IJMMS 2004:6, 273–283
PII. S0161171204302206
http://ijmms.hindawi.com
© Hindawi Publishing Corp.

DETERMINANTAL GENERATING FUNCTIONS OF COLORED SPANNING FORESTS

GREGORY M. CONSTANTINE and MARIUS G. BULIGA

Received 24 February 2003

The color type of a spanning forest of a graph with colored edges is defined and, subsequently, it is proved that the generating function of such spanning forests is obtained as the formal expansion of a certain determinant. An analogous determinantal expansion yields the generating function of all spanning forests of a given color type that contain a specific subforest. Algorithms are described for obtaining a list of all colored spanning trees and spanning forests of any graph with colored edges based on symbolic calculation.

2000 Mathematics Subject Classification: 05C05, 05A15, 15A15.

1. Colored spanning forests. We start out by defining colored trees and forests in a graph with colored edges. The generating function of such objects is then introduced. Relying on the concept of tree-monomial matrices, we then prove that the generating function of a certain type of colored spanning forests is equal to the formal expansion of a determinant.

Let G be a graph with $n+1$ vertices denoted by $1,2,\ldots,n+1$. Denote by C a set of cardinality s, the elements of which are called *colors*. Assume that each edge of G is colored with one of the s colors in C. To edge ij, we associate the indeterminate x_{ij}. When the color c of edge ij is of the essence, we append an additional index and write x_{ijc}. Superscripts of the form $x_{ijc}^{(k)}$ are used to allow for multiple edges of the same color between two vertices. We refer to Berge [4] and Constantine [7, Chapter 4] for the basic concepts of graph theory. By a *forest*, we understand a cycle-free subgraph of G. The connected components of a forest are called *trees*. If the forest is such that each vertex of G is also a vertex of the forest, we call it a *spanning forest*. A *spanning tree* is a connected subgraph of G having n edges. (Alternatively, a spanning tree is a spanning forest having a single connected component.) If the edges of a forest carry distinct colors, the forest is said to be *colorful*. In particular, we can talk about colorful trees, colorful spanning forests, and colorful spanning trees. Furthermore, if the emphasis is on the color of the edges of a forest, we refer to the forest as a *colored forest* with an obvious terminological extension to any subgraph of G.

The generating function of a colored tree T is the monomial

$$g(T)=\prod_{ij\in T}x_{ijc}. \tag{1.1}$$

If F is a colored forest, we define the generating function of F as

$$f(F) = \prod_{T \in F} g(T), \tag{1.2}$$

the product being the overall trees in F.

If S is a set of colored forests of G, the generating function of S is defined by

$$h(S) = \sum_{F \in S} f(F). \tag{1.3}$$

2. Tree-monomial matrices.

Let $I = \{1,\dots,n\}$. Consider the indeterminates x_{ij} with $i \neq j$, where $i \in I$ and $j \in I \cup \{n+1\}$. A matrix $M = (m_{ik})$ of dimension n is called a *tree-monomial matrix* if its entries satisfy the following conditions:

$$m_{ii} = x_{ij} \quad \text{for some } j \in I \cup \{n+1\},\ j \neq i, \tag{2.1}$$

$$m_{ik} = \begin{cases} -x_{ij} & \text{if } k = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

As we will see below, the determinant of the matrix M is either a monomial associated with a spanning tree or 0, hence the name of tree-monomial matrices. We thus distinguish it from the monomial matrices defined in Berman and Plemmons [5].

Denote by $\mathbf{L}_n$ the set of tree-monomial matrices of dimension n. For $M \in \mathbf{L}_n$, we denote by $M(1)$ the matrix obtained from M by replacing all the indeterminates in M with the number 1. Observe the following lemma.

Lemma 2.1. *If $M \in \mathbf{L}_n$, then* $\det M(1)$ *is either* 0 *or* 1.

Proof. Indeed, if each row of $M(1)$ contains one 1 and one -1, the row sums of $M(1)$ are zero, and therefore $M(1)$ is singular; thus $\det M(1) = 0$. We may therefore assume that the first row of $M(1)$ has a 1 in position 1 and the rest of the entries are 0. Expanding $\det M(1)$ along the first row, we see that $\det M(1)$ is equal to the determinant of a matrix in $\mathbf{L}_{n-1}$ which by induction has determinant 0 or 1. □

Lemma 2.2. *There is a bijection between the spanning trees with $n+1$ vertices and the elements $M \in \mathbf{L}_n$ with* $\det M(1) = 1$.

