Generating the List of Spanning Trees in $\mathrm{K}_{\mathrm{s,t}}$

Thomas Porter
Department of Mathematics,
Southern Illinois University,
Carbondale. IL 62901-4408
tporter@math.siu.edu

Abstract

The formula for the number of spanning trees in $K_{t_1,\ldots,t_p}$ is well known. In this paper we give an algorithm that generates the list of spanning trees in $K_{s,t}$.

1 Introduction

We use the standard notation and terminology which can be found, e.g., in [9]. Let $\tau(G)$ denote the number of spanning trees in a connected graph G. The graph $K_{t_1,\ldots,t_p}$ is the complete multipartite graph of order n, i.e., $t_1 + \cdots + t_p = n$. A formula for $\tau(K_{t_1,\cdots,t_p})$ has been found by Austin [1], Good [5], Eğecioğlu and Remmel [4], Lewis [7], and Clark [3].

In this paper we are interested in finding an algorithm that generates the *list* of spanning trees in $K_{s,t}$, rather than enumerating this list. The motivation was to find the number of nonisomorphic spanning trees in $K_{s,t}$.

Theorem 1.1 ([1], [3], [4], [5], [7]). *If $t_1 + \cdots + t_p = n$, where $t_i \in \mathbb{Z}^+$, then*

$$\tau(K_{t_1,\cdots,t_p}) = n^{p-2}(n-t_1)^{t_1-1}\cdots(n-t_p)^{t_p-1} = n^{p-2}\prod_{i=1}^{p}(n-t_i)^{t_i-1}.$$

Notice $K_n \cong K_{\underbrace{1,\ldots,1}_{n\text{-times}}}$. As immediate consequences of Theorem 1.1 we have:

Corollary 1.2.

$$\tau(K_n) = \tau(K_{1,\ldots,1}) = n^{n-2}(n-1)^{(1-1)} \cdots (n-1)^{(1-1)} = n^{n-2}.$$

Corollary 1.3. $\tau(K_{s,t}) = s^{t-1}t^{s-1}$.

As another illustration of the formula consider the hyperoctahedral graph $H_n = K_{\underbrace{2,\ldots,2}_{n\text{-times}}}$. We have $\tau(H_n) = 2^{2n-2}n^{n-2}(n-1)^n$. For $H_3 = K_{2,2,2} =$ octahedron, we have $\tau(H_3) = 384$; also since the dual of the octahedron is the 3-cube Q_3, we have $\tau(Q_3) = 384$.

We show another way to compute $\tau(K_{t_1,\ldots,t_p})$ by computing the eigenvalues of the Laplacian matrix of $K_{t_1,\ldots,t_p}$ The *Laplacian matrix*, $L(G)$, of a graph G of order n is defined to be $L(G) = D(G) - A(G)$, where $D(G)$ is the diagonal matrix of vertex degrees in G, and $A(G)$ is the adjacency matrix. The *Laplacian spectrum* of G is $s(G) = (\lambda_1, \ldots, \lambda_n)$, where the eigenvalues of $L(G)$ are ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0$. The following facts about the Laplacian matrix can be found in [2], [8]. It is known that $\lambda_n = 0$ with corresponding eigenvector $[1, 1, ..., 1]$ and if G is connected, then $\lambda_{n-1} > 0$. The Laplacian spectrum is graph invariant; that is, $G_1 \cong G_2$ only if $s(G_1) = s(G_2)$. Also, there is a relation between the Laplacian spectrum of a graph G with its spanning tree number $\tau(G)$.

Theorem 1.4 ([2], [8]).

$$\tau(G) = \frac{\lambda_1 \cdot \lambda_2 \cdots \lambda_{n-1}}{n}. \qquad \square$$

If one has the eigenvalues of the Laplacian matrix of G, they can be used to compute the number of spanning trees in G. The *join*, denoted by $G \vee H$, of two graphs G and H is defined to be the graph which is the union of G and H together with the additional $|V(G)||V(H)|$ edges consisting of all possible edges xy where x is a vertex in G and y is a vertex in H. Merris [8] gives a way to compute the spectrum of $G \vee H$, which we state here:

Theorem 1.5 ([8, pg. 181]). *Let G and H be graphs on disjoint sets of s and t vertices respectively. If $s(G) = (\nu_1, \ldots, \nu_{s-1}, 0)$ and $s(H) = (\mu_1, \ldots, \mu_{t-1}, 0)$, then $s(G \vee H)$ consists of the numbers*

$$t + s$$

$$s + \mu_1, s + \mu_2, \ldots, s + \mu_{t-1}$$

$$t + \nu_1, t + \nu_2, \ldots, t + \nu_{s-1}, \text{ and } 0, \text{ arranged in nonincreasing order.}$$

Let $\overline{G}$ denote the complement of G. Notice that $K_{s,t} \cong \overline{K}_s \vee \overline{K}_t$, and that $K_{t_1,\ldots,t_p} \cong \overline{K}_{t_1} \vee K_{t_2,\ldots,t_p}$.

