2010

Eigenvalue assignments and the two largest multiplicities in a Hermitian matrix whose graph is a tree

Charles R. Johnson
William & Mary, crjohn@WM.EDU

Christopher Jordan-Squire

David A. Sher

Recommended Citation

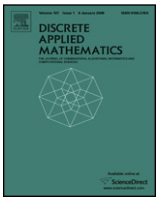
Johnson, Charles R.; Jordan-Squire, Christopher; and Sher, David A., Eigenvalue assignments and the two largest multiplicities in a Hermitian matrix whose graph is a tree (2010). *Discrete Applied Mathematics*, 158(6), 681-691.
10.1016/j.dam.2009.11.009

Discrete Applied Mathematics 158 (2010) 681–691

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam



Eigenvalue assignments and the two largest multiplicities in a Hermitian matrix whose graph is a tree[☆]

Charles R. Johnson[a,*], Christopher Jordan-Squire[b,1], David A. Sher[c,2]

[a] College of William and Mary, United States
[b] Swarthmore College, United States
[c] Johns Hopkins University, United States

ARTICLE INFO

Article history:
Received 26 July 2007
Received in revised form 17 November 2009
Accepted 24 November 2009
Available online 22 December 2009

Keywords:
Eigenvalue multiplicities
Symmetric matrices
Trees
Two largest multiplicities

ABSTRACT

Among the possible multiplicity lists for the eigenvalues of Hermitian matrices whose graph is a tree we focus upon M_2, the maximum value of the sum of the two largest multiplicities. The corresponding M_1 is already understood. The notion of assignment (of eigenvalues to subtrees) is formalized and applied. Using these ideas, simple upper and lower bounds are given for M_2 (in terms of simple graph theoretic parameters), cases of equality are indicated, and a combinatorial algorithm is given to compute M_2 precisely. In the process, several techniques are developed that likely have more general uses.

1. Introduction

Let T be a tree on n vertices. By $\mathcal{S}(T)$ we mean the collection of all n-by-n real symmetric (equivalently, complex Hermitian) matrices whose graph is T. No restriction is placed upon the diagonal entries of matrices in $\mathcal{S}(T)$, except that they are real. We are interested in the possible eigenvalue multiplicity lists of matrices in $\mathcal{S}(T)$, and their possible spectra.

For convenience, if $A \in \mathcal{S}(T)$, we place in descending order the *multiplicities* of the eigenvalues of A, irrespective of the numerical order of the eigenvalues, and refer to such a list of multiplicities as *unordered multiplicities*.

Let $\mathcal{L}(T)$ denote the set of all lists $m : m_1 \geq m_2 \geq \cdots \geq m_k$ such that $m_1, \ldots, m_k$ is the list of unordered multiplicities for some $A \in \mathcal{S}(T)$. To eliminate possible confusion, such as when multiple matrices are being discussed, we sometimes use $m_i(A)$ to refer to the ith largest multiplicity among the eigenvalues of A. We also denote $M_j(T) = \max_{A\in\mathcal{S}(T)}[m_1(A) + \cdots + m_j(A)]$. Several general facts are known about $\mathcal{L}(T)$:

(1) The list $(1, 1, \ldots, 1)$ consisting of one n times occurs for any T and is the only multiplicity list for a path. Also, the path is the only graph for which this is the only multiplicity list.
(2) $M_1(T)$ is equal to the *path cover number* $P(T)$, the smallest number of nonintersecting induced paths of T that cover all the vertices of T; this is the same as $\max(p - q)$ in which p is the number of paths remaining when q vertices have been removed from T in such a way as to leave only induced paths [2].

[☆] This work was supported in part by National Science Foundation grant DMS-0353510.

[*] Corresponding author. Tel.: +1 757 221 2014; fax: +1 757 221 7400.
E-mail addresses: crjohnso@math.wm.edu (C.R. Johnson), cjordan1@u.washington.edu (C. Jordan-Squire), dsher@stanford.edu (D.A. Sher).

[1] Current address: Department of Mathematics, University of Washington, Box 354350, Seattle, WA 98195, United States.

[2] Current address: Department of Mathematics, Stanford University, 380 Serra Mall, Stanford, CA 94305, United States.

0166-218X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2009.11.009

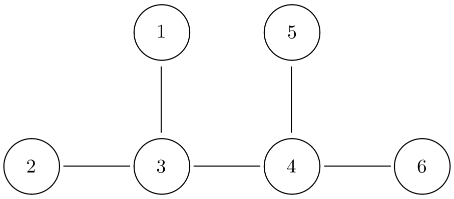


Fig. 1. An example of assignment.

(3) For each $m = (m_1, \ldots, m_k) \in \mathcal{L}(T)$, k (the number of distinct eigenvalues) is at least the diameter of T (measured in terms of vertices) [3].

(4) And, in each $m \in \mathcal{L}(T)$, there are at least two 1's [6].

Our primary purpose here is to discuss, give simple tight bounds for, and give a method for calculating $M_2(T)$. In the process we formalize and learn much about the notion of an *assignment* (see below) of eigenvalues for $A \in \mathcal{S}(T)$, indicate cases of equality in our bounds and see that $M_2(T)$ is related to several simpler characteristics of T.

A key fact in understanding $\mathcal{L}(T)$ (though this was not its original purpose) is a theorem due to Parter [8], as refined by Wiener [10] and more fully in [6]. To state it we introduce notation. If $A \in \mathcal{S}(T)$ and v is a vertex of T then $A(v)$ denotes the principal submatrix of A resulting from deleting row and column associated with v (i.e. $A(v) \in \mathcal{S}(T - v)$), and $m_A(\lambda)$ denotes the multiplicity of eigenvalue λ of matrix A. Parter's Theorem indicates that if $A \in \mathcal{S}(T)$ and $m_A(\lambda) \geq 2$, then there is at least one vertex v of T, of degree at least 3, such that $m_{A(v)}(\lambda) = m_A(\lambda) + 1$. Moreover, v may be chosen so that λ is an eigenvalue of at least three principal submatrices of A associated with branches of T at v. For this reason, we refer to any vertex v of degree greater or equal to 3 as a *high-degree vertex*, or HDV. So, perhaps counterintuitively, the multiplicity of a multiple eigenvalue actually increases in some proper principal submatrices. A *Parter vertex* is a vertex for which the eigenvalue multiplicity is positive and increases. Note that Parter's theorem guarantees the existence of at least one Parter HDV for any multiple eigenvalue. If the principal submatrix of A associated with some branch at v again has λ as a multiple eigenvalue, this theorem may again be applied to that branch. Parter vertices for λ may be removed in this fashion until (fully) fragmenting T into many subtrees in which λ occurs as an eigenvalue in each subtree at most once. Such a set of Parter vertices is called a *fully fragmented Parter set* for λ, and it is known that each successive Parter vertex is also Parter for A and λ in the original tree. When a matrix A is understood, we often, informally, refer to λ as an eigenvalue associated with a subtree.

Clearly, Parter's theorem severely limits the possible lists in $\mathcal{L}(T)$ [6]. Another limitation is the *interlacing inequalities* for principal submatrices of a Hermitian matrix [1]. Not only do they imply that $\left|m_{A(v)}(\lambda) - m_A(\lambda)\right| \leq 1$ and limit the possibilities for multiple eigenvalues to share a Parter vertex, etc., but they may more subtly limit possible lists by constraining the numerical order of the eigenvalues. This topic will be explored further in the next section. The first author has long conjectured that Parter's theorem and interlacing are the only limitations upon $\mathcal{L}(T)$, and some of the work herein is a step toward verifying this.

A quick remark on notation: throughout this paper, if A is a set or collection, then $|A|$ denotes the cardinality of A. Similarly, if A is a graph then $|A|$ is the number of vertices in A. If V is a set of vertices and A is a graph then $V \cap A$ denotes the set of vertices in both V and A. Additionally, if A is a tree we let $\mathscr{P}(A)$ denote the collection of all subtrees of A, including A, rather than the power set of the vertices in A.

2. Assignments

Suppose that a list m in $\mathcal{L}(T)$ contains a multiplicity greater than one. Then Parter's theorem implies m must have an eigenvalue distributed, via the fragmenting process described in the previous section, amongst various subtrees of T.