Proof. Let T be a spanning tree with $n+1$ vertices labeled $1,\dots,n+1$. By a process of layered removals of vertices of degree 1, described below, we associate to T a matrix $M_T \in \mathbf{L}_n$.

Fix and never remove vertex $n+1$. Let E be the set of edges (i,j) of T such that i is a vertex of degree 1 in T, and i is not vertex $n+1$. For $(i,j) \in E$, define the ith column of M_T to be 0, with the exception of entry i which equals x_{ij} and entry j which equals $-x_{ij}$, if $j \neq n+1$; if $j = n+1$, all entries in the ith column are 0 except the ith which equals x_{ij}. Remove all vertices i with $(i,j) \in E$. Repeat the process in the resulting tree, which has strictly fewer edges. The process terminates when only vertex $n+1$ remains. At this stage, we produced a matrix $M_T \in \mathbf{L}_n$. The matrix $M_T(1)$ has a determinant equal to 1. To see this, write $M_T(1) = (a_{ij})$. We have

$$\det M_T(1) = \sum \operatorname{sgn}(\sigma) \prod_i a_{i\sigma(i)}, \tag{2.3}$$

with the sum extending over all permutations σ on n points. Except for the identity permutation, any other σ has nontrivial cycles which correspond to cycles in T. Since T is a tree, it has no cycles, thus $\prod_i a_{i\sigma(i)} = 0$. Therefore, $\det M(1) = \prod_i a_{ii} = 1$.

Conversely, let $M \in \mathbf{L}_n$ and assume that $\det M(1) = 1$. There exists i such that row i of $M(1)$ has 1 in position i and all its other entries are 0; else, the row sums of $M(1)$ would be zero, contradicting the nonsingularity. The cofactor of the ith entry in row i is a matrix $M_i(1)$, with $M_i \in \mathbf{L}_{n-1}$ and $\det M_i(1) = 1$. By induction, there is a spanning tree T_i on vertices $\{1,\ldots,n+1\} - \{i\}$ corresponding to M_i. The spanning tree T obtained from T_i by appending edge $(i,n+1)$, where i is a vertex of degree 1, corresponds to the matrix M.

It is evident that distinct M's yield distinct spanning trees, and distinct spanning trees lead to distinct M's. The correspondence we describe is therefore a bijection. (It is similar to the *marimba stick* bijection discussed in Merris [12].) This ends the proof. □

Denote by $T \leftrightarrow M_T$ the bijection we constructed. The bijection is obtained by fixing vertex $n+1$ and then constructing the matrix M_T by a process of layered removals of vertices of degree 1 (different from $n+1$) which we informally call *prunning*. We call vertex $n+1$ the *root* of tree T and say that T is *rooted* at $n+1$. Furthermore, it is clear from the construction of M_T that M_T is completely determined by its diagonal entries. If the ith diagonal entry of M_T is x_{ij}, we call edge (i,j) the *tail* of vertex i. The correspondence $M_T \leftrightarrow \{(i,j) : (i,j) \text{ is the tail of } i,\ 1 \le i \le n\} = P_T$ is therefore a bijection. It helps to interchangeably and advantageously make use of the bijections $T \leftrightarrow M_T \leftrightarrow P_T$. Clearly, instead of $n+1$, we could a priori have selected any vertex as the root in the above construction.

By our previous remarks, it follows that

$$\det(M_T) = \prod_{i=1}^{n} x_{ij} = \prod_{ij\in T} x_{ij} = g(T). \tag{2.4}$$

This leads us to the following conclusion.

Proposition 2.3. *The generating function of a spanning tree T is equal to the determinant of the associated matrix M_T. (In symbols, $g(T) = \det(M_T)(= \prod_{i=1}^{n} x_{ij})$.)*

For any set S of spanning trees, we therefore have

$$g(S) = \sum_{T\in S} \det(M_T). \tag{2.5}$$

We will see examples of sets of spanning trees and spanning forests where the sum of determinants reduces to the expansion of a single determinant.

