On the Hardness of Approximating Stopping and Trapping Sets in LDPC Codes

Andrew McGregor
Center for Information Theory and Applications
University of California, San Diego
andrewm@ucsd.edu

Olgica Milenkovic
Dept. of Electrical and Computer Engineering
University of Colorado, Boulder
milenkov@colorado.edu

***Abstract*— We prove that approximating the size of the smallest trapping set in Tanner graphs of linear block codes, and more restrictively, LDPC codes, is NP-hard. The proof techniques rely on reductions from three NP-hard problems, the set cover, minimum three-dimensional matching, and the minimum Hamming distance problem. The ramifications of our findings are that methods used for estimating the height of the error-floor of long LDPC codes, centered around trapping set enumeration, cannot provide accurate worst-case performance predictions.**

I. Introduction

In the past decade, the search for efficient and near-optimal decoding algorithms for linear block codes culminated with the rediscovery and generalization of the notion of sparse codes and iterative message passing algorithms. Iterative decoders can approach the Shannon limit of reliable communication with polynomial time complexity, provided that they operate on long, low-density parity-check (LDPC) codes. This is achieved by using graphical representations of codes, termed Tanner graphs, that contain a small number of edges. On such graphs, probabilistic inference of the form of iterative message passing is known to have linear complexity.

The performance of linear block codes under iterative decoding, and the performance of LDPC codes in particular, depends on the structural properties of their underlying Tanner graphs. For each channel-decoder pair, there exist vertex configurations in the code graph on which a given iterative decoder fails. For some frequently encountered Discrete Memoryless Channels (DMCs), such configurations are known as *near-codewords* [9], *trapping and stopping sets* [3], [12], *pseudocodewords* [7], or *instantons* [13].

It is known that maximum-likelihood (ML) decoders fail when transmission errors are confined to Tanner graph configurations corresponding to codewords, while iterative decoders usually fail to make correct decisions on (strictly) larger sets of configurations. For example, iterative edge-removal (ER) decoders for coding over the Binary Erasure Channel (BEC) fail on stopping sets [3], a subset of which are the codewords themselves. For the Additive White Gaussian Noise (AWGN) channel and sum-product decoding, failures arise due to subsets of vertices in the code graph termed *near-codewords*.

As a result, iterative decoders exhibit sub-optimal performance compared to ML decoders, and this performance loss most frequently manifests itself in terms of the appearance of *error-floors* in the Bit-Error-Rate (BER) curve of the code.

The error-floor phenomena is a problem of focal importance in the theory of iterative decoding, since many practical applications of codes on graphs require extremely low operational BERs. Since such low BERs are well beyond the scope of current Monte-Carlo simulation techniques, several methods were proposed for estimating the height of the error-floor based on enumerating small trapping sets and exploring dominant instantons [12], [13]. These techniques perform fairly accurately for codes of short and moderate length, but they are time consuming, and no rigorous analytical study of the termination criteria for the search procedure is known.

Recently, it was shown that the problem of finding the smallest stopping set in an arbitrary code graph is NP-hard to approximate up to a constant term [10]. In [17], it was shown that finding the smallest k-*out set*, which represents a straightforward generalization of the notion of a stopping set, is NP-hard as well. The results in [17] do not capture the fact that trapping sets are usually characterized in terms of two parameters. Furthermore, the notion of a trapping set is meaningful only in conjunction with a fixed decoding method. Finally, no hardness results for approximating k-out sets or more general trapping sets are currently known.

The main contribution of our work are three-fold. First, we improve upon the hardness results of approximating stopping sets, presented in [10]. Second, we provide a set of new results regarding the hardness of

finding the smallest trapping sets for the Zyablov-Pinsker (ZP) decoder [19], [18], and the product-sum decoder. The third, and most important result, asserts that these hardness results carry over to the case of LDPC code graphs.

The paper is organized as follows. Section II introduces the trapping set structures under investigation, as well as their corresponding decoding algorithms. Section III provides a brief overview of a class of NP-hard problems that are used in the reduction proofs of our main result. Section IV contains results regarding the hardness of approximating trapping sets, while Section V specializes these results for the class of sparse code graphs.

