Computing the Stopping Distance of a Tanner Graph is NP-hard

K. Murali Krishnan Priti Shankar

IISc-CSA-TR-2006-13
http://archive.csa.iisc.ernet.in/TR/2006/13/

Computer Science and Automation
Indian Institute of Science, India

December 2006

Computing the Stopping Distance of a Tanner Graph is NP-hard

K. Murali Krishnan
Priti Shankar
Department of Computer Science and Automation
Indian Institute of Science, Bangalore - 560012, India.

Abstract

Two decision problems related to the computation of stopping sets in Tanner graphs are shown to be NP-complete. It follows as a consequence that there exists no polynomial time algorithm for computing the stopping distance of a Tanner graph unless P=NP.

1 Introduction

Stopping sets were introduced in [1] for the analysis of erasure decoding of LDPC codes. It was shown that the iterative decoder fails to decode to a codeword if and only if the set of erasure positions is a superset of some stopping set in the Tanner graph [8] used in decoding. Considerable analysis has been carried out on the distribution of stopping set sizes in LDPC code ensembles, giving valuable insight into the asymptotic performance of message-passing decoding on LDPC ensembles — see for example [2, 3]. Since small stopping sets are directly responsible for poor performance of iterative decoding algorithms, it is of interest to determine the size of the smallest stopping set in a Tanner graph, called the *stopping distance* of the graph. Construction of codes for which there are Tanner graphs that do not contain small stopping sets has been studied — see for example [4, 5]. The stopping distance of the graph, is of interest as it gives the minimum number of erasures that can cause iterative decoding to fail.

The relationship between stopping distance and other graph parameters like girth has been explored in [6] where it is shown that large girth implies high stopping distance. Pishro-Nik and Fekri [12] showed that by adding a suitable number of parity checks the stopping distance of a Tanner graph

of a code as the minimum numb
code, such that the stopping dis
equal to the minimum distance
for a code can be increased to the maximum possible, viz., the minimum distance of the code. Schwartz and Vardy [7] define the *stopping redundancy*

It is clear that if either STOPPING SET or STOPPING DISTANCE can be solved in polynomial time, then invoking the algorithm at most $|L|$ times, the problem of actually finding the stopping distance of a Tanner graph can be solved. Conversely, if there is a polynomial time algorithm for finding the stopping distance of a given Tanner graph G, then we can use the algorithm to solve STOPPING DISTANCE since G has stopping distance less than or equal to t if and only if G contains a stopping set of size less than or equal to t. Note that it is not immediately clear how to solve STOPPING SET in polynomial time even if a polynomial time algorithm for computing the stopping distance of a Tanner graph is known.

The notion of NP-completeness was introduced in [11], and is well established in the computer science literature for the analysis of the computational complexity of problems (see [9, 10] for a detailed account). Typically, a problem is posed as a decision problem, i.e., one where the solution consists of answering it with a *yes* or a *no*. All inputs for which the answer is a *yes* form a set. We identify this set with the problem. A decision problem A belongs to the class NP if there exists a polynomial time algorithm Π such that, for all $x \in A$, there exists a string y (called a *certificate* for membership of x in A), with $|y|$ polynomially bounded in $|x|$, such that Π accepts (x, y), whereas, for all $x \notin A$, Π rejects (x, y) for any string y presented to the algorithm. In other words, problems in NP are precisely those for which membership verification is polynomially solvable. We say a decision problem A is *polynomial time many-one reducible* to a decision problem B if there exists a polynomial time algorithm Π' such that, given an instance x of A, Π' produces an instance z of B satisfying $z \in B$ if and only if $x \in A$. In such case, we write $A \preceq_p B$. A problem $A \in$ NP is NP-complete if for every $X \in$ NP, $X \preceq_p A$. It is generally believed that NP-complete problems have no polynomial time algorithms.

Given an undirected graph (not necessarily bipartite) $G = (V, E)$, $S \subseteq V$ is a *vertex cover* in G if for all $(u, v) \in E$ either $u \in S$ or $v \in S$ or both. We will be using in our reductions the following decision problems associated with the computation of vertex covers in a graph.

