International Journal of Pure and Applied Mathematics
Volume 11 No. 1 2004, 47-61

ON MULTICOLORED ISOMORPHIC SPANNING TREE PARALLELISMS OF K_n AND A CLASSIFICATION FOR 6 AND 8 VERTICES

Marius Buliga[1 §], Gregory Constantine[2], Richard Melka[3]
[1,3]Department of Mathematics
University of Pittsburgh-Bradford
Bradford, PA 16701, USA
[1]e-mail: buliga@pitt.edu
[3]e-mail: melka@pitt.edu
[2]Department of Mathematics
University of Pittsburgh
Pittsburgh, PA 15260, USA
e-mail: gmc@euler.math.pitt.edu

Abstract: For each proper edge-coloration of K_n with $n-1$ colors we seek the number of ways that the edges can be partitioned into edge disjoint multicolored isomorphic spanning trees. A spanning tree is multicolored if all $n-1$ colors occur among its edges. A complete solution is provided for $n=6$ and 8. An algorithm for obtaining the list of all multicolored spanning trees is described and used in the case of K_n for small values of n.

AMS Subject Classification: 05C15, 05C05
Key Words: multicolored tree, orthogonal latin squares, multicolored matching

1. Parallelisms of Multicolored Isomorphic Trees

Throughout this paper K_s, denotes the complete graph on s vertices. We color the edges of K_{2n} with $2n-1$ colors by applying one color to each edge. Basic

Received: October 22, 2003

[§]Correspondence author

terminology and notation on graph theory is found in Berge [4]. The coloration is *proper* if whenever two edges intersect they carry different colors. A spanning tree is called *multicolored* if no two of its edges have the same color. Two trees are *edge disjoint* if they do not share common edges. Two (not necessarily different) graphs with colored edges are *equivalent* if there exists a bijection σ between the sets of vertices and a bijection η between the sets of colors such that (i, j) is an edge of color c if and only if $(\sigma(i), \sigma(j))$ is an edge of color $\eta(c)$. If the two graphs are in fact isomorphic, but not equivalent as colored graphs, we call the two colorations nonequivalent. In case of K_{2n}, if a proper edge-coloration of K_{2n} exists such that the edges can be partitioned into edge disjoint isomorphic multicolored spanning trees we obtain a *multicolored isomorphic tree parallelism* (MITP, for short) for K_{2n}. Our general interest is in finding as much information as possible about the MITP's for all nonequivalent proper colorations of K_{2n}. The general problem proves difficult even with lesser restrictions. In their paper (see [6, p. 310]), Brualdi and Hollingsworth posed the problem of producing such a decomposition without requiring that the spanning trees be isomorphic. They were able to produce two such multicolored spanning trees for K_{2n} for any proper coloration, and conjectured that n such spanning trees exist. In this paper we impose the additional constraint that all spanning trees in question are isomorphic, discuss some aspects of the general problem, and produce an exhaustive analysis for K_6 and K_8.

When no coloring is involved, it is well known that the edges of K_{2n} can be partitioned into isomorphic spanning trees (paths, for example). Each of these spanning trees can easily be made multicolored, but the resulting edge coloration of K_{2n} usually fails to be proper; indeed, there exist proper colorations of K_8 that do not contain any multicolored paths. A complete description can be found in Section 3. Such a partition of the edges of K_m can be viewed as a parallelism as defined in Cameron [8] with an additional property due to color. Specifically, finding a partition as described above corresponds to an arrangement of the edges of K_{2n} into an array of $2n - 1$ rows and n columns such that each row contains all edges of some color (these edges form a perfect matching due to the fact that the graph is properly colored) and the edges in each column form a (necessarily multicolored) spanning tree, the isomorphism type of which does not change from column to column. We ask, therefore, for a double parallelism of K_{2n}, one present in the rows of the array (perfect matchings) and the other in the columns that consist of edge disjoint spanning trees of a fixed isomorphism type.

The generating function of the multicolored spanning trees in any edge colored graph can be expressed as a sum of a formal discriminant; cf. Bapat and Constantine [3]. Algorithms for finding multicolored spanning trees are discussed in Broersma and Li [5]. We apply the algorithm written by Buliga [7] to obtain tree parallelisms for complete graphs on a small number of vertices. A description of the general algorithm appears in the last part of the paper.