Theorem 1.6. *With $t_1 + t_2 + \cdots + t_p = n$, the Laplacian spectrum of $K_{t_1,\ldots,t_p}$ consists of the eigenvalues:*

$$\underbrace{(t_1 + \cdots + t_p), (t_1 + \cdots + t_p), \ldots, (t_1 + \cdots + t_p)}_{(p-1)\text{-times}};$$

and

$$\underbrace{(n - t_i), (n - t_i), \ldots, (n - t_i),}_{(t_i - 1)\text{-times}} \quad \text{for all } i = 1, \ldots, p.$$

Proof. The proof is by induction on p. For the ground case $p = 2$, we have $K_{s,t} \cong \overline{K}_s \vee \overline{K}_t$. Also, $s(\overline{K}_s) = (0, 0, \ldots, 0)$ and $s(\overline{K}_t) = (0, 0, \ldots, 0)$. Hence by Theorem 1.5, the eigenvalues of the Laplacian matrix of $K_{s,t}$ consists of the numbers $(s + t)$;

$$\underbrace{s, s, \ldots, s}_{(t-1)\text{-times}}; \qquad \underbrace{t, t, \ldots, t}_{(s-1)\text{-times}}.$$

Now, assume the inductive hypothesis for all $K_{a_1,\ldots,a_k}$ where $k < p$. Consider $K_{t_1,\ldots,t_p}$; we have $K_{t_1,\ldots,t_p} \cong \overline{K}_{t_p} \vee K_{t_1,\ldots,t_{p-1}}$. By the inductive hypothesis, the Laplacian eigenvalues of $K_{t_1,\dots,t_{p-1}}$ are

$$\underbrace{(t_1+\cdots+t_{p-1}),(t_1+\cdots+t_{p-1}),\dots,(t_1+\cdots+t_{p-1})}_{(p-2)\text{-times}};$$

and with $q = t_1+\cdots+t_{p-1}$,

$$\underbrace{(q-t_i),(q-t_i),\dots,(q-t_i),}_{(t_i-1)\text{-times}} \quad \text{for all } i=1,\dots,p-1.$$

Now with $s(\overline{K}_{t_p}) = (0,0,\dots,0)$, we apply Theorem 1.5 to $\overline{K}_{t_p} \vee K_{t_1,\dots,t_p} \cong K_{t_1,\dots,t_p}$, yielding that the Laplacian eigenvalues of the $K_{t_1,\dots,t_p}$ are:

$$\underbrace{(t_1+\cdots+t_p),(t_1+\cdots+t_p),\dots,(t_1+\cdots+t_p)}_{(p-1)\text{-times}};$$

and

$$\underbrace{(n-t_i),(n-t_i),\dots,(n-t_i),}_{(t_i-1)\text{-times}} \quad \text{for all } i=1,\dots,p.$$

□

Theorem 1.7. *If $t_1+\cdots+t_p = n$, where $t_i \in \mathbb{Z}^+$, then*

$$\tau(K_{t_1,\dots,t_p}) = n^{p-2}\prod_{i=1}^{p}(n-t_i)^{t_i-1}.$$

Proof. This follows immediately by combining Theorems 1.4 and 1.6. □

The reader familiar with the Laplacian spectrum will notice there is a faster, more direct way to compute the Laplacian eigenvalues of $K_{t_1,\dots,t_p}$. That is, to first compute the Laplacian spectrum of $K_{t_1}+K_{t_2}+\cdots+K_{t_p}$; and then use the known relations between $s(G)$ and $s(\overline{G})$ for any graph G and its complement $\overline{G}$. Here, notice $\overline{K_{t_1,\dots,t_p}} \cong K_{t_1,\dots,t_p}$. But we choose to establish Theorem 1.6 by using the join operator.

2 The Algorithm

We now discuss the main objective of the paper, to algorithmically list the set of spanning trees in $K_{s,t}$. We use $[n]$ to denote the set $\{1, \ldots, n\}$ and $s \geq 2$. A spanning tree in $K_{s,t}$ is formed by merging two objects; a rooted directed tree of order s, and a partition $(A_1, \ldots, A_s)$ of $\{y_1, y_2, \ldots, y_t\}$ where $|A_1| \geq 1$, and $|A_j| \geq 0$ for $2 \leq j \leq s$. In the graph $K_{s,t}$ we label the s vertices in one partite set with the integers in $[s]$, and for the t vertices in the other partite set we label with $\{y_1, \ldots, y_t\}$. We use the notion that for the two partite sets s and t, the *vertices in the partite set of size s are colored red*, and the *vertices in the partite set of size t are colored blue.* To each of the s^{s-2} labeled trees on s-vertices we form a rooted oriented tree as follows:

Given a fixed tree T of order s, with $V(T) = [s]$:

2(a) Draw the tree as a rooted tree with the vertex labeled 1 as the designated root.

