STORAGE CODES ON TRIANGLE-FREE GRAPHS WITH ASYMPTOTICALLY UNIT RATE

ALEXANDER BARG, MOSHE SCHWARTZ, AND LEV YOHANANOV

Abstract. A storage code on a graph $G$ is a set of assignments of symbols to the vertices such that every vertex can recover its value by looking at its neighbors. We consider the question of constructing large-size storage codes on triangle-free graphs. Previously it was shown that there are infinite families of binary storage codes of rate converging to $3/4$, constructed on coset graphs of binary linear codes. Here we construct such codes of rate asymptotically approaching one, thereby offering a complete solution to this problem.

Equivalently, this question can be phrased as a version of hat-guessing games on graphs (e.g., P.J. Cameron e.a., Electronic J. Comb. 2016). In this language, we construct triangle-free graphs with success probability of the players approaching one as the number of vertices tends to infinity. Equivalently again, there exist linear index codes on such graphs of rate approaching zero.

1. Introduction

Suppose we are given a connected graph $G(V, E)$ on $N$ vertices. Denote by $\mathcal{N}(v)$ the neighborhood of $v$ in $G$, i.e., the set of vertices of $G$ adjacent to $v$. A storage code for $G$ is a set $\mathcal{C}$ of vectors $c = (c_v)_{v \in V} \in \mathbb{F}_q^N$ such that for every $c \in \mathcal{C}$ and $v \in V$ the value of the coordinate $c_v$ is uniquely determined by the values of $c_w$, $w \in \mathcal{N}(v)$. More formally, suppose that for every vertex $v$ there is a function $f_v : 2^\mathcal{N} \rightarrow \mathbb{F}_q$ such that for every $c \in \mathcal{C}$ and every $v \in V$ the value $c_v = f_v((c_w)_{w \in \mathcal{N}(v)})$, where we assume some implicit fixed ordering of the vertices.

The concept of storage codes was introduced around 2014 in [6, 10] and studied in subsequent papers [7, 2]. In coding theory literature, this concept was motivated by a more general notion of codes with locality [5], which has enjoyed considerable attention during the last decade [8].

The main problem associated with storage codes is constructing codes of large size, expressed as a function of the parameters of the graph. To make the comparison easier, define the code rate $R_q(\mathcal{C}, G) := \log_q(|\mathcal{C}|)/N$, and let $R_q(G) = \max_{\mathcal{C}} R_q(\mathcal{C}, G)$. The first observation is that for dense graphs it is easy to construct codes of large size or rate; for instance, if the graph is complete, $G = K_N$, then the condition of vertex recovery from its neighbors will be satisfied if the codevectors satisfy a global parity check, i.e., $\sum_{v \in V} c_v = 0$ for every $c \in \mathcal{C}$. In this case $|\mathcal{C}| = q^{N-1}$ and $R(\mathcal{C}) = R_q(K_N) = \frac{N-1}{N}$ for any $q$. A variant of this idea clearly applies if $G$ is a dense graph, in particular, if it has many cliques, and therefore it makes sense to focus on the case of graphs with no cliques at all, i.e., triangle-free graphs. This case was afforded some attention in the literature both on storage codes and on guessing games on graphs which we discuss next.

The following version of guessing games on graphs, introduced in [9], turns out to be equivalent to the construction problem of storage codes. The vertices are assigned colors
out of a finite set $Q$ of size $q$, and each vertex attempts to guess its color based on the colors of its neighbors. The game is won if all the vertices correctly guess their colors. The strategy may be agreed upon before the start of the game, but no communication between the vertices is allowed once the colors have been assigned. Suppose that the assignment $x \in Q^N$ is chosen randomly from the available options, and the goal of the game is to minimize the probability of failure. The following strategy connects this game with storage codes. Let $\mathcal{C}$ be a $q$-ary storage code for the graph. Every vertex $v$ assumes that the assignment is a codeword in $\mathcal{C}$ and guesses its color from its neighbors, then the probability of success is $P_s(\mathcal{C}, G) = \frac{|\mathcal{C}|}{q^N}$. This quantity is sometimes expressed via the guessing number of $G$, defined as $\text{gn}_q(G) = N + \log_q \max P_s(\mathcal{C}, G)$ [3], so in our notation $\frac{1}{N}(\text{gn}_q(G)) = R_q(G)$.

It was soon realized that the problem of finding storage codes is also equivalent to constructing procedures for linear index coding with side information graph $G$. We refer to the introduction of [2] for a brief overview of the known results on storage codes as well as the connections to guessing and linear index coding, and to [1] for a more detailed presentation.

This paper develops the work [2], and it shares with it the general approach to the construction of storage codes. In particular, as in [2], we confine ourselves to linear binary codes and we assume that the repair function of a vertex $f_v$ is simply a parity check. In other words, the value $c_v = \sum_{w \in \mathcal{N}(v)} c_w$ for all $c \in \mathcal{C}$ and all $v \in V$, where the sum is computed modulo 2. This also relies on an implicit assumption that vertex recovery relies on the full parity, i.e., all the neighbors contribute to the recovery (the general definition does not include this stipulation). Thus, the parities of the vertices are given by the corresponding rows of the matrix $A + I$, where $A = A(G)$ is the adjacency matrix of $G$ and $I$ is the identity (in other words, we are adding a self-loop to every vertex $v \in V$). For a given graph $G$ define the augmented adjacency matrix $\bar{A}(G) := A(G) + I$. The dimension of the code $\mathcal{C}$ equals $\dim(\mathcal{C}) = \frac{N}{q} - \text{rk} \bar{A}(G)$, and thus we are interested in constructing graphs for which $\bar{A}(G)$ has the smallest possible rank given the number of vertices $N$.