II. Problem Setup

Let H be a fixed parity-check matrix of $\mathcal{C}$, and let the bipartite graph $G = (L \cup R, E)$ be such that the columns of H are indexed by the *variable nodes* in $\mathcal{L}$, and the rows of H are indexed by *check nodes* in $\mathcal{R}$. For $i \in L$ and $j \in R$, $(i, j) \in E$ if and only if $H_{i,j} = 1$. For a set $S \subset L$, the notation $\Gamma(S)$ is reserved for the set of neighbors of S in R. The graph G is called the Tanner graph of $\mathcal{C}$, with parity-check matrix H. If the parity-check matrix of a code is sparse, the corresponding code is called a Low-Density Parity-Check (LDPC) code.

The messages passed between vertices of the Tanner graphs during iterative decoding depend on the particular choice of the transmission channel. For a detailed description of decoding procedures for signaling over the BEC channel (the well known edge removal (ER) algorithm), the BSC channel (the Zyablov-Pinsker (ZP) algorithm), and AWGN channel (the product-sum algorithm), the interested reader is referred to [3], [19], [18], [12].

Note that for the AWGN channel, and message-passing algorithms, no precise analytic characterization of all failure modes is known. Extensive computer simulations [12], [8] show that errors are usually confined to *near codewords*, that resemble codewords in so far that they result in a very small number of unsatisfied check equations. We focus our attention on three such configurations, defined below.

Definition 1 (BEC Stopping Sets): Given a bipartite graph $G = (L \cup R, E)$, we say $S \subset L$ is a *stopping-set* if the degree of each vertex in $\Gamma(S)$ in the induced graph G_S on $S \cup \Gamma(S)$ is at least two.

Definition 2 (BSC ZP-Trapping Sets): Let $G = (L \cup R, E)$ be a regular bipartite graph with left-degree ℓ. We say $S \subset L$ is a ZP trapping set if the induced graph G_S on $S \cup \Gamma(S)$ is such that all vertices in S are connected to less than $\ell - \lfloor (\ell - 1)/2 \rfloor$ odd degree vertices in G_S.

Definition 3 (AWGN Trapping Sets): Given a bipartite graph $G = (L \cup R, E)$ we say $S \subset L$ is an (a, b) trapping set T if $|S| = a$ and the induced graph on $S \cup \Gamma(S)$ is such that $\Gamma(S)$ has exactly b vertices of odd degree. Similarly, we say that $S \subset L$ is an elementary (a, b) trapping set T_e if b vertices in $\Gamma(S)$ have degree one, and $|\Gamma(S)| - b$ have degree two.

Definition 4 (AWGN Trapping Sets T_c): Given a bipartite graph $G = (L \cup R, E)$ we say $S \subset L$ is good if the induced graph G_S on $S \cup \Gamma(S)$ is such that the majority of vertices of G_S in $\Gamma(S)$ have even degree. A subset T_c of L is a *BSC Trapping Set* if T_c and $L \setminus T_c$ are both good.

We are concerned with the worst-case computational complexity of the following problems.

1) MinStop: Find the stopping set of minimum cardinality.
2) MinT_{ZP}: Find the ZP-trapping set of minimum cardinality.
3) MinT: For a given a, find a trapping set T with smallest value of the parameter b.
4) MinT_e: For a given b, find an elementary trapping set T_e with smallest value of the parameter a.
5) MinT_c: Find the T_c trapping set of minimum cardinality.

We show that all five problems described above are NP-hard. Furthermore, we show that is also NP-hard to approximate the optimal value for these problems, e.g., there exists a value $\alpha > 1$ such that if there existed a polynomial-time algorithm that always returned a value at most a factor α larger (or smaller in the case of a maximization problem) than the optimum value then $P = NP$. Approximation algorithms is an important area of study in theoretical computer science. See [16] for further details.