For example, for T as in Fig. 1 the multiplicity list $(2, 2, 1, 1)$ does occur. Vertices 3 and 4 are two possible Parter vertices for the two multiple eigenvalues. If α and β are the two multiple eigenvalues, then vertex 3 must be Parter for α, say, and only α, while 4 must be Parter for β and only β. This implies that for there to be an $A \in \mathcal{S}(T)$ with the indicated multiplicity list, we must have $m_{A(3)}(\alpha) = 3$ and $m_{A(4)}(\beta) = 3$, which imply that α must be an eigenvalue of $A[\{1\}]$, $A[\{2\}]$ and $A[\{4, 5, 6\}]$ and β an eigenvalue of $A[\{5\}]$, $A[\{6\}]$ and $A[\{1, 2, 3\}]$. Here, $A[J]$, $J \subset \{1, \ldots, n\}$, denotes the principal submatrix of A lying in rows and columns J. This distribution of eigenvalues among subtrees is possible (as is known and easily seen), and is the only way to attain the list $(2, 2, 1, 1)$. The key point is that Parter's Theorem requires that if the list is to occur then the multiple eigenvalues must appear in certain subtrees. The above distribution of eigenvalues to subtrees is an example of an assignment. The formal definition follows.

Definition 2.1 (*Assignment*). Let T be a tree on n vertices and let

$$\left(p_1, p_2, \ldots, p_k, 1^{n - \sum_{i=1}^{k} p_i}\right)$$

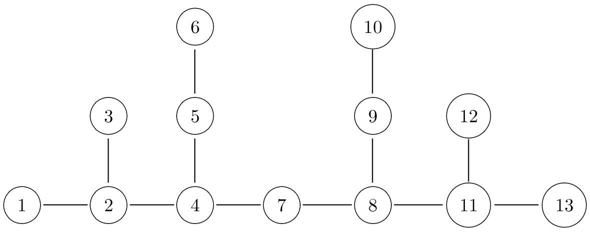


Fig. 2. A problem with interlacing.

be a non-increasing list of positive integers, with $\sum_{i=1}^{k} p_i \le n$. The notation 1^l denotes that the last l entries of the list are 1. These will be the desired eigenvalue multiplicities. Note that some of the p_i's may be 1. Then, an *assignment* $\mathcal{A}$ is a collection $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ of k collections $\mathcal{A}_i$ of subtrees of T, corresponding to eigenvalues with multiplicities $m_i(A)$, with the following properties.

(1) (Specification of Parter vertices) For each i, there exists a set V_i of vertices of T such that
   (1a) Each subtree in $\mathcal{A}_i$ is a connected component of $T - V_i$.
   (1b) $|\mathcal{A}_i| = p_i + |V_i|$.
   (1c) For each vertex $v \in V_i$, there exists a vertex x adjacent to v such that x is in one of the subtrees in $\mathcal{A}_i$.

(2) (No overloading) We require that no subtree S of T is assigned more than $|S|$ eigenvalues; define $c_i(S) = |\mathcal{A}_i \cap \mathcal{P}(S)| - |V_i \cap S|$, the difference between the number of subtrees contained in S and the number of Parter vertices in S for the ith multiplicity. Then we require that $\sum_{i=1}^{k} \max(0, c_i(S)) \le |S|$ for each $S \in \mathcal{P}(T)$. If this condition is violated at any subtree, then that subtree is said to be *overloaded*.

We also refer to the ith eigenvalue as being "assigned" to each subtree in $\mathcal{A}_i$.

The usage of assignments in practice is simpler than the definition suggests. In the example at the beginning of this section, the formal assignment of the eigenvalues is the two collections of subtrees $\mathcal{A}_1 = \{\{1\}, \{2\}, \{4, 5, 6\}\}$ for α and $\mathcal{A}_2 = \{\{1, 2, 3\}, \{5\}, \{6\}\}$ for β, with Parter vertices $V_1 = \{3\}$ and $V_2 = \{4\}$.

We will also use the following weaker variations of an assignment. An *assignment candidate* is a collection of vertices and components satisfying condition (1), but not necessarily (2). Similarly, a *near-assignment* is a collection of vertices and components satisfying conditions (1a), (1b), and (2), but not necessarily (1c). We also define a *near-assignment candidate* to be a similar collection satisfying (1a) and (1b) but not necessarily (1c) or (2).

We call an assignment $\mathcal{A}$ for a tree T *realizable* if there exists a matrix $B \in \mathcal{S}(T)$ with multiplicity list $(p_1, p_2, \ldots, p_k, 1^{n-\sum_{i=1}^{k} p_i})$ and eigenvalues $(s_1, s_2, \ldots, s_k)$ corresponding to the p_i, such that, for each i between 1 and k:

(1) For each subtree R of T in $\mathcal{A}_i$, s_i is a multiplicity 1 eigenvalue associated with R. Also, for each connected component Q of $T - V_i$ that is not in $\mathcal{A}_i$, s_i is not an eigenvalue of $B[Q]$ (i.e. the submatrix of B corresponding to vertices of Q).
(2) For each vertex c in V_i, c is Parter for s_i.
(3) All eigenvalues of B other than the s_i have multiplicity 1.

In this case, we also call the multiplicity list $(p_1, p_2, \ldots, p_k, 1^{n-\sum_{i=1}^{k} p_i})$ *realizable*.

By Parter's theorem, for any $A \in \mathcal{S}(T)$, T a tree, and any $\lambda \in \sigma(A)$, $m_A(\lambda) \ge 2$, there must be a fully fragmenting Parter set for λ [5]. It follows that, for any multiplicity list $m \in \mathcal{L}(T)$, there must be an assignment for the multiple eigenvalues. This was not the original intent of Parter's theorem, and the mentioned refinements are needed; the fact is of sufficient import that we record it here.

Theorem 2.2. *If T is a tree and $m \in \mathcal{L}(T)$ includes multiplicities greater than* 1, *then there is an assignment for m.*

The necessity of assignments raises a natural question. Is the existence of a valid assignment not only necessary, but also sufficient for a multiplicity list to exist? We have determined all multiplicity lists for trees on fewer than 12 vertices, and assignments are sufficient to that point [4]. Unfortunately, they are not sufficient in general, as numerical order for the eigenvalues, via interlacing, can cause subtle difficulties for larger trees.

Example 2.3. Consider the graph shown in Fig. 2 and the multiplicity list (3, 3, 3, 1, 1, 1, 1). We construct an assignment for this graph. Let the three multiple eigenvalues be α, β, and γ. Each must have at least two non-adjacent Parter vertices, since each has multiplicity 3 and no vertex has degree more than 3. Vertices 2 and 11 can be Parter for one eigenvalue each, while vertices 4 and 8 can be Parter for two. But two eigenvalues cannot be Parter at both 4 and 8 and still have multiplicity 3, since the subtree {7} is too small to assign both eigenvalues to it. The only remaining possibility is that vertices 2 and 8 are Parter for α, vertices 4 and 11 are Parter for β, and vertices 4 and 8 are Parter for γ.

Each eigenvalue is assigned to every connected component of the tree when the indicated Parter vertices are removed. There are 5 components and 2 Parter vertices for each eigenvalue, so each has multiplicity 3. It is easy to see that there is no overloading. The most nearly overloaded subtrees are {1, 2, 3}, and {11, 12, 13}; each has 3 vertices and 3 assigned eigenvalues. So we have a valid assignment.

However, the subtree {1, 2, 3} has eigenvalues α, β, and γ. Its principal submatrix {1, 3} has eigenvalues α and α. This means, by the interlacing inequalities, that the numerical value of α must be between the values of β and γ. But the same logic, applied to {11, 12, 13}, gives us that the numerical value of β must be between the values of α and γ. Since $\beta \neq \alpha$, we have a contradiction. Since this is the only possible assignment for this multiplicity list, we conclude that $(3, 3, 3, 1, 1, 1) \notin \mathcal{L}(T)$, despite the existence of an assignment.

We know of no such example on fewer vertices, or with fewer vertices of degree 3 and higher, or with fewer than 3 multiple eigenvalues. It is relatively easy to see that there are no such examples for trees of path cover number 2. One implication of what we do here is that these order concerns do not prevent the computation of M_2; as we will see, M_2 can be computed by considering assignments rather than matrices, and thus it can be thought of as a fundamentally combinatorial object.

First we develop an important technical lemma that we have found to be quite useful more broadly, for example in the construction of matrices with a given list for a given tree. It allows us to almost freely choose one eigenvalue of a matrix whose graph is a given tree, as long as we have control over a diagonal entry. Since much about multiplicities and numerical values of eigenvalues may be determined only by branches at the vertex associated with the diagonal entry, the lemma is a powerful tool for refining multiplicity lists.

We use the notation that E_{ij} denotes the matrix with a 1 in the (i, j) position and zeroes elsewhere.

