Enumerating and Counting Cycles in Bipartite Graphs

Thomas R. Halford and Keith M. Chugg
Communications Sciences Institute
Electrical Engineering Dept.
University of Southern California
Los Angeles, CA, USA, 90089-2565
{halford,chugg}@usc.edu

I. INTRODUCTION

Exactly counting cycles and paths in arbitrary graphs is known to be hard [4]. Alon, Yuster and Zwick presented methods for counting cycles of length less than 8 in [1]; however, their methods are prohibitively complex for larger cycles. This work addresses the simpler problem of counting and enumerating cycles and paths in bipartite graphs. We restrict our attention to bipartite graphs since the graphs used to define codes (e.g. Tanner graphs [7], factor graphs [6], normal graphs [5]) are bipartite. This work provides a method for enumerating paths and cycles of arbitrary lengths in bipartite graphs using symbolic matrix multiplication. This work also provides closed form expressions for the number of cycles of length g, $g+2$ and $g+4$ incident on each vertex in bipartite graphs with girth g. The cycle structure of a graphical model, especially short cycles, has been observed empirically to affect the performance of the iterative message-passing heuristic used in the decoding of modern codes [3]. This motivates this work into characterizing this cycle structure.

The rest of this summary is organized as follows. Section II provides a brief review of the graph theory used in this paper. Section III outlines the symbolic cycle enumeration algorithm. Section IV outlines the closed form expressions for cycle counting. Section V outlines extensions of this work that we are currently considering.

II. REVIEW OF GRAPH THEORY AND NOTATION

A graph $G(\mathcal{V},\mathcal{E})$ consists of a finite non-empty set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$ which is any subset of the pairs $\{\{u,v\}: u,v \in \mathcal{V}, u \neq v\}$ [2]. A *walk* in $G(\mathcal{V},\mathcal{E})$ is a sequence of vertices $v_1, v_2, \cdots, v_n$ such that $v_i \in \mathcal{V}$ and $\{v_i, v_{i+1}\} \in \mathcal{E}$. A walk is called *closed* if $v_1 = v_n$. A walk is called a *trail* if the edges that form the walk are distinct; a trail is called a *path* if the vertices are distinct. A closed walk is a *cycle* if the vertices $v_1, v_2, \cdots, v_{n-1}$ are distinct. We introduce the term $(k, n-1-k)$-*lollipop*[1] walk to describe walks where the vertices $v_1, v_2, \cdots, v_{n-1}$ are distinct and $v_n = v_k$ for some $k \in \{1, 2, \cdots, n-1\}$. Cycles of length $2k$ are thus $(0, 2k)$-lollipop walks. Figure 1 illustrates a $(4,2)$-lollipop walk; figure 2 illustrates a $(2,4)$-lollipop walk.

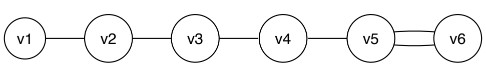


Fig. 1. A (4,2)-lollipop walk from v_1 to v_5.

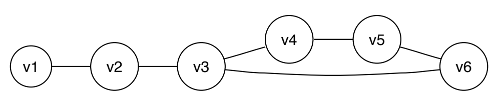


Fig. 2. A (2,4)-lollipop walk from v_1 to v_3.

A graph $G(\mathcal{V},\mathcal{E})$ is bipartite with vertex classes $\mathcal{U}$ and $\mathcal{W}$ if $\mathcal{V} = \mathcal{U} \cup \mathcal{W}$, $\mathcal{U} \cap \mathcal{W} = \emptyset$ and each edge joins a vertex in $\mathcal{U}$ to one in $\mathcal{W}$ [2]. All cycles in bipartite graphs contain an even number of edges and all $(k, n-1-k)$-lollipop paths have $n-1-k = 0 \mod 2$. Associated with the bipartite graph $G(\mathcal{U} \cup \mathcal{W}, \mathcal{E})$ is the $|\mathcal{U}| \times |\mathcal{W}|$ edge matrix $E = [e_{ij}]$ where $e_{ij} = 1$ if $(u_i, w_j) \in \mathcal{E}$ and 0 otherwise.

[1]This terminology is nonstandard and should not be confused with the lollilpop graph of [8].

III. Enumerating Cycles in Bipartite Graphs

Consider a set $\mathcal{A}$ that corresponds to a labelling of the vertices of a bipartite graph $G(\mathcal{U} \cup \mathcal{W}, \mathcal{E})$ and the set $\mathcal{B}$ consisting of all words formed from $\mathcal{A}$ along with the special elements 0 and 1:

$$\mathcal{A} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\} \cup \{w_1, w_2, \cdots, w_{|\mathcal{W}|}\} \tag{1}$$

$$\mathcal{B} = \{0, 1\} \cup \mathcal{A} \cup \mathcal{A}^2 \cup \cdots \cup \mathcal{A}^n \cdots \tag{2}$$

Define multiplication on elements of $\mathcal{B}$ by the following rules:

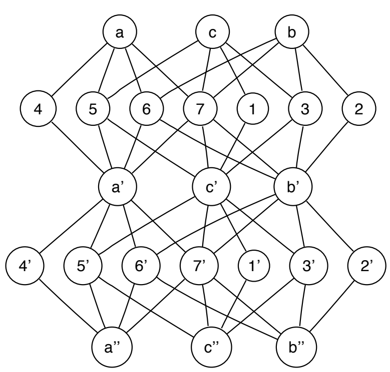
1) For any $b \in \mathcal{B}$, $b0 = 0b = 0$.
2) For any $b \in \mathcal{B}$, $b1 = 1b = b$.
3) For any word $avbvc \in \mathcal{B}$ where $a, b, c \in \mathcal{B}$ and $v \in \mathcal{A}$, $avbvc = 0$.

Using these rules we can enumerate paths and cycles in bipartite graphs via symbolic matrix multiplication. Let F be the the following matrix, which we denote the *symbolic edge matrix*:

$$F = \left([u_1 u_2 \cdots u_{\mathcal{U}}]^T [w_1 w_2 \cdots u_{\mathcal{W}}]\right) \cdot E \tag{3}$$

The non-zero elements in F correspond to the edges in $G(\mathcal{U} \cup \mathcal{W}, \mathcal{E})$ and each element describes which two vertices the corresponding edge joins. It is straightforward to show that $\left(FF^T\right)^{k-1}$ enumerates the paths of length $2k-2$ from vertices in $\mathcal{U}$ to vertices in $\mathcal{U}$ while $\left(F^T F\right)^{k-1}$ enumerates paths of length $2k-2$ from vertices in $\mathcal{W}$ to vertices in $\mathcal{W}$. At each symbolic matrix multiplication, paths with repeated vertices are removed by the third rule of multiplication. In order to count cycles of length $2k$ from vertices in $\mathcal{U}$ to vertices in $\mathcal{U}$, we multiply $\left(FF^T\right)^{k-1}$ by FF^T but don't apply the third rule of multiplication. We can thus enumerate cycles and paths of arbitrary length in a bipartite graph via symbolic matrix multiplication.

As an example consider the Tanner graph of $(7,4)$ Hamming code with codeword symbol nodes labelled $\{1,2,3,4,5,6,7\}$ and parity check nodes labelled $\{a, b, c\}$. Figure 3 illustrates the concatenation of 4 such graphs. The diagonal elements of FF^TFF^T (where the third multiplication rule of the final symbolic matrix multiplication has been neglected) enumerate the paths from a to a'', b to b'' and c to c'' in this concatenated graph. Hence, FF^TFF^T enumerates the 4 distinct 4 cycles.



$$\begin{aligned} FF^TFF^T|_{(0,0)} &= a6b'7'a'' + a7b'6'a'' + \\ &\quad a5c'7'a'' + a7c'5'a'' \\ FF^TFF^T|_{(1,1)} &= b6a'7'b'' + b7a'6'b'' + \\ &\quad b3c'7'b'' + b7c'3'b'' \\ FF^TFF^T|_{(2,2)} &= c5a'7'c'' + c7a'5'c'' + \\ &\quad c3b'7'c'' + c7b'3'c'' \end{aligned}$$

Fig. 3. Concatenated Tanner graph and diagonal elements of FF^TFF^T for the $(7,4)$ Hamming code.

IV. Counting Cycles in Bipartite Graphs

The cycle enumeration algorithm of the previous section is very complex because the matrix multiplication is done over symbolic elements. Counting cycles in bipartite graphs is significantly easier: here we show closed form expressions for counting cycles. We first define 8 matrices that count the number of paths and lollipop walks in a bipartite graph $G(\mathcal{U} \cup \mathcal{W}, \mathcal{E})$. Let $P_{2k}^{\mathcal{U}}$ be a $|\mathcal{U}| \times |\mathcal{U}|$ matrix where the $(i, j) - th$ element is the number of paths of length $2k$ from $u_i \in \mathcal{U}$ to $u_j \in \mathcal{U}$; similarly define $P_{2k}^{\mathcal{W}}$. By definition, the diagonal elements of $P_{2k}^{\mathcal{U}}$ and $P_{2k}^{\mathcal{W}}$ are zero. Let $P_{2k+1}^{\mathcal{U}}$ be a $|\mathcal{U}| \times |\mathcal{W}|$ matrix where the $(i, j) - th$ element is the number of paths of length $2k + 1$ from $u_i \in \mathcal{U}$ to $w_j \in \mathcal{W}$; similarly define $P_{2k+1}^{\mathcal{W}}$. Let $L_{(2k',k-2k')}^{\mathcal{U}}$ be a $|\mathcal{U}| \times |\mathcal{U}|$ matrix where the $(i, j) - th$ element is the number of $(2k', k - 2k')$-lollipop walks from $u_i \in \mathcal{U}$ to $u_j \in \mathcal{U}$; similarly define $L_{(2k',k-2k')}^{\mathcal{W}}$. Let $L_{(2k'+1,k-2k'-1)}^{\mathcal{U}}$ be a $|\mathcal{U}| \times |\mathcal{W}|$ matrix where the $(i, j) - th$ element is the number of $(2k' + 1, k - 2k' - 1)$-lollipop walks from $u_i \in \mathcal{U}$ to $w_j \in \mathcal{W}$; similarly define $L_{(2k'+1,k-2k'-1)}^{\mathcal{W}}$.