Problem 3 *VERTEX COVER(=): Given a graph G and a positive integer t, does G contain a vertex cover of size equal to t?*

Problem 4 *VERTEX COVER: Given a graph G and a positive integer t, does G contain a vertex cover of size at most t?*

VERTEX COVER is shown to be NP-complete in [10, p. 190]. VERTEX COVER(=) is shown to be NP-complete in [9, pp. 949–950] (in fact, in

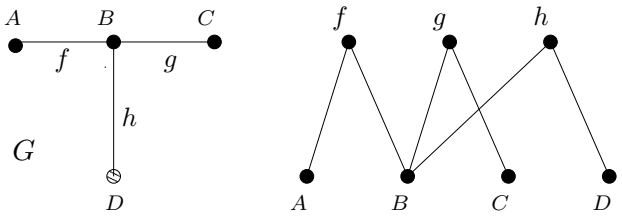

Figure 1: A graph G and its vertex-edge incidence graph

this reference, the authors refer to this problem as the VERTEX COVER problem).

In the following section we show that both STOPPING DISTANCE and STOPPING SET are NP-complete by establishing polynomial time many-one reductions from VERTEX COVER and VERTEX COVER(=) respectively to the above problems.

3 Hardness of STOPPING DISTANCE

Let $(G = (V, E), t)$ be an instance of the VERTEX COVER problem. Let $|V| = n$, $|E| = m$. Excluding trivial cases of the problem we may assume $1 \leq t \leq n-1$. We shall make the further assumption that G is connected. It is not hard to show that both VERTEX COVER and VERTEX COVER(=) remain NP-complete even when restricted to connected graphs.

The vertex-edge incidence graph of G is the undirected bipartite graph $G' = (L, R, E')$ with $L = V$, $R = E$ and edges (e, u) and (e, v) in E' for each $e = (u, v) \in E$. Fig. 1 shows the vertex-edge incidence graph for a graph G with $n = 4$ and $m = 3$.

The advantage of assuming that G is connected arises out of the following lemma:

Lemma 3.1 *Let $G' = (L, R, E')$ be the vertex-edge incidence graph of a connected graph $G = (V, E)$. Let S be a stopping set in G'. Then $S = L$.*

Proof Suppose to the contrary that $L \setminus S \neq \emptyset$. Then, as G is connected there exists $v \in L \setminus S$ and $u \in S$ such that $(u, v) \in E$. Let $e = (u, v)$. Then $e \in N(S)$. Since S is a stopping set $|N(\{e\}) \cap S| \geq 2$. But the only neighbors of e in G' are u and v. Hence $v \in S$ contradicting $v \in L \setminus S$. ■

We construct an undirected bipartite graph $G'' = (L, R, E'')$ as follows: $L = \bigcup_{i=0}^{m+1} L_i$, $R = \bigcup_{j=0}^{m+1} R_j$, where, $R_0 = \{z_1, ..., z_{m-1}\}$, $R_j = \{u_j^r, u \in V\}$ for $2 \leq j \leq m+1$, R_1 and L_0 are copies of E, the edge set of G and $L_i = \{u_i^\ell, u \in V\}$ for $1 \leq i \leq m+1$. Edges in G'' are connected as follows:

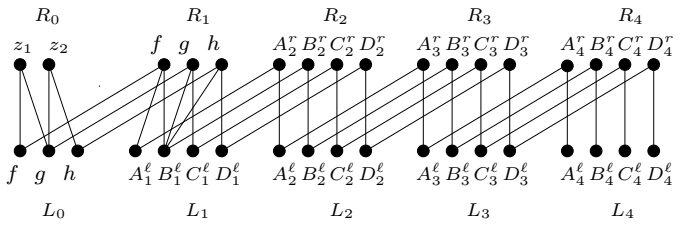

Figure 2: Construction of G''

- Connect $u_i^\ell \in L_i$ to $u_i^r \in R_i$, $2 \leq i \leq m+1$.
- Connect $u_i^\ell \in L_i$ to $u_{i+1}^r \in R_{i+1}$, $1 \leq i \leq m$.
- For each $e = (u, v)$ in E, connect $e \in R_1$ to u and v in L_1.
- For each $e \in E$, connect $e \in L_0$ to $e \in R_1$.
- For the purpose of defining the edges between R_0 and L_0, temporarily re-label vertices in L_0 as $e_1, e_2, ..e_m$ in some arbitrary way. Add the edges (e_i, z_i) for $1 \leq i \leq m-1$ and the edges (e_i, z_{i-1}) for $2 \leq i \leq m$.

