Conf-9505145--1

SAND94–3082
Unlimited Release
Printed November 1994

Distribution
Category UC–405

A Spectral Algorithm for the Seriation Problem

Jonathan E. Atkins†
Erik G. Boman‡
Bruce Hendrickson§

Abstract.

Given a set of objects and a correlation function f reflecting the desire for two items to be near each other, find all sequences π of the items so that correlation preferences are preserved; that is if $\pi(i) < \pi(j) < \pi(k)$ then $f(i,j) \geq f(i,k)$ and $f(j,k) \geq f(i,k)$. This *seriation problem* has numerous applications, for instance, solving it yields a solution to the consecutive ones problem. We present a spectral algorithm for this problem that has a number of interesting features. Whereas most previous applications of spectral techniques provided bounds or heuristics, our result is an algorithm for a nontrivial combinatorial problem. Our analysis introduces powerful tools from matrix theory to the theoretical computer science community. Also, spectral methods are being applied as heuristics for a variety of sequencing problems and our result helps explain and justify these applications. Although the worst case running time for our approach is not competitive with that of existing methods for well posed problem instances, unlike combinatorial approaches our algorithm remains a credible heuristic for the important cases where there are errors in the data.

† Dept. Mathematics, University of Michigan, 3217 Angell Hall, Ann Arbor, MI 48109–1003. atkinsje@math.lsa.umich.edu.

‡ Scientific Computing & Computational Mathematics, Bldg. 460, Stanford Univ., Stanford, CA 94305–2140. boman@sccm.stanford.edu.

§ Applied & Numerical Math Dept., Sandia National Labs, Albuquerque, NM 87185–1110. bah@cs.sandia.gov.

* This work was supported by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research, and was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract No. DE-AC04-76DP00789.

1. Introduction. Many problems in computer science involve ordering a set of items in such a way that closely coupled elements are placed near each other. This is the underlying problem in such diverse applications as genomic sequencing, sparse matrix profile reduction and determining efficient database geometries. In this paper we present a *spectral* algorithm for this class of problems. Unlike traditional combinatorial methods, our approach uses an eigenvector of a matrix to order the items. Our main result is that this approach correctly solves a large class of ordering problems, including the consecutive ones problem [4].

More formally, consider a set of n objects we wish to sequence; that is we wish to bijectively map the elements to the integers $1, \ldots, n$. We also have a real valued *correlation function* (sometimes called a *similarity function*) $f(i,j) = f(j,i)$ which reflects the desire for items i and j to be near each other in the sequence. The correlation function can be thought of as a weighted graph or as a symmetric matrix.

We now wish to find all ways to sequence the elements so that the correlations are *consistent*; that is, if π is our permutation of elements and $\pi(i) < \pi(j) < \pi(k)$ then $f(i,j) \geq f(i,k)$ and $f(j,k) \geq f(i,k)$ for all i, j and k. Not all correlation functions allow for a consistent sequencing. If a consistent ordering is possible we will say the problem is *well posed*. Determining an ordering from a correlation function is what we will call the *seriation problem*, reflecting its origins in archeology [14, 16]. The seriation problem arises naturally in many other application areas, for example musical chronology [9] and psychometrics [1].

Although we are not aware of any published results, we believe well–posed seriation problems are easy to solve; they are certainly polynomial and perhaps linear. For instance, there is a linear time algorithm for the consecutive ones problem [4]. However, many ordering problems are generated by experimental data, which can contain some inconsistencies. In this case there may be no consistent solution, so we would instead like to find a best approximation. Unfortunately, this often makes the problem NP–complete [8]. Most combinatorial algorithms for well–posed problems break down when the data is inconsistent, limiting their value for many problems.

In this paper, we present a new algorithm for these problems that correctly solves well–posed instances, while serving as a credible heuristic if the data is inconsistent. This technique is already being used with success to address problems in genomic mapping [8], envelope reduction [2] and database organization [3]. Our goal here is not to analyze the approach as an approximation algorithm, but rather to prove that it correctly solves well–posed seriation problems.

In terms of worst case run time our approach is not competitive with the best algorithms for well–posed instances of these problems, but our results are still interesting for several reasons. Besides the ability of our approach to gracefully handle inconsistent

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

data, our technique provides a unifying paradigm for a wide range of problems. It is also a novel use of spectral methods, which we hope will stimulate further research. Whereas most previous applications of spectral techniques provided bounds for NP–complete problems or heuristics, our result is an *algorithm* for a nontrivial combinatorial problem. In addition, the techniques we require to prove our results are likely to be unfamiliar to many members of the theoretical computer science community, particularly our use of some classical results from matrix theory. We hope to make the community aware of some of these techniques which may find uses in other settings. Finally, spectral methods are already being applied with success to a variety of sequencing problems, and our result lends credence to their use.

This paper is organized in the following way. In the next section we introduce the mathematical notation and the results from matrix theory that we will need later. We also describe a spectral heuristic for ordering problems which motivates the rest of the paper. The theorem which underpins our algorithm is proved in §3. Several additional results in §4 lead us to an algorithm and its analysis in §5. We describe some applications in §6. Our results motivate several open problems which we mention in §7.