For a $d$-regular graph it is easy to construct a code of rate $1/2$. This is a folklore result accomplished by placing bits on the edges and assigning to every vertex the $d$-vector of bits written on the edges incident to it. The rate of the obtained code over the alphabet of size $2^d$ is readily seen to be $1/2$. Constructing codes of higher rate on triangle-free graphs turned out to be difficult, and until recently only isolated examples were known in the literature [3]. Improving upon these results, the authors of [2] constructed an infinite family of triangle-free graphs $G_r$ with $|V(G_r)| = 2^{r+1}$ vertices, $r \geq 4$, that afford storage codes of rate $R = \frac{1}{4} - 2^{-r^2}$. Moreover, [2] also gave an example of a triangle-free graph on $N = 2^{16}$ vertices with $\text{rk}(\bar{A}) = 11818/65536$, i.e., code rate slightly above 0.8196. The authors of [2] speculated that there may exist families of triangle-free graphs that afford storage codes of rate approaching one. As our main result, we confirm this conjecture.

**Theorem 1.** There exists an infinite family of connected triangle-free graphs $(G_m)_m$ on $N = N(m)$ vertices such that $\frac{1}{N} \text{rk}(\bar{A}(G_m)) \to 0$ as $m \to \infty$. In other words,

$$\lim_{m \to \infty} R_2(G_m) = 1$$

and

$$\text{gn}_2(G_m) = N(1 - o(1)).$$
If we let $R_2(N) = \max R_2(G_N)$ to be the largest code rate over all triangle-free graphs on $N$ vertices, then this theorem implies that $\lim \sup_{N \to \infty} R_2(N) = 1$. Since $R_q(N) \geq R_2(N)$ for all alphabets with $q \geq 2$ a power of a prime, this claim is also true for all such $q$.

Graphs in the family $(G_m)_m$ are constructed as coset graphs of binary linear codes. Their explicit description will be given in Sec. 5 after we develop the tools for analyzing the rank of adjacency matrices of coset graphs. Along the way we illustrate the usefulness of our approach by giving simple proofs of some of the earlier results concerning storage codes on coset graphs.

2. THE CONSTRUCTION OF [2] AND OUR APPROACH

To remind the reader the definition of coset graphs, suppose we are given a binary linear code $C \in F^n$ with an $r \times n$ parity-check matrix $H$, where $F = \mathbb{F}_2$. This means that $C = \ker(H)$ and $\dim(C) \leq n - r$; in other words, that $H$ is allowed to contain dependent rows. Denote by $S$ the set of columns of $H$. Now consider the Cayley graph $G = \text{Cay}(F^r, S)$ on the group $(F^r)_+$ with generators in $S$, where $r = n - k$ is the number of rows of $H$. The vertex set of $G$ is $V(G) = F^r$, and a pair of vertices $v, v'$ are connected if there is a column (generator) $h \in S$ such that $v = v' + h$. Since the group is Abelian, the graph $G$ is undirected. The vertices of $G$ can be also viewed as cosets in $F_2^n/C$, with two cosets connected if and only if the Hamming distance between them (as subsets) is one. It is clear that $G$ is triangle-free once the minimum distance of the code $C$ is at least four. We will assume this throughout and note that in particular, this implies that the columns in $H$ are distinct.

Having constructed $G$ from $C$, we consider the code $\mathcal{C} = \ker(\hat{A}(G))$. Clearly, $\mathcal{C}$ is a storage code for $G$. Below, to distinguish between the seed codes $C$ and storage codes $\mathcal{C}$, we call the former small codes and the latter big codes. Generally, finding the parameters of $\mathcal{C}$ from the parameters of $C$ is a nonobvious task, and even if the small code is simply a repetition code, computing the rank of $\hat{A}$ is not immediate. This problem in fact was considered earlier in the context of constructing quantum codes [4], and we refer to [2] for a brief discussion of this connection. The family of storage codes in [2] is constructed starting with the parity-check matrix $H_r$ obtained from the parity-check matrix of the extended Hamming code of length $2^{r-1}$ (see Sec. 5 for more details). Below we often find it convenient to include $0_r$ (the all-zero vector of length $r$) in the set $S$, and then the adjacency matrix $A(G)$ includes ones on the diagonal. Accordingly, in such cases we do not use the notation $\hat{A}$ and write $A$ instead.

The approach taken in [2] relies on detailed analysis of the action of the adjacency operator $A$ on the space of functions $f : F^r \to F$. While it indeed yields the value of the rank for the graphs $G$ obtained from $H_r$, extending it to other families of codes looks difficult. The approach taken here relies on the following observations. Generators $h \in S$ act on $F^r$ as permutations $\sigma_h$, and clearly $\sigma^2_h(v) = v$ for every $v \in F^r$. Therefore for a given $h \neq 0$, $\sigma_h$ can be written as a product of disjoint transpositions (cycles of length 2), $\sigma_h = (v_0, v_1)(v_2, v_3) \ldots (v_{2r-2}, v_{2r-1})$, where $v_{2i} + v_{2i+1} = h$ for all $i \geq 0$ (note that the labeling of the vectors depends on $h$). For any vector $v \in F^r$, not necessarily a generator, the action of $\sigma_v$ can be written as a $2^r \times 2^r$ permutation matrix, which we denote below by $\Gamma_v$. For every $v, u, w \in F^r$ we have $(\Gamma_v)_{u,w} = 1(v = u + w)$, i.e., every row contains a single 1 in the column $u + v$. In particular, $\Gamma_0 = I$. Denote by
\( \mathcal{M}_r = \{ \Gamma_v \mid v \in F^r \} \) the collection of matrices \( \Gamma_v \), and note that it forms a multiplicative group \( (\mathcal{M}_r, \times) \cong (F^r)^+ \); thus, \( \Gamma_v^2 = I \) and \( \Gamma_v \Gamma_w = \Gamma_w \Gamma_v \) for all \( v, w \).

