Noname manuscript No.
(will be inserted by the editor)

Continuation Methods for Approximate Large Scale Object Sequencing

Xenophon Evangelopoulos ·
Austin J. Brockmeier · Tingting Mu ·
John Y. Goulermas

Received: date / Accepted: date

Abstract We propose a set of highly scalable algorithms for the combinatorial data analysis problem of seriating similarity matrices. Seriation consists of finding a permutation of data instances, such that similar instances are nearby in the ordering. Applications of the seriation problem can be found in various disciplines such as in bioinformatics for genome sequencing, data visualization and exploratory data analysis. Our algorithms attempt to minimize certain p-SUM objectives, which also arise in the problem of envelope reduction of sparse matrices. In particular, we present a set of graduated non-convexity algorithms for vector-based relaxations of the general p-SUM problem for $p \in \{2, 1, \frac{1}{2}\}$ that can scale to very large problem sizes. Different choices of p emphasize global versus local similarity pattern structure. We conduct a number of experiments to compare our algorithms to various state-of-the-art combinatorial optimization methods on real and synthetic datasets. The experimental results demonstrate that compared to other approaches, the proposed algorithms are very competitive and scale well with large problem sizes.

Keywords Combinatorial Data Analysis · Combinatorial Optimization · Graduated non-Convexity · Sequencing · Seriation.

X. Evangelopoulos and J.Y. Goulermas
Department of Computer Science, University of Liverpool, U.K.
E-mail: {x.evangelopoulos, j.y.goulermas}@liverpool.ac.uk

A.J. Brockmeier and T. Mu
School of Computer Science, University of Manchester, U.K.
E-mail: {austin.brockmeier, tingting.mu}@manchester.ac.uk

1 Introduction

Seriation is an exploratory combinatorial data analysis method that aims at reordering data objects to capture and identify patterns and trends of gradually varying similarities in the data. The general objective of the resulting reordering is to position more similar objects proximately and dissimilar ones further apart. The original motivation for seriation arose in the field of archeology, when Sir Flinders Petrie used sequencing to infer the chronological order of a set of graves based on the artifacts recovered from them (Hodson, 1968). The problem of seriation was mathematically formalized by Kendall (1971). Since then, it has been studied and successfully put to practice in several other areas, such as sociology and psychology (Liiv, 2010), gene sequencing (Fulkerson and Gross, 1965), and bioinformatics (Tsafrir et al, 2005; Tien et al, 2008; Recanati et al, 2017). Seriation can also be used in exploratory data visualization (Havens and Bezdek, 2012) as a means for rearranging similarity or dissimilarity matrices, so that global patterns (e.g., the number or tendency of clusters) can be identified. For this purpose, it has been applied to reveal patterns in microarray data (Tien et al, 2008), and to arrange words or documents in text mining based on their co-occurrence statistics (Mavroeidis and Bingham, 2010); the latter work also includes the reordering of word-by-document similarity matrices for the purpose of tracking the flow of conversations. A broad overview of different applications and miscellaneous theoretical details of seriation is presented by Liiv (2010) and Hahsler et al (2008). More recent works include the systematic experimental analysis of seriation methods and measures by Hahsler (2017), mechanisms for comparing and fusing generated orderings by Goulermas et al (2016), and the introduction of various modeling formulations and solution procedures for robust seriation by Recanati et al (2018).

Seriation methods employ heuristics or combinatorial optimization procedures in order to identify orderings that maintain object proximities according to their pairwise (dis)similarities. They typically act on a symmetric similarity (dissimilarity) matrix to simultaneously interchange its rows and columns, such that its entries decrease (increase) monotonically while departing from the main diagonal. Formally, given an $n \times n$ symmetric similarity matrix $\mathbf{A}$, the goal of seriation is to find an ideal row and column reordering, such that $A_{ik} \leq \min(A_{ij}, A_{jk})$, for all i, j, k with $1 \leq i \leq j \leq k \leq n$; in other words bring it to a Robinsonian[1] form.

One consistent objective for seriation is the p-SUM (Juvan and Mohar, 1992), defined as $\frac{1}{p}\sum_{i,j=1}^{n} A_{ij}\,|i-j|^p$, since for all $p > 0$, an optimal ordering that renders any pre-Robinsonian[2] matrix to a Robinsonian one can be found (Laurent and Seminaroti, 2015). The p-SUM problem, which was initially introduced in the context of the matrix envelope reduction problem (George and

[1] Named after William S. Robinson who mathematically formalized the seriation problem (Robinson, 1951).

[2] Any symmetric (dis)similarity matrix that can be symmetrically permuted to become Robinsonian.

Pothen, 1994), describes a class of objective functions that can be modeled as instances of the quadratic assignment problem (QAP) (Burkard et al, 1999), where a Toeplitz Robinsonian dissimilarity matrix is involved to represent positional differences of the objects. Different values of p confer different penalties on similar objects that are far apart in the linear ordering. Various instances of this problem have been studied, with the most widespread being the $p = 2$ case, which is referred to as the 2-SUM problem. In the context of seriation, the 2-SUM objective is known as the inertia criterion when it is applied to dissimilarity values (Hahsler et al, 2008). The 2-SUM objective penalizes the squared difference of the coordinates between similar instances, and can be expressed as a quadratic function of a permutation vector involving a graph Laplacian matrix (the details can be found later in Section 3.3).

Another specific case of the p-SUM is the 1-SUM problem, also known as the optimal linear arrangement problem (George and Pothen, 1994), which is more difficult to analyze in terms of a spectral approximation and bounds, as it is no longer a quadratic function of the permutation vector. In comparison with the 2-SUM objective function which relies on squared positional differences of the objects, the 1-SUM uses absolute differences. Finally, interesting p-SUM instances for seriation are the cases when $p < 1$, corresponding to quasi ℓ_p-norms, as they are less sensitive to large positional differences and relatively more sensitive to local ordering, and can therefore prioritize local neighborhoods of similar objects.

As a QAP instance, the p-SUM is an NP-hard combinatorial problem with $\mathcal{O}(n!)$ possible discrete solutions corresponding to permutations (Çela, 2013). Therefore, solving optimally such seriation formulations can be impractical when the problem size is large. In the ideal and infrequent case where the data yield a pre-Robinsonian similarity matrix, an optimal solution can be identified in polynomial time (Barnard et al, 1993; Atkins et al, 1998) by sorting the patterns according to the corresponding entries of the Fiedler vector (Fiedler, 1973), which is the eigenvector associated with the smallest nonzero eigenvalue. However, when the similarity matrix is not pre-Robinsonian, this spectral solution is only guaranteed to approximately minimize the 2-SUM problem. Therefore, alternative approaches for the p-SUM problem are desirable.

There exist various directions for solving QAP problems (Anstreicher, 2003; Burkard et al, 1999; Burkard and Çela, 1999; Loiola et al, 2007). Examples of exact QAP algorithms include branch-and-bound (Brusco and Stahl, 2001), cutting plane methods (Bazaraa and Sherali, 1982) and dynamic programming approaches (Christofides and Benavent, 1989). As exact methods can only be used for QAP instances of small sizes, suboptimal algorithms and heuristics that maintain good running performance have been very popular. Some of them include improvement methods, such as local search, tabu search (Glover and Laguna, 1997), simulation approaches such as simulated annealing, and population-based heuristics such as evolutionary optimization (Mühlenbein, 1989). Besides these, there are relaxation-based algorithms in the context of graph matching (Vogelstein et al, 2015; Lyzinski et al, 2016). Particularly for the 2-SUM case, recent works (Fogel et al, 2013; Lim and Wright, 2014; Fogel et al, 2015) have shown how the relaxations of the 2-SUM problem can be solved using interior-point methods relying on either matrix- or vector-based formulations. However, these relaxations may yield solutions far from the optimum permutation and there is no guarantee that the nearest permutation will minimize the original objective.

Relaxation methods have mostly been applied to the 2-SUM problem but not the general p-SUM. Our contribution is to propose a set of first-order optimization methods for minimizing certain p-SUM objectives. The methodology combines first-order optimization with graduated non-convexity, which successively transforms the relaxation to a concave problem, so that the final solution is guaranteed to be a permutation. We previously showed (Evangelopoulos et al, 2017) that this approach outperforms other convex relaxation methods for the 2-SUM problem and scales very well with large datasets. Additionally, while previous methods rely on extra ordering information to achieve good performance, our method does not have such requirement. Here, we extend this work by proposing algorithms for approximately solving the 1-SUM and $\frac{1}{2}$-SUM objectives. The proposed methodologies are able to scale up to problem sizes unattainable with existing approaches, and additionally, apart from the noiseless cases they outperform the spectral approximation algorithms which are the most computationally efficient approaches. To the best of our knowledge, this is the first time that highly scalable algorithms for the p-SUM problem with $p < 2$ have been proposed.

The rest of the paper is organized as follows. In Section 2, we present recent developments in the field and the current state-of-the-art algorithms. In Section 3, we give a detailed description for each of the proposed algorithms, with the different subsections presenting various formulations and optimization-related aspects. Section 4 contains detailed experimental evaluations and comparisons with regard to the performance of the algorithms, while relevant analyses and conclusions are presented in Section 5.

2 Relation to Existing Methods

The most extensively studied instance of the p-SUM problem is the 2-SUM one because it is amenable to a much more convenient algebraic formulation. The most recent approaches approximate the 2-SUM problem via convex relaxations. Specifically, Fogel et al (2013, 2015) formulate their relaxation over the set of doubly stochastic matrices which is known to be the convex hull of the permutation matrices, while Lim and Wright (2014) use sorting networks to generate a set of linear constraints in order to perform the optimization in terms of the permutahedron (Goemans, 2015), which is the convex hull of all permutation vectors. In both cases, interior point methods are used to optimize a regularized version of the 2-SUM problem that can be written as a quadratic program with additional linear constraints. The permutahedron-based method performs better and is considerably faster as it uses an order of $\mathcal{O}(n \log^2 n)$

variables and constraints. Furthermore, both approaches can be used to solve a semi-supervised instance of seriation as they both accommodate the use of additional ordering constraints.

Nevertheless, the aforementioned convex relaxation approaches do not outperform spectral ordering unless additional ordering constraints are used. Moreover, they suffer from scalability issues and when the input size increases significantly, even commercial solvers cannot alleviate the need for demanding computational resources. Furthermore, recent work (Vogelstein et al, 2015; Lyzinski et al, 2016) on solving general QAP problems suggests that convex relaxations do not always outperform indefinite formulations. Towards this direction, Lim and Wright (2016a) present a new framework for approximating general QAP problems formulated in terms of sorting networks, and use a continuation procedure (Blake, 1983; Rangarajan and Chellappa, 1990; Liu and Qiao, 2014) that starts by solving a convex relaxation of the problem and then gradually converts it to a concave one, to finally yield a local optimum to the original discrete problem. A similar approach was followed by Zaslavskiy et al (2009), where instead of employing an objective function with a convex and nonconvex component as used in typical continuation methods, the authors follow the solution path of a linear combination of two different relaxations of the initial problem, one convex and one concave, in order to approximately solve it.

