Desgins, Codes and Cryptography manuscript No.
(will be inserted by the editor)

New Lower Bounds for the Shannon Capacity of Odd Cycles

K. Ashik Mathew · Patric R. J. Östergård

Received: date / Accepted: date

Abstract The Shannon capacity of a graph G is defined as $c(G) = \sup_{d\geq 1}(\alpha(G^d))^{\frac{1}{d}}$, where $\alpha(G)$ is the independence number of G. The Shannon capacity of the cycle C_5 on 5 vertices was determined by Lovász in 1979, but the Shannon capacity of a cycle C_p for general odd p remains one of the most notorious open problems in information theory. By prescribing stabilizers for the independent sets in C_p^d and using stochastic search methods, we show that $\alpha(C_7^5) \geq 350$, $\alpha(C_{11}^4) \geq 748$, $\alpha(C_{13}^4) \geq 1534$, and $\alpha(C_{15}^3) \geq 381$. This leads to improved lower bounds on the Shannon capacity of C_7 and C_{15}: $c(C_7) \geq 350^{\frac{1}{5}} > 3.2271$ and $c(C_{15}) \geq 381^{\frac{1}{3}} > 7.2495$.

Keywords cube packing · independent set · Shannon capacity · zero-error capacity

Mathematics Subject Classification (2000) 05C69 · 94A24

1 Introduction

The Shannon capacity of a graph is an important information-theoretic parameter and plays a central role in the study of the zero-error capacity of a noisy communication channel represented by the graph [18]. A communication channel transmitting p different symbols can be represented by a graph G with vertex set $V(G)$ and edge set $E(G)$ in the following way: $V(G)$ is the set of transmitted symbols, and for $v_1, v_2 \in V(G)$, $\{v_1, v_2\} \in E(G)$ if the symbols v_1 and v_2 are indistinguishable. The *Shannon capacity* of G is defined as

$$c(G) = \sup_{d\geq 1}(\alpha(G^d))^{\frac{1}{d}},$$

K. Ashik Mathew and P. R. J. Östergård
Department of Communications and Networking
Aalto University School of Electrical Engineering
P.O. Box 13000, 00076 Aalto, Finland
E-mail: ashikmathewk@gmail.com; patric.ostergard@aalto.fi

where $\alpha(G)$ is the independence number of G and the graph strong product is assumed [23]. For a survey of some of the early results related to the Shannon capacity of graphs, see [17].

Algebraic tools for the study of the Shannon capacity were proposed by Haemers [12,13] while the Shannon capacity of digraphs was investigated by Alon [1]. See also [2,3,10,24] for some related studies.

For a channel transmitting p symbols represented by the elements of the set $\mathbb{Z}_p = \{0,1,2,\ldots,p-1\}$ and where two distinct symbols s and t are indistinguishable if $s - t \equiv \pm 1 \pmod{p}$, the graph that represents the channel is C_p, the cycle on p vertices. If p is even, then $c(C_p) = p/2$. It was shown by Lovász [19] in 1979 that $c(C_5) = \sqrt{5}$, but finding the Shannon capacity of C_p for $p \geq 7$ and odd is still open [5].

It is well known that the independence number of C_p^d, the dth power (under strong product) of a cycle C_p, is same as the number of hypercubes of side 2 that can be packed in a discrete d-dimensional torus of width p, denoted by $G(d,p)$ [9]. See Figure 1 for a visualization of such a 2-dimensional packing and a corresponding independent set in C_5^2. Representing independent sets as a packing of cubes often gives a more comprehensible model (visually) to work with. This is especially true when we talk about symmetries, though for the cause of adhering to formalism, we stick to a rather algebraic notion to discuss symmetries in the remaining sections.

Cube packings and their different variants also form the basis of several classical and well-studied problems in combinatorics [9,16]. The function $G(d,p)$ has been studied thoroughly, and exact values and bounds have been published in [4] and later studies. Several of these results have been obtained using exhaustive and stochastic computational methods. For example, Baumert *et al.* [4] used exhaustive search to show that $G(3,7) = 33$, and Vesel and Žerovnik [25] proved that $G(4,7) \geq 108$ with simulated annealing. The current authors used another stochastic (local search) method, tabu search, to obtain lower bounds for the capacity of triangular graphs [20]. (Triangular graphs are closely related to cycle graphs; the capacity problem for triangular graphs can be studied via a generalization of the cube packing problem.)

Many of the best known cube packings possess some kind of symmetry. For example, the packing in Figure 1 has a symmetry generated by $(a,b) \to (a+2,b+1)$ (addition modulo 5). This symmetry generates a group of order 5. Several additional examples can be found in the constructions of [4].

