

# Consecutive ones property and spectral ordering

Niko Vuokko\*

## Abstract

A 0-1 matrix where in each column the 1s occur consecutively is said to have the consecutive ones property. This property and its approximations are important in a variety of applications, e.g., in DNA analysis and paleontology. Checking whether the matrix rows can be permuted so that this property holds can be done in polynomial time. However, an exact solution rarely exists in applications. Thus, methods that produce good approximations with a minimal number of 0s between 1s in the columns are needed. The spectral ordering method gives a solution to the consecutive ones problem and has empirically been shown to work well in other cases too. This paper constructs a theoretical basis for the connection between the consecutive ones property and spectral ordering. We represent the approximation problem as a problem of minimizing a certain matrix norm. We show how the spectral ordering method affects this norm and show that the method is optimal within a class of algorithms. We analyze also popular normalization schemes for spectral ordering and give our recommendation on their use for the C1P problem. We use these theoretical results to explain the experimental performance of different approximate methods and their differences.

Keywords: zero-one matrices, spectral ordering, consecutive ones property, combinatorial optimization, Frobenius norm

## 1 Introduction

Discrete optimization involves combinatorial problems that require making an optimal choice from a finite set of alternatives. The powerful tools of calculus are not usually available for use in these problems and so the problems are regularly very difficult to solve exactly and many common problems of the field are **NP**-hard. Therefore approximate methods are needed, but for the same reasons it is sometimes difficult to justify and analyze these approximations. When this analysis is possible, it is often done through the use of calculus and an analogy to a simpler continuous case. In this paper, we adapt the discrete optimization problem of consecutive ones to a problem in calculus and show how this can be used to analyze the problem and the

methods used for approximately solving it.

Suppose that  $\mathbf{A}$  is a 0-1 matrix. We define  $\mathbf{A}$  to have the *consecutive ones property* (abbreviated *C1P*), if the non-zero entries occur consecutively in each of the columns of  $\mathbf{A}$ . We call  $\mathbf{A}$  a *pre-C1P* matrix if it is possible to order  $\mathbf{A}$ 's rows so that the result has C1P. C1P has its roots in interval graphs and their incidence matrices, but the need for exploring the property first arose from the problem of aligning and reconstructing DNA sequences, see [22,24]. A good introduction to the theory of C1P can be found in [10] or [15].

In practical applications of C1P it is very rare to encounter data that actually has the C1P or pre-C1P property. Therefore the actual practical problem is to manipulate the data so that the result adheres as much as possible to the C1P property. These kinds of approximations and especially their quality are very important for the practical applications of C1P, because their performance and applicability is primarily driven by the approximation quality in the C1P problem. Thus it is rather disappointing that this approximation problem has received little attention among researchers whereas the pure C1P and pre-C1P properties are quite well understood.

For the approximation problem, the most common data manipulations allowed by applications are permutations of rows or columns, although in some cases also direct changes to data values are allowed. In the following, the term “*maximizing the C1P property*” is used when referring to this approximation problem. We will give two related, but different, exact definitions for this problem in Section 2.

In [1] a new way for finding an exact C1P-permutation of matrix rows was introduced. In the paper, the authors prove that *if* such a permutation exists then a method called spectral ordering will find it. *Spectral ordering* (also called *spectral sorting*) is a method used for serially ordering objects based on their similarities with each other. The method was first introduced in [2]. Since then it has mostly been used as an aid for spectral clustering, but it has gained a status as a method of its own too. For a good modern introduction to the method see [5].

Even though the results published in [1] apply only to the case of pre-C1P matrices, the spectral ordering

\*HIIT, Helsinki University of Technology, Finland

method has also been used for maximizing the C1P property in general, see [9]. Despite its distinct success, this practice has always been purely heuristic without any theoretical justification beyond the results of Atkins et al. in [1].

The main contribution of this paper is that we theoretically verify and justify the connection between C1P and spectral ordering. We will prove that maximizing the C1P property, that is, consecutiveness of non-zeros in a 0-1 matrix  $\mathbf{A}$ , is equivalent to minimizing the Frobenius norm of a certain matrix derived from  $\mathbf{A}$ . As a main result, we prove that the spectral ordering method minimizes the Frobenius norm of this derived matrix' strongest principal component. Thus, spectral ordering is the optimal method for solving the problem using only 1-dimensional approximations. As a supporting, simple result we also prove that solving the problem with higher dimensional approximations is **NP**-hard. Finally the original result of [1] in the case of pre-C1P matrices is shown to be a special case of our analysis and results.

The rest of this paper is organized as follows: Section 2 defines the concepts and problems at hand in more detail. Section 3 describes the theoretical results devised in the paper and analyses their relation to the C1P problem. Section 5 gives explanations on how the theoretical results of Section 3 influence the practical results given by various variants of the spectral ordering algorithm. Finally Section 6 summarizes the work and gives suggestions on future directions for research.

**1.1 Related work** The first polynomial algorithm for deciding whether a matrix is pre-C1P was constructed in [10]. In [3] a new data structure called *PQ-tree* was introduced that can be created in linear time and manipulated to give all the permutations that put a matrix into C1P form. This work was then extended in [16] using a modified version of PQ-trees called *PQR-trees* (not to be mixed with Prediction of Query Runtime trees in database theory). These trees can function even with non-pre-C1P matrices although performance is not linear anymore. Also other schemes to test for the pre-C1P property have been suggested, see e.g. [13].

PQ- and PQR-trees are commonly used to construct structures that can be used to find an underlying C1P form of a matrix. Neither of the structures however provide a way to maximize the C1P property of an arbitrary matrix to any degree. For that matter, the consecutive ones property itself has been examined extensively, but the more important practical problem of approximately maximizing the C1P property has received limited attention among researchers.

In [18] a branch and bound approach was used for maximizing the C1P property in order to solve a set

cover problem for railway network optimization. Solving the set cover problem was done by first permuting the data to almost-C1P form after which the rest of the problem becomes significantly simpler.

Combining partial orders to order a set of objects was studied in [20]. This technique can also be used to approximately maximize the C1P property when it is expressed as a serialization problem.

Spectral ordering is used for maximizing the C1P property in [9] to sort excavation sites chronologically using fossils found from each site as data. In paleobiology the term "Lazarus event" is used for the occurrence of having zeros in between ones i.e. species disappearing and reappearing in time.

