertices are dominated and separated), and that these tasks do not overlap. So the $n - q$ non-codewords

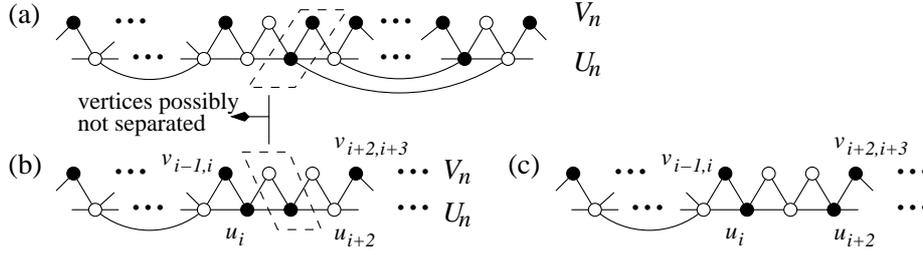


Figure 3: (a) How to treat isolated non-codewords in V_n ; (b) and (c) How to treat a duet.

in V_n require (at least) $n - q$ codewords in U_n , and the total number of codewords is (at least) n . \diamond

Thus, the conjecture is proved:

Corollary 3 For all integers $n \geq 3$, we have: $\gamma^{ID}(S_n) = n$. \diamond

3 Characterization of the Optimal Codes

We construct all the optimal identifying codes in S_n by first choosing the configuration of codewords/non-codewords in V_n ; this configuration is one of the two given in Figure 2. Then we complete the code by choosing the remaining codewords in U_n : for each isolated non-codeword in V_n , we arbitrarily choose one of the two vertices in U_n that dominates it, see Figure 3(a); for each duet in V_n , we arbitrarily choose two vertices among the three vertices in U_n that dominate at least one vertex of the duet, see Figure 3(b)–(c). We claim that, except in some cases that will be described below, the code C_n thus constructed is identifying. Since it contains n elements, it is optimal.

(1) Obviously, all the vertices are dominated by C_n .

(2) The vertices in U_n are separated between themselves:

two consecutive vertices are separated because there is no single in C_n ;
two non-consecutive vertices are separated because there is no quartet in C_n .

(3) The vertices in U_n are separated from the vertices in V_n : indeed, unless

- (a) there is exactly one codeword u_i in U_n , or
- (b) there are exactly two codewords u_i, u_{i+1} in U_n ,

the codewords in U_n perform this separating task. In Case (a) however, u_i and the codeword in V_n that u_i dominates are not separated (cf. Figure 3(a)); this case can be avoided only if we do not take exactly $n - 1$ codewords in V_n . In Case (b), if u_i and u_{i+1} have been put in the code to dominate and separate $v_{i,i+1}$ and $v_{i+1,i+2}$, then u_{i+1} and $v_{i,i+1}$ are not separated, (cf. Figure 3(b)), but by taking rather u_i and u_{i+2} , like in Figure 3(c), we can overcome this difficulty.

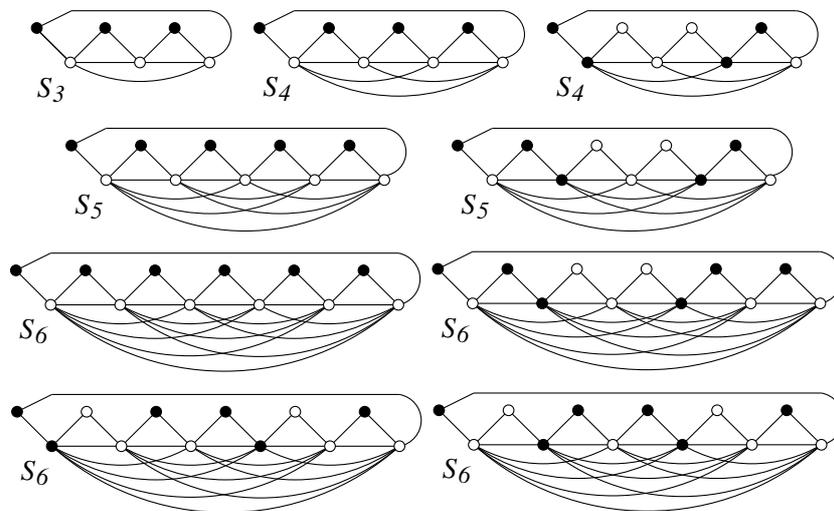


Figure 4: All the nonisomorphic optimal identifying codes.

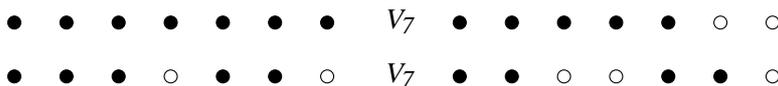


Figure 5: Distributions of codewords in V_7 .

- (4) The vertices in V_n are separated between themselves:
- (a) The codewords in V_n are separated from all the other vertices in V_n by themselves.
 - (b) The non-codewords in V_n are separated between themselves by the codewords in U_n that dominate them.

Summarizing: in order to construct all the optimal identifying codes in S_n , we choose a configuration of codewords and non-codewords in V_n such as given by Figure 2, *with the exception of* the configuration with exactly $n - 1$ codewords in V_n . We complete the code with vertices in U_n , in the way described in the first paragraph of this Section, being careful, in the case when $v_{i,i+1}$ and $v_{i+1,i+2}$ are the *only* non-codewords in V_n , to take u_i and u_{i+2} as codewords, *not* u_i and u_{i+1} *nor* u_{i+1} and u_{i+2} .

