On Encodings of Spanning Trees

Glenn H. Hurlbert*
Department of Mathematics and Statistics
Arizona State University
Tempe, AZ 85287-1804
email: hurlbert@asu.edu

May 29, 2007

*Partially supported by National Security Agency grant #MDA9040210095.

Abstract

Deo and Micikevicius recently gave a new bijection for spanning trees of complete bipartite graphs. In this paper we devise a generalization of Deo and Micikevicius's method, which is also a modification of Olah's method for encoding the spanning trees of any complete multipartite graph $K(n_1, \ldots, n_r)$. We also give a bijection between the spanning trees of a planar graph and those of any of its planar duals. Finally we discuss the possibility of bijections for spanning trees of DeBriujn Graphs, Cubes, and regular graphs such as the Petersen graph that have integer eigenvalues.

2000 AMS Subject Classification: 05C05, 05C85, 68R10

1 Introduction

Given a graph $G = (V, E)$ with vertices $V = V(G)$ and edges $E = E(G)$, a *spanning tree* $T = (V, E')$ of G is a connected acyclic subgraph of G. One very natural and old problem is to determine the number $\tau(G)$ of labelled spanning trees for a fixed graph G, or better yet, a formula for each in a family $\mathcal{G} = \{G_n\}_{n=0}^{\infty}$.

The first and most famous result of this kind is due to Cayley [5], who proved that $\tau(K_n) = n^{n-2}$, where K_n is the complete graph on n vertices. A wonderful collection of many different proofs of this result is found in [16]. The first bijective method of this result was given by Prüfer [18]. It is possibly the simplest proof of Cayley's Theorem. A generalization of Prüfer's encoding was given by Olah [17] for complete multipartite graphs $K(n_1, \ldots, n_r)$. This graph has $n = \sum_{i=1}^{r} n_i$ vertices, partitioned into sets of sizes $n_1, \ldots, n_r$, and edges between every pair of vertices from different parts. The encoding proves

Theorem 1 $\tau(K(n_1, \ldots, n_r)) = n^{r-2} \prod_{i=1}^{r} (n - n_i)^{n_i - 1}$.

Another wonderful bijection for these graphs is found in [10, 11], where the authors give a full description of the q-analog properties of their bijections.

Recently, a new method has been found by Deo and Micikevicius [9] that allows one both to encode and decode a spanning tree of K_n in such a way that the diameter, center, and radius of the tree can be calculated directly from the code without having to resort to other algorithms that operate on the tree itself. In this paper we devise a generalization of Deo and Micike-

vicius's method which is also a modification of Olah's method for encoding the spanning trees of any complete multipartite graph $K(n_1, \ldots, n_r)$.[1] Our method shares the benefits of finding the diameter, center, and radius directly. We describe this method in Section 2.

We discuss the possibility of bijections for spanning trees of integral graphs, such as the Petersen graph and d-dimensional cubes, and of De-Bruijn Graphs in Section 3. We also give a bijection between the spanning trees of a planar graph and those of any of its planar duals. The section includes some open problems.

