# Transactions Letters

## Random Redundant Iterative Soft-in Soft-out Decoding

Thomas R. Halford, *Member, IEEE*, and Keith M. Chugg, *Member, IEEE*

***Abstract*—This letter presents an iterative soft-in soft-out (SISO) decoding algorithm based on redundant Tanner graphs that is applicable to arbitrary linear block codes. The proposed algorithm utilizes the permutation group of a code in order to efficiently and randomly generate redundant parity-checks.**

***Index Terms*—Iterative decoding, linear block codes, permutation groups, redundant Tanner graphs.**

### I. Introduction

WHEREAS modern codes are designed with cyclic graphical models in mind, classical linear block codes by and large were not. Graphical models of such codes which imply decoding algorithms with desired performance and complexity characteristics must thus be sought. The Tanner graphs [1] that are used to decode low-density parity-check (LDPC) codes are a natural starting point in the search for good graphical models of classical linear block codes. Although Tanner graphs imply very low-complexity decoding algorithms, most classical linear block codes are defined by high-density, rather than low-density, parity-check matrices and the performance of the decoding algorithms implied by these models is poor. Specifically, a recent result demonstrates that the Tanner graphs corresponding to many classical linear block codes *necessarily* contain cycles of length four [2] and it has been observed empirically (cf., [3], [4]) that four-cycles can be detrimental to Tanner graph decoding performance.

A number of authors have considered SISO decoding algorithms for linear block codes that utilize redundant Tanner graphs. Jiang and Narayanan developed a SISO decoding algorithm for cyclic linear block codes that effectively decodes on the graph implied by a different parity-check matrix at each decoding iteration by randomly cyclicly shifting soft information [5]. Kothiyal *et al.* developed a SISO decoding algorithm for arbitrary linear block codes that decodes on the graph implied by a different parity-check matrix at every iteration which is chosen adaptively in order to minimize the propagation of unreliable soft information [6]. Related algorithms were presented in [7], [8]. Furthermore, a number of authors have considered redundant graphical models in other contexts such as the iterative decoding of one-step majority logic decodable codes [9], LDPC decoding [10], [11], [12], pseudo-noise sequence acquisition [13], and iterative decoding on the binary erasure channel [14].

This work present a novel redundant Tanner graph SISO decoding algorithm - random redundant iterative decoding (RRD) - that is both practically realizable and applicable to arbitrary linear block codes. The remainder of this work is organized as follows. Section II defines redundant Tanner graphs and describes the iterative decoding algorithms implied by such models. Section III presents the proposed RRD algorithm. The performance of the proposed algorithm is examined empirically in Section IV. Concluding remarks are given in Section V.

### II. Redundant Tanner Graphs

Let $\mathcal{C}$ be an $[n,k,d]$ linear block code with $r \geq n-k \times n$ parity-check matrix $H=[h_{ij}]$. The codes considered in this work are binary; however, the proposed algorithm can be applied to codes over arbitrary fields. Associated with $H$ is the bipartite Tanner graph $\mathrm{TG}(H)=(\mathcal{U}\cup\mathcal{W},\mathcal{E})$. Vertices in $\mathcal{U}=\{u_i\}_{i=0}^{r-1}$ represent the single parity-check constraints (SPCs) corresponding to the rows of $H$. Vertices in $\mathcal{W}=\{w_j\}_{j=0}^{n-1}$ represent the repetition constraints (RCs) corresponding to the columns of $H$. An edge in $\mathcal{E}$ connects $u_i$ to $w_j$ if and only if $h_{ij}=1$. If $r$ is strictly greater then $n-k$ then $H$ is a *redundant* parity-check matrix and the corresponding Tanner graph is denoted redundant. This work focuses specifically on redundant parity-check matrices where $r$ is a multiple of $n-k$, $r=m(n-k)$; the Tanner graph corresponding to such a matrix is denoted a *degree-$m$ redundant Tanner graph*.