III. A Set of NP-hard Problems

Our proofs will be based on reductions from the NP-hard problems *Minimum-Set-Cover* [11], *Minimum-Distance* [14], [15], [4], and *Maximum-Three-Dimensional-Matching* problems [2]. We briefly describe these problems

1) MinSetCover: Given a set of sets $\mathcal{S} = \{S_1, \ldots, S_a\}$, find $\mathcal{S}' \subset \mathcal{S}$ of minimum cardinality such that $\cup_{S \in \mathcal{S}'} S = \cup_{S \in \mathcal{S}} S$. The problem is NP-hard and not approximable within $c \log |\cup_{S \in \mathcal{S}} S|$, for some constant $c > 0$. A special promise version of MinSetCover where $|S_i| \leq K$ is also NP-hard and Approximation-complete. We call this problem MinKSetCover.
2) Max3DMatch: Given a set $T \subset X \times X \times X$, find a set $S \subset T$ such that no elements in S agree in

any co-ordinate. The problem is Approximation-complete. Generalizations of this problem that involve the constraint that no element of X appears more than $r \geq 3$ times is also known to be Approximation-complete [5].

3) MAXLD: Given a code $\mathcal{C}$ specified by an $m \times n$ parity-check matrix H (we may assume H has linearly independent rows), a vector $s \in F_2^m$, and an integer $\omega > 0$, determine if there is a vector $x \in F_2^n$ with weight bounded from above by ω, such that $H\,x^T = s$.
4) MINCODEWORD: Given a code $\mathcal{C}$ specified by an $n \times k$ generator matrix M (we may assume M has linearly independent rows) find the smallest weight of a non-zero codeword. The problem is not approximable within an additive error that is linear in the block length of the code, n, and provided that $NP \subsetneq RQP$, not approximable within the factor $2^{\log^{1-\epsilon}(n)}$, for any $\epsilon > 0$.

IV. RESULTS

A. Hardness of Approximation for MINSTOP

In what follows, we provide a proof that MINSTOP is not approximable within $o(\log n)$, unless $P = NP$. This results improves upon the finding in [10], where the weaker claim that MINSTOP cannot be approximated within any positive constant was established. This improvement is a consequence of the fact that our proof relies on reduction from the MINSETCOVER, rather than the Minimum Vertex Cover problem [10].

Theorem 1: There is no polynomial time $o(\log n)$-approximation algorithm for MINSTOP unless $P = NP$.

Proof: The proof is by a reduction from MINSETCOVER: Let $n = \left|\cup_{i\in[a]} S_i\right|$ and without loss of generality assume that $S \subset [n]$ for each $S \in \mathcal{S}$.

Form a bipartite graph $G = (L \cup R, E)$ with nodes,

$$L = \{u_1, ...u_a, x, y\}$$
$$R = \{v_1, ...v_n, w_1, ..., w_a, z\}$$

and edges,

$$E = \{(u_i, v_j) : j \in S_i\} \cup \{(u_i, w_i : i \in [a]\} \cup \{(x, v) : v \in R\} \cup \{(y, v) : v \in \{w_1, ..., w_a, z\}\}\ .$$

We show next that G has stopping distance $2 + t$ if and only if the minimum set cover is t. The proof then follows from the claim above, and the fact that there is no polynomial algorithm returning an $o(\log n)$ approximation of MINSETCOVER, unless $P = NP$.

Let S be a stopping set. Consequently,

1) If $(x \in S$ or $y \in S)$, then $(x \in S$ and $y \in S)$ since otherwise $d_{G_S}(z) = 1$.
2) If $x \in S$ then $u_i \in S$ for some i since otherwise $d_{G_S}(v_j) = 1$ for some $j \in [n]$.
3) If $u_i \in S$ then $(x \in S$ or $y \in S)$ since otherwise $d_{G_S}(w_i) = 1$.

Therefore, if S is non-empty $x, y, _i \in S$ for some i. But then $d_{G_S}(v_j) \geq 2$. However this means that for all $j \in [n]$, $d_{G_{S\setminus\{x,y\}}}(v_j) \geq 1$. Therefore, S being a stopping set implies that the included u_i nodes correspond to a covering of $[n]$.