2 Complete Multipartite Graphs

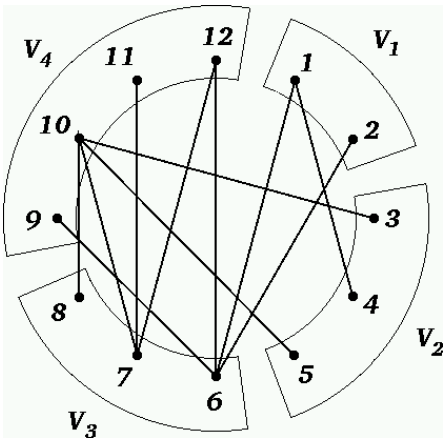
2.1 Encoding a Tree

Let T be a fixed spanning tree of $K = K(n_1, \ldots, n_r)$, and let $n = \sum_{j=1}^{r} n_j$ be the number of its vertices. Let the partition of the vertices V given by K be $V_1, \ldots V_r$, where each $|V_j| = n_j$, and define $s_j = \sum_{i=1}^{j} n_i$. Then we may assume that $V_j = \{s_{j-1} + 1, \ldots, s_j\}$. Define $V(k) = j$, where $k \in V_j$. The code for T will be a set of sequences $C = \{C_0, C_1, \ldots, C_r\}$, where the length of C_0 is $r - 2$ and the length of each C_j $(1 \leq j \leq r)$ is $n_j - 1$. Thus the total length of the code is $(r-2) + \sum_{j=1}^{r}(n_j - 1) = n - 2$, as expected. Figure 1 shows the algorithm **E** used to encode T by C. The sequence C_0 can contain any of the labels, while every other sequence C_j will contain the labels of

[1] It is worth correcting a small typographical error that appears in Deo and Micikevicius's paper: In each of lines 4 and 5 of Figure 3 in [9] the variable *used*[i] should be replaced by *used*[$C(i)$].

1. **set** $Q_0 = \emptyset$
2. **for** i from 1 to n **do**
   (a) **if** $\deg(i) = 1$
      i. **then** add i to the tail of Q_0
3. **set** $i = 1, j = 0$
4. **while** $i \leq n - 2$ **do**
   (a) **set** $Q_{j+1} = \emptyset$
   (b) **while** $Q_j \neq \emptyset$ and $i \leq n - 2$ **do**
      i. remove k from the head of Q_j
      ii. **if** $|C_{V(k)}| < n_i - 1$
         A. **then** add k^+ to the tail of $C_{V(k)}$
         B. **else** add k^+ to the tail of C_0
      iii. remove k from T
      iv. **if** $\deg(k^+) = 1$
         A. **then** add k^+ to the tail of Q_{j+1}
      v. increment i
   (c) increment j

Figure 1: Algorithm **E** for encoding a tree T by a code C



C_0	10	6	
C_1	6		
C_2	10	1	
C_3	10	12	
C_4	6	7	7

Figure 2: The encoding C of a particular tree T

the neighbors, at appropriate stages of the algorithm, of the vertices in V_j (in some specified order). The set of all possible such codes has cardinality matching the right hand side of Theorem 1. When k is a leaf of T denote its unique neighbor in T by k^+.

Figure 2 shows an example of **E** encoding a tree T.

Proposition 2 *Algorithm* **E** *is injective.*

Proof. For a given tree T and for $0 \leq t \leq \text{radius}(T)$ denote by $T^{(t)}$ the subgraph of T induced by the vertices $Q_t \cup_{j=0}^{t-1} (Q_j \cup \text{nbhd}(Q_j))$, where $\text{nbhd}(W)$ is the set of vertices adjacent to some vertex of W. Suppose that Algorithm **E** encodes the trees T_1 and T_2 by the same code C. According to **E** any vertex that does not appear in C is a leaf, and so T_1 and T_2 have the same set of leaves; i.e. $Q_0(T_1) = Q_0(T_2)$ and $T_1^{(0)} = T_2^{(0)}$. Using induction we'll assume that $Q_{t-1}(T_1) = Q_{t-1}(T_2)$ and $T_1^{(t-1)} = T_2^{(t-1)}$ for some $t > 0$. Because C encodes both trees, the neighbors of Q_{t-1} are determined by C and

so $T_1^{(t)} = T_2^{(t)}$. This also determines the leaves in the remaining tree at this stage, and so $Q_t(T_1) = Q_t(T_2)$. Hence $T_1 = T_2$. □

2.2 Decoding a Sequence

Suppose C is a code satisfying the conditions described in Section 2.1. We describe Algorithm **D** (see Figure 3), which constructs from C the tree T that Algorithm **E** encodes as C. Values of the n-tuple U are initialized in the loop at line 2 to one less than the degrees of the vertices so that the leaves become evident, forming the queue Q_0 (we set $n_0 = r - 1$) in the loop at line 3. The process of constructing T is facilitated by the iterative construction in the loop at line 5 of each queue $Q_1, Q_2, \ldots$, essentially reproducing them in the same manner done by **E** . The variables k and k' are redefined each time they are removed; that is, in line 5(a)i for example, k is defined to be the head of Q_j and then it is removed.

