Which Codes Have Cycle-Free Tanner Graphs?

Tuvi Etzion
Technion — Israel Institute of Technology
Department of Computer Science
Haifa 32000, Israel

Ari Trachtenberg
Digital Computer Laboratory
University of Illinois at Urbana-Champaign
1304 W. Springfield, Urbana, IL 61801

Alexander Vardy
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main Street, Urbana, IL 61801

October 28, 1998

Abstract

If a linear block code $\mathbb{C}$ of length n has a Tanner graph without cycles, then maximum-likelihood soft-decision decoding of $\mathbb{C}$ can be achieved in time $O(n^2)$. However, we show that cycle-free Tanner graphs cannot support good codes. Specifically, let $\mathbb{C}$ be an (n, k, d) linear code of rate $R = k/n$ that can be represented by a Tanner graph without cycles. We prove that if $R \geq 0.5$ then $d \leq 2$, while if $R < 0.5$ then $\mathbb{C}$ is obtained from a code of rate ≥ 0.5 and distance ≤ 2 by simply repeating certain symbols. In the latter case, we prove that

$$d \;\leq\; \left\lfloor \frac{n}{k+1} \right\rfloor + \left\lfloor \frac{n+1}{k+1} \right\rfloor \;<\; \frac{2}{R}$$

Furthermore, we show by means of an explicit construction that this bound is tight for all values of n and k. We also prove that binary codes which have cycle-free Tanner graphs belong to the class of graph-theoretic codes, known as cut-set codes of a graph. Finally, we discuss the asymptotics for Tanner graphs *with* cycles, and present a number of open problems for future research.

Keywords: Linear codes, Tanner graphs, minimum distance, iterative decoding

*A version of this paper appeared in *IEEE Trans. Inf. Theory*, 45:6, pp. 2173, September 1999.

1. Introduction

Iterative decoding algorithms on factor graphs [11] have become a subject of much active research in recent years [1, 2, 4, 5, 9, 11, 16, 17, 18, 22, 29, 30]. For example, the well-known turbo codes and turbo decoding methods [5, 4] constitute a special case of this general approach to the decoding problem. Factor-graph representations for turbo codes were introduced in [29, 30], where it is also shown that turbo decoding is an instance of a general decoding procedure, known as the sum-product algorithm. Another extensively studied [8, 27] special case is trellis decoding of block and convolutional codes. It is shown in [9, 30] that the Viterbi algorithm on a trellis is an instance of the min-sum iterative decoding procedure, when applied to a simple factor graph. The forward-backward algorithm on a trellis, due to Bahl, Cocke, Jelinek, and Raviv [3], is again a special case of the sum-product decoding algorithm. More general iterative algorithms on factor graphs, collectively termed the "generalized distributive law" or GDL, were studied by Aji and McEliece [1, 2]. These algorithms encompass maximum-likelihood decoding, belief propagation in Bayesian networks [10, 20], and fast Fourier transforms as special cases.

It is proved in [2, 11, 26, 30] that the min-sum, the sum-product, the GDL, and other versions of iterative decoding on factor graphs all converge to the optimal solution if the underlying factor graph is cycle-free. If the underlying factor graph has cycles, very little is known regarding the convergence of iterative decoding methods.

This work is concerned with an important special type of factor graphs, known as Tanner* graphs. The subject dates back to the work of Gallager [12] on low-density parity-check codes in 1962. Tanner [26] extended the approach of Gallager [12, 13] to codes defined by general bipartite graphs, with the two types of vertices representing code symbols and checks (or constraints), respectively. He also introduced the min-sum and the sum-product algorithms, and proved that they converge on cycle-free graphs. More recently, codes defined on sparse (regular) Tanner graphs were studied by Spielman [22, 25], who showed that such codes become asymptotically good if the underlying Tanner graph is a sufficiently strong expander. These codes were studied in a different context by MacKay and Neal [16, 18], who demonstrated by extensive experimentation that iterative decoding on Tanner graphs can approach channel capacity to within about 1 dB. Latest variants [17] of these codes come within about 0.3 db from capacity, and outperform turbo codes.

In general, a Tanner graph for a code $\mathbb{C}$ of length n over an alphabet A is a pair $(\mathcal{G}, \mathcal{L})$, where $\mathcal{G} = (V, E)$ is a bipartite graph and $\mathcal{L} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ is a set of codes over A, called *behaviors* or *constraints*. We denote the two vertex classes of $\mathcal{G}$ by $\mathcal{X}$ and $\mathcal{Y}$, so that $V = \mathcal{X} \cup \mathcal{Y}$. The vertices of $\mathcal{X}$ are called *symbol vertices* and $|\mathcal{X}| = n$, while the vertices

***Note on terminology.** The term *Tanner graph* was first used by Wiberg, Loeliger, and Kötter [30] to refer to the more general graphs introduced in [30]. These were later termed TWL graphs by Forney [9], although *TWLK graphs* would have been more appropriate. By now, the term *factor graphs* is almost universally used in this context, which leaves *Tanner graphs* available to refer to the kind of factor graphs actually studied by Tanner [26]. The emphasis in this paper (as in all of the literature [7, 16, 18, 22, 26] on the subject) is on a special type of Tanner graphs that come with simple parity-check constraints. These Tanner graphs include the graphs underlying Gallager's low-density parity-check codes [12, 13].

of $\mathcal{Y}$ are called *check vertices* and $|\mathcal{Y}| = r$. There is a one-to-one correspondence between the constraints $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r$ in $\mathcal{L}$ and the check vertices $y_1, y_2, \ldots, y_r$ in $\mathcal{Y}$, so that the length of the code $\mathcal{C}_i \in \mathcal{L}$ is equal to the degree of the vertex $y_i \in \mathcal{Y}$, for all $i = 1, 2, \ldots, r$. A *configuration* is an assignment of a value from A to each symbol vertex $x_1, x_2, \ldots, x_n$ in $\mathcal{X}$. Thus a configuration may be thought of as a vector of length n over A. Given a configuration $\chi = (\chi_1, \chi_2, \ldots, \chi_n)$ and a vertex $y \in \mathcal{Y}$ of degree δ, we define the *projection* χ_y of χ on y as a vector of length δ over A obtained from χ by retaining only those values that correspond to the symbol vertices adjacent to y. Specifically, if $\{x_{i_1}, x_{i_2}, \ldots, x_{i_\delta}\} \subseteq \mathcal{X}$ is the neighborhood of y in $\mathcal{G}$, then $\chi_y = (\chi_{i_1}, \chi_{i_2}, \ldots, \chi_{i_\delta})$. A configuration χ is said to be *valid* if all the constraints are satisfied, namely if $\chi_{y_i} \in \mathcal{C}_i$ for all $i = 1, 2, \ldots, r$. The code $\mathbb{C}$ represented by the Tanner graph $(\mathcal{G}, \mathcal{L})$ is then the set of all valid configurations.

While the foregoing definition of Tanner graphs is quite general, the theory and practice of the subject [7, 16, 17, 18, 22, 26] is focused almost exclusively on the simple special case where all the constraints are single-parity-check codes over $\mathbb{F}_2$. This work is no exception, although we will provide for the representation of linear codes over arbitrary fields by considering the zero-sum codes over $\mathbb{F}_q$ rather than the binary single-parity-check codes. It seems appropriate to call the corresponding Tanner graphs *simple*. Notice that in the case of simple Tanner graphs, the set of constraints $\mathcal{L}$ is implied by definition, so that one can identify a simple Tanner graph with the underlying bipartite graph $\mathcal{G}$. All of the Tanner graphs considered in this correspondence, except in Section 5.3, are simple. Thus, for the sake of brevity, we will henceforth omit the quantifier "simple." Instead, when we consider the general case in Section 5.3, we will use the term *general* Tanner graphs.

We can think of a (simple) Tanner graph for a binary linear code $\mathbb{C}$ of length n as follows. Let H be an $r \times n$ parity-check matrix for $\mathbb{C}$. Then the corresponding Tanner graph for $\mathbb{C}$ is simply the bipartite graph having H as its $\mathcal{X}, \mathcal{Y}$ adjacency matrix. It follows that the number of edges in any Tanner graph for a linear code $\mathbb{C}$ of length n is $O(n^2)$. Thus, if we can represent $\mathbb{C}$ by a Tanner graph *without cycles*, then maximum-likelihood decoding of $\mathbb{C}$ can be achieved in time $O(n^2)$, using the min-sum algorithm for instance.