The nodes corresponding to a covering of $[n]$ in addition to x and y is a stopping set since every node on the right hand side (R) is in the neighborhood and has degree at least two. Hence $\text{MINSTOP}(G) = 2 + \text{MINSETCOVER}(\mathcal{S})$ where $\text{MINSTOP}(G)$ denotes the cardinality of the minimum stopping set and $\text{MINSETCOVER}(\mathcal{S})$ denotes the cardinality of the minimum set cover. ■

We note that under the stronger assumption that $NP \not\subset DTIME(n^{\text{polylog}\,n})$, the stronger results that there is no poly-time approximation algorithm for MINSTOP within $2^{(\log n)^{1-\epsilon}}$, for any $\epsilon > 0$, was shown in [10].

B. Hardness of Approximation for MINT$_{ZP}$ and MINT

We show next that the problems MINT$_{ZP}$ and MINT are computationally at least as hard as the MINCODEWORD problem.

Theorem 2: There is no polynomial time $O(1)$-approximation algorithm for MINT$_{ZP}$, unless $P = NP$.

Proof: Note first that the MINCODEWORD problem is $O(1)$-hard to approximate even under the restriction that the Tanner graph of the code is left regular. This follows directly from the results in [2] and [14].

Given an Tanner graph T on $L \cup R$ that is left regular say with degree $\lfloor (l-1)/2 \rfloor + 1$, for each node $u \in L$ create $l - \lfloor (l-1)/2 \rfloor - 1$ new nodes in R each connected to u. Call the new Tanner graph T'. Then any $S \subset L$ is a ZP-trapping set in T' iff S is the support of a codeword in T. ■

Theorem 3: There is no polynomial time $O(1)$-approximation algorithm for MINT, unless $P = NP$.

Proof: The proof is by a reduction from MINCODEWORD. We construct the Tanner graph $(L \cup R, E)$ of the dual code $C^\perp$ where

$$L = \{u_1, ...u_k\}, R = \{v_1, ...v_n\}, \text{ and}$$
$$E = \{(u_i, v_j) : G_{ij} = 1\}.$$

Note that for each $S \subset L$, $\Gamma(S)$ corresponds to a codeword. Hence, if we have an α-approx to the min-trapping set problem for any a, then this gives an α

approx to the minimum weight codeword problem by running through all values of a and taking the minimum of the resulting b's.

But, since it is hard to $O(1)$-approximate MINCODEWORD, it is NP-hard to $O(1)$-approximate MINT [4]. ∎

C. Hardness of Approximation for MINT_e

Theorem 4: There is no polynomial time algorithm for solving the MINT_e problem, unless $P = NP$.

Proof: The proof is based on showing that a polynomial time algorithm for solving MINT_e can be used for solving the MAX3DMATCH problem, and is based on similar arguments as those used for showing that MAXLD is NP-complete [2]. To this end, let us construct the *matching incidence matrix* D as follows. Let $\{T =_j (x_j, y_j, z_j)\}$, $j = 1, \ldots, t$, be the collection of ordered triples in $T \subset X \times X \times X$, where $|T| = t$, and $|X| = n$. Then D is a $3\,n \times t$ dimensional zero-one matrix, with entries

$$\begin{aligned} 1 \le i \le n: &\;\; D_{i,j} = 1, \text{ iff } x_j = i; \\ n+1 \le i \le 2n: &\;\; D_{i,j} = 1, \text{ iff } y_j = i; \\ 2n+1 \le i \le 3n: &\;\; D_{i,j} = 1, \text{ iff } z_j = i. \end{aligned}$$

Assume next that there exists a polynomial time algorithm for the MINT_e problem. Construct D for a given matching problem, set $b = 3 \times n$, and run the MINT_e algorithm on D. The algorithm will produce a set of a columns, indexed by a set of triples T_{j_i}, $i = 1, \ldots, b$, of T. Each row in the submatrix induced by the triples has weight one, which follows from the definition of an elementary trapping set. Consequently, these triples represent a matching for T. This implies that no polynomial time algorithm for the MINT_e problem exists, unless P=NP. ∎

D. Hardness of Approximation for MINT_c

We first prove a hardness of approximation result for the problem MINGOOD, of finding a good set of minimum cardinality. We then use this to show a hardness of approximation result for MINT_c.