Implementational aspects programmed in C++, *Mathematica* and *Splus*, along with issues of optimization in the presence of graph coloration, are found in Buliga [6].

3. Spanning trees of a specific type. The bijection $T \leftrightarrow P_T$ allows us to classify a spanning tree as a sequence of n possibly repeated colors $c_1 \cdots c_n$, where c_i is the color of the tail of vertex i. The induced correspondence $T \to w = c_1 \cdots c_n$ is a *function* but not necessarily a bijection. Spanning trees mapped into the same sequence w form an equivalence class which we denote by S_w; we call w the color *type* of a spanning tree in S_w. The type of a spanning tree in S_w, with $w = c_1 \cdots c_n$, is described in words by saying that vertex i has a tail of color c_i, $1 \le i \le n$.

Our next result expresses the generating function of the colored spanning trees in which vertex i has a tail of color c_i, $1 \le i \le n$, as the formal expansion of a single determinant obtained directly from the coloration of the edges of G. We first introduce some terminology.

Let H be a graph with vertices labeled $1,\dots,n+1$. We assume that all edges of H carry the same color c. The *Kirchhoff matrix* of H is a symmetric $(n+1)$-dimensional vertex-versus-vertex matrix with entries described as follows: for $i \neq j$, the (i,j)th entry is equal to $-x_{ijc}$ (in case of multiple edges, it equals $-\sum_k x_{ijc}^{(k)}$); it is equal to 0 if there are no edges present between vertices i and j. The ith diagonal entry is equal to the negative of the sum of the off-diagonal entries in row i. Delete the row corresponding to vertex $n+1$ and the column corresponding to vertex $n+1$ in the Kirchhoff matrix of H; denote the resulting matrix by $K(H)$. Call $K(H)$ the *reduced Kirchhoff matrix* of H.

The Kirchhoff matrix is sometimes called the Laplacian. Recent contributions and extensions of the matrix-tree theorem appeared in Lewin [11] and Moon [13, 14]. Operations on the graph that increase its number of spanning trees are found in Kelmans [9]. Applications to statistics are highlighted in Ouellette [15] and Constantine [8].

For a graph G with vertices labeled $1,\dots,n+1$ and each edge colored with one of s possible colors, let G_c be the subgraph with $n+1$ vertices whose edges are the edges of G of color c. Denote by $K(G_c)$ the reduced Kirchhoff matrix of G_c. Our result may now be stated as follows.

Theorem 3.1. *In a graph G with $n+1$ vertices, the generating function of the set of spanning trees, each having the property that vertex i has a tail of color c_i, is obtained as the formal determinant of the matrix whose ith column is equal to the ith column of matrix $K(G_{c_i})$, $1 \le i \le n$.*

Proof. We are interested in $g(S_w)$, where $w = c_1 \cdots c_n$. Denote by $K(S_w)$ the matrix whose ith column is equal to the ith column of $K(G_{c_i})$, $1 \le i \le n$. Use the linearity of the determinant in each of its rows (or columns) to express $\det(K(S_w))$ as a sum of determinants of tree-monomial matrices; in fact, if we use d_i to denote the number of distinct indeterminates in the ith column of $K(S_w)$, the sum contains $\prod_{i=1}^n d_i$ determinants of tree-monomial matrices. By Lemmas 2.1 and 2.2, the formal determinant of a tree-monomial matrix is either 0 or the generating function of a spanning tree of type w. Since for each i, all indeterminates x_{ijc_i}, with edge ij of color c_i, occur in the ith column on $K(S_w)$, the generating function of every spanning tree of type $w = c_1 \cdots c_n$ appears as a term in the expansion of $\det(K(S_w))$. There are no repetitions since a repetition can occur if and only if a tree-monomial matrix is repeated; but different monomial matrices (the determinats of which occur) in the expansion contain different sets of indeterminates and they are therefore distinct. This ends the proof. □

Expressed in symbols, we deduce that

$$g(S_w) = \sum_{T \in S_w} \det(M_T) = \det(K(S_w)). \tag{3.1}$$

Denote by $K(S_w;1)$ the matrix obtained from $K(S_w)$ by substituting 1 for x_{ijc} for all i, j, and c. The last equation yields $|S_w|$, the cardinality of the set of spanning trees of type w:

$$|S_w| = \det(K(S_w;1)). \tag{3.2}$$

It is self-evident that the entries of $K(S_w;1)$ can be expressed directly in terms of incidences in the graph G; the ith diagonal entry, for example, is the number of edges of color c_i incident with vertex i. To save space, we will not restate this last equation in graph-theoretical language.

