Journal of Combinatorics
Volume 8, Number 2, 371–387, 2017

# Enumeration of constrained subtrees of trees*

Hua Wang and Shuai Yuan

Starting from the work of Székely and Wang, where the extremal trees and binary trees that maximize or minimize the number of subtrees are characterized, the examination of the numbers of subtrees has been an interesting topic providing applications and questions in Phylogeny reconstruction, Graph Theory, Number Theory, and Computer Science. We present linear-time algorithms for enumerations of subtrees under various constraints. Such specific categories of subtrees including but not limited to those of given orders and those containing vertices of a given set, are of interests due to their applications.

Keywords and phrases: Tree, subtree, enumeration.

## 1. Introduction

For a tree (connected acyclic graph) $T$, a subtree is simply a connected subgraph of $T$. When every vertex is considered different, the number of subtrees (where isomorphic subtrees consisting of different vertices are considered different) has been an interesting topic since the appearance of [8], where the extremal trees and binary trees that maximize or minimize the number of subtrees are studied. While formulating an explicit formula for the maximum number of subtrees of binary trees, a novel binary representation of integers was proposed and further studied in [2]. A concept similar in nature is the number of leaf-containing subtrees, whose extremal structures were studied for binary trees [9]. The leaf-containing subtrees of a binary tree turned out to be exactly the acceptable residue configurations, the number of which bounds the complexity of Knudsen's multiple parsimony alignment algorithm with affine gap cost in phylogenetic tree reconstruction [6].

At the same time, the extremal structures that maximize or minimize the number of subtrees coincide with the extremal trees for many other graph invariants. In particular, the correlations between several graph-theoretical indices were analyzed in [13] where the number of subtrees and the well-known Wiener index (sum of distances between all pairs of vertices) [15]

*This work was partially supported by grants from the Simons Foundation (#245307).

were found to have the strongest "negative" correlation. Because of these connections with other important parameters of trees, the number of subtrees of various categories of trees have been further studied. In [5], the maximum number of subtrees of a tree with given order and maximum degree is identified, matching the analogous work on the Wiener index [1]. This study was further generalized to trees with a given degree sequence [17, 18] due to the special roles played by such trees in applications. The number of subtrees also provides a novel definition of the "middle part" of a tree [8], which was further investigated and compared with the well-established distance-based middle parts of graphs [10, 11].

The first paper focusing on the enumeration of subtrees seems to be [16], where a polynomial time algorithm is presented for counting subtrees of a general tree. Independently, [14] conducted similar studies with the order of subtrees taken into consideration. Note that the "total order" or "average order" of subtrees have been of interest in the examination of "densities" of trees [3, 4, 7, 12].

We first introduce some general terminologies and notations in Section 2 including the previous work in [14, 16]. Given the importance of subtrees containing some vertices from a given set (for instance, the leaf-containing subtrees), we consider the enumeration of such subtrees in Section 3 and present the corresponding algorithm. This result, besides providing the enumeration of leaf-containing subtrees as a special case, is also useful in related studies of weighted trees with a special set of vertices. In Section 4, we consider the more general question of enumerating subtrees with specific number of "special vertices". The enumeration of general subtrees of given order then follows as an immediate consequence. Lastly, we summarize our work as well as mentioning questions that can be resolved in similar ways but rather tedious to present.

## 2. Some terminologies and previous work

Throughout this paper, we let $T = (V(T), E(T); f, g)$ be a weighted tree with vertex set $V(T) = \{v_1, v_2, ..., v_n\}$, edge set $E(T) = \{e_1, e_2, ..., e_{n-1}\}$, vertex-weight function $f : V(T) \to \mathbb{R}$ and edge-weight function $g : E(T) \to \mathbb{R}$ unless defined otherwise. If $f = g = 1$ in such a tree, we call $T$ a *simple tree* denoted by $T = (V(T), E(T))$.

For a tree $T$, let $\tau(T)$ be the set of subtrees and $\tau_k(T)$ the set of subtrees of order $k$. For any vertex $v \in V(T)$ and any set $A \subset V(T)$, denote by $\tau(T; v)$ (resp. $\tau(T; A)$) the set of subtrees of a tree $T$, each of which contains $v$ (resp. vertices in $A$). Let $\tau_k(T; v)$ (resp. $\tau_k(T; A)$) be the set of subtrees with order

$k$ of a tree $T$, each of which contains $v$ (resp. vertices in $A$). For technical reasons, we also define $\tau_k(T; v; u)$ to be the set of subtrees of $T$ consisting of $v$ but not $u$ and of order $k$.

Following the terminologies in [16], we call $F_T(k)$ the *generating function* of subtrees of given order $k$ of a weighted tree $T = (V(T), E(T); f, g)$ (i.e., subtrees in $\tau_k(T)$), defined as

$$F_T(k) = \sum_{T_k \in \tau_k(T)} \omega(T_k)$$

where

$$\omega(T_k) = \prod_{v \in V(T_k)} f(v) \prod_{e \in E(T_k)} g(e)$$

is the product of the weights of the vertices and edges of $T_k$, called the *weight of* $T_k$.

