Transactions Letters

Comparison of Constructions of Irregular Gallager Codes

David J. C. MacKay, Simon T. Wilson, *Associate Member, IEEE*, and Matthew C. Davey

***Abstract*—The low-density parity check codes whose performance is closest to the Shannon limit are "Gallager codes" based on irregular graphs. We compare alternative methods for constructing these graphs and present two results. First, we find a "super-Poisson" construction which gives a small improvement in empirical performance over a random construction. Second, whereas Gallager codes normally take N^2 time to encode, we investigate constructions of regular and irregular Gallager codes that allow more rapid encoding and have smaller memory requirements in the encoder. We find that these "fast encoding" Gallager codes have equally good performance.**

***Index Terms*—Channel coding, error correction coding, Gaussian channels, graph theory, iterative probabilistic decoding, random codes.**

I. Introduction

GALLAGER codes [3], [4] are low-density parity check codes constructed at random subject to constraints on the weight of each row and of each column. The original *regular* Gallager codes have very sparse random parity check matrices with uniform weight t per column and t_r per row. (We will also use the term "regular" for codes that have nearly uniform weight columns and rows—for example, codes which have some weight 2 columns and some weight 3 columns.) These codes are asymptotically good and can be practically decoded with Gallager's sum–product algorithm giving near Shannon limit performance when large block lengths are used [6]–[8]. Regular Gallager codes have also been found to be competitive codes for short block-length code-division multiple-access (CDMA) applications [10].

Recent advances in the performance of Gallager codes are summarized in Fig. 1. The rightmost curve shows the performance of a regular binary Gallager code with rate 1/4. The best known binary Gallager codes are *irregular* codes whose parity check matrices have *nonuniform* weight per column [5]; the performance of one such code is shown by the second curve from the right. The best known Gallager codes of all are Gallager codes defined over finite fields $GF(q)$ [1], [2]. The remaining two solid curves in Fig. 1 show the performance of a regular Gallager code over $GF(16)$ [2] and

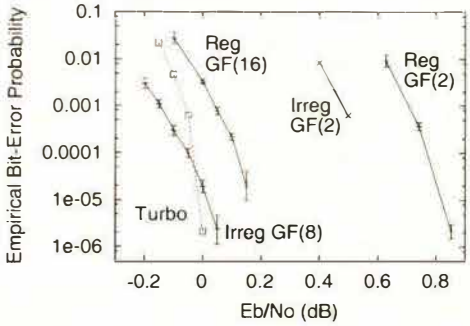


Fig. 1. Empirical results for Gaussian channel, rate 1/4 left–right: irregular LDPC, $GF(8)$ blocklength 24 000 bits; JPL Turbo, blocklength 65 536 bits; regular LDPC, $GF(16)$, blocklength 24 448 bits; irregular LDPC, $GF(2)$, blocklength 64 000 bits; regular LDPC, $GF(2)$, blocklength 40 000 bits. (Reproduced from [1].)

an irregular code over $GF(8)$ with bit-error probability of 10^{-4} at $E_b/N_0 = -0.05$ dB [1]. In comparing this code with the rate 1/4 turbo-code shown by the dotted line, the following points should be noted. 1) The transmitted blocklength of the irregular Gallager code is only 24 000 bits, whereas that of the turbo-code is 65 536 bits. 2) The errors made by the Gallager codes were all detected errors, whereas turbo-codes make undetected errors at high signal-to-noise ratio. This difference is not caused by a difference in the decoding algorithm: both codes are decoded by the sum–product algorithm [9]. Turbo-codes make undetected errors because *they have low-weight codewords*. For Gallager codes, the rate of occurrence of undetected errors is extremely small because they have good distance properties (the minimum distance scales linearly with the blocklength) [4]. In all our experiments with Gallager codes of block length greater than 10 000 and column weight at least 3, undetected errors have never occurred.