We mention some uses of parallelisms of complete graphs that appear in the literature. It is known that every minimum biclique decomposition of K_n has a multicolored spanning tree. Liu and Schwenk, [11], study the number of such multicolored spanning trees in such decompositions, and find the maximum and minimum among all possible decompositions. In particular, they determine the extreme numbers of multicolored spanning trees among all acyclic decompositions.

An application of parallelisms of complete designs to the analysis of population genetics data is found in Banks [1]. We briefly describe the main idea. Mitochondrial DNA pairwise mismatch distributions have been used to make inferences about paleodemographic events such as population expansions, bottlenecks, and the rate of population growth. Sometimes the mismatch histogram shows multiple modes. These modes count the number of genetically distinct subpopulations (plus additional modes between these), and the intermode distance gives information about the amount of heterogeneity. In practice, in order to save financial resources, biologists simply take a random sample $x_1, \ldots, x_n$ of n elements from the population and calculate all $\binom{n}{2}$ (mismatch) genetic distances $\{d(x_i, x_j) : i \neq j\}$. This is the mismatch data with which one works. Distances are associated to edges of K_n. Distances from the same element of the population are clearly not independently distributed. This introduces biases in the analysis of data. Partitioning of these distances in accordance to parallelisms of K_n organize the data into independent random smples to which known statistical analyses can then be applied. We refer to Banks [1] for details.

Discussion of colored matchings and design parallelisms to parallel computing appear in Harari [9].

Section 2 introduces the color type of a colored spanning tree. It is then proved that the set of spanning trees of a given color type can be obtained as the expansion of a formal determinant. This refines the result of Bapat and Constantine [3] and offers a more detailed way of examining the set of multicolored spanning trees in any graph. The result forms the basis for the algorithm found in Buliga [7]. Section 3 adapts the algorithm to finding all multicolored tree parallelisms for all nonequivalent proper colorations of K_8 and K_6. We find only one solution for K_6 and observe that, in the case of K_8, there are, overall, only 7 nonisomorphic spanning trees that provide solutions; these appear in Figure 1 below. Section 4 provides a listing of the algorithm used to derive these results.

2. Colored Spanning Trees

Let G be a graph of order $n+1$ with edges colored with colors from the set $C = \{c_1, \ldots, c_s\}$. An indeterminate x_{ljc_i} is associated to each edge (l, j) having color c_i. The generating function of a tree T is defined by $g(T) = \prod_{(l,j)\in T} x_{ljc_i}$. In a similar way, the generating function of a set of trees S is $g(S) = \sum_{T\in S} g(T)$.

For an arbitrary color c_i, G_{c_i} represents the subgraph containing all edges of G having color c_i. The Kirchoff matrix of G_{c_i} is a symmetric $(n+1)$ dimensional vertex-versus-vertex matrix with the entries k_{lj} defined by:

$$k_{lj} = \begin{cases} -\sum_k x_{ljc_i}, & \text{if } l \neq j, \text{ and } x_{ljc_i} \text{ edge of color } c_i \text{ between } l \text{ and } j, \\ 0, & \text{if } l \neq j, \text{ and there are no edges between } l \text{ and } j, \\ -\sum_{m\neq l} k_{lm}, & \text{if } l = j. \end{cases}$$

By erasing the last row and last column from this matrix we obtain the reduced Kirchoff matrix $K(G_{c_i})$.

A generating function for the multicolored spanning trees of a graph was provided in Bapat and Constantine [3]. The authors defined the *mixed discriminant* of n matrices $A^k = (a^k_{ij})$, $k \in \{1, \ldots, n\}$, $i, j \in \{1, \ldots, n\}$ as:

$$D(A^1, \ldots, A^n) = \frac{1}{n!} \sum_{\sigma\in S_n} \begin{vmatrix} a_{11}^{\sigma(1)} & \ldots & a_{1n}^{\sigma(n)} \\ \vdots & \ldots & \vdots \\ a_{n1}^{\sigma(1)} & \ldots & a_{nn}^{\sigma(n)} \end{vmatrix},$$

where S_n is the set of permutations on n elements. In order to obtain the term of the sum corresponding to σ, we take the first column from $A^{\sigma(1)}$, the second column from $A^{\sigma(2)}$, and continues until lastly taking the n-th column from $A^{\sigma(n)}$ and then computes the determinant of this matrix. It was shown that if n colors are used to color the edges of G, then the formal expansion of $n!$ times the mixed discriminant of $K(G_{c_1}), \ldots, K(G_{c_n})$ produces the list without repetitions of all multicolored spanning trees of the graph.