Finally, Figure 4 gives all the nonisomorphic optimal identifying codes in S_n , for $n = 3$, $n = 4$, $n = 5$ and $n = 6$, with *decreasing* number of codewords in V_n . Figure 5 gives, for $n = 7$, the four possible configurations for V_n , giving respectively one, one, three and one nonisomorphic optimal identifying codes in S_7 .

An open question is to determine the number of nonisomorphic optimal identifying codes in S_n for $n > 7$, but in the following Section, we manage

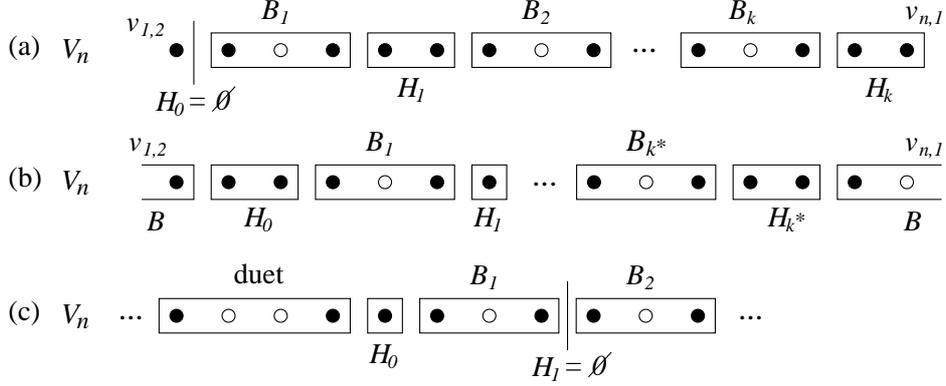


Figure 6: Examples of partitions of V_n : (a) no duet, $v_{1,2}$ not in a brick; (b) no duet, $v_{1,2}$ in the brick B ; (c) one duet.

to count all the optimal identifying codes in S_n , isomorphic or not.

4 Counting the Optimal Codes

4.1 The case when there is no duet in V_n

Consider the set $V_n = \{v_{1,2}, v_{2,3}, \dots, v_{n-1,n}, v_{n,1}\}$ and an optimal identifying code C_n such that there is no duet in V_n ; for $i \in \{1, \dots, n\}$, the set $\{v_{i-1,i}, v_{i,i+1}, v_{i+1,i+2}\}$ is called a *brick* if $v_{i-1,i} \in C_n$, $v_{i,i+1} \notin C_n$, and $v_{i+1,i+2} \in C_n$. Let k be the number of bricks in V_n ; k is also the number of (isolated) non-codewords in V_n , and we have seen previously that $k \in K_n = \{0\} \cup \{2, 3, \dots, \lfloor \frac{n}{3} \rfloor\}$. Two sets of vertices of V_n are said to be *consecutive* if both consist of consecutive vertices in V_n , and if the last vertex of one set and the first vertex of the other set are consecutive.

Assume first that $v_{1,2}$ does not belong to any brick; in particular, $v_{1,2} \in C_n$. Going from $v_{2,3}$ to $v_{n,1}$, we can partition $V_n \setminus \{v_{1,2}\}$ into $2k + 1$ consecutive sets in the following way (see Figure 6(a)):

$$V_n \setminus \{v_{1,2}\} = H_0 \cup B_1 \cup H_1 \cup \dots \cup H_{k-1} \cup B_k \cup H_k, \quad (1)$$

where the sets H_i contain only codewords and can have size zero, and the sets B_i are bricks (of size 3). Let $h_i = |H_i|$. The partitions of type (1) are characterized by the sizes of the sets H_i , and their number is given by the number of solutions of

$$\sum_{i=0}^k h_i = n - 1 - 3k, \quad 0 \leq h_i, \quad (2)$$

which is known to be $\binom{n-2k-1}{k}$, see, e.g., [8]. We have seen that for each isolated non-codeword $v_{j,j+1}$ in V_n , we can choose as codeword in U_n either u_j or u_{j+1} . So, under these conditions (no duet, no brick containing $v_{1,2}$), the number of optimal identifying codes is

$$\alpha_n = \sum_{k \in K_{n-1}} 2^k \binom{n-2k-1}{k} = \sum_{k=0}^{\lfloor \frac{n-1}{3} \rfloor} 2^k \binom{n-2k-1}{k} - 2(n-3). \quad (3)$$

Assume next that $v_{1,2}$ belongs to one brick (hence $k \neq 0$), and let $k^* = k - 1$. There are three possibilities according to the position of $v_{1,2}$ in the brick (leftmost, middle or rightmost). Then we can apply the previous argument, with the following differences nonetheless: $k^* \in \{1, 2, \dots, \lfloor \frac{n}{3} \rfloor - 1\}$, and the partition is done on the $n - 3$ vertices surrounding the brick containing $v_{1,2}$, with k^* bricks and $k^* + 1$ sets H_i ; see Figure 6(b). So (2) becomes

$$\sum_{i=0}^{k^*} h_i = n - 3 - 3k^*, \quad 0 \leq h_i,$$

which admits $\binom{n-2k^*-3}{k^*}$ solutions. Keeping in mind that we have three positions for $v_{1,2}$ inside its brick, and two possibilities in the choice of the codeword in U_n which dominates $v_{1,2}$, we can see that the number of optimal identifying codes in this case is given by

$$\beta_n = 3 \times 2 \times \sum_{k^*=1}^{\lfloor \frac{n-3}{3} \rfloor} 2^{k^*} \binom{n-2k^*-3}{k^*}. \quad (4)$$

So, in the absence of duet, the number of optimal identifying codes in S_n is equal to

$$\delta_n = \alpha_n + \beta_n, \quad (5)$$

where α_n and β_n are given by (3) and (4), respectively.