Flipping entries from zero to one and vice-versa by a minimal amount to reach a "banded" and therefore a C1P form for a matrix was recently studied in [11]. In the paper spectral ordering and spanning tree methods were used to maximize the C1P property.

## 2 Definition of concepts

We will now define the objective of maximizing the C1P property in detail and provide two related metrics for evaluating optimization performance. The first metric calculates the sum of the total number of zeros between the first and last non-zero entry in each of the columns. This metric will be called  $m_z$ . The second metric  $m_c$  sums up the number of consecutive series of zeros between the first and last non-zero entry in each column. Therefore both metrics look at each column in a matrix independently and then simply add up the scores for each column to get the metric value for the whole matrix.

Supposing  $\mathbf{v} = (01100101000111)^T$  we get metric values  $m_z(\mathbf{v}) = 6$  and  $m_c(\mathbf{v}) = 3$ . There is no general rule on which metric to choose as it depends on the application. In this paper, we will concentrate in analyzing  $m_c$ , although in the experiments we will give results also for the  $m_z$  metric as it gives an interesting insight to the methods' differences. Our goal thus is to sort the rows of  $\mathbf{A}$  so that  $m_c(\mathbf{A})$  is minimized. More precisely, if  $\mathbf{A}$  is an  $m \times n$  matrix then our goal is to find the  $m \times m$  permutation matrix  $\mathbf{P}^* = \arg \min_{\mathbf{P}} m_c(\mathbf{PA})$ .

**2.1 Spectral ordering** Next we will introduce the spectral ordering method analyzed in this work. For ease of presentation the eigenvectors and singular vectors corresponding to the  $i^{th}$  largest (smallest) eigenvalue or singular value will be called the  $i^{th}$  largest (smallest) eigenvectors or singular vectors.

In the spectral ordering method first an  $m \times m$  similarity matrix  $\mathbf{W}$  is constructed from the data matrix  $\mathbf{A}$ . Then let the diagonal matrix  $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_m)$

contain the row sums of  $\mathbf{W}$ . Matrix  $\mathcal{L} = \mathbf{D} - \mathbf{W}$  is called the *Laplacian* of  $\mathbf{W}$ . It is a central concept in spectral graph theory, see e.g. [17]. In this paper we will use only the dot product of the matrix rows as their similarity measure and thus  $\mathbf{W} = \mathbf{A}\mathbf{A}^T$  in the following. From this it follows that the values  $d_i$  can in this situation be interpreted as the amount of interaction and constraints each row of  $\mathbf{A}$  has with the other rows.

It can be proven that the Laplacian  $\mathcal{L}$  is positive semidefinite. This implies that it has an eigenvalue decomposition  $\mathcal{L} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T$  and also that all its eigenvalues are non-negative. Furthermore it is easily seen that  $\mathcal{L}\mathbf{1} = \mathbf{0}$  and so  $\mathbf{1} = (1, 1, \dots, 1)^T$  is an eigenvector of the Laplacian corresponding to its smallest eigenvalue 0. The second smallest eigenvector of  $\mathcal{L}$  is called the *Fiedler vector*, see [7].

The spectral ordering method calculates a permutation matrix  $\mathbf{P}$  that sorts the Fiedler vector of the similarity matrix' Laplacian  $\mathcal{L}$  to a monotonous order. These orderings are called the *spectral orders* of the data. Matrix  $\mathbf{PA}$ , which is simply the data  $\mathbf{A}$  with its rows permuted by  $\mathbf{P}$ , is called the spectrally ordered version of  $\mathbf{A}$ . Notice that the Fiedler vector of  $\mathbf{PA}$  is then by definition in monotonous order.

Spectral clustering and ordering results are generally concluded to improve when using some normalization scheme, see e.g. [5, 21]. Mainly two different normalizations are in use. First one is a symmetric normalization SYM that is related to e.g. graph commute times, see [14]. NCUT is a more popular scheme introduced in [19] that has strong connections to random walks. In contrast, the unnormalized method described above is sometimes referred to as RATIOCUT, see [12]. In both normalizations the Laplacian matrix is normalized after which the algorithm continues as usual. In symmetric normalization  $\mathcal{L}$  is replaced with  $\mathcal{L}_{\text{sym}} = \mathbf{D}^{-1/2}\mathcal{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ . NCUT normalization uses  $\mathcal{L}_{\text{N}} = \mathbf{D}^{-1}\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$  instead. The eigenvectors of these two normalized Laplacians are quite similar:

$$\begin{aligned} \mathcal{L}_{\text{N}}\mathbf{x} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{x} &\iff (\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{D}\mathbf{x} \\ &\stackrel{\mathbf{x}=\mathbf{D}^{-1/2}\mathbf{z}}{\iff} \mathcal{L}_{\text{sym}}\mathbf{z} = (\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})\mathbf{z} = \lambda\mathbf{z} \\ (2.1) \qquad \qquad \qquad &\iff \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{z} = \zeta\mathbf{z}, \end{aligned}$$

where  $\zeta = 1 - \lambda$ . The smallest eigenvalue of these two normalized Laplacians is also 0 as can be seen from Equation (2.1). Also the corresponding eigenvector for  $\mathcal{L}_{\text{N}}$  is  $\mathbf{1}$  just as with  $\mathcal{L}$ , but for  $\mathcal{L}_{\text{sym}}$  it is  $\mathbf{D}^{1/2}\mathbf{1} = (\sqrt{d_1}, \sqrt{d_2}, \dots, \sqrt{d_n})$ . Furthermore the Fiedler vector of  $\mathcal{L}_{\text{N}}$  is actually the Fiedler vector of  $\mathcal{L}_{\text{sym}}$  multiplied by  $\mathbf{D}^{-1/2}$  or equivalently the second largest eigenvector of  $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$  multiplied by  $\mathbf{D}^{-1/2}$ .

### 3 Analysis of the C1P maximization problem

The goal of this section is to first give an analytical representation of the C1P maximization problem. The resulting matrix formula is then manipulated and opportune parts of it are decomposed by their spectrum. As a result of this development we get a representation of the original problem in which the connection between the problem and spectral structures of related matrices becomes apparent. This connection is then further refined with spectral ordering in mind in the next section.