Proposition 3 *Algorithm* **D** *is injective.*

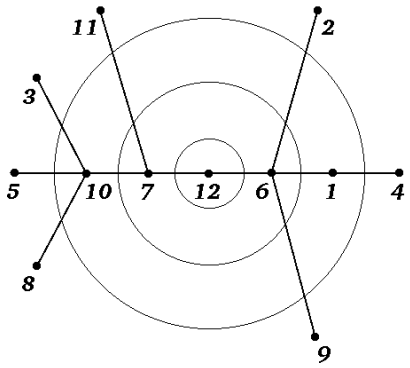
Proof. By the above discussion, Algorithm **D** is the realization of the argument given in the proof of Proposition 2. Hence Algorithm **D** is the inverse of Algorithm **E** . □

One can check Algorithm **D** against Figure 2.

Theorem 4 *When Algorithm* **D** *halts, we have* $\text{diameter}(T) = 2j + |Q_j| - 1$, $\text{radius}(T) = j + |Q_j| - 1$, *and* $\text{center}(T) = Q_j$, *where* j *is given by its value when* **D** *halts.*

1. **for** i from 1 to n **do**
   (a) **set** $U(i) = 0$
2. **for** j from 0 to r **do**
   (a) **set** $Q_j = 0$
   (b) **for** i from 1 to $n_j - 1$ **do**
      i. increment $U(C_j(i))$
3. **for** i from 1 to n **do**
   (a) **if** $U(i) = 0$
      i. **then** add i to the tail of Q_0
4. **set** $i = 1, j = 0$
5. **while** $i \leq n - 2$ **do**
   (a) **while** $Q_j \neq \emptyset$ **do**
      i. remove k from the head of Q_j
      ii. **if** $C_{V(k)} \neq \emptyset$
         A. **then** remove k' from the head of $C_{V(k)}$
         B. **else** remove k' from the head of C_0
      iii. add $\{k, k'\}$ to $E(T)$
      iv. decrement $U(k')$
      v. **if** $U(k') = 0$
         A. **then** add k' to the tail of Q_{j+1}
      vi. increment i
      vii. **if** $i = n - 1$
         A. **then if** $Q_j \neq \emptyset$
            (I) **then** remove k from the head of Q_j
            (II) **else** remove k from the head of Q_{j+1}
   (b) increment j
6. remove k' from the head of Q_j
7. add $\{k, k'\}$ to $E(T)$

Figure 3: Algorithm **D** for decoding a tree T from a code C



Q_0	2	3	4	5	8	9	11
Q_1	1	10					
Q_2	6	7					
Q_3	12						

Figure 4: The layers $Q_1, \ldots, Q_3$ of a particular tree T

Proof. This can be seen most easily by recognizing that, for every $i > 0$, every vertex $k \in Q_i$ is adjacent to some vertex in Q_{i-1} (see Figure 4). Thus set of Q_is "peel off" the layers of T from the outside in. The formulas take into account the cases from line 5(a)viiA of Algorithm **D** that distinguish whether the final queue has size 1 or 2, respectively. □

Therefore, an algorithm that calculates these values directly from C, without reference to T, is given by removing lines 5(a)iii and 7 from Algorithm **D** . This differs from Olah's (and Prüfer's) method, which needs to run a separate algorithm on T for calculating diamater, radius and center.