Other instances of the p-SUM problem, especially for $p < 1$, have not been studied extensively in the seriation literature. Juvan and Mohar (1992, 1993) are the first to present a theoretical analysis on the minimization of the p-SUM problem for $p = 1, 2$, and ∞ using a spectral method. George and Pothen (1994) investigate the specific cases of 1-SUM and 2-SUM and their close connection to the matrix envelope reduction problem (George and Liu, 1981), as the former problem is expressed via the sum of spreads of the nonzero entries in each row, while the latter uses the sum of squared spreads. Most of the problems analyzed for the different p-SUM employed spectral methods. Such methods were also used by Helmberg et al (1995) to obtain lower bounds on the bandwidth problem. In this work we present alternative methodologies that enable us to solve an approximation of different p-SUM problems in a more efficient way than other convex relaxations and spectral methods.

3 Proposed Methodology

3.1 Preliminaries and Basic Notations

Let $\boldsymbol{\pi}$ denote a permutation vector consisting of the rearrangement of the integers $1, \ldots, n$. The set of $n!$ distinct permutations (which for convenience are treated here as vectors) is denoted by $\mathcal{P}^n$. Each permutation describes the rearrangement of the entries of an n-dimensional vector, with one convention being that the element at position π_i is moved to position i. This transformation can be explicitly represented by an $n \times n$ matrix $\mathbf{\Pi}$ from the set of permutation matrices $\mathcal{M}^n$ with elements defined by

$$\mathit{\Pi}_{ij} = \begin{cases} 1, & \text{if } \pi_i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

This also allows $\mathbf{\Pi}$ to be converted to its corresponding permutation via $\mathbf{\Pi e} = \boldsymbol{\pi}$, where $\mathbf{e} = (1, 2, ..., n)^\top$ is the identity permutation.

Many combinatorial problems involving the optimal arrangement of objects can be modeled by objective functions parametrized by permutation vectors or matrices. In particular, the aforementioned QAP describes models that are quadratic with respect to a permutation matrix, and can be expressed as

$$\mathrm{QAP}(\mathbf{A}, \mathbf{B}) \triangleq \mathrm{tr}\left[\mathbf{A\Pi B}^\top \mathbf{\Pi}^\top\right] = \sum_{i,j=1}^{n} A_{ij} B_{\pi_i \pi_j}, \tag{2}$$

where the problem depends on the two parameter matrices $\mathbf{A}$ and $\mathbf{B}$.

For seriation we are interested in specific QAP instances, where $\mathbf{A}$ is a non-negative[3] symmetric data-dependent matrix that encapsulates the pairwise similarities between n objects. $\mathbf{B}$ is a Toeplitz Robinsonian dissimilarity matrix with elements $B_{ij} = \frac{1}{p} |i - j|^p$ for some $p > 0$. It acts as the seriation template with elements increasing across diagonals while moving away from the main one. In this case, the QAP corresponds to the p-SUM problem (George and Pothen, 1994)

$$\mathrm{QAP}(\mathbf{A}, \mathbf{B}) = \frac{1}{p} \sum_{i,j=1}^{n} A_{ij} |\pi_i - \pi_j|^p. \tag{3}$$

When $\mathbf{A}$ is Robinsonian, the identity permutation optimizes the QAP (Laurent and Seminaroti, 2015), and if $\mathbf{A}$ is pre-Robinsonian, then a solution can be found in polynomial time (Atkins et al, 1998). Different cases for p yield different types of problems. For example, for $p = 1, 2$ and in the limit of ∞, we obtain the 1-SUM or optimal linear arrangement, the 2-SUM, and the bandwidth minimization problem, respectively (this relies on the more conventional problem definition of $(\sum A_{ij} |\pi_i - \pi_j|^p)^{\frac{1}{p}}$). Approximate solutions for this problem can be searched for with a variety of QAP approximation methods, including simulated annealing, tabu search, and evolutionary methods (Loiola et al, 2007).

3.2 Problem Relaxations

Recent work on the 2-SUM (Fogel et al, 2013; Lim and Wright, 2014; Fogel et al, 2015) has considered convex relaxations on the set of permutation matrices and also on permutation vectors. The relaxed feasible sets are the convex hull of permutation matrices which is the Birkhoff polytope, i.e., the

[3] Even if there are negative entries, adding a constant to the matrix does not change the minimizing permutation.

set of doubly stochastic matrices $\mathcal{B}^n \triangleq \{\mathbf{X} : \mathbf{X}\mathbf{1} = \mathbf{X}^\top\mathbf{1} = \mathbf{1}, X_{ij} \geq 0\}$, and the convex hull of permutation vectors which is the permutahedron (Goemans, 2015) denoted as $\mathcal{PH}^n$. These are directly related by enumerating all contributing permutations; that is, for each $\mathbf{X} = \sum_{i=1}^{n!} a_i \mathbf{\Pi}_i \in \mathcal{B}^n$, we have $\mathbf{x} = \mathbf{X}\mathbf{e} = \sum_{i=1}^{n!} a_i \boldsymbol{\pi}_i \in \mathcal{PH}^n$, where the ith vertex correspondence between the polytopes is through $\boldsymbol{\pi}_i = \mathbf{\Pi}_i\mathbf{e}$, and the coefficients of the convex combination satisfy $a_i \geq 0$ and $\sum_{i=1}^{n!} a_i = 1$.

For the p-SUM problem, possible relaxations can be expressed as

$$\min_{\mathbf{x}\in\mathcal{PH}^n} \frac{1}{p}\sum_{i,j} A_{ij} \left|x_i - x_j\right|^p , \tag{4}$$

or, in matrix form, as

$$\min_{\mathbf{X}\in\mathcal{B}^n} \operatorname{tr}\left[\mathbf{A}\mathbf{X}\mathbf{B}^\top\mathbf{X}^\top\right] , \tag{5}$$

where $B_{ij} = \frac{1}{p}\left|i-j\right|^p$. The first objective function, for $p \geq 1$ and $A_{ij} \geq 0$ is convex, since it is non-negative combination of convex functions $|\cdot|^p$ applied to the linear functions $x_i - x_j$, with $i, j \in \{1, \ldots, n\}$. The second objective depends on $\mathbf{A}$ and $\mathbf{B}$, but these can be adjusted in their diagonals before relaxation to become convex[4]. Nonetheless, this convexity is not useful. For example, the constant vector $\frac{n+1}{2}\mathbf{1}$ which lies at the barycenter of the permutahedron, minimizes the relaxed problem in Eq.(4) since all $x_i - x_j = 0$.

In order to find non-trivial solutions further from the barycenter and closer to the vertices, as the norm of each permutation vector is constant and maximal over the relaxed set, we attempt to maximize the norm of the relaxed solution while simultaneously minimizing the original objective. Using a trade-off parameter $\mu > 0$, this may lead to the following regularized objective

$$\min_{\mathbf{x}\in\mathcal{PH}^n} \frac{1}{p}\sum_{i,j} A_{ij} \left|x_i - x_j\right|^p - \mu \left\|\mathbf{x}\right\|_2^2 . \tag{6}$$

3.3 Regularized 2-SUM Relaxation

Due to its quadratic form, the 2-SUM case is amenable to more convenient algebraic manipulations and it has therefore attracted further attention by recent works (Barnard et al, 1993; Atkins et al, 1998; Fogel et al, 2013; Lim

[4] In general any QAP can be easily modified before relaxation, so that its relaxed version can assume a convex or concave form. For example, for the formulation $\operatorname{tr}\left[\mathbf{A\Pi B\Pi}^\top\right]$ for seriation, the symmetric similarity matrix $\mathbf{A}$ could have negative eigenvalues, and the seriation template $\mathbf{B}$ always has eigenvalues of both signs (being a hollow matrix). In this case, the formulation $\operatorname{tr}\left[(\mathbf{A} - \lambda_1(\mathbf{A})\mathbf{I})\ \mathbf{\Pi}\ (\mathbf{B} - (\mu\lambda_n(\mathbf{B}) + (1-\mu)\lambda_1(\mathbf{B}))\mathbf{I})\,\mathbf{\Pi}^\top\right]$ will transform the objective from a convex to a concave, whilst adjusting μ from within $(-\infty, 0]$ to within $[1, +\infty)$.

and Wright, 2014; Fogel et al, 2015). In particular, the associated QAP can be reformulated into an equivalent one parametrized by a rank-1 matrix as

$$\begin{aligned}\mathrm{QAP}(\mathbf{A},\mathbf{B}) &= \frac{1}{2}\sum_{i,j=1}^{n} A_{ij}(\pi_i^2+\pi_j^2-2\pi_i\pi_j)\\ &= \sum_{i=1}^{n}\pi_i^2\sum_{j=1}^{n}A_{ij}-\sum_{i,j=1}^{n}\pi_i\pi_j A_{ij}\\ &= \boldsymbol{\pi}^\top\left(\mathrm{dg}\left(\mathbf{A1}\right)-\mathbf{A}\right)\boldsymbol{\pi}=\boldsymbol{\pi}^\top\mathbf{L_A}\boldsymbol{\pi}\\ &= \mathrm{tr}[\mathbf{L_A}\boldsymbol{\Pi}\mathbf{e}\mathbf{e}^\top\boldsymbol{\Pi}^\top]=\mathrm{QAP}(\mathbf{L_A},\mathbf{e}\mathbf{e}^\top),\end{aligned}$$

where $\mathrm{dg}\left(\mathbf{x}\right)$ returns a diagonal matrix with elements from a vector $\mathbf{x}$. The matrix $\mathbf{L_A}\triangleq\mathrm{dg}\left(\mathbf{A1}\right)-\mathbf{A}$ is defined to be the graph Laplacian, and is guaranteed to be positive semidefinite for symmetric non-negative $\mathbf{A}$, since $f(\mathbf{x})\triangleq\mathbf{x}^\top\mathbf{L_A}\mathbf{x}=\frac{1}{2}\sum_{i,j}A_{ij}(x_i-x_j)^2\geq 0,\forall\mathbf{x}\in\mathbb{R}^n$.