As an example, consider the $[8,4,4]$ extended Hamming code $\mathcal{C}_8$ which can be represented equivalently by the two parity-check matrices:

$$H_1=\begin{bmatrix}1&1&1&1&0&0&0&0\\0&0&1&1&1&1&0&0\\0&0&0&0&1&1&1&1\\0&1&1&0&0&1&1&0\end{bmatrix},\qquad H_2=\begin{bmatrix}0&1&0&1&0&1&0&1\\1&1&0&0&0&0&1&1\\1&0&1&0&1&0&1&0\\0&1&0&1&1&0&1&0\end{bmatrix}. \tag{1}$$

A redundant parity-check matrix for $\mathcal{C}_8$ can be formed by concatenating the rows of $H_1$ and $H_2$ to form $H_R$. The corresponding degree-2 redundant Tanner graph $\mathrm{TG}(H_R)$ is illustrated in Figure 1 where Forney's normal graph convention has been adopted [15]. Degree-$m$ redundant Tanner graphs imply the following standard iterative decoding algorithm (note that the message passing schedule described below is reasonable but not unique). For each set of checks $H_i$,

Paper approved by C. Schlegel, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received March 15, 2006; revised November 15, 2006. This paper was presented in part at IEEE International Symposium on Information Theory, Seattle, WA, July 2006.

T. R. Halford was with the University of Southern California. He is now with TrellisWare Technologies Inc., Glendale, CA (e-mail: thalford@trellisware.com).

K. M. Chugg is with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA (e-mail: chugg@usc.edu).

Digital Object Identifier 10.1109/TCOMM.2008.060105.

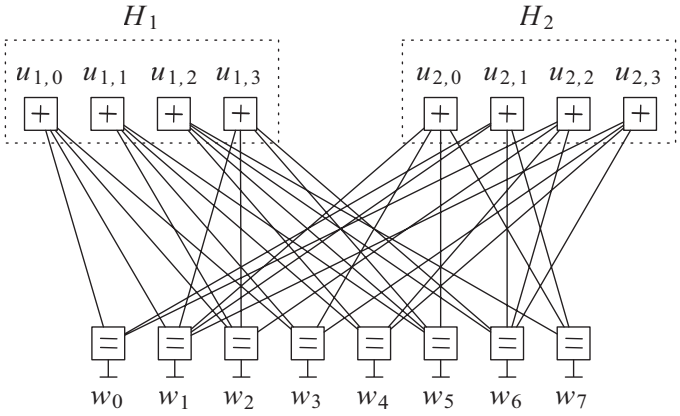


Fig. 1. Degree-2 redundant Tanner graph for the $[8, 4, 4]$ extended Hamming code corresponding to the concatenation of the rows of $H_1$ and $H_2$. Single parity-scheck constraints are illustrated by square vertices labeled with '+' symbols and the repetition constraints corresponding to codeword bits are represented by square vertices labeled with '=' symbols.

$i = 1, \ldots, m$, denote by $\mathbf{M}\downarrow_i$ the vector of messages passed from checks to variables and denote by $\mathbf{M}\uparrow_i$ the vector of messages passed from variables to checks. Note that for all $i = 1, \ldots, m$, $\mathbf{M}\downarrow_i$ is initialized to zero[1]. For $i = 1, \ldots, m$, the $\mathbf{M}\uparrow_i$ messages are first updated at the RCs using the channel observations and the $\mathbf{M}\downarrow_j$ messages (for $j \neq i$). The $\mathbf{M}\downarrow_i$ messages are then updated at the SPCs using $\mathbf{M}\uparrow_i$. This updating procedure repeats for a prescribed number of iterations, $I$. Note that this schedule can be modified so that a number of decoding iterations are performed on each set of checks before proceeding to the next.