In the current work, stochastic computational methods will be combined with the idea of prescribing symmetries of packings. By prescribing symmetries, one is able to speed up the computer search. Obviously, such a search has a possibility of success only if there are packings with the given symmetries. By exploiting possible symmetries in as exhaustive manner as possible, we are able to show that $\alpha(C_7^5) \geq 350$, $\alpha(C_{11}^4) \geq 748$, $\alpha(C_{13}^4) \geq 1534$, and $\alpha(C_{15}^3) \geq 381$. These bounds further imply that $c(C_7) \geq 350^{\frac{1}{5}} > 3.2271$ and $c(C_{15}) \geq 381^{\frac{1}{3}} > 7.2495$.

The paper is organized as follows. In Section 2, the approach of prescribing symmetries is considered, and a stochastic local search method for finding packings is discussed in Section 3. In Section 4, the results are summarized and tabulated. Specific packings are listed in the Appendix.

	4	4	5	5
3	4	4		3
3		2	2	3
1	1	2	2	
1	1		5	5

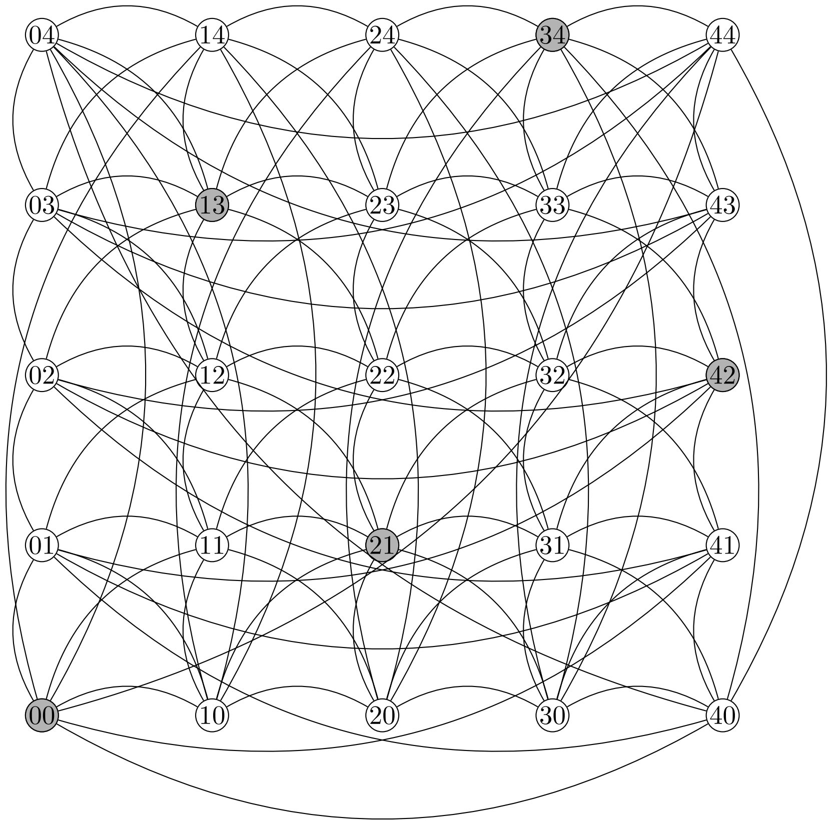

Fig. 1 Packing a torus with 2-dimensional cubes and a corresponding independent set in C_5^2

2 Prescribing Symmetries of Independent Sets

The graph $G = C_p^d$ is conveniently discussed in the framework of codes. Let $V(G) = \{0, 1, \dots, p-1\}^d$, the set of all codewords of length d over $\mathbb{Z}_p$. For $v \in V(G)$, we denote $v = (v_1, v_2, \dots, v_d)$. Now we define the set of edges as

$$E(G) = \{\{v, v'\} : v, v' \in V(G) \text{ and } \max_{1 \le i \le d} \min\{|v_i - v'_i|, p - |v_i - v'_i|\} < 2\}. \qquad (1)$$

This definition further shows the one-to-one correspondence between an independent set in C_p^d and a packing in the discrete d-dimensional torus of width p by (hyper)cubes of side 2 (with their centers—or any other specific position of the cubes—in the position given by the element of the independent set).

For small parameters, one may find the independence number of C_p^d using Cliquer [22] or some other available software. However, growing parameters makes the use of such exact algorithms infeasible at some point. One way of handling large instances of combinatorial search problems is to prescribe symmetries [15, Chapter 9].

Symmetries of an independent set in a graph G are elements of the automorphism group of G—denoted by $\mathrm{Aut}(G)$—that stabilize the independent set.

Let $N[v]$ denote the closed neighborhood of the vertex v, that is, $N[v] = \{v\} \cup \{v' : \{v,v'\} \in E(G)\}$. A graph G is called *thin* if $N[u] \neq N[v]$ whenever $u \neq v$. A *prime* graph G is one that cannot be written as $G = G_1 \boxtimes G_2$ (strong product of G_1 and G_2) for non-trivial graphs G_1 and G_2.