Lemma 2.4 (*Fixed Branches Lemma*). *Suppose that T is a tree, v is a vertex of T, $\lambda \in \mathbb{R}$, and $A \in \mathcal{S}(T)$ is such that $\lambda \notin \sigma(A(v))$. Then there exists exactly one $x \in \mathbb{R}$ such that $\lambda \in \sigma(A + xE_{vv})$.*

Note also that if $m_{A(v)}(\lambda) \geq 2$, then by the interlacing inequalities, $\lambda \in \sigma(A + xE_{vv})$ for any x.

Proof. Consider $\det(A + xE_{vv} - \lambda I)$. It is a linear polynomial in x, and the coefficient of x in that polynomial is $\det(A(v) - \lambda I)$. So unless $\det(A(v) - \lambda I) = 0$, the equation $\det(A + xE_{vv} - \lambda I) = 0$ has exactly one solution. Suppose $\det(A(v) - \lambda I) = 0$. Then λ is an eigenvalue of $A(v)$, which is a contradiction that completes the proof. □

We use the above lemma to prove the principal result of this section. We first observe that any two distinct eigenvalues of $A \in \mathcal{S}(T)$ may be changed to any other two distinct numbers (without changing the multiplicity list) via translation and scalar multiplication applied to A to obtain $B \in \mathcal{S}(T)$. This idea has been used previously (e.g. [9]) and ensures that any one or two eigenvalues of a matrix in $\mathcal{S}(T)$ may be chosen freely.

Theorem 2.5. *Given a tree T on $n = p_1 + p_2 + l$ vertices, a multiplicity list $(p_1, p_2, 1^l)$, a near-assignment of this list for T, and any distinct real numbers α and β, there exists an $A \in \mathcal{S}(T)$ which satisfies the following conditions:*

If R is a connected component of $T - V_1$, α is an eigenvalue of R if and only if $R \in \mathcal{A}_1$. Similarly, if S is a connected component of $T - V_2$, β is an eigenvalue of S if and only if $S \in \mathcal{A}_2$.

Before giving the proof, note that, by interlacing, the matrix A constructed by this theorem has $M_A(\alpha) \geq p_1$ and $M_A(\beta) \geq p_2$, so we immediately have as a consequence:

Corollary 2.6. *For any tree T, if there exists a near-assignment of the list $(p_1, p_2, 1^l)$ to T, then $M_2(T) \geq p_1 + p_2$.*

By this corollary and Theorem 2.2, the question of determining $M_2(T)$ reduces to a question of determining the maximum value of $p_1 + p_2$ among all lists $(p_1, p_2, 1^l)$ that may be near-assigned to T. This shows that M_2 is purely graph theoretic in nature.

Now we present the proof of Theorem 2.5.

Proof. We prove Theorem 2.5 by induction on the number of vertices n. The claim is not hard to show directly for, say, $n \leq 4$ (in fact we have done so up through $n \leq 11$). Thus we assume $n \geq 5$ and proceed by induction.

Let $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$ be a near-assignment for the list $(p_1, p_2, 1^l)$, with V_1 and V_2 the sets of vertices associated with $\mathcal{A}_1$ and $\mathcal{A}_2$. In the trivial case that $V_1 = V_2 = \emptyset$, then A exists as in the claim because (1^{l+2}) is always realizable with any n distinct real numbers as eigenvalues [6]. In particular we may construct such an A with α, β, both, or neither as eigenvalues. Hence we suppose that $V_1 \cup V_2 \neq \emptyset$. Then there is a peripheral $v \in V_1 \cup V_2$, in the sense that all other vertices in $V_1 \cup V_2$ lie in one branch T' of T at v.

For the first case, suppose that $v \in V_1 \cap V_2$. Then each connected component of $T - V_1$ or $T - V_2$ is contained in some branch of T at v; hence the condition that we need to show depends only on the construction of the branches. But each branch is strictly smaller than T, so we can just apply the inductive hypothesis to construct the part of A corresponding to each branch, and then fill in the remaining entries of A in any way we like that ensures $A \in S(T)$.

The two remaining cases are $v \in V_1 \backslash V_2$ and $v \in V_2 \backslash V_1$. Our proof will apply to both cases, so without loss of generality we assume $v \in V_1 \setminus V_2$. Then every element of $T - V_1$ is contained in a branch of T at v, and there is some connected component S of $T - V_2$ such that $v \in S$. Recalling that every other vertex in $V_1 \cup V_2$ is in one branch T' of T at v, we construct a new near-assignment $\mathcal{A}'$ for T' by restricting $\mathcal{A}$ to T'; that is, for $i = 1, 2$, $V_i' = V_i \cup T'$, and $\mathcal{A}_i'$ consists of every subtree in $\mathcal{A}_i$ contained entirely in T. All the conditions of a near-assignment are clearly satisfied for some appropriate multiplicity list $(p_1', p_2', 1^{l'})$. So as T' is strictly smaller than T, we may construct the part of our matrix corresponding to T' by the inductive hypothesis. We can construct each of the other branches of T at v in the exact same way. Then fill in the entries corresponding to the edges at v with ones (or anything nonzero), and fill in the entry corresponding to v with zero, to get a matrix $\bar{A} \in S(T)$.

In fact, $\bar{A}$ is almost what we need. The one connected component we have not yet accounted for is S, because it is the only one not contained in a branch. We need to ensure that β either is or is not (depending on whether $S \in \mathcal{A}_2$) an eigenvalue of the part of our matrix which corresponds to S. So let $\bar{A}[S]$ be the submatrix of $\bar{A}$ corresponding to S, and consider $S(v)$. It consists of $S \cap T'$ (if it is nonempty) and all of the other branches of T at v. But when we constructed these pieces, $S \cap T'$ was a connected component of $T' - V_2'$ that was not in $\mathcal{A}_2'$, so the corresponding part of $\bar{A}$ does not have β as an eigenvalue (this is where we use the "only if" part of the inductive hypothesis). Similarly, each of the other branches does not have β as an eigenvalue. Putting this together, we see that β is not an eigenvalue of $\bar{A}[S(v)]$. So by the Fixed Branches Lemma, there is exactly one $x \in \mathbb{R}$ such that $\beta \in \sigma(\bar{A}[S] + xE_{vv})$. If $S \in \mathcal{A}_2$, then let $A = \bar{A} + xE_{vv}$; if $S \notin \mathcal{A}_2$, then let $A = \bar{A} + yE_{vv}$ for some $y \neq x$. Changing the entry corresponding to v does not affect any of the other conditions, so in either case the resulting A will have all the required properties. This completes the proof of Theorem 2.5. □

3. Upper and lower bounds for M_2

The focus of this section is to give bounds on $M_2(T)$ (defined in the introduction). If $\mathcal{A}$ is an assignment of $(p_1, p_2, 1^l)$ to T with $p_1 + p_2 = M_2(T)$, we refer to $\mathcal{A}$ as an *M_2-maximal assignment*. Notice for every tree T, some matrix in $\mathcal{S}(T)$ maximizes M_2 and thus has an assignment that maximizes M_2. By eliminating all sets of vertices in the definition except those corresponding to the two largest multiplicity eigenvalues, we obtain an assignment to T of the form $(p_1, p_2, 1^l)$ with $p_1 + p_2 = M_2(T)$; hence every tree has an M_2-maximal assignment.

The following two definitions will be useful in several of our proofs, particularly ones using inductive methods. They enable us to speak concretely about the outer vertices of a tree.

Definition 3.1 (*Peripheral HDV, Peripheral Arm*). Given a tree T and a high-degree vertex v, v is a *peripheral HDV* of T if and only if there is a branch of T at v that contains all the other high-degree vertices in T. A *peripheral arm* of a tree T is a branch of T at a peripheral HDV such that the branch does not itself contain any HDV.

We will also use several classes of trees to illustrate and motivate our theorems. Recall that a *pendant* vertex is a vertex of degree 1. A *star* is a tree that has one central high-degree vertex and a number of pendant vertices attached to that central vertex. A *generalized star* is a tree with just one HDV. A *double star* is a tree with exactly two HDVs that are adjacent, and all other vertices adjacent to one of them. A *double generalized star* is a tree with exactly two HDVs that are adjacent.

The following technical lemma simplifies consideration of assignments of the form $(p_1, p_2, 1^l)$. It shows that in such an assignment or near-assignment, if condition (2) of Definition 2.1 fails for some subtree, then it also fails for a single vertex. Thus, as long as we can check that no single vertex is overloaded, we will know that condition (2) is satisfied.