Along the same line, let

$$F_T(k; v) = \sum_{T_k \in \tau_k(T; v)} \omega(T_k)$$

be the generating function of subtrees in $\tau_k(T; v)$. Similarly, $F_T(k; v; u)$ denotes the generating function of subtrees of $\tau_k(T; v; u)$.

Given a weighted tree $T = (V(T), E(T); f, g)$ of order $n > 1$ and pendant edge $e = (u, v_0)$ with leaf $u$, we will use $T' = (V(T'), E(T'); f', g')$ to denote the tree $T - \{u\}$ of order $n - 1$ (Figure 1) where $f'$ and $g'$ are $f$ and $g$ restricted to $V(T')$ and $E(T')$.


Figure 1: Generating $T'$ from $T$.

With the above set up, [14, 16] provided recursive formulation of $F_T(k)$ and consequently the corresponding algorithm to enumerate subtrees of given order. The fundamental idea of the approaches to be presented, as that in [14, 16], is to repetitively contracting pendant edges and record the changes of corresponding parameters through updating the label of $v_0$. The specific labeling system and updating procedure depends on different questions.

## 3. Subtrees containing vertices of a given set

Motivated by the concept of leaf-containing subtrees (i.e., subtrees that contain at least one of the original leaf vertices) [6, 9], we consider the enumeration of the more general objects in this section. For a given set of vertices $\mathbb{S} \subset V(T)$, we use $\Omega_{\mathbb{S}}(T)$ to denote the set of subtrees of $T$ containing at least one vertex from $\mathbb{S}$. Given a vertex $v$, $\Omega_{\mathbb{S}}(T;v)$ is the set of subtrees of $T$ containing $v$ and at least one vertex from $\mathbb{S}$.

To enumerate the number of subtrees of a tree $T_0 := T$ that contain at least one vertex from $\mathbb{S} \subset V(T)$, we label each $v \in V(T)$ by $(b_T(1;v), b_T(2;v))$ with

$$b_T(1;v) = \begin{cases} 0 & v \notin \mathbb{S}; \\ 1 & v \in \mathbb{S}; \end{cases}$$

and

$$b_T(2;v) = 1 - b_T(1;v)$$

for all $v \in V(T)$ at the begining (when $T = T_0$).

Let

$$\phi(T) = \omega(T) \cdot m(T)$$

where

$$m(T) = \prod_{v \in V(T)} (b_T(1;v) + b_T(2;v)),$$

the generating function $F_T(\mathbb{S})$ of subtrees containing at least one vertex in $\mathbb{S}$ is defined as

$$F_T(\mathbb{S}) = \sum_{T_s \in \Omega_{\mathbb{S}}(T)} \phi(T_s).$$

When a tree $T$ is reduced to $T'$ (Figure 1, starting from $T = T_0$), we define

$$b_{T'}(i;v_0) = \begin{cases} f(u)g(e)\left[(b_T(1;v_0) + b_T(2;v_0))b_T(1;u) + b_T(1;v_0)b_T(2;u)\right] \\ \quad + b_T(1;v_0) & i = 1; \\ f(u)g(e)b_T(2;v_0)b_T(2;u) + b_T(2;v_0) & i = 2. \end{cases}$$

and

$$b_{T'}(i;v) = b_T(i;v)$$

for any $v \neq v_0, u$.

We show the following "recursion" of the function $F_T(\mathbb{S})$.

**Theorem 3.1.** *From $T$ to $T'$ in the process of contracting pendant edges (Figure 1), we have*

$$F_T(\mathbb{S}) = F_{T'}(\mathbb{S}) + f(u)b_T(1; u).$$

*Proof.* First we partition the sets $\tau(T)$ and $\tau(T')$ of subtrees of $T$ and $T'$ as

$$\tau(T) = \tau_1 \cup \tau_{1'} \cup \tau_2 \cup \tau_3$$

and

$$\tau(T') = \tau_1' \cup \tau_2'$$

where

- $\tau_1$ is the set of subtrees of $T$ containing $v_0$ but not $u$;
- $\tau_{1'}$ is the set of subtrees of $T$ containing the edge $e = (u, v_0)$;
- $\tau_2$ is the set of subtrees of $T$ containing neither $u$ nor $v_0$;
- $\tau_3$ is the set of subtrees of $T$ containing $u$ but not $v_0$;
- $\tau_1'$ is the set of subtrees of $T'$ containing $v_0$;
- $\tau_2'$ is the set of subtrees of $T'$ not containing $v_0$.

It is easy to observe

1. bijections
$$\theta_1 : T_1 \mapsto T_1'$$
between $\tau_1$ and $\tau_1'$, and
$$\theta_2 : T_2 \mapsto T_2'$$
between $\tau_2$ and $\tau_2'$;
2. the bijection between $\tau_{1'}$ and $\tau_1$ defined through
$$\tau_{1'} = \{T_1 + u | T_1 \in \tau_1\},$$
where $T_1 + u$ is the tree obtained from $T_1$ by attaching a pendant edge $(v_0, u)$ at vertex $v_0$ of $T_1$;
3. that $\tau_3$ is the single element set that contains the single vertex subtree $\{u\}$.