4.2 The case when there is one duet in V_n

Now we assume that C_n is an optimal identifying code such that there is one duet in V_n (we have seen previously that there can be at most one duet).

First we choose the position of the duet; we have n possibilities. The duet, together with its left and right neighbouring codewords, occupies four vertices. Then we proceed as before, with k bricks, $k \in \{0, 1, 2, \dots, \lfloor \frac{n-4}{3} \rfloor\}$ (no restriction on the number of bricks when there is a duet): we have to choose a partition on the $n - 4$ vertices surrounding the duet and its left and right neighbours (see Figure 6(c)), so we have to consider the number of solutions of

$$\sum_{i=0}^k h_i = n - 4 - 3k, \quad 0 \leq h_i,$$

which is $\binom{n-2k-4}{k}$. We have seen that there is no choice for the two codewords in U_n taking care of the duet, and we can conclude that the number of optimal identifying codes is given by

$$\omega_n = n \times \sum_{k=0}^{\lfloor \frac{n-4}{3} \rfloor} 2^k \binom{n-2k-4}{k} \quad (6)$$

in case of a duet.

4.3 Conclusion

Gathering (3), (4), (5) and (6) gives the total number of optimal identifying codes, which is

$$\begin{aligned} \Phi_n = & \sum_{k=0}^{\lfloor \frac{n-1}{3} \rfloor} 2^k \binom{n-2k-1}{k} + 6 \cdot \sum_{k=1}^{\lfloor \frac{n-3}{3} \rfloor} 2^k \binom{n-2k-3}{k} \\ & + n \cdot \sum_{k=0}^{\lfloor \frac{n-4}{3} \rfloor} 2^k \binom{n-2k-4}{k} - 2(n-3). \end{aligned} \quad (7)$$

The first values given by (7) are $\Phi_3 = 1$, $\Phi_4 = 5$, $\Phi_5 = 6$, $\Phi_6 = 19$, $\Phi_7 = 50$, $\Phi_8 = 89$ and $\Phi_9 = 160$.

In order to obtain an asymptotic estimation of Φ_n , we need to approximate $M_n = \max_{k \in \{0, \dots, \lfloor n/3 \rfloor\}} 2^k \binom{n-2k}{k}$. Setting $N = n - 2k$ and $\varepsilon = k/N$ leads to study $\max_{\varepsilon \in]0, 1[} 2^{N\varepsilon} \binom{N}{N\varepsilon}$, and, using a standard approximation on the binomial coefficient (see, e.g., (2.4.3) in [2]), to study $\max_{\varepsilon \in]0, 1[} N(\varepsilon + H(\varepsilon))$, where $H(x) = -x \log_2(x) - (1-x) \log_2(1-x)$ is the (binary) *entropy function*. Going back to n , we have to search for $\max_{\varepsilon \in]0, 1[} n \frac{\varepsilon + H(\varepsilon)}{1+2\varepsilon}$, that is, to search for $\max_{\varepsilon \in]0, 1[} \frac{\varepsilon + H(\varepsilon)}{1+2\varepsilon}$.

This maximum is reached for $\varepsilon_0 \approx 0.41025$ and $k_0 \approx 0.22535n$, and is approximately 0.76181, which leads finally to $M_n \gtrsim 2^{0.76181n}$. Since $n^* = 2n$ is the order of the complete sun S_n , we have, in terms of order, about $2^{0.38091n^*}$ different optimal identifying codes in the complete sun.

Let us now consider briefly the problem of finding graphs of order n^* with a large number of optimal identifying codes, see [5]. The complete suns are graphs for which the optimal identifying codes have size equal to half their order, and as such could be candidates, since $(\gamma^{ID}(G))^{n^*}$, the number of all subsets of vertices of size $\gamma^{ID}(G)$ in any graph G of order n^* , is maximum when $\gamma^{ID}(G) = n^*/2$. However, the above result, $2^{0.38091n^*}$, is modest when compared to the best known result, namely $2^{0.77003n^*}$ ([5, Th. 11]).

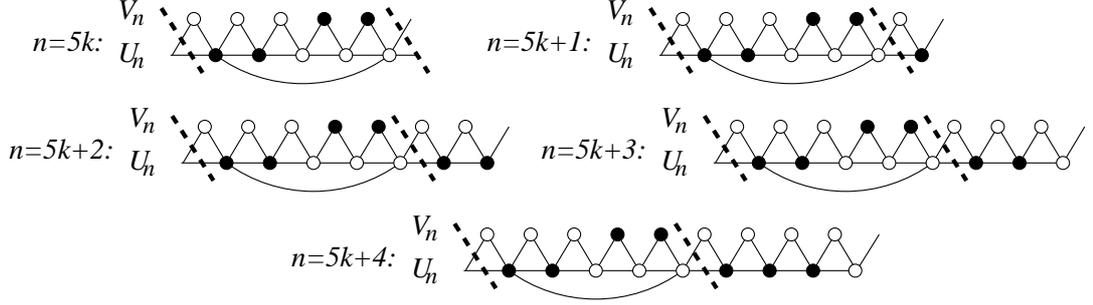


Figure 7: Constructions of LD codes in S_n .

5 Determining $\gamma^{LD}(S_n)$

The first two values of n are easy to study, and we have: $\gamma^{LD}(S_3) = 3$, $\gamma^{LD}(S_4) = 3$. From now on, we assume that $n \geq 5$.

We first give five constructions, according to the congruences of n modulo 5, proving that for $n \geq 5$, we have:

$$\gamma^{LD}(S_n) \leq \left\lfloor \frac{4n+2}{5} \right\rfloor. \quad (8)$$

These constructions are given by Figure 7 and are straightforward to check (remember that only the non-codewords have to be tested).

