IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

VOL. E97-A NO. 4
APRIL 2014

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

LETTER

Finding Small Fundamental Instantons of LDPC Codes by Path Extension

Junjun GUO[†], ***Nonmember***, **Jianjun MU**[†a)], ***Member***, **Xiaopeng JIAO**[†], ***and*** **Guiping LI**[†], ***Nonmembers***

SUMMARY In this letter, we present a new scheme to find small fundamental instantons (SFIs) of regular low-density parity-check (LDPC) codes for the linear programming (LP) decoding over the binary symmetric channel (BSC). Based on the fact that each instanton-induced graph (IIG) contains at least one short cycle, we determine potential instantons by constructing possible IIGs which contain short cycles and additional paths connected to the cycles. Then we identify actual instantons from potential ones under the LP decoding. Simulation results on some typical LDPC codes show that our scheme is effective, and more instantons can be obtained by the proposed scheme when compared with the existing instanton search method.

key words: *instanton, error floor, low-density parity-check (LDPC) codes, linear programming (LP)*

1. Introduction

Although the linear programming (LP) decoding for low-density parity-check (LDPC) codes has been intensively investigated in recent years owing to its maximum-likelihood (ML) certification property and provable performance bound [1], [2], the error floor phenomenon is still the main obstacle to practical applications of LDPC codes for the LP decoding. One of the major reasons causing the error floor performance is the sub-optimality of the decoding methods such as the iterative decoding and LP decoding [3]. When the problematic noise configurations, so-called instantons, arise in the received vectors, the LP decoder would output fractional pseudo-codewords. The knowledge of instantons, especially small fundamental instantons (SFIs), can be used to design LDPC codes with low error floors and approximately estimate the LP decoding performance. Thus, it is important to search for those instantons with various sizes in regular LDPC codes. However, identifying all instantons is a difficult task in general [4], [5]. In [6], the authors devised a pseudo-codeword search (PCS) algorithm for the LP decoding over the continuous channels. By searching for pseudo-codewords over the AWGN channel, one can obtain instantons as the PCS's byproducts. In [7], Chilappagari *et al.* proposed an iterative instanton search algorithm (ISA) for the random input over the BSC. If the input vector contains sufficient bit-flipped errors, the ISA can converge to an instanton in finite steps with at most twice the number of flipped bits. It is difficult to enumerate the list of instonstans with a given size due to the random inputs of the algorithms proposed in [6] and [7]. Moreover, the approach to brute-force search for instantons based on the parity-check matrix becomes computationally prohibitive as the code length and instanton size increase.

By observing a number of the Tanner subgraphs induced by instantons of regular LDPC codes, it is well known that each troublesome subgraph consists of at least one short cycle with connected paths (if any). Inspired by the idea of finding dominant trapping sets based on the strategy of short cycle expansion [8], we propose an efficient SFIs search scheme of regular LDPC codes for the LP decoding over the BSC. Firstly, we use a simple cycle enumeration method to generate a set of short cycles. Secondly, we extend possible paths for each short cycle and determine potential SFIs in a heuristical manner. Finally, we identify all actual instantons from those potential SFIs by using the LP decoding. The proposed scheme may also be modified to find instantons of irregular LDPC codes.

The remainder of this letter is organized as follows. In Sect. 2, the notions of LDPC code, LP decoding, support set and instanton are introduced. We proposes the SFIs search scheme in Sect. 3. In Sect. 4, some numerical experiments are provided. Finally, Sect. 5 concludes the letter.

2. Preliminaries

Let H denote an $m \times n$ parity-check matrix of a binary LDPC code C. The matrix H can be represented by a Tanner graph $G = (V \cup W, E)$. The set of variable nodes is $V = \{v_1, v_2, \cdots, v_n\}$ and the set of check nodes is $W = \{w_1, w_2, \cdots, w_m\}$. The edge set is $E = \{(v_i, w_j)|H_{i,j} = 1\}$. The girth g is the length of the shortest cycle in G. The set of neighbors of a variable node v_i is denoted as $N(v_i)$. Similarly, $N(w_j)$ represents the set of neighbors of a check node w_j.

We assume that a codeword vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is transmitted over the BSC. The received vector is $\mathbf{r}$. The LP decoding for the LDPC code C over the BSC can be described as

$$\texttt{min.}\ \|\mathbf{x} - \mathbf{r}\|_1,\ \texttt{subject to}\ \mathbf{x} \in P. \tag{1}$$

In Eq. (1), the symbol $\|\cdot\|_1$ denotes ℓ_1 norm and the polytope $P = \bigcap_{1 \le j \le m} conv(C_j)$ where each local codeword $x \in C_j$ satisfies the jth check-parity constraint and the $conv(C_j)$ is the convex hull of the local code C_j. The LP decoding result is obtained by solving convex optimization problem over P.

