Codes in the q-ary Lee Hypercube

IRENE CHARON[1], OLIVIER HUDRY[1], ANTOINE LOBSTEIN[2]
[1]Institut polytechnique de Paris, T´el´ecom Paris, 91123 Palaiseau
[2]Laboratoire Interdisciplinaire des Sciences du Num´erique
(UMR 9015), CNRS, Universit´e Paris-Saclay, 91400 Orsay
FRANCE

Abstract: Let $F_q = \{0, 1, \ldots, q-1\}$ be an alphabet of size q, so that F_q^n is the q-ary hypercube of dimension n. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two elements in F_q^n. The *Lee distance* between x and y is equal to $\sum_{i=1}^{n} \min(|x_i - y_i|, q - |x_i - y_i|)$. Let $C \subseteq F_q^n$; C is called a *code*. Given an integer radius $r \geqslant 1$, we consider three types of codes with respect to the Lee distance: an r-*dominating code* C (also called an r-*covering code*) is such that any element $x \in F_q^n$ is within distance r from at least one codeword $c \in C$ (then c r-*dominates* x); an r-*locating-dominating code* C is (i) r-dominating and (ii) such that any two vertices x, y in $F_q^n \setminus C$ are r-dominated by distinct sets of codewords; an r-*identifying code* C is (i) r-dominating and (ii) such that any two vertices x, y in F_q^n are r-dominated by distinct sets of codewords. We look for minimum such codes. For the above three types of codes, we give tables of upper bounds on their smallest cardinalities, for alphabet size $q \in \{4, 5, 6\}$, dimension n up to 7, and radius r up to 5. These bounds are obtained mainly by using different heuristics (greedy, descent, noising). We conclude with conjectures and open problems.

Key-Words: Dominating codes, Covering codes, Locating-dominating codes, Identifying codes, Hamming distance, Lee distance, q-ary hypercube, Graph theory, Combinatorial optimization, Heuristics.

Received: May 17, 2021. Revised: February 16, 2022. Accepted: March 15, 2022. Published: April 13, 2022.

1 Introduction

We consider the q-ary hypercube of dimension n, endowed with the Lee distance and, using different heuristics, we search for subsets of vertices, with the smallest possible sizes, satisfying different properties that we are now going to define in the more general setting of a finite, undirected, connected graph $G = (V, E)$.

1.1 General notation and definitions

For a given integer radius $r \geqslant 1$ and a given distance d, the *closed r-neighbourhood* of a vertex $v \in V$ is the set $N_r[v] = \{u \in V : 0 \leqslant d(u, v) \leqslant r\}$.

A *code* is simply a subset of V, whose elements are called *codewords*.

Two vertices within distance r from each other are said to r-*dominate* each other. A vertex is r-dominated *by a code* C if it is r-dominated by at least one codeword. Two vertices x and y are r-*separated* by a vertex z if we have $z \in N_r[x]$ and $z \notin N_r[y]$ or if we have $z \in N_r[y]$ and $z \notin N_r[x]$ (note that z can be equal to x or y). Two vertices x and y are r-separated *by a code* C if they are r-separated by at least one codeword.

Definition 1. An r-*dominating code* (r-D code for short) is a code $C \subseteq V$ such that any vertex has at least one codeword at distance at most r:

$$\forall\, v \in V,\ \exists\, c \in C \text{ such that } d(v, c) \leqslant r.$$

Equivalently, every vertex is r-dominated by C, or $\cup_{c \in C} N_r[c] = V$, which implies:

$$\sum_{c \in C} |N_r[c]| \geqslant |V|. \tag{1}$$

Dominating codes constitute a wide topic in graph theory, explored mostly in the case $r = 1$. See, e.g., [16], [17].

Definition 2. An r-*locating-dominating code* (r-LD code for short) is an r-D code $C \subseteq V$ which moreover satisfies:

$$\forall\, v_1 \in V \setminus C,\ \forall\, v_2 \in V \setminus C,\ v_1 \neq v_2 :\ N_r[v_1] \cap C \neq N_r[v_2] \cap C.$$

Equivalently, a code C is an r-LD code if C is r-dominating and r-separates all the pairs of non-codewords. An r-LD code always exists in G (e.g., $C = V$), and we can see that the previous definition states that every non-codeword is characterized (or *located*) by the set of codewords r-dominating it.

Definition 3. An r-*identifying code* (r-Id code for short) is an r-D code $C \subseteq V$ which moreover satisfies:

$$\forall\, v_1 \in V,\ \forall\, v_2 \in V,\ v_1 \neq v_2 :\ N_r[v_1] \cap C \neq N_r[v_2] \cap C.$$

Equivalently, a code C is an r-Id code if C is r-dominating and r-separates all the pairs of vertices. Here, every vertex, whether it is a codeword or not,

is characterized (or *identified*) by the codewords r-dominating it. It is easy to see that an r-Id code exists in G if and only if we have:

$$\forall\, v_1 \in V,\ \forall\, v_2 \in V,\ v_1 \neq v_2 :\ N_r[v_1] \neq N_r[v_2].$$

Locating-dominating codes were introduced in 1983 [30], but for more easily accessible sources, see [26] or [8]. The term "identifying code" is used in [20], which certainly marks the starting point for the blossoming of works on this topic, but the concept is already contained in [27]. For both locating-dominating and identifying codes, see the ongoing bibliography at [23].

We usually search for the smallest possible codes and introduce the notation $\gamma_r(G)$, $\mathrm{LD}_r(G)$, and $\mathrm{Id}_r(G)$ for the smallest sizes of an r-D, an r-LD, and an r-Id code (when there is one), respectively. Any code with this size is called *optimal* with respect to the desired property. Obviously, every r-Id code is r-LD and every r-LD code is an r-D code; hence:

$$\gamma_r(G) \leqslant \mathrm{LD}_r(G) \leqslant \mathrm{Id}_r(G). \qquad (2)$$

1.2 The q-ary Lee hypercube of dimension n

We now restrict our attention to a specific class of graphs.

Let q be an integer, $q \geqslant 2$. Let $F_q = \{0, \ldots, q-1\}$ be an alphabet of size q. All calculations will be modulo q. The *q-ary hypercube of dimension n*, denoted F_q^n, is the Cartesian product of F_q, carried out n times: its elements are the q^n vertices or vectors $v = (v_1, v_2, \ldots, v_n)$, where $v_i \in F_q$ for $1 \leqslant i \leqslant n$. One can define two distances on F_q^n:

- the *Hamming distance* between $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$ in F_q^n is given by:

$$d_H(u, v) = |\{i \in \{1, 2, \ldots, n\} : u_i \neq v_i\}|$$

or equivalently $d_H(u, v) = \sum_{i=1}^{n} \min\{1, |u_i - v_i|\}$;

- the *Lee distance* between $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$ in F_q^n is given by:

$$d_L(u, v) = \sum_{i=1}^{n} \min\{|u_i - v_i|, q - |u_i - v_i|\}.$$

Note that both distances coincide for $q \in \{2, 3\}$. If we draw edges between any two vectors which are at distance one, we obtain, for given q and n, two graphs, according to the distance we consider. We shall use the obvious notation H_q^n and L_q^n; by our previous observation, we have $H_2^n = L_2^n$ and $H_3^n = L_3^n$.

Example. For $r = 2$ in L_4^2:

- $C = \{(0,0); (2,1)\}$ is a 2-D code (for instance, (3, 2) is 2-dominated by (2, 1)) but not a 2-LD code: for instance, (1, 1) and (3, 0) are not 2-separated by C since they are both 2-dominated by (0, 0) and $(2, 1)$, nor a 2-Id code; hence $\gamma_2(L_4^2) \leqslant 2$;

- $C = \{(0,1); (1,2); (2,0); (2,1); (3,1)\}$ is a 2-LD code but not a 2-Id code: for instance (2, 1) and (1, 1) are both 2-dominated by the five codewords and thus are not 2-separated by C; hence $\mathrm{LD}_2(L_4^2) \leqslant 5$;

- $C = \{(0,0); (0,1); (0,2); (1,0); (2,0); (3,3)\}$ is a 2-Id code; hence $\mathrm{Id}_2(L_4^2) \leqslant 6$.