2(b) An edge (i, j) of T is given the orientation $(\overrightarrow{i, j})$, i.e., $j \in N^+(i)$, if $d(1, j) < d(1, i)$.

In other words, with the designated root labeled 1, we orient each edge of T *towards* the root.

We use $\overrightarrow{T}$ to denote the oriented tree formed from T using the above definition. Notice that for each red vertex $j \in \{2, \ldots, s\}$ in $\overrightarrow{T}$, $|N^+_{\overrightarrow{T}}(j)| = 1$, otherwise we would have a cycle in T. We also interchangeably use the term *di-tree* in place of oriented tree.

Now, to each of the s^{s-2} di-trees $\overrightarrow{T}$, we pair them with partitions $(A_1, ..., A_s)$ of $\{y_1, y_2, \ldots, y_t\}$ that satisfy the following:

For a fixed di-tree $\overrightarrow{T}$, it is paired with *each* partition $(A_1, ..., A_s)$ where;

2(i) For all vertices j in $\overrightarrow{T}$, where $|N^-(j)| > 0$, it is required that $A_j \neq \emptyset$.

2(ii) $A_1 \neq \emptyset$.

We remark that 2(ii) follows from 2(i) and 2(b) above since $d(1,1) < d(1,j)$ for any $j \neq 1$.

For a fixed di-tree $\overrightarrow{T}$ and an associated partition $(A_1,\ldots,A_s)$ of $\{y_1,\ldots,y_t\}$, that satisfies 2(i) and 2(ii) above; we call such a pair a *legal pair*, and denote them as $(\overrightarrow{T},(A_1,\ldots,A_s))$. The pair $(\overrightarrow{T},(A_1,\ldots,A_s))$ generates the following spanning trees in $K_{s,t}$.

Given the partition $(A_1,\ldots,A_s)$, for each A_j, where $|A_j| > 0$, we join the vertex j in the red partite set to *all* of the blue vertices in A_j. Notice at this stage we have a subgraph of $K_{s,t}$ formed having t edges. Now to form a spanning tree we add to this subgraph the following $s-1$ edges. For each red vertex i in $\overrightarrow{T}$, where $2 \leq i \leq s$; let j denote the out-neighbor of i in $\overrightarrow{T}$, i.e., $(\overrightarrow{i,j})$ is an arc in $\overrightarrow{T}$, we then add an edge of the form (i, y_x) where y_x is *any vertex* in A_j. Note from 2(i) above, that $A_j \neq \emptyset$.

We now show this is indeed a spanning tree. We remark that when we add the edge (i, y_x) as mentioned above, there are $|A_j|$ choices for this y_x. Moreover, given the pair $(\overrightarrow{T},(A_1,\ldots,A_s))$, for each $j \in [s]$, let $k_j = |A_j|$ and let $\{x_1, x_2, \ldots, x_r\}$ be precisely the subset of red vertices of $\overrightarrow{T}$ in $\{2,\ldots,s\}$ where $|N^-(x_i)| \neq 0$. Then the number of spanning trees generated by this *fixed legal pair* $(\overrightarrow{T},(A_1,\ldots,A_s))$ is $k_{x_1}^{|N^-(x_1)|} k_{x_2}^{|N^-(x_2)|} \cdots k_{x_r}^{|N^-(x_r)|}$.

Theorem 2.1. *With the above notation, the number of spanning trees generated by a single legal pair $(\overrightarrow{T},(A_1,\ldots,A_s))$ is*

$$k_{x_1}^{|N^-(x_1)|} k_{x_2}^{|N^-(x_2)|} \cdots k_{x_r}^{|N^-(x_r)|}.$$

Proof. A term in the above product, say $k_{x_i}^{|N^-(x_i)|}$, comes from the algorithm as follows. When we add the last $s-1$ edges, for each of the $|N^-(x_i)|$ red vertices in $N^-(x_i)$, there are $|A_{x_i}| = k_{x_i}$ blue vertices to choose from in A_{x_i}in which to form an edge. □

This observation will lead to some enumeration that we do later. Now to show our algorithm generates a spanning tree, we first note that by construction it produces a spanning subgraph of $K_{s,t}$ with $s+t-1$ edges. Since any connected graph of order n and size $n-1$ is a tree, we need to show our subgraph is connected. Also, since any graph that contains a $u-v$ walk also contains a $u-v$ path, we show the existence of walks between any two vertices. There are three cases to check. If $W = x_1, x_2, \ldots, x_p$ is a walk then we use W^{-1}, to denote the walk $W^{-1} = x_p, x_{p-1}, \ldots, x_1$.