The resulting QAP form above is very practical as it can be used in a relaxed version of the 2-SUM, expressed in either of the following forms

$$\min_{\mathbf{x}\in\mathcal{PH}^n}\mathbf{x}^\top\mathbf{L_A}\mathbf{x}\equiv\min_{\mathbf{X}\in\mathcal{B}^n}\mathbf{e}^\top\mathbf{X}^\top\mathbf{L_A}\mathbf{X}\mathbf{e}. \tag{7}$$

It is clear that the objective function is convex since, in terms of the first form, the Hessian $\mathbf{L_A}$ is positive semidefinite. In terms of the matrix form, the objective can be rewritten as $\mathrm{vec}\left(\mathbf{X}\right)^\top(\mathbf{e}\mathbf{e}^\top\otimes\mathbf{L_A})\,\mathrm{vec}\left(\mathbf{X}\right)$, where $\otimes$ denotes the Kronecker product, and the Hessian $\mathbf{e}\mathbf{e}^\top\otimes\mathbf{L_A}$ is positive semidefinite. However, the optimal solution to this relaxed formulation is the barycenter $\frac{1}{n}\mathbf{1}\mathbf{1}^\top$ of $\mathcal{B}^n$, since $[\mathbf{L_A}\mathbf{1}]_i=\sum_k\mathbf{A}_{ik}-\sum_j\mathbf{A}_{ij}=0$, which gives $\mathbf{1}^\top\mathbf{L_A}\mathbf{1}=\sum_i[\mathbf{L_A}\mathbf{1}]_i=0$ that corresponds to the minimum of the objective function.

The objective in Eq.(7) can be modified in line with the regularization described in Section 3.2 to produce a non-trivial solution. For example, the objective $\mathbf{x}^\top\mathbf{L_A}\mathbf{x}-\mu\left\|\mathbf{x}\right\|_2^2$ can be used, but this precludes convexity for any $\mu>0$. An alternative modification for the 2-SUM minimization problem with a concave regularizer is suggested by Fogel et al (2013) and Lim and Wright (2014) as

$$\min_{\mathbf{x}\in\mathcal{PH}^n}\left\{f_\mu(\mathbf{x})\triangleq\mathbf{x}^\top(\mathbf{L_A}-\mu\mathbf{H})\mathbf{x}=\mathbf{x}^\top\mathbf{L_A}\mathbf{x}-\mu\left\|\mathbf{H}\mathbf{x}\right\|_2^2\right\}. \tag{8}$$

The use of the above regularizer leaves the sought optimization intact, since by using the constant matrix $\mathbf{J}=\mathbf{1}\mathbf{1}^\top$ and the centering one $\mathbf{H}=\mathbf{I}-\frac{1}{n}\mathbf{J}$, we have

$$\begin{aligned}\left\|\mathbf{x}\right\|_2^2 &= \left\|(\mathbf{H}+\tfrac{1}{n}\mathbf{J})\mathbf{x}\right\|_2^2=\left\|\mathbf{H}\mathbf{x}\right\|_2^2+\tfrac{2}{n}\mathbf{x}^\top\mathbf{H}\mathbf{J}\mathbf{x}+\left\|\tfrac{1}{n}\mathbf{J}\mathbf{x}\right\|_2^2\\ &= \left\|\mathbf{H}\mathbf{x}\right\|_2^2+\tfrac{n(n+1)^2}{4},\end{aligned}$$

where we make use of the facts that $\mathbf{H}\mathbf{J}=\mathbf{0}$, and that for any $\mathbf{x}\in\mathcal{PH}^n$, $\mathbf{J}\mathbf{x}=\frac{n(n+1)}{2}\mathbf{1}$. Note that this equivalence between the two regularizers holds independently of the problem relaxation from $\mathcal{P}^n$ to $\mathcal{PH}^n$.

The same can also be observed for the matrix formulation, after adding a constant c to one of the QAP matrix parameters. Specifically, the optimization of $\mathrm{QAP}(\mathbf{A}+c\mathbf{J},\mathbf{B})$ is unaffected (again either before or after the relaxation to $\mathcal{B}^n$), as it changes by the constant quantity $c\mathbf{1}^\top\mathbf{B}\mathbf{1}$. In such case, replacing $\mathbf{A}$ by $\tilde{\mathbf{A}}=\mathbf{A}-\frac{\mu}{n}\mathbf{J}$ yields $\mathbf{L}_{\tilde{\mathbf{A}}}=\mathbf{L}_{\mathbf{A}}-\mu\mathbf{I}+\frac{\mu}{n}\mathbf{J}=\mathbf{L}_{\mathbf{A}}-\mu\mathbf{H}$. The new matrix may no longer be positive semidefinite (it is not a proper Laplacian matrix as $\mathbf{A}-\frac{\mu}{n}\mathbf{J}$ may have negative entries) and the resulting minimization is not always convex.

Although the objective in Eq.(8) is generally non-convex because it is the difference of convex functions, convexity can be preserved for values of μ that keep $\mathbf{L}_{\mathbf{A}}-\mu\mathbf{H}$ positive semidefinite. Note that the constant vector is an eigenvector of both $\mathbf{L}_{\mathbf{A}}$ and $\mathbf{H}$ with an associated eigenvalue $\lambda_1=0$. Consequently, choosing $\mu\leq\lambda_2(\mathbf{L}_{\mathbf{A}})$ ensures convexity (henceforth, for the eigenvalues λ_i of a matrix $\mathbf{X}$ we assume the ordering $\lambda_1(\mathbf{X})\leq\ldots\leq\lambda_n(\mathbf{X})$). Moreover, choosing $\mu\geq\lambda_n(\mathbf{L}_{\mathbf{A}})$ ensures that this matrix is negative semidefinite, which renders the objective concave. Therefore, adjusting μ from $\lambda_2(\mathbf{L}_{\mathbf{A}})$ to $\lambda_n(\mathbf{L}_{\mathbf{A}})$ can gradually transform the relaxed 2-SUM problem from a convex, to an indefinite and finally to a concave problem. In general, except for the concave form, the relaxed solutions may lie in the interior of the polytope and far from the set of sought permutations. However, in the concave form, the solution will necessarily lie at the boundaries. We exploit this fact and use a continuation scheme to successively find relaxed solutions moving from the convex to the concave case, which is a common approach for similar problems (Zaslavskiy et al, 2009; Xia, 2010; Liu and Qiao, 2014).

3.4 First-order Optimization with Graduated Non-convexity

Given an initial feasible solution $\mathbf{x}^{(0)}$ and a current value for μ, we now show how to solve the relaxed and regularized 2-SUM problem using first-order optimization. In particular, we employ conditional gradient, also known as the Frank-Wolfe (FW) algorithm (Frank and Wolfe, 1956), to ensure the optimization variable at each iteration remains within the convex hull of $\mathcal{P}^n$. We note that other first-order methods, such as projected gradient descent (Bertsekas, 1995) which over the permutahedron can be equally efficient per iteration (Lim and Wright, 2016b), could also be employed. However, FW can produce sparse iterates for certain cases of convex optimization problems, adapts to norm-free smoothness and does not need a projection step (Jaggi, 2013; Bubeck, 2015). Due to its simplicity we use it throughout this work.

The FW update at iteration $k+1$ can be written as

$$\mathbf{x}^{(k+1)}=\alpha\mathbf{x}^{\star}+(1-\alpha)\mathbf{x}^{(k)}, \tag{9}$$

where

$$\mathbf{x}^{\star}=\operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{PH}^n}\langle\nabla f_\mu(\mathbf{x}^{(k)}),\mathbf{x}\rangle, \tag{10}$$

and $\alpha \in [0,1]$ is the step size. The gradient descent direction is based on optimizing a linearization of the objective function f_μ in Eq.(8) over the constraint set, given by

$$\tilde{f_\mu}(\mathbf{x}) = f_\mu(\mathbf{x}^{(k)}) + \langle \nabla f_\mu(\mathbf{x}^{(k)}), \mathbf{x} - \mathbf{x}^{(k)} \rangle, \tag{11}$$

where $\frac{1}{2}\nabla f_\mu(\mathbf{x}^{(k)}) = \mathbf{L_A}\mathbf{x}^{(k)} - \mu\mathbf{H}\mathbf{x}^{(k)}$. The solution $\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathcal{PH}^n} \tilde{f_\mu}(\mathbf{x})$ is necessarily a permutation, since a bounded linear program is optimized at a vertex of the constraint set. To calculate it, we use Hardy-Littlewood-Pólya's rearrangement theorem (Hardy et al, 1952), that states that two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ assume the minimum shuffled inner product when sorted in opposite orders. This happens, for example, when the permutations $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$ order two given vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ descending and ascending, respectively, or equivalently when $\boldsymbol{\tau}(\boldsymbol{\pi}^{-1})$ reorders $\boldsymbol{a}$ while $\boldsymbol{b}$ is kept in its original order. In this situation, by setting $\boldsymbol{a} = \nabla f_\mu(\mathbf{x}^{(k)})$ and $\boldsymbol{b} = \mathbf{e}$, we obtain the permutation $\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathcal{P}^n} \langle \boldsymbol{a}, \mathbf{x}\rangle = \boldsymbol{\pi}^{-1}$ (or in permutation matrix format $\arg\min_{\mathbf{\Pi}\in\mathcal{M}^n} \langle \mathbf{e}, \mathbf{\Pi}^\top \boldsymbol{a}\rangle$) whose inverse ($\boldsymbol{\pi}$) sorts the gradient descending.

Given $\mathbf{x}^\star$, the optimal step size α can then be easily computed in closed form, as $f_\mu(\alpha\mathbf{x}^\star + (1-\alpha)\mathbf{x}^{(k)})$ is quadratic in α. Since its second and first order coefficients are correspondingly $\gamma_2 = (\mathbf{x}^\star - \mathbf{x}^{(k)})^\top(\mathbf{L_A} - \mu\mathbf{H})(\mathbf{x}^\star - \mathbf{x}^{(k)}) = f_\mu(\mathbf{x}^\star - \mathbf{x}^{(k)})$ and $\gamma_1 = \langle \nabla f_\mu(\mathbf{x}^{(k)}), \mathbf{x}^\star - \mathbf{x}^{(k)}\rangle$, the optimizing step within $[0,1]$ is (from convexity and optimizing step we have $\gamma_1 \le f_\mu(\mathbf{x}^\star) - f_\mu(\mathbf{x}^{(k)}) \le 0$)

$$\alpha = \begin{cases} \min\left(\frac{-\gamma_1}{2\gamma_2}, 1\right), & \text{if } \gamma_2 > 0, \\ 0, & \text{if } \gamma_2 \le 0 \ \wedge \ f_\mu(\mathbf{x}^\star) \ge f(\mathbf{x}^{(k)}), \\ 1, & \text{if } \gamma_2 \le 0 \ \wedge \ f_\mu(\mathbf{x}^\star) < f(\mathbf{x}^{(k)}). \end{cases} \tag{12}$$

As previously mentioned, to solve the problem in Eq.(7) we use a continuation scheme that starts from a solution to a convex instance of the problem in Eq.(8). In each iteration we increase μ by multiplying it with a user-defined parameter $\gamma > 1$ and solve the new problem until the solution becomes discrete, which is guaranteed in the concave case. This graduated non-convexity approach (Blake, 1983; Rangarajan and Chellappa, 1990) yields a sequence of relaxed solutions that ultimately lead to a local optimum of the original discrete problem. The procedure can be started at a permutation or any point around the barycenter. However, we have experimentally observed that starting from the ordering of the Fiedler vector, frequently leads to better solutions in terms of 2-SUM value and therefore we use that as a starting point (the continuation scheme almost always converges to a different solution except for pre-Robinsonian cases). We note here that calculating this ordering does not require an extra initial eigen-decomposition, since in our setting this is already performed in order to determine the initial parameter μ_0. The method converges when α reaches near-zero values. Algorithm 1, referred to as Graduated non-Convexity Relaxation (GnCR), summarizes the main steps of this vector-based graduated non-convexity approach to solve the relaxed regularized 2-SUM problem.