Observe that whereas the reduced Kirchhoff matrix is always symmetric, it is not the same case for $K(S_w)$.

4. Spanning trees containing a specific subtree. Fix a subtree T of G with t vertices. By an eventual relabeling of vertices, we may assume that $n-t+2,\dots,n+1$ are the t vertices of T. Unless otherwise stated, whenever the context requires that a root be specified, that root will be vertex $n+1$ of the graph G. Fix a type $w = c_1 \cdots c_n$. Let $S_{w,T}$ denote the set of colored spanning trees of G of type w, each containing T as a subtree. We attempt to find $g(S_{w,T})$, the generating function of the set of such spanning trees. Denote by $K(S_{w,T})$ the matrix whose ith column is equal to the ith column of matrix $K(G_{c_i})$, $1 \le i \le n-t+1$. The vertices by which matrix $K(S_{w,T})$ is indexed are vertices not in the tree T.

Theorem 4.1. *The generating function* $g(S_{w,T})$ *is equal to the formal expansion of the product of determinants* $\det(M_T)\det(K_{w,T})$.

Proof. Let G' be the graph obtained from G by contracting the tree T to a new vertex r and preserving all adjacencies and all indeterminates attached to the edges as they appear in the original graph G. We form the color matrices G'_{c_i} by treating the new vertex r as a root. Observe that matrix $K(S_{w,T})$ has as its ith column the ith column of matrix G'_{c_i}. Theorem 3.1 yields now the generating function of the spanning trees of G' as the expansion of $\det(K(S_{w,T}))$, provided that we substitute the indeterminates x_{ijc} with a new indeterminate x_{irc} for all vertices $j \in T$. Therefore, by multiplying each term in $\det(K(S_{w,T}))$ by the generating function $g(T) = \det(M_T)$ of tree T, we obtain the generating function of all spanning trees of type w that contain T as a subtree. The main point being that a spanning tree of G of the kind we seek is obtained from a spanning tree of G', a specification of the edges of the spanning tree of G' that connect to T, with the tree T itself appended. It follows that $g(S_{w,T}) = \det(M_T)\det(K(S_{w,T}))$, as stated. This ends the proof. □

If we consider all edges colored with the same color, we obtain the following corollary.

Corollary 4.2. *The generating function of the set of all spanning trees of a graph G containing a given subtree T is equal to the product between the generating function of T and the determinant of the matrix obtained from the Kirchhoff matrix of G by deleting the rows and columns corresponding to the vertices of T.*

Furthermore, if we replace in the corollary each indeterminate x_{ij} by the number 1, we obtain a formula for the number of spanning trees of G containing a given subtree T. This numerical result was first proved in Lewin [11].

5. Rooted colored spanning forests. In this section, we assume that G is a graph with vertices labeled $1,\dots,n$. Let $F=(T_1,\dots,T_k)$ be a forest in G with connected components T_i, $1\le i\le k$. When the tree T_i is rooted at vertex v_i, we say that F is a *rooted forest* at vertices $(v_1,\dots,v_k)$. To each rooted tree T_i, with t_i vertices, we associate M_{T_i}, the $(t_i-1)\times(t_i-1)$ matrix obtained by layered removals of vertices of degree 1 while converging toward the root v_i, as described in detail in the proof of Lemma 2.2. In this case, we label the rows and columns of M_{T_i} by the vertices present in the tree T_i, except for the root v_i. To F, we now associate the direct sum of matrices M_{T_i} in the sense that we place the zeroes for all entries other than those previously defined. We denote the resulting $(n-k)\times(n-k)$ matrix by M_F. The generating function of F is $g(F)=\prod_{i=1}^{k} g(T_i)=\prod_{i=1}^{k}\det(M_{T_i})=\det(M_F)$ as is readily seen from definition (1.2) and Proposition 2.3.