For algorithmic purposes it is more helpful to classify the colored spanning trees of a graph by their color type. A colored tree is a tree with colored edges.

We associate to each colored tree a color type by a pruning process. Supposing that tree T has n edges, we form a sequence w of length n containing all the colors of the edges of T in the following way: first we fix vertex $n+1$ as the root of T and consider the set V_1 containing all vertices of degree one in T, different from the root. We put on position i in w the symbol c_j if the edge that links vertex i has color c_j. We perform this operation for all vertices in V_1. Next, we eliminate from T the vertices in V_1 and the corresponding edges, obtaining a new tree T_2. We form again the set V_2 of all vertices of degree one in T_2, different from the root and continue completing w and pruning the tree. We stop when we obtain a tree that contains only the root $n+1$. The sequence w is the *color type* of the tree.

We will present in this section a way of determining all spanning trees of a graph having a certain color type. The notion of *monomial marices* and some of their properties are first introduced.

An n-dimensional matrix M is called monomial matrix if the entries m_{ik} satisfy the relations:

$$\begin{cases} m_{ll} = x_{ljc_i}, \text{ where } j \in \{1,2,\ldots,n,n+1\}, \ j \neq l\,, \\ m_{lk} = -x_{ljc_i}, \text{ if } k = j \neq l\,, \\ m_{lk} = 0, \text{ if } k \neq l, \ k \neq j\,. \end{cases} .$$

If we replace in M all indeterminates by 1, a new matrix $M(1)$ is obtained. It is easy to notice that for any monomial matrix M, the determinant of $M(1)$ is 0 or 1. It can be proved by induction that there is a bijection between the set of n dimensional matrices M, with $\det M(1) = 1$ and the set of spanning trees that have $n+1$ vertices.

With respect to colored trees of a given color type we give the following theorem.

Theorem 2.1. *The generating function of the set of spanning trees of a graph G with $n+1$ vertices having the color type $w = c_1 \ldots c_n$, can be expressed as the formal determinant of the matrix whose i-th row is equal to the i-th row of the reduced Kirchoff matrix of color c_i, for $1 \leq i \leq n$.*

Proof. Let S_w be the set of all colored spanning trees having the color type

$$w = c_1 \ldots c_n.$$

For each color c_i that appears in w, we consider the subgraph G_{c_i} containing the edges of G having color c_i.

The matrix $K(S_w)$ is formed by taking the i-th row of the reduced Kirchoff matrix $K(G_{c_i})$ and placing it as the i-th row of $K(S_w)$, for all $1 \leq i \leq n$.

By using the linearity of a determinant in its rows, we express $\det(S_w)$ as a sum of determinants of monomial matrices. According to the properties of monomial matrices, the determinant of each matrix in the sum is equal to 0, or to the product of the elements on the main diagonal, which is the generating function for a colored spanning tree having color type w. Since for each i all indeterminates x_{ljc_i} with edge (l,j) having color c_i occur in the i-th row of $K(S_w)$, the generating function of every spanning tree of type $w = c_1 \dots c_n$ appears as a term in the expansion of $\det K(S_w)$. We know that to each spanning tree T it corresponds a unique monomial matrix M_T. Since different monomial matrices that appear in the linear combination of $K(S_w)$ contain different sets of indeterminates, they are all distinct. Hence, the generating function $g(T)$ of each spanning tree $T \in S_w$ appears only once in the formal expansion of $\det K(S_w)$. We obtain that

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

This ends the proof of the theorem. □

We use Theorem 2.1 to produce partitions of K_{2n} into isomorphic multicolored spanning trees. In order to do this, we find first the list of all multicolored spanning trees in K_{2n}. The complete graph is entered as a list of edges containing symbols having the form x_{lji}. We produce the list P of permutations on n elements, and for each symbol $w \in P$ we form the matrix $K(S_w)$ and compute its determinant. According to Theorem 2.1, each determinant contains all spanning trees with color type w without repetitions. Since all symbols in w are different and we go trough all $w \in P$, we obtain the list C containing all multicolored trees in the graph without repetitions.