### A. Practical Implementation via Massive Redundancy and Permutation Groups

From the viewpoint of practical low-complexity implementation, the decoding algorithm described above suffers from two drawbacks:

1) For the standard message passing rules, the intermediate messages vectors $\mathbf{M}\downarrow_i$ must be stored for $i = 1, \ldots, m$ resulting in an $m$-fold increase of the memory required with respect to standard Tanner graph decoding.
2) Since each parity-check matrix $H_i$ defines a different set of checks, there is either an $m$-fold increase in the number of SPC trellises which must be implemented or the SPC trellises must be implemented in a reconfigurable fashion.

The first drawback may be addressed by using a massively redundant Tanner graph. If a large degree of redundancy is used then $I$ (the number of decoding iterations on the aggregate model) can be set to one and the intermediate message vectors $\mathbf{M}\downarrow_i$ need not be stored. Before addressing the second drawback, we review permutation groups of codes.

Let $\mathcal{C}$ be a block code of length $n$. The *permutation group* of $\mathcal{C}$, $\mathrm{Per}(\mathcal{C})$, is the set of permutations of coordinate places which send $\mathcal{C}$ onto itself [16]. The permutation groups of many classical linear block codes are well-understood (cf., [16], [17]) and for codes with short block lengths, permutation groups can be obtained using the methods described in [18], [19].

Returning to the $[8, 4, 4]$ extended Hamming code, it can be shown that [16][2]:

$$\boldsymbol{\sigma} = (5, 3, 1, 7, 0, 6, 4, 2) \in \mathrm{Per}(\mathcal{C}_8) \tag{2}$$

It is readily verified that applying $\boldsymbol{\sigma}$ to the columns of $H_1$ (as defined in (1)) yields $H_2$ so that $H_2 = \boldsymbol{\sigma} H_1$. In light of this example, it is clear that redundant parity-checks for a given code $\mathcal{C}$ can be generated by applying permutations drawn from $\mathrm{Per}(\mathcal{C})$ to the columns of some initial parity-check matrix $H$. Observe that decoding with soft-input vector $\mathbf{SI}$ on $\mathrm{TG}(\boldsymbol{\beta} H)$ (where $\boldsymbol{\beta} \in \mathrm{Per}(\mathcal{C})$) is equivalent to decoding with soft-input vector $\boldsymbol{\beta}^{-1}\mathbf{SI}$ on $\mathrm{TG}(H)$. It is this observation that allows for the efficient implementation of redundant Tanner graph decoding: provided that the redundant parity-checks are column permuted versions of some base matrix $H$, redundant Tanner graph decoding can be implemented by permuting soft information vectors and decoding with a *constant* set of constraints.

### B. Redundant Tanner Graphs and Previous Work

From the above discussion, it is apparent that Kothiyal *et al.*'s adaptive belief propagation (ABP) algorithm [6] and Jiang and Narayanan's stochastic shifting based iterative decoding (SSID) algorithm [5] are redundant Tanner graph decoding algorithms. Kothiyal *et al.*'s scheme adaptively chooses new parity-check sets based on soft information reliability. Although the ABP algorithm can be applied to arbitrary block codes, it does not imply a practical low-complexity implementation because the check sets change with every iteration. Furthermore, the ABP algorithm requires the computationally expensive Gaussian elimination of potentially large parity-check matrices at every iteration. Jiang and Narayanan's scheme is an example of a practical, low-complexity redundant Tanner graph decoding algorithm for cyclic codes which uses the permutation group approach described above. The random redundant decoding algorithm proposed in this work is, in fact, an extension of Jiang and Narayanan's algorithm to arbitrary block codes with a known permutation group.