We define the *type* of a colored forest $F=(T_1,\dots,T_k)$ rooted at $(v_1,\dots,v_k)$ to be $(w_1,\dots,w_k)$, where w_i is the type of the colored tree T_i. Let S be the set of colored spanning forests of G rooted at $(v_1,\dots,v_k)$ of type $(w_1,\dots,w_k)$; we will find a determinantal expression for the generating function of S. Computer algorithms for finding spanning forests in certain classes of graphs are found in Annan [1]. Extensions of such algorithms to colored forests appear in Buliga [6].

Identify k vertices $(v_1,\dots,v_k)$ in G. Denote by $\bar{G}$ the graph obtained from G by adding a new vertex $n+1$ and joining it to vertices $v_1,\dots,v_k$. Color each of the k new edges by a new color $s+1$. Associate the indeterminate x to each of the k new edges. Denote by G_{c_i} the subgraph of $\bar{G}$ on $n+1$ vertices with edges of color c_i, $1\le i\le s+1$. Let $K(G_{c_i})$ be the $n\times n$ reduced Kirchhoff matrix of the graph G_{c_i} upon deletion of the row and column corresponding to vertex $n+1$ in the Kirchhoff matrix of the graph G_{c_i}. We are now in a position to enunciate our next theorem.

Theorem 5.1. *The generating function of the set of spanning forests rooted at $(v_1,\dots,v_k)$ of a graph G with n vertices, having the property that vertex i has a tail of color c_i, is equal to x^{-k} times the formal determinant of the $n\times n$ matrix whose ith column is equal to the ith column of matrix $K(G_{c_i})$, $1\le i\le n$.*

Proof. Deletion of vertex $n+1$ from a spanning tree of $\bar{G}$ yields a spanning forest of the type we want. This association is a bijection. The result now follows by applying Theorem 3.1 to obtain the generating function of the spanning trees of $\bar{G}$ having the same type as the spanning forests in question. □

An extension of Theorem 4.1 yields the generating function of the spanning forests of G of a given type that contain a given subforest. Let K be the $n\times n$ matrix whose

ith column is equal to the ith column of matrix $K(G_{c_i})$, $1 \le i \le n$, and let U be a set of vertices in G. Denote by $K(U)$ the matrix obtained from K by deleting the rows and columns whose indices correspond to vertices in U.

Theorem 5.2. *Let $G(U)$ be the subgraph of a graph G induced by a subset U of vertices of G. If $F(G(U)) = (T_1,\dots,T_k)$ is a spanning forest of $G(U)$ with trees $\{T_i\}$ as components, the generating function of the set of spanning forests of G rooted at $(v_1,v_2,\dots,v_k)$, with $v_i \in T_i$, having the property that vertex i has a tail of color c_i and that each tree of such a spanning forest contains precisely one tree of $F(G(U))$, is equal to the generating function of $F(G(U))$ times the formal determinant of $K(U)$.*

Proof. As in the proof of Theorem 5.1, we add a new vertex $n+1$ and join it to vertices $v_1,\dots,v_k$ obtaining the graph $\bar{G}$. Analogously, we produce $\bar{G}(U)$ by adding the new vertex $n+1$ to $G(U)$. Notice again the bijection between the spanning trees of $\bar{G}$ and the spanning forests of G. We denote by $F(G(U)) \cup \{n+1\}$ the tree obtained by adding vertex $n+1$ to the forest $F(G(U))$. In view of this bijection, we see that the generating function of the spanning forests in G containing $F(G(U))$ is equal to the product between x^{-k} and the generating function of the spanning trees in $\bar{G}$ containing the tree $F(G(U)) \cup \{n+1\}$. By Theorems 4.1 and 5.1, the last expression is $x^{-k} \cdot g(F(G(U)) \cup \{n+1\}) \cdot \det K(U) = x^{-k} \cdot x^k \cdot g(F(G(U))) \cdot \det K(U) = g(F(G(U))) \cdot \det K(U)$. Here $g(H)$ denotes the generating function of the forest H. This ends the proof. □

The result represents a colored generating function version of the numerical uncolored version that appears in Lewin [11].

