ON SPANNING TREES OF CERTAIN GRAPHS

Glenn H. Hurlbert
Department of Mathematics
Arizona State University
Tempe, AZ 85287–1804

Abstract

Many classes of graphs have simple and elegant formulas for counting the number of spanning trees contained in them, but few have equally simple proofs. Here we present several such classes and offer two methods of attack.

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 subgraph of G having no cycles. That is, T is a connected graph with $V(T) = V(G)$, $E'(T) \subseteq E(G)$, and $|E'| = |V| - 1$. One natural and very old problem is to determine the number $t(G)$ of labelled spanning trees for a fixed graph G, or better yet, a formula for each in a family $\mathcal{G} = \{G_1, G_2, \dots\}$. (In [3] Biggs refers to $t(G)$ as the *complexity* of G.)

The first result in this direction was due to Cayley [6], who proved

Theorem 1.1 $t(K_n) = n^{n-2}$.

This result was originally stated in terms of counting the number of labelled trees on n vertices, his motivation coming from the enumeration of certain chemical isomers.

Our motivation for investigating spanning trees arises from counting the number of k-ary de Bruijn cycles of order n (to be discussed in section 3). This count is tied to the number of Eulerian circuits in a certain graph, which can be derived from its complexity. Another important application of spanning tree enumeration is in the computation of the total resistance along an edge in an electrical network (see [4]).

Kirchoff [12] was later able to find a general algebraic method to find $t(G)$, known as the *Matrix Tree Theorem*. Let $A = A(G)$ be the *adjacency*

Research supported in part by NSF grant DMS 9201467.

matrix of G; i.e., $a_{ij} = 1$ if vertex v_i is adjacent to vertex v_j and $a_{ij} = 0$ otherwise. Let D be the diagonal matrix of degrees, where d_{ii} is the degree of vertex v_i.

Theorem 1.2 *$t(G)$ is the value of any cofactor of the matrix $(D - A)$.*

This theorem is very handy for determining the complexity of a specific graph, but it can be difficult to use for an infinite family of graphs, especially if the graphs tend to get large very quickly. The following theorem is of a similar nature and is due to Temperly [18].

Theorem 1.3 *$t(G) = \det(J + D - A)/n^2$, where $|v(G)| = n$ and J is the $n \times n$ matrix of all 1's.*

The advantage of this theorem over the previous one is the corollary below (see [3]).

Corollary 1.4 *Let G be r-regular with n vertices and let $A(G)$ have eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n-1} < \lambda_n = r$. Then $t(G) = \frac{1}{n} \prod_{i=1}^{n-1} (r - \lambda_i)$.*

This is a particularly nice formulation if one knows the *spectra* (sets of eigenvalues) of a family of graphs, but they can be equally difficult to discover. On the other hand, if one could determine the complexity of a family of graphs, then one might also use that information to compute their spectra.

Still, none of these theorems tell us combinatorially *why* K_5 has 125 spanning trees, or *how* to list all 2,000 spanning trees of the Petersen graph. Prüfer [16], however, gave a proof of Cayley's theorem by presenting a bijection between the set of spanning trees and the set of sequences $(a_1, \ldots, a_{n-2})$ with each $a_i \in [n] = \{1, 2, \ldots, n\}$. His method answers the first but not the second question, and we will discuss it, along with a generalization due to Oláh [13], in section 2. Recently, Eğecioğlu and Remmel [7] found other bijections which may even be of greater use and which we will outline as well. Probably the most important use of such bijective proofs lies in the generation of random spanning trees (see [15]).

Before proceeding, however, we mention one appealing result involving planar graphs (one may consult [3] for many other theorems on complexity). If G is planar and H is its face dual, then $t(H) = t(G)$. The proof in [3] invokes Kirchhoff's Matrix Tree Theorem, but a more direct method is as follows. Draw G on the surface of a paper sphere and choose a spanning tree. Now take your scissors and cut along the edges of the tree. Lie the paper cutout flat on a table and notice that the faces have formed a tree, and that this construction is clearly 1–1.

2 Bijective Methods

We first present a generalization of theorem 1.1 whose proof is due to Oláh [14].