Let

$$\phi_1(T; v_0) = \frac{\phi(T)}{b_T(1; v_0) + b_T(2; v_0)} b_T(1; v_0)$$

and

$$\phi_2(T; v_0) = \frac{\phi(T)}{b_T(1; v_0) + b_T(2; v_0)} b_T(2; v_0),$$

we have

$$\phi(T) = \phi_1(T; v_0) + \phi_2(T; v_0)$$

and

$$\frac{\phi_1(T; v_0)}{b_T(1; v_0)} = \frac{\phi_2(T; v_0)}{b_T(2; v_0)}. \tag{1}$$

Note that in both expressions in (1) one may have the undetermined form $\frac{0}{0}$. In this case we are simply using them as symbolic expressions to denote the true value $\frac{\phi(T)}{b_T(1;v_0)+b_T(2;v_0)}$.

From the bijections

$$T_1 \leftrightarrow T_{1'} \leftrightarrow T_1'$$

between subtrees $T_1 \in \tau_1$, $T_{1'} \in \tau_{1'}$, $T_1' \in \tau_1'$, we have

$$V(T_1) = V(T_1') = V(T_{1'}) - \{u\}$$

and

$$E(T_1) = E(T_1') = E(T_{1'}) - \{uv_0\}.$$

For any $T_1 \in \tau_1$, we have

$$\begin{aligned}
&\frac{\phi_1(T_1; v_0)}{b_{T_1}(1; v_0)} \\
&= \frac{\phi(T_1)}{b_{T_1}(1; v_0) + b_{T_1}(2; v_0)} \\
&= \frac{\prod_{v \in V(T_1)} f(v) \prod_{e \in E(T_1)} g(e) \prod_{v \in V(T_1)} (b_{T_1}(1; v) + b_{T_1}(2; v))}{b_{T_1}(1; v_0) + b_{T_1}(2; v_0)} \\
&= \prod_{v \in V(T_1)} f(v) \prod_{e \in E(T_1)} g(e) \prod_{v \in V(T_1) \setminus \{v_0\}} (b_{T_1}(1; v) + b_{T_1}(2; v)).
\end{aligned} \tag{2}$$

Similarly,

$$\begin{aligned}
&\frac{\phi_1(T_{1'}; v_0)}{f(u)g(uv_0)b_{T_{1'}}(1; v_0)\left(b_{T_{1'}}(1; u) + b_{T_{1'}}(2; u)\right)} \\
&= \frac{\phi(T_{1'})}{f(u)g(uv_0)\left(b_{T_{1'}}(1; v_0) + b_{T_{1'}}(2; v_0)\right)\left(b_{T_{1'}}(1; u) + b_{T_{1'}}(2; u)\right)} \\
&= \frac{\prod_{v \in V(T_{1'})} f(v) \prod_{e \in E(T_{1'})} g(e) \prod_{v \in V(T_{1'})} (b_{T_{1'}}(1; v) + b_{T_1'}(2; v))}{f(u)g(uv_0)\left(b_{T_{1'}}(1; v_0) + b_{T_{1'}}(2; v_0)\right)\left(b_{T_{1'}}(1; u) + b_{T_{1'}}(2; u)\right)}
\end{aligned}$$

$$= \prod_{v\in V(T_{1'})\setminus\{u\}} f(v) \prod_{e\in E(T_{1'})\setminus\{uv_0\}} g(e) \prod_{v\in V(T_{1'})\setminus\{u,v_0\}} (b_{T_{1'}}(1;v)+b_{T_{1'}}(2;v))$$
$$= \prod_{v\in V(T_1)} f(v) \prod_{e\in E(T_1)} g(e) \prod_{v\in V(T_1)\setminus\{v_0\}} (b_{T_1}(1;v)+b_{T_1}(2;v))$$

(3)

for any $T_{1'} \in \tau_{1'}$ (that is mapped to $T_1$ through the bijection) and

$$\frac{\phi_1(T_1';v_0)}{b_{T_1'}(1;v_0)}$$
$$= \frac{\phi(T_1')}{b_{T_1'}(1;v_0)+b_{T_1'}(2;v_0)}$$
$$= \frac{\prod_{v\in V(T_1')} f(v) \prod_{e\in E(T_1')} g(e) \prod_{v\in V(T_1')}(b_{T_1'}(1;v)+b_{T_1'}(2;v))}{b_{T_1'}(1;v_0)+b_{T_1'}(2;v_0)}$$
$$= \prod_{v\in V(T_1')} f(v) \prod_{e\in E(T_1')} g(e) \prod_{v\in V(T_1')\setminus\{v_0\}} (b_{T_1'}(1;v)+b_{T_1'}(2;v))$$
$$= \prod_{v\in V(T_1)} f(v) \prod_{e\in E(T_1)} g(e) \prod_{v\in V(T_1)\setminus\{v_0\}} (b_{T_1}(1;v)+b_{T_1}(2;v))$$

(4)

for any $T_1' \in \tau_1'$ (that is mapped to $T_1$ through the bijection).