Given a legal pair $(\overrightarrow{T}, (A_1, \ldots, A_s))$, and a subgraph H in $K_{s,t}$ generated by the algorithm, we have:

Lemma 2.2. *There exists a walk between any two red vertices in H.*

Proof. Consider two red vertices i, j, where i and j are in $\{1, ..., s\}$. We show there are two walks W_i and W_j, here W_i is a walk from i to 1, and W_j is a walk from j to 1. Hence, the walk $W_i W_j^{-1}$ is a walk from i to j.

Let $\overrightarrow{P}_i = i, x_1, x_2, \ldots, x_n, 1$ denote the oriented path from i to 1 in $\overrightarrow{T}$. By the definition of the algorithm, since $(\overrightarrow{i, x_1})$ is an arc in $\overrightarrow{T}$, there is an edge in H of the form iu_1, where $u_1 \in A_{x_1}$. Also by the algorithm, x_1u_1 is one of the initial t edges of the construction. Hence, so far we have the path i, u_1, x_1. We repeat this with the arc $(\overrightarrow{x_1, x_2})$ in $\overrightarrow{P}$. Again, by the definition of the algorithm, we have that there exists an edge in H of the form x_1u_2, where $u_2 \in A_{x_2}$, also x_2u_2 is one of the initial t edges of the construction. At this point we have the walk i, u_1, x_1, u_2, x_2 in H. We continue this process and form the walk $i, u_1, x_1, u_2, x_2, u_3, \ldots, u_n, x_n$ in H. Now, since $(\overrightarrow{x_n, 1})$ is an arc in $\overrightarrow{T}$, we have by the algorithm that there is an edge in H of the form $x_n a_1$, where $a_1 \in A_1$; also $1a_1$ is one of the initial t edges of the construction. Hence $W_i = i, u_1, x_1, u_2, x_2, \ldots, x_n, a_1, 1$ is an $i-1$, i.e., (i to 1)-walk in H. We can apply the same idea starting with the oriented path $\overrightarrow{P}_j$ from j to 1 in $\overrightarrow{T}$ and obtain a $j-1$ (j to 1)-path W_j in

H. Consequently $W_i W_j^{-1}$ is an $i-j$ path in H. □

Lemma 2.3. *There exists a walk between any two blue vertices.*

Proof. Consider two blue vertices y_m, y_n ,where y_m and y_n are in $\{y_1, ..., y_t\}$. From our partition $(A_1, \ldots, A_s)$ if y_m and y_n are in the same partition set, say, A_x, then for red vertex x in $[s]$, $y_m x$ and $y_n x$ are amongst the initial t edges of our construction, hence y_m, x, y_n is a $y_m - y_n$ path and we are done. Otherwise, they are from different partition sets, say y_m is in A_ℓ and y_n is in A_k. Then, from our construction, $y_m \ell$ and $y_n k$ are amongst the initial t edges; also by Lemma 2.2, since ℓ and k are red vertices there exists an $\ell - k$ walk, which we denote by W, henceforth; y_m, ℓ, W, y_n is a walk from y_m to y_n. □

Lemma 2.4. *There exists a walk between any pair of red and blue vertices.*

Proof. Consider a blue vertex y and a red vertex j. From our initial partition $(A_1, \ldots, A_s)$, if y is in A_j, then by definition of our construction, the edge jy is one of the initial t edges and we are done. Otherwise, y is in some other partite set say, A_x. Now by definition, xy is one of the initial t edges in our construction, and from Lemma 2.2 there exists a walk from x to j, denoted by W; henceforth y, x, W is a walk from y to j. □

Combining the above cases yields:

Theorem 2.5. *Any subgraph of $K_{s,t}$ that is generated by the algorithm using a legal pair $(\overrightarrow{T}, (A_1, \ldots, A_s))$ is a spanning tree.*

Proof. Let H be a subgraph of $K_{s,t}$ generated by the algorithm. We have by construction, H is a spanning subgraph with $s+t-1$ edges. Combining Lemmas 2.2, 2.3, 2.4, we have shown H is connected, consequently H is a spanning tree. □

We now want to show that any spanning tree in $K_{s,t}$ is generated uniquely by the algorithm, i.e., from any spanning tree, we will extrapolate a *unique* legal pair from whence it was generated. Let Q be a spanning tree, we will remove $s-1$ edges from Q to reveal the unique legal pair $(\overrightarrow{T}, (A_1, \ldots, A_s))$ that generated it. Consider our special red vertex 1. From amongst the other $s-1$ red vertices, namely $2, 3, \ldots, s$, we form a subset of these which we call *vertices at level* 1, denoted V_1 where $V_1 = \{j | N_Q(j) \cap N_Q(1) \neq \emptyset\}$. We now begin to build our rooted tree T with designated root 1, by first adjoining to 1 all vertices in V_1. Note that $V_1 \neq \emptyset$, otherwise Q would be disconnected.