Lemma 3.2 (*Overloading Lemma*). *If T is a tree and $\mathcal{A}$ is an assignment candidate (or a near-assignment candidate) for T for a multiplicity list of the form $(p_1, p_2, 1^\ell)$, but $\mathcal{A}$ is not an assignment (or a near-assignment, respectively), then there must exist a single vertex in T that is overloaded by $\mathcal{A}$.*

Proof. For notational convenience, we write $V_i(S)$ for $V_i \cap S$ and $\mathcal{A}_i(S)$ for $\mathcal{A}_i \cap \mathcal{P}(S)$ throughout the proof. We make the argument assuming that $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$ is a near-assignment candidate for T with Parter vertices V_1 and V_2. The statement for assignment candidates is then a special case.

We prove the claim by contradiction. Assume that T has an overloaded subtree but no overloaded vertex. We further assume that T is the smallest tree on more than one vertex for which such a near-assignment candidate exists. By the definition of overloading, there is a smallest subtree $S \subset T$ which violates part 2 of Definition 2.1. This means that $c_1(S) + c_2(S) = |\mathcal{A}_1(S)| + |\mathcal{A}_2(S)| - |V_1(S)| - |V_2(S)| > |S|$.

If $V_1(S) = V_2(S) = \emptyset$, then $c_1(S)+c_2(S) \leq 1+1 \leq |S|$, since by hypothesis we know that $|S| \geq 2$. This is a contradiction, and so either $V_1(S)$ or $V_2(S)$ is nonempty. Suppose without loss of generality that there is a vertex $v \in V_1(S)$. Let $S_1, S_2, \ldots, S_k$ be the branches of S at v. Since S is the smallest overloaded subtree and each S_i is strictly smaller, we have for each i that $|\mathcal{A}_1(S_i)| + |\mathcal{A}_2(S_i)| - |V_1(S_i)| - |V_2(S_i)| \leq |S_i|$; hence this inequality is also true when we sum over i between 1 and k.

But $v \in V_1$, so $|V_1(S)| = 1 + \sum_i |V_1(S_i)|$. Also, v is not contained in any element of $\mathcal{A}_1(S)$, so each component of $\mathcal{A}_1(S)$ is contained in one of the S_i; hence $|\mathcal{A}_1(S)| = \sum_i |\mathcal{A}_1(S_i)|$. We can also clearly see that $|V_2(S)| \geq \sum_i |V_2(S_i)|$, and since there can be at most one component of $\mathcal{A}_2(S)$ that contains v and hence is not in one of the S_i, we have that $|\mathcal{A}_2(S)| \leq 1 + \sum_i |\mathcal{A}_2(S_i)|$. Of course, $|S| = 1 + \sum_i |S_i|$.

Putting this all together:

$$\begin{aligned} c_1(S) + c_2(S) &= |\mathcal{A}_1(S)| + |\mathcal{A}_2(S)| - |V_1(S)| - |V_2(S)| \\ &\leq \sum_i |\mathcal{A}_1(S_i)| + \left(1 + \sum_i |\mathcal{A}_2(S_i)|\right) - \left(1 + \sum_i |V_1(S_i)|\right) - \sum_i |V_2(S_i)| \\ &= \sum_i (|\mathcal{A}_1(S_i)| + |\mathcal{A}_2(S_i)| - |V_1(S_i)| - |V_2(S_i)|) \leq \sum_i |S_i| < |S|, \end{aligned}$$

which contradicts the fact that S is overloaded. This completes the proof. □

By this lemma combined with Corollary 2.6, we see that if there exists a near-assignment candidate of the list $(p_1, p_2, 1^l)$ to T which has no overloaded single vertices, then $M_2(T) \geq p_1 + p_2$. We will use this observation repeatedly throughout the remainder of the paper.

For any tree T, we let $X(T)$ be the set of pendant vertices in T. We consider inequalities relating $M_2(T)$ and $|X(T)|$, ultimately giving a very nice lower bound for M_2. We begin with the following lemma, which shows that the change in M_2 is restricted when a pendant vertex is added to a tree.

Lemma 3.3. *If x is a pendant vertex of $T + x$, then $M_2(T + x) \leq M_2(T) + 1$.*

Proof. Let $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$ be an M_2-maximal assignment, with multiplicity list $(p_1, p_2, 1^l)$, for $T + x$. Let V_1 and V_2 be the associated Parter vertices, and $v \in T$ be the vertex adjacent to x. In each case, we use $\mathcal{A}$ to construct a near-assignment $(p'_1, p'_2, 1^{l'})$ with $p'_1 + p'_2 \geq p_1 + p_2 - 1$; applying Corollary 2.6 then completes the proof.

Suppose $v \notin V_1 \cup V_2$. Then any subtree in $\mathcal{A}_1$ or $\mathcal{A}_2$ containing p also contains v. For $i = 1, 2$, let $\mathcal{B}_i$ be the collection of subtrees of T resulting from removing x from all subtrees in $\mathcal{A}_i$. Then $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2\}$, with the same Parter vertices, is a near-assignment candidate of the multiplicity list $(p_1, p_2, 1^{l-1})$ for T. By Lemma 3.2, if $\mathcal{B}$ is not a near-assignment, then some single vertex of T must be overloaded. It is clear from this construction that the single vertex could only be v. If v is overloaded, then $\{v\} \in \mathcal{B}_i$ for $i = 1, 2$. By removing $\{v\}$ from $\mathcal{B}_2$ we obtain a near-assignment of $(p_1, p_2 - 1, 1^l)$ to T.

Now suppose $v \in V_1$. For $\mathcal{A}$ to be an M_2-maximal assignment, $\{x\}$ must be in $\mathcal{A}_1$ or $\mathcal{A}_2$. Otherwise maximality is violated because $\{x\}$ could be added to $\mathcal{A}_1$ to give an assignment of $(p_1 + 1, p_2, 1^{l-1})$ to $T + x$. If $v \in V_1 \cap V_2$, then $\{x\}$ lies in exactly one of $\mathcal{A}_1$ or $\mathcal{A}_2$. Removing $\{x\}$ from whichever of $\mathcal{A}_1$ or $\mathcal{A}_2$ contains it yields a near-assignment for $(p_1 - 1, p_2, 1^l)$ or $(p_1, p_2 - 1, 1^l)$ to T.

If $v \in V_1 \setminus V_2$, then remove $\{x\}$ from $\mathcal{A}_1$ and also remove x from all subtrees in $\mathcal{A}_2$. This yields a near-assignment for the list $(p_1 - 1, p_2, 1^l)$ to T, since $\{v\}$ cannot be overloaded — after all, $v \in V_1$. The final possibility, $v \in V_2 \setminus V_1$, is handled by the same argument. □

As a special case we directly compute M_2 for both generalized stars and simple double stars. We know all of the multiplicity lists for these objects from [7].

Example 3.4. For a generalized star S, suppose we have f arms of length 1 and g arms of length 2 or greater. Then $M_2(S)$ is achieved by assigning one eigenvalue to every arm and (if $g \geq 2$) a second to every arm of length 2 or greater. So $M_2(S) = f + 2g - 2$. If $g \leq 1$, it is just $f + g + 1 - 1 = f + g$, since the maximal assignment is achieved with only one high multiplicity, which is assigned to all arms. Note that the generalized star has $f + g$ pendant vertices, so in either case $M_2(S) \geq |X(T)|$.

Let $D_{p,q}$ denote the double star in which one HDV has p, and the other q, pendant vertices attached to it. All multiplicity lists are known for $D_{p,q}$ for all $p, q \geq 2$ [4]. In particular, if p and q are both greater than or equal to 2, we achieve M_2 by making each central vertex Parter for one multiple eigenvalue. This gives the multiplicity list $(p, q, 1^{|D_{p,q}|-p-q})$, which maximizes M_2. So $M_2(D_{p,q}) = p+q$, which is the number of pendant vertices. (If p or q is less than 2, we have a generalized star or a path.)

This shows that $M_2(T) \geq |X(T)|$ for both generalized stars and simple double stars. This property is true in general, giving a lower bound for M_2.

Theorem 3.5. *Let T be a tree on n vertices. Then $M_2(T) \geq |X(T)|$.*

Proof. We prove this by induction on n, the number of vertices in T. The base cases may be observed by direct calculation for $n \leq 10$, using known multiplicity lists [4]. We also have the result for generalized stars and for double stars.

Now assume the claim is true for trees on η vertices with $\eta \leq n$. Let T be a tree on $n + 1$ vertices.