The excellent performance of irregular Gallager codes is the motivation for this paper, in which we explore ways of further enhancing these codes.

The irregular codes of Luby, Mitzenmacher, Shokrollahi, and Spielman [5] have parity check matrices with both nonuniform weight per row and nonuniform weight per column. It has not yet been established whether both of these nonuniformities are desirable. In our experience with codes for noisy channels, performance is more sensitive to the distribution of column weights. In this paper, we concentrate on irregular codes with the weight per row as uniform as possible.

We can define an irregular Gallager code in two steps. First, we select a *profile* that describes the desired number of columns of each weight and the desired number of rows of

Paper approved by S. B. Wicker, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received August 11, 1998; revised January 27, 1999. This paper was presented in part at the 1998 Allerton Conference on Communications, Control, and Computing, Allerton House, IL, September 1998.

The authors are with the Department of Physics, University of Cambridge, Cambridge CB3 0HE, U.K. (e-mail: mackay@mrao.cam.ac.uk; stw11@mrao.cam.ac.uk; mcdavey@mrao.cam.ac.uk).

Publisher Item Identifier S 0090-6778(99)07784-3.

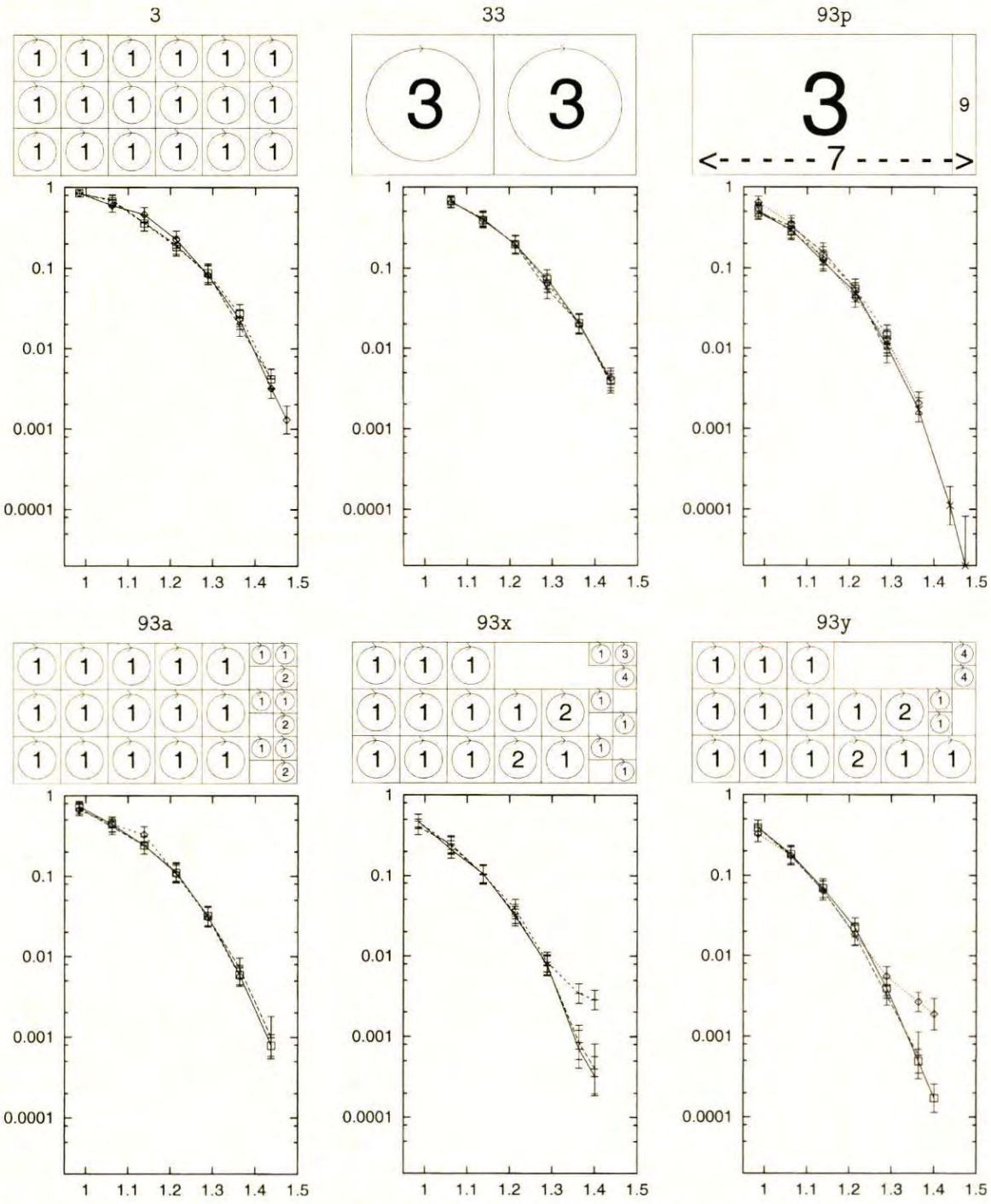


Fig. 2. Upper panels: constructions of regular and irregular codes. Lower panels: performance of these codes. The construction types shown are regular, (3, 33), Poisson (93p), sub-Poisson (93a), super-Poisson (93x), and super-Poisson (93y). Notation for upper panels for all constructions except 93p: an integer represents a number of permutation matrices superposed on the surrounding square. Horizontal and vertical lines indicate the boundaries of the permutation blocks. Notation for the Poisson construction 93p: integers "3" and "9" represent column weights. The integer "7" represents the row weight. Lower panels show the performance of several random codes of each construction. Vertical axis: block error probability. Horizontal axis: E_b/N_0 in dB. All codes have $N = 9972$ and $K = M = 4986$. All errors were detected errors, as is usual with Gallager codes.