Let $V_1 = \{v_{11}, v_{12}, \ldots, v_{1p_1}\}$. We then define *vertices at level* 2, V_2 as follows:

$$V_2 = \{j \mid N_Q(j) \cap N_Q(v_{1i}) \neq \emptyset \text{ for some vertex } v_{1i} \in V_1\}.$$

We denote V_2 by $V_2 = \{v_{21}, v_{22}, \ldots, v_{2p_2}\}$. We continue in this way and define vertices at level k, V_k as:

$$V_k = \{j \mid N_Q(j) \cap N_Q(v_{(k-1)_i}) \neq \emptyset \text{ for some vertex } v_{(k-1)_i} \in V_{k-1}\}.$$

We denote V_k by $V_k = \{v_{k1}, v_{k2}, \ldots, v_{kp_k}\}$. We also define $V_0 = \{1\}$.

For a given graph $G = (V, E)$, and a vertex subset $S \subset V$ with $S = \{u_1, u_2, \ldots, u_\ell\}$ we define $N_G(S)$ as $N_G(S) = \bigcup_{i=1}^{\ell} N_G(u_i)$.

Lemma 2.6. *For any vertex j in V_k, with $k \geq 1$, $|N_Q(j) \cap N_Q(V_{k-1})| = 1$.*

Proof. Suppose there exists a vertex $j \in V_k$, where $|N_Q(j) \cap N_Q(V_{k-1})| \geq 2$.

Case 1. Suppose there is a single vertex $u \in V_{k-1}$ with $|N_Q(j) \cap N_Q(u)| \geq 2$. Let x, y denote two blue vertices in the intersection set. We then have u, x, j, y, u is a 4-cycle in Q, contradicting that Q is a tree.

Case 2. Suppose there exist two vertices $a, b \in V_{k-1}$ with $|N_Q(a) \cap N_Q(j)| \geq 2$ and $|N_Q(b) \cap N_Q(j)| \geq 2$.

Case 2(i). Suppose $N_Q(a) \cap N_Q(b) \cap N_Q(j) \neq \emptyset$. Let u be a common vertex in these intersecting sets. Let A denote the path from a–to–1 in Q, and let B denote the path from 1–to–b in Q. We then have A, B, u, a contains a cycle contradicting that Q is a tree.

Case 2(ii). Let $x \in N_Q(a) \cap N_Q(j)$ and $y \in N_Q(b) \cap N_Q(j)$. Let A denote the path from a–to–1 in Q and B the path from 1–to–b in Q. We then have x, a, A, B, y, j, x is a cycle in Q contradicting that Q is a tree. □

Lemma 2.7. *For any vertex $j \in V_k$, with $k \geq 1$, $N_Q(j) \cap N_Q(v_{(k-1)i}) \neq \emptyset$ for exactly one vertex $v_{(k-1)i}$ in V_{k-1}.*

Proof. We have by Lemma 2.6 that $|N_Q(j) \cap N_Q(V_{k-1})| = 1$. Let $x \in N_Q(j) \cap N_Q(a) \cap N_Q(b)$ for two distinct vertices $a, b \in V_{k-1}$. Let A denote the a–to–1 path in Q and B the 1–to-b path in Q. We then have x, a, A, B, x contains a cycle contradicting that Q is a tree. □

We continue this process of building the sequence of vertex levels $V_1, V_2, \ldots, V_m$ until $\{1\} \cup V_1 \cup \cdots V_m = \{1, 2, \ldots, s\}$. Let $V_1 = \{v_{11}, v_{12}, \ldots, v_{1p_1}\}$, by Lemma 2.6 to each v_{1i} in V_1 $|N_Q(v_{1i}) \cap N_Q(1)| = 1$. Let t_{1i} denote this single blue vertex, i.e., $\{t_{1i}\} = N_Q(v_{1i}) \cap N_Q(1)$. More generally, with $V_k = \{v_{k1}, \ldots, v_{kp_k}\}$ let blue vertex t_{ki} be defined by $\{t_{ki}\} = N_Q(v_{ki}) \cap N_Q(V_{k-1})$. So, (v_{ki}, t_{ki}) is an edge in Q for all $1 \leq k \leq m$, $1 \leq i \leq p_k$. Notice $|V_1| + |V_2| + \cdots + |V_m| = p_1 + p_2 + \cdots + p_k = s - 1$. So given a spanning tree Q in $K_{s,t}$, to show that it comes by the algorithm from a *unique* legal pair $(\overrightarrow{T}, (A_1, \ldots, A_s))$. We first remove from Q a selected set of $s-1$ edges that will reveal $\overrightarrow{T}$ and $(A_1, \ldots, A_s)$.