6. Examining some known results.

If all edges of G are of one color, then there is only one color type. All spanning trees of G are in the same class, and our main result implies that *the generating function for the spanning trees of G is equal to the formal expansion of the determinant of the reduced Kirchhoff matrix of G*. This is the well-known matrix-tree theorem due to Kirchhoff [10].

A generating function for the set of all colorful spanning trees of G was obtained by Bapat and Constantine [3] as the formal expansion of the mixed discriminant of matrices $K(G_c)$, where c ranges over a set of n colors. We refer the reader to [3, page 232] for a definition of the mixed discriminant. This result is recovered by observing that the generating function for the set of all colorful spanning trees of G is equal to

$$\sum_{w \in B} \det\left(K(S_w)\right), \tag{6.1}$$

where B is the set of all sequences consisting of n colors selected out of s available colors. It is evident that the mixed discriminant is just the sum of $\det(K(S_w))$ when w ranges over the $n!$ sequences, possible to make with n distinct colors. Moreover, the expansion of the mixed discriminant contains many formal determinantal expansions that are actually zero since S_w is empty unless spanning trees of type w exist in G. By examining the color incidences at each vertex, we obtain meaningful information on the color types that correspond to actual colored spanning trees of that type, thus avoiding redundant determinantal evaluations included in the mixed discriminant.

By selecting as color types all sequences of length n having n_i elements of color c_i out of a choice of s colors, Bapat and Constantine [3] express the generating function of such spanning trees as a mixed discriminant. The result is obtained as a special case of Theorem 3.1 by summing $\det(K(S_w))$ over all such type choices w. In general, since the subsets S_w are disjoint, the generating function of any union of classes is

$$g(\cup_w S_w) = \sum_w \det(K(S_w)). \tag{6.2}$$

The results on colorful spanning forests are recovered by observing that the same generating function is obtained by rooting the forest $F = (T_1, \dots, T_k)$ in $\prod_{j=1}^{k} t_j$ ways, with t_j being the number of vertices in tree T_j.

7. Algorithms using symbolic manipulation. In the first part of this section, we present an algorithm for obtaining the list of all spanning trees of a specific color type for a given graph G. This algorithm is then extended to obtain all spanning trees containing a given tree. At the end of the section, we generate all colorful spanning trees for the complete graph K_8 colored in a specific way (coloring by matchings).

The input to the main procedure for the first algorithm consists in the list graph (containing all edges x_{ijc} of the graph), the variable ord (representing the order of the graph), and the required color type w for the spanning trees. The symbol s represents the cardinality of the set of colors.

```
Procedure P1(graph, ord, $w$)
begin
  allcolm = GEN(graph, ord)
  $d$ = DET($w$, allcolm)
end.
```

The variable allcolm represents a list containing s color matrices (each having dimension $(\text{ord}-1)\times(\text{ord}-1)$) for the given graph. This list is generated using the procedure GEN.

```
Procedure GEN(graph, ord)
begin
  initialize all entries in allcolm with 0
  for $x_{ijc}$ ∈ graph, do
  begin
    add $x_{ijc}$ to entry $(i,i)$ in allcolm[$c$]
    if $j$! = ord, then
    begin
      subtract $x_{ijc}$ from entry $(i,j)$ in allcolm[$c$]
      subtract $x_{ijc}$ from entry $(j,i)$ in allcolm[$c$]
      add $x_{ijc}$ to entry $(j,j)$ in allcolm[$c$]
    end;
  end;
  return allcolm
end.
```

This procedure requires $O(m)$ time and $O(s \cdot \text{ord}^2)$ storage space, where m represents the number of edges of the graph G.

The procedure DET is used to form the matrix $K(S_w)$, as described in Section 3, and to compute its determinant.

```
Procedure DET($w$, allcolm)
begin
  for $c_p \in w$, do
  begin
    copy column $p$ of allcolm[$c_p$] into column $p$ of $K(S_w)$
  end;
  return the determinant of $K(S_w)$
end.
```

This procedure requires $O(\text{ord}^2)$ space. The output of this procedure is 0 (if there are no spanning trees of color type w) or a sum of monomials, each monomial representing a spanning tree of the required type.