**3.1 Problem representation** To help in the analysis we introduce an  $(m-1) \times m$  matrix  $\mathbf{S}$  first used in [1] for the problem. Matrix  $\mathbf{S}$  is defined as a sort of differential operator,

$$\mathbf{S} = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}_{(m-1) \times m}.$$

The following lemma formalizes the C1P maximization problem as a calculus formula.

LEMMA 3.1. Let  $\widehat{\mathbf{A}}$  denote matrix  $\mathbf{A} = (a_{ij})$  with an extra zero-filled row added to the top and to the bottom:

$$\widehat{\mathbf{A}} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \\ 0 & 0 & \dots & 0 \end{pmatrix}.$$

Minimizing the  $m_c$  metric on a 0-1 matrix  $\mathbf{A}$  is equivalent to finding the row permutation  $\mathbf{P}$  that minimizes the Frobenius norm of matrix  $\mathbf{SP}\widehat{\mathbf{A}}$  and keeps the extra zero rows in place. More precisely, the solution to maximizing the C1P property is

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} m_c(\mathbf{PA}) = \arg \min_{\mathbf{P}} \|\mathbf{SP}\widehat{\mathbf{A}}\|_{\text{F}}^2.$$

(Here  $\mathbf{P}$  is written to act on both  $\mathbf{A}$  and  $\widehat{\mathbf{A}}$  since it has no effect on the two extra rows.)

*Proof.* The difference matrix  $\mathbf{SPA}$  (row differences after  $\mathbf{A}$ 's rows are permuted) has entry values -1, 0 or 1. The non-zero elements exist on locations where a series of same values ends and another begins. Each consecutive string of ones creates two non-zero entries to a column of  $\mathbf{SPA}$ , a 1 where the string begins and a -1 where it ends.

The only exception to this rule are the top and bottom row of  $\mathbf{A}$ . However replacing  $\mathbf{A}$  with  $\widehat{\mathbf{A}}$  fixes

this problem if we also force the permutation  $\mathbf{P}$  to keep these two extra rows in place. After this modification it is easily seen that the number of non-zero entries in  $\mathbf{SPA}$  is twice the number of consecutive strings of 1s and so equal to  $m_c(\mathbf{PA}) + 2n$  (supposing  $\mathbf{A}$  has no columns with only zeros). For example, if  $\mathbf{PA}$  is in C1P form, then  $\mathbf{SPA}$  has exactly  $2n$  non-zero entries. This number of non-zero entries is clearly equal to  $\|\widehat{\mathbf{SPA}}\|_{\mathbf{F}}^2$ , because all the entries of  $\widehat{\mathbf{SPA}}$  are either -1, 0 or 1.  $\square$

REMARK 3.1. *The modification of  $\mathbf{A}$  into  $\widehat{\mathbf{A}}$  has little effect to any of the following analysis and its results. This is especially true for large row counts of  $\widehat{\mathbf{A}}$ . Therefore, for simplicity we will rather study the problem of solving*

$$(3.2) \quad \mathbf{P}^* = \arg \min_{\mathbf{P}} \|\mathbf{SPA}\|_{\mathbf{F}}^2$$

in the rest of the paper.

Based on Lemma 3.1 we can now concentrate in analyzing the connection between spectral ordering of  $\mathbf{A}$  and the Frobenius norm  $\|\mathbf{SPA}\|_{\mathbf{F}}^2$ . As a first step, we will insert the singular value decomposition  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  to Equation (3.2). Let  $k = \min(m, n)$  be the number of  $\mathbf{A}$ 's singular values so that, e.g.,  $\mathbf{U}$  is an  $m \times k$  matrix. We will also denote

$$\begin{aligned} \mathbf{U} &= (\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_k) = (u_{ij})_{m \times k}, \\ \mathbf{\Sigma} &= \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k) \end{aligned}$$

and give  $\mathbf{P}$  a dual meaning as a permutation of the set  $\{1, 2, \dots, m\}$ . By decomposing the Frobenius norm to parts by the columns we get

$$(3.3) \quad \begin{aligned} \|\mathbf{SPA}\|_{\mathbf{F}}^2 &= \left\| \mathbf{SPU}\mathbf{\Sigma}\mathbf{V}^T \right\|_{\mathbf{F}}^2 \stackrel{*}{=} \|\mathbf{SPU}\mathbf{\Sigma}\|_{\mathbf{F}}^2 \\ &= \sum_{j=1}^k \sigma_j^2 \|\mathbf{SP}\mathbf{u}_j\|_{\mathbf{F}}^2 \\ &= \sum_{j=1}^k \sigma_j^2 \sum_{i=1}^{m-1} (u_{\mathbf{P}(i+1),j} - u_{\mathbf{P}(i),j})^2. \end{aligned}$$

In the spot marked by \* we use the fact that Frobenius norm does not depend on the orthonormal basis used and therefore we may drop  $\mathbf{V}^T$  from the formula.

### 3.2 Computational complexity of the problem

The following lemma shows that finding the permutation  $\mathbf{P}^*$  that minimizes the value in Formula (3.3) is **NP**-hard in general case. This result is somewhat similar to previously proven results and presented here for completeness and clarity.

LEMMA 3.2. *Consider a real-valued matrix  $\mathbf{X} = (x_{ij})_{m \times k}$  and positive real numbers  $c_j$  for  $j = 1, 2, \dots, k$ . Finding the permutation  $\pi$  of the set  $\{1, 2, \dots, m\}$  that minimizes*

$$(3.4) \quad \sum_{j=1}^k c_j^2 \sum_{i=1}^{m-1} (x_{\pi(i+1),j} - x_{\pi(i),j})^2$$

is solvable in polynomial time when  $k = 1$  and **NP**-hard for  $k \geq 2$ .

*Proof.* Case  $k = 1$ : We may let  $c_1 = 1$ , write  $x_i = x_{i,1}$  and also visualize the elements  $x_i$  as separate nodes in a graph. Suppose that permutation  $\pi$  puts the elements of  $x$  into ascending order. Now the sum of Formula (3.4) can be seen as calculating the square length of a path in the graph traversing through each element  $x_i$  in  $\pi$ -order. If we draw this path as a diagram

$$x_{\pi(1)} \leq x_{\pi(2)} \leq \dots \leq x_{\pi(m)}$$

we notice that each of the intervals  $[x_{\pi(i)}, x_{\pi(i+1)}]$  will be included in at least one part of *any* path that traverses through all the points  $x_i$ . If the path traverses the points in  $\pi$ -order the intervals are included in exactly on part of the path. Thus the path length and Formula (3.4) are minimized by sorting the  $x_i$  in ascending (or any monotonous) order.