Our proof uses a reduction from MINCODEWORD. Let H be the $n \times (n-k)$ parity check of some code. We may assume that the code specified by H includes at least one codeword in addition to the all-zero vector (i.e., the code has dimension at least one). This gives rise to the graph $G' = (L' \cup R', E')$ where

$$\begin{aligned} &L' = \{x_1, \ldots, x_n\}, R' = \{y_1, \ldots, y_m\}, \text{ and} \\ &E' = \{(x_i, y_j) : H_{ij} = 1\}. \end{aligned}$$

We will create a bipartite graph $G = (L \cup R, E)$ by augmenting G' with gadgets that we call "Zig"s and "Or"s. These gadgets will ensure that the minimum cardinality of a good set is approximately proportional to the minimum weight of any codeword.

Before describing the proof, we state two definitions. For a subgraph G'' of G we define

$$\begin{aligned} \text{Cost}_S(G'') =& |S \cap V(G'')| \\ \text{Disc}_S(V(G'')) =& |\{v \in G_S \cap V(G'') \cap R : d_{G_S}(v) \text{ even}\}| \\ & - |\{v \in G_S \cap V(G'') \cap R : d_{G_S}(v) \text{ odd}\}|. \end{aligned}$$

Given this notation we can rephrase the problem MINGOOD as estimating,

$$\min_{S \subset L : \text{Disc}_S(G) \ge 0} \text{Cost}_S(G) \;\; .$$

For each $x \in L'$ we add the $\mathsf{Zig}(x)$ graph. This graph consists of $3(m-1)$ nodes

$$\begin{aligned} L(\mathsf{Zig}(x)) &= \{v_1, \ldots, v_{m-1}\} \\ R(\mathsf{Zig}(x)) &= \{u_1, \ldots, u_{m-1}, w_1, \ldots, w_{m-1}\} \end{aligned}$$

and edges,

$$\begin{aligned} E(\mathsf{Zig}(x)) = &\{(u_i, v_i), (v_i, w_i) : i \in [m-1]\} \cup \\ &\{(v_i, w_{i+1} : i \in [m-2]\} \cup \{(x, w_1)\}. \end{aligned}$$

The intuition behind $\mathsf{Zig}(x)$ is that if x is in the trapping set then the nodes $L(\mathsf{Zig}(x))$ will also be in the trapping set. This is formalized in the lemma below.

Lemma 1: For all $x \in L'$, $\text{Disc}_S(\mathsf{Zig}(x)) \le 0$ and $\text{Disc}_S(\mathsf{Zig}(x)) = 0$ iff $\mathsf{Zig}(x) \cap L \subset S$.

For each $y \in R'$, we add a $\mathsf{Or}(y)$ graph. Let $\Gamma(y) \cap L' = \{u_1, \ldots, u_{k'}\}$. Let $k = 2^{\lceil \log_2 k' \rceil}$. The construction $\mathsf{Or}(y)$ consists two node sets $L(\mathsf{Or}(y))$ and $R(\mathsf{Or}(y))$. Consider a binary tree on the nodes $\{u_1, \ldots, u_{k'}\}$, where $u_{k'}$ is repeated $k - k'$ times. $L(\mathsf{Or}(y))$ consists of a $n-1$ nodes corresponding to the internal nodes of the tree, i.e. $L(\mathsf{Or}(y))$ equals

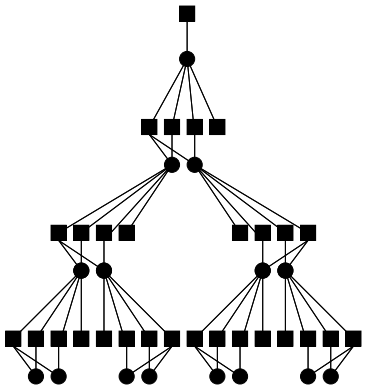
$$\{v_{u_1 \vee u_2}, \ldots, v_{u_{k-1} \vee u_k}, v_{u_1 \vee u_2 \vee u_3 \vee u_4}, \ldots, v_{u_1 \vee u_2 \vee \ldots \vee u_k}\}$$

For each internal node v with children u and w, we add four new nodes: all these nodes are connected to v, the first and third are connected to u and the first and second are connected to w. If v is the root of the tree we also add one more new node which is connected to v. Let $R(\mathsf{Or}(y))$ be the set of such nodes and let $E(\mathsf{Or}(y))$ be the set of such edges. Finally, let $f(S, y)$ be equal to