Computationally, the proposed method is highly efficient, since each update only requires a single matrix-vector multiplication to compute the gradient

Algorithm 1 The main steps of GnCR.

```
Input  : Laplacian matrix $\mathbf{L_A} = \mathrm{dg}(\mathbf{A1}) - \mathbf{A}$ where $\mathbf{A} \in \mathbb{R}^{n \times n}_{\geq 0}$, initial regularization
         $\mu_0 \leq \lambda_2(\mathbf{L_A})$, initial spectral solution $\mathbf{x}_0$, continuation parameter $\gamma > 1$
Output: Order of final solution $\mathbf{x}$
$\mu \leftarrow \mu_0$
$\mathbf{x} \leftarrow \mathbf{x}_0$
while $\mu \leq \lambda_n(\mathbf{L_A})$ do
    while not converged do
        $\mathbf{x}^\star \leftarrow \arg\min_{\boldsymbol{\pi} \in \mathcal{P}^n} \langle \nabla f_\mu(\mathbf{x}), \boldsymbol{\pi} \rangle$
        $\alpha \leftarrow \arg\min_{a \in [0,1]} f_\mu(a\mathbf{x}^\star + (1-a)\mathbf{x})$
        $\mathbf{x} \leftarrow \alpha\mathbf{x}^\star + (1-\alpha)\mathbf{x}$
    end
    $\mu \leftarrow \gamma\mu$
end
```

vector (where any sparsity and/or low-rank structure of $\mathbf{A}$ can be exploited) and the sorting of the gradient vector, which has complexity $\mathcal{O}(n \log n)$. For example, if $\mathbf{A} = \mathbf{M}\mathbf{M}^\top$ where $\mathbf{M}$ is a sparse matrix with Tn non-zero entries, then the time complexity of each gradient computation $\frac{1}{2}\nabla f_\mu(\mathbf{x}) = \mathbf{D}\mathbf{x} - \mathbf{M}(\mathbf{M}^\top\mathbf{x}) - \mu\mathbf{H}\mathbf{x}$, where $\mathbf{D} = \mathrm{dg}\big(\mathbf{M}(\mathbf{M}^\top\mathbf{1})\big)$, is $\mathcal{O}(Tn)$ due to the sparse matrix with vector multiplication. Likewise, the function evaluation can be calculated as $f_\mu(\mathbf{x}) = \frac{1}{2}\langle \nabla f_\mu(\mathbf{x}), \mathbf{x} \rangle$.

As convergence is concerned, a rate of $\mathcal{O}(\frac{1}{\sqrt{t}})$ (where t is the number of iterations) for non-convex objectives is known for the FW method (Lacoste-Julien, 2016), which applies here since the objective is not necessarily convex for all varying values of μ. Specifically, it is shown that the minimal FW gap is upper bounded by the quantity $\frac{\max\{2h_0, C_{f_\mu}\}}{\sqrt{t+1}}$, for an objective f_μ as defined in Eq. (8). The quantity $h_0 = f_\mu(\mathbf{x}_0) - \min_{\mathbf{x} \in \mathcal{PH}^n} f_\mu(\mathbf{x})$ is the initial global suboptimality, and C_{f_μ} the related curvature constant defined over f_μ. Due to the regularization the latter becomes

$$C_{f_\mu} = \sup_{\substack{\mathbf{x},\mathbf{s} \in \mathcal{PH}^n,\, \alpha \in (0,1], \\ \mathbf{y} = \mathbf{x} + \alpha(\mathbf{s} - \mathbf{x})}} \tfrac{2}{\alpha^2}\mathcal{D}_f(\mathbf{y}, \mathbf{x}) - \tfrac{2\mu}{\alpha^2}\left\|\mathbf{H}(\mathbf{x} - \mathbf{y})\right\|_2^2 \leq C_f,$$

since $\mathcal{D}_f(\mathbf{y}, \mathbf{x}) - \mu\left\|\mathbf{H}(\mathbf{x} - \mathbf{y})\right\|_2^2 \leq \mathcal{D}_f(\mathbf{y}, \mathbf{x})$, where $\mathcal{D}_f$ is the Bregman distance over f. We note that for adaptive FW variants, such as the away steps, the pairwise FW and the fully corrective, a linear and sublinear convergence rate for strongly convex and convex problems has been shown, respectively (Lacoste-Julien and Jaggi, 2015). In our case however, experiments showed that such variants yield negligible benefit in the solution quality, and can even sometimes increase the overall running time (e.g., each step of the fully corrective FW has significant computational demands as a quadratic optimization is realized over the polytope defined by an active set of permutations).

3.5 A Smoothed Regularized Relaxation for the 1-SUM

We now consider the 1-SUM or optimal linear arrangement problem (George and Pothen, 1994), which is harder to analyze as it no longer assumes a quadratic function of the permutation vector. Although a convex function, no regularized form can be employed in this case as in Eq. (8), since $\mu > 0$ cannot control the convexity of the formulation. Additionally, the non-smoothness of this problem resulting from the absolute terms in $\sum_{i,j=1}^{n} A_{ij} \left|x_i - x_j\right|$, prevents the use of a gradient approach (subgradient methods may not be suitable for the regularized formulation that assumes non-convex forms). Therefore, we propose a smooth approximation of the 1-SUM problem in order to enable us to utilize the continuation scheme of Section 3.4.

We employ a pseudo-Huber function (Fountoulakis and Gondzio, 2016) of the form

$$\psi_\delta(x) = \sqrt{\delta^2 + x^2} - \delta, \tag{13}$$

which has bounded and Lipschitz continuous first and second derivatives. Other formulations of the pseudo-Huber functions were previously used in Hartley and Zisserman (2004) and González-Recio and Forni (2011). Figure 1 sketches $\psi_\delta(x)$ for different values of the parameter $\delta > 0$. This form is a smooth approximation of the Huber loss penalty function (Huber, 1992), and approximates $|x|$ as δ approaches zero. Unlike the Huber loss function, which is only first-order differentiable, the pseudo-Huber function is second-order differentiable, a fact essential to the convexity analysis of the continuation process, as shown later in this section. The first two derivatives of the pseudo-Huber function are

$$\psi_\delta^{'}(x) = \frac{x}{\sqrt{\delta^2 + x^2}} = \frac{x}{\psi_\delta(x) + \delta}, \tag{14}$$

$$\psi_\delta^{''}(x) = \frac{\delta^2}{(\delta^2 + x^2)^{\frac{3}{2}}} = \frac{\delta^2}{(\psi_\delta(x) + \delta)^3}, \tag{15}$$

and as $\psi_\delta^{''}(x) > 0$, it is a strictly convex function.

By using the pseudo-Huber loss, we can formulate a smooth approximation for the 1-SUM problem of Eq.(4) for p=1. The new objective is defined as

$$\phi_\delta(\mathbf{x}) \triangleq \sum_{i,j=1}^{n} A_{ij} \psi_\delta(x_i - x_j), \tag{16}$$

and is also convex for non-negative A_{ij} as a non-negative combination of convex functions applied to the linear functions $x_i - x_j$ (this also can be shown from the Hessian of $\phi_\delta(\mathbf{x})$ being diagonally dominant).

The first and second order gradients of $\phi_\delta(\mathbf{x})$ (for symmetric $\mathbf{A}$) assume the simple-to-calculate forms of

$$\frac{\partial \phi_\delta(\mathbf{x})}{\partial x_i} = 2 \sum_{k=1}^{n} A_{ik} \frac{(x_i - x_k)}{\sqrt{\delta^2 + (x_i - x_k)^2}} = 2 \sum_{k=1}^{n} A_{ik} \psi_\delta^{'}(x_i - x_k), \tag{17}$$

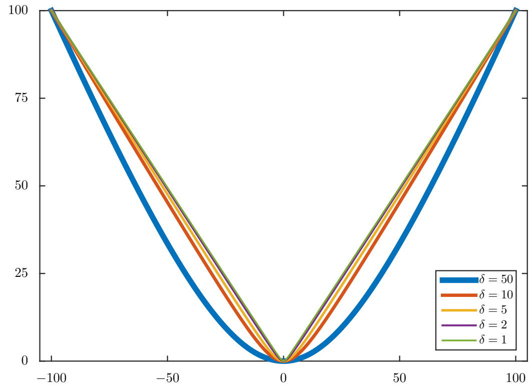


Fig. 1: Plots of the pseudo-Huber function $\psi_\delta(x)$ scaled within $[0, 100]$, for different parameter values δ.

and

$$\frac{\partial^2 \phi_\delta(\mathbf{x})}{\partial x_i \partial x_j} = \begin{cases} -\, 2A_{ij}\ \psi_\delta^{''}(x_i - x_j), & \text{if } i \neq j, \\ \ \ 2\sum\limits_{\substack{k=1 \\ k\neq i}}^{n} A_{ik}\ \psi_\delta^{''}(x_i - x_k), & \text{if } i = j. \end{cases} \tag{18}$$

Since the minimization of $\phi_\delta(\mathbf{x})$ leads to the trivial barycenter solution and in order to apply a continuation scheme, we solve instead the regularized form, defined as

$$\min_{\mathbf{x}\in\mathcal{PH}^n} \left\{ \phi_{\delta,\mu}(\mathbf{x}) \triangleq \phi_\delta(\mathbf{x}) - \mu \left\|\mathbf{Hx}\right\|_2^2 \right\}. \tag{19}$$