2. Mathematical background. In this section we define terms and review some classical results from matrix theory that we will need in our proofs. The use of matrix concepts is useful because the correlation function defined above can be considered as a real, symmetric matrix. A permutation of the items corresponds to a symmetric permutation of this matrix. The question of whether or not the ordering problem is well posed can also be asked as a property of this matrix. Specifically, assume the matrix has been permuted to reflect a consistent solution to the ordering problem. The off–diagonal matrix entries must now be non–increasing as we move away from the diagonal. More formally, we will say a matrix $A \in \mathbb{R}^{n\times n}$ is an R–matrix[1] if and only if A is symmetric, has all non–negative off–diagonal entries and

$$\begin{aligned} a_{ij} &\leq a_{ik} \qquad \text{for } j < k < i, \\ a_{ij} &\geq a_{ik} \qquad \text{for } i < j < k. \end{aligned}$$

If A can be symmetrically permuted to become an R–matrix, then we say that A is *pre–R*. Note that pre–R matrices correspond precisely to well posed ordering problems.

When π is a permutation of the natural numbers $\{1,\ldots,n\}$ and x is a column vector, i.e. $x^t = [x_1,\ldots,x_n]$, we will denote by x^π the permutation of x by π, i.e. $x_i^\pi = x_{\pi(i)}$. Similarly, A^π is the symmetric permutation of A by π, i.e. $a_{i,j}^\pi = a_{\pi(i),\pi(j)}$.

[1] This class of matrices is named after W. S. Robinson who first defined this property in his work on seriation methods in archeology [14].

We denote by e the vector whose entries are all 1, by e_i the vector consisting of zeros except for a 1 in position i, and by I the identity matrix. A matrix $A \in \mathbb{R}^{n \times n}$ is *reducible* if there exists a permutation π such that

$$A^\pi = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix},$$

where $B \in \mathbb{R}^{r \times r}$ and $D \in \mathbb{R}^{(n-r)\times(n-r)}$ and $0 < r < n$. If no such permutation exists then A is *irreducible.*

We say that λ is an *eigenvalue* of A if $Ax = \lambda x$ for some vector $x \neq 0$. The corresponding eigenvector x is an *eigenvector.* For real, symmetric matrices the eigenvectors can be constructed to be pairwise orthogonal. The *(algebraic) multiplicity* of the eigenvalue λ is defined as the number of times λ occurs as a root in the characteristic polynomial $p(z) = \det(A - zI)$ where I is the identity matrix. A value that occurs only once is called *simple.* We write $A > 0$ ($A \geq 0$) and say A is positive (non–negative) if all its elements a_{ij} are positive (non–negative). A real vector x is *monotone* if $x_i \leq x_{i+1}$ for all $1 \leq i < n$ or if $x_i \geq x_{i+1}$ for all $1 \leq i < n$. A vector is *strictly monotone* if the above inequalities are strict.

We define the *Laplacian* of a matrix $A \geq 0$ to be $L = D - A$, where D is a diagonal matrix with $d_{ii} = \sum_{j=1}^{n} a_{ij}$. It is not hard to show that the eigenvalues of L are all non–negative, and that e is an eigenvector with eigenvalue zero. If A (and hence L) is irreducible, then zero is a simple eigenvalue of L. The second lowest eigenvalue of a Laplacian matrix L is called the *Fiedler value* and a corresponding eigenvector is called a *Fiedler vector* in recognition of the pioneering work of Miroslav Fiedler [5, 6]. More formally, the Fiedler value is the value attained by

$$\min_{x^t e = 0, x^t x = 1} x^t L x,$$

and a Fiedler vector is a vector which achieves the Fiedler value. When we refer to the Fiedler value/vector of A, we always mean the second lowest eigenvalue/vector of the Laplacian of A.

The cornerstone of our analysis is a classical result in matrix theory due to Perron and Frobenius [13]. The particular formulation below can be found on page 46 of [15].

THEOREM 2.1 (PERRON–FROBENIUS). *Let M be a real, non–negative matrix. If we define $\rho(M) = \max_i |\lambda_i(M)|$, then*

1. *$\rho(M)$ is an eigenvalue of M, and*
2. *there is a vector $x \geq 0$ such that $Mx = \rho(M)x$.*

The final concept we need is that of a *PQ–tree*, a data structure introduced by Booth and Lueker [4] to efficiently encode a set of permutations. A PQ–tree over a set

$U = \{u_1, u_2, \ldots, u_n\}$ is a rooted, ordered tree whose leaves are elements of U and whose internal nodes are distinguished as either P–nodes or Q–nodes. A PQ–tree is *proper* when the following three conditions hold:

1. Every element $u_i \in U$ appears precisely once as a leaf.
2. Every P–node has at least two children.
3. Every Q–node has at least three children.

Two PQ–trees are said to be equivalent if one can be transformed into the other by applying a sequence of the following two equivalence transformations:

1. Arbitrarily permute the children of a P–node.
2. Reverse the children of a Q–node.

Conveniently, the equivalence class represented by a PQ–tree corresponds to a set of permutations with precisely the properties we will need.