Theorem 1 ([14, Theorem 7.18]) *For a graph $G = G_1 \boxtimes G_2 \boxtimes \cdots \boxtimes G_n$ where G_1, G_2, ..., G_n are connected, thin and prime graphs, the automorphism group of G is isomorphic to the automorphism group of the disjoint union of graphs $G_1, G_2, \ldots, G_n$.*

We use the conventional notation of D_n for the dihedral group of order $2n$ and S_n for the symmetric group of degree n.

Theorem 2 ([11]) *If G is a connected graph and nG denotes the graph representing n disjoint copies of G, then $\mathrm{Aut}(nG)$ is the wreath product $\mathrm{Aut}(G) \wr S_n$.*

Theorem 3 *The automorphism group of C_p^d for $p > 3$ is isomorphic to the wreath product $D_p \wr S_d$.*

Proof We first show that C_p with vertex set $V(C_p) = \{0,1,2,\ldots,p-1\}$ and edge set $E(C_p) = \{\{u,v\} : u - v \equiv \pm 1 \pmod p\}$ is thin and prime for $p > 3$. Consider any two distinct vertices $x, y \in V(C_p)$. If $\{x,y\} \notin E(C_p)$, $x \notin N[y]$ and so $N[x] \neq N[y]$. Suppose $\{x,y\} \in E(C_p)$. This means that, w.l.o.g., $x - y \equiv 1 \pmod p$. Consider the vertex $z = (x+1) \bmod p$. Clearly, $\{x,z\} \in E(C_p)$ and $\{y,z\} \notin E(C_p)$, leading to $N[x] \neq N[y]$. So, C_p is thin. Now we observe that C_p is prime by the following argument. By definition, if $G = G_1 \boxtimes G_2$ is connected, both G_1 and G_2 are connected. Since $K_2 \boxtimes K_2 = K_4$, the strong product of any two graphs with at least one edge each has K_4 as a subgraph. Since C_p does not have K_4 as a subgraph, C_p is prime. Now that C_p is connected, thin and prime, Theorem 1 can be applied. Since the automorphism group of C_p is isomorphic to the dihedral group D_p, the result follows from Theorem 2.

The order of the group $\mathrm{Aut}(C_p^d)$ is $|\mathrm{Aut}(C_p^d)| = |D_p \wr S_d| = (2p)^d d!$. In the framework of codes, introduced earlier, elements of $\mathrm{Aut}(C_p^d)$ act by a permutation of the coordinates followed by permutations of the coordinate values (separately for each coordinate) that have the form

$$i \to (ai + b) \bmod p,\ a \in \{-1, 1\},\ b \in \mathbb{Z}_p, \tag{2}$$

Given two codes corresponding to independent sets or packings of cubes, we say that these are *equivalent* if one of the codes can be obtained from the other with a mapping in the action of $\mathrm{Aut}(C_p^d)$. Such mappings from a code onto itself—which are formally *stabilizers* of the code (and the corresponding independent set and the packing) under the action of $\mathrm{Aut}(C_p^d)$—are said to form the automorphism group of the code.

The automorphism group of a code is a subgroup of $\mathrm{Aut}(C_p^d)$. When prescribing possible automorphism groups, we therefore consider subgroups of $\mathrm{Aut}(C_p^d)$ up to conjugacy. Moreover, we reduce the number of groups to consider by explicitly restricting the computations to cyclic groups. (In this manner we are still able to cover a large part of the groups, since most large groups that are omitted will have a cyclic subgroup amongst the groups considered.)

Having prescribed an automorphism group of a code, the action of the group partitions all possible codewords into orbits. In the framework of independent sets, we now get instances of the maximum weight independent set problem. The vertex set consist of all *admissible* orbits: the pairs of codewords in the set must fulfill the distance criterion in (1). The weight of a vertex is the number of codewords in the orbit. Finally, edges are inserted whenever no pairs of codewords, one from each of the orbits, violate the distance criterion in (1).

By prescribing automorphism groups, we can extend the range of parameters of codes for which the running time of Cliquer (which can also handle weighted graphs) or similar software is feasibly short. Moreover, by also changing the computational approach from being exact to becoming stochastic, we can extend the range of parameters even further. Such an approach will be discussed next.

3 Stochastic Search for Weighted Independent Sets

The graphs obtained in the previous section are weighted. In general, let G be an arbitrary graph, each vertex of which has a positive integer weight. The problem of finding the maximum weight of an independent set in G is obviously a generalization of the maximum independent set problem, which we get by letting all weights be 1. Note that since an independent set corresponds to a clique in the complement graph, any discussion of independent sets and related algorithms apply to cliques and vice versa. The maximum (weight) independent set problem is surveyed in [7], in the framework of cliques.