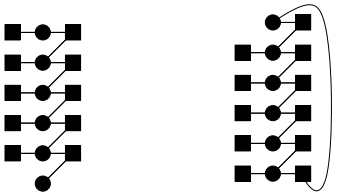
$$\{v_{u_i \vee \ldots \vee u_j} \in L(\mathsf{Or}(y)) : |S \cap \{u_i, \ldots, u_j\}| \ge 1\} \;\; .$$

We have the following lemma relating S, the $\mathsf{Or}()$, and the set f.

Lemma 2: For all $y \in G_S \cap R$, $\text{Disc}_S(\mathsf{Or}(y)) \le -1$ with equality iff $S \cap L(\mathsf{Or}(y)) = f(S, y)$.



(a) $\mathsf{Or}(y)$



(b) $\mathsf{Zig}(x)$ (c) Ballast

Fig. 1. Reduction from MINCODEWORD to MINT$_c$.

Proof: Consider two nodes $v_{u_i \vee \ldots \vee u_j}$ and $v_{u_{j+1} \vee \ldots \vee u_k}$ in $L(\mathsf{Or}(y))$. The lemma follows from the observation that if either node is in S then,

$$\mathrm{Disc}_S(\Gamma(v_{u_i \vee \ldots \vee u_j}) \cap \Gamma(v_{u_{j+1} \vee \ldots \vee u_k}) \cap \Gamma(v_{u_i \vee \ldots \vee u_k}))$$

being 0 implies that $v_{u_i \vee \ldots \vee u_k} \in S$. ∎

Note that the graph G, constructed in the previous section, has $|L| \leq mn + 2n(n-k)$ and $|R| \leq (n-m) + n^2$.

Lemma 3: $\mathrm{Disc}_S(G) \geq 0$ iff $S \cap L'$ is a codeword and for each $x \in S \cap L'$, $\mathsf{Zig}(x) \cap L \subset S$.

Proof: Let $|\Gamma(S) \cap R'|$ and $\mathrm{Disc}_S(G')$. Then, invoking Lemma 1 and 2 shows that

$$\begin{aligned}
&\mathrm{Disc}_S(G) \\
&= \mathrm{Disc}_S(G') + \sum_{x \in L'} \mathrm{Disc}_S(\mathsf{Zig}(x)) + \sum_{y \in R'} \mathrm{Disc}_S(\mathsf{Or}(y)) \\
&\leq \mathrm{Disc}_S(G') - \sum_{x \in L'} I_{\mathsf{Zig}(x) \cap L \not\subset S} - |\Gamma(S) \cap R'|.
\end{aligned}$$

Therefore, $\mathrm{Disc}_S(G) \geq 0$ implies that

$$\begin{aligned}
\forall y \in R',\ d_{G_S}(y) &= 0 \bmod 2 \\
\forall x \in S \cap L',\ \mathsf{Zig}(x) &\subset G_S.
\end{aligned}$$

Again, using Lemma 1 and 2, one can show that if $\forall y \in R',\ d_{G_S}(y) = 0 \bmod 2$ and $\forall x \in S \cap L',\ \mathsf{Zig}(x) \subset G_S$, then S is good. ∎

Theorem 5: There is no polynomial time $O(1)$-approximation algorithm for MINGOOD unless $P = NP$.

Proof: Assume that S is a good set such that $S \leq \alpha$MINGOOD for some constant α. By Lemma 3 and Lemma 2,

$$|S| = |S \cap L'|m + \sum_{y \in \Gamma(S \cap L')} |f(S,y)|$$

and $S \cap L'$ corresponds to a codeword. But $\sum_{y \in \Gamma(S \cap L')} |f(S,y)| \leq 2n(n-k)$ and so by setting m sufficiently large we get a constant approximation for MINCODEWORD. But no such approximation exists unless $P = NP$ [4]. ∎