With these definitions we can now formulate a simple heuristic for the seriation problem that will motivate the remainder of the paper. This heuristic is at the heart of the more complex algorithms we will devise. We begin by constructing a simple penalty function whose value will be small when closely correlated items are close to each other, $g(\pi) = \sum_{(i,j)} f(i,j)(\pi_i - \pi_j)^2$. Unfortunately, minimizing g is difficult due to the discrete nature of the permutation (we believe it to be NP–hard, although we are not aware of any proof). Instead we approximate it by a function of continuous variables x_i that maintains much of the structure of g, $h(x) = \sum_{(i,j)} f(i,j)(x_i - x_j)^2$. Note that the value of h does not change if we add a constant to each x value, so we need to add a constraint like $\sum_i x_i = 0$. Also, the minimum value is trivially zero when all the x_i's are zero, so we need a second constraint like $\sum_i x_i^2 = 1$. The resulting minimization problem is now well defined.

$$(1) \qquad \text{Minimize } \sum_{(i,j)} f(i,j)(x_i - x_j)^2$$
$$\text{subject to: } \sum_i x_i = 0, \text{ and } \sum_i x_i^2 = 1.$$

The solution to this continuous problem can be used as a heuristic for sequencing. Merely construct the solution vector x, sort the elements x_i and sequence based upon their sorted order. Even if the problem is not well–posed, this approach generates an ordering that tries to keep highly correlated elements near each other. As mentioned above, this technique is being used for a variety of sequencing problems [2, 3, 8].

One reason this heuristic is attractive is that the minimization problem has an elegant solution. We can rewrite $h(x)$ as $x^t L x$ where L is the Laplacian matrix of the correlation function. The lowest value attainable by $x^t L x$ is achieved by a multiple of the first eigenvector, e. This vector is disallowed by the first constraint, and due to

pairwise orthogonality all other eigenvectors satisfy that constraint. Consequently, a solution to the constrained minimization problem is just a Fiedler vector.

3. The key theorem. Our main result is that a modification of the simple heuristic presented in §2 is actually an algorithm for well posed instances of the seriation problem. Completely proving this will require us to deal with the special cases of multiple Fiedler vectors and ties within the Fiedler vector, but the key theorem is the following.

THEOREM 3.1. *If A is an R–matrix then it has a monotone Fiedler vector.*

Proof. Our proof uses the Perron–Frobenius Theorem 2.1. The non–negative vector in that theorem will consist of differences between neighboring entries in the Fiedler vector of the Laplacian of A.

First define the matrix $S \in \mathbb{R}^{(n-1)\times n}$ as

$$S = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}.$$

Note that for any vector x, $Sx = (x_2 - x_1, \ldots, x_n - x_{n-1})^t$. Define $T \in \mathbb{R}^{n\times(n-1)}$ by

$$T = \begin{bmatrix} 0 & & & \\ 1 & & & \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

It is easy to verify that T is a right inverse of S, i.e. $ST = I_{n-1}$, and that $TS = I_n - ee_1^t$. Define $M = SLT$, where L is the Laplacian of A. We will show that x is an eigenvector of L other than e if and only if Sx is an eigenvector of M.

$$\begin{aligned} Lx &= \lambda x, \quad x \neq \alpha e \\ SLx &= \lambda Sx, \; x \neq \alpha e \\ SL(I - ee_1^t)x &= \lambda Sx, \; x \neq \alpha e \\ SLTSx &= \lambda Sx, \; x \neq \alpha e \\ My &= \lambda y, \text{ where } y = Sx \neq 0. \end{aligned}$$

The transformation from the second to the third line follows from $Le = 0$. Equivalence holds between all the above equations, so λ is an eigenvalue for both L and M for eigenvectors of L other than e. Hence the eigenvalues of M are the same as eigenvalues $2, \ldots, n$ of L, and the eigenvectors of M are differences between neighboring entries of the corresponding eigenvectors of L.

It is easily seen that $(SL)_{i,k} = -l_{i,k} + l_{i+1,k}$ for all i, k, so

$$\begin{aligned} m_{i,j} &= \sum_{k=1}^{n} (SL)_{i,k} T_{k,j} \\ &= \sum_{k=j+1}^{n} (-l_{i,k} + l_{i+1,k}) \\ &= \sum_{k=j+1}^{n} (a_{i,k} - a_{i+1,k}). \end{aligned}$$

Since by assumption, A is an R–matrix, $a_{i,k} \leq a_{i+1,k}$ for $i < k+1$, and therefore $m_{i,j} \leq 0$ for $i < j$. For $i > j$ we can use the fact that $\sum_{k=1}^{n} l_{i,k} = 0$ to obtain

$$\begin{aligned} m_{i,j} &= \sum_{k=j+1}^{n} (-l_{i,k} + l_{i+1,k}) \\ &= \sum_{k=1}^{j} (l_{i,k} - l_{i+1,k}) \\ &= \sum_{k=1}^{j} (-a_{i,k} + a_{i+1,k}). \end{aligned}$$

Again, from the R–matrix property we conclude that $m_{ij} \leq 0$ for $i > j$. Consequently, all the off-diagonal elements in M are non–positive.