We next turn to the lower bound which we want to prove:

$$\gamma^{LD}(S_n) \geq \left\lfloor \frac{4n+2}{5} \right\rfloor.$$

The following lemma, where (a) and (b) match Lemma 1 for identifying codes, will prove useful.

Lemma 4 *Let C_n be a LD code in S_n , and $i \in \{1, \dots, n\}$. Then:*

- (a) $C_n \cap \{v_{i-1,i}, u_i, u_{i+1}, v_{i+1,i+2}\} = \emptyset$ is impossible;
- (b) $C_n \cap \{v_{i-1,i}, u_i, v_{i,i+1}\} = \emptyset$ is possible for at most one value of i ;
- (c) $C_n \cap \{u_i, v_{i,i+1}, u_{i+1}\} = \emptyset$ is impossible;
- (d) $C_n \cap \{u_i, v_{i,i+1}, v_{i+1,i+2}, u_{i+2}\}$ is impossible.

Proof. (a) Because u_i and u_{i+1} would not be separated.

(b) Assume on the contrary that there exists $j \in \{1, \dots, n\}$, $j \neq i$, such that $C_n \cap \{v_{j-1,j}, u_j, v_{j,j+1}\} = \emptyset$; then u_i and u_j are not separated.

(c) Because $v_{i,i+1}$ must be dominated.

(d) Because $v_{i,i+1}$ and $v_{i+1,i+2}$ must be separated. \diamond

In the following, we shall say that the sets $\{v_{i-1,i}, u_i, v_{i,i+1}\}$ and $\{u_i, v_{i,i+1}, u_{i+1}\}$ are *triangles*, denoted by ∇ and \triangle , respectively. We shall say that

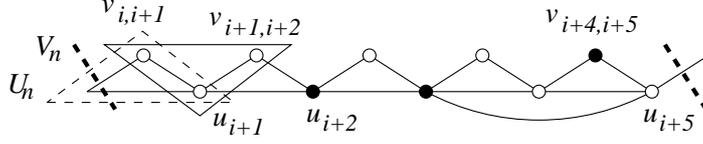


Figure 8: The 10-slice $T_{i,10}$ with three codewords, one in V_n and two in U_n .

a triangle is *empty* if it contains no codeword. Now, with this terminology, Lemma 4(b) states that there is at most one empty triangle ∇ , and Lemma 4(c) states that there is no empty triangle Δ .

Amidst all the LD codes in S_n with size $\gamma^{LD}(S_n)$, we take one, and we denote it by C_n . Assume that $V_n \cap C_n = \emptyset$; then the vertices in U_n cannot be separated, unless all of them but one are codewords. This implies $|C_n| \geq n - 1 \geq \lfloor \frac{4n+2}{5} \rfloor$, for $n \geq 5$. So from now on, we can take for granted that V_n contains at least one codeword.

Step 1. Any set $\{v_{i,i+1}, v_{i+1,i+2}, \dots, v_{i+k-1,i+k}, u_{i+1}, u_{i+2}, \dots, u_{i+k}\}$, $i \in \{1, \dots, n\}$, $k \in \{1, \dots, n\}$, is denoted by $T_{i,2k}$ and is called a $2k$ -*slice*. In this first step, our goal is to show that for all $i \in \{1, \dots, n\}$,

$$|T_{i,10} \cap C_n| \geq 4, \quad (9)$$

with the exception of three cases which will be specifically treated. We are going to study how many codewords can be in $V_n \cap T_{i,10}$, for any i .

- (1) If there are four or five codewords in $V_n \cap T_{i,10}$, then (9) is satisfied.
- (2) Next, assume that there are three codewords in $V_n \cap T_{i,10}$; then, because there are four triangles Δ in $T_{i,10}$, at least one more codeword is necessary in $U_n \cap T_{i,10}$, and (9) is satisfied.
- (3) We now turn to the case when there are two codewords (among five vertices) in $V_n \cap T_{i,10}$; all the $\binom{5}{2} = 10$ configurations require at least two more codewords in $U_n \cap T_{i,10}$, and again (9) is satisfied.
- (4) If $|V_n \cap C_n \cap T_{i,10}| = 1$, then, among the five possibilities, one can see, in particular using Lemma 4, that only one is possible with only two codewords in $U_n \cap T_{i,10}$, see Figure 8, where one can see an empty triangle ∇ .
- (5) If $V_n \cap C_n \cap T_{i,10} = \emptyset$, then, among the $\binom{5}{3} = 10$ configurations for $U_n \cap T_{i,10}$, one can see, again using Lemma 4, that only two of them can possibly lead to LD codes with three codewords in U_n ; they are given in Figure 9. Note that both contain an empty triangle ∇ .

So far, the above five cases show that for all $i \in \{1, \dots, n\}$, the set $C_n \cap T_{i,10}$ has size either (a) at least 4, or (b) equal to 3; in the latter case, only three configurations are possible, given by Figures 8 and 9, and at most

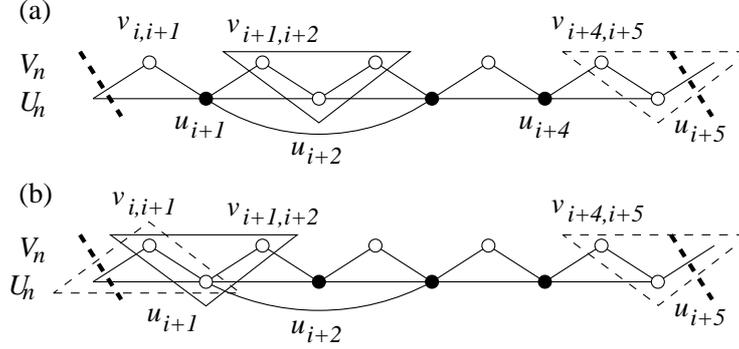


Figure 9: The 10-slice $T_{i,10}$ with three codewords, all of them in U_n .