each weight. The parity check matrix of a code can be viewed as defining a bipartite graph with "bit" vertices corresponding to the columns and "check" vertices corresponding to the rows. Each nonzero entry in the matrix corresponds to an edge connecting a bit to a check. The profile specifies the degrees of the vertices in this graph.

way that satisfies the constraints. (In the case of nonbinary Gallager codes, we also need to choose an algorithm for assigning values to the nonzero entries in the matrix.)

This paper has two parts. In the first part (Section III), we compare alternative construction methods for a fixed profile in order to find out whether the construction method matters. In

motivation for this second study is that the only drawback of regular Gallager codes compared to turbo-codes for CDMA applications appears to be their greater encoding complexity [10].

In the experiments presented here, we study binary codes with rate 1/2 and blocklength about $N = 10\,000$. We simulate an additive white Gaussian noise channel in the usual way [2] and examine the block error probability as a function of the signal-to-noise ratio. The error bars we show are one standard deviation error bars on the estimate of the logarithm of the block error probability $\hat{p}$ defined. Thus, when we observe r failures out of n trials, $p_{\pm} = \hat{p}\exp(\pm\sigma_{\log p})$ where $\sigma_{\log p} = \sqrt{(n-r)/(rn)}$.

II. Constructions

We compare the following methods.

Poisson: The edges are placed "completely at random," subject to the profile constraints and the rule that you cannot put two edges between one pair of vertices, which would correspond to a double entry in the parity check matrix. One way to implement a Poisson construction is to make a list of all the columns in the matrix, with each column appearing in the list a number of times equal to its weight, then make a similar list of all the rows in the matrix, each row appearing with multiplicity equal to its weight, and then map one list onto the other by a random permutation, taking care not to create duplicate entries [5].

A variation of this construction is to require that no two columns in the parity check matrix have an overlap greater than one, i.e., forbid cycles of length 4 in the graph. (Similar to construction 1A in [8].) A second variation requires that the graph have no cycles of length less than some 1. (Similar to construction 1B in [8].) This constraint can be quite hard to enforce if the profile includes high weight rows or columns.