To finalize the proof of the hardness result for MINT$_c$, we need to further augment our graph G with multiple "Ballast" constructions. We call the resulting graph G^+. The intuition behind Ballast is that no nodes from Ballast will be chosen in S while the multiple copies of Ballast will ensure that the complement of S is also good. A single Ballast consists of nodes

$$\begin{aligned}
L(\mathsf{Ballast}) &= \{u_1, \ldots, u_l\} \\
R(\mathsf{Ballast}) &= \{v_1, \ldots, v_l, w_1, \ldots, w_l\}
\end{aligned}$$

and edges,

$$\begin{aligned}
E(\mathsf{Ballast}) = \{(u_i, v_i) : i \in [l]\} &\cup \{(v_i, u_{i+1}) : i \in [l-1]\} \\
&\cup \{(v_l, u_1)\} \cup \{(u_i, w_i) : i \in [l-1]\}.
\end{aligned}$$

We consider setting $l = n|L|$ and adding $|R|$ copies of Ballast to G.

Lemma 4: $\mathrm{Disc}_S(\mathsf{Ballast}) = 1$ if $L(\mathsf{Ballast}) \subset S$ and $\mathrm{Disc}_S(\mathsf{Ballast}) \leq 0$ otherwise.

Proof: The result follows because

$$|\Gamma(S \cap L(\mathsf{Ballast}))| \geq |S \cap L(\mathsf{Ballast})|$$

with equality iff $L(\mathsf{Ballast}) \subset S$. ∎

Lemma 5: Assuming there exists a non-zero codeword, there is a good set in G. Furthermore, any good set in G is a trapping set for G^+.

Proof: Let S' be the subset of L' corresponding to the minimum weight codeword. Let

$$S = S' \cup \left(\bigcup_{x \in S'} L(\mathsf{Zig}(x)) \right) \cup \left(\bigcup_{y \in \Gamma(S')} f(S,y) \right).$$

Then S is a good set in G. For the second part of the lemma note that by Lemma 4, for $S \subset L$, $\mathrm{Disc}_{\bar{S}}(G^+) \geq |R| - |R| = 0$. ∎

Theorem 6: There is no poly-time $O(1)$-approximation algorithm for MINT$_c$ unless $P = NP$.

Proof: Assume that S is a trapping set such that $S \leq \alpha \text{MINT}_c$ for some constant α. By Lemma 5, we know that $|S| \leq \alpha |L|$ and hence S does not include all LHS nodes of any copy of Ballast because doing so would imply that $|S| \geq |L(\text{Ballast})| = n|L|$. But then by Lemma 4, we may assume that no nodes from Ballast are included in S because removing all such nodes from S increases $\text{Disc}_S(G)$. Consequently S must be a subset of L. Since any good subset of L is a trapping set, $\text{MINGOOD}(G) = \text{MINT}_c(G^+)$. But, by Theorem 5, there is no constant approximation for the problem MINGOOD, which completes the proof of the theorem. ■

V. HARDNESS OF PROBLEMS FOR SPARSE CODES

The fact that a problem is NP-hard usually does not imply that a special instance of the problems is NP-hard. Since iterative decoding algorithms have both linear-time complexity and offer good decoding performance only for special classes of codes, it is important to establish the analogues of the results in Section IV for such codes. We provide next a set of results asserting the hardness of approximating the minimum distance and smallest stopping and trapping set in LDPC codes, i.e., codes for which each vertex, except for possibly a constant number, in the Tanner graph $G = (L \cup R, E)$ has degree at most ℓ, for some constant $\ell > 2$.

Theorem 7: MAXLD and MINCODEWORD are NP-hard for LDPC codes.

Proof: The proof is a direct consequence of the fact that the parity-check matrix used in the reduction from the MAX3DMATCH to the MAXLD problem is sparse (it has column weight three, and the row weight can be bounded as well by invoking the constraint that every element of X cannot appear more than $r \geq 3$ times). The second result follows from the observation that there exists a polynomial-time reduction algorithm from MAXLD to MINCODEWORD problem [14], [15]. ■

As a consequence of the above finding, all trapping set problems described in Section IV, for which the hardness was established in terms of reductions from the MINCODEWORD problem, remain NP-hard for the class of LDPC codes.

Theorem 8: MINSTOP is NP-hard for LDPC codes.