Theorem 2.1 *Let $a_1, \ldots, a_r$ be positive integers, $n = a_1 + \cdots + a_r$, and $V_i = \{v_{i,1}, \ldots, v_{i,a_i}\}$ for $1 \leq i \leq r$. Let $K(a_1, \ldots, a_r)$ be the complete r-partite graph having vertices $V = \cup_{i=1}^r V_i$ and edges $(v_{i,j}, v_{i',j'})$ if and only if $i \neq i'$. Then*

$$t(K(a_1, \ldots, a_r)) = n^{r-2} \prod_{i=1}^{r} (n - a_i)^{a_{i-1}}.$$

Note that if each $a_i = 1$, then this reduces to Cayley's theorem. In this case the method of Oláh also reduces to that of Prüfer.

Proof. We will later find it convenient to assume $a_1 \leq \cdots \leq a_r$. Let $G = K(a_1, \ldots, a_r)$ and label vertex $v_{i,j}$ by $x_{i,j} = \sum_{k=0}^{i-1} a_k + j$ $(a_0 = 0)$. Given an integer m, let $[m] = \{1, \ldots, m\}$ and for $s \geq 0$, let $[m]+s = \{1+s, \ldots, m+s\}$. For each $1 \leq i \leq r$, let $R_i = [a_i] + \sum_{k=0}^{i-1} a_k$ and $S_i = [n] \backslash R_i$. Notice that V_i has been labelled with integers from R_i, and V has been labelled in order by $[n]$.

We call $P = P_0 P_1 \cdots P_r$ a *generalized Prüfer sequence* if it is of the following type. For $1 \leq i \leq r$, $P_i = (p_{i,1}, \ldots, p_{i,a_i-1})$ is any sequence of $a_i - 1$ integers from S_i, and $P_0 = (p_{0,1}, \ldots, p_{0,r-2})$ is any sequence of $r-2$ integers from $[n]$. Clearly, there are $n^{r-2} \prod_{i=1}^r (n - a_i)^{a_i - 1}$ such generalized Prüfer sequences, our task being to set up a 1–1 correspondence between these and the set of all spanning trees of G.

Given a spanning tree T we exhibit its sequence as follows. Find the leaf vertex of $T = T_0$ with the smallest label, say vertex $v_{i,j}$, and record the label $x_{i',j'}$ of its unique neighbor in T_0, $v_{i',j'}$, in the first available position of P_i. Remove vertex $v_{i,j}$ and edge $(v_{i,j}, v_{i',j'})$ from T_0 to form T_1 and iterate this procedure with the following proviso. If $v_{i,j}$ is the last remaining vertex of V_i in T, the label of its neighbor will be recorded in the first available position of P_0, rather than P_i. When only one edge remains (in T_{n-2}) the algorithm halts, and it is easily checked that we have produced a generalized Prüfer sequence. For example, the tree in figure 1 yields the sequence $P = (5,7)(3)(8,8)(5,10)(8,1,1)$.

Figure 1.

On the other hand, let P be a generalized Prüfer sequence, and let us reconstruct the spanning tree. Let L be the list of all labels $[n]$. We first find the smallest label $x_{i,j}$ in $L = L^{(0)}$ not appearing in $P = P^{(0)}$ and let $x_{i',j'}$ be the first available label from P_i. We then draw an edge between $v_{i,j}$ and $v_{i',j'}$, remove $x_{i,j}$ from $L^{(0)}$ to form $L^{(1)}$, remove $x_{i',j'}$ from P_i (and $P^{(0)}$) to form $P^{(1)}$, and iterate the procedure. Here, our proviso is that if $x_{i,j}$ is the last remaining label in R_i (i.e., P_i is now empty) then we let $x_{i',j'}$ be the first available label from P_0. After $n-2$ steps, $P^{(n-2)}$ is empty and two labels $x_{i,j}$ and $x_{i',j'}$ remain in $L^{(n-2)}$. We then include the final edge $(v_{i,j}, v_{i',j'})$. The proof that this forms a spanning tree is identical to Prüfer's argument and is omitted (since we have the right number of edges, we need only check that there are no cycles). $\square$