Case  $k \geq 2$ : First, the problem is easily seen to be equivalent (by logarithmic space reductions) with question "Does a  $\pi$  exist for which Formula (3.4)'s value is at most  $R \geq 0$ ?". Secondly, this equivalent decision problem is clearly in **NP**. The problem can now be seen as a travelling salesman problem in  $k$ -dimensional space using distance measure  $d(x, y) = \|x - y\|_2^2$ . This distance function is not a metric, but if  $x$  and  $y$  are binary vectors, the hamming distance of them equals  $d(x, y)$ . After this remark it is simple to reduce the *hamming travelling salesman problem* that was proven **NP**-complete in [6] to this decision problem. Thus the decision version of Lemma 3.2 is **NP**-complete.  $\square$

REMARK 3.2. *More precisely the problem of Lemma 3.2 is log-space equivalent with the travelling salesman problem that uses squared  $L_2$  norm as the distance function. Therefore the problem resides in complexity class  $\mathbf{FP}^{\mathbf{NP}}$ .*

We may view the C1P maximization problem as a travelling salesman problem not only in the space of principal components, but in binary  $\{0, 1\}^n$  space for matrix  $\mathbf{A}$ . This interpretation helps to give intuition to the problem and several C1P maximization methods are based on it. As we will soon see, spectral ordering solves the same travelling salesman problem using only one dimension as input.

However, the biggest difference between spectral ordering and other C1P maximization methods is that spectral ordering tries to solve the related travelling salesman problem in spectral space instead of the original binary space. Similar to any other case of spectral methods, this can be seen as a data-driven approach, since the original binary space is an arbitrary representation choice from the viewpoint of the data.

**REMARK 3.3.** *If  $\mathbf{A}$  is a pre-C1P matrix, spectral ordering solves the related travelling salesman problem optimally (as proven in [1]) by looking at only one principal component. This means that the cities in the travelling salesman problem of Lemma 3.2 above have to be located more or less along a straight line.*

**3.3 Trace-based analysis** We may analyze the Frobenius norm minimization problem also through using the connection  $\|\mathbf{M}\|_F^2 = \text{tr}(\mathbf{M}\mathbf{M}^T)$  between Frobenius norm and matrix trace. Using the linearity of the trace function we get

$$\begin{aligned} \|\mathbf{SPA}\|_F^2 &= \text{tr}(\mathbf{SPA}\mathbf{A}^T\mathbf{P}^T\mathbf{S}^T) \\ &= \text{tr}(\mathbf{SP}(\mathbf{D} - \alpha\mathbf{I} + \alpha\mathbf{I} - \mathcal{L})\mathbf{P}^T\mathbf{S}^T) \\ &= \text{tr}(\mathbf{SP}(\mathbf{D} - \alpha\mathbf{I})\mathbf{P}^T\mathbf{S}^T) + \\ (3.5) \quad &\text{tr}(\mathbf{SP}(\alpha\mathbf{I} - \mathcal{L})\mathbf{P}^T\mathbf{S}^T), \end{aligned}$$

where we let  $\alpha$  to be any positive real number. In the last form in Equation (3.5),  $\mathbf{D} - \alpha\mathbf{I}$  is a diagonal matrix. Therefore we can calculate that the first term of the form,

$$\begin{aligned} \text{tr}(\mathbf{SP}(\mathbf{D} - \alpha\mathbf{I})\mathbf{P}^T\mathbf{S}^T) &= \underbrace{\sum_{i=1}^m 2(d_i - \alpha)^2}_{\text{constant}} \\ &\quad - (d_{\mathbf{P}^{-1}(1)} - \alpha)^2 - (d_{\mathbf{P}^{-1}(m)} - \alpha)^2, \end{aligned}$$

depends only on two elements of  $\mathbf{D}$  placed to the top and bottom by  $\mathbf{P}$ . Furthermore, this quantity is an exact constant, if we compute Equation (3.5) for  $\hat{\mathbf{A}}$  (see Remark 3.1). Based on these notions and Equation (3.5), we decide that to minimize  $\|\mathbf{SPA}\|_F^2$ , it is enough to minimize the second term of the last form in Equation (3.5) and thus to solve

$$(3.6) \quad \mathbf{P}^* = \arg \min_{\mathbf{P}} \text{tr}(\mathbf{SP}(\alpha\mathbf{I} - \mathcal{L})\mathbf{P}^T\mathbf{S}^T).$$

In the next step, we need the eigenvalue decomposition  $\mathcal{L} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T$ , where  $\mathbf{R} = (\mathbf{r}_1\mathbf{r}_2\dots\mathbf{r}_k)$  and  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$  with eigenvalues  $\lambda_i$  sorted ascending. If we set  $\alpha > \max \lambda_i$ , we can rewrite Equa-

tion (3.6) as

$$\begin{aligned} \mathbf{P}^* &= \arg \min_{\mathbf{P}} \text{tr} \left( \mathbf{SP} \left( \sum_{i=1}^k (\alpha - \lambda_i) \mathbf{r}_i \mathbf{r}_i^T \right) \mathbf{P}^T \mathbf{S}^T \right) \\ (3.7) \quad &= \arg \min_{\mathbf{P}} \sum_{i=1}^k (\alpha - \lambda_i) \|\mathbf{SP}\mathbf{r}_i\|_2^2, \end{aligned}$$

where each multiplier  $\alpha - \lambda_i$  is positive. The most significant terms of the sum in Formula (3.7) are those corresponding to the smallest eigenvalues  $\lambda_i$  of the Laplacian  $\mathcal{L}$ . One should note however that the high significance of those terms in the sum is largely a heuristic claim, because the influence of the norm factors in the sum cannot be evaluated.

#### 4 Connection to spectral ordering

Following Lemma 3.2 we must now find a way to approximately minimize Formula (3.3) or Formula (3.7). Standard approximation algorithms for metric travelling salesman problems can't be used, because the distance function  $d(x, y) = \|x - y\|_2^2$  in Formula (3.3) does not fulfill the triangle inequality. Regardless, these algorithms can be used as heuristics and this approach was followed in [11].