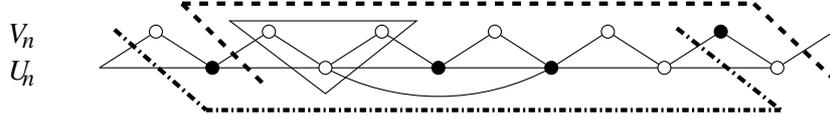


Figure 10: Two 10-slices containing each three codewords.

one of them can appear, at most once, because they all contain an empty triangle ∇ ; more precisely, this is true only if no intersection is allowed: see for instance Figure 10, where the configurations of Figures 8 and 9(a) coexist, but share eight vertices. Since our goal is to split S_n into disjoint 10-slices, we shall go on saying, with a slight abuse of language, that at most one of them appears at most once.

Moreover, in Figure 8, $u_i \in C_n$, by Lemma 4(c), and $v_{i+5,i+6} \in C_n$, in order to avoid the pattern of Lemma 4(a). The same is true in Figure 9(a), by Lemma 4(d)-(b), as well as in Figure 9(b), by Lemma 4(c)-(b). We shall say that in these three cases, the left outside and right outside neighbours of $T_{i,10}$ must be codewords.

We can now proceed to Step 2, in which we show how, according to the congruences of n modulo 5, we can split S_n into 10-slices and one smaller slice, and we prove the lower bound.

Step 2. Given C_n , for $k \in \{1, \dots, n\}$ and $j \in \{0, \dots, 2k\}$, we say that a $2k$ -slice containing exactly (respectively, at least) j codewords is a $(2k, j)$ -slice (respectively, a $(2k, \geq j)$ -slice). With this notation, what we have just proved in Step 1 is that our graph, when arbitrarily split into disjoint 10-slices (plus one r -slice, $r \in \{0, 2, 4, 6, 8\}$), can only contain at most one $(10, 3)$ -slice, and $(10, \geq 4)$ -slices. Moreover, any $(10, 3)$ -slice has its left and right outside neighbours in the code, which in particular helps to show that

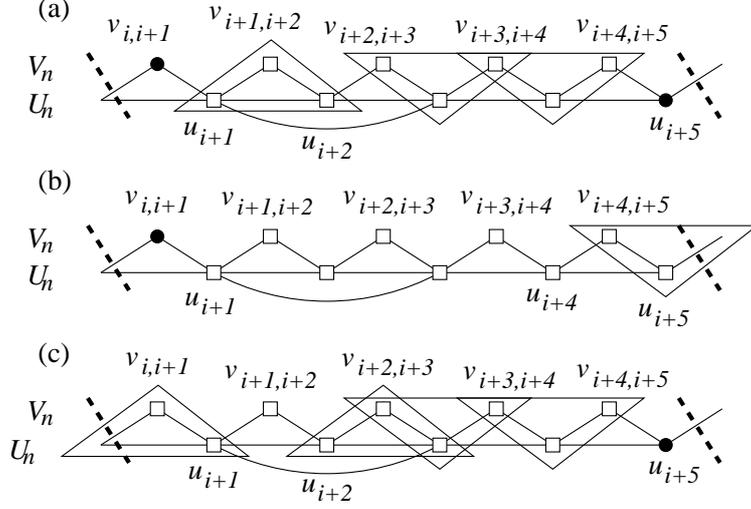


Figure 11: Illustration of Lemma 5, the $(10, 4)$ -slice $T_{i,10}$.

a $(10, 3)$ -slice cannot give by itself a LD code in S_5 , and proves that

$$\gamma^{LD}(S_5) \geq 4. \quad (10)$$

The following lemma gives information on the $(10, 4)$ -slices when there is a $(10, 3)$ -slice in S_n .

Lemma 5 *Assume that we have one $(10, 3)$ -slice $T_{j,10}$ in S_n . Let $T_{i,10}$ be any $(10, 4)$ -slice with no intersection with $T_{j,10}$. Then:*

- (a) *It is impossible to have $v_{i,i+1} \in C_n$ and $u_{i+5} \in C_n$ simultaneously;*
- (b) *If $v_{i,i+1} \in C_n$, then $v_{i+5,i+6} \in C_n$;*
- (c) *If $u_{i+5} \in C_n$, then $u_i \in C_n$.*

Proof. The fact that we have a $(10, 3)$ -slice forbids that an empty triangle ∇ appears in $T_{i,10}$. So in Figure 11, there is at least one codeword in each of the triangles ∇ . Also, by Lemma 4(c), there is at least one codeword in the triangles Δ . We represent only the triangles which are used in the proof.

(a) Assume that $v_{i,i+1} \in C_n$ and $u_{i+5} \in C_n$, see Figure 11(a). If $v_{i+1,i+2} \in C_n$ or $u_{i+1} \in C_n$, then $v_{i+2,i+3}$ and $v_{i+3,i+4}$ are still to be dominated and separated, which is impossible with only one additional codeword. So $v_{i+1,i+2} \notin C_n$, $u_{i+1} \notin C_n$, which implies that $u_{i+2} \in C_n$ (triangle Δ). Then the remaining codeword is at the intersection of the two triangles ∇ , $v_{i+3,i+4}$, but we have the forbidden pattern of Lemma 4(a).