Let $e$ be the edge $uv_0$, now we have

(5)
$$\frac{\phi_1(T_1\in\tau_1;v_0)}{b_{T_1}(1;v_0)} = \frac{\phi_1(T_{1'}\in\tau_{1'};v_0)}{f(u)g(e)b_{T_{1'}}(1;v_0)\,(b_{T_{1'}}(1;u)+b_{T_{1'}}(2;u))} = \frac{\phi_1(T_1'\in\tau_1';v_0)}{b_{T_1'}(1;v_0)},$$

and similarly

(6)
$$\frac{\phi_2(T_1\in\tau_1;v_0)}{b_{T_1}(2;v_0)} = \frac{\phi_2(T_{1'}\in\tau_{1'};v_0)}{f(u)g(e)b_{T_{1'}}(2;v_0)\,(b_{T_{1'}}(1;u)+b_{T_{1'}}(2;u))} = \frac{\phi_2(T_1'\in\tau_1';v_0)}{b_{T_1'}(2;v_0)}.$$

Again, the expressions $\frac{\phi_1(T_1\in\tau_1;v_0)}{b_{T_1}(1;v_0)}$, $\frac{\phi_1(T_1'\in\tau_1';v_0)}{b_{T_1'}(1;v_0)}$, $\frac{\phi_2(T_1\in\tau_1;v_0)}{b_{T_1}(2;v_0)}$ and $\frac{\phi_2(T_1'\in\tau_1';v_0)}{b_{T_1'}(2;v_0)}$ are merely denoting the true value provided in (5) and (6) when $\frac{0}{0}$ occurs.

Consequently

$$\sum_{T\in\tau_1} \phi(T)$$

$$
\begin{aligned}
&= \sum_{T\in\tau_1} (\phi_1(T;v_0)+\phi_2(T;v_0)) \\
&= \sum_{T\in\tau_1} \frac{\phi_1(T\in\tau_1;v_0)}{b_T(1;v_0)} b_T(1;v_0) + \sum_{T\in\tau_1} \frac{\phi_2(T\in\tau_1;v_0)}{b_T(2;v_0)} b_T(2;v_0) \\
&= \sum_{T\in\tau_1'} \frac{\phi_1(T\in\tau_1';v_0)}{b_T(1;v_0)} b_{T\in\tau_1}(1;v_0) + \sum_{T\in\tau_1'} \frac{\phi_2(T\in\tau_1';v_0)}{b_T(2;v_0)} b_{T\in\tau_1}(2;v_0) \qquad (7)
\end{aligned}
$$

and

$$
\begin{aligned}
&\sum_{T\in\tau_{1'}} \phi(T) \\
&= \sum_{T\in\tau_{1'}} (\phi_1(T;v_0)+\phi_2(T;v_0)) \\
&= \sum_{T\in\tau_{1'}} \frac{\phi_1(T\in\tau_{1'};v_0)}{f(u)g(e)b_T(1;v_0)\left(b_T(1;u)+b_T(2;u)\right)} \\
&\cdot f(u)g(e)b_T(1;v_0)\left(b_T(1;u)+b_T(2;u)\right) \\
&+ \sum_{T\in\tau_{1'}} \left(\frac{\phi_2(T\in\tau_{1'};v_0)}{f(u)g(e)b_T(2;v_0)\left(b_T(1;u)+b_T(2;u)\right)} \cdot f(u)g(e)b_T(2;v_0)b_T(1;u)\right) \\
&+ \sum_{T\in\tau_{1'}} \left(\frac{\phi_2(T\in\tau_{1'};v_0)}{f(u)g(e)b_T(2;v_0)\left(b_T(1;u)+b_T(2;u)\right)} \cdot f(u)g(e)b_T(2;v_0)b_T(2;u)\right).
\end{aligned}
$$

Using $b_{T\in\tau_{1'}}(.;.)$ to denote the labeling in the corresponding trees in the above expression, we have

$$
\begin{aligned}
&\sum_{T\in\tau_{1'}} \phi(T) \\
&= \sum_{T\in\tau_1'} \frac{\phi_1(T\in\tau_1';v_0)}{b_T(1;v_0)} \cdot f(u)g(e)b_{T\in\tau_{1'}}(1;v_0)\left(b_{T\in\tau_{1'}}(1;u)+b_{T\in\tau_{1'}}(2;u)\right) \\
&+ \sum_{T\in\tau_1'} \left(\frac{\phi_2(T\in\tau_1';v_0)}{b_T(2;v_0)} \cdot f(u)g(e)b_{T\in\tau_{1'}}(2;v_0)b_{T\in\tau_{1'}}(1;u)\right) \\
&+ \sum_{T\in\tau_1'} \left(\frac{\phi_2(T\in\tau_1';v_0)}{b_T(2;v_0)} \cdot f(u)g(e)b_{T\in\tau_{1'}}(2;v_0)b_{T\in\tau_{1'}}(2;u)\right) \qquad (8)
\end{aligned}
$$