Our approach has a number of common features with [2], but it is phrased entirely in terms of permutation matrices and their combinations, which supplant actions of the generators of the group. This view enables us to isolate subblocks of the adjacency matrix, and to manipulate those subblocks to rearrange the matrix so that it becomes possible to control the rank.

3. Properties of permutation matrices

In this section, we formulate some results about combinations (products and sums) of the matrices \( \Gamma_v \), where the computations are performed modulo 2. Throughout we assume some fixed order of the vectors in \( F^r \). Our first lemma expresses the following obvious fact: acting on \( F^r \) first by \( u \) and then by \( v \) is the same as acting by \( u + v \).

**Lemma 2.** For any \( v, w \in F^r \), \( \Gamma_v \Gamma_w = \Gamma_{v+w} \).

**Proof.** First, \( (\Gamma_v \Gamma_w)_{u,u'} = 1 \) if and only if \( (\Gamma_v)_{u,z} = 1 \) and \( (\Gamma_w)_{z,w'} = 1 \), i.e., \( v = u + z \) and \( w = z + u' \), or \( v + w = u + u' \). On the other hand, by definition \( (\Gamma_{v+w})_{u,u'} = 1 \) if and only if \( v + w = u + u' \). \( \square \)

For any \( u, w \in F^r \) there exists exactly one \( v \in F^r \) such that \( (\Gamma_v)_{u,w} \neq 0 \), and thus \( \sum_v \Gamma_v = 1_{2^r \times 2^r} \) (the \( 2^r \times 2^r \) all-ones matrix). Thus, if a binary matrix \( A \) can be written as a sum of permutations, this representation is unique (up to an even number of repeated summands). Therefore, if \( A_V := \sum_{v \in V} \Gamma_v \), where \( V \subset F^r \) is some subset, is a sum of permutation matrices, the quantity \( s(A_V) := |V| \) is well defined. Let \( \mathcal{P}_r := \{ A_V \mid V \subset F^r \} \) be the set of all sums of the permutation matrices. Below we usually suppress the subscript \( V \) from the notation.

**Proposition 3.** The set \( \mathcal{P}_r \) with operations \( + \) and \( \cdot \) forms a commutative matrix ring with identity.

**Proof.** By definition, the matrices in \( \mathcal{P}_r \) form a commutative additive group. The remaining properties of the ring follow immediately from Lemma 2 which reduces sums of products of \( \Gamma \)'s to simply sums. \( \square \)

As above, the rank in the next lemma is computed over \( \mathbb{F}_2 \).

**Lemma 4.** Let \( A, B \in \mathcal{P}_r \).

1. If \( s(A) \) is odd, then \( \text{rk}(A) = 2^r \).
2. If both \( s(A) \) and \( s(B) \) are odd, then \( s(AB) \) is odd.
3. If \( s(A) \) is odd, then there exists \( C \in \mathcal{P}_r \), such that \( AC = CA = B \).
4. For any \( \Gamma \in \mathcal{M}_r, \text{rk}(A) = \text{rk}(A\Gamma) \).

**Proof.** (1) For odd \( \ell \) let \( A = \sum_{i=1}^\ell \Gamma_{v_i} \) such that \( \Gamma_{v_i} \in \mathcal{M}_r \). Thus,

\[
A^2 = \left( \sum_{i=1}^\ell \Gamma_{v_i} \right)^2 = \sum_{i=1}^\ell \Gamma_{v_i}^2 + \sum_{i \neq j} \Gamma_{v_i+v_j}.
\]

The last sum has every term appearing twice, so it vanishes mod 2, and the first is formed of an odd number of identities \( I \), so it equals \( I \). Thus, \( A \) is nonsingular.
(2) For $\ell$ and $m$ odd, let $A = \sum_{i=1}^{\ell} \Gamma_{\ell_i}, B = \sum_{j=1}^{m} \Gamma_{w_j} \in \mathcal{P}_r$ such that $\Gamma_{\ell_i}, \Gamma_{w_j} \in \mathcal{M}_r$. Therefore,

$$AB = \sum_{i=1}^{\ell} \Gamma_{\ell_i} \cdot \sum_{j=1}^{m} \Gamma_{w_j} = \sum_{i=1}^{\ell} \sum_{j=1}^{m} \Gamma_{\ell_i+w_j}.$$ 

Since $\ell$ and $m$ are odd, the number of terms in this sum is odd. If some of them are repeated, they cancel in pairs, without affecting the parity.

(3) Let $C = AB$. Since $\mathcal{P}_r$ is closed under multiplication, $C \in \mathcal{P}_r$. Next, $CA = AC = A^2B = B$.

(4) This is obvious since multiplying by a permutation matrix over any field preserves the rank.

The next lemma establishes another simple property of the matrices in $\mathcal{P}_r$.

**Lemma 5.** Let $A_1, A_2$, and $B$ be matrices in $\mathcal{P}_r$ and suppose that $s(B)$ is odd. Let $D = (A_1|A_2)$ and $D' = (A_1B|A_2B)$, then $\text{RowSpan}(D) = \text{RowSpan}(D')$.

**Proof.** Observe that

$$D' = (A_1B|A_2B) = (BA_1|BA_2) = B(A_1|A_2) = BD,$$

$$D = (A_1|A_2) = (BA_1B|BA_2B) = B(A_1B|A_2B) = BD'.$$

The first of these relations implies that $\text{RowSpan}(D') \subseteq \text{RowSpan}(D)$, and the second implies the reverse inclusion. □

It is clear that Lemma 5 holds even if the matrices $D, D'$ have the form

$$D = (A_1|A_2|\ldots|A_t), \quad D' = (A_1B|A_2B|\ldots|A_tB),$$

where $A_i \in \mathcal{P}_r, i \in [t]$ and where $s(B)$ is odd.