However, both intuition and experimentation (cf. [16]) suggest that powerful codes cannot be represented by cycle-free Tanner graphs. The notion that cycle-free Tanner graphs can support only weak codes is, by now, widely accepted. Our goal in this correspondence is to make this "folk knowledge" more precise. We provide rigorous answers to the question: Which codes can have cycle-free Tanner graphs?

Our results in this regard are two-fold: we derive a characterization of the structure of such codes and an upper bound on their minimum distance. The upper bound (Theorem 5) shows that codes with cycle-free Tanner graphs provide extremely poor trade-off between rate and distance for each fixed length. This indicates that at very high signal-to-noise ratios these codes will perform badly. In general, however, the minimum distance of a code does not necessarily determine its performance at signal-to-noise ratios of practical interest. Indeed, there exist codes — for example, the turbo codes of [4, 5] — that have low minimum distance, and yet perform very well at low signal-to-noise ratios. The development of analytic bounds on the *performance* of cycle-free Tanner graphs under iterative decoding is a challenging problem, which is beyond the scope of this work.

Nevertheless, our results on the *structure* of the corresponding codes indicate that they are very likely to be weak: their parity-check matrix is much too sparse to allow for a reasonable performance even at low signal-to-noise ratios.

The rest of this paper is organized as follows. We start with some definitions and auxiliary observations in the next section. In Section 3, we show that if an (n, k, d) linear code $\mathbb{C}$ can be represented by a cycle-free Tanner graph and has rate $R = k/n \geq 0.5$, then $d \leq 2$. We furthermore prove that if $R < 0.5$, then $\mathbb{C}$ is necessarily obtained from a code of rate ≥ 0.5 and minimum distance ≤ 2 by simply repeating certain symbols in each codeword. Theorem 5 of Section 4 constitutes our main result: this theorem gives an upper bound on the minimum distance of a general linear code that can be represented by a cycle-free Tanner graph. Furthermore, the bound of Theorem 5 is exact. This is also proved in Section 4 by means of an explicit construction of a family of (n, k, d) linear codes that attain the bound of Theorem 5 for all values of n and k. Asymptotically, for $n \to \infty$, the upper bound takes the form:

$$d \lesssim 2\lfloor 1/R \rfloor \tag{1}$$

and an immediate consequence of (1) is that asymptotically good codes with cycle-free Tanner graphs do not exist. We show in Section 5 that the same is true for Tanner graphs *with* cycles, unless the number of cycles increases exponentially with the length of the code. We also show in Section 5 that for every binary code $\mathbb{C}$ that can be represented by a cycle-free Tanner graph, there exists a graph $\mathcal{G}$ such that $\mathbb{C}$ is the dual of the cycle code of $\mathcal{G}$. This establishes an interesting connection between codes with cycle-free Tanner graphs and the well-known [6, 15, 14, 21, 24] class of graph-theoretic *cut-set* codes. Finally, we conclude this paper with a partial analysis of general Tanner graphs.

2. Preliminaries

Let $H = [h_{ij}]$ be an $r \times n$ matrix, with entries drawn from the finite field $\mathbb{F}_q$ of order q. We let $*$ denote a nonzero entry in H. Given H, we define a bipartite graph T as follows: the vertex set of T consists of the set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ of symbol vertices and the set $\mathcal{Y} = \{y_1, y_2, \ldots, y_r\}$ of check vertices; there is an edge (y_i, x_j) in T if and only if $h_{ij} = *$. Thus the neighborhood of the vertex $y_i \in \mathcal{Y}$ corresponds to the i-th row of H, and the neighborhood of the vertex $x_j \in \mathcal{X}$ corresponds to the j-th column of H. We say that T is the Tanner graph of H, and denote $T = T(H)$. It is obvious that every matrix defines a unique Tanner graph. Over $\mathbb{F}_2$, the converse is also true: every bipartite graph T defines a unique binary matrix H such that $T = T(H)$, which is the $\mathcal{X}, \mathcal{Y}$ adjacency matrix of T.

We say that a bipartite graph T *represents* the linear code $\mathbb{C}$, or simply that T is a Tanner graph for $\mathbb{C}$, if there exists a parity-check matrix H for $\mathbb{C}$ such that T is the Tanner graph of H. In general, a given linear code can be represented by many distinct Tanner graphs. On the other hand, over $\mathbb{F}_2$, a given Tanner graph represents a unique binary code.

We say that a matrix H is *cycle-free* if the corresponding Tanner graph $T(H)$ is cycle-free. Notice that every submatrix of a cycle-free matrix is also cycle-free. We say that a linear code $\mathbb{C}$ over $\mathbb{F}_q$ is *cycle-free* if there *exists* a cycle-free parity-check matrix for $\mathbb{C}$.

Observe that if the matrices H and H' differ by a permutation of rows and columns then the Tanner graphs $T(H)$ and $T(H')$ are isomorphic. On the other hand, if H and H' differ by a sequence of elementary row operations then $T(H)$ and $T(H')$ are generally not isomorphic. Thus it is possible to have two parity-check matrices for the same code, one of which is cycle-free while the other is not. It is also possible to have two cycle-free Tanner graphs for the same code that are not isomorphic.

Example. Suppose that a parity-check matrix H for the $(5,2,3)$ binary linear code $\mathbb{C}$ is given by

$$H \;=\; \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The corresponding Tanner graph $T(H)$ is shown in Figure 1a. Notice that H is not cycle-free, since the sequence of edges $(x_1,y_1),(y_1,x_3),(x_3,y_2),(y_2,x_1)$ constitutes a cycle.

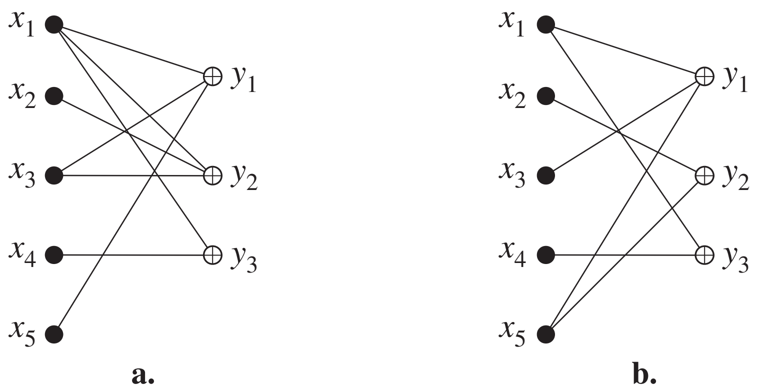


Figure 1. *Tanner graph with cycles and cycle-free Tanner graph for the same code*

However, the code $\mathbb{C}$ is, in fact, cycle-free since adding the first row of H to the second row produces a cycle-free parity-check matrix H' for $\mathbb{C}$. The graph $T(H')$ shown in Figure 1b is a cycle-free Tanner graph for $\mathbb{C}$. □

The following simple lemma will serve as our starting point. This lemma is well-known in graph theory – see, for instance, West [28, p. 52] – and we omit the proof.

Lemma 1. *A graph $\mathcal{G} = (V,E)$ is cycle-free if and only if $|E| = |V| - \omega(\mathcal{G})$, where $\omega(\mathcal{G})$ denotes the number of connected components in $\mathcal{G}$.*

A cycle-free graph consisting of a single connected component is called a tree, and thus a multiple-component cycle-free graph is also known as a forest. For trees, we have $|E| = |V| - 1$ by Lemma 1. Since every forest contains at least one tree, we have

$$|E| \;\le\; |V| - 1 \tag{2}$$

for any cycle-free graph. If M is an $m \times n$ matrix, then the number of vertices in $T(M)$ is $m+n$ and the number of edges in $T(M)$ is equal to $\mathrm{wt}(M)$ — the total number of nonzero entries in M. Thus if M is cycle-free, then $\mathrm{wt}(M) \le m+n-1$ in view of (2).

3. The structure of cycle-free codes

We start with a simple theorem, which gives a tight upper bound on the minimum distance of high-rate cycle-free linear codes.