Permutations: We can build parity check matrices by superposing random permutation matrices [4]. The convenience of this method depends on the profile. There are many ways of laying out these permutation matrices to satisfy a given profile. We will distinguish "*super-Poisson*" and "*sub-Poisson*" constructions.

- In a super-Poisson construction, the distribution of high weight columns per row has greater variance than a Poisson distribution.
- In a sub-Poisson construction, the distribution of high weight columns per row has smaller variance than a Poisson distribution.

III. Comparing Poisson, Super-Poisson, and Sub-Poisson Constructions

A. Profiles and Constructions Studied in this Paper

*1) Regular Codes—*3 *and* 33*:* As our baseline, we study regular Gallager codes with weight per column exactly $t = 3$ and weight per row exactly $t_r = 6$. We construct parity check matrices satisfying this profile from permutation matrices in an integer (for example, "3") denotes the superposition inside that square of that number of random permutation matrices. The matrices are generated at random subject to the constraint that no two nonzero entries coincide.

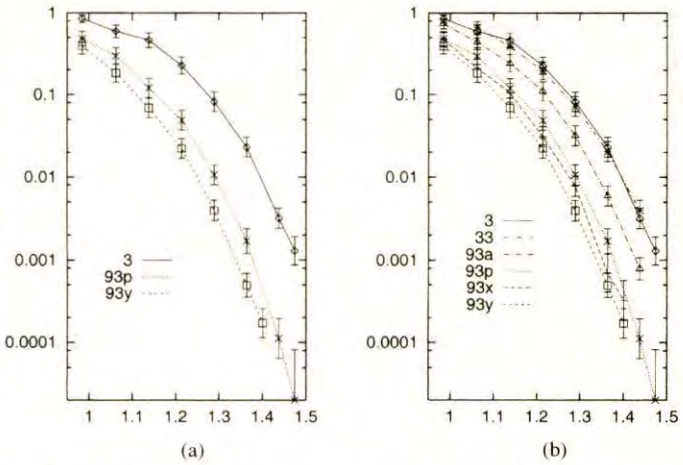


Fig. 3. (a) Comparison of one representative of each of the constructions: 3 (regular), 93p (Poisson) and 93y (super-Poisson). (b) Representatives of all six constructions in Fig. 2. Vertical axis: block error probability. Horizontal axis: E_b/N_0 in dB.

TABLE I
The Two Profiles Studied in this Paper

Profile 3	Column weight	Fraction of columns	Row weight	Fraction
	3	1	6	1
Profile 93	Column weight	Fraction of columns	Row weight	Fraction
	3	11/12	7	1
	9	1/12		

*2) Irregular Codes—*93p, 93a, 93x, *and* 93y*:* We chose the profile "93" shown in Table I. It has columns of weight 9 and of weight 3; all rows have weight 7. Note that this profile only differs from the regular profile "3" in that some extra 1's are added to 1/12 of the columns. We emphasize that this profile has not been carefully optimized, so the results of this paper should not be taken as describing the best that can be done with irregular binary Gallager codes. We chose this profile because it lends itself to interesting experiments.

We will refer to the bits that connect to nine checks as "elite" bits. We use four different constructions that match this profile, named as follows. These constructions are depicted diagrammatically in the upper panels of Fig. 2.

*Poisson—*93p*:* In this construction, while most checks will connect to one or two elite bits, a fraction of them will connect to more than two elite bits, and some will connect to none.

*Sub-Poisson—*93a*:* This construction allocates exactly one or two elite bits to each check.

Super-Poisson: 93x and 93y are, respectively, moderately and very super-Poisson. In 93y, one third of the checks are connected to *four* elite bits, one third are connected to *one*, and one third are connected to *none*.