Note that, by our definition of $b_T(1;v_0)$ and $b_T(2;v_0)$ for a subtree of $T'$, the first term in (7) and the first two terms in (8) sum up to

$$\sum_{T\in\tau_1'}\left(\frac{\phi_1(T\in\tau_1';v_0)}{b_T(1;v_0)}b_T(1;v_0)\right)$$

and the last terms in (7) and (8) sum up to

$$\sum_{T\in\tau_1'}\left(\frac{\phi_2(T\in\tau_1';v_0)}{b_T(2;v_0)}b_T(2;v_0)\right).$$

Thus we have

$$\begin{aligned}
&\sum_{T\in\tau_1}\phi(T)+\sum_{T\in\tau_{1'}}\phi(T)\\
&=\sum_{T\in\tau_1'}\left(\frac{\phi_1(T\in\tau_1';v_0)}{b_T(1;v_0)}b_T(1;v_0)\right)+\sum_{T\in\tau_1'}\left(\frac{\phi_2(T\in\tau_1';v_0)}{b_T(2;v_0)}b_T(2;v_0)\right)\\
&=\sum_{T\in\tau_1'}\phi_1(T\in\tau_1';v_0)+\sum_{T\in\tau_1'}\phi_2(T\in\tau_1';v_0)\\
&=\sum_{T\in\tau_1'}\phi(T).
\end{aligned}$$

Through the earlier established bijections, it is easy to see that

$$\sum_{T\in\tau_2}\phi(T)=\sum_{T\in\tau_2'}\phi(T)$$

and

$$\sum_{T\in\tau_3}\phi(T)=f(u)b_T(1;u).$$

Then

$$\begin{aligned}
F_T(\mathbb{S})&=\sum_{T\in\tau_1}\phi(T)+\sum_{T\in\tau_{1'}}\phi(T)+\sum_{T\in\tau_2}\phi(T)+\sum_{T\in\tau_3}\phi(T)\\
&=\sum_{T\in\tau_1'}\phi(T)+\sum_{T\in\tau_2'}\phi(T)+f(u)b_T(1;u)\\
&=F_{T'}(\mathbb{S})+f(u)b_T(1;u) \qquad (9)
\end{aligned}$$

as desired. □

As a direct application of this theorem, we have the following algorithm that provide $F_T(\mathbb{S})$ as the output.

**ALGORITHM**
**Step 1.**
Initialization:

$$b_T(1;v) = \begin{cases} 0 & v \notin \mathbb{S}; \\ 1 & v \in \mathbb{S}; \end{cases}$$

and

$$b_T(2;v) = 1 - b_T(1;v)$$

for all $v \in V(T)$.
$N = 0$.
**Step 2.**
Contraction:

- Choose a pendant edge $e = (u, v_0)$ with leaf $u$;
- Update $b_T(i;v)$ with $b_{T'}(i;v)$ for all $v \in V(T')$;
- Update $N := N + f(u)b_T(1;u)$;
- Update $T := T'$.

**Step 3.**

- If $|T| = 1$, go to Step 4;
- Otherwise, go to Step 2.

**Step 4.**
Update $N := N + f(v)b_T(1;v)$ where $v$ is the only vertex in $V(T)$.
Output $F_T(\mathbb{S}) = N$.

**Remark 1.** *When $f = g = 1$, this algorithm computes the number of subtrees of $T$ when $\mathbb{S} = V(T)$ and the number of leaf-containing subtrees when $\mathbb{S}$ is the set of leaves of $T$.*

In Figure 2, an example is provided for a tree with $n = 7$, $f = g = 1$ and vertices of $\mathbb{S}$ denoted by larger nodes.

## 4. Subtrees containing given number of vertices of a given set

In this section we discuss the enumeration of subtrees of $T$ that contain a given number $k$ of the vertices from $\mathbb{S} \subset V(T)$. The essential idea is the same as that of the previous section, but notations are more technical and we skip some details. We denote by $c_T(i;v)$ the number of subtrees of $T$ (the original tree) containing $v$ and exactly $i$ vertices from $\mathbb{S}$. Each vertex $v \in V(T)$

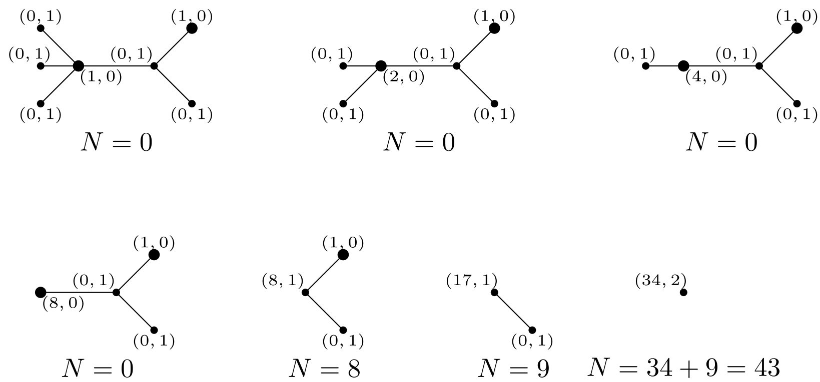


Figure 2: Example.

will be labeled with the $(k+1)$-tuple $(c_T(k;v), c_T(k-1;v)\cdots, c_T(0;v))$ in this examination. Note that these labels are later only used to record the information needed.