It is interesting to note that if $G = K_n$ then the sequence from Prüfer's original proof is simply P_0. One can also notice that if vertex $v_{i,j}$ has degree $d_{i,j}$ in tree T, then the label $x_{i,j}$ appears $d_{i,j} - 1$ times in P and so $\sum_i \sum_j (d_{i,j} - 1) = n - 2$, or $\sum_i \sum_j d_{i,j} = 2(n-1)$. In other words, the sum of the degrees is twice the number of edges. But unlike the case $G = K_n$, this condition is not sufficient for the existence of a spanning tree of $K(a_1, \ldots, a_r)$ with such degrees $d_{i,j}$.

Given two sequences $B = (b_1, \ldots, b_m)$ and $C = (c_1, \ldots, c_m)$ with $b_1 \geq \cdots \geq b_m$ and $c_1 \geq \cdots \geq c_m$, we say $B \leq C$ whenever $b_i \leq c_i$ for $1 \leq i \leq m$. Given $a_1 \leq \cdots \leq a_r$, let $\Delta_i = (\delta_{i,1}, \ldots, \delta_{i,a_i})$, where $\delta_{i,j} = n - a_i$ for $1 \leq j \leq a_i$, and $\Delta = \Delta_1 \cdots \Delta_r$. Let D be the degree sequence of a fixed spanning tree of $K(a_1, \ldots, a_r)$, taken in nonincreasing order. Then, clearly $D \leq \Delta$. It is a simple exercise to show that this second condition combined with the previous condition is sufficient for the existence of a spanning tree with the prescribed degrees. Unfortunately, it is not easy to count the number of such trees for a given degree sequence. In the case $G = K_n$, the number is $\binom{n-2}{d_1-1,\ldots,d_n-1}$.

Our second bijective method is due to Eğecioğlu and Remmel [7] and works for counting spanning trees of complete r-partite graphs with $r \leq 3$. They have recently extended their method to work for all r [8]. We illustrate their method with $K(12, 14)$ and show the bijection between all its spanning trees and all functions

$$f : \begin{cases} \{2, 3, \ldots, 12\} \to \{13, 14, \ldots, 26\} \\ \{13, 14, \ldots, 25\} \to \{1, 2, \ldots, 12\}, \end{cases}$$

of which there are $12^{13} \cdot 14^{11}$.

Before proceeding, we note that the number of spanning trees in a graph is equal to the number of spanning trees *rooted* at a specified vertex r. A *rooted* spanning tree (T, v) is a spanning tree in which each edge (x, y) is given an orientation pointing toward v. That is, if $xy \cdots v$ is the unique path from x to v then we use the orientation xy for edge (x, y). That complexity is independent of the choice of a root v is seen by simply rooting each spanning tree at v. In this method we will speak only of trees rooted at vertex v_n (for a graph on n vertices).

Given the function

i	2	3	4	5	6	7	8	9	10	11	12
f	25	18	26	19	24	15	26	26	24	14	17

i	13	14	15	16	17	18	19	20	21	22	23	24	25
f	5	3	11	12	8	5	12	5	2	5	1	3	12

let us construct the corresponding rooted tree. Draw the directed graph with arcs $v_i v_{f(i)}$ and make two observations. First, there is no arc leaving v_1 and so the component containing it is a tree rooted at v_1. Likewise with v_{26}. Second, each other component consists of a unique cycle with, possibly, rooted trees hanging off of it (see figure 2).

Figure 2.

Now order the components as follows. In each cycle pick the smallest label to represent the entire component. Pick the roots to represent the tree components. The components are then ordered left to right according to the smaller representative. Each component itself is then drawn so that the vertices of the cycle appear on the main line (see figure 3) with its representative rightmost and edges directed as shown. The process of including the dotted arcs and removing the slashed is then visually quite natural and constructs the desired rooted tree.

Figure 3.

The inverse process is straightforward. Given the above tree we can recover the function by considering the unique path from v_1 to v_{26}. Ignoring v_1 and v_{26} we direct an arc from the smallest label v_3 to the "beginning" v_{15}, from the next smallest v_5 to the "new beginning" v_{18}, and so on, including these arcs and excluding those which connect one cycle to another (or to v_1 or v_{26}). That such a "bipartite" function is recovered is not difficult to reason.