Manuscript received September 2, 2013.
Manuscript revised November 6, 2013.
[†]The authors are with the School of Computer Science and Technology, Xidian University, Xi'an, 710071, P.R. China.
a) E-mail: jjmu@xidian.edu.cn
DOI: 10.1587/transfun.E97.A.1001

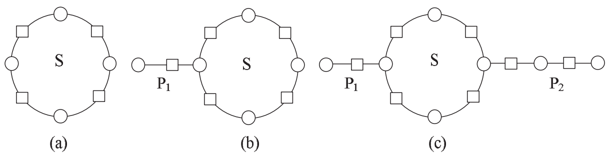

Fig. 1 Instanton-induced graphs consist of cycles with the same size. The symbol ∘ presents the variable node and □ denotes the check node, and each variable node is an element of the instanton.

If this result is integral, then we find a codeword. Otherwise, it is a fractional pseudo-codeword and the decoder fails.

Definition 1 [1]: The *support set* of a vector $\mathbf{u}$ is defined as the set of indices of all non-zero elements in $\mathbf{u}$. Generally, the support set of a vector $\mathbf{u}$ is denoted by $supp(\mathbf{u})$.

Definition 2 [7]: An *instanton* $\mathbf{i}$ is a binary vector, as the LP decoding input over the BSC, which can be decoded to the fractional pseudo-codeword. For a binary input vector $\mathbf{r}$ where $supp(\mathbf{r}) \subset supp(\mathbf{i})$, it is decoded to the integer pseudo-codeword. Conversely, the decoder declares failure where $supp(\mathbf{i}) \subseteq supp(\mathbf{r})$. The size of $\mathbf{i}$ is the cardinality of $supp(\mathbf{i})$.

The instanton can be described by the instanton-induced graph (IIG) of Tanner graphs. A typical IIG I of regular LDPC codes consists of a short cycle S and several possible paths $\Gamma = \{P_1, P_2, \cdots, P_t\}$ where t is the number of paths and all paths in Γ are connected to S. Figure 1 shows a set of IIGs with the same length of short cycles.

Definition 3: A *fundamental instanton* is a special instanton whose IIG has the following properties: (1) The number of connected paths in Γ is not more than 2, namely, $t \leq 2$; (2) The number of variable nodes in S are not less than that of ones in Γ.

We have empirically observed that almost all instantons are fundamental instantons. More importantly, small-size fundamental instantons are more harmful than others and they dominate the error floor performance of LDPC codes.

3. Searching for Small Fundamental Instantons (SFIs)

In this section, a simple search scheme is proposed to find SFIs of LDPC codes over the BSC. Firstly, we enumerate short cycles of a given Tanner graph. Secondly, in order to find potential SFIs, we can construct possible IIGs by progressively extending paths connected to short cycles. Finally, we identify actual SFIs from potential instantons under the LP decoding.

3.1 Enumerating Short Cycles of Tanner Graph

The cycle [9], [10] is one of important components in the IIG. For a given SFI, its IIG is closely related to short cycles. Thus we must first find as many short cycles with various lengths as possible in order to identify the SFIs. Here, we find short cycles based on the depth-first search (DFS). The procedure of enumerating a short cycle can be described as follows. First of all, we pick a variable node v_1 as a visited root node, and then progressively search the neighbor node to obtain the path $P = (v_1 w_1 v_2 w_2 \cdots w_\ell)$. If w_ℓ has a neighbor variable node that just is the root node v_1, a cycle with length 2ℓ is obtained as shown in Fig. 2. Otherwise, we continue to search the neighbor node of the previous one by backtracking approach until a cycle is found. Repeating the above steps, we can obtain all short cycles with the given length 2ℓ.

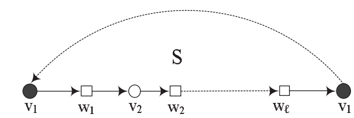

Fig. 2 Illustration for the example of spanning cycle by extending paths.

3.2 Finding Possible IIGs Based on Path Extension

The key idea to determine a potential SFI is to construct its possible IIG that contains at least one short cycle and paths (if any) connected to the cycle. Assuming that the size of SFIs is k, the search procedure starts from a set of short cycles of size 2ℓ ($\ell \leq k$). If $\ell = k$, a possible IIG is only comprised by the short cycle. If $\ell < k$, an IIG can be constructed by extending path(s) over the short cycle until the size of the potential SFI is up to k. The detailed description of the algorithm for finding possible IIGs is given in Algorithm 1.