These three codes are optimal, see Table 2.

The study of r-dominating codes in H_q^n (these codes are also called *r-covering codes* or simply *r-coverings* in this case) is of importance for coding and information theory, see [7] and its large bibliography, updated at [22]; the case $q = 3$ can lead to applications for football bets, see, e.g., [15]. Also in H_q^n, results on "good" locating-dominating and identifying codes can be found in, among many others, [18], [3], [12], [10]. Much less is known when considering L_q^n for $q \geqslant 4$, see [1], [20], [21], [9], [28], [11].

From now on, we consider only the Lee distance, that is, the graph L_q^n with $q \geqslant 4$. Due to the regularity of the graph, the size of the r-neighbourhood is the same for all vertices; we denote by $\mathcal{V}_{n,r,q}$ the size of $N_r[v]$ in L_q^n, where v is any vertex.

Consider $v = (v_1, v_2, \ldots, v_n) \in L_q^n$. When q is even, the vertex $(v_1 + \frac{q}{2}, v_2 + \frac{q}{2}, \ldots, v_n + \frac{q}{2})$ is at Lee distance $\delta_e = n \times \frac{q}{2}$ from v, and all other vertices are at distance at most $\delta_e - 1$ from v. When q is odd, the 2^n vertices $(v_1 + \frac{q\pm1}{2}, v_2 + \frac{q\pm1}{2}, \ldots, v_n + \frac{q\pm1}{2})$ are at distance $\delta_o = n \times \frac{q-1}{2}$ from v, and the remaining vertices are at distance at most $\delta_o - 1$.

As a first consequence, if n, q and r are such that $r \geqslant \delta_e$ for even q, or $r \geqslant \delta_o$ for odd q, then all vertices are within distance r from each other in L_q^n. Then obviously:

Remark 1. If n, q and r are such that $r \geqslant n \times \frac{q}{2}$ for even q, or $r \geqslant n \times \frac{q-1}{2}$ for odd q, then $\gamma_r(L_q^n) = 1$, $\mathrm{LD}_r(L_q^n) = q^n - 1$ and L_q^n admits no r-identifying code. Otherwise, it is easy to see that L_q^n admits r-identifying codes.

Assume now that q is even and n, q and r are such that $r = n \times \frac{q}{2} - 1$. Then all vectors are within distance r from each other, except for the pairs $(v_1, v_2, \ldots, v_n)$ and $(v_1 + \frac{q}{2}, v_2 + \frac{q}{2}, \ldots, v_n + \frac{q}{2})$, which are at distance $r + 1$ from one another. In other words, the r-th power (or r-transitive-closure) of L_q^n, denoted $(L_q^n)^r$, is the complete graph deprived of a perfect matching. Then it is not very difficult to see that $\gamma_1((L_q^n)^r) = 2$, $\mathrm{LD}_1((L_q^n)^r) = \frac{q^n}{2}$ and $\mathrm{Id}_1((L_q^n)^r) = q^n - 1$, which in turns implies the following:

Remark 2. If n, q and r are such that q is even and

$r = n \times \frac{q}{2} - 1$, then $\gamma_r(L_q^n) = 2$, $\mathrm{LD}_r(L_q^n) = \frac{q^n}{2}$ and $\mathrm{Id}_r(L_q^n) = q^n - 1$.

Remark 3. Using (1) and (2) and Remarks 1 and 2, one obtains the following bounds:

$$\frac{q^n}{\mathcal{V}_{n,r,q}} \leqslant \gamma_r(L_q^n) \leqslant \mathrm{LD}_r(L_q^n) \leqslant \mathrm{Id}_r(L_q^n) \leqslant q^n - 1. \quad (3)$$

Our goal here is to use different heuristics, already used elsewhere for other optimization problems, in order to construct r-D, r-LD and r-Id codes as small as possible in L_q^n. As a result, for the above three types of codes, Tables 2, 3 and 4 will provide upper bounds on their smallest cardinalities, for alphabet size $q \in \{4, 5, 6\}$, dimension n up to 7, and radius r up to 5. Define the *density* of a code C as the ratio $|C|/q^n$ between the cardinality of C and the number of vertices; Tables 5, 6 and 7 will provide the densities of the computed codes, also for alphabet size $q \in \{4, 5, 6\}$, dimension n up to 7, and radius r up to 5.

2 The methods for obtaining r-D, r-LD or r-Id codes

2.1 The combinatorial optimization problems

The combinatorial optimization problems that we want to solve are the following: given n, r and q, minimize $|C|$ where C is an r-D or an r-LD or an r-Id code in L_q^n. We call these problems (n, r, q)-DP, (n, r, q)-LDP and (n, r, q)-IdP respectively. In the general case of any graph, the decision problems associated with these problems are NP-complete (see [6], [14], [2] or [24] for more references).

In order to solve (n, r, q)-DP, (n, r, q)-LDP and (n, r, q)-IdP, we solve subproblems that we define now. For any subset C of V, let $\alpha(C)$ be the number of vertices which are not r-dominated by C, $\zeta(C)$ be the number of pairs of vertices belonging to $V \setminus C$ which are not r-separated by C, and $\xi(C)$ be the number of pairs of vertices belonging to V which are not r-separated by C. We set: $f_D(C) = \alpha(C)$, $f_{LD}(C) = \alpha(C) + \zeta(C)$ and $f_{Id}(C) = \alpha(C) + \xi(C)$. Observe that:
- C is an r-D code if and only if $f_D(C)$ is equal to 0;
- C is an r-LD code if and only if $f_{LD}(C)$ is equal to 0;
- C is an r-Id code if and only if $f_{Id}(C)$ is equal to 0.

The subproblems associated with (n, r, q)-DP, (n, r, q)-LDP and (n, r, q)-IdP consist in generating codes C and in trying to minimize $f_D(C)$, $f_{LD}(C)$ or $f_{Id}(C)$ respectively. If we find a code C_0 with $f_D(C_0) = 0$, $f_{LD}(C_0) = 0$ or $f_{Id}(C_0) = 0$, then C_0 is an r-D or an r-LD or an r-Id code respectively and we may try again with a smaller code.

In order to minimize f_D, f_{LD} or f_{Id}, we apply a greedy algorithm, some metaheuristics (namely, a descent method and a noising method; for references on metaheuristics, see for instance [29] or [25] among many possibilities) and, for small instances, an exact enumerative method.

2.2 Greedy method

The *greedy method* performed here consists in adding new vertices successively to the code C in construction until C becomes an r-D or an r-LD or an r-Id code according to the studied problem. More precisely, at each step of the method, we choose to add one vertex; the chosen vertex is such that the decrease of f_D for (n, r, q)-DP, of f_{LD} for (n, r, q)-LDP, or of f_{Id} for (n, r, q)-IdP is as large as possible. The aim of such additions of extra vertices is to obtain a value equal to 0 for f_D, f_{LD} or f_{Id}, in which case the current code is an r-D or an r-LD or an r-Id code respectively.

We may start from the empty set for C. Other possibilities consist in starting from a random subset of V or, for (n, r, q)-LDP and (n, r, q)-IdP, from an r-dominating set of L_q^n (for instance, from the solution obtained for (n, r, q)-DP); this allows to save time but may build less good r-LD or r-Id codes (similarly, we may start from an r-LD code for (n, r, q)-IdP).