In [7] bijections are constructed for spanning trees at other vertices v_i and for complete r-partite graphs with $r \leq 3$. Only recently [8] has this method been pushed to handle $r > 3$. In addition, Remmel and Eğecioğlu give a full description of the q-analog properties of their bijections.

3 Other Graphs

Define the *k-ary de Bruijn graph of order n*, $dB(n,k)$, as follows. Its vertices are all n-tuples $(b_1, \ldots, b_n)$ with each $b_i \in [k]$, and it has directed arcs from $(b_1, b_2, \ldots, b_n)$ to $(b_2, \ldots, b_n, b_{n+1})$. Notice that this is a connected digraph with indegree = outdegree = k, and so is *Eulerian*. That is, we can traverse each edge in the graph exactly once while returning to the same vertex we started from. Since each arc in $dB(n, k)$ corresponds to a k-ary $(n+1)$-tuple $(b_1, \ldots, b_{n+1})$, this gives us a constructive existence proof for a *k-ary de Bruijn cycle of order* $(n+1)$, which is defined to be a cyclic listing of k^{n+1} digits with the property that every k-ary $(n+1)$-tuple appears exactly once consecutively on the list. These objects have been rediscovered several times (see [5, 9, 10]) and have very many interesting applications and generalizations (see [11]).

Since there is a one-to-one correspondence between the de Bruijn cycles of order $(n+1)$ and the Eulerian circuits in $dB(n, k)$, the problem of counting such cycles is reduced to enumerating the Eulerian circuits of $dB(n, k)$. The following theorem is implicit in the work of many authors.

Theorem 3.1 *Let G be an Eulerian di-graph with out-degree sequence $d_1, \ldots, d_n$. Let $t(G)$ be the number of spanning trees of G rooted at some vertex v_i, and let $eul(G)$ be the number of Eulerian circuits in G. Then*

$$eul(G) = t(G) \prod_{i=1}^{n} (d_i - 1)!$$

Proof. Fix an Eulerian circuit starting with the arc $v_i v_j$. After traversing the entire circuit label the edges as follows. At vertex v_x label the d_x out-edges in the reverse of the order they were traversed, except that at vertex v_i do this for all arcs other than $v_i v_j$, choosing labels from $\{2, \ldots, d_i\}$. Notice that 1) the arcs labelled "1" form a spanning tree rooted at v_i, and 2) this labelling induces a permutation of the $d_x - 1$ out-edges at v_x not in the tree. The proof is complete by recognizing that the choice of any spanning tree rooted at v_i, together with a "starter" edge $v_i v_j$ (fixed throughout) and a permutation of the non-tree out-edges at each vertex, determines an Eulerian circuit simply by traversing arcs according to the maximum label available at each vertex. $\square$

The generalization of the Matrix Tree Theorem to directed graphs, often called the BEST Theorem (see [2, 17]), leads to the following results.

Theorem 3.2 $t(dB(n, k)) = k^{k^n - 1 - n}$.

Corollary 3.3 *The number of k-ary de Bruijn cycles of order n is $(k!)^{k^{n-1}}/k^n$.*

The nice thing about theorem 3.2 is that $k^{k^n-1-n} = k^{-n} k^{k^n-1}$. That is, it is of the same form as theorem 1.1, corollary 1.4, and theorem 2.1, namely $1/|V(G)|$ times the product of $|V(G)| - 1$ factors. Aesthetically, this makes sense—one factor for each edge divided by the number of choices for a root. And in this case each factor equals the out-degree k in the graph. So the natural method to attempt is that of Eğecioğlu and Remmel, namely to fix a root and a function that picks an out-edge for each non-root vertex. Unfortunately, there is much more to "repair" here than previously.

Another popular family of graphs consists of the *n-dimensional cubes* Q^n, which have binary n-tuples as vertices and undirected edges between vertices of hamming distance one (the toggle of one bit).