Another approach to approximative minimization of the Frobenius norm is to look at only a part of its sum form letting the rest make up an error term. However, Lemma 3.2 shows us that it is not only **NP**-hard to find the optimal permutation  $\mathbf{P}^*$  of Lemma 3.1 in general, but that it is also **NP**-hard to gain optimal solutions for approximations of more than one dimension, i.e., using  $k \geq 2$  terms in the approximation. Therefore, we will use the positive case for  $k = 1$  in Lemma 3.2 to approximate a solution. This means choosing one single eigenvector leaving the rest aside, sorting it to monotonous order and ordering  $\mathbf{A}$ 's rows similarly. Accepting some uncertainty, we now (heuristically) conclude that the largest singular vectors of  $\mathbf{A}$  or similarly the smallest eigenvectors of  $\mathcal{L}$  contribute the most to formulas in (3.3) and (3.7). Therefore we will concentrate only in the two most important vectors in each case.

Now look at only the two first terms of the sum in Formula (3.7) that have the largest multipliers  $\alpha - \lambda_i$  and discard the rest. Interestingly, the first term in Formula (3.7) is actually zero since the smallest eigenvector of the Laplacian  $\mathcal{L}$  is the constant vector  $\mathbf{1} = (1, 1, \dots, 1)$  and thus  $\mathbf{SP}\mathbf{r}_1 = (0, 0, \dots, 0)$  regardless of the permutation  $\mathbf{P}$ .

This important observation leaves the second term of the sum in Formula (3.7) as the most influential term,

leaving us with a feasible problem:

$$\begin{aligned} \mathbf{P}^* &= \arg \min_{\mathbf{P}} \operatorname{tr} (\mathbf{S}\mathbf{P} (\alpha - \lambda_2) \mathbf{r}_2 \mathbf{r}_2^T \mathbf{P}^T \mathbf{S}^T) \\ &= \arg \min_{\mathbf{P}} \|\mathbf{S}\mathbf{P}\mathbf{r}_2\|_2^2. \end{aligned}$$

The solution  $\mathbf{P}^*$  to this minimization problem is the permutation that sorts the second smallest eigenvector  $\mathbf{r}_2$  of  $\mathcal{L}$ , which is exactly the solution given by the unnormalized spectral sorting algorithm. Thus, we have proved the following theorem.

**THEOREM 4.1.** *The spectral ordering algorithm gives the optimal solution to minimizing the strongest term of Formula (3.7).  $\square$*

By connecting the pieces from Lemma 3.2 and Theorem 4.1 we can call the spectral ordering method to give an optimal feasible solution to the C1P maximization problem in its own class of approach to the problem:

**COROLLARY 4.1.** *The spectral ordering algorithm solves the C1P maximization problem optimally for the strongest principal component of SPA. Any solution for two or more components is NP-hard to find.*

**4.1 C1P and normalized spectral ordering** Next we will move on to analyzing the normalized versions of spectral ordering. As one might expect, they relate to maximizing the C1P property for, not the original data matrix, but for modified versions of it. The significant practical problems arising from this difference are discussed later in the section. Anyhow, differing normalizations have a very strong position in many applications of spectral methods. Already this fact makes their analysis in the context of spectral ordering interesting. Furthermore, the analysis can also be used to gain additional insight into the effects of normalization in spectral applications.

For the NCUT-normalized modification we need to start by also modifying our objective. This is done by normalizing the data matrix  $\mathbf{A}$  to  $\mathbf{D}^{-1}\mathbf{A}$ , a normalization that may not always be desired. We will also let matrix  $\mathbf{D}^{-1/2}\mathbf{A}$  have the singular value decomposition  $\mathbf{F}\Psi\mathbf{G}^T$ , where  $\mathbf{F} = (\mathbf{f}_1 \mathbf{f}_1 \dots \mathbf{f}_r)$ ,  $\Psi = \operatorname{diag}(\psi_1, \psi_2, \dots, \psi_r)$  and the singular values  $\psi_i$  are sorted descending by their absolute values. This implies the eigenvalue decomposition  $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{A}^T\mathbf{D}^{-1/2} = \mathbf{F}\Psi^2\mathbf{F}^T$  and thus by Section 2 the NCUT-normalized Laplacian  $\mathcal{L}_N$  has eigenvalues  $1 - \psi_i^2$  with corresponding eigenvectors  $\mathbf{D}^{-1/2}\mathbf{f}_i$ . With these preparations we use similar manipulations than with Equation (3.3) to

calculate

$$\begin{aligned} \|\mathbf{S}\mathbf{P}\mathbf{D}^{-1}\mathbf{A}\|_{\mathbb{F}}^2 &= \|\mathbf{S}\mathbf{P}\mathbf{D}^{-1/2} (\mathbf{D}^{-1/2}\mathbf{A})\|_{\mathbb{F}}^2 \\ (4.8) \qquad &= \sum_{i=1}^r \psi_i^2 \|\mathbf{S}\mathbf{P}\mathbf{D}^{-1/2}\mathbf{f}_i\|_2^2. \end{aligned}$$

As described in Section 2, the eigenvector  $\mathbf{D}^{-1/2}\mathbf{f}_1$  of the Laplacian  $\mathcal{L}_N$  corresponding to eigenvalue 0 is the constant vector  $\mathbf{1}$  and we can once again discard the first term in Equation (4.8). Thus the strongest term of the sum in (4.8) is minimized by sorting the Fiedler vector of  $\mathcal{L}_N$ .

**COROLLARY 4.2.** *NCUT-normalized spectral ordering maximizes the C1P property of the normalized data  $\mathbf{D}^{-1}\mathbf{A}$  the same way as the unnormalized method does for the unmodified data  $\mathbf{A}$ .*

Using very similar conclusions than with NCUT we can analyze the SYM-normalized spectral sorting algorithm. For this purpose we need to normalize the data  $\mathbf{A}$ , this time as  $\mathbf{D}^{-1/2}\mathbf{A}$ :

$$(4.9) \qquad \|\mathbf{S}\mathbf{P}\mathbf{D}^{-1/2}\mathbf{A}\|_{\mathbb{F}}^2 = \sum_{i=1}^r \psi_i^2 \|\mathbf{S}\mathbf{P}\mathbf{f}_i\|_2^2.$$

The second term of the resulting sum is the value minimized by SYM-normalized spectral sorting. The first term of the sum however doesn't disappear, since  $\mathbf{f}_1 = (\sqrt{d_1}, \sqrt{d_2}, \dots, \sqrt{d_m})$ . Therefore the more there is variation in the values of  $\mathbf{D}$  and thus in the difficulty of finding a good position for each of the rows of  $\mathbf{A}$ , the bigger is the first term of the sum and the less has SYM-normalized spectral sorting effect on the minimization objective.