The example in Fig. 2 illustrates the construction of G'' for the graph G in Fig. 1. The graph G'' consists of a copy of the vertex-edge incidence graph of G (vertex sets L_1 and R_1). Additionally, there are m copies of the vertex set V on the left $(L_2, L_3, ..., L_{m+1})$ and right $(R_2, R_3, ...R_{m+1})$. The connections between R_0 and L_0 ensure that any stopping set in G'' containing any one vertex in L_0 must contain the whole of L_0. The vertex u_i^r in R_i has neighbors u_{i-1}^ℓ and u_i^ℓ for each $2 \leq i \leq m+1$ and each $u \in V$. This ensures that if a stopping set S in G'' contains u_i^ℓ for some $i \in \{1, 2, ..m+1\}$ then all the $m+1$ vertices $u_1^\ell, u_2^\ell, ..., u_{m+1}^\ell$ must be present in S. These observations, summarized below, play a crucial role in the arguments that follow.

Observation 3.2 *A stopping set S' in G'' satisfies $u_i^\ell \in S'$ for some $1 \leq i \leq m+1$ if and only if it satisfies $u_i^\ell \in S'$ for* every *$1 \leq i \leq m+1$. Moreover either $L_0 \subseteq S'$ or $L_0 \cap S' = \emptyset$.*

The following two claims establish the connection between vertex covers in G and stopping sets in G''.

Lemma 3.3 *If G contains a vertex cover S of size t for some $1 \leq t \leq n-1$ then G'' contains a stopping set of size $t(m+1)+m$.*

Proof Consider the set $S' = L_0 \cup \{u_i^\ell : u \in S, 1 \leq i \leq m+1\}$ in G''. Clearly S' has $t(m+1)+m$ elements. Let $w \in N(S')$. Then either $w = u_i^r$ for some $u \in S$, $i \in \{2, 3, ...m+1\}$ or $w \in R_1$ or $w \in R_0$. In the first case, both u_i^ℓ and u_{i-1}^ℓ are neighbors of w. If $w \in R_1$, then by construction, w must correspond to some edge $e = (u, v)$ in E. Since $L_0 \subseteq S'$, $e \in L_0$ is a neighbor of w. Since S is a vertex cover in G, either u or v or both must belong to S. Hence one or both of u_1^ℓ and v_1^ℓ are neighbors of w which belong to S'. Finally if $w \in R_0$, then both the neighbors of w are in L_0, and therefore in S'. Thus in all cases w has at least two neighbors in S'. Consequently S' is a stopping set. ■

We now prove that every stopping set in G'' of size less than $n(m+1)$ must correspond to some vertex cover of size t in G for some $1 \leq t \leq n-1$ and must have size exactly $t(m+1)+m$

Lemma 3.4 *Let S' be a stopping set in G'' of size less than $n(m+1)$. Then the following must hold:*

- $L_0 \subseteq S'$,
- $|S'| = t(m+1)+m$ *for some* $1 \leq t \leq n-1$ *and* $|S' \cap L_i| = t$ *for every* $1 \leq i \leq m+1$
- $S = \{u \in V : u_1^\ell \in S'\}$ *is a vertex cover of size t in G.*

Proof Suppose L_0 is not contained in S'. Then by Observation 3.2, $L_0 \cap S' = \emptyset$. Since $S' \neq \emptyset$, There must be some $u \in V$ and $i \in \{1, 2, .., m+1\}$ such that $u_i^\ell \in S'$. By Observation 3.2, $u_1^\ell \in S'$. Since vertices in the set R_1 are connected only to L_1 and L_0, every neighbor of S' in R_1 must have two neighbors in $S' \cap L_1$ in order for S' to satisfy the conditions of a stopping set. In other words, $S' \cap L_1$ must be a stopping set in the subgraph of G'' induced by the vertices $L_1 \cup R_1$. Note that this subgraph is the vertex-edge incidence graph of G. Applying Lemma 3.1 we get $S' \cap L_1 = L_1$. Hence Observation 3.2 shows that $S' = \bigcup_{i=1}^{m+1} L_i$. But in that case $|S'| = n(m+1)$, a contradiction. Hence $L_0 \subseteq S'$ and $|L_1 \cap S'| < n$. Let $|S' \cap L_1| = t$ for some $1 \leq t \leq n-1$. Applying Observation 3.2 once again, $|S' \cap L_i| = t$ for all $1 \leq i \leq m+1$. Hence $|S'| = t(m+1)+m$.