Theorem 2.8. *For any spanning tree Q in $K_{s,t}$, it is generated by a unique legal pair $(\overrightarrow{T}, (A_1, \ldots, A_s))$.*

Proof. We first reveal $\overrightarrow{T}$ by building $\overrightarrow{T}$ level-by-level. Let 1 be the designated root. First, attach all vertices in V_1 to 1 (oriented towards 1). To each vertex j in V_2 attach the arc (j,i) where i is by Lemma 2.7 the unique vertex in V_1 with $N_Q(j) \cap N_Q(i) \neq \emptyset$. More generally, vertices in V_k are attached towards vertices in V_{k-1} as follows. To each vertex j in V_k we attach the arc $(\overrightarrow{j,i})$ where i is, by Lemma 2.7, the unique vertex in V_{k-1} with $N_Q(j) \cap N_Q(i) \neq \emptyset$. The resulting unique oriented tree is $\overrightarrow{T}$ in the algorithm. To reveal the associated partition $(A_1, \dots, A_s)$ we remove the $s-1$ edges $\{((v_{11}, t_{11}), (v_{12}, t_{12}), \dots, (v_{1p_1}, t_{1p_1})) \cdots ((v_{m1}, t_{m1}), (v_{m2}, t_{m2}) \dots, (v_{mp_m}, t_{np_n}))\}$. Let H denote this subgraph of Q. Then H is a collection of disjoint stars, and to each red vertex j we define $A_j = N_H(j)$ yielding the partition $(A_1, \dots, A_s)$. The pair $(\overrightarrow{T}, (A_1, \dots, A_s)$ is legal since if $(\overrightarrow{i,j})$ is an arc in $\overrightarrow{T}$, we have i is in some level V_k and j is in V_{k-1} with $N_Q(i) \cap N_Q(j) \neq \emptyset$, consequently, $A_j \neq \emptyset$. □

Combining Theorems 2.5 and 2.8 we show some enumeration.

3 Some Enumeration

Using the notation of Thm. 2.1, there are $k_1^{|N^-(1)|} k_{x_1}^{|N^-(x_1)|} \; k_{x_2}^{|N^-(x_2)|} \cdots k_{x_r}^{|N^-(x_r)|}$ trees generated by the fixed pair $(\overrightarrow{T}, (A_1, \dots, A_s))$, hence for the fixed tree $\overrightarrow{T}$ there are in total

$$\begin{aligned} &\sum k_1^{|N^-(1)|} k_{x_1}^{|N^-(x_1)|} \cdots k_{x_r}^{|N^-(x_r)|} \binom{t}{k_1} \binom{t-k_1}{k_2} \\ &\qquad \times \binom{t-k_1-k_2}{k_3} \times \cdots \binom{t-k_1-\cdots-k_{s-1}}{k_s} \qquad (1) \\ &= \sum_{k_1+\cdots+k_s=t} k_1^{|N^-(1)|} k_{x_1}^{|N^-(x_1)|} \cdots k_{x_r}^{|N^-(x_r)|} \frac{t!}{k_1! k_2! \cdots k_s!} \end{aligned}$$

spanning trees generated.

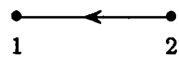
Let $H(\overrightarrow{T})$ denote the above formula for a fixed di-tree $\overrightarrow{T}$. Hence we have:

Theorem 3.1. *$\tau(K_{s,t}) = \sum_{\overrightarrow{T}} H(\overrightarrow{T})$, where the sum is over all of the s^{s-2} rooted di-trees $\overrightarrow{T}$.*

We illustrate Thm. 3.1 for the cases $s = 2, 3, 4, 5$. The enumeration involves the usual techniques involving multinomial identities.

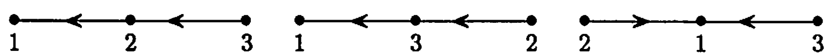
Example 3.1. $\mathbf{K_{2,n}}$

For the case $K_{2,n}$ we have only one tree, namely

$\overrightarrow{T} =$ 

From Theorem 3.1 we have $\tau(K_{2,n}) = \sum\limits_{k_1=1}^{n} k_1\binom{n}{k_1}$. Let $f(x,y) = (x+y)^n = \sum \binom{n}{k_1} x^{k_1} y^{n-k_1}$, then $\frac{\partial}{\partial x} f = n(x+y)^{n-1} =$
$\sum k_1 \binom{n}{k_1} x^{k_1-1} y^{n-k_1}$, letting $x = y = 1$ we obtain $\tau(K_{2,n}) = n(1+1)^{n-1} = n2^{n-1}$.

Example 3.2. $\mathbf{K_{3,n}}$ For the case $K_{3,n}$ we have three di-trees



$\overrightarrow{T}_1$ $\overrightarrow{T}_2$ $\overrightarrow{T}_3$

by Eq. (1) we have, $H(\overrightarrow{T}_1) = \sum k_1 k_2 \frac{n!}{k_1!k_2!k_3!}$, $H(\overrightarrow{T}_2) = \sum k_1 k_3 \frac{n!}{k_1!k_2!k_3!}$, and $H(\overrightarrow{T}_3) = \sum k_1^2 \frac{n!}{k_1!k_2!k_3!}$.