Proof: The proof follows along the same lines as the proof of Theorem 1, with the MINSETCOVER problem replaced by the MINKSETCOVER problem. ■

Theorem 9: MINT_e is approximation-complete for LDPC codes.

Proof: The proof follows along the same lines as the proof of Theorem 4, with the three-dimensional matching problem replaced by its constraint version involving a bounded number ℓ of appearances of each element in X. ■

Our findings imply that, despite the existence of polynomial time algorithms for finding a minimum weight codeword of LDPC codes with column-weight two [6], no such algorithms exist for the general class of LDPC codes, provided that $P \neq NP$.

REFERENCES

[1] A. Barg, "Some New NP-Complete Coding Problems", *Problemy Peredachi Informatsii*, Vol. 30, pp. 23-–28, 1994, (in Russian).

[2] E. R. Berlekamp, R. J. McEliece, and H. Van Tilborg, "On the Inherent Intractability of Certain Coding Problems, *IEEE Trans. Inform. Theory*, Vol. 24, pp. 384 – 386, May 1978.

[3] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke, "Finite Length Analysis of Low-Density Parity-Check Codes," *IEEE Trans. on Inform. Theory*, Vol. 48, No. 6, pp. 1570 – 1579, June 2002.

[4] I. Dumer, D. Micciancio and M. Sudan, "Hardness of Approximating the Minimum Distance of a Linear Code", *IEEE Trans. on Inform. Theory*, Vol. 49, No. 1, pp. 22-37, 2003.

[5] M. Garey, D. Johnson, "Computers and Intractibility: a Guide to the Theory of NP-Completeness", *W.H. Freeman*, 1979.

[6] X. Y. Hu and M. Fossorier, "On the Computation of the Minimum Distance of LDPC Codes", *preprint*.

[7] R. Koetter, "Iterative Coding Techniques, Pseudocodewords, And Their Relationship", *Workshop on Applications of Statistical Physics to Coding Theory*, Santa Fe, New Mexico, January 2005.

[8] S. Laendner and O. Milenkovic, "Algorithmic and Combinatorial Analysis of Trapping Sets in Structured LDPC Codes," *Proceedings of WirelessCom 2005*, Hawaii, June 2005.

[9] D. MacKay and M. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes," *Electronic Notes in Theoretical Computer Science*, Vol. 74, 2003, URL: http://www.elsevier.nl/locate/entcs/volume74.html.

[10] K. Murali Krishnan and L. Sunil Chandran, "Hardness of Approximation Results for the Problem of Finding the Stopping Distance in Tanner Graphs", *FSTTCS*, pp. 69–80, 2006.

[11] R. Raz and S. Safra, "A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP", *Proceedings of Symposium on the Theory of Computing*, STOC'1997, pp. 475-484, El Paso, May 1997.

[12] T. Richardson, "Error-floors of LDPC Codes," *Proceedings of the 41st Annual Conference on Communication, Control and Computing*, pp. 1426–1435, September 2003.

[13] M. Stepanov and M. Chertkov, "Instanton Analysis of Low-Density Parity-Check Codes in the Error-Floor Regime", *Proceedings of the International Symposium on Information Theory*, ISIT'2007, pp. 552-556, Seattle, July 2007.

[14] A. Vardy, "Algorithmic Complexity in Coding Theory and the Minimum Distance Problem", *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, El Paso, Texas, pp. 92 – 109, 1997.

[15] A. Vardy, "The Intractability of Computing the Minimum Distance of a Code", *IEEE Trans. Inform. Theory*, Vol. 43, pp. 1757—1766, November 1997.

[16] V. V. Vazirani, "Approximation Algorithms", *Springer*, 2001

[17] C. C. Wang, S. Kulkarni, and V. Poor, "Exhausting Error-Prone Patterns in LDPC Codes", *preprint*.

[18] K. Zigangirov, A. Pusane, D. Zigangirov, and D. Costello, "On the Error Correcting Capability of LDPC Codes", *preprint*.

[19] V. Zyablov and M. Pinsker, "Estimates of the Error-Correction Complexity of Gallager's Low-Density Codes", *Problems of Information Transmission*, Vol. 11, No. 1, pp. 18–28, January 1976.