Suppose there exists a pendant vertex x of T that is adjacent to a degree-2 vertex v. In that case, $|X(T)| = |X(T - x)|$. By induction, $M_2(T - x) \geq |X(T - x)|$. So let $\mathcal{A}$ be an M_2-maximal assignment for $T - x$, with V_1, V_2 being the sets of Parter vertices and $\mathcal{A}_1, \mathcal{A}_2$ the collections of subtrees, and then modify it by adding x to any subtree in $\mathcal{A}_1 \cup \mathcal{A}_2$ that contains v. The result will be an assignment for T, with the same multiplicities of the multiple eigenvalues and an additional 1, and so $M_2(T) \geq M_2(T - x) \geq |X(T - x)| = |X(T)|$.

Thus we may suppose instead that each pendant vertex is adjacent to a high-degree vertex. $T - X(T)$ is again a tree and is nonempty except in trivial cases, so let $Y(T)$ be the set of its pendant vertices. We may assume that $T - X(T) - Y(T)$ is nonempty, because if it were empty then T would be at worst a double star, for which we already know the result. Now if $x \in X(T)$, $|X(T - x)| + 1 = |X(T)|$, and so it suffices to find $x \in X(T)$ such that $M_2(T) \geq M_2(T - x) + 1$. Let $\mathcal{A}$ be an M_2-maximal assignment for $T - x$ of a list of the form $(p_1, p_2, 1^l)$.

Suppose that some vertex $v \in Y(T)$ is adjacent in T to $k > 2$ vertices in $X(T)$. Let x be one of those vertices. It is easy to see that because of maximality, no vertex in $X(T)$ is in any V_i; we could modify $\mathcal{A}$ by removing it from V_i and adjusting subtrees as necessary, and this would increase M_2. We can also assume that $v \in V_i$ for some i; if not, $\mathcal{A}$ could be modified by adding v to V_1, removing any subtree containing v from $\mathcal{A}_1$, and then putting each of the $k - 1$ pendant vertices in $\mathcal{A}_1$. This increases $|V_1|$ by 1 and $|\mathcal{A}_1|$ by at least $k - 2 > 0$, so it does not decrease either p_1 or p_2, and the Overloading Lemma guarantees that nothing is overloaded. Thus we may assume that $v \in V_i$. Now expand the near-assignment $\mathcal{A}$ to T by adding

x to $\mathcal{A}_i$ (where $v \in V_i$) and enlarging any subtree containing v to include x as well. This increases p_i by 1 and leaves the other multiplicities unchanged, so we see by Corollary 2.6 that $M_2(T) \geq M_2(T - x) + 1$.

It remains to consider cases in which each vertex $v \in Y(T)$ is adjacent in T to exactly two vertices in $X(T)$. In this case, $T - X(T) - Y(T)$ is a nonempty tree, so it has some pendant vertex u, which is adjacent in T to some number j of vertices in $Y(T)$. Let v be one of these vertices, let x and x' be its two adjacent pendant vertices, and then let $\mathcal{A}$ be an M_2-maximal assignment for $T - x$ as above.

Suppose that u is not in $V_1 \cap V_2$; then $\exists\, i$ such that $u \notin V_i$. We can assume that $v \in V_i$ for the following reason: if it is not, then by maximality, there must be a subtree $S \in \mathcal{A}_i$ containing v and x'. Modify the assignment by removing S, adding v to V_i, and adding $S - v - x'$ and x' to $\mathcal{A}_i$; $u \notin V_i$, so $S - v - x' \neq \emptyset$ and p_i is thus unchanged. So this modification produces an assignment $\mathcal{A}$ where M_2 is maximal and $v \in V_i$. Now expand this assignment to T exactly as in the previous case (add x to $\mathcal{A}_i$, et cetera) to see that $M_2(T) \geq M_2(T - x) + 1$.

Finally, suppose that $u \in V_1 \cap V_2$. We may assume by maximality that none of the j vertices in $Y(T)$ that are adjacent to u is in either V_1 or V_2, and also that every one of those, together with its adjacent pendant vertex (if v) or two pendant vertices (if not v) is in both $\mathcal{A}_1$ and $\mathcal{A}_2$; it is impossible to have any other arrangement in the j outer branches with a higher M_2. So modify $\mathcal{A}$ as follows. Remove u from V_2, and remove any element of $\mathcal{A}_2$ adjacent to u as well. There are at most $j+1$ of these. Now add each of the j vertices in $Y(T)$ that are adjacent to u to V_2. Then add the subtree S of $(T-x) - V_2$ containing u to $\mathcal{A}_2$, and add each of the $2j-1$ pendant vertices adjacent to the j vertices to $\mathcal{A}_2$ as well. This creates a new assignment $\mathcal{A}'$ with no overloading in which p_1 was unchanged. However, $|V_2|$ increased by $j-1$, and $|\mathcal{A}_2|$ increased by at least $2j-(j+1) = j-1$, so $\mathcal{A}'$ has at least the same p_2 and thus must be M_2-maximal as well. But now $v \in V_2'$, so expand the assignment to T as in the other cases to see that $M_2(T) \geq M_2(T - x) + 1$. This completes the proof. □

There is also a simple upper bound for M_2, related to the diameter of the tree.

Theorem 3.6. *Let T be a tree on n vertices and d be the diameter of T, measured in vertices. Then $M_2(T) \leq n + 2 - d$.*

Proof. By the remarks in the first paragraph of this section, there exists a list $(p_1, p_2, 1^l)$ and an assignment of this list to T where $p_1 + p_2 = M_2(T)$. Then $d \leq \ell + 2$, since d is a lower bound on the number of distinct eigenvalues, by comment (3) in the introduction. Thus $M_2(T) = n - l \leq n + 2 - d$. □

These bounds are often good estimates, but they are not exact. An example showing that the lower bound is not an equality is the generalized star on 7 vertices with 3 arms, each of length 2. We call this graph S_3. This graph has 3 pendant vertices, but admits the multiplicity list (2, 2, 1, 1, 1), so $M_2 = 4$. This is, in fact, the smallest counterexample to the lower-bound equality statement.

To notice that both bounds can be violated simultaneously by arbitrarily large amounts, consider a generalized star G with m arms of length m ($m \geq 3$). Then, using the terminology from Example 3.4, $f = 0$ and $g = m$. Thus $M_2(G) = 2m - 2$. But $|X(G)| = m$, and $n+2-d = (m^2+1+2-(2m+1)) = m^2-2m+2$. So $(n+2-d) - M_2(G) = m^2-4m+4 = (m-2)^2$, and also $M_2(G) - |X(G)| = m-2$. By making m large enough both bounds can be simultaneously inexact by arbitrarily large amounts.

4. M_2 in special cases

In this section we give characterizations of M_2 for several special classes of trees, in the process showing several important connections among M_2, M_1, and combinatorial information about a tree.

The first class of trees we consider are caterpillars. Recall that a tree is a *caterpillar* if there exists a diameter of the tree such that all vertices of the tree either lie on the diameter or are adjacent to a vertex in the diameter. The class of caterpillars is a natural one to consider in this context because it is the class of trees that do not contain S_3, the generalized star with one central vertex and three arms of length 2, as an induced subtree. The graph S_3 is an important example because it is the smallest tree for which the equation $M_2(T) = |X(T)|$ does not hold; in fact, the following theorem shows that all trees that do not contain S_3 as an induced subtree will necessarily have M_2 equal to the number of pendant vertices.

Theorem 4.1. *If T is a caterpillar, then $|X(T)| = M_2(T)$.*

Proof. Note that for any caterpillar, $n + 2 - d$, where d is the diameter, is exactly the number of pendant vertices since a caterpillar consists of a diameter with pendant vertices hung off of it. Therefore, the two bounds combine to give $V(T) \leq M_2(T) \leq n + 2 - d$, showing that $M_2(T) = V(T)$ (and $M_2(T) = n + 2 - d$). □

Another class of trees with nice properties is the following:

Definition 4.2 (*Segregated Trees*). A tree T is called segregated iff no two HDVs of T are adjacent.

To understand why segregated trees are important in this context, we first give an example.

Example 4.3. The tree in Fig. 3 has an HDV, namely the vertex labeled 6, that is not in any V_i in any maximal assignment. For example, an M_2-maximal assignment for two multiple eigenvalues, both with multiplicity 3, is to put vertices 3 and 8 in $V_1 \cap V_2$ and then put each of the subtrees $\{1, 2\}$, $\{4, 5\}$, $\{6, 7\}$, $\{9, 10\}$. and $\{11, 12\}$ in $\mathcal{A}_1 \cap \mathcal{A}_2$. $M_2(T)$ is in fact exactly 6, since the maximum multiplicity of any individual eigenvalue is equal to the path cover number $P(T)$, which is 3. Also, note that the example given is *not* a segregated tree.