Algorithm 1 An algorithm for finding possible IIGs.

Input: Tanner graph G, size of the SFI k, set of short cycles Θ.
1: $I \leftarrow \emptyset$.
2: **repeat**
3: Select a cycle S from Θ as the possible IIG Φ and let ϕ be the number of variable nodes of Φ.
4: **if** $\ell = k$ and there exists a path P' between any two variable nodes of S **then**
5: $I \leftarrow I \cup S$.
6: Goto Line 18.
7: **end if**
8: **repeat**
9: Take a variable node v from S to extend a path P, add P to Φ where $\phi \leq k$.
10: **while** $\phi < k$ **do**
11: Select other variable node v' from S and continue to extend a new path P'.
12: Add P' to Φ.
13: **end while**
14: **if** $\phi = k$ **then**
15: $I \leftarrow I \cup \Phi$.
16: **end if**
17: **until** all cases of the path extension of S are considered.
18: **until** all cycles in Θ are exhausted.
Output: set of possible IIGs I.

Remarks:

(1) It should be noted that the relationship between the instanton size k and cycle length 2ℓ is satisfied with the following condition $k - \ell \leq 2\ell \leq k$.

(2) It is worth pointing out that the size of smallest fundamental instantons k_{min} is related to the fractional distance

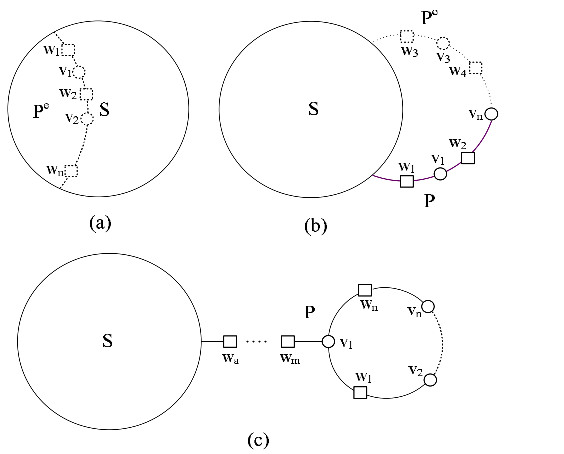

Fig. 3 Possible path extention cases of IIGs. (a) If an IIG is only comprised of a cycle S, one can extend an additional path $P^e = (w_1v_1w_2v_2\cdots w_n)$ between any two variable nodes in S which can construct the shorter cycle together with fractional paths of S. (b) There exits a complementary path $P^c = (w_3v_3w_4\cdots w_n)$ such that the extended path $P = (w_1v_1w_2\cdots v_n)$ forms a closed loop with the fraction of cycle S. (c) Different from (b), the extended path $P = (w_a\cdots w_mv_1w_1v_2\cdots v_nw_nv_1)$ is a lollipop graph.

d_{frac} of the code, and $k_{min} \geq \lceil d_{frac}/2 \rceil$ [7].

(3) In Line 4 of Algorithm 1, an additional path P connected to the original cycle S can generate a new cycle whose length is not more than the original one's. From Line 8 to 17, the paths around the cycle can be extended, which is based on the heuristical manner as illustrated in Figs. 3(b)–(c).

3.3 Identifying Actual SFIs for LP Decoding

Once potential SFIs are obtained, we can identify actual SFIs from them using LP decoding. Without loss of generality, we assume that all-zero codeword is transmitted over the BSC. Let $\mathbf{x}'$ denote the codeword biased noise configuration, and $\mathbf{x}' = (1, 1, 1, \cdots, 1, 0, 0, 0, 0, \cdots, 0)$. Here, it is assumed that the first small bits of the codeword are involved in the SFI. Then we take the vector $\mathbf{x}'$ as the input of LP decoding. If the LP decoding result is not an integral solution, the actual SFI related to $\mathbf{x}'$ is found.

3.4 Computational Complexity

The total computational complexity for finding SFIs by path extension is mainly determined by three parts: (1) enumerating short cycles, (2) identifying potential instantons by using the Algorithm 1, and (3) verifying whether potential instantons are actual ones under the LP decoding. For Part (1), it is obvious that in a depth-first algorithm, the computational complexity of enumerating short cycles of length 2ℓ for a length n LDPC code is $O(n(d_vd_w)^{\ell})$, where d_v denotes the variable node degree and d_w denotes the check node degree. For Algorithm 1, assuming that N_c is the number of cycles, the complexity of Algorithm 1 for finding potential SFIs with a given size k is approximately $O(N_c\ell^2(k-\ell)(d_vd_w)^{k-\ell})$.