**Lemma 6.** The action of $\mathcal{M}_r$ on the set $\mathcal{P}_r$ partitions this set into equivalence classes.

**Proof.** Transitivity follows by the remark before Lemma 2 (or from the lemma itself): if $A_1 = \Gamma_{u}A_2$ and $A_2 = \Gamma_{v}A_3$ then $A_1 = \Gamma_{w}A_3$ for $w = u+v$. □

Denote by $[A]$ the equivalence class of $A \in \mathcal{P}_r$ and note that, since the action of $\mathcal{M}_r$ is faithful, $|[A]| = |\mathcal{M}_r| = 2^r$.

**Lemma 7.** Let $D_1, i = 1,\ldots,t$ and $A$ be matrices from $\mathcal{P}_r$ and let $D = (D_1|D_2|\ldots|D_t)$. If $D_i \in [A], i = 1,\ldots,t$ then $\text{rk}(A) = \text{rk}(D)$.

**Proof.** For simplicity, let $t = 2$ and let $D_1 = A\Gamma_u$ and $D_2 = A\Gamma_v$ for some $u, v \in F^r$. Clearly $\text{rk}(A\Gamma_u|A\Gamma_v) = \text{rk}(A\Gamma_u|A\Gamma_u)$ since $A\Gamma_u$ and $A\Gamma_v$ share the same column set. Next, we have

$$\text{rk}(D) = \text{rk}(D_1|D_2) = \text{rk}(A\Gamma_u|A\Gamma_v)$$

$$= \text{rk}(A\Gamma_u|A\Gamma_u) = \text{rk}(\Gamma_uA|\Gamma_uA) = \text{rk}(\Gamma_u(A|A)) = \text{rk}(A).$$

□
4. Partitioning the adjacency matrix

Suppose we are given a small code \( C \in F^n \), and let \( S = \{h_1, \ldots, h_n\} \) be the set of columns of its parity-check matrix.

**Lemma 8.** Let \( A \) be the adjacency matrix of the graph \( G = \text{Cay}(F^r, S) \), where \( S \) may or may not include \( 0 \), then

\[
A = \Gamma_{h_1} + \Gamma_{h_2} + \cdots + \Gamma_{h_n}.
\]

**Proof.** Let \( v \in F^r \) be a vertex in \( G \). The nonzero entries in the row \( A_v \) correspond to the vectors (columns) \( w \) such that \( w = v + h \) for some \( h \in S \). Each of these entries appears in row \( v \) of the matrix \( \Gamma_h \), and the sum of those rows forms row \( v \) in \( A \). \( \square \)

Our aim at this point is to operate on submatrices of \( A \), transforming it to a more convenient form while preserving the rank. With this in mind, we will define permutations derived from the matrix \( H \) but acting on subspaces of \( F^r \).

For integers \( s, t, 1 \leq s \leq t \leq r \) and a vector \( x = (x_1, \ldots, x_r) \in F^r \) let\( \text{x}^{(s,t)} = (x_s, x_{s+1}, \ldots, x_t) \). Given \( \ell \leq r \) and a vector \( u \in F^\ell \), let \( S_u \) be the set of suffixes of the columns whose prefix is \( u \):

\[
S_u = \{ h^{(\ell+1,r)} | h \in S, h^{(1,\ell)} = u \}.
\]

Finally, consider permutations \( \Gamma_v, v \in F^{r-\ell} \) acting on \( F^{r-\ell} \) by adding vectors from \( S_u \).

For a given \( u \) define the matrix

\[
D_u = \sum_{v \in S_u} \Gamma_v.
\]

Assume by definition that if \( S_u = \emptyset \) then \( D_u = 0_{2^{r-\ell} \times 2^{r-\ell}} \).

To give an example, let \( r = 3, \ell = 1 \), and take

\[
H = \begin{bmatrix}
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

then \( S_0 = \{00, 10\} \), \( S_1 = \{00, 11\} \) and

\[
D_0 = I + \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}, \quad D_1 = I + \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix},
\]

where we assumed that the rows and columns of the matrices are indexed by \( F^2 \) in lexicographical order.

Now let us look at how the matrices defined in (2) relate to the adjacency matrix of the graph \( G \). For \( \ell \in \{1, 2, \ldots, r\} \), let \( F^\ell = \{v_0, v_1, \ldots, v_{2^\ell-1}\} \) where the vectors are ordered lexicographically. Starting with a small code \( C \) with a fixed parity-check matrix \( H \), let us form the matrices \( D_{v_0}, D_{v_1}, \ldots, D_{v_{2^\ell-1}} \).
Lemma 9. The matrix $A$ can be written in the form

$$A = \begin{pmatrix} v_0 & v_1 & \cdots & v_{2^\ell-1} \\ v_0 + v_0 & v_0 + v_1 & \cdots & v_0 + v_{2^\ell-1} \\ v_1 + v_0 & v_1 + v_1 & \cdots & v_1 + v_{2^\ell-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{2^\ell-1} + v_0 & v_{2^\ell-1} + v_1 & \cdots & v_{2^\ell-1} + v_{2^\ell-1} \end{pmatrix},$$

where the $2^{r-\ell} \times 2^{r-\ell}$ blocks $D_{v_i+v_j}$ are defined in (2).

Proof. Given $x, y \in F^r$, we have $A_{x,y} = \mathbb{1}(x + h = y)$ for some $h \in S$. Let $v_i = x^{(1,\ell)}$ and $v_j = y^{(1,\ell)}$ be the $\ell$-prefixes of $x$ and $y$ and let $D_{v_i,v_j}$ be the block in $A$ at the intersection of the stripes $v_i$ and $v_j$. Our goal is to show that $D_{v_i,v_j} = D_{v_i+v_j}$.