Following the same procedure as in Figure 1, we use $\mathscr{F}$ and $\mathscr{F}'$ denote set of subtrees of $T$ and that of $T'$. We need to introduce a few more notations before getting to the main result of this section.

Let

$$\lambda_k(T) = \sum_{\sum_{v\in V(T)} i_v = k} \left( \prod_{v\in V(T)} c_T(i_v; v) \right),$$

where the summation picks an $k \geq i_v \geq 0$ for each vertex $v \in V(T)$ under the condition that the sum of these values is $k$. For a vertex $v \in V(T)$, we define

$$\lambda_k(T; v; j) = \sum_{\sum_{u\in V(T)\setminus\{v\}} i_u = k-j} c_T(j; v) \left( \prod_{u\in V(T)\setminus\{v\}} c_T(i_u; u) \right).$$

It is obvious that

$$\lambda_k(T) = \sum_{j=0}^{k} \lambda_k(T; v; j)$$

when $v \in V(T)$.

For technical reasons, we also define $c_T(i;e)$ and $\lambda_k(T;e;j)$ accordingly for an edge $e \in E(T)$. The generating function of subtrees of a labeled weighted tree $T$, each containing $k$ vertices from $\mathbb{S}$, is then denoted by

$$F_T(k;\mathbb{S}) = \sum_{T_s \in \mathscr{F}} w(T_s)\lambda_k(T_s).$$

**Remark 2.** *Note that in the above summation, the subtree $T_s$ has non-negative contribution only if $\sum_{v_i \in V(T_s)} i_{v_i} \geq k$, where $i_{v_i}$ is the largest subscript $0 \leq i \leq k$ such that $c_T(i;v) > 0$ in the label $(c_T(k;v), c_T(k-1;v) \cdots, c_T(0;v))$.*

To deal with the operation from $T$ to $T'$, we provide the following recursive definition of the label of $v_0$:

$$c_{T'}(i;v_0) = c_T(i;v_0) + f(u)g(e) \sum_{m+n=i;\ m,n \geq 0} c_T(m;v_0)c_T(n;u), \quad 0 \leq i \leq k.$$

Note that except for the labels of the original tree $T = T_0$, $c_T(i;v)$ will be simply used for recording information and not necessarily reflect its original definition on the current tree.

Let $e$ be the edge $uv_0$ in Figure 1 and the sets $\tau_1$, $\tau_2$, $\tau_{1'}$, $\tau_3$, $\tau_1'$ and $\tau_2'$ defined as before. Noting the bijections

$$T_1 \leftrightarrow T_{1'} \leftrightarrow T_1'$$

as in the previous section and let

$$\begin{aligned} &\lambda_k(T;e;j) \\ &= \sum_{\sum_{v \in V(T)\setminus\{u,v_0\}} i_v = k-j} \left( \sum_{m+n=j} c_T(m;v_0)c_T(n;u) \right) \left( \prod_{v \in V(T)\setminus\{u,v_0\}} c_T(i_v;v) \right), \end{aligned}$$

We have, for any $T_1 \in \tau_1$, $T_{1'} \in \tau_{1'}$, $T_1' \in \tau_1'$, $1 \leq k \leq n$ and $0 \leq j \leq k$,

$$\frac{w(T_1)\lambda_k(T_1;v_0;j)}{c_{T_1}(j;v_0)} = w(T_1) \sum_{\sum_{v \in V(T_1)\setminus\{v_0\}} i_v = k-j} \left( \prod_{v \in V(T_1)\setminus\{v_0\}} c_{T_1}(i_v;v) \right),$$

$$\begin{aligned} &\frac{w(T_{1'})\lambda_k(T_{1'};e;j)}{f(u)g(e)\sum_{m+n=j} c_{T_{1'}}(m;v_0)c_{T_{1'}}(n;u)} \\ &= \frac{w(T_{1'})}{f(u)g(e)} \cdot \frac{\lambda_k(T_{1'};e;j)}{\sum_{m+n=j} c_{T_{1'}}(m;v_0)c_{T_{1'}}(n;u)} \end{aligned}$$

$$= w(T_1) \sum_{\sum_{v\in V(T_{1'})\setminus\{u,v_0\}} i_v = k-j} \left( \prod_{v\in V(T_{1'})\setminus\{u,v_0\}} c_{T_{1'}}(i_v; v) \right)$$

$$= w(T_1) \sum_{\sum_{v\in V(T_1)\setminus\{v_0\}} i_v = k-j} \left( \prod_{v\in V(T_1)\setminus\{v_0\}} c_{T_1}(i_v; v) \right),$$

and

$$\frac{w(T_1')\lambda_k(T_1'; v_0; j)}{c_{T_1'}(j; v_0)}$$

$$= w(T_1) \cdot \frac{\lambda_k(T_1'; v_0; j)}{c_{T_1'}(j; v_0)}$$

$$= w(T_1) \sum_{\sum_{v\in V(T_1')\setminus\{v_0\}} i_v = k-j} \left( \prod_{v\in V(T_1')\setminus\{v_0\}} c_{T_1'}(i_v; v) \right)$$

$$= w(T_1) \sum_{\sum_{v\in V(T_1)\setminus\{v_0\}} i_v = k-j} \left( \prod_{v\in V(T_1)\setminus\{v_0\}} c_{T_1}(i_v; v) \right).$$