It can be observed from Eq.(18), that the Hessian $\nabla^2\phi_\delta(\mathbf{x})$ happens to be equal to the Laplacian $\mathbf{L_G} = \mathrm{dg}\left(\mathbf{G1}\right) - \mathbf{G}$, where $\mathbf{G}$ is a hollow matrix with off-diagonal elements the negated mixed partials, and is centered. This allows us to apply a continuation scheme following the same reasoning as in Section 3.3. Particularly, setting an initial value for $\mu \leq \lambda_2(\nabla^2\phi_\delta(\mathbf{x}))$ enables us to start from a convex instance of the objective $\phi_{\delta,\mu}(\mathbf{x})$, and by gradually increasing μ we can eventually convert it into concave. As in the GnCR algorithm, during each iteration of the continuation process the FW method is used, but here the step size is estimated with a golden section search (Bertsekas, 1995). Unlike GnCR, we do not use the ordering of the Fiedler vector as a starting point for the continuation procedure since the spectral solution approximates 2-SUM problem and not the 1-SUM. However, initial experimentations showed that depending on the similarity matrix (for instance when it is close to pre-Robinsonian) such an initialization could help, but the gain was very small to offset the extra computation. For this method, we start from around the barycenter and specifically, the midpoint between the barycenter $\frac{n+1}{2}\mathbf{1}$ and $\mathbf{e}$. Experimental tests on the sensitivity of the algorithm to the δ parameter reveal that within $[\frac{n}{50}, \frac{n}{10}]$, a sufficiently small δ can be found that ensures good performance. However, very small choices of δ have shown to result to ill-conditioning, something also verified by Fountoulakis and Gondzio (2016). The parameter choice for δ can rely on a grid search in the interval $[\frac{n}{50}, \frac{n}{10}]$ performed in parallel or just set initially by the user. We refer to this “Huberized” 1-SUM algorithm as H-GnCR.

3.6 A Kernel Annealing Approach for the Quasi p-SUM

Depending on the employed objective function, seriation can focus on the global or more localized aspects of ordering (Earle and Hurley, 2015; Hahsler, 2017). Emphasis on the local ordering corresponds to prioritizing neighborhoods of similar objects as opposed to the global ordering that additionally separates dissimilar objects. After having investigated the p-SUM objective $\frac{1}{p}\sum_{i,j} A_{ij}\,|x_i - x_j|^p$ for $p = 1, 2$, we now consider the case of $p < 1$. The motivation is that the optimization becomes more sensitive to small differences $|x_i - x_j|$ than in the $p \geq 1$ case, which encourages more local object placements. Figure 2 exemplifies the effects of localized ordering for three p-SUM cases on a toy dataset.

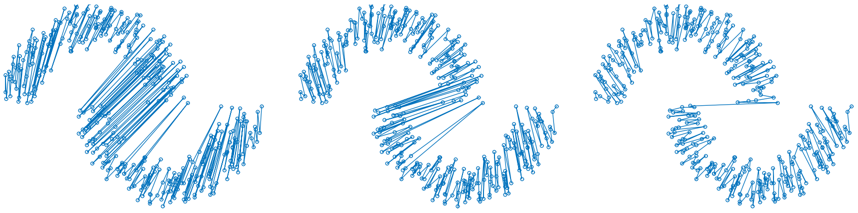


Fig. 2: Seriated points of the Double moons dataset, for the 2-SUM (left), 1-SUM (center), and $\frac{1}{2}$-SUM (right). The order is implied by the lines connecting the points consecutively. The rightmost sequence follows better the local ordering as it avoids moving back and forth between the two moons.

One difficulty with the $p < 1$ case is that the objective is non-convex and non-smooth and prevents the application of the proposed continuation-based optimization scheme. As an alternative, we use an approximation through a series of indefinite functions. In particular, we use the Cauchy distribution-based kernel (Basak, 2008) defined as $K_\sigma(x - y) = \frac{1}{1+\frac{(x-y)^2}{\sigma^2}}$, and we approximate the term $|x_i - x_j|^p$ with the function

$$\xi_\sigma(x) = 1 - K_\sigma(x) = \frac{x^2}{\sigma^2 + x^2}. \tag{20}$$

The scale parameter σ can be used to approximate the effects of the penalty contributions for cases of $p < 1$. Figure 3 presents some plots to demonstrate the behavior of ξ_σ for various values of σ.

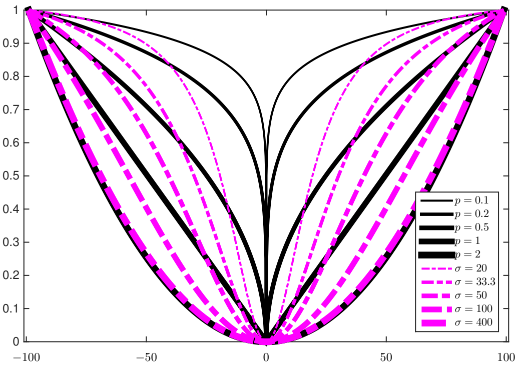


Fig. 3: Plots of the function ξ_σ (dotted lines) and $|x|^p$ (solid), for different values of the parameters σ and p. Both functions are scaled within $[0, 1]$. For larger σ, the former can locally approximate x^2, but for smaller kernel sizes it behaves more similar to $|x|^p$ for $p < 1$.

Unlike the pseudo-Huber function, this kernel-based smoothing function is not convex. The first and second derivatives are

$$\xi_\sigma^{'}(x) = \frac{2\sigma^2 x}{(\sigma^2 + x^2)^2}, \tag{21}$$

$$\xi_\sigma^{''}(x) = \frac{2\sigma^4 - 6\sigma^2 x^2}{(\sigma^2 + x^2)^3}, \tag{22}$$

and the sign of $\xi_\sigma^{''}$ is dependent on the input and the positive scale parameter σ; specifically, it is non-negative when $\sigma \geq x\sqrt{3}$.

Substituting ξ_σ in the objective of Eq.(4), gives

$$\varphi_\sigma(\mathbf{x}) \triangleq \sum_{i,j=1}^{n} A_{ij}\xi_\sigma(x_i - x_j). \tag{23}$$

It can be seen that in order to have $\xi_\sigma(x_i - x_j)$ convex when x_i and x_j are components of $\mathbf{x} \in \mathcal{PH}^n$, we need $\sigma \geq (n-1)\sqrt{3}$. Another observation is that if we restrict attention for $\xi_\sigma(x)$ within $[1-n, n-1]$ and scale accordingly, then we have $\lim_{\sigma\to\infty} \frac{\xi_\sigma(x)}{\xi_\sigma(n-1)} = \frac{x^2}{(n-1)^2}$. This shows that for large σ, Eq.(23) approximates the 2-SUM problem, as normalizing $\xi_\sigma(x)$ by $\frac{1}{\xi_\sigma(n-1)}$ and x^2 by $\frac{1}{(n-1)^2}$ does not affect the optimization.

Since the Hessian $\nabla^2\varphi_\sigma(\mathbf{x})$ is written in a form similar to Eq.(18), the regularized form can be given similarly to that of Section 3.5. That is

$$\min_{\mathbf{x}\in\mathcal{PH}^n} \left\{ \varphi_{\sigma,\mu}(\mathbf{x}) \triangleq \varphi_\sigma(\mathbf{x}) - \mu \left\|\mathbf{Hx}\right\|_2^2 \right\}, \tag{24}$$

where $\varphi_{\sigma,\mu}(\mathbf{x})$ is convex for $\sigma \geq (n-1)\sqrt{3}$ and $\mu \leq \lambda_2(\nabla^2\varphi_\sigma(\mathbf{x}))$.

Although any value $p < 1$ can be potentially useful to recover the local order, here we focus on the $\frac{1}{2}$-SUM objective, which experimentally appeared to be more sensitive in capturing local structure within the proposed setup. We follow a heuristic annealing of the scale parameter σ whose value is gradually decreased. In each step, a continuation scheme is realized with an increasing μ until the problem becomes concave (based on empirical observations, we only need to ensure we start with a convex setup for $\varphi_{\sigma,\mu}(\mathbf{x})$ for the initial and largest σ value, while the remaining steps may start from being indefinite). For experimentation, we let σ vary within the interval $[\frac{n}{5}, 4n]$ in order for ξ_σ to capture various profiles of $|x|^p$. Each solution obtained from a σ step is recorded and used to initialize the subsequent step but from a shifted location to avoid solution stagnation. We finally report the solution that amongst the recorded minimizes the $\frac{1}{2}$-SUM value. However, the method can be used independently of the p-SUM formulation to suit a given application. For example, one can instead seek the solution that minimizes the δ_{count} measure from Section 4.2 or any other measure that captures local order. It has to be noted that although $\xi_\sigma(x)$ is, as shown in Figure 3, only a rough approximation of $|x|^{\frac{1}{2}}$, when used in an annealing scheme of the σ parameter with restarts, it results to good solutions in terms of the $\frac{1}{2}$-SUM value. We refer to this heuristic approximation as C-GnCR, and we summarize its main steps in Algorithm 2.

Algorithm 2 The main steps of C-GnCR.

```
Input  : Similarity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}_{\geq 0}$, continuation rate $\gamma > 1$, decreasing series of kernel
         sizes $\sigma_1, \dots, \sigma_m$, and initial regularization $\mu_0 \leq \lambda_2(\nabla^2 \varphi_{\sigma_1})$
Output: Order of final solution $\mathbf{x}$
  $\mathbf{x}^{(0)} \leftarrow \mathbf{e}$
  for $k = 1;\ k \leq m;\ k \leftarrow k + 1$ do
      $\sigma \leftarrow \sigma_k$
      $\mu \leftarrow \mu_0$
      $\mathbf{x} \leftarrow \frac{1}{2}(\mathbf{x}^{(k-1)} + \frac{n+1}{2}\mathbf{1})$
      while $\mathbf{x} \notin \mathcal{P}^n$ do
          while not converged do
              $\mathbf{x}^\star \leftarrow \arg\min_{\boldsymbol{\pi} \in \mathcal{P}^n} \langle \nabla \varphi_{\sigma_k,\mu}(\mathbf{x}), \boldsymbol{\pi} \rangle$
              $\alpha \leftarrow \arg\min_{a \in [0,1]} \varphi_{\sigma_k,\mu}(a\mathbf{x}^\star + (1-a)\mathbf{x})$
              $\mathbf{x} \leftarrow \alpha\mathbf{x}^\star + (1-\alpha)\mathbf{x}$
          end
          $\mu \leftarrow \gamma\mu$
      end
      $\mathbf{x}^{(k)} \leftarrow \mathbf{x}$
end
$k^\star = \arg\min_{k \in \{1,\dots,m\}} \sum_{i,j} A_{ij} \left| x_i^{(k)} - x_j^{(k)} \right|^{\frac{1}{2}}$
  $\mathbf{x} \leftarrow \mathbf{x}^{(k^\star)}$
```

The recent work of Recanati et al (2018) on robust seriation is using a formulation that controls error contributions to reduce sensitivity on outliers. In this respect, this can be an additional motivation for using the Cauchy-based kernel here, as for small σ values it has a similar limiting effect. We note that we also tested other approximation functions, such as the Gaussian (the Laplacian and the log-kernel are not applicable since they both are non-smooth functions), but the Cauchy-based shows the best overall performance when used in the proposed annealing process (see Table 8). Nonetheless, the choice of the approximating function may depend on the given problem.