The greedy method can be improved thanks to the following two remarks.
1. For (n, r, q)-LDP or (n, r, q)-IdP, when a vertex v is added to C, the complexity for evaluating the variations of $\zeta(C)$ or $\xi(C)$ is about $\mathcal{V}_{n,r,q} \times (q^n - \mathcal{V}_{n,r,q})$. Another way to compute this variation consists in scanning a list containing the pairs of vertices which are not r-separated and, for each such pair $\{a, b\}$ of this list, in checking whether v is a r-neighbour of a but not of b or of b but not of a. The complexity is then about the number of pairs of vertices which are not r-separated. So, when this number becomes less than $\mathcal{V}_{n,r,q} \times (q^n - \mathcal{V}_{n,r,q})$, we build an array containing the pairs of vertices which are not r-separated. Of course, we update this array when a vertex is added to the current code. This speeds up the greedy method, and even more and more as the method runs since the number of pairs of vertices which are not r-separated decreases (anyway, we cannot apply this trick from the beginning, because it usually requires a too large memory-space).
2. When we look for a vertex which maximizes the decrease of the objective function, we keep all the vertices yielding this maximum decrease in a list L. Let μ denote this maximum decrease. Then we go through the list L. For each vertex v of L, if adding v to the current code still allows to decrease the

objective function by μ with respect to the current code (this is necessarily the case for the first element of L), then v is added to the current code before considering the next element of L. This avoids the computation of the maximal decrease at each step, which is usually long; thus this considerably speeds up the greedy algorithm, specially at the end of its run.

2.3 Descent method (iterative improvement method)

In order to improve the r-D or r-LD or r-Id code provided by the greedy algorithm (or by any other constructive method), we design a *descent method* (also called *iterative improvement method*, or still *hill climbing method* for a maximization problem).

We start from an r-D or an r-LD or an r-Id code C. We wish to decrease the cardinality of C by removing vertices of C, one by one. For this, we first remove a vertex randomly chosen in C. We update the objective function f_D, f_{LD} or f_{Id} according to the problem. Then we try to restore an r-D or an r-LD or an r-Id code without changing the current number of codewords. If we succeed, we apply the same process (i.e., removing a vertex) to the new r-D or r-LD or r-Id code.

In order to recover an r-D or an r-LD or an r-Id code, we perform *elementary transformations* (or *local transformations*) with respect to the current code C. An elementary transformation consists here in removing a vertex u belonging to C and, simultaneously, in adding a vertex v which does not belong to C. The vertices u and v are randomly chosen. We say that the transformation is *good* if the objective function f_D, f_{LD} or f_{Id} does not increase (the variation of the objective function can be equal to 0); otherwise, we say that it is *bad*.

Observe a main difference with respect to a usual design of a descent method: in order to ensure the convergence of the process, a good transformation is quite often required to involve a strict decrease of the objective function (this variation cannot be equal to 0). It appears from our experiments that allowing transformations which do not change the value taken by the objective function may be preferable. Indeed, when we cannot improve the current configuration with only one transformation, then it is necessary to perform several transformations if we want to escape a local minimum in order to reach a better configuration. The negative counterpart of this is that the process can cycle without improving the current configuration; it is then necessary to upper bound the number of iterations performed by the method.

So, in the descent method, we try a given number of elementary transformations, which is a parameter of the method (of course directly related to the CPU time that we want to spend to find an r-D or an r-LD or an r-Id code). Only the good transformations are actually performed: the objective function can never increase. If the objective function f_D, f_{LD} or f_{Id} according to the problem becomes equal to 0, then the current code is again an r-D or an r-LD or an r-Id code respectively; in this case, we remove a randomly chosen vertex from the new r-D code or r-LD code or r-Id code and we start a new descent. On the contrary, if we perform the total number of trials without obtaining a new r-D or r-LD or r-Id code, we stop the whole process.

2.4 Noising method

This metaheuristic (see [5] for a review of its principles and of applications) can be seen as a generalization of the simulated annealing method (as well as other metaheuristics; see [4]). The version of the noising method applied in our experiments is very close to the usual scheme of simulated annealing, from which we draw the terminology of *temperature*. In particular, a bad transformation may be accepted, unlike in the descent method.

Four parameters are involved: the initial temperature T_0, the number of temperature changes, the number of elementary transformations tried for each value of the temperature and a fixed real number λ with $0 < \lambda < 1$, meant to control the geometric decrease of the temperature T: when T changes, T becomes λT. The values of these parameters depend on the considered problem. For example, to find a 2-D code of size 30 for (5, 2, 4)-DP, we applied the noising method with an initial temperature equal to 50, 100 temperature changes, 10^7 elementary transformations and $\lambda = 0.95$ (the CPU time was equal to 268 014 seconds, i.e. a little more than three days).

Consider an elementary transformation involving a variation Δ of the objective function f_D, f_{LD} or f_{Id} (with $\Delta > 0$ when we deal with a bad elementary transformation). For each tried transformation, we draw a real number p between 0 and 1 with a uniform distribution. The transformation is accepted if we have $\Delta \leqslant pT$ (hence, the main difference here with simulated annealing is that, in simulated annealing, the acceptance criterion would be $\Delta \leqslant -T \ln p$; the results obtained by a classic simulated annealing are about the same as the ones reported below; the noising method applied here avoids the computations of logarithms). When T becomes less than 1, the behaviour of the noising method at the end of its run is qualitatively the same as the one of a descent.

2.5 Removing the unnecessary vertices

At the end of any of the previous methods, we consider each vertex v of the computed r-D or r-LD

or r-Id code according to the problem. If removing v leaves anyway a suitable code, we definitively remove v.

According to our experiments, this simple principle may improve the current configuration quite often, especially when many elementary tranformations were performed with a variation of the objective function equal to 0 (see above). Indeed, these zero-variation transformations then achieve a "mixing" that increases diversity by changing the codewords inside the current configuration, which can be profitable to move away from a local minimum.

2.6 Exact method

For small values of n and q, we also looked for minimum r-D or r-LD or r-Id codes by using an exact method. Such a method is based on an exhaustive enumeration of the codes. We start with a code C of low cardinality k. While C is not an r-D or an r-LD or an r-Id code according to the considered problem, we generate the next (according to the lexicographic order) code of cardinality k. If all the codes of size k have been tried in vain, we increase k and we repeat the same process.

Another way to apply this enumerative method is to start with an r-D or an r-LD or an r-Id code of size k and to decrease k: while we find a suitable code with a given cardinality k, we decrease k; when we do not find any longer a suitable code of cardinality k, we conclude that $k+1$ is the minimum size. This is particularly useful for checking the optimality of a given r-D or r-LD or r-Id code of cardinality k, when k is not too large: we try all the $(k-1)$-sized codes to check that none of them is appropriate. In the tables below, this is indicated by the key "$*b$"; more generally, a star "*" denotes the minimum cardinality of an r-D or an r-LD or an r-Id code.

3 Tables of results

Tables of bounds for identifying codes are given in [11] for $4 \leqslant q \leqslant 6$, $2 \leqslant n \leqslant 7$ and $1 \leqslant r \leqslant 5$. This is why we chose the same values for the parameters q, n, r in our study, even if we considered codes other than identifying, namely dominating and locating-dominating.

The lower bounds $\lceil \frac{q^n}{\mathcal{V}_{n,r,q}} \rceil$ given in Table 1 are weak, specially for locating-dominating codes and identifying codes, but probably also for dominating codes. Indeed, even for small values of r and n, we may observe that the lower bounds and the minimum values are not equal (for instance, for $n=3$, $r=1$, $q=4$: 10 *versus* 12, see Table 2; for $n=2$, $r=2$, $q=5$: 2 *versus* 3, see Table 3; for $n=2$, $r=2$, $q=6$: 3 *versus* 4, see Table 4). For identifying codes, better lower bounds exist for $q=4$ (see [21] for $r=1$ and [11] for $r>1$), and for $q>4$, q even and $r=1$ [20]. They are given in Tables 2, 3 and 4, together with the upper bounds (all constructive).