Let $S_{v_i+v_j}$ be the set defined in (1). Given $t, u \in F^{r-\ell}$, the element

$$(D_{v_i,v_j})_{t,u} = \mathbb{1}(t + h' = u),$$

where $h' \in S_{v_i+v_j}$ is an $(\ell - r)$ tail vector. Rephrasing and using (2),

$$D_{v_i,v_j} = \sum_{h' \in S_{v_i+v_j}} \Gamma_h' = D_{v_i+v_j}.$$

\[\square\]

4.1. Zero-codeword-only codes and repetition codes. As an example of using the above approach, consider the adjacency matrices of coset graphs of the zero-codeword-only codes and repetition codes. Their ranks are known [4, 2], but we rederive them using the tools developed in the previous sections. Let $J = \Gamma_1$ be the matrix defined as $J_{u,v} = \mathbb{1}(u + v = 1_r)$, where $1_r$ denotes the all-ones vector of length $r$. Since we assumed that the vectors in $F^r$ are ordered lexicographically, $J$ is antidiagonal.

Proposition 10. [4, Prop.9] Let $C = \{0_r\}$ be the zero-codeword code with $S$ given by the standard basis $e_i, i = 1, \ldots, r$. The adjacency matrix $A_r$ of the coset graph $\text{Cay}(F^r, S)$ has rank $2^r$ if $r$ is even and $2^{r-1}$ if $r$ is odd.

Proof. If $r$ is odd, then $A_r = \sum_{i=1}^r \Gamma e_i$ has full rank by Lemma 4(1). Let us consider the case of $r$ even, writing the matrix $C$ as in Lemma 9. Let $\ell = 1$ and note that $D_0 = A_{r-1}$ and $D_1 = I_{r-1}$, both of rank $r - 1$. Then

$$A_r = \begin{pmatrix} A_{r-1} & I \\ I & A_{r-1} \end{pmatrix}.$$

Since $s(A_{r-1})$ is odd, Lemma 5 implies that we can multiply the upper stripe by $A_{r-1}$ block-by-block without affecting the row space in this part, thus with no effect on the rank. Upon multiplying, we obtain

$$\begin{pmatrix} I & A_{r-1} \\ I & A_{r-1} \end{pmatrix},$$

which obviously is of rank $2^{r-1}$.

\[\square\]
We could of course simply eliminate all the entries in one of the submatrices \( A_{r-1} \) by row operations, but the above procedure models our approach in other constructions. We next exemplify it in a more complicated case of repetition codes. Let \( C' \) be the repetition code of length \( r+1 \) and redundancy \( r \) defined by the parity-check matrix \( H' = [I|1_r] \), where \( I \) is the identity matrix of order \( r \). Form the matrix \( H = [H'|0_r] \) and consider the code \( C \) of length \( r+2 \) for which \( H \) is a parity-check matrix. In the next lemma, we compute the rank of the adjacency matrix of the coset graph of \( C \).

**Proposition 11.** Let \( S = \{e_1, \ldots, e_r, 1, 0_r\} \). The adjacency matrix \( A_r \) of the coset graph \( \text{Cay}(F^r, S) \) satisfies

\[
\text{rk}(A_r) = \begin{cases} 
2^r & \text{if } r \text{ is odd} \\
\frac{1}{2}(2^r - 2^r) & \text{if } r \text{ is even} 
\end{cases}
\]

**Proof.** If \( r \) is odd then \( s(A_r) \) is odd, and by Lemma 4(1) \( A_r \) is a full-rank matrix, i.e., \( \text{rk}(A_r) = 2^r \). For even \( r \) we prove the result by induction on \( r \). Take \( r = 2 \), then \( H_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \). Since \( S = F^2 \), the matrix \( A_2 \) is an all-ones matrix of rank 1, verifying the base case.

Let us assume that \( \text{rk}(A_{r-2}) = \frac{1}{2}(2^r - 2^r) \) and let us consider the matrix \( A_r \). We decompose it into blocks taking \( \ell = 2 \) in the construction of Lemma 9. Observe that

\[
D_{(0,0)} = A_{r-2} + J, \quad D_{(0,1)} = I, \quad D_{(1,0)} = I \text{ and } D_{(1,1)} = J.
\]

By Lemma 9 we obtain

\[
A_r = \begin{pmatrix}
I & A_{r-2} + J & I & J \\
I & A_{r-2} + J & J & I \\
J & I & A_{r-2} + J & I \\
J & I & I & A_{r-2} + J
\end{pmatrix}.
\]

Next multiply the top horizontal stripe by \( A_{r-2} + J \) and the bottom one by \( J \), then we obtain the following matrix:

\[
\begin{pmatrix}
I & A_{r-2} + J & A_{r-2} + J & A_{r-2} J + I \\
I & A_{r-2} + J & J & I \\
J & I & A_{r-2} + J & I \\
J & J & I & A_{r-2} J + I
\end{pmatrix}.
\]

Since \( s(A_{r-2} + J) \) is odd, by Lemma 5 this has no effect on the rank. Next, let us eliminate the bottom stripe adding to it the first three stripes row-by-row, and after that cancel two matrices in each of the two middle stripes. Overall we obtain

\[
\begin{pmatrix}
I & 0 & 0 & A_{r-2} J + I \\
0 & A_{r-2} + J & 0 & A_{r-2} J + I \\
0 & A_{r-2} + J & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}.
\]

Let \( D_1 = (A_{r-2}|A_{r-2} J) \) and \( D_2 = (A_{r-2} A_{r-2}) \). By Lemma 7, \( \text{rk}(D_1) = \text{rk}(D_2) = \text{rk}(A_{r-2}) \). Therefore, considering the first three stripes independently, we obtain

\[
\text{rk}(A_r) = \frac{1}{4} \cdot 2^r + \frac{1}{2} \cdot (2^r - 2^r) + \frac{1}{2} \cdot (2^r - 2^r) + \frac{1}{2} \cdot (2^r - 2^r) = \frac{1}{2} \cdot (2^r - 2^r).
\]

\[\square\]
5. A new code family

The starting point of the construction is the binary Hamming code of length $2^r - 1, r \geq 2$. Let $\mathcal{H}_r$ denote its parity-check matrix written in the standard form in which all the $r$-columns are ordered lexicographically with $0_{r−1}|1$ on the left and $1_r$ on the right. Since $\mathcal{H}_r$ contains all the nonzero columns, its coset graph is a complete graph $K_{2^r}$.