3 Other Graphs

Notice that the spanning tree formulas for each of the graph families discussed so far follow the familiar pattern of

$$\tau(G) \;=\; \frac{1}{|V|} \prod_{i=1}^{|V|-1} f_i \;,$$

where $V = V(G)$ and f_i is some set of factors. Such a pattern naturally opens itself to possible encodings. The inner product of factors counts rooted spanning trees, while the outer factor unroots the trees. In this section we explore some examples that have natural sets with which to match in 1-to-1 correspondence.

3.1 Integral Graphs

Our first example is the family of r-regular graphs. For any graph G let $A = A(G)$ denote the adjacency matrix of G, $D = D(G)$ denote the diagonal matrix of vertex degrees of G, and J be the $n \times n$ matrix of all ones. One can use Temperly's theorem [21] that $\tau(G) = \det(J + D - A)/n^2$ to show that if G is r-regular and A has eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ (it is known that $\lambda_{n-1} < \lambda_n = r$) then

$$\tau(G) \;=\; \frac{1}{n} \prod_{i=1}^{n-1} (r - \lambda_i)$$

(see [3, 6]). Of course, this formula is useful from the point of view of encodings only if every the eigenvalue of G is an integer; i.e. G is an *integral graph*. For $r = 3$, for example, there are only 13 such graphs (see [4, 19]), one of which is the Petersen graph P, having eigenvalues (with multiplicities) $3, 1^5, -2^4$. This gives rise to the following representative open problem.

Problem 5 *Find a bijective proof for the equation*

$$\tau(P) \;=\; 2000 \;=\; \left(\frac{1}{10}\right) 5^4 2^5 \;.$$

In this case one could build a set of 9 arcs in P by taking the natural image of one of the 5^4 spanning trees of K_5 and orienting the remaining 5 "matching" edges between the two 5-cycles of P in 2^4 ways. But then how to map the resulting digraph to a rooted tree?

There are only finitely many connected integral graphs for each r [7]. However, there are infinite families of integral graphs, such as complete graphs. Another example [15] is the d-dimensional cube Q^d, whose vertices are the binary d-tuples and whose edges join d-tuples whose coordinates differ in one position. Let $w(v)$ be the number of 1s in the d-tuple v.

Problem 6 *Find a bijective proof for the equation*

$$\tau(Q^d) \;=\; 2^{-d} \prod_{j=1}^{d} (2j)^{\binom{d}{j}} \;=\; \left(\frac{1}{2^d}\right) \prod_{v \neq 0^d} 2w(v) \;.$$

Here, one could build a set of $2^d - 1$ arcs in Q^d by orienting, for each vertex $v \neq 0^d$, one of the $w(v)$ "down" edges (from v to a neighbor with fewer ones) in one of two ways. Still, how to map the resulting digraph to a rooted tree?

Other interesting examples can be found in the survey [2].

3.2 DeBruijn Graphs

Define the digraph $B(d, k)$ whose vertices are all the k-ary d-tuples and whose oriented edges go from each vertex $(a_1, a_2, \ldots, a_d)$ to each vertex $(a_2, \ldots, a_d, a_{d+1})$. Kirchoff's Matrix Tree Theorem [14] states that $\tau(G)$ equals the value of any cofactor of the matrix $D - A$. In order to generalize to digraphs it is

useful to think of $\tau(G)$ as the number of spanning trees of G that are rooted at a fixed vertex. This formulation of τ now works for both directed and undirected graphs G. An analogue of the Matrix Tree Theorem for digraphs is known as the BEST Theorem [1, 20], which yields a formula for $\tau(B(d,k))$.

Problem 7 *Find a bijective proof for the equation*

$$\tau(B(d,k)) \;=\; k^{k^d-1-d} \;=\; \left(\frac{1}{k^d}\right) \prod_{v \neq 0^d} k \;.$$

A natural set of digraphs for a bijection arises by choosing, for each vertex $v \neq 0^d$, one of its k out-arcs. The question of how to map the resulting digraph to a tree rooted at 0^d remains.