B. Results

1) Variability Within Each Construction: For each con-

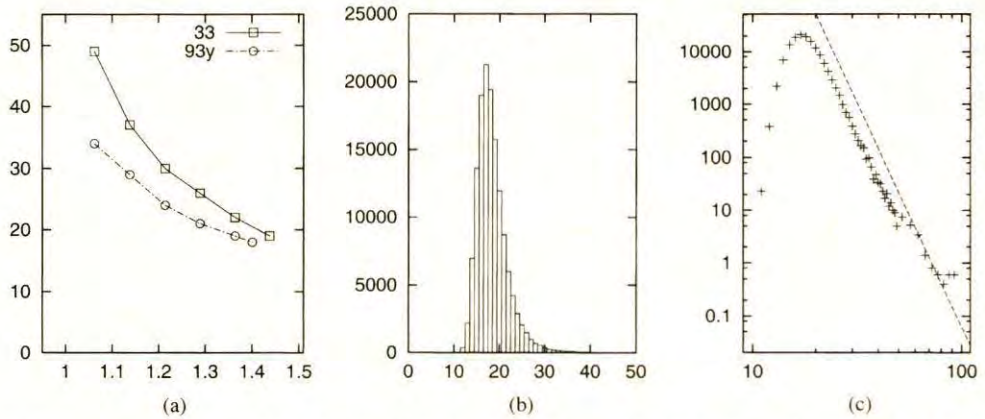
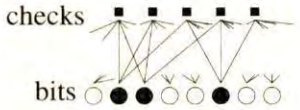


Fig. 4. (a) Comparison of irregular 93y with regular 33 code. Vertical axis: median number of iterations. Horizontal: E_b/N_0 in dB. (b) Histogram of number of iterations for 93y code at $E_b/N_0 = 1.4$. (c) Log/log plot of iterations histogram showing that the tail of the distribution is well approximated by a power law. The straight line has slope -8.5. Above 50, iterations were binned into intervals of 5.

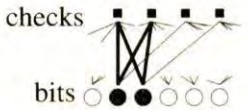
studied were of rate 1/2, with blocklength $N = 4986$. The results are shown in Fig. 2. We see no significant variability among the 3, 33, 93i (Poisson) or 93a (sub-Poisson) codes. But among the super-Poisson codes, 93x and 93y, there is some variability with some codes showing an error floor.

2) Explanation of Error Floors: In both cases, the error floor has a simple cause. The most frequent error under these conditions is a reconstructed transmission which differs from the correct codeword in exactly three bits—the same three bits every time. These bits, which have weight 3 columns in the parity check matrix, are connected to just five checks with the topology shown below



(1)

If the three bits shaded grey are flipped into the wrong state, then the syndrome vector changes sign in the fifth check only. The sum–product algorithm is unable to extricate itself from this state. As the block length of the code is increased, the probability of this topology's occurrence falls. It is also possible to modify the construction algorithm for Gallager codes such that cycles of length 4, like this are forbidden (as

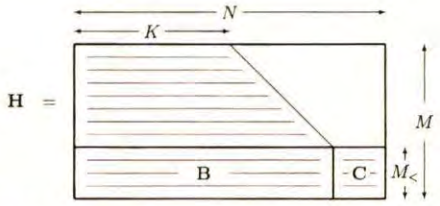


(2)

in construction 1A of [8]). This modification is sufficient to prevent the topology shown in (1) from occurring. In principle, it is possible for a code to have a minimum distance of 4 even when the minimum cycle length is 6. However, for randomly constructed codes, the minimum distance increases linearly with the blocklength, for almost all codes [4].

We discard the two codes with error floors in the subsequent comparisons.

3) Comparison of Constructions: The six families are



Encoding procedure:

Bits $t_1 \dots t_K$ are defined to be source bits.

Bits $t_{K+1} \dots t_{N-M_<}$ are set in sequence, using the mth parity check to determine t_{K+m}. This costs $(N-M_<)t_r$ computational operations, where t_r is the typical weight per row.