4 Experimental Results

We present a series of experiments in order to compare the proposed algorithms[5] with other relevant methods in terms of both utility and scalability. Section 4.1 presents experimental results from comparisons with state-of-the-art algorithms for seriation and various heuristics that approximately solve the QAP. We use several datasets with different characteristics ranging from synthetic to real. Section 4.2 contains a detailed comparison among the different p-SUM algorithms, and highlights the utility of each one in sequencing problems using interpretable supervised measures. Finally, in Section 4.3 we test the scalability of the algorithms, and in Section 4.4 we test their performance on image seriation problems.

4.1 Benchmark Evaluation

In this section we experiment with the following methods:

- *GnCR*: the graduated non-convexity 2-SUM relaxation in Algorithm 1.
- *H-GnCR*: the 1-SUM method relying on the pseudo-Huber approximation.
- *C-GnCR*: the annealing-based quasi $\frac{1}{2}$-SUM method in Algorithm 2.
- *Spectral*$_A$: the spectral method (Barnard et al, 1993) that sorts the entries of the Fiedler vector of the unnormalized Laplacian.
- *Spectral*$_B$: the spectral method (Ding and He, 2004) that sorts the entries of the Fiedler vector of the normalized Laplacian.
- *vRCR* (Vector-regularized convex 2-SUM relaxation): minimizes problem (8) using an interior point solver (we only use the tie-breaking constraint). Its implementation was provided to us by the authors (Lim and Wright, 2014).
- *vRCR*$_2$: variant of *vRCR* that minimizes problem (8) using FW on the permutahedron with the tie-breaking constraint (Lim and Wright, 2014); also used to solve problems (19) and (24).
- *FAQ*: the fast approximate QAP method (Vogelstein et al, 2015), based on the relaxation on the Birkhoff polytope and the Frank-Wolfe method.

[5] The code for the proposed algorithms and other evaluated methods is included in our Matlab toolbox for seriation, available at `http://pcwww.liv.ac.uk/~goulerma/software/seriation.zip`.

– *SA*: a simulated annealing-based optimizer (Brusco and Stahl, 2000).

We note that other population-based heuristics (Kennedy and Eberhart, 1995), (Yang, 2008) were also tried, but they showed to perform worse than SA and therefore were not included in our results. Each algorithm is implemented in MATLAB ver.9.3. For timing comparisons we use a 2.93 GHz 12-Core Intel Xeon desktop with 16 GB of memory. Typical parameters are $\gamma = 1.05$ and μ_0 set to the second smallest eigenvalue of each corresponding Hessian. Sections 3.5 and 3.6 discuss in detail the parameter choices for δ and σ for the H-GnCR and C-GnCR methods, respectively.

We selected a range of real and synthetic datasets, associated either with a similarity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}_{\geq 0}$, or a data matrix $\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_n]^\top$ for which case we assume that $A_{ij} = \left|\mathbf{m}_i^\top \mathbf{m}_j\right|$. These sets include:

- Real datasets from the `seriation` R-package (Hahsler et al, 2008):
  – *Munsingen*: a 59×70 binary matrix $\mathbf{M}$.
  – *Psych24*: a 24×24 similarity matrix $\mathbf{A}$.
  – *Gene expression (wood)*: a 136×6 $\mathbf{M}$.
  – *Zoo*: a 101×16 $\mathbf{M}$.
- Other real datasets :
  – *Votes*: a 232×16 binary matrix $\mathbf{M}$ (Dheeru and Karra Taniskidou, 2017).
  – *Facebook ego-network*: a 324×324 similarity matrix $\mathbf{A}$ (Leskovec and Krevl, 2014).
  – *Elutriation gene expression*: a 301×14 $\mathbf{M}$ (Alter et al, 2000).
- Datasets from the SuiteSparse Matrix Collection (Davis and Hu, 2011):
  – *CAT*: a 85×85 similarity matrix $\mathbf{A}$.
  – *DWT*: a 59×59 similarity matrix $\mathbf{A}$.
- Synthetic datasets:
  – *Markov chains* (Lim and Wright, 2014): a 100×100 $\mathbf{A}$, that is the covariance matrix of 50 independent linear Markov chains, with each one generated as $X_i = bX_{i-1} + \epsilon_i, \quad i \in \{1, \ldots, 100\}$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $b = 0.999$, and $\sigma = 0.5$.
  – *Artificial graves*: a 100×200 binary $\mathbf{M}$, that models the incidence of artifacts in graves assuming that the occurrence rate of each artifact follows a Gaussian curve. Specifically, each grave is associated with a time-point $t_i \sim \mathcal{U}(0, 1)$. The probability that the jth artifact will appear in a grave is defined as $\Pr(M_{i,j} = 1) = \alpha_i \beta_j \exp(-\frac{\|t_i - \mu_j\|}{2\sigma_j^2})$, where $\alpha_i \sim \text{Lognormal}(\log(0.3), .3)$, $\beta_j \sim \mathcal{U}(0, 1)$, $\mu_j \sim \mathcal{U}(-1, 2)$, and the standard deviation σ_j is distributed with a truncated Jeffrey's prior between $[0.01, 0.25]$.
  – *Robinsonian*$_N$: formed from an $N \times M$ binary 0–1 matrix $\mathbf{M}$ that has the consecutive ones property (C1P), that is its rows can be rearranged such that the ones in every column form a single contiguous sequence (Fulkerson and Gross, 1965).

	GnCR	Spectral_A	Spectral_B	FAQ	SA	vRCR_2	vRCR
Munsingen	0	0.440	0.500	0.534	0.135	0.665	0.617
Artificial graves	0.013	1.244	0.206	0.287	0	1.723	1.632
Markov chains	0.003	0.177	0.149	0.001	0	0.524	0.244
Psych24	0	0.018	0.041	0.075	0	0.164	0.146
Zoo	0	0.282	0.117	0.009	0	0.398	0.246
Gene expression	0.002	0.051	0.125	0	0	0.177	0.119
Double moons	0	0.004	0.010	0	0	0.178	0.178
Facebook	0	0.767	0.894	0.293	0.077	2.707	1.831
CAT	0	0.090	0.105	0.198	0.019	0.576	0.413
DWT	0	0.125	0.127	1.668	0.298	1.386	0.679
Votes	0	0.001	0.008	0	0	0.061	0.036
Elutriation	0.005	0.114	0.111	0	0	0.150	0.096
Average	**0.002**	0.276	0.199	0.255	0.044	0.726	0.520

Table 1: Deviation from the best 2-SUM value across the 12 datasets.

	H-GnCR	Spectral_A	Spectral_B	FAQ	SA	vRCR_2
Munsingen	0	0.155	0.199	0.470	0.030	0.522
Artificial graves	0.050	0.421	0.114	0.273	0	0.655
Markov chains	0.011	0.090	0.064	0.001	0	0.235
Psych24	0.076	0.015	0.026	0.073	0	0.150
Zoo	0.026	0.155	0.067	0.005	0	0.236
Gene expression	0.037	0.033	0.066	0	0	0.215
Double moons	0.006	0.023	0.028	0.175	0	0.335
Facebook	0.018	0.346	0.403	0.175	0	1.081
CAT	0.033	0.085	0.115	1.311	0	0.504
DWT	0	0.154	0.157	1.433	0.153	0.827
Votes	0.001	0.001	0.004	0	0	0.322
Elutriation	0.006	0.057	0.055	0	0	0.074
Average	0.022	0.128	0.108	0.326	**0.015**	0.429

Table 2: Deviation from the best 1-SUM value across the 12 datasets.

- *Double moons*: a 100 × 100 **A**, that generates points that form two half moons in 2-D space. The **A** similarity matrix is computed using a Gaussian kernel (Baudat and Anouar, 2001).

We evaluate the utility of the proposed algorithms by comparing their objective function values with a number of seriation methods that can solve different p-SUM problems. All evaluations are run over multiple randomly shuffled instances of the available datasets. We additionally use the weighted Robinson events (WRE) measure (Hahsler et al, 2008) to assess the Robinsonian structure of a similarity matrix. Since the values on different datasets are not comparable, for interpretability we report a normalized value for each measure that quantifies the deviation from the best performer for that dataset. For the ith dataset the deviation for the jth algorithm is defined as

$$\Delta_{i,j} = \frac{score_{i,j} - best_i}{best_i}. \tag{25}$$

Additionally, we report the overall average deviations for each algorithm across all datasets. Tables 1, 2 and 3 show the normalized deviation from the

	C-GnCR	$\text{Spectral}_{\text{A}}$	$\text{Spectral}_{\text{B}}$	FAQ	SA	vRCR_2
Munsingen	0	0.077	0.100	0.374	0.017	0.181
Artificial graves	0.031	0.198	0.080	0.202	0	0.269
Markov chains	0.001	0.045	0.028	0.001	0	0.078
Psych24	0.007	0.013	0.019	0.052	0	0.039
Zoo	0.005	0.079	0.034	0.004	0	0.080
Gene expression	0.002	0.019	0.033	0	0	0.081
Double moons	0.005	0.026	0.028	0.071	0	0.069
Facebook	0.003	0.193	0.214	0.104	0	0.461
CAT	0.030	0.143	0.163	0.817	0	0.286
DWT	0	0.131	0.129	0.636	0.034	0.292
Votes	0	0.001	0.003	0	0	0.024
Elutriation	0.003	0.027	0.026	0	0	0.038
Average	0.007	0.079	0.072	0.189	**0.004**	0.158

Table 3: Deviation from the best $\frac{1}{2}$-SUM value across the 12 datasets.