Lemma 12. Consider the matrix $(\mathcal{H}_r^T [0_{(2^r−1)\times m}])^T, m \geq 1$ and let $S$ be the set of its columns. Let $G = \text{Cay}(F^{r+m}, S)$, then $\text{rk}(\tilde{A}(G)) = 2^m$.

Proof. We give two proofs, of which the second paves the way for later results in this section.

There are $2^r$ vertices $v \in F^{r+m}$ that share a common $m$-suffix. They form a clique $K_{2^r}$, which is a connected component of the graph $G$. Thus $\tilde{A}(G)$ is a $2^m \times 2^m$ block-diagonal matrix with each block of rank 1, and so its rank is $2^m$.

Alternatively, with Lemma 9 in mind, let $\ell = r$ and note that for any fixed $r$-prefix $u$ the set $S_u = \{0_m\}$. Thus the matrix $\tilde{A}(G)$ can be written as a $2^r \times 2^r$ block matrix with each block equal to $I_{2^m}$, confirming again that its rank is $2^m$. \qed

The approach taken in [2], as well as in our work, is to add rows and columns to $\mathcal{H}_r$ in order to remove the codewords of weight 3 in the small code, while keeping the rank $\text{rk}(\tilde{A})$ low. The parity-check matrix $H_r$ of the small code is formed of $s$ rows at the top and some combinations of the matrices $\mathcal{H}_r$ underneath them. The resulting big codes are denoted by $\mathcal{C}_{s,r}$. The construction is recursive, starting with $s = 2$ (the base case) and adding one extra row to the top part in each step.

5.1. The case of $s = 2$. Consider the following $(r+2) \times (2^r+2)$ parity-check matrix of the small code:

$$H_2 = \begin{pmatrix}
0 & 0 & 0 & \ldots & 0 & 1 & 1 \\
0 & 1 & 1 & \ldots & 1 & 0 & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 1 & 1 \\
\mathcal{H}_r & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 1 & 1
\end{pmatrix}.$$ (3)

By inspection, once we discount the first column, the new matrix has rank 4, so its coset graph is triangle-free. The family of big codes $\mathcal{C}_{2,r}$ obtained from $H_2$ was originally discovered in [2] (up to a minor difference that is not essential for the results). As shown there, the rate of the storage code is $R(\mathcal{C}_{2,r}) = 3/4 - 2^{-r}$ for all $r \geq 4$. While the argument in [2] is somewhat complicated, here we give a straightforward proof based on the decomposition of Lemma 9 with $\ell = 2$.

Proposition 13. Let $S$ be the set of columns of the matrix $H_2$ and let $G_2 := \text{Cay}(F^r, S)$ be the coset graph of the code ker($H_2$). Then $\text{rk}(A(G_2)) \leq 2^{r+2}/4 + 4$.

Proof. Let $P$ be the adjacency matrix of $K_{2^r}$. We partition the adjacency matrix of $G_2$ into blocks of order $2^r$ according to the 2-prefix of the generator. The result can be written in the form

$$A(G_2) = \begin{pmatrix}
I & P + I & I & I \\
P + I & I & I & I \\
I & I & I & P + I \\
I & I & P + I & I
\end{pmatrix} = \begin{pmatrix}
I & I & I \\
I & I & I \\
I & I & I \\
I & I & I
\end{pmatrix} + \begin{pmatrix}
0 & P & 0 & P \\
P & 0 & 0 & 0 \\
0 & 0 & 0 & P \\
0 & 0 & P & 0
\end{pmatrix},$$
where the locations of $P$ correspond to $D_{01}$; see (2) and Lemma 9. Since $\text{rk}(P) = 1$, the result follows. □

**Corollary 14.** The rate of the storage code is

$$R(\mathcal{C}_{2,r}) \geq 1 - \text{rk}(A)/2^{r+2} = \frac{3}{4} - 2^{-(r+4)} \to \frac{3}{4}$$

as $r \to \infty$.

Next we increase the dimensions of the matrix. We show this procedure in detail for $s = 3$ and then state and prove the general claim.

5.2. The case of $s = 3$. Form the matrix $(H_2|H_2)$ and

(1) add a new row $0_{2r+2}1_{2r+2}$ at the top of this matrix.

We obtain an $(r+3) \times (2r+1+4)$ matrix of the form

$$
\begin{pmatrix}
0 & 0 & 0 & \ldots & 0 & 0 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 \\
0 & 0 & 0 & \ldots & 0 & 1 & 1 & 0 & 0 & 0 & \ldots & 0 & 1 & 1 \\
0 & 1 & 1 & \ldots & 1 & 0 & 1 & 0 & 1 & 1 & \ldots & 1 & 0 & 1 \\
\vdots & & & & & \mathcal{H}_r & \vdots & \vdots & \mathcal{H}_r & \vdots & \\
0 & 0 & 0 & 0 & & & 0 & 0 & & & & & 0 & 0
\end{pmatrix}.
$$

(4)

This matrix contains linearly dependent triples that do not include the first column. To take care of them,

(2) add $r$ rows of zeros at the bottom,

(3) replicate the column $1|0_{2r+2}$ immediately to the right of the vertical divider $2^r - 1$ times,

(4) replace the $r \times (2^r - 1)$ matrix of zeros at the bottom of the new columns with $\mathcal{H}_r$.