Bits $t_{N-M_<+1} \dots t_N$ are set equal to:

$$\mathbf{C}^{-1}\mathbf{B}\mathbf{t}^* \bmod 2$$

where $\mathbf{t}^* = (t_1 \dots t_{N-M_<})^T$ and $\mathbf{C}^{-1}$ is the inverse of $\mathbf{C}$ in modulo 2 arithmetic. $\mathbf{C}^{-1}$ can be stored in $M_<^2$ bits of memory. The product $\mathbf{B}\mathbf{t}^*$ can be computed in $M_< t_r$ computational operations, and the multiplication by $\mathbf{C}^{-1}$ takes $M_<^2$ operations.

Fig. 5. Upper panel: general form of a fast-encoding Gallager code. Horizontal stripes indicate low-weight rows. The diagonal line is a line of 1's. The matrices $\mathbf{B}$ and $\mathbf{C}$ are of dimension $M_< \times (N - M_<)$ and $M_< \times M_<$, respectively. Lower panel: the fast encoding method to generate a codeword and its computational cost, assuming an appropriate representation of the sparse matrix.

clear ranking of the other constructions, as follows:

$$3 < 93a < 93p < 93x < 93y.$$

Thus, we find that at least for the 93 profile, sub-Poisson constructions are inferior to Poisson constructions, and super-Poisson constructions are significantly superior. In the case of 93y, we see an improvement of about 0.05 dB.

4) Decoding Times: Not only do these irregular codes outperform the regular codes, they require fewer iterations as illustrated in Fig. 4(a), which compares the median number of iterations of the irregular code 93y and the regular code 33. Note that 93y requires 7/6 times more operations per iteration due to the increased weight of the matrix, so the total decoding times are similar.

Fig. 4(b) and (c) shows that the distribution of decoding times is heavy tailed. At $E_b/N_0 = 1.4$, the tail is well approximated by the power law: $P(\tau) \sim \tau^{-8.5}$, where τ is