In both cases of normalization, NCUT or SYM, the spectral ordering ends up maximizing the C1P property for a modified version of the data matrix, where each element is normalized by taking the whole data row into account. This may cause several problems. First of all, for binary matrices the difference of two consecutive non-zero elements is always zero. This is not the case for normalized data, which makes the results difficult to forecast or understand. Second, a non-constant strongest eigenvector, as in the case of SYM-normalization, reduces the impact of spectral ordering on the optimization problem as a whole. As the third and most fundamental reason, the normalizations done to the data in cases of normalized spectral methods normalize the data by rows. This happens despite that in the C1P problem different columns are highly independent units that should not be directly compared with each other. On the other hand, this observation raises the question

whether normalizing data by columns could be of any use in the problem.

This problem with normalization can also be seen in the experiments in Section 5, where normalized spectral ordering seems to perform worse than unnormalized. Therefore, whereas normalization for spectral methods is highly recommended in applications of clustering, it is not desirable in the applications of C1P maximization.

As the last part of our analysis we examine the modified spectral ordering method used in e.g. [9]. In this scheme dot product similarities of rows are replaced by cosine similarities, i.e., the similarity matrix is given as  $\widehat{\mathbf{W}} = \mathbf{T}^{-1/2} \mathbf{A} \mathbf{A}^T \mathbf{T}^{-1/2}$ , where  $\mathbf{T} = \text{diag}(\sum_j a_{1j}, \sum_j a_{2j}, \dots, \sum_j a_{mj})$  contains the row sums of  $\mathbf{A}$ . Aside from this change, the algorithm is the same. Let us now denote the diagonal row sum matrix of  $\widehat{\mathbf{W}}$  with  $\widehat{\mathbf{D}}$ , the resulting Laplacian with  $\widehat{\mathcal{L}} = \widehat{\mathbf{D}} - \widehat{\mathbf{W}}$  and let the Laplacian have the eigenvalue decomposition  $\widehat{\mathcal{L}} = \widehat{\mathbf{U}} \widehat{\mathbf{\Lambda}} \widehat{\mathbf{U}}^T$  where  $\widehat{\mathbf{U}} = (\widehat{\mathbf{u}}_1 \widehat{\mathbf{u}}_2 \dots \widehat{\mathbf{u}}_k)$  and  $\widehat{\mathbf{\Lambda}} = \text{diag}(\widehat{\lambda}_1, \widehat{\lambda}_2, \dots, \widehat{\lambda}_k)$  with eigenvalues  $\widehat{\lambda}_i$  sorted ascending. Now our objective is to solve

$$\begin{aligned}
\mathbf{P}^* &= \arg \min_{\mathbf{P}} \|\mathbf{S} \mathbf{P} \mathbf{A}\|_{\text{F}}^2 \\
&= \arg \min_{\mathbf{P}} \text{tr} \left( \mathbf{S} \mathbf{P} \mathbf{T}^{1/2} \left( \widehat{\mathbf{D}} - \widehat{\mathcal{L}} \right) \mathbf{T}^{1/2} \mathbf{P}^T \mathbf{S}^T \right) \\
&= \arg \min_{\mathbf{P}} \left[ \text{tr} \left( \mathbf{S} \mathbf{P} \mathbf{T}^{1/2} \left( \widehat{\mathbf{D}} - \alpha \mathbf{I} \right) \mathbf{T}^{1/2} \mathbf{P}^T \mathbf{S}^T \right) \right. \\
&\quad \left. + \text{tr} \left( \mathbf{S} \mathbf{P} \mathbf{T}^{1/2} \left( \alpha \mathbf{I} - \widehat{\mathcal{L}} \right) \mathbf{T}^{1/2} \mathbf{P}^T \mathbf{S}^T \right) \right] \\
(4.10) \quad &= \arg \min_{\mathbf{P}} \sum_{i=1}^k \left( \alpha - \widehat{\lambda}_i \right)^2 \left\| \mathbf{S} \mathbf{P} \mathbf{T}^{1/2} \widehat{\mathbf{u}}_i \right\|_2^2,
\end{aligned}$$

where  $\alpha$  is chosen bigger than  $\max \widehat{\lambda}_i$ . The last simplification is done similarly as for Equation (3.5), since  $\mathbf{T}^{1/2} \left( \widehat{\mathbf{D}} - \alpha \mathbf{I} \right) \mathbf{T}^{1/2}$  is a diagonal matrix and so the term corresponding to it in Equation (4.10) can be dropped. The first term of the final norm sum is built on vector

$$\begin{aligned}
\mathbf{T}^{1/2} \widehat{\mathbf{u}}_1 &= \mathbf{T}^{1/2} \mathbf{1} \\
(4.11) \quad &= \left( \sqrt{\sum a_{1j}}, \sqrt{\sum a_{2j}}, \dots, \sqrt{\sum a_{mj}} \right)
\end{aligned}$$

and so affects the end results. Also in this spectral ordering modification using cosine similarities, the permutation that sorts the Fiedler vector  $\widehat{\mathbf{u}}_2$  is chosen as the approximated solution. Based on Equation (4.10) one could argue that choosing the permutation that sorts the vector  $\mathbf{T}^{1/2} \widehat{\mathbf{u}}_2$  could be more useful in maximizing the C1P property. This variation however performed

consistently worse in the experiments that were conducted than the simple sorting of  $\widehat{\mathbf{u}}_2$ .

## 5 Experiments

In this section we provide explanations on the performance differences based on the work in Section 3. We also compare the methods to a reference method introduced in [4] that approximates a minimum weight Hamiltonian path by constructing a minimum weight spanning tree. In the core this reference method simply gives an approximate solution to the related travelling salesman problem in the original binary space. We will also give explanations on the performance differences based on the work in Section 3. The purpose of this experimental evaluation is not to compare the methods on their performance, but to compare their differences in the light of the theoretical work done in the paper.