The problem of deciding whether there is an independent set of weight at least k in a graph G is NP-complete, and no polynomial-time general algorithms are known. However, the problem has a wide variety of applications and is of general importance, so a lot of effort has been put on developing exact as well as stochastic algorithms. In many of the stochastic algorithms, one maintains an independent set I, which is repeatedly altered throughout the search. For unweighted graphs, the alteration step commonly consists of the removal of *one* vertex from I followed by the addition of a nonnegative number of vertices.

Montemanni and Smith [21] modify the approach described above by removing not one but *many* vertices at a time from I. After removing a set of vertices, an exact algorithm (like Cliquer [22]) can be used to find a set of vertices to add that have the largest possible weight. In some sense, this approach lies in between basic stochastic search and exact algorithms.

The main decision to be made in the approach by Montemanni and Smith is the choice of vertices to remove from the independent set I. In [21] vertices are removed in a random manner. When one thinks about this problem in the context of cube packings, removal means removing (hyper)cubes. When cubes are removed, there will be holes in the packing. But when such holes are not connected, we have a situation equivalent to that of sequentially removing a smaller number of cubes in different parts of the packing. One may therefore try to modify the approach to make sure that only one hole emerges in each step.

To this end, the second author realized (in unpublished work) that a (heuristic) metric

$$d_h : V(G) \times V(G) \to \mathbb{R}$$

may be used for instances of the maximum weight independent set problem that come from packing problems. Specifically, in the removal step a random vertex $v \in I$ is chosen and all vertices v' fulfilling $d_h(v, v') \leq b$ for some given value of b are removed from I. Some experimenting is typically needed to find a proper metric and value of b for the problem at hand and its instances. This approach has been used, for example, to find new q-analog packings [8].

To arrive at the metric needed in the current work, we utilize the Lee distance

$$d_L(v, v') = \sum_{i=1}^{d} \min\{|v_i - v'_i|, p - |v_i - v'_i|\},$$

where $v, v' \in \{0, 1, \ldots, p-1\}^d$, to get

$$d_h(o, o') = \min_{v \in o, v' \in o'} d_L(v, v')$$

for two orbits of codewords o, o' under the action of the prescribed group. Recall that we have one vertex in $V(G)$ for each orbit, so we now have a required metric.

We here outline the new algorithm and present it as Algorithm 1 in pseudocode. The pseudocode algorithm has two parameters: the graph G and the parameter for the heuristic metric, b.

The iterative steps of the algorithm are carried out in a loop. There are several possibilities for the loop condition. In the current work, where we have old bounds that we want to improve, we may iterate until such an improvement has taken place. Of course, this might never happen, so some timeout must in any case be built into the loop condition. Also note that, to optimize the performance of stochastic algorithms, one should always consider restarting the search at regular intervals, rather than carrying out one very long search.

Parts of the pseudocode algorithm are implemented as calls to functions. The function RANDOMELEMENT returns a random element from a set, MAXINDSET returns a maximum weight independent set in a graph (for this, Cliquer [22] was used in the current work), and WEIGHT gives the total weight of an independent set.

The variable I contains the independent set that is processed, and I_r is such a set encountered during the search with the maximum weight. Any method (greedy,

Algorithm 1 Partial neighborhood search

```
procedure SEARCH($G$:weighted graph, $b$:integer)
 1: Initialize independent set $I \subset V(G)$
 2: $I_r \leftarrow I$
 3: while LOOP-CONDITION-IS-TRUE do
 4:     $i \leftarrow$ RANDOMELEMENT($I$)
 5:     $I' \leftarrow \{v \in I : d_h(v,i) > b\}$
 6:     $T \leftarrow V(G) \setminus (\cup_{v \in I'} N[v])$
 7:     $T \leftarrow T \setminus \{i\}$
 8:     $I \leftarrow I' \cup$ MAXINDSET($G[T]$)
 9:     if WEIGHT($I$) > WEIGHT($I_r$) then
10:         $I_r \leftarrow I$
11:     end if
12: end while
13: Output $I_r$
end procedure
```

stochastic, etc.) can be used to obtain an initial (maximal) independent set in line 1. The other variables in the algorithm are auxiliary. For $T \subseteq V(G)$, the subgraph of G induced by T is denoted by $G[T]$. The statement in line 7 prevents the independent set I from remaining unchanged.