To complete the proof of the lemma, it is sufficient to prove that $S = \{u \in V : u_1^\ell \in S'\}$ is a vertex cover of G. Since $L_0 \subseteq S'$, $R_1 \subseteq N(S')$. Since every vertex e in R_1 has only one neighbor in the set L_0, for S' to satisfy the stopping set condition e must have a neighbor in $L_1 \cap S'$. Then, by construction $\{u \in V : u_1^\ell \in S'\}$ must be a vertex cover in G as required. ■

As a consequence of Lemma 3.3 and Lemma 3.4 we have:

Corollary 1 *G has a vertex cover of size t if and only if G'' has a stopping set of size $t(m+1)+m$, $1 \leq t \leq n-1$. Hence $(G,t) \in$ VERTEX COVER(=) if and only if $(G'', t(m+1)+m) \in$ STOPPING SET.*

Corollary 2 *G has a vertex cover of size at most t if and only if G'' has a stopping set of size at most $t(m+1)+m$, $t \in \{1,2,..,n-1\}$. Hence $(G,t) \in$ VERTEX COVER if and only if $(G'', t(m+1)+m) \in$ STOPPING DISTANCE.*

We are now ready to prove:

Theorem 3.5 *STOPPING DISTANCE and STOPPING SET are NP-complete*

Proof

Since G'' can be constructed from G in polynomial time ($O(mn)$ time suffices), it follows that VERTEX COVER(=) $\preceq_p$ STOPPING SET and VERTEX COVER $\preceq_p$ STOPPING DISTANCE from Corollary 1 and Corollary 2 respectively. It is easy to verify whether a given set of left vertices of a bipartite graph forms a stopping set in time linear in the size of the graph. Hence both STOPPING DISTANCE and STOPPING SET belong to the class NP. ■

As a consequence, we have:

Corollary 3 *There is no polynomial time algorithm for computing the stopping distance of a Tanner graph unless P=NP.*

4 Acknowledgment

The authors would like to thank Dr. L. Sunil Chandran for useful discussions, and the anonymous referees for their helpful comments .

References

[1] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory.*, vol. 48, no. 6, pp. 1570-1579, June 2002.

[2] C. Di, A. Montanari and R. Urbanke, "Weight distribution of LDPC code ensembles: Combinatorics meets statistical physics," in *Proc. IEEE Int. Symp. Inform. Theory*, Chicago, IL., July 2004, p. 102.

[3] A. Orlitsky, K. Viswanathan, and J. Zhang, "Stopping set distribution of LDPC code ensembles," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, March 2005, pp. 929-953.

[4] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, "Construction of irregular LDPC codes with low error floors," in *Proc. IEEE Int. Conf. Comm.*, Seattle, Washington, May 2003, pp. 3125-3129.

[5] A. Ramamoorthy and R. Wesel, "Construction of short block length irregular LDPC codes," in *Proc. IEEE Int. Conf. Comm.*, Paris, June 2004, pp. 410-414.

[6] A. Orlitsky, R. Urbanke, K. Viswanathan, and J. Zhang, "Stopping sets and girth of Tanner graphs," in *Proc. IEEE Int. Symp. Inform. Theory*, Lausanne, June 2002, p. 2.

[7] M. Schwartz and A. Vardy, "On the stopping distance and the stopping redundancy of codes," *IEEE Trans. Inform. Theory* vol. 52, no. 3, pp. 922-932, March 2006.

[8] R. Michael Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5 pp. 533-547, Sept 1981.

[9] T. H. Cormen, C. E. Leicerson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, 1990.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, 1979.

[11] S. Cook, "The complexity of theorem proving procedures," in *Proc. Third ACM Ann. Symposium on Theory of Computing*, Shaker Heights, Ohio, May 1971, pp.151-158.

[12] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 3, pp. 439-454, March 2004.

[13] J. Han and P. Siegel, "Improved upper bounds on stopping redundancy," preprint available at: *http://www.arXiv.org*, cs.IT/0511056.

[14] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Trans. Inform. Theory*, vol. 46 no. 6 pp. 1757-1766, Nov. 1997.

[15] E. R. Berlekamp, R. J. McEliece, and H. C. A van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. 24 no. 3, pp. 384-386, May 1978.