The procedure P2 presented below determines all colored spanning trees having color type w that contain a given tree t.

```
Procedure P2(graph, ord, $w$, $t$)
begin
  allcolm = GEN(graph, ord)
  $l = \emptyset$
  for $x_{ijc} \in t$, do
  begin
    if ($i \notin l$ and $i \neq$ ord), then
      $l = l \cup \{i\}$
    if ($j \notin l$ and $j \neq$ ord), then
      $l = l \cup \{j\}$
  end;
  for $c_p \in w$, do
  begin
    copy column $p$ of allcolm[$c_p$] into column $p$ of $K(S_w)$
  end;
  obtain $K_{w,t}$ by erasing from $K(S_w)$ all rows and columns indicated by elements of $l$
  $gt = 1$
  for $x_{ijc} \in t$, do
  begin
    $gt = gt * x_{ijc}$
  end;
  return $gt * \det(K_{w,t})$
end.
```

The list l contains all vertices in the tree t that are different from the root. The elements of l determine which rows and columns are erased from the matrix $K(S_w)$.

The generating function of all spanning trees containing tree t having color type w is returned as the product between the generating function gt of t and the determinant of the new matrix $K_{w,t}$.

As an illustration of the use of these routines, consider the complete graph K_8 with edges colored by using seven colors. The coloration has the properties that all edges having color c_i $(1 \leq i \leq 7)$ cover all vertices of K_8 and are vertex disjoint. We say that K_8 is colored by matchings. Wallis [16] showed that there are six possible nonequivalent colorations by matchings for K_8. We try to obtain a decomposition of K_8 into isomorphic colorful spanning trees that are edge disjoint. In order to do this, we first find the list of all colorful spanning trees of K_8. We produce the list P of permutations on seven elements, and for each symbol $w \in P$, we find the colorful spanning trees having color type w by using the algorithm P1. Since all symbols in w are different and we go through all $w \in P$, we obtain the list C of all colorful spanning trees in the graph without repetitions.

For each spanning tree $t \in C$, we then form the adjacency matrix and find its eigenvalues. We partition the list C into equivalence classes containing trees that have the same eigenvalues. This is a necessary condition, which is also sufficient for K_8, to allow sorting of the spanning trees into isomorphism classes. The list of classes is denoted by ISO. For each equivalence class $\mathrm{ISO}[i]$, we pick an element t_1 and find the list ISO_1 with all trees from $\mathrm{ISO}[i]$ that are edge disjoint with t_1. Next, we pick an element t_2 in ISO_1 and find the list ISO_2 with all trees from ISO_1 that are edge disjoint with t_2. We continue this process until we obtain ISO_7. If $\mathrm{ISO}_7 = \{t_8\} \neq \emptyset$, then $(t_1, t_2, \ldots, t_8)$ represents a partition of K_8 into isomorphic colorful spanning trees that are edge disjoint. The method yields all possible decompositions of the edges of K_8 into isomorphic spanning trees. It turns out that there are exactly seven nonisomorphic trees for which such a decomposition is possible, and precisely one tree that is common to all six nonequivalent colorings by mathings.

Applications of such decompositions to the analysis of mytochondrial DNA data in population genetics appear in Banks et al. [2].

Acknowledgment. This work is funded in part by a Scaife Family Foundation Grant.

References

[1] J. D. Annan, *A randomised approximation algorithm for counting the number of forests in dense graphs*, Combin. Probab. Comput. **3** (1994), no. 3, 273–283.

[2] D. Banks, G. Constantine, D. A. Merriwether, and R. LaFrance, *Nonparametric inference on mtDNA mismatches*, J. Nonparametr. Statist. **11** (1999), no. 1–3, 215–232.

[3] R. B. Bapat and G. Constantine, *An enumerating function for spanning forests with color restrictions*, Linear Algebra Appl. **173** (1992), 231–237.

[4] C. Berge, *Principles of Combinatorics*, Academic Press, New York, 1971.

[5] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Applied Mathematics, vol. 9, Society for Industrial and Applied Mathematics (SIAM), Pennsylvania, 1994.

[6] M. Buliga, *On the enumeration of colorful spanning trees in a colored graph*, Ph.D. thesis, Department of Mathematics, University of Pittsburgh, Pennsylvania, 2002.