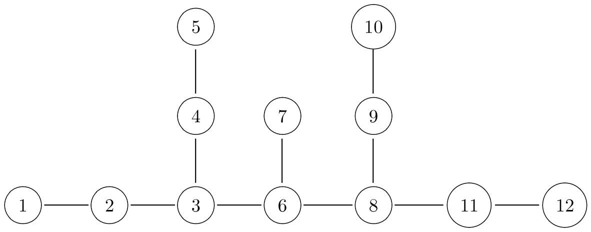


Fig. 3. Not every HDV need be in a V_i in an M_2-maximal assignment.

In contrast to trees such as the example given above, segregated trees have several nice properties and a simple characterization of M_2 (though not as simple as it is for caterpillars). This occurs because of the space between HDVs, which, as the next lemma shows, forces all HDVs to be in $V_1 \cup V_2$ in some M_2-maximal assignment, unlike the example above.

Lemma 4.4. *If T is a segregated tree then any M_2-maximal assignment for T of a list $(p_1, p_2, 1^l)$ has every HDV in $V_1 \cup V_2$. Furthermore, there exists an M_2-maximal assignment of this form in which $p_1 = P(T)$ and all HDVs are in V_1.*

Proof. Suppose that $\mathcal{A}$ is an M_2-maximal assignment for T of the form $(p_1, p_2, 1^l)$, and that v is an HDV which is not Parter for either α or β. Modify the assignment by removing a subtree containing v from $\mathcal{A}_1$ if necessary, putting v in V_1, and assigning α to every component of $T - V_1$ adjacent to v. Because of the segregation, this will not create overloading. And also because of the segregation, this increases $|V_1|$ by 1 and $|\mathcal{A}_1|$ by at least 2, which means that p_1 increases by at least 1, while p_2 remains the same. This contradicts the maximality of $\mathcal{A}$. Hence all maximal assignments of this form have each HDV in $V_1 \cup V_2$.

Now pick one of these assignments. Construct a new assignment as follows: if possible, pick an HDV v that is in $V_2 \setminus V_1$, instead put it in $V_1 \setminus V_2$, and assign α to every subtree adjacent to v and β to the subtree of $T - V_2$ containing v. Repeat this until there are no more such HDVs. This will not overload any single vertex (because nothing that is in $V_1 \cap V_2$ is changed, this process never creates a single vertex that is in both $\mathcal{A}_1$ and $\mathcal{A}_2$). And when we make one of these replacements, $|V_1 \cup V_2|$ is unchanged. If v has degree, k then we could lose at most one member of $\mathcal{A}_1$ and at most k members of $\mathcal{A}_2$, but we gain k members of $\mathcal{A}_1$ and one member of $\mathcal{A}_2$. So $|\mathcal{A}_1 \cup \mathcal{A}_2|$ does not decrease, and so $p_1 + p_2$ does not decrease. Thus the new assignment is still maximal, and since each HDV was in $V_1 \cup V_2$ at the start, each HDV ends up in V_1. And since each HDV is of degree at least three with no adjacent HDVs, this does in fact maximize the multiplicity of α, and so $p_1 = P(T)$. □

Definition 4.5. Suppose T is a segregated tree with vertices $1, \ldots n$. Let R be the set of high-degree vertices of T. Every component of $T - R$ will be an induced path. Replace each such path of length m by a path of length $m - 1$, and re-assemble the graph. In the case that there are components (paths) of length 1: if the component is a pendant vertex of T, remove it. If it is an interior vertex of T, perform a reverse edge subdivision (eliminate the vertex and place an edge between the vertices it had been connected to). We call the resulting graph $\hat{T}$.

The effect of creating $\hat{T}$ from T is to remove all pendant vertices and to shorten by 1 the distance between each pair of HDVs. Of course, $\hat{T}$ need not be segregated. This definition lets us state our main theorem on segregated trees:

Theorem 4.6. *If T is a segregated tree, then $M_2(T) = P(T) + P(\hat{T})$. Moreover, $(P(T), P(\hat{T}), 1^l)$ can be assigned to T.*

Proof. By Lemma 4.4, we can choose an M_2-maximal assignment $\mathcal{A}$ in which each HDV is in $|V_1|$, $p_1 = P(T)$, and each of the $P(T) + |V_1|$ paths induced by the removal of the HDVs is in $\mathcal{A}_1$. Some, but not necessarily all, of the HDVs will also be in V_2. Notice that the vertices removed from T to produce $\hat{T}$ are in one-to-one correspondence with the induced paths in $\mathcal{A}_1$, and one vertex is removed from each such path. Now in $\hat{T}$, there exists a set of k HDVs such that when they are removed, the resulting graph has j connected components and $j - k = P(\hat{T})$. Modify $\mathcal{A}$ as follows: remove everything in V_2 and $\mathcal{A}_2$, replace V_2 with this collection of k vertices, and put every induced subtree of $T - V_2$ in $\mathcal{A}_2$. This creates an assignment candidate, so we must remove from $\mathcal{A}_2$ any single vertex subtree which already belonged to $\mathcal{A}_1$ to eliminate overloading. However, consider any of the j connected components of $\hat{T} - V_2$. If the analogous component in T has more than one vertex, we do not have to remove it from $\mathcal{A}_2$. If it does have exactly one vertex, the only way that can occur is for the vertex to be an HDV in T (otherwise it would shrink to nothing in $\hat{T}$). That vertex would be adjacent in T to an element of V_2, but T is segregated, so this cannot happen. Thus $|\mathcal{A}_2| \geq k$, so $p_2 \geq j - k = P(\hat{T})$, and so $M_2(T) \geq P(T) + P(\hat{T})$.

We claim now that $M_2(T) = P(T) + P(\hat{T})$. Suppose for contradiction that it were larger, say $P(T) + r$ for some $r \geq P(\hat{T})$. Then we would be able to achieve that with an M_2-maximal assignment of the form in Lemma 4.4. So there would be a set W_2 of HDVs in T for which $T - W_2$ had $r + |W_2|$ connected non-singleton components (non-singleton because all the singletons are in $\mathcal{A}_1$ and thus cannot be in $\mathcal{A}_2$). None of these components would shrink to nothing in $\hat{T}$, and so $\hat{T} - W_2$ would have $r + |W_2|$ connected components, so $P(\hat{T}) \geq r$, which is a contradiction. This tells us that $M_2(T) = P(T) + P(\hat{T})$, and also tells us at the same time that p_2 above must equal $P(\hat{T})$. So we have an M_2-maximal assignment of the form $(P(T), P(\hat{T}), 1^l)$. □

This theorem tells us several nice things about segregated trees. For example, we see that M_1 and M_2 are achieved simultaneously for segregated trees; in general, this is not true. There are counterexamples in the list in [4]. Also, we note that the formula $M_2(T) = P(T) + P(\hat{T})$ does not hold in general, even if we try to extend the definition of $\hat{T}$. For example, if we try to extend it in the obvious way by saying that we leave unchanged edges connecting adjacent HDVs, the double star $D_{2,2}$ is easily seen to be a counterexample.

5. An algorithm for M_2

According to our prior results, the question of determining M_2 for a given tree T reduces to a question of finding an optimal assignment to T. In this section, we give a theorem that allows the calculation of M_2 from simple reductions of the tree, without actually constructing any assignment. The idea is to remove sets of vertices from the tree in such a way that we know the effect of each reduction on M_2. The process may be continued until a path, for which M_2 is 2 (or 1 if the path is length 1), is reached. Then the original M_2 may be calculated.

Throughout this section, we will consider a peripheral HDV v in a tree T. The subtree of T consisting of v and its peripheral arms will be called S- however, if v is the only *HDV* in T, we will let S be v and all but one of its peripheral arms (chosen arbitrarily). The point is that S should be a generalized star containing everything except a single branch of T at v. We define the forest F to be $S - v$, and let w be the one vertex adjacent to v that is not in F.

In order to prove the reduction theorem, we will consider assignments of lists with at most two multiple eigenvalues, of the form $(p_1, p_2, 1^l)$; all assignments, near-assignments, et cetera in this section will be of this form unless explicitly stated otherwise. The sets of Parter vertices will be V_1 and V_2, and the collections of subtrees will be $\mathcal{A}_1$ and $\mathcal{A}_2$.