## III. Proposed Decoding Algorithm

Algorithm 1 describes the proposed decoding algorithm. The inner **for**-loop of Algorithm 1 describes an efficient redundant Tanner graph decoding algorithm with the addition of a damping coefficient $\alpha$. The outer **for**-loop of Algorithm 1 iterates over different values of $\alpha$. By varying the damping coefficient $\alpha$, the algorithm avoids local minima in the solution space. Many authors have considered the introduction of such damping coefficients in iterative soft decoding algorithms to achieve this end (cf., [20]). For practical implementations where a large number of outer iterations is undesirable from a

[1]Throughout this work, decoding is assumed to be performed in the $-\log(\cdot)$ domain, i.e., either min-sum or min*-sum processing is assumed.

[2]Throughout this work, permutations of $n$ coordinate places (indexed from 0) are described by $n$-tuples. For example, the application of the permutation $(1, 4, 0, 2, 3, 6, 5)$ to a 7 bit codeword $(c_0, c_1, c_2, c_3, c_4, c_5, c_6)$ yields the permuted codeword: $(c_2, c_0, c_3, c_4, c_1, c_6, c_5)$. The identity permutation is denoted $\boldsymbol{\epsilon}$ and the inverse of a permutation $\boldsymbol{\beta}$ is denoted $\boldsymbol{\beta}^{-1}$.

time complexity standpoint, a single damping coefficient (or a small set of coefficients) could be used depending on the operating noise power.

Algorithm 1 takes as input a received soft information vector, $\mathbf{SI}$, a parity-check matrix for the code, $H$, and four parameters:

- $\alpha_0$: The initial damping coefficient.
- $I_1$: The number of Tanner graph decoding iterations to perform per inner iteration.
- $I_2$: The maximum number of inner iterations to perform per outer iteration. Each inner iteration considers a different random permutation of the codeword elements.
- $I_3$: The maximum number of outer iterations to perform. Each outer iteration uses a different damping coefficient.

Let $\mathbf{s}$ be the sum of the input soft information $\mathbf{SI}$ and the output soft information produced by all previous inner iterations. During the $i_2$-th inner iteration, $I_1$ Tanner graph decoding iterations are performed on $\mathrm{TG}(H)$ with damping coefficient $\alpha$ and soft input $\mathbf{s}$ producing the soft output vector $\mathbf{s}'$ and hard decision $\mathbf{c}'$. The cumulative soft information vector $\mathbf{s}$ is then updated to include $\mathbf{s}'$. The inner iteration concludes by applying a random permutation $\boldsymbol{\theta}$ from the permutation group of the code to $\mathbf{s}$. Decoding concludes when either a valid codeword is returned by the Tanner graph decoding step or when a maximum number of iterations is reached. Before returning the final soft output and hard decision vectors, the random permutations are undone by applying the inverse of the product of the permutations that were applied to $\mathbf{s}$.

### A. Initial Parity-Check Matrix Selection

It was observed empirically that the performance of the proposed decoding algorithm depends heavily on the choice of parity-check matrix (and thus Tanner graph) used to represent the code. It is widely accepted that the performance of the decoding algorithms implied by Tanner graphs are adversely affected by short cycles (cf., [3], [21]). Algorithm 2 searches for a suitable parity-check matrix by greedily performing row operations on an input binary parity-check matrix $H$ in order to reduce the number of short cycles contained in the Tanner graph defined by $H$. The operation of Algorithm 2 requires that short cycles in bipartite graphs can be counted efficiently; such an algorithm was described in [3].

### B. Generation of Random Permutation Group Elements

Algorithm 1 requires the efficient generation of random elements of the permutation group of a code. Cellar *et al.* presented an algorithm for generating random elements of an arbitrary finite group in [22]. Briefly, suppose that $G$ is a finite group with size $k$ generating set[3]:

$$X = \{x_0, x_1, \ldots, x_{k-1}\}. \tag{3}$$

Cellar *et al.*'s *product-replacement* algorithm constructs a vector $\mathbf{S} = (s_0, s_1, \ldots, s_{N-1})$ of length $N > k$ containing all of the elements of $X$ with repeats. The basic operation of this algorithm is to randomly choose two elements of $\mathbf{S}$, $s_i$ and $s_j$, and to replace $s_j$ by the product of $s_i$ and $s_j$. Generation of random group elements is initialized by performing this basic operation $K$ times. After initialization, successive basic operations yield random elements of $G$ by returning the updated value of $s_j$. Note that the execution of this basic operation requires the generation of two random integers and only one group multiplication and is thus efficient (permutation multiplication is particularly easy). Also note that after every execution, the elements contained in $\mathbf{S}$ generate $G$. Cellar *et al.* found that setting $N = \max(2k + 1, 10)$ and $K = 60$ provides near-uniform random generation of group elements in practice.

[3]That is, every element $g \in G$ can be expressed as a finite product $g = x_{i_1}^{n_1} x_{i_2}^{n_2} \cdots x_{i_t}^{n_t}$ where $x_{i_j} \in X$ and $n_j \in \mathbb{N}$ (the set of natural numbers) for all $j$.

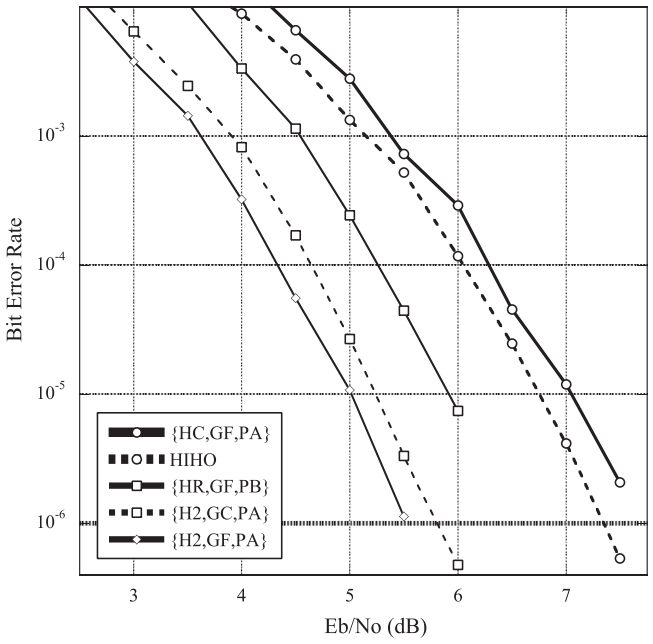


Fig. 2. Bit error rate performance comparison of different decoding algorithms for the $[63, 39, 9]$ BCH code. Binary antipodal signaling on an additive white Gaussian noise channel is assumed.

## IV. Simulation Results

The performance of the proposed algorithm is investigated empirically in this section for the $[63, 39, 9]$ BCH code $\mathcal{C}_{63}$. The performance of Algorithm 1 was studied using three choices of input parity-check matrix (labeled HC, H2, and HR), two choices of permutation (sub)group (labeled GF and GC), and two choices of parameter sets (labeled PA and PB) all of which are defined below. Figure 2 compares the bit error rate (BER) performance of four combinations of these choices to algebraic hard-in hard-out (HIHO) decoding.

- HC: The standard $24 \times 63$ cyclic parity-check matrix for $\mathcal{C}_{63}$ (cf., [16]). The corresponding Tanner graph contains $32,625$ four-cycles and $6,981,190$ six-cycles.
- H2: The output of Algorithm 2 when HC is used as input. The corresponding Tanner graph contains $3,162$ four-cycles and $212,301$ six-cycles.
- HR: A $1200 \times 63$ redundant parity-check matrix generated by concatenating $50$ column-permuted versions of H2 (using permutations drawn randomly from $\mathrm{Per}(\mathcal{C}_{63})$).

**Input**: Length $n$ soft-input vector $\mathbf{SI}$.
$n-k \times n$ binary parity-check matrix $H$.
Parameters $I_1$, $I_2$, $I_3$, $\alpha_0$.
**Output**: Length $n$ soft-output vector $\mathbf{SO}$.
Length $n$ hard-decision vector $\mathbf{HD}$.