Based on these definitions, there is a simple recursion for the path matrices described by equations (4)-(7) with the initial conditions $P_0^{\mathcal{U}} = P_0^{\mathcal{W}} = I$ and $L_{(0,0)}^{\mathcal{U}} = L_{(0,0)}^{\mathcal{W}} = 0$.

$$P_{2k+1}^{\mathcal{U}} = P_{2k}^{\mathcal{U}} E - \sum_{i=0}^{k-1} L_{(2i+1,2k-2i)}^{\mathcal{U}} \quad (4) \qquad P_{2k+1}^{\mathcal{W}} = P_{2k}^{\mathcal{W}} E^T - \sum_{i=0}^{k-1} L_{(2i+1,2k-2i)}^{\mathcal{W}} \quad (6)$$

$$P_{2k}^{\mathcal{U}} = P_{2k-1}^{\mathcal{U}} E^T - \sum_{i=0}^{k-1} L_{(2i,2k-2i)}^{\mathcal{U}} \quad (5) \qquad P_{2k}^{\mathcal{W}} = P_{2k-1}^{\mathcal{W}} E - \sum_{i=0}^{k-1} L_{(2i,2k-2i)}^{\mathcal{W}} \quad (7)$$

To prove (4) note that a path of length $2k+1$, $v_1, v_2, \cdots, v_{2k+1}$, is formed by augmenting a single edge to a path of length $2k$, $v_1, v_2, \cdots, v_{2k}$, such that $v_{2k+1} \neq v_i \ \forall i \in [1, 2k]$. All walks $v_1, v_2, \cdots, v_{2k}, v_{2k+1}$ where the first $2k$ vertices form a path are counted by $W_{2k}^{\mathcal{U}} E$; $W_{2k}^{\mathcal{U}} E$ counts both paths and lollipop walks, hence (4). Equations (5)-(7) are proved in a similar manner. From (4)-(7) we see that cycle counting has been reduced to counting lollipop walks. We have determined closed form expressions for the lollipop walk matrices that are required to count cycles of length g, $g + 2$ and $g + 4$ in bipartite graphs of girth g. Each of these expressions contains only matrix multiplications; the complexity of counting short cycles in a bipartite graph $G(\mathcal{U} \cup \mathcal{W}, \mathcal{E})$ is thus $\mathcal{O}(n^3)$ where $n = |\mathcal{U} \cup \mathcal{W}|$.

As examples, consider the $(23, 12)$ systematic Golay code and a $(21, 7)$ LDPC code due to [9]. If we let the set of codeword symbol nodes correspond to $\mathcal{W}$ and the set of parity check node to $\mathcal{U}$ then E is the parity check matrix. Our method counts 452 4-cyles, 10,319 6-cycles and 199,994 8-cycles in the Tanner graph of the Golay code. The $(21, 7)$ LDPC code was designed with girth 12. Our method counts 28 12-cycles, 0 14-cycles and 21 16-cycles in the Tanner graph of the LDPC code.

V. Future Work

The simple recursion of (5)-(8) hides the high complexity of counting arbitrary lollipop walks. We have initially restricted our focus to counting short cycles; an immediate extension of this work is to count cycles of length $g + 6$, $g + 8$, etc. A second extension is to reformulate the recursion as a generating function, the language of algebraic combinatorics. We are currently investigating how elementary row operations on E over $GF(2)$ affect the number of short cycles and formalizing the complexity of the cycle enumeration algorithm.

References

[1] N. Alon, R. Yuster, and U. Zwick, "Finding and counting given length cycles," *Algorithmica*, vol. 17, no. 3, pp. 209–223, 1997.

[2] V. Bryant, *Aspects of Combinatorics*, 2nd ed. New York, NY: Cambridge University Press, 1995.

[3] K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*. Kluwer Academic Publishers, 2001.

[4] J. Flum and M. Grohe, "The parameterized complexity of counting problems," in *Proc. IEEE Symp. Foundations of Comp. Sci.*, November 2002, pp. 538–547.

[5] G. D. Forney, Jr., "Codes on graphs: normal realizations," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 520–548, February 2001.

[6] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

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

[8] A. G. Thomason, "Hamiltonian cycles and uniquely edge-colourable graphs," *Annals of Discrete Mathematics*, vol. 3, pp. 259–268, 1978.

[9] H. Zhang and J. M. F. Moura, "The design of structured regular LDPC codes with large girth," in *Proc. Globecom Conf.*, Taipei, Taiwan, December 2003, pp. 4022–4027.