Hence

$$\frac{w(T_1)\lambda_k(T_1; v_0; j)}{c_{T_1}(j; v_0)}$$

$$= \frac{w(T_{1'})\lambda_k(T_{1'}; e; j)}{f(u)g(e)\sum_{m+n=j} c_{T_{1'}}(m; v_0) c_{T_{1'}}(n; u)} = \frac{w(T_1')\lambda_k(T_1'; v_0; j)}{c_{T_1'}(j; v_0)}$$

We now have the necessary tools to show the following.

**Theorem 4.1.** *From $T$ to $T'$ in the process of contracting pendant edges (Figure 1), we have*

$$F_T(k; \mathbb{S}) = F_{T'}(k; \mathbb{S}) + f(u)c_T(k; u).$$

*Proof.* First note that

$$\sum_{T_s\in\mathscr{F}\cap\tau_1} w(T_s)\lambda_k(T_s) + \sum_{T_s\in\mathscr{F}\cap\tau_{1'}} w(T_s)\lambda_k(T_s)$$

$$= \sum_{j=0}^{k} \left( \sum_{T_1\in\mathscr{F}\cap\tau_1} w(T_1)\lambda_k(T_1; v_0; j) + \sum_{T_{1'}\in\mathscr{F}\cap\tau_{1'}} w(T_{1'})\lambda_k(T_{1'}; e; j) \right)$$

$$
\begin{aligned}
&= \sum_{j=0}^{k} \left( \frac{\sum_{T_1 \in \mathscr{F} \cap \tau_1} w(T_1)\lambda_k(T_1; v_0; j)}{c_{T_1}(j; v_0)} c_{T_1}(j; v_0) \right) \\
&\quad + \sum_{j=0}^{k} \left( \frac{\sum_{T_{1'} \in \mathscr{F} \cap \tau_{1'}} w(T_{1'})\lambda_k(T_{1'}; e; j)}{f(u)g(e) \sum_{m+n=j} c_{T_{1'}}(m; v_0) c_{T_{1'}}(n; u)} f(u)g(e) \right. \\
&\quad \left. \cdot \sum_{m+n=j} c_{T_{1'}}(m; v_0) c_{T_{1'}}(n; u) \right) \\
&= \sum_{j=0}^{k} \left( \frac{\sum_{T_1' \in \mathscr{F}' \cap \tau_1'} w(T_1')\lambda_k(T_1'; v_0; j)}{c_{T_1'}(j; v_0)} \cdot c_T(j; v_0) \right) \\
&\quad + \sum_{j=0}^{k} \left[ \frac{\sum_{T_1' \in \mathscr{F}' \cap \tau_1'} w(T_1')\lambda_k(T_1'; v_0; j)}{c_{T_1'}(j; v_0)} \right. \\
&\quad \left. \cdot \left( f(u)g(e) \sum_{m+n=j} c_T(m; v_0) c_T(n; u) \right) \right] \\
&= \sum_{T_s \in \mathscr{F}' \cap \tau_1'} w(T_s)\lambda_k(T_s).
\end{aligned}
$$

From the definitions and bijections established in the previous section, we also have

$$
\sum_{T_s \in \mathscr{F} \cap \tau_2} w(T_s)\lambda_k(T_s) = \sum_{T_s \in \mathscr{F}' \cap \tau_2'} w(T_s)\lambda_k(T_s)
$$

and

$$
\sum_{T_s \in \mathscr{F} \cap \tau_3} w(T_s)\lambda_k(T_s) = f(u) c_T(k; u).
$$

Hence

$$
\begin{aligned}
F_T(k; \mathbb{S}) &= \sum_{T_s \in \mathscr{F}} w(T_s)\lambda_k(T_s) \\
&= \sum_{T_s \in \mathscr{F} \cap \tau_1} w(T_s)\lambda_k(T_s) + \sum_{T_s \in \mathscr{F} \cap \tau_{1'}} w(T_s)\lambda_k(T_s) \\
&\quad + \sum_{T_s \in \mathscr{F} \cap \tau_2} w(T_s)\lambda_k(T_s) + \sum_{T_s \in \mathscr{F} \cap \tau_3} w(T_s)\lambda_k(T_s) \\
&= \sum_{T_s \in \mathscr{F}' \cap \tau_1'} w(T_s)\lambda_k(T_s) + \sum_{T_s \in \mathscr{F}' \cap \tau_2'} w(T_s)\lambda_k(T_s) + f(u) c_T(k; u). \\
&= F_{T'}(k; \mathbb{S}) + f(u) c_T(k; u).
\end{aligned}
$$

□

Similar to before, the algorithm for enumerating such subtrees immediately follows from the above theorem.