Table 1 Distribution of instantons for (155, 64, 20) Tanner code.

Instantons	5	6	7	8
ISA	155	≈ 2300	$\approx 6.4 \times 10^5$	$\approx 3.8 \times 10^7$
Our scheme	155	6, 297	960, 730	42, 106, 153

Table 2 Lists of almost all smallest fundamental instantons for the *204.33.484* code.

Induced by $6-cycles$	Induced by $8-cycles$
$(4, 82, 122, 131)^*$	$(4, 82, 122, 131)^*$
$(9, 41, 178, 203)^*$	$(9, 41, 178, 203)^*$
$(17, 34, 107, 161)$	$(70, 109, 155, 190)$
$(25, 161, 172, 202)^*$	$(25, 161, 172, 202)^*$
$(25, 80, 161, 172)^*$	$(25, 80, 161, 172)^*$
$(32, 39, 46, 132)^*$	$(32, 39, 46, 132)^*$
$(53, 70, 109, 155)$	$(53, 70, 109, 190)$
$(53, 70, 155, 190)^*$	$(53, 70, 155, 190)^*$
$(53, 109, 155, 190)$	$(100, 161, 172, 181)$
-	$(161, 166, 172, 181)$

Assuming that N_p is the number of potential instantons, the running time of Part (3) is equal to the time taken by N_p calls to an LP solver implemented by the simplex algorithm.

4. Numerical Results

In this section, distributions of SFIs of several typical LDPC codes are provided and the prediction result of FER performance is obtained based on the knowledge of SFIs.

4.1 Distributions of SFIs

Example 1: Here, we apply the proposed scheme to find SFIs of the (155, 64, 20) Tanner code which is constructed algebraically [12]. The girth of this code is 8 and d_{frac} is 8.3498 [7]. The size of smallest fundamental instantons is $k_{min} = 5$, which is not less than a half of d_{frac}. Firstly, we enumerate short cycles with various sizes, such as 8, 10, 12, 14 and 16-cycles. Then we find potential SFIs and further check them under the LP decoding. There are 155 different instantons with size 5 derived from 930 isomorphic IIGs.

It should be pointed out that the key observation we made is that the instantons searched by ISA are almost SFIs. It can be seen from Table 1 that we obtain more instantons with various sizes when compared with the results obtained by ISA.

Example 2: In this example, we consider the regular *204.33.484* MacKay code constructed randomly [13]. The girth of this code is $g = 6$, and the size of smallest fundamental instantons is $k_{min} = 4$. It is worth noting that some instanton patterns may be counted more than one time for short cycles with various sizes. For instance, the set of intantons with size 4 consists of 9 instantons induced by 6-cycles, 10 instantons induced by 8-cycles, and 6 instantons induced by both 6-cycles and 8-cycles. Therefore, the number of smallest fundamental instantons of this code is $9 + 10 - 6 = 13$. All of typical SFI patterns are listed in Table 2.

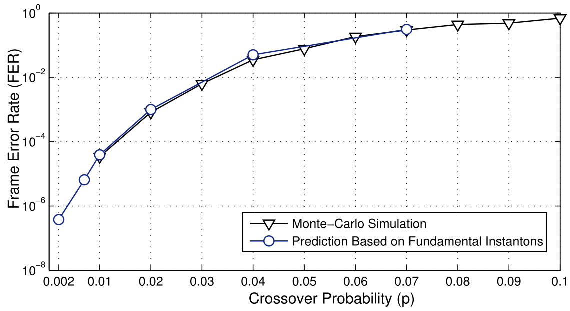

Fig. 4 Comparison of the FER curves plotted by using Monte-Carlo simulations and prediction approach based on the knowledge of SFIs for the *PEGReg100* × *50* code.

Remark: Instanton support vectors labeled '*' are induced by both 6-cycles and 8-cycles.

Example 3: We take the regular *PEGReg100* × *50* code as the experimental code. The code was constructed by PEG approach [11]. The length of this code is 100, and the code rate is 0.5. We collect all short cycles with length 6, 8, 10, 12 and 14 respectively. There are 226 SFIs with size 4 extended from 6-cycles and 8-cycles, 12104 SFIs with size 5 extended from cycles with lengths 6, 8 and 10, 845126 SFIs of size 6 extended from cycles with lengths 8, 10 and 12, and 24639219 SFIs with size 7 extended from cycles with lengths 8, 10, 12 and 14.