We pause to present explicitly what is implicitly found in the work of several authors, namely an encoding of the eulerian circuits of a digraph induced by the collection of its rooted spanning trees. For a given digraph G we say that a spanning tree is *rooted at vertex* v if, from each vertex $w \neq v$, the unique path from w to v has every edge oriented toward v.

Theorem 8 *Let G be an eulerian digraph with outdegree sequence $d_1, \ldots, d_n$. Let $\tau(G)$ be the number of spanning trees of G rooted at a fixed vertex v, and let $\rho(G)$ be the number of eulerian circuits in G. Then*

$$\rho(G) \;=\; \tau(G) \prod_{i=1}^{n} (d_i - 1)! \;.$$

Proof. Fix an eulerian circuit starting with the edge vu; without loss of generality we assume $v = v_1$ and $u = v_2$. After traversing the entire circuit

label the edges as follows. At vertex v_i label the d_i outedges in the reverse of the order they were traversed, except that at vertex v_1 do this for all edges other than v_1v_2, choosing labels from $\{2, \ldots, d_i\}$. Notice that (1) the arcs labelled '1' form a spanning tree T rooted at v_1 and (2) this labelling induces a permutation of the $d_i - 1$ outedges of v_i not in the tree. Whereas (2) is plain to see, (1) requires a short justification. Consider the two arcs labelled '1' that are incident with some fixed vertex w. Because the $d^-(w) = d^+(w)$, the eulerian circuit labelled the entering arc before the leaving arc. Thus any consistently oriented path of arcs labelled '1' are labelled in the order induced by the orientation. Thus T is acyclic and hence a spanning tree rooted at the only vertex with no leaving arc of label '1', namely v_1.

The proof is completed by recognizing that the choice of any spanning tree rooted at v, together with a "starter" edge vu (fixed throughout), and a permutation of the non-tree outedges at each vertex, determines an eulerian circuit simply by traversing arcs according to the maximum unused label available at each vertex. Clearly, this algorithm halts at a vertex if and only if all of its leaving edges have already been traversed, which can happen only at v. If the closed trail C produced isn't eulerian then some eulerian subgraph of G wasn't traversed by C. □

Theorem 8 is useful in the enumeration and generation of DeBruijn cycles. A k-ary DeBruijn cycle of order d is an infinite, periodic k-ary string of period k^d having the property that every k-ary d-tuple appears as a contiguous block exactly once in each period. Such cycles were thought to have been discovered

independently by DeBruijn [8] and Good [13], but actually were discovered earlier by Flye-Saint Marie [12]. The standard proof of the existence of DeBruijn cycles comes from their one-to-one correspondence with the set of all eulerian circuits of $B(d-1,k)$ which, knowing $\tau(B(d-1,k))$, implies that there are $(k!)^{k^{d-1}}/k^d$ DeBruijn cycles, a fact that is found in [12] without proof.

3.3 Planar graphs

Here we discuss one final problem, not of encodings of spanning trees but rather of bijections between them — in particular, between spanning trees of planar graphs and those of their planar duals. Let G be a planar graph and G_e one of its embeddings in the plane. Then define H_e to be its planar dual, having a vertex for each face of G and an edge between every pair of vertices coresponding to incident faces of G_e.

Theorem 9 *For any planar graph G, every planar embedding G_e and its corresponding planar dual H_e satisfy $\tau(G_e) = \tau(H_e)$.*

The proof given in [3] invokes the Matrix Tree Theorem, but a more appealing bijective method follows. The idea comes from imagining a drawing G_e on the surface of a sphere, using scissors to cut the surface of the sphere along the edges of one of its spanning trees T, and then laying the cut surface stretched flat on the plane to reveal its dual tree.