4 Results

By applying the approaches discussed in this paper and using more than 2 CPU-years in the stochastic search, we have obtained independent sets that attain the following bounds for $G(d,p)$: $G(5,7) \geq 350$, $G(4,11) \geq 748$, $G(4,13) \geq 1534$, and $G(3,15) \geq 381$. These also lead to improved lower bounds on the Shannon capacity of C_7 and C_{15}: $c(C_7) \geq 350^{\frac{1}{5}} > 3.2271$ and $c(C_{15}) \geq 381^{\frac{1}{3}} > 7.2495$. The previous best known lower bounds for $c(C_7)$ and $c(C_{15})$ were $108^{\frac{1}{4}} > 3.2237$ ([25]) and $380^{\frac{1}{3}} > 7.2431$ ([4]), respectively. The best known lower bounds for $c(C_p)$ for other small cycles of odd length are: $c(C_9) \geq 81^{\frac{1}{3}} > 4.3267$ ([4]), $c(C_{11}) \geq 148^{\frac{1}{3}} > 5.2895$ ([4]) and $c(C_{13}) \geq 247^{\frac{1}{3}} > 6.2743$ ([6]). The Lovász ϑ-function [19] gives an upper bound for the Shannon capacity of an odd cycle C_p and then also for $G(n,d)$:

$$c(C_p) \leq \frac{p\cos\frac{\pi}{p}}{1+\cos\frac{\pi}{p}}, \tag{3}$$

$$G(d,p) \leq \left(\frac{p\cos\frac{\pi}{p}}{1+\cos\frac{\pi}{p}}\right)^d. \tag{4}$$

Using (3), we get $c(C_7) < 3.3177$, $c(C_9) < 4.3601$, $c(C_{11}) < 5.3864$, $c(C_{13}) < 6.4042$, and $c(C_{15}) < 7.4172$.

The currently best known upper and lower bounds for $G(d,p)$ are listed in Table 1 together with keys. Only one key is provided in cases where the value can be obtained using more than one method.

Table 1 Bounds on $G(d,p)$

$p \backslash d$	1	2	3	4	5
5	$^{a}2^{a}$	$^{a}5^{a}$	$^{c}10^{f}$	$^{c}25^{d}$	$^{c}50$–55^{j}
7	$^{a}3^{a}$	$^{a}10^{a}$	$^{f}33^{f}$	$^{h}108$–115^{d}	$^{k}350$–401^{j}
9	$^{a}4^{a}$	$^{a}18^{a}$	$^{e}81^{d}$	$^{c}324$–361^{j}	$^{c}1458$–1575^{j}
11	$^{a}5^{a}$	$^{a}27^{a}$	$^{e}148^{d}$	$^{k}748$–814^{d}	$^{c}3996$–4477^{d}
13	$^{a}6^{a}$	$^{a}39^{a}$	$^{g}247^{i}$	$^{k}1534$–1605^{d}	$^{c}9633$–10432^{d}
15	$^{a}7^{a}$	$^{a}52^{a}$	$^{k}381$–390^{d}	$^{b}2720$–2925^{d}	$^{c}19812$–21937^{d}

Key to Table 1.

^{a}	$G(1,p) = \lfloor p/2 \rfloor$, $G(2,p) = \lfloor (p^2 - p)/4 \rfloor$ [4, Theorem 2]
^{b}	$G(d,p) \geq 1 + G(d,p-2)(p^d - 2^d)/(p-2)^d$ [4, Corollary 2]
^{c}	$G(d,p) \geq G(d_1,p)G(d-d_1,p)$ [4, Corollary 3]
^{d}	$G(d,p) \leq G(d-1,p)p/2$ [4, Lemma 2]
^{e}	Baumert *et al.* [4, Theorem 3]
^{f}	Baumert *et al.* [4, Theorem 4]
^{g}	Baumert *et al.* [4, Theorem 6]
^{h}	Vesel and Žerovnik [25]
^{i}	Bohman, Holzman, and Natarajan [6]
^{j}	Lovász [19], here (4)
^{k}	This paper, see Appendix

Acknowledgement

The work was supported in part by the Academy of Finland, Project No. 289002.

Appendix

We here list codes giving the four new lower bounds. The permutation of coordinates is the identity permutation in all generators of the groups, and $a = 1$ for all value permutations in (2). We therefore present the groups by simply listing the values of b for the d value permutations of a generator.

$G(5,7) \geq 350$:

Generator: (0, 1, 1, 5, 1)
Group order: 7
Orbit representatives: (0, 5, 6, 6, 0), (0, 0, 6, 6, 0), (3, 3, 0, 6, 0), (0, 5, 2, 1, 0), (2, 5, 6, 5, 0), (1, 3, 0, 0, 0), (2, 3, 2, 0, 0), (2, 1, 0, 4, 0), (2, 5, 2, 1, 0), (0, 2, 1, 2, 0), (4, 2, 6, 3, 0), (4, 0, 0, 3, 0), (5, 1, 2, 3, 0), (3, 6, 1, 5, 0), (4, 5, 0, 2, 0), (3, 4, 5, 2, 0), (1, 6, 1, 6, 0), (2, 3, 6, 4, 0), (5, 5, 1, 4, 0), (5, 3, 1, 3, 0), (6, 4, 0, 1, 0), (0, 0, 2, 2, 0), (6, 0, 1, 0, 0), (5, 1, 5, 0, 0), (5, 6, 6, 0, 0), (5, 3, 5, 1, 0), (0, 3, 6, 5, 0), (2, 0, 2, 1, 0), (4, 0,