4.2 Prediction of FER Performance

Using the knowledge of SFIs obtained by the proposed scheme, one can predict the FER performance without the Monte-Carlo (MC) simulations. Given the crossover probability p, $FER(p)$ tends to 0 as the number of errors $e < k_{min}$ and $FER(p)$ tends to 1 as $e > d_{min}$ where d_{min} denotes the Hamming distance of the specific LDPC code. We can use Eq. (2) to approximately evaluate the FER performance [12].

$$FER(p) \approx \sum_{e=k_{min}}^{d_{min}} \binom{n}{e} p^e (1-p)^{n-e} \sum_{k=k_{min}}^{e} \frac{\binom{e}{k} N_k}{\binom{n}{k}}, \qquad (2)$$

where N_k denotes the number of instantons of size k.

Figure 4 shows the comparison between the $FER(p)$ curves of *PEGReg100* × *50* code obtained by the prediction approach and the MC simulations. It can be seen that the predicted results are almost the same as the ones obtained by the MC simulations. Figure 4 also illustrates that it is easy to predict the FER performance at the small values of p. However, it is difficult to obtain the FER performance at the small values of p by using the MC simulations.

5. Conclusions

Based on the relationship between instantons and cycles in Tanner graphs of regular LDPC codes, we propose an efficient scheme to find SFIs of regular LDPC codes for the LP decoding over the BSC. Experimental results demonstrate the distribution of fundamental instantons with given sizes for the specific LDPC code. More importantly, the proposed scheme can find more instantons than the ISA.

Acknowledgments

This research is supported by the National Nature Science Foundation of China (Grant Nos. 61271004 and 61001132).

References

[1] J. Feldman, M.J. Wainwright, and D.K. Karger, "Using linear programming to decode binary linear codes," IEEE Trans. Inf. Theory, vol.51, no.3, pp.954–972, March 2005.
[2] M. Helming, S. Ruzika, and A. Tanatmis, "Mathematical programming decoding of binary linear codes: Theory and algorithms," IEEE Trans. Inf. Theory, vol.58, no.7, pp.4753–4769, July 2012.
[3] T.J. Richardson, "Error floors of LDPC codes," Proc. 41st Annu. Allerton Conf. Commun., Contr. Comput., 2003.
[4] G.B. Kyung and C.C. Wang, "Finding the exhaustive list of small fully aborbing sets and designing the corresponding low error-floor decoder," IEEE Trans. Inf. Theory, vol.60, no.6, pp.1487–1497, June 2012.
[5] S.K. Chilappagari, M. Chertkov, M.G. Stepanov, and B. Vasic, "Instanton-based techniques for analysis and reduction of error floors of LDPC codes," IEEE J. Sel. Areas Commun., vol.27, no.6, pp.855–865, Aug. 2009.
[6] M. Chertkov and M. Stepanov, "An efficient pseudocodeword search algorithm for linear programming decoding of LDPC codes," IEEE Trans. Inf. Theory, vol.54, no.4, pp.1514–1520, April 2008.
[7] S.K. Chilappagari, M. Chertkov, and B. Vasic, "An efficient instanton search algorithm for LP decoding of LDPC codes over the BSC," IEEE Trans. Inf. Theory, vol.57, no.7, pp.4417–4426, July 2011.
[8] M. Karimi and A.H. Banihashemi, "Efficient algorithm for finding dominant trappings sets of LDPC codes," IEEE Trans. Inf. Theory, vol.58, no.11, pp.6942–6958, Nov. 2012.
[9] T.R. Halford and K.M. Chugg, "An algorithm for counting short cycles in bipartite graph," IEEE Trans. Inf. Theory, vol.52, no.1, pp.287–292, Jan. 2006.
[10] M. Karimi and A.H. Banihashemi, "Message-passing algorithms for counting short cycles in a graph," IEEE Trans. Commun., vol.61, no.2, pp.485–495, Feb. 2013.
[11] X. Hu, E. Eleftheriou, and D.M. Arnold, "Regular and irregular progressive edge-worth Tanner graphs," IEEE Trans. Inf. Theory, vol.51, no.1, pp.386–398, Jan. 2005.
[12] R.M. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," Proc. ICSTA 2001, Ambleside, UK, July 2001.
[13] D.J.C. MacKay, Encyclopedia of Sparse Graph Codes. [Online]. Avaliable: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html