We will implicitly use a couple of facts. First, in any M_2-maximal assignment, no pendant vertex is in a V_i; for if it were, we could remove it, expand other subtrees as necessary, and increase $p_1 + p_2$. Second, we note the following combination of Corollary 2.6 and the Overloading Lemma: if there exists a near-assignment candidate of $(p_1, p_2, 1^l)$ to T with no overloaded single vertex, then $M_2(T) \geq p_1 + p_2$. This will be used to avoid worrying about condition (1d) or condition (2) on subtrees that are not single vertices. Third, we note that any near-assignment with $p_1 + p_2 = M_2(T)$ must actually be an assignment; if it were not, we could remove any vertices in V_i that are not adjacent to an element of $\mathcal{A}_i$ and strictly increase $p_1 + p_2$, which contradicts the maximality.

Now we give a technical lemma.

Lemma 5.1. *Suppose that v is a peripheral HDV in T. Then there exists an M_2-maximal assignment in which $v \in V_1 \cup V_2$. Moreover, if v has at least two peripheral arms of length at least 2, then there exists an M_2-maximal assignment in which $v \in V_1 \cap V_2$. If v has at most one peripheral arm of length at least 2, then there exists an M_2-maximal assignment in which v is in exactly one of V_1 and V_2.*

Proof. To prove the first statement, consider an M_2-maximal assignment $\mathcal{A}$ for T in which $v \notin V_1 \cup V_2$. Modify this assignment as follows. First, if there is a subtree $R \in \mathcal{A}_1$ with $v \in R$, remove it (note that the peripheral arms would also have to be in R). Then put v in V_1 and put two of the peripheral arms in $\mathcal{A}_1$. We get at least a near-assignment with no overloaded single vertices, and p_1 does not decrease, while p_2 remains unchanged. Hence the new near-assignment must also be M_2-maximal, so it must be an assignment.

To prove the second statement, consider an M_2-maximal assignment for T. By what we just proved, we may assume that $v \in V_2$. But then apply the same procedure as with the first statement: remove any $R \in \mathcal{A}_1$ with $v \in R$, put v in V_1, and put the two longest peripheral arms in $\mathcal{A}_1$. This avoids overloading problems, as neither of those arms is a single vertex. This, analogously, gives an M_2-maximal assignment of the desired type.

For the third statement, let $\mathcal{A}$ be an M_2-maximal assignment for T; again, we use the first statement to assume that $v \in V_1$. If $v \notin V_2$, we are done, so assume $v \in V_1 \cap V_2$. By maximality, each length-1 arm is in one of $\mathcal{A}_1$ or $\mathcal{A}_2$, and any longer arm is in both. There also might be some subtree R containing w in $\mathcal{A}_2$. Modify the assignment as follows: remove v from V_2, put every length-1 arm that is in $\mathcal{A}_2$ in $\mathcal{A}_1$ instead, remove the longer arm from $\mathcal{A}_2$ if applicable, and if there is an R as above, remove it from $\mathcal{A}_2$ and put $R \cup S$ in $\mathcal{A}_2$ instead. This gives a near-assignment, and decreases $|V_1| + |V_2|$ by 1 while decreasing $|\mathcal{A}_1| + |\mathcal{A}_2|$ by at most 1. Hence $p_1 + p_2$ does not decrease. So the result is still an M_2-maximal assignment, of the desired type. □

Now we present the main theorem, which allows us to make a reduction of the tree near any peripheral HDV with known effect upon M_2. Iterative application of this theorem allows us to calculate M_2 for any tree.

Theorem 5.2 (*Reduction Theorem*). *Let T be a tree and v a peripheral HDV, with S and F as defined earlier in this section. Suppose that S has f arms of length 1 and g arms of length at least 2. Then:*

(A) *If $g \geq 2$, then $M_2(T - S) = M_2(T) - f - 2g + 2$.*
(B) *If $g \leq 1$, then $M_2(T - F) = M_2(T) - f - g + 1$.*

Proof. *Part A*: Let $\mathcal{A}$ be an M_2-maximal assignment for $T - S$. Add back S, put v in $V_1 \cap V_2$, put each of the $f + g$ peripheral arms in $\mathcal{A}_1$, and put each of the g peripheral arms of length at least 2 in $\mathcal{A}_2$. This creates an assignment for T in which p_1 increases by $f + g - 1$ and p_2 increases by $g - 1$, so $p_1 + p_2$ increases by $f + 2g - 2$. Thus by Corollary 2.6, $M_2(T) \geq M_2(T - S) + f + 2g - 2$.

Conversely, by Lemma 5.1, there is an M_2-maximal assignment $\mathcal{A}$ for T in which $v \in V_1 \cap V_2$. Remove v from V_1 and V_2, then remove each arm from whichever $\mathcal{A}_i$ it might be in, then remove S to get an assignment for $T - S$. Now we have reduced

$|V_1| + |V_2|$ by 2. And each arm of length 1 could have been in at most one $\mathcal{A}_i$, while each arm of length 2 could have been in both, so we have reduced $|\mathcal{A}_1| + |\mathcal{A}_2|$ by at most $f + 2g$. Hence $p_1 + p_2$ has decreased by at most $f + 2g - 2$, so by Corollary 2.6, $M_2(T - S) \geq M_2(T) - f - 2g + 2$. Putting this together with the other inequality, we see that $M_2(T - S) = M_2(T) - f - 2g + 2$.

Part B: Let $\mathcal{A}$ be an M_2-maximal assignment for $T - F$. v is pendant, so by maximality, $v \notin V_1 \cup V_2$. There are several cases.

If $w \in V_1 \cap V_2$, then v is in exactly one $\mathcal{A}_i$ by maximality. Take it out of that $\mathcal{A}_i$, add back F, put $v \in V_1$, put each of the $f + g$ arms in $\mathcal{A}_1$, and put S in $\mathcal{A}_2$. This gives an assignment for T where $|V_1| + |V_2|$ has increased by 1, and $|\mathcal{A}_1| + |\mathcal{A}_2|$ has increased by $f + g$, so $p_1 + p_2$ has increased by $f + g - 1$.

If $w \in V_1 \setminus V_2$ (the case of $V_2 \setminus V_1$ is the same), then by maximality, $v \in \mathcal{A}_1$. Also by maximality, the connected component R of $T - V_2$ which contains v and w is in $\mathcal{A}_2$; it is bigger than a single vertex, so by the Overloading Lemma it will not create an overloading problem. Now take v out of $\mathcal{A}_1$, take R out of $\mathcal{A}_2$, add back F, put S in $\mathcal{A}_1$, put $v \in V_2$, put $R - v$ in $\mathcal{A}_2$, and put each of the $f + g$ arms in $\mathcal{A}_2$. Again, since $w \notin \mathcal{A}_1$, putting $R - v$ in $\mathcal{A}_2$ will not cause overloading, so we get a near-assignment for T. The process increases $|V_2|$ by 1, increases $|\mathcal{A}_2|$ by $f + g$, and leaves everything else unchanged, so $p_1 + p_2$ increases by $f + g - 1$.

If $w \notin V_1 \cup V_2$, then for each i, w and v are contained in some connected component R_i of $T - V_i$; by maximality, $R_i \in \mathcal{A}_i$ for each i, as if it were not, putting it in would create a better near-assignment with no overloading. Now: add back F, remove each R_i from $\mathcal{A}_i$, put $v \in V_1$, put $R_2 \cup F$ in $\mathcal{A}_2$, put $R_1 - v$ in $\mathcal{A}_2$, and put each of the $f + g$ arms in $\mathcal{A}_1$. This increases $|V_1|$ by 1, increases $|\mathcal{A}_1|$ by $f + g$, and leaves everything else unchanged. Additionally, there is no overloading of a single vertex and so we have a near-assignment for T with $p_1 + p_2$ having increased by $f + g - 1$.

Thus, in each case, we have a near-assignment to T with $p_1 + p_2 = M_2(T - F) + f + g - 1$, so by Corollary 2.6, $M_2(T) \geq M_2(T - F) + f + g - 1$.

Conversely, by Lemma 5.1, there is an M_2-maximal assignment $\mathcal{A}$ for T in which v is exactly one of V_1 or V_2. Without loss of generality, assume that $v \in V_1$. By the proof of Lemma 5.1, we can also assume that each peripheral arm is in $\mathcal{A}_1$, and that the connected component R of $T - V_2$ which contains S is in $\mathcal{A}_2$. There are again multiple cases.