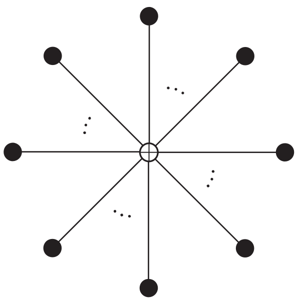
Theorem 2. *Let $\mathbb{C}$ be an (n,k,d) cycle-free linear code of rate $k/n \geq 0.5$. Then $d \leq 2$.*

Proof. Let H be the $r \times n$ cycle-free parity-check matrix for $\mathbb{C}$. We assume w.l.o.g. that H has full row-rank and $r = n-k$, since otherwise we can remove the linearly dependent rows of H while preserving the cycle-free property. Let η_i denote the number of columns of weight i in H. If $\eta_0 \neq 0$ then $d = 1$, and we are done. Otherwise, we have

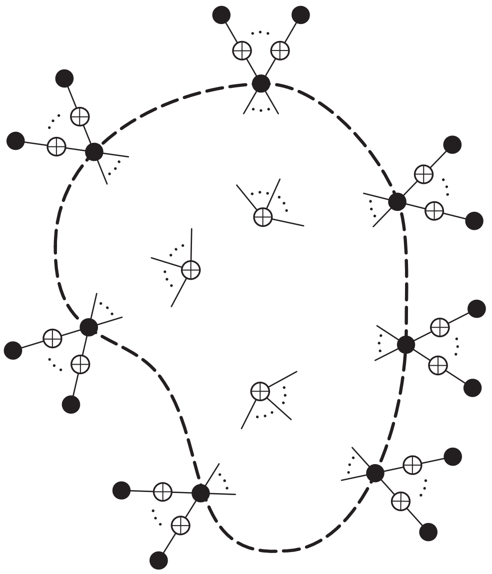
$$\eta_1 + 2(n-\eta_1) \;=\; \eta_1 + 2(\eta_2+\eta_3+\cdots+\eta_r) \;\leq\; \mathrm{wt}(H) \;\leq\; n+r-1 \qquad (3)$$

in view of (2). Substituting $r = n-k$ into (3), this inequality readily reduces to $\eta_1 \geq k+1$. Since $k/n \geq 0.5$, it follows that $k \geq r$ and $\eta_1 \geq r+1$. This means that the number of weight-one columns in H is greater than the number of rows in H. Hence H contains at least two columns of weight one that are scalar multiples of each other, and $d = 2$. ∎

Theorem 2 implies that the $(n,n-1,2)$ single-parity-check code $\mathcal{E}_n$ is, in a sense, the optimal cycle-free code of rate ≥ 0.5, since all such codes have distance $d \leq 2$ and $\mathcal{E}_n$ has the highest rate. The cycle-free Tanner graph for $\mathcal{E}_n$ is depicted in Figure 2a.



a.



b.

Figure 2. *Tanner graphs for $\mathcal{E}_n$ and for a general low-rate cycle-free code*

To show that the bound of Theorem 2 is tight for all n and k, with $n/2 \leq k \leq n-1$, we may start with the single-parity-check code $\mathcal{E}_{k+1}$ and repeat any symbol (or symbols) in $\mathcal{E}_{k+1}$ until a code of length n is obtained. The following lemma shows that this always produces an $(n, k, 2)$ cycle-free code for $k \geq n/2$.

Lemma 3. *Let $\mathbb{C}$ be a cycle-free code of length n and dimension k. Fix a positive integer i, with $i \leq n$, and let $\mathbb{C}'$ be the code obtained from $\mathbb{C}$ by repeating the i-th symbol in each codeword. Then $\mathbb{C}'$ is a cycle-free code of length $n+1$ and dimension k.*

Proof. The length and dimension of $\mathbb{C}'$ are obvious. To see that $\mathbb{C}'$ is cycle-free, observe that a Tanner graph T' for $\mathbb{C}'$ can be obtained from the cycle-free Tanner graph T for $\mathbb{C}$ by introducing two new vertices x' and y' and two new edges: (x', y') and (x_i, y'). It is easy to see that this procedure does not create new cycles. ∎

Let $\mathbb{C}$ be a cycle-free code of length n, and let $\mathbb{C}^*$ be the code of length $n + \delta$ obtained from $\mathbb{C}$ by iteratively applying δ times the procedure of Lemma 3, while possibly choosing a different value of i at different iterations. We then say that $\mathbb{C}^*$ is a code obtained by *repeating symbols* in $\mathbb{C}$. To make our terminology precise, we further extend the notion of codes obtained by "repeating symbols in $\mathbb{C}$" to also include the codes obtained from $\mathbb{C}^*$ by appending all-zero coordinates. The following proposition shows that *every* low-rate cycle-free linear code has this structure.

Proposition 4. *Let $\mathbb{C}$ be an (n, k, d) cycle-free linear code over $\mathbb{F}_q$ of rate $k/n \leq 0.5$. Then, up to scaling by constants in $\mathbb{F}_q$ at certain positions, $\mathbb{C}$ is obtained by repeating symbols in a cycle-free code of rate > 0.5.*

Proof. Let H be the $r \times n$ cycle-free parity-check matrix for $\mathbb{C}$, with $r = n - k$. Then by Lemma 1 and (2), we have $\mathrm{wt}(H) \leq n + r - 1 \leq 3r - 1$, where the second inequality follows from the fact that $k/n \leq 0.5$. This implies that H contains at least one row of weight ≤ 2. If this row is of weight one, then the corresponding coordinate of $\mathbb{C}$, say the n-th coordinate, is all-zero. Otherwise, assume without loss of generality that this row is of the form $h = (0, 0, \ldots, 0, *, *)$. Then, up to scaling the last two columns of H by constants in $\mathbb{F}_q$, we may further assume that $h = (0, 0, \ldots, 0, 1, -1)$. This would mean that the n-th symbol in $\mathbb{C}$ is a repetition of the preceding symbol. In both cases, we can puncture-out the n-th coordinate of $\mathbb{C}$, and iteratively repeat the argument until a cycle-free code of rate > 0.5 is obtained. ∎

Loosely speaking, Proposition 4 implies that every cycle-free code $\mathbb{C}$ of rate ≤ 0.5 can be represented by a Tanner graph whose structure is shown in Figure 2b. The dashed line in Figure 2b encloses a cycle-free Tanner graph for a code $\mathbb{C}'$ of rate > 0.5 and distance ≤ 2. It follows that to establish a bound on the minimum distance of low-rate cycle-free codes, we need to determine an optimal choice for $\mathbb{C}'$ in Figure 2b and an optimal sequence of symbol repetitions. This problem is considered in detail in the next two sections.

Specifically, we will show in the next section that the single-parity-check code constitutes an optimal choice for $\mathbb{C}'$, and every symbol should be repeated equally often.

4. The minimum distance of cycle-free codes

The following theorem gives an upper bound on the minimum distance of cycle-free linear codes. Later in this section, we will show that this bound is exact for all values of n and k.

Theorem 5. *Let $\mathbb{C}$ be an (n,k,d) cycle-free linear code over $\mathbb{F}_q$. Then*

$$d \;\le\; \left\lfloor \frac{n}{k+1} \right\rfloor + \left\lfloor \frac{n+1}{k+1} \right\rfloor \tag{4}$$

Observe that for $k/n \ge 0.5$, the bound in (4) reduces to $d \le 2$. This simple special case was dealt with in Theorem 2. The proof of Theorem 5 for general n and k is considerably more involved. This proof will be presented in Section 4.2, after we establish a series of auxiliary lemmas in the next subsection.

4.1. Groundwork: auxiliary lemmas

For the sake of brevity, we will consider only binary codes, although our proof readily extends to codes over an arbitrary finite field. Furthermore, with a slight abuse of notation, we will not distinguish between equivalent codes: namely, given a parity-check matrix H for a code $\mathbb{C}$, we will often freely permute the columns of H while still referring to the resulting matrix as a parity-check matrix for $\mathbb{C}$.

Let $\mathbb{C}$ be an (n,k,d) cycle-free binary linear code, and let H be an $r \times n$ cycle-free parity-check matrix for $\mathbb{C}$, where $r = n-k$. We say that H is in *s-canonical form*, if this matrix has the following structure:

$$H \;=\; \left[\begin{array}{c|c} A & \mathbf{0} \\ \hline B & I_s \end{array}\right] \tag{5}$$

where all the rows of B have weight ≤ 1, and I_s is the $s \times s$ identity matrix, for some s in the range $0 \le s \le r$. Notice that if $s = 0$ then (5) reduces to $H = A$ (which means that every matrix is in 0-canonical form), while if $s = r$ then the corresponding canonical form is $H = [B\,|\,I_r]$. We will use the shorthand $H = A\|_s B$ to denote the s-canonical form in (5).