$\alpha \leftarrow \alpha_0$;
**for** $1 \leq i_3 \leq I_3$ **do**
  $\boldsymbol{\Theta} \leftarrow \boldsymbol{\epsilon}$;
  $\mathbf{s} \leftarrow \mathbf{SI}$;
  **for** $1 \leq i_2 \leq I_2$ **do**
    Perform $I_1$ decoding iterations of $\mathbf{s}$ on $\mathrm{TG}(H)$ with damping coefficient $\alpha$ and place soft output in $\mathbf{s}'$ and resulting hard decision in $\mathbf{c}'$;
    $\mathbf{s} \leftarrow \mathbf{s} + \mathbf{s}'$;
    **if** $H\mathbf{c}' = 0$ **then**
      Apply $\boldsymbol{\Theta}^{-1}$ to $\mathbf{s}$ and $\mathbf{c}'$;
      $\mathbf{SO} \leftarrow \mathbf{s} - \mathbf{SI}$;
      $\mathbf{HD} \leftarrow \mathbf{c}'$;
      **return** $\mathbf{SO}$ *and* $\mathbf{HD}$
    **end**
    $\boldsymbol{\theta} \leftarrow$random element of $\mathrm{Per}(\mathcal{C})$;
    Apply $\boldsymbol{\theta}$ to $\mathbf{s}$;
    $\boldsymbol{\Theta} \leftarrow \boldsymbol{\theta}\boldsymbol{\Theta}$;
  **end**
  $\alpha \leftarrow \alpha_0 + (1-\alpha_0)\frac{i_3}{I_3-1}$;
**end**

**Algorithm 1**: Random redundant iterative decoding.

GF: The full permutation group $\mathrm{Per}(\mathcal{C}_{63})$ of $\mathcal{C}_{63}$ which is generated by the 6 permutations:

$$\boldsymbol{\sigma}^{(j)} = \left(1, 2^j+1, 2\cdot 2^j+1, \ldots, 62\cdot 2^j+1\right) \quad (4)$$

for $0 \leq j \leq 5$ (where each permutation element is taken modulo $2^m-1$) [17].

GC: The cyclic subgroup of $\mathrm{Per}(\mathcal{C}_{63})$ generated by $\boldsymbol{\sigma}^{(0)}$ alone.

PA: $\alpha_0 = 0.08$, $I_1 = 2$, $I_2 = 50$, and $I_3 = 20$.

PB: $\alpha_0 = 0.08$, $I_1 = 2$, $I_2 = 1$, and $I_3 = 20$.

Optimal SISO decoding of $\mathcal{C}_{63}$ is intractable in practice. Specifically, the least complex known optimal SISO decoding algorithm for this code is obtained by decoding on a sixteen stage trellis for the dual of an extension of $\mathcal{C}_{63}$ containing $2^{15}$ states [23]. Random redundant decoding with {H2,GF,PA}, which is suboptimal yet tractable in practice, is shown to outperform algebraic HIHO decoding by approximately 1.75 dB at a BER of $10^{-6}$ in Figure 2. Figure 2 demonstrates the sensitivity of the perfomance of random redundant decoding to the choice of initial parity-check matrix: the {H2,GF,PA} decoder outperforms the {HC,GF,PA} decoder by approximately 2 dB at a BER of $10^{-5}$. The {H2,GF,PA} decoder outperforms the {H2,GC,PA} decoder by approximately 0.25 dB at a BER of $10^{-6}$. The former considers all possible permutations in $\mathrm{Per}(\mathcal{C}_{63})$, rather than only those corresponding to cyclic shifts, and is in some sense more random than the latter, which is equivalent to Jiang and Narayanan's algorithm [5]. Note finally that although the {H2,GF,PA} and {HR,GF,PB} decoders can be viewed as different message-passing schedules on the same graphical model, the former outperforms the latter by approximately 0.9 dB at a BER of $10^{-5}$. We attribute the performance difference to the fact that the flooding schedule used in the {HR,GF,PB} decoder does not allow for the full exploitation of the gains that can be realized by damping thus indicating that the performance of the proposed random redundant decoding algorithm may be attributed to both the use of redundancy and the use of damping to slow convergence.