Proof. Consider one of the spanning trees T of G_e. Define the subgraph T' of H_e, by removing those edges of H_e which cross the edges of T. Because

of Euler's formula, the number of edges of T' is one less than the number of its vertices. Moreover, as is well known, minimal cutsets of G_e correspond to cycles in H_e, and so T' is acyclic. Hence T' is a spanning tree of H_e. The construction is clearly one-to-one, completing the proof. □

References

[1] T. van Aardenne-Ehrenfest, N.G. de Bruijn, Circuits and trees in ordered linear graphs, Simon Stevin 28 (1951) 203–217.

[2] K. Balińska, D. Cvetković, Z. Radosavljecić, S. Simić, D. Stevanović, A survey on integral graphs, Univ. Beograd, Publ. Elektrotehn. Fak., Ser. Mat. **13** (2002), 42–65.

[3] N. Biggs, Algebraic Graph Theory, Cambridge, London, 1974.

[4] F.C. Bussemaker, D. Cvetković, There are exactly 13 connected, cubic, integral graphs, Univ. Beograd, Publ. Elektrotehn. Fak., Ser. Mat. Fiz. **544–576** (1976), 43–48.

[5] A. Cayley, A theorem on trees, Quart. J. Math. **23** (1889) 376–378. Collected papers, Cambridge 13 (1897) 26–28.

[6] D.M. Cvetković, The spectral method for determining the number of trees, Publ. Inst. Math. (Beograd) **11** (**25**) (1971), 135–141.

[7] D.M. Cvetković, Cubic interval graphs, Univ. Beograd, Publ. Elektrotehn. Fak., Ser. Mat. Fiz. **498–541** (1975), 107–113.

[8] N.G. de Bruijn, A combinatorial problem, Proc. Nederl. Akad. Wetensch **49** (1946) 758–764.

[9] N. Deo, P. Micikevicius, A new encoding for labeled trees employing a stack and a queue, Bull. Inst. Combin. Appl. **34** (2002), 77–85.

[10] O. Eğecioğlu, J. Remmel, Bijections for Cayley trees, spanning trees, and their q-analogs, J. Combin. Theory Ser. A **42** (1986) 15–30.

[11] O. Eğecioğlu, J. Remmel, A Bijection for Spanning Trees of Complete Multipartite Graphs, Congr. Numer. **100** (1994), 225–243.

[12] C. Flye-Sainte Marie, Solution to problem number 58, l'Inter. Math. **1** (1894) 107–110.

[13] I. J. Good, Normally recurring decimals, J. London Math. Soc. **21** (1946), 167–169.

[14] G. Kirchhoff, Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme gefuhrt wird, Ann. Phys. Chem. **72** (1847) 497–508. Gesammelte Abhandlungen, Leipzig (1882) 22–33.

[15] L. Lovász, Combinatorial Problems and Exercises, Akadémiai Kiadó, 1979, North Holland, 1993.

[16] J.W. Moon, Various proofs of Cayley's formula for counting trees, in: F. Harary, L. Beineke (eds.), A Seminar on Graph Theory, Holt, Rinehart and Winston, New York, 1967, pp. 70–78.

[17] G. Oláh, A problem on the enumeration of certain trees (Russian), Stud. Sci. Math. Hungar. **3** (1968) 71–80.

[18] H. Prüfer, Neuer Beweis eines Satzes über Permutationen, Arch. Math. Phys. **27** (1918) 742–744.

[19] A.J. Schwenk, Exactly thirteen connected cubic graphs have integral spectra, Theory and Applications of Graphs (Proc. Internal. Conf. Western Michigan Univ., Kalamazoo, Mich. May 11-15, 1976, eds. Y. Alavi, D. Lick), Springer-Verlag, 1978, 516–533.

[20] C.A.B. Smith, W.T. Tutte, On universal paths in a network of degree 4, Amer. Math. Monthly **48** (1941) 233–237.

[21] H.N.V. Temperly, On the mutual cancellation of cluster integrals in Mayer's fugacity series, Proc. Phys. Soc. **83** (1964) 3–16.