Define the *spectrum* of a graph G,

$$\operatorname{Spec}(G) = \begin{pmatrix} \lambda_1 & \cdots & \lambda_s \\ m_1 & \cdots & m_s \end{pmatrix},$$

according to the adjacency matrix $A(G)$ having eigenvalues $\lambda_1, \dots, \lambda_s$ of multiplicities $m_1, \dots, m_s$, respectively. In [3] we find that

$$\operatorname{Spec}(Q^n) = \begin{pmatrix} n & n-2 & n-4 & \cdots & n-2j & \cdots & -n \\ 1 & n & \binom{n}{2} & \cdots & \binom{n}{j} & \cdots & 1 \end{pmatrix}$$

so that $t(Q^n) = 2^{-n} \prod_{j>0} (2j)^{\binom{n}{j}}$.

In this case we might try the following. Draw Q^n by levels. That is, level j consists of all vertices of *weight* j, where the weight of a vertex is the number of 1's in its string. One can then see that a vertex at level j is adjacent to j vertices at level $j-1$, and so a possible function might be to specify for each root (weight 0) vertex a specific edge to the previous level, and directing it either upward or downward. The function seems natural enough, but the resulting structure seems extremely difficult to repair in the sense of making it a spanning tree rooted at $(0, 0, \dots, 0)$. Although there are no cycles, the arc orientations are hard to interpret.

The *Odd Graph* $\mathcal{O}_k$ has all subsets of $[2k-1]$ of size $(k-1)$ for vertices and has edges between disjoint subsets. See [3] for $\operatorname{Spec}(\mathcal{O}_k)$, but the first tempting case is the *Petersen Graph* $\mathcal{P} = \mathcal{O}_3$.

$$\operatorname{Spec}(\mathcal{P}) = \begin{pmatrix} 3 & -2 & 1 \\ 1 & 4 & 5 \end{pmatrix}$$

and so $t(\mathcal{P}) = (1/10)5^4 2^5 = 2{,}000$ (this is 8 times $t(K_5)$). Here may be a fine place to blend both Prüfer and E/R methods.

As one last challenge we wish to reintroduce the reader to *metacirculant graphs* (see [1] for the actual definitions). Whereas the direct product of two cycles is the graph of the group $\mathbf{Z}_m \times \mathbf{Z}_n = \langle p \rangle \times \langle \sigma \rangle$ with p and σ satisfying

$|p| = m$, $|\sigma| = n$, and $p\sigma p^{-1} = \sigma$, a metacirculant graph $G(m, n, \alpha)$ only differs in that $p\sigma p^{-1} = \sigma^\alpha$ for some α. For example, we have the Petersen graph $\mathcal{P} \cong G(2, 5, 2)$. With these graphs, a Prüfer-type code may be the only way to discover their complexity. We end with the following

Conjecture 3.4 *$t(G(m, n, \alpha))$ depends only on m and n and not on α.*

4 Acknowledgments

Many thanks to Dan Archdeacon for uncovering [14] and to Hal Kierstead for showing me [7].

References

[1] Alspach, B. & Parsons, T.D., A construction for vertex-transitive graphs, *Canad. J. Math.* **34** (1982) 307–318.

[2] van Ardenne-Ehrenfest, T. & de Bruijn, N.G., Circuits and trees in ordered linear graphs, *Simon Steven* **28** (1951) 203–217.

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

[4] Bollobás, B., *Graph Theory, An Introductory Course*, Springer-Verlag, New York, 1979.

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

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

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

[8] Eğecioğlu, O. & Remmel, J., preprint.

[9] Flye-Sainte Marie, C., Solution to problem number 58, *l'Intermediaire des Mathematiciens* **1** (1894) 107–110.

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

[11] Hurlbert, G.H., Universal cycles–on beyond de Bruijn, Ph.D. Thesis (1990).

[12] KIRCHHOFF, G., 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.

[13] MOON, J.W., Various proofs of Cayley's formula for counting trees, *A Seminar on Graph Theory* (F. Harary and L. Beineke, eds.), Holt, Rinehart and Winston, New York, 1967.

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

[15] PALMER, E., *Graphical Evolution*, John Wiley & Sons, New York, 1985.

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

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

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