(b) Assume that $v_{i,i+1} \in C_n$ and $v_{i+5,i+6} \notin C_n$, see Figure 11(b). By (a) just above, we have $u_{i+5} \notin C_n$. Now necessarily $v_{i+4,i+5} \in C_n$ (triangle ∇). Then by Lemma 4(a), $u_{i+4} \in C_n$ or $v_{i+3,i+4} \in C_n$, and there remains only one codeword, which cannot dominate and separate $v_{i+1,i+2}$ and $v_{i+2,i+3}$.

(c) Assume that $u_{i+5} \in C_n$ and $u_i \notin C_n$, see Figure 11(c). By (a), we have $v_{i,i+1} \notin C_n$, which implies $u_{i+1} \in C_n$ (leftmost triangle \triangle). But $v_{i,i+1}$ and $v_{i+1,i+2}$ are not separated yet, so $v_{i+1,i+2} \in C_n$ or $u_{i+2} \in C_n$. If it is $v_{i+1,i+2}$, there remain two empty disjoint triangles, one \triangle and one ∇ , with only one more codeword available. If it is u_{i+2} , then the fourth codeword is the vertex at the intersection of the two triangles ∇ , $v_{i+3,i+4}$, but we have the forbidden pattern of Lemma 4(a). \diamond

Remark 6 *The previous lemma is still true if we weaken its condition, by assuming now that there is in S_n an empty triangle ∇ with no intersection with $T_{i,10}$.*

We are now going to consider the different congruences of n modulo 5.

Proposition 7 *For every integer $k \geq 1$, we have $\gamma^{LD}(S_{5k}) = 4k$.*

Proof. The upper bound (8) gives $\gamma^{LD}(S_{5k}) \leq 4k$. By (10), we can assume that $k > 1$. The only way to have fewer than $4k$ codewords in a LD code is to have one $(10, 3)$ -slice and $k - 1$ $(10, 4)$ -slices, in which case we have $4k - 1$ codewords. Assume wlog that the $(10, 3)$ -slice is $T_{1,10}$, and consider the $k - 1$ $(10, 4)$ -slices $T_{6,10}, T_{11,10}, \dots, T_{5k-4,10}$. We have observed just before inequality (10) that $T_{1,10}$ has its right outside neighbour in the code, i.e., $v_{6,7} \in C_n$. Then, by Lemma 5(b), $v_{11,12} \in C_n, \dots, v_{5k-4,5k-3} = v_{n-4,n-3} \in C_n$, which implies, by Lemma 5(a) applied with $i = n - 4$, that $u_1 \notin C_n$, contradicting the fact that $T_{1,10}$ has its left outside neighbour in the code. Note that we could have gone the other way round, using Lemma 5(c): $T_{1,10}$ has its left outside neighbour in the code, so $u_1 \in C_n, u_{5k-4} \in C_n, \dots, u_{11} \in C_n$, which implies that $v_{6,7} \notin C_n$, which contradicts the fact that $T_{1,10}$ has its right outside neighbour in the code.

Anyway, we have proved that the lower and upper bounds coincide. \diamond

Proposition 8 *For every integer $k \geq 1$, we have $\gamma^{LD}(S_{5k+1}) = 4k + 1$.*

Proof. The upper bound (8) gives $\gamma^{LD}(S_{5k+1}) \leq 4k + 1$. Assume first that $k = 1$. Is $|C_6| = 4$ possible?

If there is a $(10, 3)$ -slice in S_6 , then, since its left and right outside neighbours are codewords, we have five codewords. So we can exclude a $(10, 3)$ -slice.

Consider a $(2, \geq 1)$ -slice (there must be one!). Wlog, it is $T_{1,2}$. Then the 10-slice $T_{2,10}$ is a $(10, \geq 4)$ -slice, and we have at least five codewords. So $\gamma^{LD}(S_6) = 5$.

From now on, $k > 1$. Assume first that there is a $(10, 3)$ -slice, say $T_{2,10}$, and consider the 2-slice $T_{1,2}$. Because u_2 , the left outside neighbour of $T_{2,10}$, is a codeword, $T_{1,2}$ is a $(2, \geq 1)$ -slice, and because there can be only one $(10, 3)$ -slice, the $k - 1$ 10-slices $T_{7,10}, \dots, T_{5k-3,10}$ are $(10, \geq 4)$ -slices. This implies, if we want to get at most $4k$ codewords, that $T_{1,2}$ is a $(2, 1)$ -slice,

that the $k - 1$ 10-slices other than $T_{2,10}$ are $(10, 4)$ -slices, and that we have exactly $4k$ codewords. Because the right outside neighbour of $T_{2,10}$, $v_{7,8}$, is a codeword, and using repeatedly Lemma 5(b), we have $v_{7,8} \in C_n$, $v_{12,13} \in C_n$, \dots , $v_{5k+2,5k+3} = v_{1,2} \in C_n$, i.e., $T_{1,2}$ has its two elements, u_2 and $v_{1,2}$, in C_n , a contradiction. So there is no $(10, 3)$ -slice in S_{5k+1} .

As in the case $k = 1$, consider a $(2, \geq 1)$ -slice, say $T_{1,2}$. Then the k 10-slices $T_{2,10}$, $T_{7,10}$, \dots , $T_{5k-3,10}$, which are $(10, \geq 4)$ -slices, lead to at least $4k + 1$ codewords.