Now let β be a value greater than $\max_i\{\lambda_i, m_{ii}\}$, where λ_i are the eigenvalues of M. Then $\tilde{M} = \beta I - M$ is non–negative with eigenvalues $\tilde{\lambda}_i = \beta - \lambda_i$. Also, $\tilde{M}$ and M share the same set of eigenvectors. By Theorem 2.1, there exists a non–negative eigenvector y of $\tilde{M}$ corresponding to the largest eigenvalue of $\tilde{M}$. But y is also an eigenvector of M corresponding to M's smallest eigenvalue. And this is just Sx, where x is a Fiedler vector of L. Since $y = Sx$ is non–negative, the corresponding Fiedler vector of L is non–decreasing and the theorem follows. □

THEOREM 3.2. *Let A be a pre–R matrix with a simple Fiedler value and a Fiedler vector with no repeated values. Let π_1 (π_2) be the permutation induced by sorting the values in the Fiedler vector in increasing (decreasing) order. Then A^{π_1} and A^{π_2} are R–matrices, and no other permutations of A produce R–matrices.*

Proof. First observe that if x is the Fiedler vector of A, then x^{π} is the Fiedler vector of A^{π}. So applying a permutation to A merely changes the order of the entries in the Fiedler vector. Now let π_* be a permutation such that A^{π_*} is an R–matrix. By Theorem 3.1 x^{π_*} is monotone since x is the only Fiedler vector. Since x has no repeated values, π_* must be either π_1 or π_2. □

Theorem 3.2 provides the essence of our algorithm for the seriation problem, but it is too restrictive. In particular, the Fiedler value must be simple and the Fiedler vector must contain no repeated values. We will show how to remove these limitations in the next section.

4. Removing the restrictions. Several observations about the seriation problem will simplify our analysis. First note that if we add a constant to all the correlation values the set of solutions is unchanged. Consequently, we can assume without loss of generality that the smallest value of the correlation function is zero. Note that subtracting the smallest value from all correlation values does not change whether or not the matrix is pre–R. In our algebraic formulation this translates into the following.

LEMMA 4.1. *Let A be an $n \times n$ matrix and let $\bar{A} = A - \alpha ee^t$. A vector x is a Fiedler vector of A iff x is a Fiedler vector of $\bar{A}$. So without loss of generality we can assume that the smallest off–diagonal entry of A is zero.*

Next observe that if A is reducible then the seriation problem can be decoupled. The irreducible blocks of the matrix correspond to connected components in the graph of the nonzero values of the correlation function. We can solve the subproblems induced by each of these connected components, and link the pieces together in an arbitrary order. More formally, we have the following lemma.

LEMMA 4.2. *Let A_k be the irreducible blocks of a pre–R matrix A, and let π_k be permutations that make the corresponding blocks become R–matrices. Then any permutation comprised by concatenating the π_k's will make A become an R–matrix. In terms of a PQ–tree, the π_k permutations should be children of a single P–node.*

With these preliminaries, we will now assume that the smallest off–diagonal value is zero and that the matrix is irreducible. As the following three lemmas and theorem show, this is sufficient to ensure that the Fiedler vector is unique.

LEMMA 4.3. *Let A be an $n \times n$ R–matrix with a monotone Fiedler vector x. If $\mathcal{J} = [r, s]$ is a maximal interval such that $x_r = x_s$, then for any $k \notin \mathcal{J}$, $a_{r,k} = a_{r+1,k} = \ldots = a_{s,k}$.*

Proof. We can without loss of generality assume x is non-decreasing since $-x$ is also a Fiedler vector. We will show that $a_{r,k} = a_{s,k}$ for all $k \notin \mathcal{J}$, and since A is an R–matrix then all elements between $a_{r,k}$ and $a_{s,k}$ must also be equal. Consider rows r and s in the equation $Lx = \lambda x$.

$$\sum_{k=1}^{n} (l_{s,k} - l_{r,k}) x_k = \lambda (x_s - x_r) = 0$$

Since L is the Laplacian, we know that $\sum_{k=1}^{n} l_{i,k} = 0$ for all i. We get

$$\begin{aligned} 0 &= \sum_{k=1}^{n} (l_{sk} - l_{rk})(x_r - x_k) \\ &= \sum_{k=1}^{r-1} \underbrace{(l_{s,k} - l_{r,k})}_{\geq 0} \underbrace{(x_r - x_k)}_{>0} + \sum_{k=s+1}^{n} \underbrace{(l_{s,k} - l_{r,k})}_{\leq 0} \underbrace{(x_r - x_k)}_{<0} \end{aligned}$$

where we have used the fact that x is non-decreasing. Because all terms in the sum are non-negative, all terms must be exactly zero. By assumption, $x_k \neq x_r$ for $k \notin \mathcal{J}$ and consequently $l_{r,k} = l_{s,k}$ for $k \notin \mathcal{J}$ and the result follows. □

The proof of the following lemma requires detailed algebra but is not fundamental to what follows. Consequently, it is relegated to the end of this section.