Lemma 6. *Let $H = A\|_s B$ be a cycle-free binary matrix in s-canonical form, and suppose that $s < r$. Then at least one of the following statements is true:*

- $\bullet$. *The matrix A contains a row of weight two or less;*
- $\diamond$. *The matrix A contains three identical columns of weight one;*
- $\star$. *The matrix A contains two identical columns of weight one, and furthermore the row of A which contains the nonzero entries of these two columns has weight three.*

Proof. Let $T(A)$ be the Tanner graph of A. Evidently $T(A)$ is a subgraph of $T(H)$, obtained by retaining only the first $n-s$ symbol vertices $x_1, x_2, \ldots, x_{n-s}$, the first $r-s$ check vertices $y_1, y_2, \ldots, y_{r-s}$, and all the edges between these vertices. Since $T(H)$ is cycle-free by assumption, so is $T(A)$. We now construct another graph $\mathcal{G}$, called the *row-graph* of A, whose vertex set $y_1, y_2, \ldots, y_{r-s}$ corresponds to the rows of A. The edge set of $\mathcal{G}$ is derived from the columns of A of weight ≥ 2, so that a column of weight w in A contributes $w-1$ edges to $\mathcal{G}$. An example illustrating the construction of the row-graph $\mathcal{G}$ for a 6×8 cycle-free matrix is depicted in Figure 3.

$$\begin{array}{cc} \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{array} & \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{array} \end{array}$$

a.

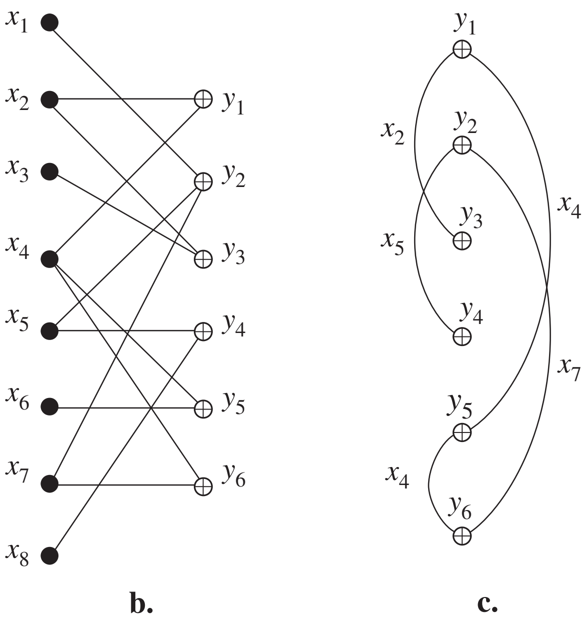


Figure 3. *A cycle-free matrix, its Tanner graph, and its row-graph $\mathcal{G}$*

Specifically, there is an edge between y_i and y_j in $\mathcal{G}$ iff $i < j$ and there exists a column $(a_1, a_2, \ldots, a_{r-s})^t$ in A, such that $a_i = a_j = 1$ while $a_{i+1} = a_{i+2} = \cdots = a_{j-1} = 0$. Notice that each such edge (y_i, y_j) in $\mathcal{G}$ corresponds to a path of length two in $T(A)$: namely $(y_i, x_p), (x_p, y_j)$, where p denotes the position at which the column $(a_1, a_2, \ldots, a_{r-s})^t$ is to be found in A. It follows that if there is a cycle in $\mathcal{G}$, then there is a cycle in $T(A)$. Since $T(A)$ is cycle-free, then so is $\mathcal{G}$. As such, $\mathcal{G}$ necessarily contains at least two vertices of degree ≤ 1, in view of (2). Let y^* be one such vertex in $\mathcal{G}$, and let $a^* = (a_1, a_2, \ldots, a_{n-s})$ be the corresponding row of A. If $\mathrm{wt}(a^*) \leq 2$, then ($\bullet$) is true. If $\mathrm{wt}(a^*) = 3$ and $\deg y^* = 0$, then ($\diamond$) is true. If $\mathrm{wt}(a^*) = 3$ and $\deg y^* = 1$, then ($\star$) is true. Finally, if $\mathrm{wt}(a^*) \geq 4$, then ($\diamond$) is true, regardless of whether $\deg y^* = 0$ or $\deg y^* = 1$. ∎

We will say that an $r \times n$ matrix H is in *reduced canonical form*, if $H = A\|_s B$ and either $s = r$ or all the rows of A have weight ≥ 3.

Lemma 7. *Let $\mathbb{C}$ be an (n, k, d) cycle-free binary linear code. Then there exists a cycle-free parity-check matrix for $\mathbb{C}$, which is in reduced canonical form.*

Proof. Let H be an arbitrary cycle-free parity-check matrix for $\mathbb{C}$. We first put H in s-canonical form, for the highest possible s, by means of row and column permutations. This is achieved by considering all the rows of H of weight one, for which the nonzero entry $*$ is contained in a column of weight one, and all the rows of H of weight two such that at least one of the two $*$ is contained in a column of weight one. Under an appropriate column permutation, these rows of H will form the submatrix $[B\,|\,I_s]$ in (5). If there are no such rows, then $s = 0$ and $H = A$. Since row and column permutations preserve the cycle-free property of H, this procedure produces a cycle-free parity-check matrix $H' = A\|_s B$ for $\mathbb{C}$, which is in canonical form, although not necessarily reduced.

Since $H' = A\|_s B$ is full-rank by assumption, all the rows of A have weight ≥ 1. The key observation is that certain elementary operations on the rows of H' allow us to eliminate rows of weight one and two in A, while still preserving the cycle-free property.

Indeed, suppose that A has a row $(a_1, a_2, \ldots, a_{r-s})$ of weight one, with the single $*$ in position i. Then we can add this row to all the rows of H' that are nonzero at position i. This procedure is equivalent to deleting all but one of the edges incident at the symbol vertex x_i in $T(H')$, which certainly does not create new cycles. Following this procedure, the single $*$ in $(a_1, a_2, \ldots, a_{r-s})$ is contained in a column of weight one. Hence we can transform the resulting cycle-free parity-check matrix for $\mathbb{C}$ into the form $A'\|_{s+1}B'$, by means of row and column permutations, thereby eliminating the row of weight one in A.

Now suppose that A has a row $(a_1, a_2, \ldots, a_{r-s})$ of weight two, with the two $*$ in positions i and j, and let y^* be the corresponding check vertex in $T(H')$. We again add this row to all the rows of H' that are nonzero at position i. Let $(b_1, b_2, \ldots, b_n)$ be such a row, and let y' denote the corresponding check vertex of $T(H')$. If $b_j = b_i = *$, then $T(H')$ contains the cycle $(x_i, y^*), (y^*, x_j), (x_j, y'), (y', x_i)$. Since $T(H')$ is cycle-free, we conclude that $b_i = *$ while $b_j = 0$. Hence, adding $(a_1, a_2, \ldots, a_n)$ to $(b_1, b_2, \ldots, b_n)$ corresponds to deleting the edge (y', x_i) in $T(H')$ while introducing a new edge (y', x_j), as illustrated in Figure 4.

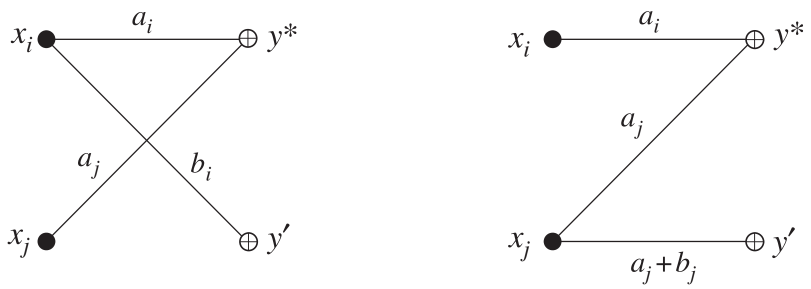


Figure 4. *Part of the Tanner graph for $\mathbb{C}$ before and after the elementary row operation*

However, this new edge cannot close a cycle, since $T(H')$ is cycle-free and a path from y' to x_j already exists in $T(H')$: indeed $(y', x_i), (x_i, y^*), (y^*, x_j)$ is such a path. Following all these elementary row operations, the $*$ at position i in $(a_1, a_2, \ldots, a_n)$ is contained in a column of weight one. Thus we can again transform the resulting cycle-free parity-check matrix for $\mathbb{C}$ into the form $A'\|_{s+1}B'$, thereby eliminating the row of weight two in A.