For each spanning tree $t \in C$ we 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 $n = 3$ and 4, to allow sorting of the spanning trees into isomorphism classes. The list with these equivalence classes is denoted by ISO. For each equivalence class $ISO[i]$ we pick an element t_1 and find the list ISO_1 with all trees from $ISO[i]$ 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 disjoint with t_2. We continue this process until we get ISO_{n-1}. If $ISO_{n-1} = \{t_n\} \neq \emptyset$, then $(t_1, t_2, \dots, t_n)$ represents an isomorphic partition of K_{2n} into isomorphic multicolored spanning trees.

In the next section we present complete results for K_6 and K_8. Section 4 explicitly describes the algorithm that implements the steps presented above.

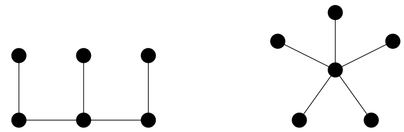

Figure 1: Isomorphism classes of spanning trees for K_6

3. Multicolored Isomorphic Tree Parallelisms for K_6 and K_8

In this section, we present all MITP's for K_6 and K_8. For K_6, there exists only one proper edge coloration and is given in the following table:

12	35	46
13	24	56
14	25	36
15	26	34
16	23	45

There are sixty-six multicolored spanning trees. However, this coloring admits only two isomorphism classes of spanning trees and they are given in the following table:

However, only the first of these two classes, the non-star, induces a partition of the edges of K_6 into MITP's:

MITP's for K_6

12	35	46
56	24	13
36	25	14
34	15	26
23	16	45

Note that each row of this table is a permutation of the corresponding row in the edge coloring of K_6.

For K_8, on the other hand, there are six nonequivalent proper edge colorations. They are:

Coloring 1 Coloring 2 Coloring 3

12	34	56	78
13	24	57	68
14	23	58	67
15	26	37	48
16	25	38	47
17	28	35	46
18	27	36	45

12	34	56	78
13	24	57	68
14	23	58	67
15	26	37	48
16	25	38	47
17	28	36	45
18	27	35	46

12	34	56	78
13	24	57	68
14	23	58	67
15	26	38	47
16	27	35	48
17	28	36	45
18	25	37	46

Coloring 4

12	34	56	78
13	24	57	68
14	23	58	67
15	27	38	46
16	28	37	45
17	25	36	48
18	26	35	47

Coloring 5

12	34	56	78
13	24	57	68
14	25	38	67
15	27	36	48
16	28	37	45
17	23	46	58
18	26	35	47

Coloring 6

12	34	56	78
13	25	47	68
14	26	38	57
15	27	36	48
16	28	37	45
17	23	46	58
18	24	35	67

Although these colorings admit several isomorphisms classes of spanning trees only seven of them induce a partition of the edges of K_8 into MITP's:

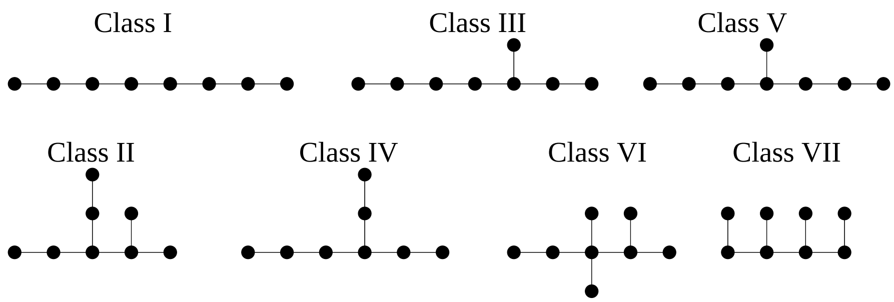

Figure 2: Isomorphism classes of spanning trees for K_8

We briefly summarize our findings: For Coloring 1, there are 2304 multicolored spanning trees. Coloring 1 admits five isomorphism classes of spanning trees and Class III provides a MITP for K_8. For Coloring 2, there are 2304 multicolored spanning trees. Coloring 2 admits seventeen isomorphism classes of spanning trees and Classes I, III, IV, and VI provide MITP's for K_8, and these are illustrated below. For Coloring 3, there are 2304 multicolored spanning trees. Coloring 3 admits eighteen isomorphism classes of spanning trees and Classes I, III, IV, VI, and VII provide MITP's for K_8. For Coloring 4, there are 2304 multicolored spanning trees. Coloring 4 admits seventeen isomorphism classes of spanning trees and Classes II, III, IV, and VI provide MITP's for K_8. For Coloring 5, there are 2312 multicolored spanning trees. Coloring 5 admits twenty isomorphism classes of spanning trees and all classes provide MITP's for K_8 in this case. Finally, for Coloring 6, there are 2318 multicolored spanning trees. Coloring 6 admits eighteen isomorphism classes of spanning trees and Classes III and V provide MITP's for K_8. It is interesting to note that the number of multicolored spanning trees is approximately the same in each of the colorings as well as the number of isomorphism classes with exception of the first. This stands in some contrast to the sizes of the automorphism groups for these nonequivalent colorations, the cardinality of which show considerable variance, cf. Wallis [12].

We now exhibit the MITP's for K_8 induced by Coloring 2:

MITP's for K_8 (Coloring 2)

Class I

12	56	34	78
24	13	68	57
67	23	58	14
48	15	37	26
38	47	16	25
17	28	45	36
35	46	27	18

Class III

12	78	34	56
24	13	57	68
67	58	14	23
48	37	15	26
38	25	16	47
36	45	28	17
35	46	27	18

Class IV

12	34	56	78
24	13	57	68
58	67	14	23
37	48	26	15
16	25	38	47
45	17	28	36
27	35	46	18

Class VI

12	34	78	56
24	13	68	57
23	67	58	14
48	37	15	26
25	38	47	16
17	45	28	36
46	27	35	18

4. Algorithm for Generating Colored Trees

In this section we present an algorithm that illustrates the result obtained in Section 2 which was then applied to obtain results in Section 3. The software package used to implement this algorithm was *Mathematica 4.0*.

It is known that in general the complexity of an algorithm for finding all spanning trees in a complete graph is exponential since the number of objects that have to be enumerated is exponential. The first part of the algorithm consists in finding all multicolored spanning trees in K_{2n}. We found that the complexity of this part is $O(((2n-1)!)^2)$.

Algorithms for finding colred paths in an arbitrary colored graph G are discussed in Li and Broersma [10]. They analyze the complexity of finding a path of G from a given vertex s to another given vertex t with as few different colors and of finding one with as many colors as possible. It was shown that the first problem is polynomial-time solvable, and that the second problem is NP-hard. A similar problem is discussed in Broersma and Li [5] for spanning trees.

We present first the main procedure. The input to this procedure consists in the list *graph* (containing all edges x_{lji} of K_{2n}) and the variable *ord* (representing the order of the graph). In our case $ord = 2n$.

Procedure PARALLELISMS(*graph, ord***)**
begin
  $allcolm = \textbf{GEN}(graph, ord)$
  generate permutations list P on $(ord - 1)$ elements
  $C = \emptyset$
  for $w \in P$ **do**
  begin
    $d = \textbf{SIMPLEDET}(w, allcolm)$
    if $d! ='' 0''$ **then**
    begin
      $tree = \textbf{CONVERT}(d)$
      $C = C \cup tree$
    end;
  end;
  $iso = \textbf{ISO\_CLASSES}(C)$
  for $i = 1$ to $length(iso)$ **do**
    $\textbf{ISOPART}(iso[i])$
end.

Mathematica provides the commands *Permutations* for finding the list P of permutations and *Length* for obtaining the length of a list. We start by generating the list *allcolm* containing the reduced Kirchoff matrices for each of the $(ord - 1)$ colors. This part is implemented by the procedure *GEN*.

Procedure GEN(*graph, ord***)**
begin
  initialize entries in all color matrices with 0
  for $x_{lji} \in graph$ **do**
  begin
    add x_{lji} to entry (l, l) in color matrix i
    if $j! = ord$ **then**
    begin
      subtract x_{lji} from entry (l, j) in color matrix i
      subtract x_{lji} from entry (j, l) in color matrix i

```
        add x_lji to entry (j, j) in color matrix i
      end;
    end;
  output all color matrices
end.
```

This procedure requires $O(n^2)$ time and $O(n^3)$ space for the complete graph K_{2n}.