LEMMA 4.4. *Let A be an irreducible $n \times n$ R–matrix with $a_{n,1} = 0$. If $\mathcal{J} = [r, s] \neq [1, n]$ is an interval such that $a_{r,k} = a_{s,k}$ for all $k \notin \mathcal{J}$, then $x_r = x_{r+1} = \ldots = x_s$ for any Fiedler vector x.*

LEMMA 4.5. *Let A be an irreducible R–matrix with $a_{n,1} = 0$, and x a monotone Fiedler vector of A. If $\mathcal{J} = [r, s]$ is an interval such that $x_r = x_{r+1} = \ldots = x_s$, then for any Fiedler vector y, $y_r = y_{r+1} = \ldots = y_s$.*

Proof. First apply Lemma 4.3 to conclude that for any $k \notin \mathcal{J}$, $a_{r,k} = a_{r+1,k} = \ldots = a_{s,k}$. Since $x^t e = 0$, it follows that $\mathcal{J} \neq [1, n]$. Now use this in conjunction with Lemma 4.4 to obtain the result. □

THEOREM 4.6. *If A is an irreducible R–matrix with $a_{n,1} = 0$, then the Fiedler value λ_2 is a simple eigenvalue.*

Proof. We will assume that λ_2 is a repeated eigenvalue and produce a contradiction. Let x and y be two linearly independent Fiedler vectors with x non–decreasing. Define $z(\theta) = \cos(\theta)x + \sin(\theta)y$, with $0 \leq \theta \leq \pi$. Let θ^* be the smallest value of θ which makes $z_k = z_{k+1}$ for some k where $x_k \neq x_{k+1}$. Such a θ^* must exist since x and y are linearly independent.

By Lemma 4.5 the indices of any repeated values in x are indices of repeated values in y and $z(\theta)$. Coupled with the monotonicity of x, this implies that $z(\theta^*)$ is monotone. By Lemma 4.5 the indices of any repeated values in $z(\theta^*)$ must be repeated in x which gives the desired contradiction. □

All that remains is to handle the situation where the Fiedler vector has repeated values. As the following theorem shows, repeated values decouple the problem into pieces which can be solved recursively.

THEOREM 4.7. *Let A be a pre–R matrix with a simple Fiedler eigenvalue and Fiedler vector x. Suppose there is some repeated value β in x and define $\mathcal{I}$, $\mathcal{J}$ and $\mathcal{K}$ to be the indices for which*

1. *$x_i < \beta$ for all $i \in \mathcal{I}$,*
2. *$x_i = \beta$ for all $i \in \mathcal{J}$,*
3. *$x_i > \beta$ for all $i \in \mathcal{K}$.*

Then π is an R–matrix ordering for A iff π or its reversal can be expressed as (π_i, π_j, π_k), where π_j is an R–matrix ordering for the submatrix $A(\mathcal{J}, \mathcal{J})$ of A induced by $\mathcal{J}$, and π_i and π_k are the restrictions of some R–matrix ordering for A to $\mathcal{I}$ and $\mathcal{K}$, respectively.

Proof. From Theorem 3.1 we know that for any R–matrix ordering x^π is monotone,

so elements in $\mathcal{I}$ must appear before (after) elements from $\mathcal{J}$ and elements from $\mathcal{K}$ must appear after (before) elements from $\mathcal{J}$. By Lemma 4.3 any $k \notin \mathcal{J}$, $a_{ik} = a_{jk}$ for all $i, j \in \mathcal{J}$. Hence the orderings of elements inside $\mathcal{J}$ must be indifferent to the ordering outside of $\mathcal{J}$ and vice versa. Consequently, the R–matrix ordering of elements in $\mathcal{J}$ depends only of $A(\mathcal{J}, \mathcal{J})$. □

Algorithmically, this theorem means that we can break ties in the Fiedler vector by recursing on the submatrix $A(\mathcal{J}, \mathcal{J})$ where $\mathcal{J}$ corresponds to the set of repeated values. In the language of PQ–trees, the subtree generated by the recursion is combined with the rest of the permutation by a Q–node.

Proof of Lemma 4.4. First we recall that the Fiedler value is the value obtained by

$$\min_{x^t e=0, x^t x=1} x^t L x = \min_{x^t e=0, x^t x=1} \sum_{i>j} a_{i,j}(x_i - x_j)^2, \tag{2}$$

and a Fiedler vector is a vector which achieves this minimum. We note that if we replace A by a matrix that is at least as large on an elementwise comparison then $x^t L x$ cannot decrease for any vector x.

We consider $A(\mathcal{J}, \mathcal{J})$, the diagonal block of A indexed by $\mathcal{J}$. By the definition of an R–matrix, all values in $A(\mathcal{J}, \mathcal{J})$ must be at least as large as $a_{r,s}$. However, $a_{r,s}$ must be greater than zero. Otherwise, by the R–matrix property $a_{i,j} = 0$ for all $i \geq r$ and $j < s$ and for all $j \geq r$ and $i < s$. But then by the statement of the theorem $a_{i,j} = 0$ for all $i \geq s$ and $j < s$ and all $j \geq r$ and $j < s$ which would make the matrix reducible.

The remainder of the proof will proceed in two stages. First we will force all the off–diagonal values in $A(\mathcal{J}, \mathcal{J})$ be $a_{r,s}$ and show the result for this modified matrix. We will then extend the result to our original matrix.

Stage 1:

We define the matrix B to be identical to A outside of $B(\mathcal{J}, \mathcal{J})$, but all off–diagonal values of B within $B(\mathcal{J}, \mathcal{J})$ are set to $a_{r,s} = \alpha$. It follows from the hypotheses that B will be an R-matrix. We let L be the Laplacian of B and define $\delta = l_{i,i}$ for $i \in \mathcal{J}$. We note that by the R–matrix property, $\delta \leq (n-1)\alpha$.