Notice by symmetry $H(\overrightarrow{T}_1) = H(\overrightarrow{T}_2)$. For $H(\overrightarrow{T}_1)$ let $f(x,y,z) = (x+y+z)^n$, then

$$\frac{\partial}{\partial x} f = n(x+y+z)^{n-1} = \sum k_1 \frac{n!}{k_1!k_2!k_3!} x^{k_1-1} y^{k_2} z^{k_3}$$

and

$$\frac{\partial^2 f}{\partial y \partial x} = n(n-1)(x+y+z)^{n-2} = \sum k_1 k_2 \frac{n!}{k_1!k_2!k_3!} x^{k_1-1} y^{k_2-1} z^{k_3}$$

letting $x = y = z = 1$ we obtain $H(\overrightarrow{T}_1) = 3^{n-2} n(n-1)$. For $H(\overrightarrow{T}_3)$, let $f(x) = (x+y+z)^n$, then

$$x\left[\frac{\partial f}{\partial x}\right] = nx(x+y+z)^{n-1} = \sum k_1 \frac{n!}{k_1!k_2!k_3!} x^{k_1} y^{k_2} z^{k_3},$$

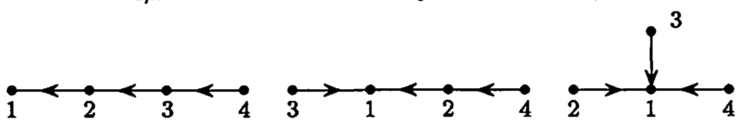
and

$$\frac{\partial}{\partial x}\left[x\frac{\partial f}{\partial x}\right] = n(x+y+z)^{n-1} + nx(n-1)(x+y+z)^{n-2}$$
$$= \sum k_1^2 \frac{n!}{k_1!k_2!k_3!} x^{k_1-1} y^{k_2} z^{k_3},$$

letting $x = y = z = 1$ we obtain $H(\overrightarrow{T}_3) = n3^{n-1} + n(n-1)3^{n-2}$. Hence $\tau(K_{3,n}) = 2n(n-1)3^{n-2} + n3^{n-1} + n(n-1)3^{n-2} = 3^{n-1}n^2$.

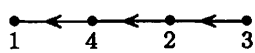
Example 3.3. $K_{4,n}$

For the case $K_{4,n}$ consider the following three di-trees,



$\overrightarrow{T}_1$ $\overrightarrow{T}_2$ $\overrightarrow{T}_3$

For any other di-tree, $\overrightarrow{T}$ of order 4 we have $H(\overrightarrow{T}) = H(\overrightarrow{T}_i)$ for some $i = 1, 2, 3$. To see this let



$\overrightarrow{T}$

be another $\overrightarrow{T}$, then, by symmetry,

$$H(\overrightarrow{T}) = \sum_{k_1+\cdots+k_4=n} k_1k_2k_4 \frac{n!}{k_1!k_2!k_3!k_4!} = \sum k_1k_2k_3 \frac{n!}{k_1!k_2!k_3!k_4!}$$
$$= H(\overrightarrow{T}_1).$$

For this case we say tree $\overrightarrow{T}$ is of type $\overrightarrow{T}_1$, of the 16 di-trees of order 4, one may draw them and check that there are six of type $\overrightarrow{T}_1$, nine of type $\overrightarrow{T}_2$ and one of type $\overrightarrow{T}_3$. Using the same techniques as in the previous illustration, starting with $f(x,y,z,w) = (x+y+z+w)^n$ we obtain

$$\begin{aligned}\tau(K_{4,n}) &= 6\sum k_1k_2k_3\frac{n!}{k_1!\cdots k_4!} + 9\sum k_1^2k_2\frac{n!}{k_1!\cdots k_4!} \\ &\qquad + \sum k_1^3\frac{n!}{k_1!\cdots k_4!} \\ &= 6n(n-1)(n-2)4^{n-3} + 9\big[n(n-1)4^{n-2} \\ &\qquad + n(n-1)(n-2)4^{n-3}\big] + n4^{n-1} \\ &\qquad + 3n(n-1)4^{n-2} + n(n-1)(n-2)4^{n-3} \\ &= n^3 4^{n-1}.\end{aligned}$$