So no LD code with $4k$ elements exists in S_{5k+1} . \diamond

Proposition 9 *For every integer $k \geq 1$, we have $\gamma^{LD}(S_{5k+2}) = 4k + 2$.*

Proof. The upper bound (8) gives $\gamma^{LD}(S_{5k+2}) \leq 4k + 2$. Assume first that $k = 1$. Is $|C_7| = 5$ possible? If there is a $(10, 3)$ -slice in S_7 , say $T_{1,10}$, then, since its two outside neighbours, u_1 and $v_{6,7}$, are codewords, we have already five codewords. So $u_7 \notin C_7$ and $v_{7,1} \notin C_7$. Then, taking the configurations of the $(10, 3)$ -slice as given by Figures 8 or 9(b), where neither $v_{i,i+1}$ (here, $v_{1,2}$) nor u_{i+1} (here, u_2) are codewords, we can see that $v_{7,1}$ and $v_{1,2}$ are not separated, i.e., at least six codewords are necessary. If we use the configuration of Figure 9(a), where $v_{i+4,i+5}$ and u_{i+5} (here, $v_{5,6}$ and u_6) are not codewords, we have the pattern forbidden by Lemma 4(a). So no $(10, 3)$ -slice is possible. If we have a $(10, 4)$ -slice, see the general case with k $(10, 4)$ -slices. If we have a $(10, 5)$ -slice, and since the 4-slice must be a $(4, \geq 1)$ -slice (existence of a \triangle), we have also at least six codewords. So $\gamma^{LD}(S_7) = 6$.

From now on, $k > 1$. Can we have $4k + 1$, or fewer, codewords? Assume first that there is a $(10, 3)$ -slice, say $T_{1,10}$. Its outside neighbours, u_1 and $v_{6,7}$, are codewords.

Consider the 4-slice $T_{6,4}$. Because $v_{6,7} \in C_n$, and because there is a triangle \triangle in $T_{6,4}$, $T_{6,4}$ is a $(4, \geq 2)$ -slice. The $k - 1$ 10-slices $T_{8,10}$, \dots , $T_{5k-2,10}$ are $(10, \geq 4)$ -slices, so that necessarily we have: one $(10, 3)$ -slice, one $(4, 2)$ -slice, $k - 1$ $(10, 4)$ -slices, and exactly $4k + 1$ codewords.

Assume first that $T_{1,10}$ is given by Figure 9(a), so that $v_{5,6} \notin C_n$, $u_6 \notin C_n$. The fact that $u_1 \in C_n \cap T_{5k-2,10}$ implies, by Lemma 5(c), that $u_{5k-2} \in C_n$, \dots , $u_8 \in C_n$, so that the two non-codewords in $T_{6,4}$ must be u_7 and $v_{7,8}$. Now the non-codewords $v_{5,6}$, u_6 , u_7 and $v_{7,8}$ form a forbidden pattern.

Assume next that we are in one of the configurations of Figures 8 or 9(b): in particular, $v_{1,2} \notin C_n$ and $u_2 \notin C_n$. Now we consider the 4-slice $T_{n-1,4}$, which contains the codeword u_1 , and a second codeword because it contains a triangle ∇ , and there is already an empty ∇ in $T_{1,10}$. Since $v_{6,7} \in C_n \cap T_{6,10}$, we have, as previously seen several times, $v_{11,12} \in C_n$, \dots , $v_{5k+1,5k+2} = v_{n-1,n} \in C_n$, i.e., $C_n \cap T_{n-1,4} = \{u_1, v_{5k+1,5k+2}\}$. But now the non-codewords u_{5k+2} , $v_{5k+2,1}$, $v_{1,2}$ and u_2 form a forbidden pattern.

Both cases lead to a contradiction, showing that there is no $(10, 3)$ -slice in S_{5k+2} .

Then, to have $4k + 1$ codewords, we can only have k $(10, 4)$ -slices and one $(4, 1)$ -slice. Take one codeword $v_{\ell, \ell+1}$ in V_n (we know that there is one), and consider the 4-slice $T_{\ell, 4}$. Because there is a triangle Δ in $T_{\ell, 4}$, there are at least two codewords in $T_{\ell, 4}$ and we cannot reach $4k + 1$ codewords. \diamond

Proposition 10 *For every integer $k \geq 1$, we have $\gamma^{LD}(S_{5k+3}) = 4k + 2$.*

Proof. The upper bound (8) gives $\gamma^{LD}(S_{5k+3}) \leq 4k + 2$. Observe that any 6-slice is necessarily a $(6, \geq 2)$ -slice: if there is only one codeword, it must be the middle vertex in U_n , because it is at the intersection of the two triangles Δ , but then there is the forbidden pattern of Lemma 4(d).

Assume first that $k = 1$. Is $|C_8| = 5$ possible? If there is a $(10, \geq 4)$ -slice, then there are at least $4 + 2 = 6$ codewords. So we assume that there is a $(10, 3)$ -slice. Its two outside neighbours are codewords, but there is still one empty triangle Δ in the 6-slice (and also one ∇), so at least one additional codeword is required. We can conclude that $\gamma^{LD}(S_8) = 6$.

Assume next that $k > 1$. The only way to have at most $4k + 1$ codewords is to have one $(10, 3)$ -slice, $k - 1$ $(10, 4)$ -slices and one $(6, 2)$ -slice, and we have exactly $4k + 1$ codewords. Wlog, the $(10, 3)$ -slice is $T_{4, 10}$, and we consider the $(6, 2)$ -slice $T_{1, 6}$. Then u_4 , the left outside neighbour of $T_{4, 10}$, is a codeword; the second codeword in $T_{1, 6}$ must belong to the triangle Δ consisting of $u_2, v_{2, 3}, u_3$, but each of the three possibilities leads to a forbidden pattern, either by Lemma 4(a) or (b) (since an empty triangle ∇ already appears in $T_{4, 10}$). This concludes the case $k > 1$. \diamond

Proposition 11 *For every integer $k \geq 1$, we have $\gamma^{LD}(S_{5k+4}) = 4k + 3$.*

Proof. The upper bound (8) gives $\gamma^{LD}(S_{5k+4}) \leq 4k + 3$. If we consider any 8-slice, say $T_{1, 8}$, it is straightforward to see (for instance by checking the nine possibilities given by the two disjoint triangles Δ) that the only possibility with two codewords in $T_{1, 8}$ that avoids any forbidden pattern is $C_n \cap T_{1, 8} = \{u_3, u_4\}$, which however contains an empty triangle ∇ , incompatible with a $(10, 3)$ -slice.