We iteratively repeat the process described in the foregoing two paragraphs, until either $s = r$ or all the rows of A have weight ≥ 3. This procedure produces a cycle-free parity-check matrix for $\mathbb{C}$, which is in reduced canonical form. ∎

If the reduced canonical form in Lemma 7 is achieved in the extreme case $s = r$, then it is easy to prove the claim of Theorem 5.

Lemma 8. *Let $\mathbb{C}$ be an (n,k,d) cycle-free binary linear code. If there is a parity-check matrix for $\mathbb{C}$ of the form $H = [B\,|\,I_r]$, where $r = n-k$ and the rows of B have weight ≤ 1, then the minimum distance of $\mathbb{C}$ satisfies the upper bound of Theorem 5.*

Proof. If B contains a column of weight w, then clearly $d \le w+1$. Since B is an $r \times k$ matrix, and all the rows of B have weight ≤ 1, we have

$$d \;\le\; \frac{\mathrm{wt}(B)}{k} + 1 \;\le\; \frac{r}{k} + 1 \;=\; \frac{n}{k} \tag{6}$$

As d is an integer, this implies that $d \le \lfloor n/k \rfloor$. It is easy to see that $\lfloor n/k \rfloor \le 2\lfloor n/(k+1) \rfloor$, unless $k = 1$ and n is odd. But, in the latter case, both (4) and (6) reduce to $d \le n$. ∎

4.2. Proof of the main result

We are now in a position to proceed with the proof of Theorem 5. Part of this proof involves tedious calculations, which will be deferred to the appendix. The proof is by induction on the length n of the code. Thus we first transform (4) into the form:

$$d \;\le\; \begin{cases} 2\left\lfloor \dfrac{n}{k+1} \right\rfloor & \text{if} \quad n+1 \not\equiv 0 \mod (k+1) \\[2ex] 2\left\lfloor \dfrac{n}{k+1} \right\rfloor + 1 & \text{if} \quad n+1 \equiv 0 \mod (k+1) \end{cases} \tag{7}$$

that is more conducive to induction on n. It can be easily seen by direct verification that equations (4) and (7) are equivalent.

As the induction basis, we may consider codes of length $n = 2$, for which the bound of Theorem 5 holds trivially. As the induction hypothesis, we assume that the minimum distance of every cycle-free linear code of length $n' < n$ satisfies the bound of Theorem 5.

The induction step is established as follows. Let $\mathbb{C}$ be an (n,k,d) cycle-free binary linear code. We may assume that $2 \le k \le n-1$, since for $k = 1$ the bound of (4) reduces to $d \le n$, while if $k = n$ then $d = 1$ and (4) obviously holds with equality.

By Lemma 7, there exists an $r \times n$ cycle-free parity-check matrix $H = A\|_s B$ for $\mathbb{C}$, which is in reduced canonical form. If $s = r$, then the induction step follows immediately from Lemma 8. Otherwise, Lemma 6 implies that either ($\diamond$) or ($\star$) is true. Observe that case ($\bullet$) of Lemma 6 does not occur, since by the definition of a reduced canonical form, the matrix A does not have rows of weight ≤ 2. Furthermore, both ($\diamond$) and ($\star$) imply that A contains at least two identical columns of weight one. Let i and j denote the positions at which these two columns are found in A. Further, let w_i and w_j denote the weight of the corresponding columns of B. Let $w = w_i + w_j + 2$.

It follows from the canonical form structure of $H = A\|_s B$ that the i-th bit, respectively j-th bit, of $\mathbb{C}$ is repeated w_i times, respectively w_j times, in the last $n-s$ positions. Further observe that the sum of the i-th and the j-th columns of H together with the corresponding $w_i + w_j$ columns of the identity matrix produces the all-zero r-tuple. Hence there is a codeword of weight $w = w_i + w_j + 2$ in $\mathbb{C}$, and $d \leq w$.

We now shorten $\mathbb{C}$ at positions i and j to obtain an (n', k', d') code $\mathbb{C}'$. That is, we consider the subcode of $\mathbb{C}$ consisting of all the codewords that are zero on positions i and j and define $\mathbb{C}'$ to be the code obtained by puncturing out the $w = 2 + (w_i + w_j)$ zero positions in this subcode. Notice that shortening $\mathbb{C}$ at positions i and j is equivalent to deleting $w_i + w_j + 2$ columns of H and $w_i + w_j$ rows of H, as illustrated in Figure 5. It is easy to see that the parameters of the resulting code $\mathbb{C}'$ satisfy

$$n' \;=\; n - w \qquad\qquad k' \;\geq\; k - 2 \qquad\qquad d' \;\geq\; d \tag{8}$$

Furthermore, since H is cycle-free by assumption, the parity-check matrix for $\mathbb{C}'$ which results by deleting rows and columns of H is also cycle-free.

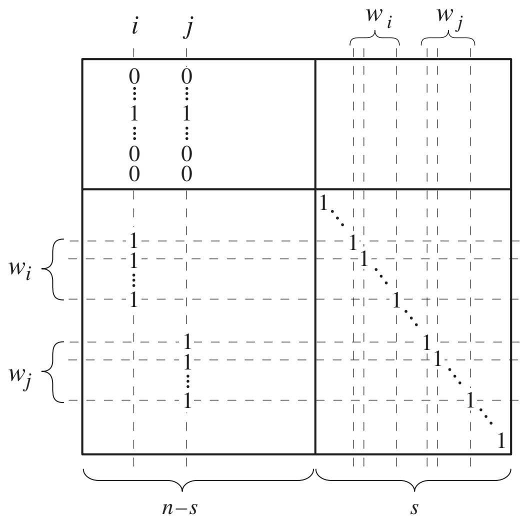


Figure 5. *Deleting rows and columns of $H = A\|_s B$ to shorten a cycle-free code*

It follows that $\mathbb{C}'$ is a cycle-free code of length $n' < n$, and we can invoke the induction hypothesis. We distinguish between two cases.

Case 1. $n' + 1 \not\equiv 0 \mod (k' + 1)$

In this case, the induction hypothesis implies that $d' \leq 2\lfloor n'/(k'+1) \rfloor$. Taking into account the relations (8) between the parameters of $\mathbb{C}$ and $\mathbb{C}'$, we obtain

$$d \;\leq\; 2\left\lfloor \frac{n'}{k'+1} \right\rfloor \;\leq\; 2\left\lfloor \frac{n-w}{k-1} \right\rfloor \;\leq\; 2\left\lfloor \frac{n-d}{k-1} \right\rfloor \tag{9}$$

where the third inequality follows from the fact that $d \leq w$. It is shown in the appendix that the relation between n, k, and d in (9) implies (7).

Case 2. $n' + 1 \equiv 0 \mod (k' + 1)$

We again apply the induction hypothesis. Notice that in this case, the upper bound of Theorem 5 may be re-written as

$$d' \;\leq\; 2\left\lfloor \frac{n'}{k'+1} \right\rfloor + 1 \;=\; 2\,\frac{n'+1}{k'+1} - 1 \tag{10}$$

where $(n'+1)/(k'+1)$ is a positive integer. Suppose that $d \leq d'$ is an even integer. Then, since the right-hand side of (10) is an odd integer, we have

$$d \;\leq\; 2\,\frac{n'+1}{k'+1} - 2 \;\leq\; 2\,\frac{n-d+1}{k-1} - 2 \tag{11}$$

where the second inequality in (11) follows from (8) along with the fact that $d \leq w$. It is shown in the appendix that (11) implies (7).