The results reported in these tables were obtained with different computers and by the application of different methods among the ones depicted above. The aim of this paper is not to study and to compare the efficiency of these methods by putting them in competition, but on the contrary to make them collaborate in order to compute the best possible codes. For these reasons, we do not report the CPU times required to compute the r-D, r-LD or r-Id codes. These CPU times can be very large, up to several weeks: for instance, we needed 699 809 seconds (i.e. a little more than 8 days) to improve by 1 a 4-D code with 119 vertices for $n=q=6$ by the descent method, and then 1 448 007 extra seconds (i.e. about 16.8 days) in order to obtain a code with 117 vertices from the code with 118 vertices.

The r-D, r-LD or r-Id codes that we computed and of which the sizes are reported in Tables 2, 3 and 4 can be found at the following address:

https://perso.telecom-paristech.fr/hudry/LeeHypercubesCodes.html

The codewords available in the files displayed at this address are encoded as integers thanks to the function φ defined as follows. Each vertex u of L_q^n, i.e. a vector $u=(u_1,u_2,\ldots,u_n)$, can be associated with an integer $\varphi(u)$ defined by $\varphi(u)=\sum_{i=1}^{n} u_i q^{i-1}$. For instance, for $q=4$, we get $\varphi(1,2,0,2)=1\times4^0+2\times4^1+0\times4^2+2\times4^3=137$. Obviously, φ is a one-to-one function between the vertices of L_q^n and the integer values between 0 and q^n-1. This representation saves space.

If we succeed in improving the values reported in Tables 2, 3 and 4 later on, we will update these files.

Keys to Tables 2, 3 and 4

Lower bounds are given for r-identifying codes, for $q=4$, and $q=6$ with $r=1$. They are in italics. Also for identifying codes, our new upper bounds are followed by the previously best known upper bounds between brackets. Keys are mainly for identifying codes, relying on [20], [21], and [11]:

a = because $\mathrm{LD}_r(G) \leqslant \mathrm{Id}_r(G)$ (but none of our heuristics could find the LD-code; this happens only once, for $n=7$, $r=1$, $q=6$)

$*b$ = optimal: exhaustive search

$*c$ = optimal: $r = n \times \frac{q}{2} - 1$ for q even (for r-D, r-LD and r-Id-codes, see Remark 2)

$*d$ = optimal: $r \geqslant n \times \frac{q}{2}$, q even, and $r \geqslant n \times \frac{q-1}{2}$, q odd (for r-D or r-LD-codes, see Remark 1)

$*e$ = optimal: the upper bound coincides with the lower bound given in Table 1

$*f$ = optimal, other cases

g, h and i refer to articles: g = [20], h = [21], i = [11].

4 Analysis of the results, conjectures and open problems

Tables 2, 3 and 4 provide upper bounds for the minimum sizes of r-D codes, r-LD codes and r-Id codes in the graphs L_q^n for $4 \leqslant q \leqslant 6$, $2 \leqslant n \leqslant 7$ and $1 \leqslant r \leqslant 5$ (remember that, for $q = 2$ and $q = 3$, Lee distance and Hamming distance are the same, yielding the equality $L_q^n = H_q^n$; so we do not give results for $q \in \{2, 3\}$: they can be found in the references given above, in Section 1).

Some of these values are optimal: they are indicated by a star in the tables. Is it possible to improve some of the others? This is a challenge...

From the values displayed in Tables 2, 3 and 4, we may observe interesting phenomena.

- Almost all the known upper bounds are significantly improved.
- For any fixed parameters n and q, $\gamma_r(L_q^n)$ is non-increasing when r increases; this is quite obvious, since the size $\mathcal{V}_{n,r,q}$ of the closed r-neighbourhood of any vertex increases with r: any vertex r-dominates more and more vertices and any r-dominating code is also an $(r+1)$-dominating code.
- This is not the case for $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$. Indeed, when r increases, the number of vertices r-dominated by a given codeword increases too, exactly as for γ_r, but not necessarily the number of r-separated vertices. More precisely, for any fixed parameters n and q, the computed values for $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$ first non-increase and then non-decrease. This suggests the following conjecture:

Conjecture 1 and open problem 1. [Unimodality with respect to r] Let n and q be given. When r increases (starting from $r = 1$), the function $\mathrm{LD}_r(L_q^n)$, after a possible non-increase, non-decreases until it reaches the value $q^n - 1$; the function $\mathrm{Id}_r(L_q^n)$, after a possible non-increase, non-decreases until L_q^n admits no r-identifying code.

If so, what are the values of r in function of n and q for which $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$ reach their minimum values?

- If now whe fix r and q, we observe that $\gamma_r(L_q^n)$ non-decreases when n increases (note that γ_r may be steady: for instance $\gamma_3(L_4^2) = \gamma_3(L_4^3) = 2$). This behaviour is systematic, as shown by the following proposition:

Proposition 1. Let r and q be given. When n decreases until $n = 2$, the function $\gamma_r(L_q^n)$ is non-increasing.

Proof. Let C_{n+1} be an optimal r-dominating code in L_q^{n+1}, and C_n be the code defined in L_q^n by $C_n = \{(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_{n+1}) : \exists j \in \{0, 1, \ldots, q-1\}$ such that $(c_1, c_2, \ldots, c_{i-1}, j, c_{i+1}, \ldots, c_{n+1}) \in C_{n+1}\}$, that is, we simply delete the i-th coordinate in C_{n+1}, for one $i \in \{1, 2, \ldots, n+1\}$. In the following, for simplicity, we take $i = n + 1$. Consider any vertex $(v_1, \ldots, v_n) \in L_q^n$. For every vertex $v = (v_1, \ldots, v_n, v_{n+1}) \in L_q^{n+1}$, there is a codeword $c = (c_1, \ldots, c_n, c_{n+1}) \in C_{n+1}$ with $d_L(v, c) \leqslant r$. Then $d_L((v_1, \ldots, v_n), (c_1, \ldots, c_n)) \leqslant r$, which proves that C_n is r-dominating in L_q^n. Hence $\gamma_r(L_q^n) \leqslant |C_n| \leqslant |C_{n+1}| = \gamma_r(L_q^{n+1})$. □

- The behaviour of $\mathrm{LD}_r(L_q^n)$ is not the same as the one of $\gamma_r(L_q^n)$ and seems more erratic when n increases with fixed r and q: for $r = 5$ and $q = 4$, $\mathrm{LD}_5(L_4^n)$ first increases ($\mathrm{LD}_5(L_4^2) = 15$, $\mathrm{LD}_5(L_4^3) = 32$) then decreases and, if we consider the computed values, finally increases again (21 for $n = 4$, 16 for $n = 5$, 25 for $n = 6$, 41 for $n = 7$). Such a behaviour may be rather surprising since the number of vertices to be r-dominated and the number of pairs of vertices to be r-separated increase dramatically: thus $\mathrm{LD}_r(L_q^n)$ may be expected to increase when n increases. Similarly, $\mathrm{Id}_r(L_q^n)$ does not inscrease with every value of n. But in fact, these phenomena seem to appear only when n is small enough with respect to r. Hence the following conjecture and open problem:

Conjecture 2 and open problem 2. [Increase of LD_r and Id_r with respect to n for n large enough] Let r and q be given. There exist integers $N_{LD}(r, q)$ and $N_{Id}(r, q)$ such that, respectively for $n \geqslant N_{LD}(r, q)$ and $n \geqslant N_{Id}(r, q)$, the values of $\mathrm{LD}_r(L_q^n)$ and of $\mathrm{Id}_r(L_q^n)$ increase when n increases.

If so, what are the values of $N_{LD}(r, q)$ and $N_{Id}(r, q)$ in function of r and q?