The resulting $(2r+3) \times (3 \cdot 2^r + 2)$ matrix $H_3$ is shown below in Eq.(5), with boxes around the rows and columns formed in steps (2)-(4).

$$
H_3 =
\begin{pmatrix}
0 & 0 & 0 & \ldots & 0 & 0 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 \\
0 & 0 & 0 & \ldots & 0 & 1 & 1 & 0 & 0 & 0 & \ldots & 0 & 1 & 1 \\
0 & 1 & 1 & \ldots & 1 & 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 1 & 1 \\
\vdots & & & & \mathcal{H}_r & \vdots & \vdots & 0 & \mathcal{H}_r & \vdots & \\
0 & 0 & 0 & & & & & & & & & & & \mathcal{H}_r & 0 & \vdots & \vdots \\
\vdots & 0 & & \vdots & \mathcal{H}_r & 0 & & & & & & & & \vdots & \vdots \\
0 & 0 & 0 & 0 & & & & & & & & & & & 0 & 0
\end{pmatrix}.
$$

(5)

In the next proposition, we establish simple properties of the matrix $H_3$. Denote by $S$ the set of its columns.

**Proposition 15.** The graph $G_3 = \text{Cay}(F^{2r+3}, S)$ is connected and triangle-free.
Proof. Let \( S_i, i = 1, 2, 3 \) be the sets of columns that contain the first, second, and third Hamming matrices, respectively, and let \( S_4 \) be the remaining set of 4 nonzero columns. Let \( 0 \in F^{2r+3} \) and \( x \in F^{2r+3}, x \neq 0 \) be two group elements. To prove the first claim, it suffices to show that there is a path in \( G_3 \) that connects them. Suppose \( x = (x_1, x_2, x_3, x_4, \ldots, x_{2r+3}) \). Since \( S_4 \) contains a basis of \( F^3 \), any assignment \( x_1x_2x_3 \) of the first three coordinates in \( x \) can be reached independently of the remaining coordinates. Since the Hamming matrices contain all the nonzero columns, it is also possible to reach any vector of the form \( (000, x_4, \ldots, x_{2r+3}) \).

For the second claim, we need to show that all triples of columns that do not include the zero column are linearly independent. This is shown by a straightforward case study. Let \( h_1, h_2, h_3 \in \{0\} \) be three such columns and let \( b = |\{h_1, h_2, h_3\} \cap S_4| \). If \( b \in \{1, 2, 3\} \) then the claim is obvious. If \( b = 0 \), then \( h_1, h_2, h_3 \) can be chosen to intersect one, two, or all three of the Hamming matrices. In each of these cases, direct inspection shows that the triples cannot add to zero. \( \square \)

Proposition 16. \( \text{rk}(A(G_3)) \leq 2^{2r} + 3 \cdot 2^{r+3} \).

Proof. Let \( P_1 \) be the adjacency matrix of the Cayley graph in \( F^r \) whose generators are the columns of the matrix \( (510_1^{(2r-1)\times 1})^T \) and \( 0_{2r} \), and let \( P_2 \) be the same for the matrix \( (0_{2r-1}51^{1\times 1})^T \) and \( 0_{2r} \). We arrange the matrix \( A(G_3) \) in a block form, where the blocks are indexed by binary 3-vectors (prefixes) ordered lexicographically. Using Lemma 9 for the matrix \( H_{3r} \), we obtain

\[
A(G_3) = \begin{pmatrix}
I & P_1 + I & I & I & P_2 + I & P_1 + I & I & I \\
P_1 + I & I & I & P_1 + I & I & P_2 + I & P_1 + I & I \\
I & I & P_1 + I & I & I & P_1 + I & P_2 + I & I \\
P_2 + I & P_1 + I & I & I & I & P_1 + I & I & I \\
P_1 + I & P_2 + I & I & I & P_1 + I & I & I & I \\
I & I & P_2 + I & P_1 + I & I & I & P_1 + I & I \\
I & I & P_1 + I & P_2 + I & I & I & P_1 + I & I \\
I & I & P_1 + I & P_2 + I & I & I & P_1 + I & I
\end{pmatrix}.
\]

The dimensions of this matrix are \( N \times N \), where \( N = 2^{2r+3} \), and each of the blocks is a square matrix of order \( 2^{2r} \). We can write \( A(G_3) \) as a sum of 4 block matrices. The first of them is the matrix \( (I)_{8\times 8} \) formed of identity blocks, and the other three are matrices with one matrix \( P_1 \) (\( P_2 \)) per row and per column. For instance, one such matrix is

\[
\begin{pmatrix}
0 & P_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & P_1 & 0 & 0 & 0 & 0 \\
0 & 0 & P_1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & P_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & P_1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & P_1 & 0
\end{pmatrix}.
\]

Now, Lemma 12 says that \( \text{rk}(P_1) = \text{rk}(P_2) = 2^r \), and thus the rank of \( A(G_3) \) is at most

\[
\text{rk}(A(G_3)) \leq 2^{2r} + 3 \cdot 8 \cdot 2^r = 2^{2r} + 3 \cdot 2^{r+3}.
\]

\( \square \)

Below we call matrices of the form (6) block permutation matrices.
Corollary 17. The rate of the storage code $C_{s,r} = \ker(A(G_s))$ is

$$R(C_{s,r}) \geq 1 - \frac{2^{2r} + 3 \cdot 2^{r+3}}{2^{2r+3}} = \frac{7}{8} - 3 \cdot 2^{-r}.$$  

The general induction step is not different from the transition from $s = 2$ to $3$. Namely, we form the matrix $H_s$ by performing steps (1)-(4) described above on the matrix $H_{s-1}$. As we will show shortly, this results in storage codes of rate $(2^s - 1)/2^s - o(1)$.