More generally, there are three types of di-trees in the above example because there are $p(3) = 3$ partitions of the integer 3, namely; 3, 2+1, 1+1+1. Here $p(n)$ denotes the number of partitions of n. For a given $\overrightarrow{T}$ the beginning terms in formula (1) are of the form $k_1^{|N^-(1)|}k_{x_1}^{|N^-(x_1)|}\ \cdots\ k_{x_r}^{|N^-(x_r)|}$. Suppose we have another tree $\overrightarrow{Q}$, if its beginning terms are of the form $k_1^{|N^-(1)|}k_{y_1}^{|N^-(y_1)|}\ \cdots\ k_{y_r}^{|N^-(y_r)|}$ and the unordered sets

$$\begin{aligned}&\{|N^-(1)|, |N^-(x_1)|\ \cdots\ |N^-(x_r)|\} \\ &\qquad = \{|N^-(1)|, |N^-(y_1)|\ \cdots\ |N^-(y_r)|\}\end{aligned} \tag{2}$$

are equal, then by the symmetry of the formula (1); $H(\overrightarrow{T}) = H(\overrightarrow{Q})$. So more generally we say trees $\overrightarrow{T}$ and $\overrightarrow{Q}$ are the same *types* if equation (2) is satisfied. Hence we partition the s^{s-2} di-trees into $p(s-1)$ equivalent classes, where two trees are equivalent if they are the same type. In the $K_{4,n}$ example, the 3 partitions $p(4-1) = 3$, $2+1$, $1+1+1$, led to the three types of classes k_1^3, $k_1^2k_y$, $k_1k_yk_z$ and as mentioned there are 1, 9, 6 trees in their respective classes.

For the case $K_{5,n}$, there are $p(5-1) = p(4) = 5$ *types* of di-trees. The 5 partitions of 4, namely, 4, $3+1$, $2+2$, $2+1+1$, $1+1+1+1$, yield the tree types k_1^4, $k_1^3k_x$, $k_1^2k_x^2$, $k_1^2k_xk_y$, $k_1k_xk_yk_z$. The reader may draw the $5^3 = 125$ trees on 5 vertices, and check that the di-trees are then partitioned into one of type k_1^4, 16 of type $k_1^3k_x$, 12 of type $k_1^2k_x^2$, 72 of type $k_1^2k_xk_y$, and 24 of type $k_1k_xk_yk_z$, with $1+16+12+72+24=125$, and using the above techniques with $f(x_1,x_2,x_3,x_4,x_5)=(x_1+x_2+x_3+x_4+x_5)^n$ one can show:

$$\begin{aligned}\tau(K_{5,n}) = 5^{n-1}n^4 &= \sum k_1^4\frac{n!}{k_1!\cdots k_5!}+16\sum k_1^3k_2\frac{n!}{k_1!\cdots k_5!}\\ &+12\sum k_1^2k_2^2\frac{n!}{k_1!\cdots k_5!}+72\sum k_1^2k_2k_3\frac{n!}{k_1!\cdots k_5!}\\ &+24\sum k_1k_2k_3k_4\frac{n!}{k_1!\cdots k_5!}.\end{aligned}$$

Remarks. For the general case $K_{s,t}$ we have there are $p(s-1)$ types of equivalent classes. For the first few values of s we were able to compute the number of di-trees in each class, but we do not have a general way to count the size of those classes. For each partition $a_1+a_2+\cdots+a_x=s-1$ of $s-1$, let $A_{a_1,\ldots,a_x}$ denote the number of di-trees in the equivalent class $k_1^{a_1}k_2^{a_2}\cdots k_x^{a_x}$, then

$$\begin{aligned}\tau(K_{s,t}) = s^{t-1}t^{s-1} = \sum_{a_1+\cdots+a_x=s-1}\ \sum_{k_1+\cdots+k_s=t} & A_{a_1,\ldots,a_x}k_1^{a_1}\\ &\times\cdots k_x^{a_x}\frac{t!}{k_1!\cdots k_s!}\end{aligned}$$

where the outer sum runs over all $p(s-1)$ partitions of $s-1$.

Acknowledgment. The author thanks the referee for providing insight and suggestions that greatly improved the paper.

References

[1] T.L. Austin, The enumeration of point labelled chromatic graphs and trees, *Canad. J. Math.* **12** (1960), 535–545.

[2] N. Biggs, *Algebraic Graph Theory*, Cambridge University Press, 1993.

[3] L. Clark, On the enumeration of multipartite spanning trees of the complete graph, *Bull. of the ICA* **38** (2003), 50–60.

[4] O. Eğecioğlu and J.B. Remmel, A bijection for spanning trees of complete multipartite graphs, *Congressus Numerantium* **100** (1994), 225–243.

[5] I.J. Good, The genrealization of Lagrange's expansion and the enumeration of trees, *Proceedings of the Cambridge Phiolosophical Society* **61** (1965), 499–517.

[6] A.K. Kelmans, The number of trees of a graph I, *Automat. Remote Control* **26** (1965), 2194–2204.

[7] R.P. Lewis, The number of spanning trees of a complete multipartite graph *Discrete Mathematics* **197/198** (199), 537–541.

[8] R. Merris, *Graph Theory*, Wiley, New York, 2001.

[9] D.B. West, *Introduction to Graph Theory*, Second Edition, Prentice-Hall, Upper Saddle River, NJ, 2001.