Now suppose that d is odd. In this case, the bound of (10) does not suffice to establish (7), and we need to use the additional structure present in statements ($\diamond$) and ($\star$) of Lemma 6. Suppose that ($\diamond$) is true, and the matrix A contains three identical columns of weight one, at positions α, β, γ. Let $w_\alpha, w_\beta, w_\gamma$ denote the weight of the corresponding columns of B. Notice that at least one of $w_\alpha + w_\beta$, $w_\alpha + w_\gamma$, $w_\beta + w_\gamma$ is an even integer. Hence we can choose the two positions i and j in Figure 5 from the three positions α, β, γ, in such a way that $w = w_i + w_j + 2$ is even. Since $d \leq w$ and d is odd, it follows that $d \leq w - 1$. In conjunction with (10), we thus obtain

$$d \;\leq\; 2\,\frac{n'+1}{k'+1} - 1 \;\leq\; 2\,\frac{n-w+1}{k-1} - 1 \;\leq\; 2\,\frac{n-d}{k-1} - 1 \tag{12}$$

It is shown in the appendix that if d is odd, then (12) implies (7). Now suppose that ($\star$) is true, and the matrix H contains a row of weight three, with the three $*$ at positions h, i, j. Then after deleting $w_i + w_j + 2$ columns of H and $w_i + w_j$ rows of H as illustrated in Figure 5, we are left with a row of weight one, with the single $*$ at position h. This means that the h-th position in $\mathbb{C}'$ is entirely zero; this position can be punctured-out without decreasing the dimension or the minimum distance. We thus obtain an (n^*, k^*, d^*) code $\mathbb{C}^*$, with $n^* = n' - 1$, $k^* = k'$, and $d^* = d'$. Applying the induction hypothesis to $\mathbb{C}^*$, we get

$$d \;\leq\; d^* \;\leq\; 2\left\lfloor \frac{n^*}{k^*+1} \right\rfloor \;\leq\; 2\left\lfloor \frac{n-w-1}{k-1} \right\rfloor \;\leq\; 2\left\lfloor \frac{n-d}{k-1} \right\rfloor \tag{13}$$

where the second inequality follows from the fact that if $n'+1 \equiv 0 \mod (k'+1)$ then $n^* + 1 \not\equiv 0 \mod (k^* + 1)$. The right-hand side of (13) is the same as relation (9), which was already considered in Case 1.

It remains to consider the case where $\mathbb{C}' = \{\mathbf{0}\}$, namely $k' = 0$. But in this case $k \leq 2$ in view of (8), and the upper bound of Theorem 5 follows directly from the Griesmer bound [19, p. 547]. Since we have now exhausted all the possibilities, this establishes the induction step, and completes the proof of Theorem 5.

4.3. Optimal cycle-free codes

While proving the upper bound of Theorem 5 required considerable effort, showing that this bound is exact is easy. We now construct a family of cycle-free codes that attain the bound of Theorem 5 with equality, for all values of n and k. The construction is quite simple: as in Section 3, we start with the single-parity-check code $\mathcal{E}_{k+1}$ of dimension k, and repeat the symbols of $\mathcal{E}_{k+1}$ until a code $\mathbb{C}$ of length n is obtained. It is obvious that the dimension of $\mathbb{C}$ is k, and by Lemma 3 this code is cycle-free. The minimum distance of $\mathbb{C}$ will depend on the sequence of symbol repetitions. The idea is to repeat each symbol in $\mathcal{E}_{k+1}$ equally often, in as much as possible. For example, for $n = 13$ and $k = 3$, we obtain the following parity-check matrix in reduced canonical form:

$$H = \left[\begin{array}{cccc|ccccccccc} 1&1&1&1&0&0&0&0&0&0&0&0&0 \\ \hline 1&0&0&0&1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&1 \end{array}\right] \qquad (14)$$

which defines a $(13, 3, 6)$ cycle-free code. In general, the number of symbols to be repeated is $k + 1$, while the number of positions available is $n - (k + 1)$. Write:

$$n - (k+1) \;=\; a(k+1) + b$$

where a, b are integers, and $0 \le b \le k$. This decomposition of the number of available positions means that in our construction exactly $k-b+1$ symbols of $\mathcal{E}_{k+1}$ will be repeated

$$a \;=\; \left\lfloor \frac{n-(k+1)}{k+1} \right\rfloor \;=\; \left\lfloor \frac{n}{k+1} \right\rfloor - 1$$

times, while the remaining b symbols of $\mathcal{E}_{k+1}$ will be repeated $a+1$ times. If $b \le k-1$, then at least two symbols of $\mathcal{E}_{k+1}$ are repeated exactly a times. Since $\mathcal{E}_{k+1}$ contains a codeword of weight 2 in every two positions, the minimum distance of the resulting code $\mathbb{C}$ is

$$d \;=\; 2 + a + a \;=\; 2 \left\lfloor \frac{n}{k+1} \right\rfloor \qquad (15)$$

If $b = k$, then only one symbol in $\mathcal{E}_{k+1}$ is repeated a times, while all the other symbols are repeated $a + 1$ times. In this case, the minimum distance of $\mathbb{C}$ is

$$d \;=\; 2 + a + (a{+}1) \;=\; 2 \left\lfloor \frac{n}{k+1} \right\rfloor + 1 \qquad (16)$$

Notice that $b = k$ if and only if $n+1 \equiv 0 \mod (k+1)$. Hence it follows from (15) and (16) that the code $\mathbb{C}$ constructed in this manner attains the bound of Theorem 5 with equality.

Figure 6 schematically shows two alternative cycle-free Tanner graphs for codes resulting from this construction (compare the Tanner graph in Figure 6a with Figure 2b).

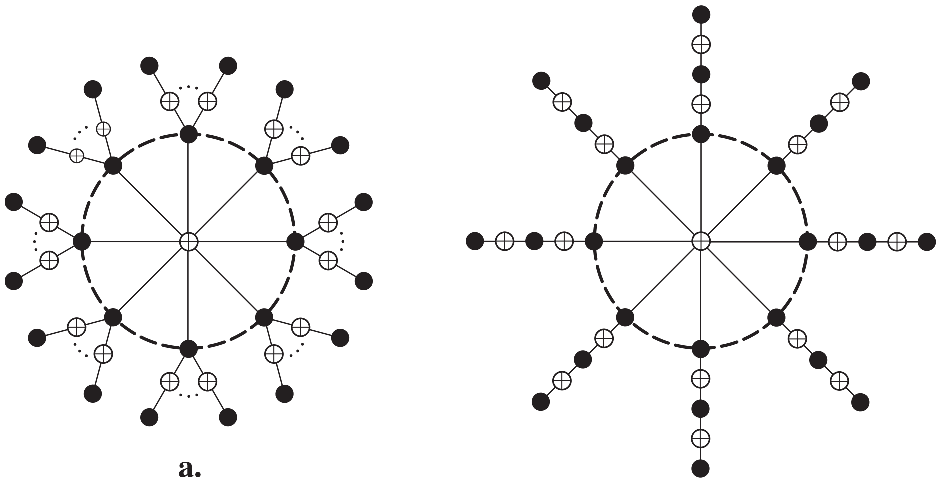


Figure 6. *Two alternative Tanner graphs for optimal cycle-free codes*

We point out that although cycle-free codes obtained by repeating symbols in $\mathcal{E}_{k+1}$ have the highest possible minimum distance, they are not the only codes with this property. For example, consider the following parity-check matrix in reduced canonical form:

$$H = \left[\begin{array}{ccccc|cccccccc} 1&1&1&0&0&0&0&0&0&0&0&0&0\\ 1&0&0&1&1&0&0&0&0&0&0&0&0\\ \hline 0&1&0&0&0&1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&1&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&1&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&1 \end{array}\right] \tag{17}$$

It is easy to see that this matrix defines a $(13,3,6)$ cycle-free code $\mathbb{C}'$, whose distance attains the bound of Theorem 5 with equality. This code was obtained by repeating symbols in a $(5,3,2)$ code. It can be readily verified that $\mathbb{C}'$ is not equivalent to the $(13,3,6)$ cycle-free code $\mathbb{C}$, defined by the parity-check matrix in (14) and obtained by repeating symbols in $\mathcal{E}_4$. For instance, $\mathbb{C}$ contains the all-one codeword, while $\mathbb{C}'$ does not.

5. Further results and open problems

In this section, we discuss three different topics: a connection between binary cycle-free codes and cut-set codes of a graph, asymptotic behavior of Tanner graphs with cycles, and the extension of the results of the previous section to general Tanner graphs. In each case, we provide a number of open problems for future research.

5.1. Cycle-free codes and graph-theoretic codes

There is an interesting connection between cycle-free codes and cut-set codes of a graph. Let $\mathcal{G} = (V, E)$ be a multigraph (a graph that may contain multiple edges with both endpoints the same) with $n = |E|$ edges and $m = |V|$ vertices. A *cut-set* in $\mathcal{G}$ is a set of edges which consists of all the edges having one endpoint in some set $X \subset V$ and the other endpoint in $V \setminus X$. Under the operation of symmetric difference, the cut-sets in $\mathcal{G}$ form a subspace of the binary vector space of all subsets of E. Hence replacing subsets of E by their characteristic vectors in $\mathbb{F}_2^n$ produces a binary linear code $\mathbb{C}(\mathcal{G})$, called the *cut-set code* of $\mathcal{G}$. The dual code of $\mathbb{C}(\mathcal{G})$ is the *cycle code* of $\mathcal{G}$, defined as the linear span of the characteristic vectors of cycles in $\mathcal{G}$. Graph theoretic codes, namely cut-set codes and cycle codes of a graph, have been extensively studied — see [6, 15, 14, 23, 24] for instance. The connection between cycle-free codes and cut-set codes of a graph can be summarized as follows.