Based on the analysis of Section 4.1, we would expect the standard unnormalized spectral ordering to beat all its three modifications. Another point of comparison is between spectral ordering that uses only one dimension in its calculations and the reference method that approximately solves the full-dimensional problem.

We use three different 0-1 data sets in the experiments. For each of them the data's rows are first randomly permuted and this result is then fed to each of the data ordering methods. Then the values of the  $m_c$  and  $m_z$  metrics of non-consecutiveness defined in Section 2 are calculated and presented for the randomly permuted data and for the ordering results. All the results of these experiments can be found in Table 1. The methods do not have any random behaviour except for minor numeric instability and so their outputs do not depend on the permutation used on their inputs. Therefore the results in Table 1 are averaged over 100 samples for the unordered matrices, but for the actual ordering methods results of one single run are shown.

**5.1 Results** The first data set, SYNTH, is a synthetic  $120 \times 100$  pre-C1P matrix. This means that there exists a row permutation for the data that puts it into exact C1P form. The purpose of the data set in the experiments is to see whether the tested methods are capable of finding the correct solution. This would mean reaching the level of zero errors with both of the tested metrics. Note that standard PQ-tree construction is guaranteed to find the correct permutation of zero error in linear time.

In the experiments the unnormalized spectral ordering algorithm finds the optimal solution as it should based on the result of Atkins et al. in [1]. The same remains true when using cosine similarities. This is

Method	SYNTH		PALEO		POWER	
	$m_c$	$m_z$	$m_c$	$m_z$	$m_c$	$m_z(\times 10^3)$
Unordered	2 882	9 073	3 097	12 435	16 478	8 653
SS	0	0	1 998	3 794	14 874	204
SYM-SS	890	972	2 116	4 492	15 174	1 470
NCUT-SS	2	2	2 078	3 834	14 402	200
Cosine-SS	0	0	1 992	3 781	14 958	195
Reference	94	2 098	1 622	6 068	9 074	4 786

Table 1: The distance from full consecutiveness as calculated by the  $m_c$  and  $m_z$  metrics. The unordered results are taken as an average from 100 random permutations of data rows. In the first column the unnormalized spectral sorting method is abbreviated SS. The SYM-normalized, NCUT-normalized and spectral sorting using cosine similarities are abbreviated SYM-SS, NCUT-SS and Cosine-SS respectively. Reference designates the results for the approximate minimum weight Hamiltonian path method.

quite unexpected, because the most significant term of Formula (4.10) is ignored by the method. To step up this surprise, the row sums in Equation (4.11) are quite uniformly distributed between 1 and 25 for the SYNTH data, which one would expect to make this ignorance even more costly.

In contrast, the normalized spectral ordering methods fail in the task of finding the correct permutation although the NCUT-normalized method gets very close. Finally also the reference method performs well but is still left substantially afar from the optimal solution, especially for the  $m_z$  metric.

Our second data set, PALEO, describes the fossils found from a number of European excavation sites, see [8]. The data contains 139 fossil genera found in 124 sites.

With this data set the reference method outperforms the spectral methods by a clear margin when using the  $m_c$  metric. On the other hand the results for the reference method are far worse than for the other methods when looking at the  $m_z$  metric. This can also be seen in Figure 1, where the spectral ordering method aligns the data much more diagonally and consistently than the reference method.

The unnormalized spectral ordering methods gave better results than the normalized versions, although the differences in the results are not easily distinguished unlike with the reference method. Compared to the results from the SYNTH data, this time the results of SYM-normalized spectral ordering are much better. This happens, because the row sums  $d_i$  that make up the vector  $\mathbf{f}_1$  in Formula (4.9) do not vary as much as with SYNTH data, thus reducing the significance of the strongest term of Formula (4.9) that is ignored by the method.

The third data set, POWER, gives a graph repre-

sentation of the topology of Western States Power Grid in the United States, see [23]. The size of the data is  $4941 \times 4941$ . There are no direct applications for the C1P property with this data set, but the test results can nevertheless be used to infer interesting facts about the power grid.

Also with this data set the reference method performs better in  $m_c$  metric than any of the spectral ordering methods, but with an even bigger margin. This poor performance of the spectral method can be explained by the eigenvalues of the data matrix' Laplacian. The quotient of the second smallest and the smallest non-zero eigenvalue of the unnormalized Laplacian is only 1.43, which means that the largest norm sum term that gets optimized by spectral ordering has only limited effect on the whole situation. As a comparison to this, with the PALEO data set, where the spectral ordering methods perform considerably better relative to the reference method, the corresponding quotient of eigenvalues is 1.85. This observation is consistent with the common phenomenon that bigger eigengaps make spectral methods perform better. In light of this consideration, it is interesting to note how decisively the spectral methods outperform the reference method even though the reference method considers all dimensions equally and should perform thus relatively well when eigengaps are small.

The poor C1P maximization results and the small eigengap explaining can also be seen in the light of the power grid topology. Simplistically, the spectral ordering methods attempt to order the data rows (power grid hubs) in a single line. The evident failure of the methods in this attempt shows that the power grid has plenty of interconnections making it robust against hub failures as it should. A bit closer look to the resulting spectral orders shows that there are



Figure 1: Data set PALEO before ordering and after ordering with unnormalized spectral sorting and with the reference method. The result of spectral ordering is more diagonalized and its  $m_z$  score is significantly smaller, see also Table 1.

no significant subsets of hubs in the grid that could be ordered in a good C1P order. Such subsets would indicate fragile parts of the grid. Another way to reach similar conclusions would be to note that the existence of several strong eigenvalues implies that the data has at least two significant dimensions and cannot be serialized easily.

**5.2 Discussion on results** It is easy to notice how the normalized spectral methods, SYM-normalization especially, seem to fall behind in performance. This notion should be compared with the observations of Section 3 about the differences of normalized and unnormalized spectral methods. Both normalized methods suffer from the fact that they try to maximize the C1P property of a modified data matrix. Furthermore the SYM-normalized spectral ordering method suffers from ignoring the most significant term in Formula (4.9). This extra handicap explains the consistent underperformance of this method against the NCUT-normalized method, less so for PALEO data than for others as already examined above.