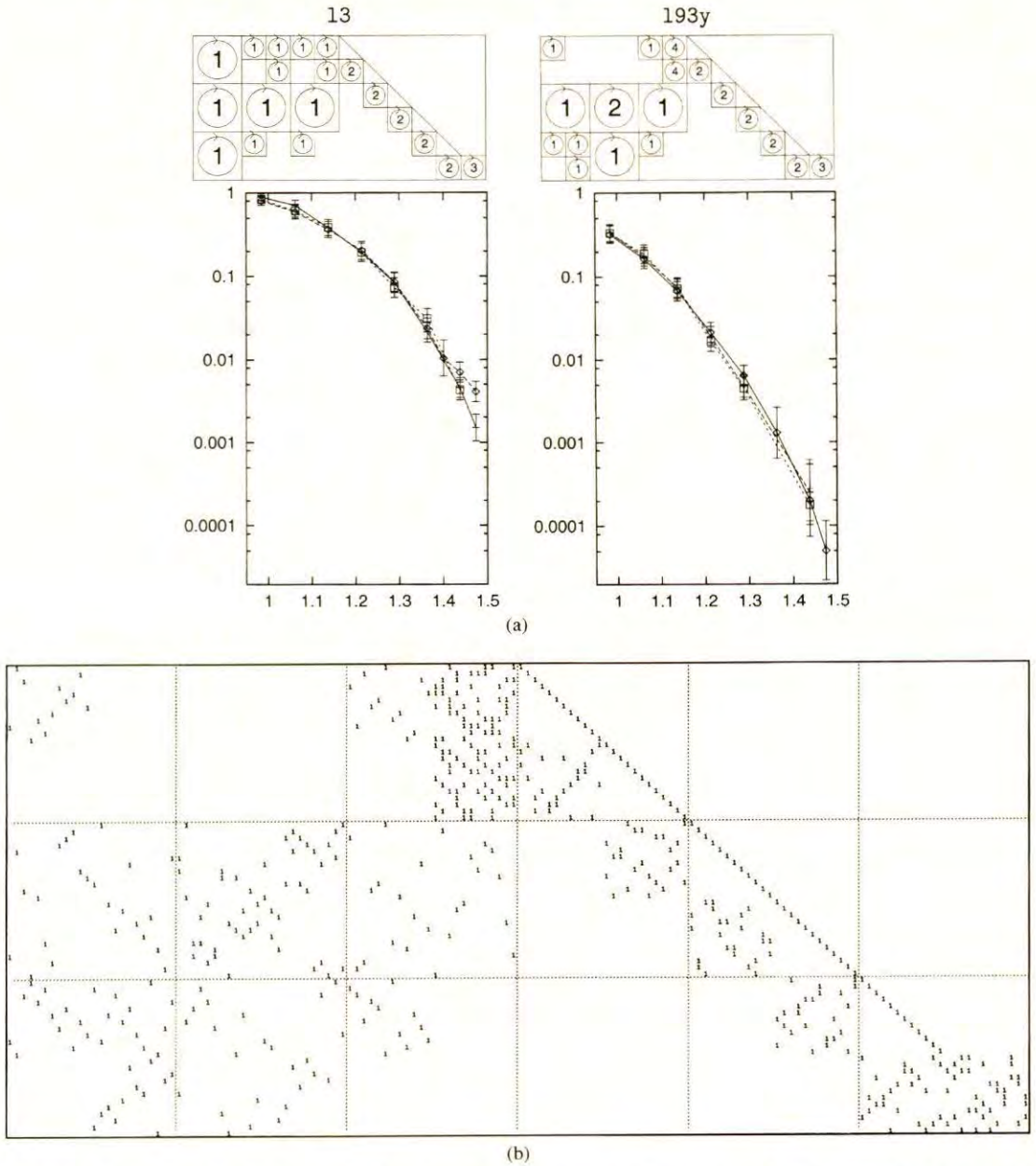


Fig. 6. (a) Upper panels: construction methods 13 and 193y. As in Fig. 2, an integer represents a number of permutation matrices superposed on the surrounding square. A diagonal line represents a line of 1's. Horizontal and vertical lines indicate the boundaries of the permutation blocks. Lower picture: variability of performance among 13 and 193y codes. Vertical axis: block error probability. Horizontal axis: E_b/N_0 in dB. All codes have $N = 9972$ and $K = M = 4986$. (b) Example of a parity check matrix with $N = 144$ made using construction 193y.

5) Unequal Error Protection: We can compare the bit-error rate of elite bits with that of standard bits. We find that when decoding fails, elite bits are more likely than standard bits to be correctly decoded. In the case of construction 93x, we found that at $E_b/N_0 = 1.3$ dB, the probability of an elite bit being in error, given that the block was incorrectly decoded, was 0.012 whereas standard bits had an error rate of 0.065. Differences remain at small E_b/N_0. For example, at $E_b/N_0 = 0.7$ dB, the error rates are 0.035 and 0.097.

IV. Fast-Encoding Gallager Codes

One of the possible drawbacks of Gallager codes is that their codes whose profiles are similar to or identical to the 3 and 93 profiles above, but which are fast-encoding. The general form of parity check matrix for a fast-encoding Gallager code is shown in Fig. 5. The parity check matrix has an almost lower-triangular structure which allows all but a small number $M_<$ of the parity bits to be computed using sparse operations. The final $M_<$ parity bits can be computed in $M_<^2$ binary operations. If $M_<$ were as small as $\sqrt{M}$, then the codes would be linear-time encodable.

We introduce two constructions, 13 and 193y ("1" for linear), shown diagrammatically in Fig. 6(a). Construction 13 has profile identical to construction 3. Construction 193y has

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.