**ALGORITHM**
**Step 1.**
Initialization:

$$c_T(0;v) = \begin{cases} 1 & v \notin \mathbb{S}; \\ 0 & v \in \mathbb{S}; \end{cases}$$
$$c_T(1;v) = \begin{cases} 0 & v \notin \mathbb{S}; \\ 1 & v \in \mathbb{S}; \end{cases}$$
$$c_T(i;v) = 0, \qquad 2 \leq i \leq k,$$

for all $v \in V(T)$ and

$$N = 0.$$

**Step 2.**
Contraction:

- Choose a pendant edge $e = (u, v_0)$ with leaf $u$;
- Update $c_T(i;v)$ with $c_{T'}(i;v)$ for $0 \leq i \leq k$ for all $v \in V(T')$;
- Update $N := N + f(u)c_T(k;u)$;
- Update $T := T'$.

**Step 3.**

- If $|T| = 1$, go to Step 4;
- Otherwise, go to Step 2.

**Step 4.**
Update $N := N + f(v)c_T(k;v)$ where $v$ is the only vertex in $V(T)$.
Output $F_T(k;\mathbb{S}) = N$.

**Remark 3.** *It is easy to see that, when $\mathbb{S} = V(T)$, the above algorithm can be used to enumerate subtrees of order $k$.*

## 5. Concluding remarks

In this note we followed the simple idea of contracting pendant edges and recording information recursively, showing that various types of subtrees (with application in many different fields) can be efficiently enumerated. A more comprehensive question, namely the enumeration of subtrees of given order that contain given number of vertices from a given set (most likely the set of leaves in applications), can be solved following the same procedure. The detailed presentation, however, seems to be rather tedious and technical.

## References

[1] M. Fischermann, A. Hoffmann, D. Rautenbach, L. A. Székely, L. Volkmann, Wiener index versus maximum degree in trees, *Discrete Appl. Math.* 122 (2002), 127–137. MR1907827

[2] C. Heuberger, H. Prodinger, On $\alpha$-greedy expansions of numbers, *Advances in Applied Mathematics* 38(4) (2007), 505–525. MR2311049

[3] R. Jamison, On the average number of nodes in a subtree of a tree, *J. Combinatorial Theory Ser. B* 35 (1983), 207–223. MR0735190

[4] R. Jamison, Monotonicity of the mean order of subtrees, *J. Combinatorial Theory Ser. B* 37 (1984), 70–78. MR0762896

[5] R. Kirk, H. Wang, Largest number of subtrees of trees with a given maximum degree, *SIAM J. Discrete Mathematics* 22(3) (2008), 985–995. MR2424834

[6] B. Knudsen, Optimal multiple parsimony alignment with affine gap cost using a phylogenetic tree, *Lecture Notes in Bioinformatics* 2812, Springer Verlag, 2003, 433–446.

[7] A. Meir, J. W. Moon, On subtrees of certain families of rooted trees, *Ars Combin.* 16 (1983), 305–318. MR0737132

[8] L. A. Székely, H. Wang, On subtrees of trees, *Advances in Applied Mathematics* 34 (2005), 138–155. MR2102279

[9] L. A. Székely, H. Wang, Binary trees with the largest number of subtrees with at least one leaf, *Congr. Numer.* 177 (2005), 147–169. MR2198660

[10] L. A. Székely, H. Wang, Extremal values of ratios: distance problems vs. subtree problems in trees, *Electron. J. Combin.* 20(1) (2013), 20, paper 67. MR3040629

[11] L. A. Székely, H. Wang, Extremal values of ratios: distance problems vs. subtree problems in trees II, to appear. MR3164035

[12] A. Vince, H. Wang, The average order of a subtree of a tree, *J. Combinatorial Theory Ser. B* 100 (2010), 161–170. MR2595700

[13] S. Wagner, Correlation of graph-theoretical indices, *SIAM J. Discrete Mathematics* 21(1) (2007), 33–46. MR2299692

[14] K. Wasa, T. Uno, H. Arimura, Constant time enumeration of bounded-size subtrees in trees and its applications, preprint.

[15] H. Wiener, Structural determination of paraffin boiling point, *J. Amer. Chem. Soc.* 69 (1947), 17–20.

[16] W. Yan, Y. Yeh, Enumeration of subtrees of trees, *Theoretical Computer Science* 369 (2006), 256–268. MR2277574

[17] X.-M. Zhang, X.-D. Zhang, D. Gray, H. Wang, The number of subtrees of trees with given degree sequence, *J. Graph Theory* 73(3) (2013), 280–295. MR3062797

[18] X.-M. Zhang, X.-D. Zhang, D. Gray, H. Wang, Trees with the most subtrees – an algorithmic approach, *J. Combinatorics* 3(2) (2012), 207–224. MR2980751

Hua Wang
Department of Mathematical Sciences
Georgia Southern University
Statesboro, GA 30460
USA
*E-mail address:* hwang@georgiasouthern.edu

Shuai Yuan
Department of Mathematics
University of South Carolina
Columbia, SC 29208
USA
*E-mail address:* syuan@email.sc.edu

Received 20 September 2014