With the  $m_z$  metric, results from all the data sets were decidedly worse for the reference method than for the spectral methods. Based on this it seems that even though the spectral methods concentrate solely in optimizing the  $m_c$  metric in one linear component, they manage to succeed very well in approximately minimizing the  $m_z$  metric. This curious upside-down situation of spectral ordering performance for the two metrics asks for more analysis. It would be especially interesting to see if spectral space travelling salesman solutions in general allow better results in terms of  $m_z$  and original binary space solutions in terms of  $m_c$ . This is potential future work on the topic.

The running times of all the methods were under

one second for SYNTH and PALEO data sets. With the POWER data set however the spectral ordering methods took 2.5 seconds on average whereas running the reference method took as much as 28 minutes. Therefore even though the reference method won the spectral ordering methods generally in  $m_c$  results, it has limited applicability to large problems. The reason for this is largely that the reference method uses all dimensions of the data. Standard solution for this would be the use of dimension reduction prior to running the algorithm. This leads us once again to a spectral space solution. Also this observation shows the need for a broad comparison of different algorithms in both spectral space and the original binary space.

## 6 Conclusion

We studied the problem of permuting the rows of a data matrix to make it as close as possible to having the consecutive ones property. We analyzed several different variations of the spectral ordering algorithm and discovered the theoretical reasons for the good performance of the algorithms on approximating solutions to the problem. This work builds a solid foundation on the use of the algorithm and also gives tools for the utilizers of the method to evaluate the precision of their experiments and to analyze the reasons for phenomena found in the data. In the future also other common methods that are heuristically used to solve the optimization problem should receive theoretical attention to justify their use and to give credibility to the obtained results.

We showed that the objective of the optimization problem is equivalent to minimizing the Frobenius norm of a certain derived matrix. We proved that the spectral sorting algorithm gives the optimal approximation to minimizing this Frobenius norm based on only the first principal component. Furthermore it was shown that

it is **NP**-hard use more than one principal component for this optimization approach. The earlier result of Atkins from [1] was also seen to be an extreme case of our analysis.

We also analyzed popular normalization schemes for spectral methods and based on the observations recommend against their use in the C1P maximization problem.

Theoretical results proven in this paper or previously were verified with experiments and the effects of theory in practice was studied. The experiments show that even though the spectral methods are not the best choice for simply minimizing the metric  $m_c$  analyzed in the paper, they are diverse and manage to do well also in the  $m_z$  metric. Considering the good speed and scalability of the spectral method, it is a workable and balanced method suitable for a varied set of tasks.

The analysis of the alternative  $m_z$  metric and its relevance on the C1P optimization problem is one important subject for future work. This analysis should also be extended to a more general comparison of spectral space and original binary space solutions with varying algorithms in both cases.

Another interesting problem would be to seek out a pathological example where spectral ordering fails in terms of the  $m_z$  metric. This could also shed light on the differences of the two metrics' theoretical properties. Furthermore, since building a PQR-tree from data is a highly efficient operation, it would be interesting also to approximate the consecutiveness of each PQR-tree and establish an algorithm for maximizing the C1P property based on PQR-trees.

## References

- [1] Jonathan E. Atkins, Erik G. Boman, and Bruce Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing*, 28:297–310, 1998.
- [2] Stephen T. Barnard, Alex Pothen, and Horst D. Simon. A spectral algorithm for envelope reduction of sparse matrices. In *Supercomputing '93: Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, pages 493–502, 1993.
- [3] Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and planarity using pq-tree algorithms. *Journal of Computational Systems Science*, 13(3):335–379, 1976.
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.
- [5] Chris Ding and Xiaofeng He. Linearized cluster assignment via spectral ordering. In *Proceedings of the 21st international conference on Machine learning*, page 30, 2004.
- [6] Jarmo Ernvall, Jyrki Katajainen, and Martti Penttonen. NP-completeness of the hamming salesman problem. *BIT Numerical Mathematics*, 25:289–292, March 1985.
- [7] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.
- [8] Mikael Fortelius. Neogene of the old world database of fossil mammals (NOW). <http://www.helsinki.fi/science/nov/>, 2006.
- [9] Mikael Fortelius, Aristides Gionis, Jukka Jernvall, and Heikki Mannila. Spectral ordering and biochronology of european fossil mammals. *Paleobiology*, 32:206–214, March 2006.
- [10] Delbert Ray Fulkerson and O.A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(3):835–855, 1965.
- [11] Gemma Garriga, Esa Junntila, and Heikki Mannila. Banded structure in binary matrices. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 292–300, 2008.
- [12] Lars Hagen and Andrew B. Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 11(9):1074–1085, 1992.
- [13] Wen-Lian Hsu. A simple test for the consecutive ones property. *Journal of Algorithms*, 43(1):1–16, 2002.
- [14] László Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is eighty*, 2:353–397, 1993.
- [15] J. Meidanis and Erasmo G. Munuera. A theory for the consecutive ones property. In *Proceedings of WSP'97 - Third South American Workshop on String Processing*, pages 194–202, 1996.
- [16] J. Meidanis and O. Porto G.P. Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88:325–354, 1998.
- [17] Bojan Mohar. The laplacian spectrum of graphs. In *Graph Theory, Combinatorics, and Applications*, volume 2, pages 871–898. Wiley, 1991.
- [18] Nikolaus Ruf and Anita Schbel. Set covering with almost consecutive ones property. *Discrete Optimization*, 1(2):215–228, 2004.
- [19] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [20] Antti Ukkonen, Mikael Fortelius, and Heikki Mannila. Finding partial orders from unordered 0-1 data. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 285–293, New York, NY, USA, 2005. ACM.
- [21] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
- [22] Michael S. Waterman and Jerrold R. Griggs. Interval graphs and maps of dna. *Bulletin of Mathematical Biology*, 48(2):189–195, 1986.

- [23] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, (393):440–442, 1998.
- [24] Peisen Zhang, Eric A .Schon, Stuart G. Fischer, Eftihia Cayanis, Janie Weiss, Susan Kistler, and Philip E. Bourne. An algorithm based on graph theory for the assembly of contigs in physical mapping of dna. *Bioinformatics*, 10(3):309–317, 1994.