	GnCR	$\text{Spectral}_{\text{A}}$	$\text{Spectral}_{\text{B}}$	FAQ	SA	vRCR_2	vRCR
Munsingen	0	0.024	0.033	0.023	0.005	0.032	0.029
Artificial graves	0.002	0.090	0.021	0.026	0	0.124	0.119
Markov chains	0.003	0.106	0.108	0.001	0	0.262	0.133
Psych24	0	0.021	0.043	0.079	0	0.169	0.159
Zoo	0.001	0.135	0.091	0	0.001	0.197	0.127
Gene expression	0.003	0.063	0.143	0	0	0.208	0.141
Double moons	0	0.002	0.004	0	0	0.019	0.019
Facebook	0	0.033	0.037	0.010	0.003	0.085	0.057
CAT	0	0.001	0.005	0.012	0.001	0.023	0.017
DWT	0	0.005	0.005	0.026	0.004	0.015	0.008
Votes	0	0	0.005	0	0	0.030	0.017
Elutriation	0.025	0.443	0.290	0	0	0.447	0.271
Average	0.003	0.077	0.065	0.015	**0.001**	0.134	0.091

Table 4: Deviation from the best WRE score when solving the 2-SUM across the 12 datasets.

best p-SUM value for each algorithm and for the 12 datasets. FAQ, SA and vRCR_2 directly solve each corresponding p-SUM problem. For the 1-SUM and $\frac{1}{2}$-SUM, the vRCR method is not included in the comparisons as it is designed to solve the 2-SUM, but we do compare with the two spectral methods as their 2-SUM solutions can perform well in near noiseless cases. Table 1 shows the normalized deviation for the 2-SUM and demonstrates that the proposed GnCR algorithm outperforms the others for the 2-SUM criterion (boldfaced table entries denote best performance). Unlike previous convex relaxation methods (Lim and Wright, 2014; Fogel et al, 2015), the proposed method can outperform both spectral methods without the use of any extra ordering information. The performance difference was assessed with a sign test, which showed that GnCR performs better than both spectral methods with a p-value of 0.0084 at a significance level of 0.05 (a Bonferroni correction for the two hypotheses tested was applied). Similarly, Table 2 shows the results for the 1-SUM case, where it is clear that the proposed H-GnCR outperforms all competing ones except SA. Nonetheless, it achieves a normalized deviation

	H-GnCR	Spectral_A	Spectral_B	FAQ	SA	vRCR_2
Munsingen	0	0.023	0.033	0.023	0.004	0.074
Artificial graves	0.011	0.095	0.026	0.031	0	0.150
Markov chains	0.017	0.106	0.108	0.001	0	0.250
Psych24	0.158	0.029	0.052	0.087	0	0.307
Zoo	0.035	0.138	0.094	0.003	0	0.244
Gene expression	0.099	0.064	0.144	0.001	0	0.463
Double moons	0.001	0.006	0.008	0.004	0	0.102
Facebook	0.002	0.035	0.039	0.012	0	0.100
CAT	0.004	0.011	0.015	0.022	0	0.063
DWT	0	0.009	0.009	0.030	0.008	0.048
Votes	0.001	0.001	0.005	0	0	0.366
Elutriation	0.042	0.443	0.290	0.001	0	0.418
Average	0.031	0.080	0.069	0.018	**0.001**	0.215

Table 5: Deviation from the best WRE score when solving the 1-SUM across the 12 datasets.

	C-GnCR	Spectral_A	Spectral_B	FAQ	SA	vRCR_2
Munsingen	0	0.024	0.034	0.129	0.013	0.062
Artificial graves	0.006	0.091	0.022	0.092	0	0.123
Markov chains	0.003	0.106	0.108	0.003	0	0.174
Psych24	0	0.025	0.047	0.202	0.005	0.122
Zoo	0.011	0.138	0.094	0.010	0	0.167
Gene expression	0.002	0.062	0.143	0.004	0	0.362
Double moons	0	0.005	0.007	0.065	0.004	0.037
Facebook	0	0.035	0.040	0.027	0.004	0.094
CAT	0	0.007	0.011	0.199	0.007	0.047
DWT	0	0.009	0.009	0.114	0.014	0.039
Votes	0	0	0.005	0.002	0.002	0.062
Elutriation	0.038	0.443	0.290	0	0	0.489
Average	0.005	0.079	0.068	0.070	**0.004**	0.148

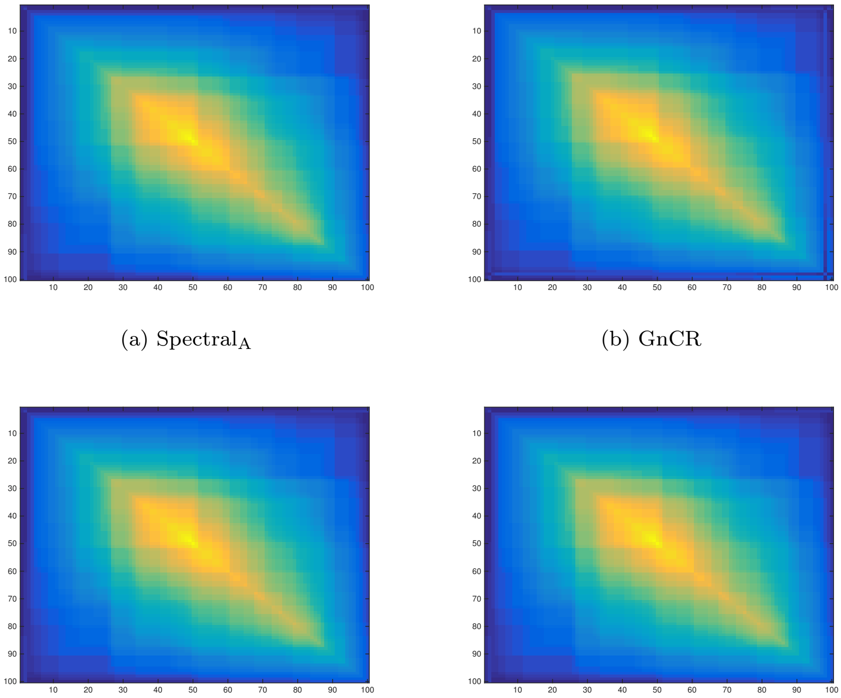
Table 6: Deviation from the best WRE score when solving the $\frac{1}{2}$-SUM across the 12 datasets.

very close to the best. Lastly, Table 3 summarizes results for the $\frac{1}{2}$-SUM case, where the proposed C-GnCR achieves the second best overall performance, having an insignificant difference from the best performer, which is SA. Tables 4, 5 and 6 show similar trends for the WRE measure, where the proposed methods achieve scores very close to the best performing method, SA.

In a second set of experiments we test the consistency of the proposed algorithms on artificial Robinsonian datasets of size $n = 100$ and $n = 500$ by comparing them against the spectral solution that can find the optimal solution in noiseless cases. For each problem size we generate 20 randomly permuted instances and find the best reordering for each dataset. We measure the 2-SUM values of each algorithm in order for the comparison to be consistent with that of the spectral solution. Table 7 shows the average 2-SUM values and running times of each algorithm. We can see that for both datasets GnCR achieves the optimal score, which is owing to the spectral initialization, and C-GnCR outperforms H-GnCR. However, C-GnCR appears to be much slower compared

	GnCR	H-GnCR	C-GnCR	Spectral$_\mathrm{A}$	
		2-SUM			
Robinsonian$_{100}$	**0.889**	0.923	0.899	**0.889**	$(\times 10^{8})$
Robinsonian$_{500}$	**2.979**	3.177	3.018	**2.979**	$(\times 10^{11})$
		Running time (s)			
Robinsonian$_{100}$	0.368	3.023	42.424	**0.002**	
Robinsonian$_{500}$	0.919	15.390	228.267	**0.038**	

Table 7: Average 2-SUM values and running times for the three proposed algorithms and Spectral$_\mathrm{A}$ using pre-Robinsonian matrices of sizes $n = 100$ and $n = 500$.



(a) Spectral$_\mathrm{A}$

(b) GnCR

(c) H-GnCR

(d) C-GnCR

Fig. 4: Reconstructions for the Robinsonian$_{100}$ dataset, for Spectral$_\mathrm{A}$ with perfect reconstruction and the three proposed methods.

to the rest of the methods due to the kernel size annealing process. Overall, the scores are comparable which supports that the underlying optimization mechanisms of the proposed methods behave consistently. Figure 4 provides graphical representations of the quality of the different reconstructions.

	$\frac{1}{2}$-SUM value	Running time (s)
Cauchy kernel	**7,664**	**6.6**
Gaussian kernel	7,690	31.2

Table 8: Average performance and running time of C-GnCR from 20 runs using the Artificial graves dataset, for two different kernel approximations.

δ_{count}			
	GnCR methods	FAQ	SA
2-SUM	**59.0**	**59.0**	59.6
1-SUM	29.8	**12.7**	13.4
$\frac{1}{2}$-SUM	18.6	**1.0**	1.2
2-SUM$_{sup}$			
	GnCR methods	FAQ	SA
2-SUM	**4,603**	**4,603**	4,605
1-SUM	3,656	4,268	**3,574**
$\frac{1}{2}$-SUM	3,494	**3,333**	3,345

Table 9: Supervised evaluation of seriation quality for different algorithms solving the general p-SUM using the Double moons dataset.

4.2 Comparison on the different p-SUM objectives

We now examine the utility in terms of seriation quality for each algorithm when solving different instances of the p-SUM. We first test the performance of C-GnCR when using different kernel approximations in Table 8. It can be seen that the Cauchy-based outperforms the Gaussian-based with respect to both objective value and running time.

Subsequently, we ascertain the ability of the different algorithms to solve the $\frac{1}{2}$-SUM problem in situations where local ordering is of particular interest. For this setting we employ 10 random repetitions of the Double moons dataset (see Figure 2) with size $n = 400$, and use the class membership to each moon to evaluate a resulting ordering $\boldsymbol{\pi}$. If we define the class label matrix as

$$C_{ij} = \begin{cases} 1, & \text{if } i, j \in \text{ same class,} \\ 0, & \text{otherwise,} \end{cases} \tag{26}$$

the first measure we propose quantifies the number of times a seriation algorithm places objects from different classes next to each other as

$$\delta_{count}(\boldsymbol{\pi}, \mathbf{C}) \triangleq \sum_{i=1}^{n-1} \left(1 - C_{\pi(i)\pi(i+1)}\right). \tag{27}$$

The second measure we use, penalizes objects from the same class that are placed far apart. It has the same form as the 2-SUM objective, but the similarity matrix $\mathbf{A}$ is replaced with the above $\mathbf{C}$; that is

$$\text{2-SUM}_{sup}(\boldsymbol{\pi}, \mathbf{C}) \triangleq \boldsymbol{\pi}^\top \mathbf{L_C} \boldsymbol{\pi}. \tag{28}$$

	Hamiltonian path		
	GnCR methods	FAQ	SA
2-SUM	**128.0**	109.2	112.8
1-SUM	144.7	184.6	**204.7**
$\frac{1}{2}$-SUM	154.6	201.3	**248.1**

Table 10: Hamiltonian path measures for different algorithms that solve the general p-SUM for the Facebook dataset.