**Input**: $r \times n$ binary parity-check matrix $H$.
**Output**: $r \times n$ binary parity-check matrix $H'$.

$H' \leftarrow H$, $r_1^\star \leftarrow -1$, $r_2^\star \leftarrow -1$, $g^\star \leftarrow$ girth of $\mathrm{TG}(H)$;
$N_{g^\star}^\star \leftarrow$ number of $g^\star$-cycles in $\mathrm{TG}(H')$;
$N_{g^\star+2}^\star \leftarrow$ number of $g^\star+2$-cycles in $\mathrm{TG}(H')$;
**repeat**
  **if** $r_1^\star \neq r_2^\star$ **then** Replace row $r_2^\star$ in $H'$ with binary sum of rows $r_1^\star$ and $r_2^\star$;
  $r_1^\star \leftarrow -1$, $r_2^\star \leftarrow -1$;
  `/* Greedily find the row operation which reduces the most short cycles. */`
  **for** $r_1, r_2 = 0, \ldots, r-1$, $r_2 \neq r_1$ **do**
    Replace row $r_2$ in $H'$ with binary sum of rows $r_1$ and $r_2$;
    $g \leftarrow$ girth of $\mathrm{TG}(H')$;
    $N_g \leftarrow$ number of $g$-cycles in $\mathrm{TG}(H')$;
    $N_{g+2} \leftarrow$ number of $g+2$-cycles in $\mathrm{TG}(H')$;
    **if** $g > g^\star$ **then**
      $g^\star \leftarrow g$, $r_1^\star \leftarrow r_1$, $r_2^\star \leftarrow r_2$, $N_g^\star \leftarrow N_g$, $N_{g+2}^\star \leftarrow N_{g+2}$;
    **end**
    **else if** $g = g^\star$ *AND* $N_g < N_g^\star$ **then** $r_1^\star \leftarrow r_1$, $r_2^\star \leftarrow r_2$, $N_g^\star \leftarrow N_g$, $N_{g+2}^\star \leftarrow N_{g+2}$;
    **else if** $g = g^\star$ *AND* $N_g = N_g^\star$ **then**
      **if** $N_{g+2} < N_{g+2}^\star$ **then** $r_1^\star \leftarrow r_1$, $r_2^\star \leftarrow r_2$, $N_{g+2}^\star \leftarrow N_{g+2}$;
    **end**
    Undo row replacement;
  **end**
**until** $r_1^\star = -1$ & $r_2^\star = -1$;
**return** $H'$

**Algorithm 2**: Greedy Tanner graph cycle reduction.

## V. Conclusion

This letter introduced random redundant iterative decoding of linear block codes which can be viewed as a generalization of Jiang and Narayanan's algorithm [5] for cyclic codes to arbitrary codes. The proposed algorithm represents a practically realizable redundant Tanner graph decoding algorithm which can be applied to any linear code with a known permutation group. An interesting direction for future work is the application of the proposed algorithm to codes which are not classical block codes, e.g., short blocklength LDPC codes with algebraic constructions which afford the ready determination of their permutation groups.

## REFERENCES


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

[2] T. R. Halford, A. J. Grant, and K. M. Chugg, "Which codes have 4-cycle-free Tanner graphs?" *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 4219–4223, Sept. 2006.

[3] T. R. Halford and K. M. Chugg, "An algorithm for counting short cycles in bipartite graphs," *IEEE Trans. Inform. Theory*, vol. 52, no. 1, pp. 287–292, Jan. 2006.