- When considering the Hamming distance, we have the following inequality ([13]; see also [3]): for $p > r \geqslant 1$, $\mathrm{Id}_r(H_2^{n+p}) \leqslant 2^p \mathrm{Id}_r(H_2^n)$. On the other hand, we have the following theorem for γ_r, thanks to the so-called *direct sum construction*:

Theorem 1. Let r and q be given. Then, for any dimension n, we have the inequality $\gamma_r(L_q^{n+1}) \leqslant q \times \gamma_r(L_q^n)$.

Proof. Let C_n be a minimum r-D code in L_q^n: $|C_n| = \gamma_r(L_q^n)$. For each codeword

$u = (u_1, u_2, ..., u_n)$ of C_n, consider the q vertices $(u_1, u_2, ..., u_n, i)$ of L_q^{n+1} with $i \in \{0, 1, ..., q-1\}$. Let C_{n+1} be the code of L_q^{n+1} thus obtained. Let $v = (v_1, ..., v_n, v_{n+1})$ be any vertex of L_q^{n+1}. Since C_n is an r-D code of L_q^n, there exists a codeword $(u_1, u_2, ..., u_n) \in C_n$ with $d_L((v_1, ..., v_n), (u_1, ..., u_n)) \leqslant r$. Hence the inequality $d_L((v_1, ..., v_n, v_{n+1}), (u_1, ..., u_n, v_{n+1})) \leqslant r$. Since $(u_1, ..., u_n, v_{n+1})$ belongs to C_{n+1}, C_{n+1} is an r-D code of L_q^{n+1} with $q \times \gamma_r(L_q^n)$ codewords. □

Note that, if the computed values for $\gamma_1(L_4^6)$ and $\gamma_1(L_4^7)$, i.e. 385 and 1540, or for $\gamma_1(L_6^4)$, $\gamma_1(L_6^5)$, $\gamma_1(L_6^6)$ and $\gamma_1(L_6^7)$, i.e. respectively 144, 864, 5 184 and 31 104, are the minimum ones, then the inequality $\gamma_1(L_q^{n+1}) \leqslant q \times \gamma_1(L_q^n)$ of the theorem would be tight for these values.

These observations lead us to the following conjecture (which would extend the inequality $\mathrm{Id}_r(H_2^{n+p}) \leqslant 2^p \mathrm{Id}_r(H_2^n)$ to any value of p, in particular for $r = p = 1$):

Conjecture 3. [Bounded increase of LD_r and Id_r when the dimension n increases by 1]. Let r and q be given. For any dimension n, we have $\mathrm{LD}_r(L_q^{n+1}) \leqslant q \times \mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^{n+1}) \leqslant q \times \mathrm{Id}_r(L_q^n)$.

- Tables 5, 6 and 7 provide the densities of the computed codes, i.e. the ratios between the computed cardinalities and the number of vertices. For γ_r, the computed densities are non-increasing both with respect to n (as announced by Theorem 1) and r (see above for the decrease with respect to r) but not with respect to q (compare for instance the – exact – densities for $\gamma_1(L_5^2)$ and $\gamma_1(L_6^2)$). For LD_r and Id_r, the computed densities are decreasing with respect to n (see Conjecture 3) but not with respect to r or to q (compare for instance the values for $\mathrm{LD}_2(L_4^2)$ and $\mathrm{LD}_3(L_4^2)$, the values for $\mathrm{LD}_1(L_4^3)$ and $\mathrm{LD}_1(L_5^3)$, the values for $\mathrm{Id}_2(L_4^2)$ and $\mathrm{Id}_3(L_4^2)$, the values for $\mathrm{Id}_1(L_4^5)$ and $\mathrm{Id}_1(L_5^5)$).

Conjecture 4. [Non-increase of the densities with respect to n]. Let r and q be given. Conjecture 3 is equivalent to the non-increase of the densities for $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$; furthermore, we conjecture that the densities for $\gamma_r(L_q^n)$, $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$ tend towards 0 when n increases.

- The compared costs of r-domination, r-location-domination and r-identification, i.e. the differences between optimal cardinalities, are studied for general graphs in [19]. While the gaps between optimal cardinalities can be very large for general graphs, the differences between the computed values for L_q^n are, for a large majority of cases, rather small with respect to the number q^n of vertices, as shown by Tables 5, 6 and 7. In particular, the computed values of $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$ become very close when n increases for given r and q: r-separating all the pairs of vertices does not cost much more than r-separating only the pairs of non-codewords. Sometimes, the computed values of $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$ are the same (see Tables 2, 3 and 4). Hence an open problem:

Open problem 3. [Equality between $\mathrm{LD}_r(L_q^n)$ and $\mathrm{Id}_r(L_q^n)$]. Can we characterize the values of n, r and q for which we have $\mathrm{LD}_r(L_q^n) = \mathrm{Id}_r(L_q^n)$?

- Figures 1, 2 and 3 allow to see the evolution of the ratios $\gamma_r(L_q^n)/\mathrm{Id}(L_q^n)$, or rather their approximations, based on the computed values of these two parameters, respectively for $q = 4$, $q = 5$ and $q = 6$. Each figure contains five curves, one for each value of $r \in \{1, 2, 3, 4, 5\}$. The x-axis represents the dimension n; the y-axis provides the approximation of $\gamma_r(L_q^n)/\mathrm{Id}(L_q^n)$. As we have $\gamma_r(L_q^n) \leqslant \mathrm{Id}(L_q^n)$, the ratios are always upper bounded by 1. In a sense, a curve illustrates, for the associated radius r, the part of the code required for the r-domination and the one required for the r-separation. Except for few values, we observe that the curves are rather increasing, which would mean that the part of the r-domination becomes more and more important inside the code when n increases, for fixed q and r. Of course, as the computed values of $\mathrm{LD}(L_q^n)$ and of $\mathrm{Id}(L_q^n)$ are quite similar, we would observe the same phenomenon if studying the ratios $\gamma_r(L_q^n)/\mathrm{LD}(L_q^n)$. This suggests the next open problem.

Open problem 4. [Increase of $\gamma_r(L_q^n)/\mathrm{LD}(L_q^n)$ and of $\gamma_r(L_q^n)/\mathrm{Id}(L_q^n)$ with respect to n for n large enough] Let r and q be given. Do there exist integers $N_{LD}(r, q)$ and $N_{Id}(r, q)$ such that, respectively for $n \geqslant N_{LD}(r, q)$ and $n \geqslant N_{Id}(r, q)$, the values of $\gamma_r(L_q^n)/\mathrm{LD}(L_q^n)$ and of $\gamma_r(L_q^n)/\mathrm{Id}(L_q^n)$ increase when n increases?

If so, what are the values of the limits of $\gamma_r(L_q^n)/\mathrm{LD}(L_q^n)$ and of $\gamma_r(L_q^n)/\mathrm{Id}(L_q^n)$? Are they equal to 1? Otherwise, is the limit of $\mathrm{LD}(L_q^n)/\mathrm{Id}(L_q^n)$ anyway equal to 1?

References:

[1] J. Astola, *The theory of Lee codes*, Research Report 1, Lappeenranta University of Technology, Department of Physics and Mathematics, 1982.

[2] D. Auger, I. Charon, O. Hudry and A. Lobstein, Complexity results for identifying codes in planar graphs, *International Transactions in Operational Research* Vol. 17, No 6, 2010, pp. 691-710.

[3] I. Charon, G. Cohen, O. Hudry and A. Lobstein, New identifying codes in the binary Hamming space, *European J. Combin.*, Vol. 31, 2010, pp. 491–501. See also: perso.telecom-paristech.fr/∼hudry/newIdentifyingNcube.html

[4] I. Charon and O. Hudry, The noising methods: a generalization of some metaheuristics, *European Journal of Operational Research*, Vol. 135, No 1, 2001, pp. 86-101.