Theorem 9. *Let $\mathbb{C}$ be a cycle-free binary linear code of length n. Then there exists a graph $\mathcal{G}$ with n edges, such that $\mathbb{C}$ is a cut-set code of $\mathcal{G}$.*

Proof. Let H be an $r \times n$ cycle-free parity-check matrix for $\mathbb{C}$, and let $T = T(H)$ be the corresponding cycle-free Tanner graph that represents $\mathbb{C}$. The following procedure converts T into a graph $\mathcal{G}$, such that $\mathbb{C}$ is the cut-set code of $\mathcal{G}$. We will describe this procedure assuming that T is a tree, in which case $\mathcal{G}$ is connected. In case T is a forest consisting of ω trees, the same procedure should be carried out independently for each tree in T, and $\mathcal{G}$ will have ω connected components.

Let $\mathcal{Y} = \{y_1, y_2, \ldots, y_r\}$ be the set of check vertices in T, and let $X_i \subseteq \mathcal{X}$ denote the neighborhood of $y_i \in \mathcal{Y}$ for $i = 1, 2, \ldots, r$. Further define $X_i^* = X_1 \cup X_2 \cup \cdots \cup X_i$. Since T is a tree, it is always possible to enumerate the check vertices in T in such a way that X_i intersects X_{i-1}^* in *one and only one* symbol vertex for all i. Given such enumeration $y_1, y_2, \ldots, y_r$, we construct $\mathcal{G}$ iteratively, check-vertex by check-vertex. First, we represent y_1 and its neighborhood X_1 by a cycle $\mathcal{G}_1$ consisting of $|X_1|$ edges and $|X_1|$ vertices. Now suppose that $X_2 \cap X_1 = \{x_2\}$. Then we create $\mathcal{G}_2$ from $\mathcal{G}_1$ by appending $|X_2| - 1$ edges — one for each symbol vertex in X_2 except x_2 — and $|X_2| - 2$ vertices, in such a way that the edges corresponding to the symbol vertices in X_2 form a cycle in $\mathcal{G}_2$. And so forth: if $X_i \cap X_{i-1}^* = \{x_i\}$, we create $\mathcal{G}_i$ from $\mathcal{G}_{i-1}$ by appending $|X_i| - 1$ edges and $|X_i| - 2$ vertices, in such a way that the edges corresponding to the symbols in X_i form a new cycle. It is easy to see that $\mathcal{G} = \mathcal{G}_r$ will contain exactly n edges and $n - (r-1)$ vertices. Furthermore, the code $\mathbb{C}^\perp$ generated by H is precisely the cycle code of $\mathcal{G}$. Since the cut-set code of $\mathcal{G}$ is the dual of its cycle code, our proof is complete. ∎

For example, the cycle-free codes defined by the parity-check matrices in (14) and (17) are cut-set codes of the graphs depicted in Figure 7a and Figure 7b, respectively.

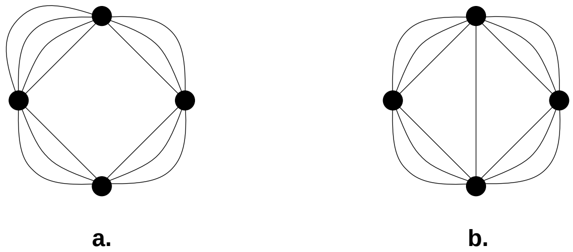


Figure 7. *Two inequivalent cycle-free cut-set codes*

For cut-set codes, it is well-known [15, 21] that $2n \geq md$, where m is the number of vertices in the underlying graph $\mathcal{G}$. Indeed, this follows immediately from the fact that if the minimum distance of $\mathbb{C}(\mathcal{G})$ is d, then every vertex of $\mathcal{G}$ must have degree at least d, otherwise the cut-set that isolates this vertex will have less than d edges. It is also well-known that $\dim \mathbb{C}(\mathcal{G}) = m - \omega(\mathcal{G})$, where $\omega(\mathcal{G})$ is the number of connected components in $\mathcal{G}$. Thus we obtain the following cut-set bound on the minimum distance of cycle-free codes

$$d \;\leq\; \frac{2n}{k+\omega(\mathcal{G})} \;\leq\; \frac{2n}{k+1} \tag{18}$$

If it is known that d is even, then the cut-set bound of (18) obviously implies Theorem 5. In general, however, Theorem 5 is stronger than the cut-set bound based on Theorem 9. Indeed, there exist cut-set codes that are not cycle-free. As a simple example, consider the $(6,3,3)$ cut-set code of the graph depicted in Figure 8, and notice that the minimum distance of this code violates the upper bound of Theorem 5.

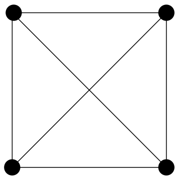


Figure 8. *A cut-set code which is not cycle-free*

In summary, we have proved that every cycle-free binary linear code is a cut-set code. We pose the converse as an open problem: which cut-set codes are cycle-free?

An answer to this question may follow from a closer look at the construction of the graph $\mathcal{G}$ from a cycle-free Tanner graph T in the proof of Theorem 9. We observe that $\mathcal{G}$ is planar, and that any two regions in $\mathcal{G}$, except for the outer region, intersect in at most one edge. Furthermore, if we remove from the *dual graph* of $\mathcal{G}$ the vertex corresponding to the outer region of $\mathcal{G}$ and all the edges adjacent to this vertex, the resulting graph is a tree (or a forest). While we believe that the cut-set code of any graph with these properties is cycle-free, we will not pursue a proof of this claim herein.

5.2. Asymptotics for Tanner graphs with cycles

It is obvious from Theorem 5 that Tanner graphs without cycles cannot support asymptotically good codes. Starting with Theorem 5, it is not difficult to show that the same is true for Tanner graphs with cycles, unless the number of cycles increases exponentially with the length of the code as $n \to \infty$. To see this, suppose the the cycle rank of a Tanner graph $T = (V, E)$ representing an (n, k, d) code $\mathbb{C}$ is $c = |E| - |V| + \omega(T)$. This means that T contains 2^c cycles and unions of disjoint cycles (cf. [21, p.137]). Now let x_i be a symbol vertex that lies on a cycle in T. Then removing x_i and all the edges incident on x_i from T produces a graph whose cycle rank is strictly less than c. This procedure is equivalent to shortening $\mathbb{C}$ at the i-th position to obtain an (n', k', d') code $\mathbb{C}'$ with $n' = n-1$, $k' \geq k-1$, and $d' \geq d$. Since the cycle rank strictly decreases each time we cut a cycle in T in this way, after repeating this procedure $t \leq c$ times we obtain a cycle-free code $\mathbb{C}^*$. Clearly $\mathbb{C}^*$ is an (n^*, k^*, d^*) code with $n^* = n - t$, $k^* \geq k - t$, and $d^* \geq d$. Thus Theorem 5 implies

$$d \;\leq\; d^* \;\leq\; 2\left\lfloor \frac{n^*}{k^*+1} \right\rfloor + 1 \;\leq\; 2\,\frac{n-t}{k-t+1}\, + 1 \tag{19}$$

Now let $\gamma = c/n$, and notice that $t/n \leq \gamma$. Hence if $\lim_{n\to\infty} \gamma = 0$, then (19) asymptotically reduces to $d \lesssim 2/R$, as in (1). Thus to support an asymptotically good sequence of codes, c must grow linearly with n, which means that the number of cycles 2^c grows exponentially with n. It would be useful to find out how the parameter $\gamma = c/n$, which has to do with the number of cycles, trades off versus the traditional asymptotic parameters $\delta = d/n$ and $R = k/n$ as $n \to \infty$. It would be also interesting to investigate, at least asymptotically, codes that have Tanner graphs of prescribed minimum girth.