3, 5, 0), (4, 4, 2, 6, 0), (4, 2, 3, 6, 0), (1, 5, 5, 3, 0), (6, 2, 4, 5, 0), (4, 4, 4, 6, 0), (6, 2, 0, 0, 0), (1, 0, 5, 4, 0), (4, 6, 4, 0, 0), (3, 1, 4, 1, 0), (3, 6, 5, 2, 0), (6, 0, 5, 4, 0), (2, 3, 3, 2, 0), (1, 1, 4, 2, 0), (1, 5, 3, 3, 0), (1, 2, 4, 4, 0), (1, 1, 1, 6, 0), (3, 1, 1, 6, 0), (0, 3, 3, 2, 0), (6, 4, 3, 4, 0), (6, 6, 3, 4, 0), (6, 5, 5, 4, 0)

$G(4,11) \geq 748$:

Generator: (1, 5, 8, 9)
Group order: 11
Orbit representatives: (9, 10, 0, 0), (7, 10, 9, 0), (5, 4, 0, 0), (5, 4, 2, 0), (2, 3, 0, 0), (7, 4, 2, 0), (7, 8, 1, 0), (1, 10, 2, 0), (2, 7, 4, 0), (0, 7, 5, 0), (6, 9, 7, 0), (6, 6, 1, 0), (8, 6, 1, 0), (8, 8, 10, 0), (4, 2, 1, 0), (5, 2, 3, 0), (6, 4, 4, 0), (5, 2, 5, 0), (3, 7, 6, 0), (2, 9, 6, 0), (4, 9, 6, 0), (1, 9, 4, 0), (1, 7, 7, 0), (5, 7, 8, 0), (6, 6, 10, 0), (3, 2, 3, 0), (8, 4, 4, 0), (3, 7, 8, 0), (10, 10, 2, 0), (0, 5, 5, 0), (9, 6, 5, 0), (4, 9, 8, 0), (4, 7, 10, 0), (0, 10, 0, 0), (7, 2, 4, 0), (7, 2, 6, 0), (7, 0, 7, 0), (6, 8, 10, 0), (0, 3, 10, 0), (8, 6, 3, 0), (10, 6, 3, 0), (3, 5, 0, 0), (1, 1, 1, 0), (0, 5, 7, 0), (2, 5, 7, 0), (2, 3, 9, 0), (1, 5, 9, 0), (3, 5, 9, 0), (3, 0, 1, 0), (4, 0, 3, 0), (2, 0, 4, 0), (3, 9, 4, 0), (4, 0, 5, 0), (6, 0, 5, 0), (9, 8, 1, 0), (10, 1, 8, 0), (8, 1, 9, 0), (1, 1, 10, 0), (10, 1, 10, 0), (0, 8, 2, 0), (9, 8, 3, 0), (9, 1, 6, 0), (10, 3, 6, 0), (8, 4, 6, 0), (5, 0, 7, 0), (1, 3, 7, 0), (10, 3, 8, 0), (9, 10, 9, 0)

$G(4,13) \geq 1534$:

Generator: (0, 1, 0, 2)
Group order: 13
Orbit representatives: (9, 6, 7, 0), (9, 8, 9, 0), (7, 1, 8, 0), (0, 7, 6, 0), (8, 10, 8, 0), (7, 11, 0, 0), (4, 11, 11, 0), (5, 4, 2, 0), (8, 12, 9, 0), (3, 7, 10, 0), (5, 7, 10, 0), (5, 0, 1, 0), (2, 0, 12, 0), (1, 7, 8, 0), (1, 7, 10, 0), (0, 7, 4, 0), (12, 0, 9, 0), (5, 2, 2, 0), (11, 2, 7, 0), (11, 0, 5, 0), (10, 4, 5, 0), (1, 5, 5, 0), (3, 8, 2, 0), (4, 0, 12, 0), (11, 0, 7, 0), (5, 12, 5, 0), (3, 5, 7, 0), (5, 12, 7, 0), (7, 12, 7, 0), (8, 12, 11, 0), (2, 12, 3, 0), (2, 3, 4, 0), (4, 10, 4, 0), (7, 2, 1, 0), (0, 9, 5, 0), (0, 9, 7, 0), (1, 5, 7, 0), (10, 4, 7, 0), (1, 1, 2, 0), (6, 10, 4, 0), (6, 10, 6, 0), (12, 11, 5, 0), (12, 11, 7, 0), (11, 2, 9, 0), (11, 4, 9, 0), (6, 3, 8, 0), (6, 5, 9, 0), (7, 1, 10, 0), (7, 3, 10, 0), (5, 9, 10, 0), (9, 10, 10, 0), (6, 5, 11, 0), (10, 1, 3, 0), (4, 5, 9, 0), (0, 11, 9, 0), (2, 11, 11, 0), (7, 4, 2, 0), (4, 6, 3, 0), (6, 6, 3, 0), (9, 12, 3, 0), (0, 0, 0, 0), (8, 1, 12, 0), (6, 7, 12, 0), (6, 9, 12, 0), (9, 10, 12, 0), (8, 8, 3, 0), (6, 8, 4, 0), (8, 10, 4, 0), (11, 2, 5, 0), (7, 6, 5, 0), (9, 6, 5, 0), (9, 8, 5, 0), (7, 8, 6, 0), (8, 10, 6, 0), (9, 8, 7, 0), (12, 1, 2, 0), (1, 3, 2, 0), (0, 12, 2, 0), (3, 9, 11, 0), (2, 3, 7, 0), (4, 3, 8, 0), (2, 9, 9, 0), (2, 11, 9, 0), (7, 0, 1, 0), (9, 1, 1, 0), (9, 12, 1, 0), (4, 8, 4, 0), (5, 1, 8, 0), (11, 0, 0, 0), (11, 11, 0, 0), (9, 10, 1, 0), (11, 12, 2, 0), (10, 3, 3, 0), (12, 10, 3, 0), (0, 2, 0, 0), (11, 2, 0, 0), (0, 9, 9, 0), (12, 2, 11, 0), (11, 4, 11, 0), (1, 9, 11, 0), (0, 11, 11, 0), (2, 2, 0, 0), (4, 2, 0, 0), (2, 4, 0, 0), (4, 4, 0, 0), (0, 11, 0, 0), (3, 6, 1, 0), (3, 10, 2, 0), (2, 1, 4, 0), (3, 12, 5, 0), (2, 1, 6, 0), (5, 1, 6, 0), (4, 3, 6, 0), (3, 7, 8, 0), (10, 6, 9, 0), (12, 0, 11, 0), (10, 6, 11, 0), (10, 8, 11, 0)

$G(3,15) \geq 381$:

Generator: (5, 0, 10)
Group order: 3
Orbit representatives: (1, 10, 4), (1, 11, 0), (10, 10, 0), (1, 2, 2), (13, 3, 2), (9, 11, 2), (2, 8, 4), (7, 11, 2), (1, 0, 2), (3, 11, 0), (5, 11, 1), (1, 10, 2), (3, 10, 4), (0, 12, 2), (12, 10, 1), (14, 10, 1), (10, 9, 2), (9, 7, 3), (8, 9, 1), (10, 9, 4), (7, 11, 4), (9, 11, 4), (7, 7, 1), (7, 7, 3), (8, 5, 3), (11, 12, 1), (13, 12, 1), (14, 0, 0), (6, 9, 1), (3, 11, 2), (6, 9, 3), (4, 8, 4), (6, 5, 0), (2, 4, 3), (1, 0, 0), (12, 1, 1), (1, 6, 1), (5, 7, 2), (2, 8, 2), (4, 9, 2), (1, 6, 3), (3, 6, 4), (4, 4, 2), (6, 5, 2), (12, 14, 1), (11, 12, 3), (13, 12, 3), (12, 5, 4), (4, 5, 0), (5, 7, 0), (2, 9, 0), (4, 9, 0), (3, 6, 2), (11, 7, 4), (11, 7, 2), (12, 6, 0), (11, 8, 0), (14, 6, 1), (0, 8, 1), (8, 9, 3), (9, 3, 3), (11, 3, 3), (4, 4, 4), (6, 4, 4), (3, 7, 0), (13, 8, 0), (13, 8, 2), (14, 6, 3), (0, 8, 3), (12, 10, 3), (14, 10, 3), (3, 2, 3), (5, 3, 0), (7, 3, 0), (10, 1, 1), (5, 2, 4), (10, 14, 3), (8, 13, 4), (1, 2, 0), (14, 2, 0), (13, 4, 0), (0, 4, 1), (2, 4, 1), (0, 4, 3), (5, 0, 1), (14, 14, 2), (12, 14, 3), (0, 12, 4), (6, 13, 2), (5, 0, 3), (5, 11, 3), (6, 13, 4), (9, 3, 1), (11, 3, 1), (8, 5, 1), (9, 7, 1), (10, 5, 2), (12, 5, 2), (10, 5, 4), (5, 6, 4), (9, 12, 0), (10, 14, 1), (8, 13, 2), (3, 0, 3), (4, 13, 3), (0, 13, 0), (2, 13, 0), (4, 13, 1), (5, 2, 2), (7, 2, 2), (9, 1, 3), (7, 0, 4), (7, 2, 4), (3, 0, 1), (3, 2, 1), (14, 1, 2), (12, 1, 3), (1, 1, 4), (14, 1, 4), (13, 3, 4), (8, 1, 0), (8, 14, 0), (7, 0, 2), (2, 13, 2), (2, 12, 4), (1, 14, 4), (14, 14, 4)