For each color type w from the list of permutations P we find the spanning trees with color type w by computing the determinant of a monomial matrix. We use the procedure *SIMPLEDET* for this purpose.

Procedure SIMPLEDET(*type, allcolm***)**

```
begin
  for c_i ∈ type do
  begin
    copy row i of allcolm[c_i] into row i of K(S_w)
  end;
  output det(K(S_w)) as a string
end.
```

This procedure requires $O(n^2)$ space. The output of this procedure is a string that contains "0" or a monomial representing a spanning tree with color type w (since we compute the determinant of a monomial matrix). A possible output of this procedure for K_6 is "x_{121}, x_{232}, x_{343}, x_{564}, x_{365}".

Next, we convert this string to a list of edges using the procedure *CONVERT*.

Procedure CONVERT(*d***)**

```
begin
  edge =""; tree = ∅;
  for comp ∈ d do
  begin
    if comp ==" " then
    begin
      tree = tree ∪ expression(edge)
      edge =""
      continue;
    end;
    append comp to edge
  end;
  tree = tree ∪ expression(edge)
  output tree
```

```
end.
```

This procedure requires $O(n)$ time and space. The spanning tree for the previous example is converted to the list $\{x_{121}, x_{232}, x_{343}, x_{546}, x_{365}\}$. We obtain in this way the list C containing all multicolored spanning trees of K_{2n}.

We use next the procedure *ISO_CLASSES* to partition C into equivalence classes containing spanning trees that are isomorphic.

```
    Procedure ISO_CLASSES(C)
begin
  iso = {{C[1]}}
  for t ∈ C do
  begin
    form adjacency matrix A(t)
    for i = 1 to length(iso) do
    begin
      if eigenvalues(A(t)) == eigenvalues(iso[i][1]) then
        iso[i] = iso[i] ∪ t
      else
        iso = iso ∪ {t}
    end;
  end;
  output iso
end.
```

For each equivalence class $iso[i]$ we try to find a partition of K_{2n} into isomorphic multicolored spanning trees by using the procedure *ISOPART*. We present this procedure below for K_6:

```
    Procedure ISOPART(isolist)
begin
  for t ∈ isolist do
  begin
    iso1 = list with trees in isolist edge disjoint with t
    for t1 ∈ iso1 do
    begin
      iso2 = list with trees in iso1 edge disjoint with t1
      if iso2 ≠ ∅ then
        begin
            output (t, t1, iso2[1])
            return
        end;
    end;
```

end;
end.

For the general case K_{2n}, we would need to use $n-1$ **for** loops.

References

[1] D. Banks, G. Constantine, A. Merriwether, R. LaFrance, Nonparametric inference on mtDNA mismatches, *Journal of Nonparametric Statistics*, **11** (1999), 215-232.

[2] R.B. Bapat, Mixed discriminants and spanning trees, *Sankhyā*, Ser. A, **54** (1992), 49-55.

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

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

[5] H. Broersma, X. Li, Spanning trees with many of few colors in edge-colored graphs, *Discuss. Math. Graph Theory*, **17**, No. 2 (1997), 259-269.

[6] R.A. Brualdi, S. Hollingsworth, Multicolored spanning trees in complete graphs, *J. Comb. Th.*, Series B, **68** (1996), 310-313.

[7] M. Buliga, *On the Enumeration of Colored Spanning Trees in a Graph*, Doctoral Dissertation, Department of Mathematics, University of Pittsburgh (2002).

[8] P.J. Cameron, *Parallelisms of Complete Designs,* London Math. Soc. Lecture Notes Series, Cambridge Univ. Press., **23** (1976).

[9] F. Harari, Parallel concepts in graph theory, *Math. Comput. Modelling*, **18**, No. 7 (1993), 101-105.

[10] X. Li, S. Zhang, H.J. Broersma, Directed paths with few or many colors in colored directed graphs, *Memorandum No. 1543*, University of Twente (2000).

[11] J. Liu, A.J. Schwenk, The number of multicolored trees in acyclic multiclique decompositions, *Congr. Numer.*, **81** (1991), 129-142.

[12] W.D. Wallis, *Combinatorial Designs*, Dekker, New York (1988).