[5] I. Charon and O. Hudry, The noising methods, in: P. Siarry (ed.), *Heuristics: Theory and Applications*, Nova Publishers, 2013, pp. 1-30.

[6] I. Charon, O. Hudry and A. Lobstein, Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard, *Theoretical Computer Science* A, Vol. 290, No 3, 2003, pp. 2109-2120.

[7] G. Cohen, I. Honkala, S. Litsyn and A. Lobstein, *Covering Codes*, Elsevier, Amsterdam, 1997.

[8] C. J. Colbourn, P. J. Slater and L. K. Stewart, Locating dominating sets in series parallel networks, *Congr. Numer.*, Vol. 56, 1987, pp. 135–162.

[9] R. Dhanalakshmi and C. Durairajan, On r-identifying codes in binary Hamming space, q-ary Lee space and incomplete hypercube, *Discrete Math., Algo. and Appl.*, Vol. 11, No 2, 2019, article No 1950027.

[10] R. Dhanalakshmi and C. Durairajan, Constructions of r-identifying codes and $(r, \leqslant \ell)$-identifying codes, *Indian J. Pure Appl. Math.*, Vol. 50, No 2, 2019, pp. 531–547.

[11] R. Dhanalakshmi and C. Durairajan, Bounds on r-identifying codes in q-ary Lee space, *Contributions to Discrete Mathematics*, Vol. 16, No 1, 2021, pp. 53–71.

[12] G. Exoo, V. Junnila, T. Laihonen and S. Ranto, Improved bounds on identifying codes in binary Hamming spaces, *European J. Combin.*, Vol. 31, 2010, pp. 813–827.

[13] G. Exoo, T. Laihonen, S. Ranto, Improved upper bounds on binary identifying codes, *IEEE Transactions on Information Theory*, Vol. 53, 2007, pp. 4255–4260.

[14] S. Gravier, R. Klasing and J. Moncel, Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs, *Algorithmic Operations Research*, Vol. 3, No 1, 2008, pp. 43-50.

[15] H. O. Hämäläinen, I. S. Honkala, S. Litsyn and P. R. J. Östergård, Football pools – a game for mathematicians, *Am. Math. Mon.*, Vol. 102, 1995, pp. 579–588.

[16] T. W. Haynes, S. T. Hedetniemi and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, 1998.

[17] T. W. Haynes, S. T. Hedetniemi and M. A. Henning (eds.), *Topics in Domination in Graphs*, Springer, Berlin, 2020.

[18] I. Honkala, T. Laihonen and S. Ranto, On locating-dominating codes in binary Hamming spaces, *Discrete Math. Theor. Comput. Sci.*, Vol. 6, 2004, pp. 265–282.

[19] O. Hudry and A. Lobstein, The compared costs of domination, location-domination and identification, *Discussiones Mathematicae-Graph Theory*, Vol. 40, 2020, pp. 127–147.

[20] M. G. Karpovsky, K. Chakrabarty and L. B. Levitin, On a new class of codes for identifying vertices in graphs, *IEEE Trans. Inform. Theory*, Vol. IT-44, 1998, pp. 599–611.

[21] J. L. Kim and S. J. Kim, Identifying codes in q-ary hypercubes, *Bull. Instit. Combin. Appl.*, Vol. 59, 2010, pp. 93–102.

[22] A. Lobstein, Covering radius. https://www.lri.fr/∼lobstein/bib-a-jour.pdf

[23] A. Lobstein, Watching systems, identifying, locating-dominating and discriminating codes in graphs, a bibliography. https://www.lri.fr/∼lobstein/debutBIBidetlocdom.pdf

[24] A. Lobstein, O. Hudry and I. Charon, Locating-domination and identification, in: T. Hayes, S. Hedetniemi, M. Henning (eds.), *Topics in Domination in Graphs*, Springer, Berlin, 2020, pp. 251-299.

[25] R. Marti, P. M. Pardalos and M. G. C. Resende (eds), *Handbook of Heuristics*, Springer, Berlin, 2018.

[26] D. F. Rall and P. J. Slater, On location-domination numbers for certain classes of graphs, *Congr. Numer.*, Vol. 45, 1984, pp. 97–106.

[27] N. S. V. Rao, Computational complexity issues in operative diagnosis of graph-based systems, *IEEE Trans. Comput.*, Vol. 42, 1993, pp. 447–457.

[28] N. V. Shinde and B. N. Waphare, Improved upper bounds for identifying codes in n-dimensional q-ary cubes, *Int. J. Appl. Comput. Math.*, Vol. 6, 2020, article No 43.

[29] P. Siarry (ed.), *Heuristics: Theory and Applications*, Nova Publishers, New York, 2013.

[30] P. J. Slater, *Domination and location in graphs*, Research Report No 93, National University of Singapore, 1983.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

The three authors contribute equally to all the aspects of this work.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

	$r=1$		$r=2$		$r=3$		$r=4$		$r=5$	
n (q^n)	$\mathcal{V}_{n,r,q}$	$\lceil \frac{q^n}{\mathcal{V}_{n,r,q}} \rceil$	$\mathcal{V}_{n,r,q}$	$\lceil \frac{q^n}{\mathcal{V}_{n,r,q}} \rceil$	$\mathcal{V}_{n,r,q}$	$\lceil \frac{q^n}{\mathcal{V}_{n,r,q}} \rceil$	$\mathcal{V}_{n,r,q}$	$\lceil \frac{q^n}{\mathcal{V}_{n,r,q}} \rceil$	$\mathcal{V}_{n,r,q}$	$\lceil \frac{q^n}{\mathcal{V}_{n,r,q}} \rceil$
$q=4$										
2 (16)	5	4	11	2	15	2	16	1	16	1
3 (64)	7	10	22	3	42	2	57	2	63	2
4 (256)	9	29	37	7	93	3	163	2	219	2
5 (1024)	11	94	56	19	176	6	386	3	638	2
6 (4096)	13	316	79	52	299	14	794	6	1145	4
7 (16384)	15	1093	106	155	470	35	1471	12	3473	5
$q=5$										
2 (25)	5	5	13	2	21	2	25	1	25	1
3 (125)	7	18	25	5	57	3	93	2	117	2
4 (625)	9	70	41	16	121	6	257	3	417	2
5 (3125)	11	285	61	52	221	15	581	6	1173	3
6 (15625)	13	1202	85	184	365	43	1145	14	2777	6
7 (78125)	15	5209	113	692	561	140	2045	39	5797	14
$q=6$										
2 (36)	5	8	13	3	23	2	31	2	35	2
3 (216)	7	31	25	9	60	4	108	2	156	2
4 (1296)	9	144	41	32	125	11	285	5	517	3
5 (7776)	11	707	61	128	226	35	626	13	1378	6
6 (46656)	13	3589	85	549	371	126	1211	39	3143	15
7 (279936)	15	18663	113	2478	568	493	2136	132	6392	44

Table 1: lower bounds given by (3), for $4 \leqslant q \leqslant 6$, $2 \leqslant n \leqslant 7$ and $1 \leqslant r \leqslant 5$.