We now define $\tilde{L} = L - (\delta + \alpha)I$ and consider the eigenvalue equation $\tilde{L}x = \tilde{\lambda}_2 x$. This matrix has the same eigenvectors as L with eigenvalues shifted by $\delta + \alpha$. Since $\tilde{l}_{ii} = \delta - (\delta + \alpha) = \alpha$ for $i \in \mathcal{J}$, all rows of $\tilde{L}$ in $\mathcal{J}$ are identical. Consequently, either all elements of x in $\mathcal{J}$ are equal, or $\tilde{\lambda}_2 = 0$ (which is equivalent to $\lambda_2 = \delta + \alpha$). We will show that irreducibility and $a_{n1} = 0$ implies $\lambda_2 \neq \delta + \alpha$, which will complete the proof of **Stage 1.**

We assume $\lambda_2 = \delta + \alpha$ and look for a contradiction. We introduce a new matrix $\hat{B}$

as follows

$$\hat{b}_{i,j} = \begin{cases} b_{i,j} & \text{if } i < r \text{ and } j < r, \\ b_{i,j} & \text{if } i > s \text{ and } j > s, \\ \alpha & \text{otherwise.} \end{cases}$$

Since B is an R–matrix, $\hat{B}$ is at least as large as B elementwise, so $\lambda_2(\hat{B}) \geq \lambda_2(B)$. We define the vector $\hat{y}$ by

$$\hat{y}_i = \begin{cases} -(n-s), & \text{if } i < r, \\ 0, & \text{if } r \leq i \leq s, \\ r-1, & \text{if } i > s, \end{cases}$$

and $\hat{x}$ to be the unit vector in the direction of $\hat{y}$. We note that $\hat{x}^t e = 0$, and that $\hat{x}^t \hat{L} \hat{x} = n\alpha$ where $\hat{L}$ is the Laplacian of $\hat{B}$. We have the following chain of inequalities.

$$\lambda_2 = \min_{x^t e = 0, x^t x = 1} x^t L x \leq \hat{x}^t L \hat{x} < \hat{x}^t \hat{L} \hat{x} = na. \tag{3}$$

The last inequality it strict since $\hat{b}_{n,1} = \alpha$ while $b_{n,1} = 0$ and $(\hat{x}_n - \hat{x}_1)^2 > 0$.

If $\lambda_2 = \delta + \alpha$ then we can combine an inequality due to Fiedler [5],

$$\lambda_2 \leq \frac{n}{n-1} \min_i l_{ii},$$

with the observation that $\min_i l_{i,i} \leq \delta$ to obtain $\lambda_2 \leq \frac{n}{n-1}\delta \leq \delta + \alpha = \lambda_2$. This can only be true if equality holds throughout, implying that $\delta = (n-1)\alpha$ and $\lambda_2 = n\alpha$. But this contradicts (3), so $\lambda_2 \neq \delta + \alpha$ and the proof of **Stage 1** is complete.

Stage 2:

We will now show that A and B have the same Fiedler vectors. Since A is elementwise at least as large as B, for any vector z, $z^t L_A z \geq z^t L_B z$ where L_A and L_B are the Laplacians of A and B respectively. From **Stage 1** we know that any Fiedler vector of B satisfies $x_r = x_{r+1} = \ldots = x_s$. In this vector $(x_i - x_j) = 0$ for $i, j \in \mathcal{J}$ so the contribution to the sum in (2) from $B(\mathcal{J}, \mathcal{J})$ is zero. But this contribution will also be zero when applied to $A(\mathcal{J}, \mathcal{J})$, and since A and B are otherwise identical, a Fiedler vector of B gives an upper bound for the Fiedler value of A; that is, $\lambda_2(A) \leq \lambda_2(B)$. It follows that the Fiedler vectors of B are also the Fiedler vectors A. □

5. A spectral algorithm for the seriation problem. We can now bring all the preceding results together to produce an algorithm for well posed instances of the ordering problem. Specifically, given a well posed correlation function we will generate all consistent orderings. Given a pre–R matrix, our algorithm constructs a PQ–tree for the set of permutations which produce an R–matrix. This **Spectral–Sort** algorithm is presented in Fig. 1.

```
    Input:   A, an n × n pre–R matrix
             U, a set of indices for the rows/columns of A
    Output:  T, a PQ–tree which gives the set of all permutations π
             such that A^π is an R–matrix
    begin
(1)   α := min_{i≠j} a_{i,j}
(1)   A := A − αee^t
(2)   {A^1, ..., A^k} := the irreducible blocks of A
(2)   {U^1, ..., U^k} := the corresponding index sets
(2)   if k > 1
(2)      for j := 1 : k
(2)         T^j := Spectral–Sort(A^j, U^j)
(2)      end
(2)      T := P–node(T^1, T^2, ..., T^k)
      else
(3)      if (n = 1)
(3)         T := u_1
(3)      else if (n = 2)
(3)         T := P–node(u_1, u_2)
         else
(4)         x := Fiedler vector for L(A)
(4)         sort x
(5)         t := number of distinct values in x
(5)         for j := 1 : t
(5)            V^j := indices of elements in x with jth value
(5)            T^j := Spectral–Sort(A(V^j, V^j), V^j)
(5)         end
            T := Q–node(T^1, ..., T^t)
         end
      end
   end
```

Fig. 1. Algorithm **Spectral-Sort.**

Let us prove that the algorithm is correct. Step (1) is justified by Lemma 4.1, and requires time proportional to the number of nonzeros in the matrix. The identification of irreducible blocks in step (2) can be performed with a breadth–first or depth–first search algorithm, also requiring time proportional to the number of nonzeros. Combining the

permutations of the resulting blocks with a P–node is correct by Lemma 4.2.