References

1. Alon, N.: On the capacity of digraphs. European J. Combin. **19**, 1–5 (1998).
2. Alon, N., Orlitsky, A.: Repeated communication and Ramsey graphs. IEEE Trans. Inform. Theory **41**, 1276–1289 (1995).
3. Ashley, J.J., Siegel, P.H.: A note on the Shannon capacity of run-length-limited codes. IEEE Trans. Inform. Theory **33**, 601–605 (1987).
4. Baumert, L.D., McEliece, R.J., Rodemich, E., Rumsey Jr., H.C., Stanley, R., Taylor, H.: A combinatorial packing problem. In: Birkhoff, G., Hall Jr., M. (eds.) Computers in Algebra and Number Theory (Proc. SIAM-AMS Sympos. Appl. Math., New York, 1970), pp. 97–108. Amer. Math. Soc., Providence (1971).
5. Bohman, T.: A limit theorem for the Shannon capacities of odd cycles I. Proc. Amer. Math. Soc. **131** 3559–3569 (2003).
6. Bohman, T., Holzman, R., Natarajan, V.: On the independence numbers of the cubes of odd cycles. Electron. J. Combin. **20**(3), #P10 (2013).
7. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Du, D.Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization, Supplement Vol. A, pp. 1–74. Kluwer, Dordrecht (1999).
8. Braun, M., Östergård, P.R.J., Wassermann, A.: New lower bounds for binary constant dimension subspace codes. Submitted for publication.
9. Brouwer, A.E., Schrijver, A.: Uniform hypergraphs. In: Schrijver A. (ed.) Packing and Covering in Combinatorics, Math. Centre Tracts No. 106, pp. 39–73. Mathematisch Centrum, Amsterdam (1979).
10. Feige, U.: Randomized graph products, chromatic numbers, and the Lovász ϑ-function, In: Proc. 27th Annual ACM Symposium on the Theory of Computing, pp. 635–640. ACM, New York (1995).
11. Frucht, R.: On the groups of repeated graphs. Bull. Amer. Math. Soc. **55**, 418–420 (1949).
12. Haemers, W.: An upper bound for the Shannon capacity of a graph. In: Lovász, L., Sós, V.T. (eds.) Algebraic Methods in Graph Theory, Colloq. Math. Soc. János Bolyai, Vol. 25, pp. 267–272. Szeged, Hungary (1978).
13. Haemers, W.: On some problems of Lovász concerning the Shannon capacity of a graph. IEEE Trans. Inform. Theory **25**, 231–232 (1979).

14. Hammack, R., Imrich, W., Klavžar, S.: Handbook of Product Graphs, Second Edition. CRC Press, Boca Raton, 2011.
15. Kaski, P., Östergård, P.R.J.: Classification Algorithms for Codes and Designs. Springer-Verlag, Berlin (2006).
16. Keller, O.H.: Über die lückenlose Erfüllung des Raumes mit Würfeln. J. Reine Angew. Math. **163**, 231–248 (1930).
17. Knuth, D.E.: The sandwich theorem. Electron. J. Combin. **1**, #A1 (1994).
18. Körner, J., Orlitsky, A.: Zero-error information theory. IEEE Trans. Inform. Theory **44**, 2207–2229 (1998).
19. Lovász, L.: On the Shannon capacity of a graph. IEEE Trans. Inform. Theory **25**, 1–7 (1979).
20. Mathew, K.A., Östergård, P.R.J., Popa, A.: On the Shannon capacity of triangular graphs. Electron. J. Combin. **20**(2), #P27 (2013).
21. Montemanni, R., Smith, D.H.: Heuristic algorithms for constructing binary constant weight codes. IEEE Trans. Inform. Theory **55**, 4651–4656 (2009).
22. Niskanen, S., Östergård, P.R.J.: Cliquer User's Guide (version 1.0). Technical report T48, Commun. Lab., Helsinki Univ. Technol., Espoo (2003).
23. Shannon, C.E.: The zero-error capacity of a noisy channel. IRE Trans. Inform. Theory **2**, 8–19 (1956).
24. Szegedy, M.: A note on the Θ number of Lovász and the generalized Delsarte bound. In: Proc. 35th Annual Symposium on Foundations of Computer Science, pp. 36–41. IEEE Computer Society Press, Los Alamitos (1994).
25. Vesel, A., Žerovnik, J.: Improved lower bound on the Shannon capacity of C_7. Inform. Process. Lett. **81**, 277–282 (2002).