If $k = 1$, to have only six codewords would lead to a $(10, 3)$ -slice with a $(8, 3)$ -slice, or a $(10, 4)$ -slice with a $(8, 2)$ -slice. In the first case, if the $(10, 3)$ -slice is $T_{5, 10}$, its outside neighbours u_5 and $v_{1, 2}$ are codewords, the third codeword in $T_{1, 8}$ is u_3 , because it is at the intersection of two triangles Δ with no codeword yet, but then the forbidden pattern of Lemma 4(d) appears. In the second case, if the $(10, 4)$ -slice is $T_{5, 10}$, and, remembering that the only two codewords in $T_{1, 8}$ are u_3 and u_4 , we can see immediately that $u_1 \in C_9$ (to avoid an empty triangle Δ) and $v_{5, 6} \in C_9$ (to avoid a second empty triangle ∇). This however contradicts, for $n = 9$ and $i = 5$, Lemma 5(a), used together with Remark 6, since there is an empty triangle ∇ outside $T_{5, 10}$. So $\gamma^{LD}(S_9) = 7$.

Now $k > 1$, and we assume that there is a $(10, 3)$ -slice, say $T_{5, 10}$. Then, to obtain $4k + 2$ codewords, we have one $(8, 3)$ -slice and $k - 1$ $(10, 4)$ -slices.

Consider the 8-slice $T_{1,8}$. Then the right outside neighbour of $T_{5,10}$, $v_{10,11} \in T_{10,10}$, is a codeword, and so are, by Lemma 5(b), $v_{15,16}, \dots, v_{5k+5,5k+6} = v_{1,2}$. Also, the left outside neighbour of $T_{5,10}$, u_5 , is a codeword. So we have $\{v_{1,2}, u_5\} \subset C_n \cap T_{1,8}$, and, exactly as above, in the first case for $k = 1$, we are led to a contradiction. Conclusion: there is no $(10, 3)$ -slice in S_{5k+4} .

Now the only way to have at most $4k + 2$ codewords is to have k $(10, 4)$ -slices and one $(8, 2)$ -slice.

Consider the $(8, 2)$ -slice $T_{1,8}$; we have, as in the second case for $k = 1$, $u_1 \in C_n$ and $v_{5,6} \in C_n$. Then again, because we are allowed to use Lemma 5 thanks to the existence of an empty triangle ∇ in the $(8, 2)$ -slice, we have, by Lemma 5(b), $v_{10,11} \in C_n, \dots, v_{5k,5k+1} = v_{n-4,n-3} \in C_n$, but the fact that $v_{n-4,n-3} \in C_n$, together with $u_1 \in C_n$, leads to a contradiction by Lemma 5(a), applied to $i = n - 4$. \diamond

We gather all our results in the following proposition.

Proposition 12 (a) $\gamma^{LD}(S_3) = 3$;
(b) For every integer $n \geq 4$, $\gamma^{LD}(S_n) = \lfloor \frac{4n+2}{5} \rfloor$. \diamond

It would be interesting to do the same as for identifying codes and be able to characterize all the optimal locating-dominating codes in S_n , but from what we have seen above, it seems quite intricate to try to do so, and this remains an open problem.

References

- [1] G. R. ARGIROFFO, S. M. BIANCHI and A. K. WAGLER: On identifying code polyhedra of families of suns, *Proc. of VIII-th ALIO/EURO Workshop on Applied Combinatorial Optimization*, Montevideo, Uruguay, Dec. 2014.
- [2] G. D. COHEN, I. S. HONKALA, S. LITSYN and A. C. LOBSTEIN: *Covering Codes*, Amsterdam: Elsevier, xxii+542 pp., 1997.
- [3] C. J. COLBOURN, P. J. SLATER and L. K. STEWART: Locating dominating sets in series parallel networks, *Congressus Numerantium*, Vol. 56, pp. 135–162, 1987.
- [4] F. FOUCAUD: The complexity of the identifying code problem in restricted graph classes, *Lecture Notes in Computer Science*, No. 8288, pp. 150–163, Springer-Verlag, 2013.
- [5] I. HONKALA, O. HUDRY and A. LOBSTEIN: On the number of optimal identifying codes in a twin-free graph, *Discrete Applied Mathematics*, Vol. 180, pp. 111–119, 2015.

- [6] M. G. KARPOVSKY, K. CHAKRABARTY and L. B. LEVITIN: On a new class of codes for identifying vertices in graphs, *IEEE Transactions on Information Theory*, Vol. IT-44, pp. 599–611, 1998.
- [7] A. LOBSTEIN: Watching systems, identifying, locating-dominating and discriminating codes in graphs, a bibliography,
<http://www.perso.enst.fr/~lobstein/debutBIBidetlocdom.pdf>
- [8] J. MATOUŠEK and J. NEŠETŘIL: *Invitation to Discrete Mathematics*, Oxford University Press, 1998.
- [9] P. J. SLATER: Domination and location in graphs, National University of Singapore, Research Report No. 93, 1983.