Step (3) handles the boundary conditions of the recursion, while in step (4) the Fiedler vector is computed and sorted. If there are no repeated elements in the Fiedler vector then the Q–node for the permutation is correct by Theorem 3.2. Step (4) is the dominant computational step and we will analyze its run time below. The recursion in step (5) is justified by Theorem 4.7.

Note that this algorithm produces a tree whether A is pre–R or not. To determine whether A is pre–R, you can apply one of the generated permutations. If the result is an R–matrix then all permutations in the PQ–tree will solve the seriation problem, otherwise it is not well–posed.

A formal complexity analysis of this algorithm is not possible without knowing the complexity of computing the Fiedler vector. Unfortunately, we are not aware of any published results on the formal computational complexity of eigenvector calculations. It is worth noting that the set of solutions to the seriation problem is only a function of the dominance relationships among the values. That is, we can modify correlation values without changing the set of solutions as long as all equalities and inequalities are preserved. This allows us to assume without loss of generality that all the values in the matrix A are small non–negative integers, less than n^2. With this simplification the computational complexity is likely to be tractable, but to our knowledge it remains an open problem.

Despite the lack of formal results, the numerical analysis community has a variety of efficient techniques for computing eigenvectors. To calculate eigenvectors corresponding to the few highest or lowest eigenvalues (like the Fiedler vector) the method of choice is known as the Lanczos algorithm. This is an iterative algorithm in which the dominant cost is a matrix–vector multiplication. The algorithm generally converges in many fewer than n iterations, often only $O(\sqrt{n})$ [12]. However, a careful analysis reveals a dependence on the difference between the distinct eigenvalues. In any case, the form of this analysis is quite different from the notion of computational complexity used in the theoretical computer science community.

However, if we assume that $O(\sqrt{n})$ iterations of a Lanczos algorithm are sufficient to compute the Fiedler vector, then the most expensive steps in algorithm **Spectral–Sort** are the generation and sorting of the eigenvector. If m is the number of nonzeros in the matrix, each matrix–vector product requires $O(m)$ work. Sorting the Fiedler vector requires $O(n \log n)$ work. If there are no repeated elements in the Fiedler vector, the total complexity of algorithm **Spectral–Sort** is $O(\sqrt{n}m + n \log n) = O(\sqrt{n}m)$ since we will assume $m \geq n$. In the worst case we have to recurse n times, giving a complexity of $O(n^{1.5}m)$.

We are not aware of any published algorithms for the well posed seriation problem,

but we believe fast algorithms for it exist. The approach of Booth and Lueker [4] for the consecutive ones problem may be extendible to this setting, possibly yielding a linear time algorithm. However, this type of approach will provide minimal insight into the more realistic cases where data errors make for problems which are not well posed.

6. Related ordering problems.

6.1. The consecutive ones problem. Ordering an R–matrix is closely related to the *consecutive ones problem.* We say that a $(0,1)$–matrix C has the *consecutive ones property* if there exists a permutation matrix Π such that for each column in ΠC, all the ones form a consecutive sequence.[2] A matrix which has this property without any rearrangement (i.e. $\Pi = I$) is in Petrie form[3] form and is called a P–matrix. Analogous to R–matrices, we say a matrix with the consecutive ones property is *pre–P.* The consecutive ones problem is: Given a pre–P matrix C, find a permutation matrix Π such that ΠC is a P–matrix.

There is a close relationship between P–matrices and R–matrices. The following results are due to D.G. Kendall and are proved in [10] and [16].

LEMMA 6.1. *If C is a P–matrix, then $A = CC^t$ is an R–matrix.*

LEMMA 6.2. *If C is pre–P and $A = CC^t$ is an R–matrix, then C is a P–matrix.*

THEOREM 6.3. *Let C be a pre–P matrix, let $A = CC^t$, and let Π be a permutation matrix. Then ΠC is a P–matrix if and only if $\Pi A \Pi^t$ is an R–matrix.*

This theorem allows us to use algorithm **Spectral–Sort** to solve the consecutive ones problem. First construct $A = CC^t$, and then apply our algorithm to A. Now act one of the permutations generated by the algorithm to C. If the result is a P–matrix then all the permutations produce consecutive ones orderings. If not, then C is not pre–P. If we again assume that the Lanczos algorithm can compute the Fiedler vector in $(\sqrt{n})$ iterations, we can exploit the product form of C to obtain a bound on running time. Since the Lanczos algorithm only requires matrix–vector multiplications, we can avoid explicitly forming A first multiply by C^t and then by C. Using this approach, if C has n columns and k ones, the cost of generating a Fiedler vector of A is $O(k\sqrt{n})$. Consequently, if there are no ties in the Fiedler vector, the run time for our algorithm is $O(\sqrt{n}k)$. With ties, in the worst case the algorithm requires $O(n^{1.5}k)$ time. This is not competitive with the linear time algorithm for this problem due to Booth and Lueker [4]. However, unlike their approach our **Spectral–Sort** algorithm degrades gracefully in the presence of errors. For most problem instances, only $O(1)$ recursion

[2] Some authors define this property in terms of rows instead of columns.