$n\,(q^n)$	$r=1$			$r=2$			$r=3$			$r=4$			$r=5$		
	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id
2 (16)	4^{*e}	6^{*b}	$7^{*f}\,[7^h]$	2^{*e}	5^{*b}	$4^i\ 6^{*f}\ [7^i]$	2^{*c}	$[8^{*c}]$	$15^{*b}[15^{*c}]$	1^{*d}	$[15^{*d}]$	-	1^{*d}	$[15^{*d}]$	-
3 (64)	12^{*b}	16	$18^h\ 19\ [21^i]$	4^{*b}	8^{*b}	$8^{*f}\ [8^i]$	2^{*e}	7^{*b}	$5^i\ 7^{*b}[7^i]$	2^{*e}	15	$4^i\ 18\ [20^i]$	2^{*c}	$[32^{*c}]$	$[63^{*c}]$
4 (256)	32	56	$56^h\ 62\ [67^i]$	12	20	$17^i\ 21\ [22^i]$	4^{*b}	13	$8^i\ 13\ [14^i]$	2^{*e}	11	$5^i\ 13\ [14^i]$	2^{*e}	21	$4^i\ 21\ [22^i]$
5 (1024)	120	203	$183^h\ 211\ [266^i]$	30	57	$42^i\ 60\ [92^i]$	12	28	$17^i\ 28\ [51^i]$	4^{*b}	16	$8^i\ 16\ [24^i]$	2^{*e}	16	$5^i\ 18\ [28^i]$
6 (4096)	385	723	$623^h\ 748\ [1064^i]$	104	174	$115^i\ 187\ [352^i]$	31	67	$35^i\ 70\ [154^i]$	12	34	$16^i\ 34\ [64^i]$	4^{*e}	25	$8^i\ 25\ [46^i]$
7 (16384)	1540	2690	$2164^h\ 2711\ [4256^i]$	341	568	$334^i\ 586\ [1407^i]$	96	182	$81^i\ 184\ [462^i]$	33	77	$32^i\ 78\ [176^i]$	13	41	$15^i\ 41\ [112^i]$

Table 2: bounds for $q=4$

$n\,(q^n)$	$r=1$			$r=2$			$r=3$			$r=4$			$r=5$		
	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id
2 (25)	5^{*e}	9^{*b}	$10^{*b}\ [10^g]$	3^{*b}	6^{*b}	$6^{*b}\ [6^i]$	2^{*e}	9^{*b}	$10^{*b}\ [11^i]$	1^{*d}	$[24^{*d}]$	-	1^{*d}	$[24^{*d}]$	-
3 (125)	20	32	$36\ [39^i]$	8^{*b}	14	$15\ [16^i]$	3^{*e}	10	$11\ [11^i]$	2^{*e}	13	$14\ [15^i]$	2^{*e}	27	$32\ [33^i]$
4 (625)	84	139	$150\ [195^i]$	20	44	$48\ [73^i]$	9	23	$25\ [42^i]$	4^{*b}	18	$18\ [18^i]$	3^{*b}	20	$20\ [20^i]$
5 (3125)	397	624	$654\ [975^i]$	98	163	$175\ [365^i]$	31	69	$70\ [160^i]$	13	37	$37\ [52^i]$	5^{*b}	27	$27\ [38^i]$
6 (15625)	1792	2811	$2912\ [4875^i]$	391	639	$685\ [1521^i]$	113	217	$219\ [624^i]$	40	93	$93\ [256^i]$	16	52	$52\ [176^i]$
7 (78125)	8422	13135	$13539\ [24375^i]$	1648	2613	$2701\ [7605^i]$	419	733	$739\ [2847^i]$	130	264	$265\ [1168^i]$	48	121	$124\ [520^i]$

Table 3: bounds for $q=5$

$n\,(q^n)$	$r=1$			$r=2$			$r=3$			$r=4$			$r=5$		
	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id
2 (36)	8^{*e}	12^{*b}	$12^g\ 14^{*b}\ [14^i]$	4^{*b}	7^{*b}	$8^{*b}\ [9^g]$	2^{*e}	7^{*b}	$8^{*b}\ [9^i]$	2^{*e}	9^{*b}	$13^{*b}\ [14^i]$	2^{*c}	$[18^{*c}]$	$[35^{*c}]$
3 (216)	36	52	$54^{*g}\ 54\ [54^{*g}]$	12	23	$24\ [27^i]$	6^{*b}	14	$14\ [16^i]$	2^{*e}	11	$11\ [13^i]$	2^{*e}	14	$14\ [16^i]$
4 (1296)	144	288	$260^g\ 310\ [324^g]$	47	94	$98\ [149^i]$	16	46	$47\ [75^i]$	9	26	$27\ [48^i]$	4^{*b}	23	$23\ [29^i]$
5 (7776)	864	1552	$1296^g\ 1609\ [1701^g]$	241	416	$445\ [756^i]$	79	167	$168\ [378^i]$	30	79	$81\ [224^i]$	14	47	$47\ [117^i]$
6 (46656)	5184	8616	$6666^g\ 8747\ [8748^g]$	1195	1992	$2039\ [2916^i]$	338	636	$644\ [1458^i]$	117	255	$259\ [729^i]$	47	126	$128\ [432^i]$
7 (279936)	31104	$[34992^a]$	$[34992^{*fg}]$	5961	9378	$10540\ [17496^i]$	1517	2583	$2882\ [8046^i]$	473	887	$901\ [4023^i]$	177	376	$389\ [2025^i]$

Table 4: bounds for $q=6$

$n\,(q^n)$	$r=1$			$r=2$			$r=3$			$r=4$			$r=5$		
	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id
2 (16)	25 %*	37.5 %*	43.75 %*	12.5 %*	31.25 %*	37.5 %*	12.5 %*	50 %*	93.75 %*	6.25 %*	93.75 %*	-	6.25* %	93.75* %	-
3 (64)	18.75 %*	25 %	29.69 %	6.25 %*	12.5 %*	12.5 %*	3.13 %*	10.94 %*	10.94 %*	3.13 %*	23.44 %	28.13 %	3.13 %*	50 %*	98.44 %*
4 (256)	12.5 %	21.88 %	24.22 %	4.69 %	7.81 %	8.20 %	1.56 %*	5.08 %	5.08 %	0.78 %*	4.30 %	5.08 %	0.78 %*	8.20 %	8.20 %
5 (1024)	11.72 %	19.82 %	20.61 %	2.93 %	5.57 %	5.86 %	1.17 %	2.73 %	2.73 %	0.39 %*	1.56 %	1.56 %	0.20 %*	1.56 %	1.76 %
6 (4096)	9.40 %	17.65 %	18.26 %	2.54 %	4.25 %	4.57 %	0.76 %	1.64 %	1.71 %	0.29 %	0.83 %	0.83 %	0.10 %*	0.61 %	0.61 %
7 (16384)	9.40 %	16.42 %	16.55 %	2.08 %	3.47 %	3.58 %	0.59 %	1.11 %	1.12 %	0.20 %	0.47 %	0.48 %	0.08 %	0.25 %	0.25 %

Table 5: upper bounds for densities for $q = 4$

$n\,(q^n)$	$r=1$			$r=2$			$r=3$			$r=4$			$r=5$		
	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id
2 (25)	20 %*	36 %*	40 %*	12 %*	24 %*	24 %*	8 %*	36 %*	40 %*	4 %*	96 %*	-	4 %*	96 %*	-
3 (125)	16 %	25.6 %	28.8 %	6.4 %*	11.2 %	12 %	2.4 %*	8 %	8.8 %	1.6 %*	10.4 %	11.2 %	1.6 %*	21.6 %	25.6 %
4 (625)	13.44 %	22.24 %	24 %	3.2 %	7.04 %	7.68 %	1.44 %	3.68 %	4 %	0.64 %*	2.88 %	2.88 %	0.48 %*	3.2 %	3.2 %
5 (3125)	12.7 %	19.97 %	20.93 %	3.14 %	5.22 %	5.6 %	0.99 %	2.21 %	2.24 %	0.42 %	1.18 %	1.18 %	0.16 %*	0.86 %	0.86 %
6 (15625)	11.47 %	17.99 %	18.64 %	2.50 %	4.09 %	4.38 %	0.72 %	1.39 %	1.40 %	0.26 %	0.60 %	0.60 %	0.10 %	0.33 %	0.33 %
7 (78125)	10.78 %	16.81 %	17.33 %	2.11 %	3.34 %	3.46 %	0.54 %	0.94 %	0.95 %	0.17 %	0.34 %	0.34 %	0.06 %	0.15 %	0.16 %