[4] L. Lan, Y. Y. Tai, L. Chen, and K. Abdel-Ghaffer, "A trellis-based method for removing cycles from bipartite graphs and construction of low density parity check codes," *IEEE Commun. Lett.*, vol. 8, no. 7, pp. 443–445, July 2004.

[5] J. Jiang and K. R. Narayanan, "Iterative soft decision decoding of Reed-Solomon codes," *IEEE Commun. Lett.*, vol. 8, no. 4, pp. 244–246, Apr. 2004.

[6] A. Kothiyal, O. Y. Takeshita, W. Jin, and M. Fossorier, "Iterative reliability-based decoding of linear block codes with adaptive belief propagation," *IEEE Commun. Lett.*, vol. 9, no. 12, pp. 1067–1069, Dec. 2005.

[7] J. Jiang and K. R. Narayanan, "Iterative soft decision decoding of Reed-Solomon codes based on adaptive parity check matrices," in *Proc. IEEE International Symp. on Information Theory*, Chicago, Il., June 2004, p. 261.

[8] A. Kothiyal and O. Y. Takeshita, "A comparison of adaptive belief propagation and the best graph algorithm for the decoding of linear block codes," in *Proc. IEEE International Symp. on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 724–728.

[9] R. Lucas, M. P. C. Fossorier, Y. Kou, and S. Lin, "Iterative decoding of one-step majority logic decodable codes based on belief propagation," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 931–937, June 2000.

[10] D. J. C. MacKay and R. M. Neal, *Codes, Systems and Graphical Models*. Springer-Verlag, 2000, ch. Evaluation of Gallager codes for short block lengths and high rate applications, pp. 113–130, eds.: B. Marcus and J. Rosenthal.

[11] Y. Kou, J. Xu, H. Tan, S. Lin, and K. Abdel-Ghaffar, "On circulant low density parity check codes," in *Proc. IEEE International Symp. on Information Theory*, Lausanne, Switzerland, June 2002, p. 200.

[12] K. Andrews, S. Dolinar, and F. Pollara, "LDPC decoding using multiple representations," in *Proc. IEEE International Symp. on Information Theory*, Lausanne, Switzerland, July 2002, p. 456.

[13] O. Yeung and K. M. Chugg, "An iterative algorithm and low complexity hardware architecture for fast acquisition of PN codes in UWB systems," *Springer J. VLSI Signal Processing (special issue on ultrawideband systems)*, vol. 43, no. 1, Apr. 2006.

[14] M. Schwartz and A. Vardy, "On the stopping distance and the stopping redundancy of codes," in *Proc. IEEE International Symp. on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 975–979.

[15] G. D. Forney, Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inform. Theory*, pp. 520–548, Feb. 2001.

[16] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1978.

[17] C.-C. Lu and L. R. Welch, "On automorphism groups of binary primitive BCH codes," in *Proc. IEEE International Symp. on Information Theory*, Trondheim, Norway, June 1994, p. 51.

[18] J. Leon, "Computing automorphism groups of error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 28, no. 3, pp. 496–511, May 1982.

[19] N. Sendrier and G. Skersys, "On the computation of the automorphism group of a linear code," in *Proc. IEEE International Symp. on Information Theory*, Washington, DC, June 2001, p. 13.

[20] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406–414, Mar. 2002.

[21] Y. Mao and A. H. Banihashemi, "A heuristic search for good low-density parity-check codes at short block lengths," in *Proc. International Conf. Communications*, vol. 1, Helsinki, Finland, June 2001, pp. 41–44.

[22] F. Celler, C. R. Leedham-Green, S. H. Murray, A. C. Niemeyer, and E. A. O'Brien, "Generating random elements of a finite group," *Commun. in Algebra*, vol. 23, pp. 4931–4948, 1995.

[23] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 1057–1064, May 1993.