[7] G. M. Constantine, *Combinatorial Theory and Statistical Design*, Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics, John Wiley & Sons, New York, 1987.

[8] ______, *Graph complexity and the Laplacian matrix in blocked experiments*, Linear and Multilinear Algebra **28** (1990), no. 1-2, 49–56.

[9] A. K. Kelmans, *Transformations of a graph increasing its Laplacian polynomial and number of spanning trees*, European J. Combin. **18** (1997), no. 1, 35–48.

[10] G. Kirchhoff, *Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird*, Ann. Phys. Chem. **72** (1847), 497–508 (German), translated in Trans. Inst. Radio Engrs. **CT 5** (1958), 4–7.

[11] M. Lewin, *A generalization of the matrix-tree theorem*, Math. Z. **181** (1982), no. 1, 55–70.

[12] R. Merris, *An edge version of the matrix-tree theorem and the Wiener index*, Linear and Multilinear Algebra **25** (1989), no. 4, 291–296.

[13] J. W. Moon, *Some determinant expansions and the matrix-tree theorem*, Discrete Math. **124** (1994), no. 1-3, 163–171.

[14] ______, *On the adjoint of a matrix associated with trees*, Linear and Multilinear Algebra **39** (1995), no. 1-2, 191–194.

[15] D. V. Ouellette, *Schur complements and statistics*, Linear Algebra Appl. **36** (1981), 187–295.

[16] W. Wallis, *Combinatorial Designs and Configurations*, Marcel Dekker, New York, 1988.

Gregory M. Constantine: Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, USA
E-mail address: gmc@euler.math.pitt.edu

Marius G. Buliga: Department of Mathematics, University of Pittsburgh at Bradford, Bradford, PA 16701, USA
E-mail address: buliga@pitt.edu

Journal of Applied Mathematics and Decision Sciences

Special Issue on Intelligent Computational Methods for Financial Engineering

Call for Papers

As a multidisciplinary field, financial engineering is becoming increasingly important in today's economic and financial world, especially in areas such as portfolio management, asset valuation and prediction, fraud detection, and credit risk management. For example, in a credit risk context, the recently approved Basel II guidelines advise financial institutions to build comprehensible credit risk models in order to optimize their capital allocation policy. Computational methods are being intensively studied and applied to improve the quality of the financial decisions that need to be made. Until now, computational methods and models are central to the analysis of economic and financial decisions.

However, more and more researchers have found that the financial environment is not ruled by mathematical distributions or statistical models. In such situations, some attempts have also been made to develop financial engineering models using intelligent computing approaches. For example, an artificial neural network (ANN) is a nonparametric estimation technique which does not make any distributional assumptions regarding the underlying asset. Instead, ANN approach develops a model using sets of unknown parameters and lets the optimization routine seek the best fitting parameters to obtain the desired results. The main aim of this special issue is not to merely illustrate the superior performance of a new intelligent computational method, but also to demonstrate how it can be used effectively in a financial engineering environment to improve and facilitate financial decision making. In this sense, the submissions should especially address how the results of estimated computational models (e.g., ANN, support vector machines, evolutionary algorithm, and fuzzy models) can be used to develop intelligent, easy-to-use, and/or comprehensible computational systems (e.g., decision support systems, agent-based system, and web-based systems)

This special issue will include (but not be limited to) the following topics:

- **Computational methods**: artificial intelligence, neural networks, evolutionary algorithms, fuzzy inference, hybrid learning, ensemble learning, cooperative learning, multiagent learning
- **Application fields**: asset valuation and prediction, asset allocation and portfolio selection, bankruptcy prediction, fraud detection, credit risk management
- **Implementation aspects**: decision support systems, expert systems, information systems, intelligent agents, web service, monitoring, deployment, implementation

Authors should follow the Journal of Applied Mathematics and Decision Sciences manuscript format described at the journal site http://www.hindawi.com/journals/jamds/. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	December 1, 2008
First Round of Reviews	March 1, 2009
Publication Date	June 1, 2009

Guest Editors

Lean Yu, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong; yulean@amss.ac.cn

Shouyang Wang, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; sywang@amss.ac.cn

K. K. Lai, Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong; mskklai@cityu.edu.hk

Hindawi Publishing Corporation
http://www.hindawi.com