Table 6: upper bounds for densities for $q = 5$

$n(q^n)$	$r=1$			$r=2$			$r=3$			$r=4$			$r=5$		
	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id	γ	LD	Id
2 (36)	22.22 %*	33.33 %*	38.89 %*	11.11 %*	19.44 %*	22.22 %*	5.56 %*	19.44 %*	22.22 %*	5.56 %*	25 %*	36.11 %*	5.56 %*	50 %*	97.22 %*
3 (216)	16.67 %	24.07 %	25 %*	5.56 %	10.65 %	11.11 %	2.78 %*	6.48 %	6.48 %	0.93 %*	5.09 %	5.09 %	0.93 %*	6.48 %	6.48 %
4 (1296)	11.11 %	22.22 %	23.92 %	3.63 %	7.25 %	7.56 %	1.23 %	3.55 %	3.63 %	0.69 %	2.01 %	2.08 %	0.31 %*	1.77 %	1.77 %
5 (7776)	11.11 %	19.96 %	20.69 %	3.10 %	5.35 %	5.72 %	1.02 %	2.15 %	2.16 %	0.39 %	1.02 %	1.04 %	0.18 %	0.60 %	0.60 %
6 (46656)	11.11 %	18.47 %	18.75 %	2.56 %	4.27 %	4.37 %	0.72 %	1.36 %	1.38 %	0.25 %	0.55 %	0.56 %	0.10 %	0.27 %	0.27 %
7 (279936)	11.11 %	12.5 %	12.5 %*	2.13 %	3.35 %	3.77 %	0.54 %	0.92 %	1.03 %	0.17 %	0.32 %	0.32 %	0.06 %	0.13 %	0.14 %

Table 7: upper bounds for densities for $q = 6$

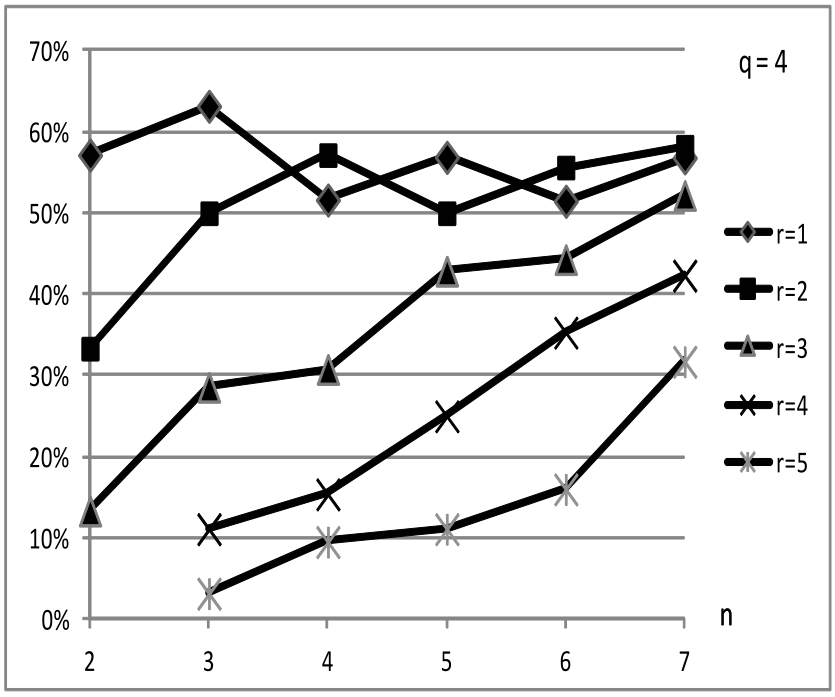

Figure 1: Approximation of the ratios $\gamma_r(L_4^n)/\mathrm{Id}(L_4^n)$ from the computed values.

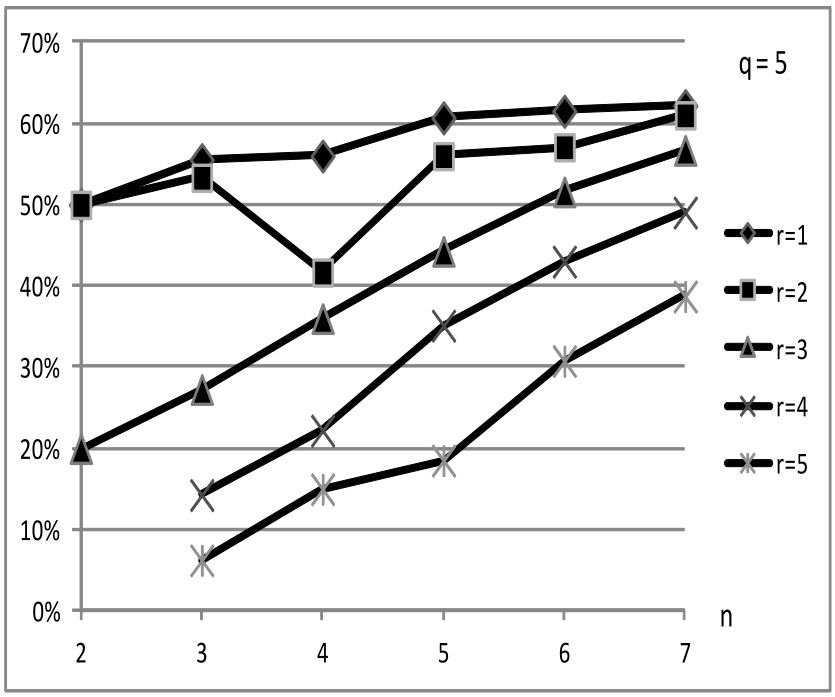

Figure 2: Approximation of the ratios $\gamma_r(L_5^n)/\mathrm{Id}(L_5^n)$ from the computed values.

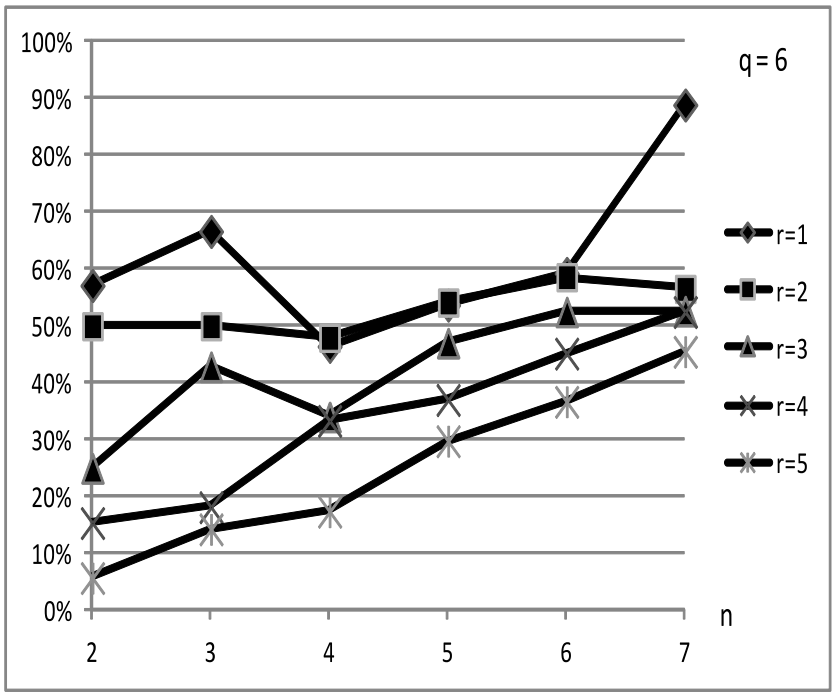

Figure 3: Approximation of the ratios $\gamma_r(L_6^n)/\mathrm{Id}(L_6^n)$ from the computed values.