Following the construction procedure, we find that the matrix $H_s$ is formed of $s$ horizontal stripes, $s - 1$ of which contain zero matrices and several Hamming matrices $H_j$. The set of columns is formed of $2^{s-1} - 1$ vertical stripes each of which contains one Hamming matrix and $2^{s-1} + 1$ other columns counting the all-zeros column. Thus the dimensions of $H_s$ are

$$((s-1)r + s) \times ((2^{s-1} - 1)(2^s - 1) + 2^{s-1} + 1).$$

Accordingly, the vertex set of the graph $G_s$ is of size $N = 2^{(s-1)r+s}$. The $N \times N$ matrix $A(G_s)$ can be written as a $2^s \times 2^s$ block matrix with each block of size $2^{(s-1)r}$. The structure of the matrix is similar to $A(G_2)$ and $A(G_3)$: namely, it is a sum of several $2^s \times 2^s$ block matrices, one of which is formed of identity matrices only. The remaining matrices are block permutation matrices of the form (6), where in each of them, the block in question is the (augmented) adjacency matrix of the Cayley graph of the kind given in Lemma 12. Namely, the set of generators of this Cayley graph is the set of columns of the matrix

$$M_j := [0_{(2^s-1) \times (j-1)} | \delta_1^T | 0_{(2^s-1) \times ((s-2)-(j-1))}]^T,$$

for some $j = 0, 1, \ldots, s - 2$ (thus overall the matrices $M_j$ are of dimensions $((s-1)r) \times (2^s - 1)$). Lemma 12 implies for each of these Cayley graphs, the adjacency matrix is of rank $2^{(s-2)r}$. Since $2^s - 1$, the number of these permutation matrices equals the total number of Hamming matrices in $H_s$, i.e., $2^{s-1} - 1$. Collecting this information, we find that

$$\text{rk}(A(G_s)) \leq 2^{(s-1)r} + (2^{s-1} - 1)2^{(s-2)r+s} \leq N(2^{-s} + 2^{s-r-1}).$$

We obtain the following result.

Theorem 18. For any $s \geq 2$ there exists a sequence of storage codes $C_{s,r}$ on triangle-free graphs on $N = 2^{(s-1)r+s}$ vertices, $r = 4, 5, \ldots$ of rate

$$R(C_{s,r}) \geq \frac{2^s - 1}{2s} - 2^{s-r-1}.$$  

Proof. Since $R = 1 - \text{rk}(A(G_s))/N$, the claim about the rate follows from (7). □

We have constructed an infinite family of code sequences $C_{s,r}$, $r = 4, 5, \ldots; s = 2, 3, \ldots$ whose rates are given by (8). To form a sequence of codes $(\mathcal{C}_m)_m$ whose rate converges to one, let both $s, r \to \infty$ such that $s = o(r)$, for instance, $s = r^\alpha, \alpha < 1$. This proves Theorem 1.

Acknowledgment. This research was partially supported by NSF grants CCF2110113 (NSF-BSF) and CCF2104489, and by a German Israeli Project Cooperation (DIP) Grant under Grant PE2398/1-1.
REFERENCES

[1] F. Arbabjolfaei and Y.-H. Kim. Fundamentals of index coding. *Foundations and Trends in Comm. Inf. Theory*, 14(3-4):163–346, 2018. doi:10.1561/0100000094.

[2] A. Barg and G. Zémor. High-rate storage codes on triangle-free graphs. *IEEE Transactions on Information Theory*, 68(12):7787–7797, 2022. doi:10.1109/TIT.2022.3191309.

[3] P. J. Cameron, A. N. Dang, and S. Riis. Guessing games on triangle-free graphs. *Electron. J. Combin.*, 23(1):Paper 48, 2016. doi:10.37236/4731.

[4] A. Couvreur, N. Delfosse, and G. Zémor. A construction of quantum LDPC codes from Cayley graphs. *IEEE Trans. Inform. Theory*, 59(9):6087–6098, 2013. doi:10.1109/TIT.2013.2261116.

[5] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin. On the locality of codeword symbols. *IEEE Trans. Inform. Theory*, 58(11):6925–6934, 2011. doi:10.1109/TIT.2012.2208937.

[6] A. Mazumdar. Storage capacity of repairable networks. *IEEE Transactions on Information Theory*, 61(11):5810–5821, 2015. doi:10.1109/TIT.2015.2472521.

[7] A. Mazumdar, A. McGregor, and S. Vorotnikova. Storage capacity as an information-theoretic vertex cover and the index coding rate. *IEEE Trans. Inform. Theory*, 65(9):5580–5591, 2019. doi:10.1109/TIT.2019.2910026.

[8] V. Ramkumar, S.B. Balaji, B. Sasidharan, M. Vajha, M. Nikhil Krishnan, and P. V. Kumar. Codes for distributed storage. *Foundations and Trends in Communications and Information Theory*, 19:547–813, 2022. doi:10.1561/0100000115.

[9] S. Riis. Information flows, graphs and their guessing numbers. *Electron. J. Combin.*, 14(1):17pp., 2007. doi:10.37236/962.

[10] K. Shanmugam and A. G. Dimakis. Bounding multiple unicasts through index coding and locally repairable codes. In *2014 IEEE International Symposium on Information Theory*, pages 296–300, 2014. doi:10.1109/ISIT.2014.6874842.

Department of ECE and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

Email address: abarg@umd.edu

School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 8410501, Israel

Email address: moshesch@bgu.ac.il

School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 8410501, Israel, and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

Email address: levuyhananov@gmail.com