[3] Sir William M. F. Petrie was a renowned archeologist who laid the foundation of mathematical methods for seriation in the 1890's.

steps will be required and then our spectral algorithm is just a factor $O(\sqrt{n})$ slower than the optimal algorithm for well–posed problems.

Several other combinatorial problems have been shown to be equivalent to the consecutive ones problem. Among these are recognizing interval graphs [4, 7] and finding dense envelope orderings of matrices [4].

6.2. Q–matrices (unimodal matrices). A *unimodal* sequence is a sequence that is first non-decreasing, then non-increasing. Define a *Q–matrix* to be a matrix where all the columns are unimodal. If there exists a row permutation Π such that ΠC is a Q–matrix, then we say C is pre–Q.

Let the *circle product* of two matrices A and B be defined by

$$(A \circ B)_{ij} = \sum_k \min(a_{ik}, b_{kj}).$$

Note that P–matrices are just a special case of Q–matrices, and that the circle product is equivalent to matrix product for $(0,1)$–matrices.

There is a relation between P–matrices and R–matrices analogous to the one between Q–matrices and R–matrices. The following results were first proved in [11] (a good reference on P–, Q–, and R–matrices is [16]).

LEMMA 6.4. *If C is a Q–matrix, then $A = C \circ C^t$ is an R–matrix.*

LEMMA 6.5. *If C is pre–Q and $A = C \circ C^t$ is an R–matrix, then C is a Q–matrix.*

THEOREM 6.6. *Let C be a pre–Q matrix, let $A = C \circ C^t$, and let Π be a permutation matrix. Then ΠC is a Q–matrix if and only if $\Pi A \Pi^t$ is an R–matrix.*

This result allows us to use our **Spectral–Sort** algorithm to identify and order Q–matrices in a manner precisely analogous to our solution of the consecutive ones problem above. The only real difference is that we cannot use the product form of A to simplify the complexity.

7. Open problems and future work. We hope that the results in this paper stimulate further work on spectral methods for discrete problems. One obvious direction to investigate is the quality of spectral algorithms as approximation algorithms for ill–posed problems. These are likely to be specialized for different applications and different metrics of solution quality.

If, as we hope, continuous techniques are to play an important role in addressing combinatorial problems, then better formal methods need to be developed for analyzing their computational complexity. In particular, we would very much like to know the complexity of calculating eigenvectors, particularly the Fiedler vector.

We also suspect that there are additional combinatorial problems which can be solved by spectral techniques. We hope our results will encourage other researchers to look for them.

Acknowledgements. We are indebted to Robert Leland for ennumerable discussions about spectral techniques, to Sorin Istrail for his insights into the consecutive ones problem, to David Greenberg for his experimental testing of our approach on simulated genomic data, and to Nabil Kahale for showing us how to simplify the proof of Theorem 3.1.

REFERENCES

[1] P. Arabie and L. J. Hubert. Combinatorial data analysis. *Annual Review of Psychology*, 43:169–203, 1992.

[2] S. T. Barnard, A. Pothen, and H. D. Simon. A spectral algorithm for envelope reduction of sparse matrices. Technical Report CS–93–49, NASA Ames Research Center, 1993.

[3] M. Berry, 1994. Personal communication.

[4] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ–tree algorithms. *JCSS*, pages 333–379, 1976.

[5] M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. Journal*, 23:298–305, 1973.

[6] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech. Math. Journal*, 25:619–633, 1975.

[7] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pac. J. Math.*, 3:835–855, 1965.

[8] D. S. Greenberg and S. C. Istrail. Physical mapping with STS hybridization: opportunities and limits. Technical report, Sandia National Labs, 1994.

[9] D. Halperin. Musical chronology by seriation. *Computers and the Humanities*, 28:13–18, 1994.

[10] D. G. Kendall. Incidence matrices, interval graphs and seriation in archaeology. *Pac. J. Math.*, 28(3):565–570, 1969.

[11] D. G. Kendall. Abundance matrices and seriation in archaeology. *Zeitschrift für Wahrscheinlichkeitstheorie*, 17:104–112, 1971.

[12] B. Parlett and D. Scott. The Lanczos algorithm with selective orthogonalization. *Math. Comp.*, 33:217–238, 1979.

[13] O. Perron. Zur Theorie der Matrizen. *Math. Ann.*, pages 248–263, 1907.

[14] W. S. Robinson. A method for chronologically ordering archaeological deposits. *American Antiquity*, 16(4):293–301, 1951.

[15] R. S. Varga. *Matrix Iterative Analysis*. Prentice–Hall, 1962.

[16] E. M. Wilkinson. Techniques of data analysis and seriation theory. In *Technische und Naturwissenschaftliche Beiträge Zur Feldarchäologie*, pages 1–142. Rheinland–Verlag, 1974.