If $w \in V_1 \cap V_2$, then $R = S$. Remove S from $\mathcal{A}_2$, remove the $f + g$ arms from $\mathcal{A}_1$, remove v from V_1, remove F, and then put $v \in \mathcal{A}_1$ to get a new near-assignment for $T - F$. $|V_1|$ decreased by 1, $|V_2|$ was unchanged, $|\mathcal{A}_1|$ decreased by $f + g - 1$, and $|\mathcal{A}_2|$ decreased by 1; so $p_1 + p_2$ decreased by $f + g - 1$.

If $w \in V_2 \setminus V_1$, then $R = S$ again, and by maximality the connected component of $T - V_1$ which contains w, call it Q, is in $\mathcal{A}_1$. Remove S from $\mathcal{A}_2$, remove the $f + g$ arms from $\mathcal{A}_1$, remove Q from $\mathcal{A}_1$, remove v from V_1, remove F, and then put $Q \cup v$ back in $\mathcal{A}_1$ and put v itself in $\mathcal{A}_2$ to get an assignment for $T - F$. Now $|V_1|$ decreased by 1, $|V_2|$ was unchanged, $|\mathcal{A}_1|$ decreased by $f + g$, and $|\mathcal{A}_2|$ was unchanged, so $p_1 + p_2$ decreased by $f + g - 1$.

If $w \in V_1 \setminus V_2$, then remove R from $\mathcal{A}_2$, remove the $f + g$ arms from $\mathcal{A} + 1$, remove v from V_1, remove F, put v in $\mathcal{A}_1$, and put $R - F$ in $\mathcal{A}_2$ to get an assignment for $T - F$. This decreased $|V_1|$ by 1, left $|V_2|$ unchanged, decreased $|\mathcal{A}_1|$ by $f + g - 1$, and left $|\mathcal{A}_2|$ unchanged, so $p_1 + p_2$ only decreased by $f + g - 2$, which is even better.

Finally, if $w \notin V_1 \cup V_2$, by maximality the connected component of $T - V_1$ which contains w, call it Q again, is in $\mathcal{A}_1$. Take Q out of $\mathcal{A}_1$, take R out of $\mathcal{A}_2$, take the $f + g$ arms out of $\mathcal{A}_\infty$, take v out of V_1, put $Q + v$ in $\mathcal{A}_1$ and put $R - F$ in $\mathcal{A}_2$. This gives a near-assignment for $T - F$. And this process decreased $|V_1|$ by 1, left $|V_2|$ unchanged, decreased $|\mathcal{A}_1|$ by $f + g$, and left $|\mathcal{A}_2|$ unchanged, so $p_1 + p_2$ decreased by $f + g - 1$.

In each of these cases, Corollary 2.6 allows us to conclude that $M_2(T - F) \geq M_2(T) - f - g + 1$. Combining it with the other inequality, we see that $M_2(T - F) = M_2(T) - f - g + 1$. This finishes the proof of the Reduction Theorem. □

We now provide an example of the determination of M_2 by application of the Reduction Theorem.

Example 5.3. Consider the graph on 13 vertices of Example 2.3. We know that the list $(3, 3, 1^7)$ is an M_2-maximal assignment - just use the assignment of $(3, 3, 3, 1^4)$ that we discussed earlier and remove all the references to γ. And 3 is the path cover number, so this will maximize M_2; M_2 for this graph is 6. What does the reduction theorem tell us?

Consider the two peripheral HDVs: vertices 2 and 11. They both have $f = 2$ and $g = 0$. So by part B, removing vertices 1 and 3 will decrease M_2 by $2 + 0 - 1 = 1$, as will removing vertices 12 and 13 afterwards. (Removing 1 and 3 has no effect on the f and g values for 11).

After removing these vertices, vertex 4 is a peripheral HDV with $f = 1$ and $g = 1$. Hence by part B, removing vertices 3, 5, and 6 will decrease M_2 by $1 + 1 - 1 = 1$. What remains is a generalized star: a single HDV, 8, with two arms of length 2 and one arm of length 1. As in the definition of S at the beginning of this section, we will let S be v and both of the length-2 arms, so that $f = 0$ and $g = 2$. [Any other choice of S would also work, by the theorem]. Then by part A, removing S decreases M_2 by $0 + 4 - 2 = 2$, and we are left with a single vertex, for which M_2 is 1.

So to arrive at the graph with $M_2 = 1$, we removed vertices in such a way that we decreased M_2 by 1 three times and by 2 once. This means that M_2 for the original graph is $1 + 3 * 1 + 1 * 2 = 6$, as noted above.

6. Conclusion

The parameter M_1 was understood fully (from a combinatorial perspective) in [JL-D]. We have now rather fully described M_2. Because of Example 2.3, it appears that M_j for j greater than or equal to 3 should be more subtle. In that example, there is an assignment (not overloaded) for which there are three eigenvalues, each of multiplicity 3, for a total hypothesized M_3 of 9. But, as the path cover number is 3 and three eigenvalues of multiplicity 3 cannot occur in any achievable multiplicity list, M_3 is actually 8 (which is easily constructed). Whether there should be a simple combinatorial description of M_3 or of $M_j, j > 3$, is unclear. For some natural subsets of trees, similar approaches may yield nice results. In fact, the following are natural and appealing questions: for what trees do all assignments imply the existence of achievable multiplicity lists, and for what sorts of assignments do multiplicity lists necessarily exist?

A further question along the lines of this work is: what pairs (m_1, m_2) occur for a given tree? Of course, we must have $m_1 \leq M_1$ and $m_1 + m_2 \leq M_2$ (and $m_1, m_2 \geq 0$). But given M_1 and M_2, there may still be further restrictions about (m_1, m_2). For example, the double star $D_{3,3}$ has path cover number 4. M_2 is equal to 6 because (3, 3, 1, 1) is in $\mathcal{L}(T)$. But (4, 2, 1, 1) is not, so that we cannot simultaneously achieve M_1 and M_2. However, even when it is possible to do so, as in the case of segregated trees, some pairs (m_1, m_2) meeting the above constraints may still not occur. For example, in the singly separated double star (two HDVs connected by a path of length 1) where each HDV has three pendant vertices, we know that $M_1 = 5$ and $M_2 = 6$. The pairs $m_1 = 5, m_2 = 1$ and $m_1 = 3, m_2 = 3$ both occur, but $m_1 = 4, m_2 = 2$ does not. An interesting subquestion is simply: what is the maximum possible value of m_2, given that $m_1 = M_1$?

References

[1] R. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, New York, 1985.

[2] C.R. Johnson, A. Leal-Duarte, The maximum multiplicity of an eigenvalue in a matrix whose graph is a tree, Linear Algebra and Multilinear Algebra 46 (1999) 139–144.

[3] C.R. Johnson, A. Leal-Duarte, On the minimum number of distinct eigenvalues for a symmetric matrix whose graph is a given tree, Mathematical Inequalities and Applications 5 (2) (2002) 175–180.

[4] C.R. Johnson, A. Leal-Duarte, C.M. Saiago, D. Sher, Eigenvalues, Multiplicities, and Graphs, in: Contemporary Mathematics Vol. 419: Algebra and its Applications, 2006, pp. 167–183.

[5] C.R. Johnson, A. Leal-Duarte, C.M. Saiago, B.D. Sutton, A.J. Witt, On the relative position of multiple eigenvalues in the spectrum of an Hermitian matrix with a given graph, Linear Algebra and its Applications 363 (2003) 147–159.

[6] C.R. Johnson, A. Leal-Duarte, C.M. Saiago, The Parter–Wiener theorem: Refinement and generalization, SIAM Journal on Matrix Analysis and Applications 25 (2) (2003) 352–361.

[7] C.R. Johnson, A. Leal-Duarte, C.M. Saiago, Inverse eigenvalue problems and lists of multiplicities of eigenvalues for matrices whose graph is a tree: The case of generalized stars and double generalized stars, Linear Algebra and its Applications 373 (2003) 311–330.

[8] S. Parter, On the eigenvalues and eigenvectors of a class of matrices, Journal of the Society for Industrial and Applied Mathematics 8 (1960) 376–388.

[9] C.M. Saiago, The Possible Multiplicities of the Eigenvalues of an Hermitian Matrix Whose Graph is a Tree, Dissertação de Doutoramento, Universidade Nova de Lisboa, 2003.

[10] G. Wiener, Spectral Multiplicity and splitting results for a class of qualitative matrices, Linear Algebra and Applications 61 (1984) 15–29.