5.3. General Tanner graphs without cycles

We now return to the case of general Tanner graphs, as defined on pp. 1–2, and observe that every general Tanner graph $(\mathcal{G}, \mathcal{L})$ can be converted into a simple Tanner graph for the same code through a vertex-splitting procedure. Indeed, let $y \in \mathcal{Y}$ be a check vertex in $\mathcal{G}$, let $\{x_{i_1}, x_{i_2}, \ldots, x_{i_\delta}\} \subseteq \mathcal{X}$ be the neighborhood of y, and let $\mathcal{C}$ be the corresponding constraint code of length δ. If $\dim \mathcal{C} = \kappa$, we split y into $\delta - \kappa$ vertices $y'_1, y'_2, \ldots, y'_{\delta-\kappa}$ and create edges between $x_{i_1}, x_{i_2}, \ldots, x_{i_\delta}$ and $y'_1, y'_2, \ldots, y'_{\delta-\kappa}$ according to a parity-check matrix H for $\mathcal{C}$. An obvious but important observation is this: if H is cycle-free, then this procedure does not create new cycles. Thus we have proved the following statement.

Proposition 10. *If a linear code $\mathbb{C}$ can be represented by a general Tanner graph $(\mathcal{G}, \mathcal{L})$ such that $\mathcal{G}$ is cycle-free and all the constraints in $\mathcal{L}$ are cycle-free, then $\mathbb{C}$ can be represented by a simple Tanner graph without cycles.*

An immediate consequence of Proposition 10 is that all the results derived so far for simple Tanner graphs, including the bound of Theorem 5, straightforwardly extend to general Tanner graphs with cycle-free constraints.

In the general case, where check constraints are not necessarily cycle-free, it appears to be very difficult to say anything about the structure/properties of the code being represented. As an example, consider a general Tanner graph for $\mathbb{C}$ which contains a single check vertex $\mathcal{Y} = \{y\}$ with the corresponding constraint code being $\mathbb{C}$ itself. The existence of this cycle-free representation for $\mathbb{C}$ obviously does not provide any information whatsoever about $\mathbb{C}$.

Notwithstanding the trivial "counter-example" discussed above, it is plausible that if the underlying Tanner graph is cycle-free, the distance of $\mathbb{C}$ should be limited by the distances of the constraint codes in some manner. Furthermore, if simple decoding is sought for, simple constraint codes must be used. It thus appears that the range of code parameters that are possible with cycle-free Tanner graphs will depend on the decoding complexity tolerated. We leave further investigation of this relation as an open problem.

A. Appendix

We will show that each of the three relations (9),(11),(12) between n, k, and d derived in Section 4.2 implies (7), providing d is an integer in (9),(11) and d is an odd integer in (12). In order to make the appendix self-contained, we now re-state these inequalities:

$$d \;\le\; 2\left\lfloor \frac{n-d}{k-1} \right\rfloor \tag{9}$$

$$d \;\le\; 2\,\frac{n-d+1}{k-1} - 2 \tag{11}$$

$$d \;\le\; 2\,\frac{n-d}{k-1} - 1 \tag{12}$$

Notice that what we are trying to establish has nothing to do with graphs or codes; this is just manipulation of integer inequalities. In particular, we have following simple lemma.

Lemma 11. *If $a \le b/c$ and a, b, c are positive integers, then $a \le (b+a)/(c+1)$.*

The proof of Lemma 11 is straightforward, and is left to the reader. We first deal with (12), assuming d is odd. Taking the common denominator and applying (twice) Lemma 11, we see that (12) implies

$$d \;\le\; \frac{2n-(k-1)}{k+1} \;=\; \frac{2(n+1)}{k+1} - 1 \tag{20}$$

Since $(d+1)/2$ is an integer for odd d, it follows from (20) that $(d+1)/2 \le \lfloor (n+1)/(k+1) \rfloor$. This may be re-written as:

$$d \;\le\; 2\left\lfloor \frac{n+1}{k+1} \right\rfloor - 1 \tag{21}$$

If $n+1 \not\equiv 0 \mod (k+1)$ then $\lfloor (n+1)/(k+1) \rfloor = \lfloor n/(k+1) \rfloor$, and (21) clearly implies (7). If $(n+1)/(k+1)$ is an integer, then (21) is precisely the equivalent form of (7) given in (10).

It is easy to see that if d is an odd integer, then (9) implies (12). Since this case was already established above, it remains to prove (9) for even d. Using once again Lemma 11, we see that (9) implies $d \le 2n/(k+1)$, or equivalently $d/2 \le n/(k+1)$. Since $d/2$ is an integer for even d, we can take the integer part of $n/(k+1)$ in the above expression. It follows that for even d, we have

$$d \ \le \ 2 \left\lfloor \frac{n}{k+1} \right\rfloor$$

which clearly implies (7). Finally, it can be readily seen that if d is an integer, then (11) implies (9). Hence, our proof of Theorem 5 is now complete.

Acknowledgement. We are grateful to Jeff Erickson, Amir Khandani, and Ralf Kötter for stimulating discussions. We would also like to thank the anonymous referees for their valuable comments, which improved the presentation of this correspondence.

References

[1] S.M. Aji and R.J. McEliece, "A general algorithm for distributing information in a graph," in *Proc. IEEE Int. Symp. Inform. Theory*, p. 6, Ulm, Germany, 1997.

[2] S.M. Aji and R.J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, submitted for publication, July 1998.

[3] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, 1974.

[4] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, 1996.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes," in *Proc. IEEE Int. Conf. on Communications*, pp. 1064-1070, Geneva, Switzerland, 1993.

[6] J. Bruck and M. Blaum, "Neural networks, error-correcting codes, and polynomials over the binary n-cube," *IEEE Trans. Inform. Theory*, vol. 35, pp. 976–987, 1989.

[7] M. Esmaeili and A.K. Khandani, "Acyclic Tanner graphs and maximum-likelihood decoding of linear block codes," preprint, May 1998.

[8] J. Feigenbaum, G.D. Forney, Jr., B.H. Marcus, R.J. McEliece, and A. Vardy, Editors., special issue on "Codes and Complexity," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1649–2064, November 1996.

[9] G.D. Forney, Jr., "The forward-backward algorithm," in *Proc. 34-th Allerton Conference on Comm., Control, and Computing*, Monticello, IL., pp. 432–446, October 1996.

[10] B.J. Frey, "Bayesian networks for pattern classification, data compression, and channel coding," Ph.D. dissertation, University of Toronto, Canada, July 1997.

[11] B.J. Frey, F.R. Kschischang, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, submitted for publication, July 1998.

[12] R.G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21–28, 1962.

[13] R.G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.

[14] S.L. Hakimi and J. Bredeson, "Graph theoretic error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 14, pp. 584–591, 1968.

[15] S.L. Hakimi and H. Frank, "Cut-set matrices and linear codes," *IEEE Trans. Inform. Theory*, vol. 11, pp. 457–458, 1965.

[16] D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, to appear, 1999.

[17] D.J.C. MacKay, "Gallager codes that are better than turbo codes," in *Proc. 36-th Allerton Conf. Comm., Control, and Computing*, Monticello, IL., September 1998.

[18] D.J.C. MacKay and R.M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electronics Letters*, vol. 32, pp. 1645–1646, 1996.

[19] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, New York: North-Holland, 1977.

[20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Kaufmann, 1988.

[21] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, 2-nd Edition, Cambridge, MA: MIT Press, 1961.

[22] M. Sipser and D.A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710–1722, 1996.

[23] P. Solé and T. Zaslavsky, "The covering radius of the cycle code of a graph," *Discrete Applied Math.*, vol. 45, pp. 63–70, 1993.

[24] P. Solé and T. Zaslavsky, "A coding approach to signed graphs," *SIAM J. Discrete Math.*, vol. 7, pp. 544-553, 1994.

[25] D.A. Spielman, "Linear-time encodable and decodable codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1723–1731, 1996.

[26] R.M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, 1981.

[27] A. Vardy, "Trellis structure of codes," to appear in HANDBOOK OF CODING THEORY, V.S. Pless and W.C. Huffman (Eds.), Elsevier, Amsterdam, 1998.

[28] D.B. West, *Introduction to Graph Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1996.

[29] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Department of Electrical Engineering, University of Linköping, Sweden, April 1996.

[30] N. Wiberg, H.-A. Loeliger and R. Kötter, "Codes and iterative decoding on general graphs," *Euro. Trans. Telecommun.*, vol. 6, pp. 513–526, 1995.