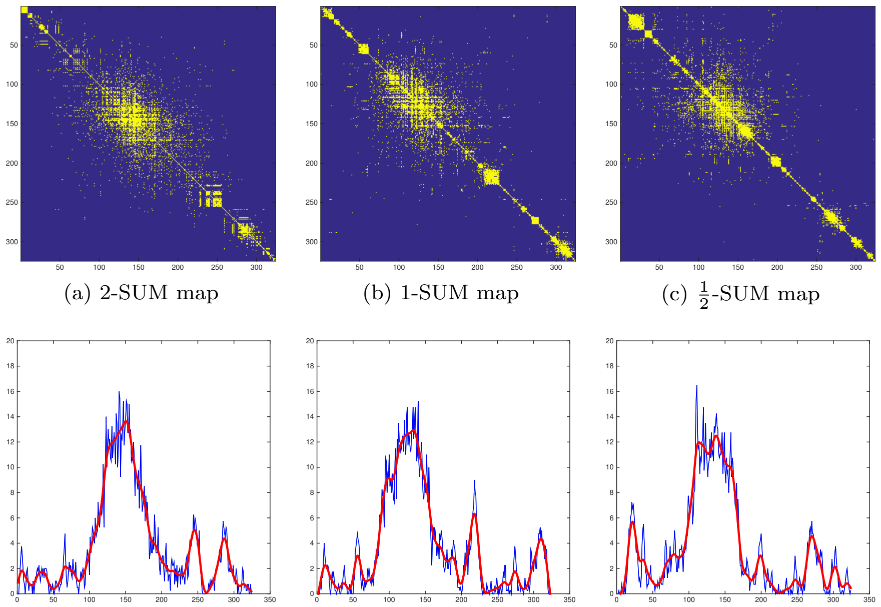
Table 9 shows the average values from both measures above. It can be seen that the algorithms solving the $\frac{1}{2}$-SUM perform much better as the sought seriation is more sensitive to the local structure. The algorithms used for the 2-SUM, that is GnCR, FAQ and SA, show a similar performance in both measures. For the 1-SUM, the proposed H-GnCR achieves the second best performance for the 2-SUM$_{sup}$, but it is the worst with respect to δ_{count}, owing to the fact that it solves a smooth approximation of the 1-SUM in contrast with FAQ and SA. For the $\frac{1}{2}$-SUM case, the proposed C-GnCR performs worse in terms of both measures, again due to the underlying approximation, but maintains a 2-SUM$_{sup}$ value very close to the best.

We further examine the effects of solving the general p-SUM on the Facebook ego-network dataset, which contains a network of connections among friends of a user (McAuley and Leskovec, 2012). In this case, seriation can be used to reveal node clustering patterns, as orderings that are more sensitive to the local structure can highlight tighter social circles. Figures 5a-5c show the effects of solving the 2-SUM, 1-SUM and $\frac{1}{2}$-SUM problems with SA, chosen here for its objective approximation quality. Figures 5d-5f display the corresponding cluster crossing curves. These are calculated as in Ding and He (2004) via summing fractions of pairwise similarities between objects. They can indicate cluster overlapping and minimum values are attained at boundaries between clusters. In this experiment we can see that smaller p yields increased number of clusters (more valleys) of smaller sizes (narrower peaks).

Since for this dataset we do not have distinct class labels, we use the Hamiltonian path (Hahsler et al, 2008) to assess the local ordering of the resulting seriation. Table 10 presents the performance of the proposed methods against FAQ and SA, across different p-SUM objectives. We can see that for all methods, as we reduce p, the measure increases (since we use similarities) which suggests more localized orderings. Furthermore, for the 2-SUM objective, GnCR outperforms both FAQ and SA, while for the 1-SUM and $\frac{1}{2}$-SUM, the proposed H-GnCR and C-GnCR perform worse.

4.3 Envelope Reduction on Big NASA Datasets

In order to test the scalability of our proposed algorithms at an even larger scale, we apply them to a collection of large ($n > 1,000$) sparse matrices taken from the SuiteSparse Matrix Collection (Davis and Hu, 2011). The quality



(a) 2-SUM map (b) 1-SUM map (c) $\frac{1}{2}$-SUM map

(d) 2-SUM crossing (e) 1-SUM crossing (f) $\frac{1}{2}$-SUM crossing

Fig. 5: Seriation of the Facebook dataset for different p-SUM losses using the SA method. The top row displays the seriated similarity matrices and the bottom row contains the corresponding cluster crossing curves.

of seriation can also be measured by the bandwidth and the envelope of the symmetric similarity matrix. The bandwidth is defined to be the maximum width of all rows (with the row width defined to be the largest distance between any non-zero element within the row and the diagonal), and the envelope size is defined to be the sum of all row widths (George and Pothen, 1994). The goal of this experiment is to examine whether the proposed methods can successfully reduce the envelope size of sparse matrices of size up to $n = 36,519$. Specifically, we use four sparse NASA datasets:

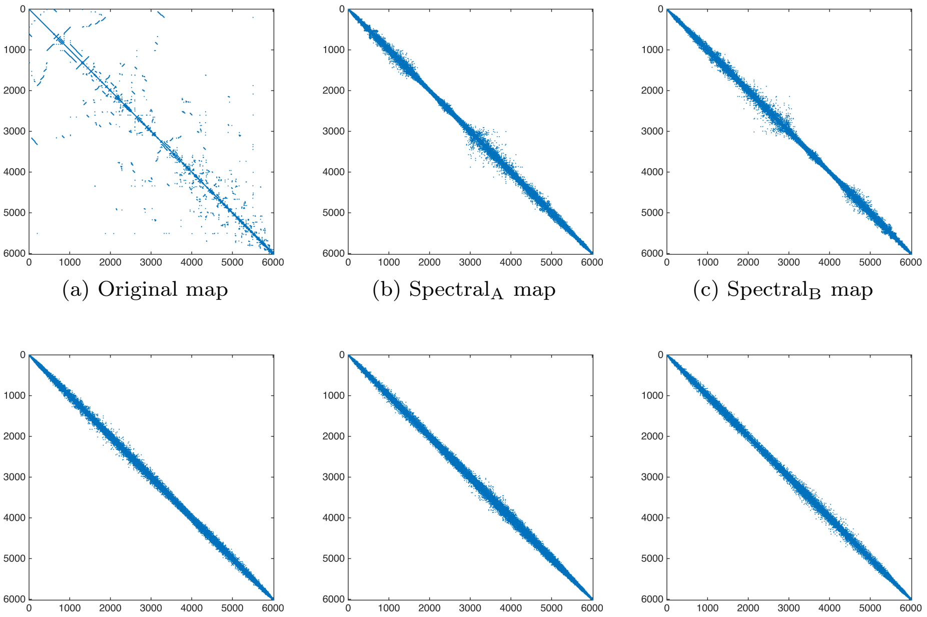
- *BARTH4*: a 6,691×6,691 binary asymmetric $\mathbf{A}_u$ with 26,439 non-zero elements, symmetrized as $\mathbf{A} = \mathbf{A}_u^\top \vee \mathbf{A}_u$ (where $\vee$ denotes elementwise OR).
- *BARTH5*: a 15,606×15,606 binary asymmetric $\mathbf{A}_u$ with 61,484 non-zero elements, symmetrized as before.
- *PWT*: a 36,519×36,519 binary symmetric $\mathbf{A}$ with 326,107 non-zero elements.
- *CAN*: a 1,072×1,072 binary symmetric $\mathbf{A}$ with 11,372 non-zero elements.

The only feasible algorithms for this setting of experiments are the three proposed methods and the two spectral methods. We present the envelope size and bandwidth of each reordered matrix as in Barnard et al (1993) and also report the p-SUM objective values and running times in Table 11. In terms of

	$\text{Spectral}_{\text{A}}$	$\text{Spectral}_{\text{B}}$	GnCR	H-GnCR	C-GnCR
Envelope size					
BARTH4	328,112	325,968	**314,006**	356,632	315,109
BARTH5	1,373,825	1,381,331	**1,373,125**	1,385,795	1,378,337
PWT	5,021,435	5,091,311	5,154,317	**4,713,766**	4,714,138
CAN	54,622	54,015	52,617	**52,523**	54,710
Bandwidth					
BARTH4	872	873	391	**363**	**363**
BARTH5	688	692	**594**	1,220	792
PWT	1071	989	903	729	**725**
CAN	312	**308**	417	804	941
2-SUM ($\times 10^{10}$)					
BARTH4	0.0105	0.0106	**0.0079**	0.0102	0.0081
BARTH5	0.0612	0.0612	**0.0568**	0.0726	0.0649
PWT	0.4565	0.4624	0.4198	0.3834	**0.3830**
CAN	0.0018	0.0018	**0.0017**	0.0022	0.0025
1-SUM ($\times 10^{7}$)					
BARTH4	0.1287	0.1286	**0.1234**	0.1403	0.1237
BARTH5	0.5455	0.5451	**0.5431**	0.5495	0.5463
PWT	2.8772	2.8901	2.8972	**2.6517**	2.6520
CAN	0.0337	0.0338	0.0328	**0.0306**	0.0318
$\frac{1}{2}$-SUM ($\times 10^{6}$)					
BARTH4	0.1910	0.1908	0.1895	0.2014	**0.1892**
BARTH5	0.6453	0.6452	0.6485	**0.6391**	0.6415
PWT	2.5703	2.5936	2.6314	**2.4950**	2.4955
CAN	0.0564	0.0565	0.0557	**0.0523**	0.0542
Running time (s)					
BARTH4	4.8	**1.9**	11.4	357.6	1,582.6
BARTH5	10.0	**8.7**	59.2	1,198.6	3,293.5
PWT	14.4	**14.2**	72.7	3,773.4	17,679.9
CAN	**0.1**	**0.1**	0.8	15.0	191.2

Table 11: Envelope size, bandwidth, objective function and running time for each algorithm for the four datasets.

envelope size, we can see that GnCR and H-GnCR show the best performance across all datasets. Nevertheless, C-GnCR maintains a good performance as well, very close to the best. Regarding the bandwidth, the proposed methods H-GnCR, GnCR and C-GnCR achieve best for the first three datasets, and $\text{Spectral}_{\text{B}}$ for the last one. It is notable however, that GnCR maintains a low bandwidth very close to the best for all cases. With regard to the 2-SUM, GnCR shows the best performance in all datasets apart from PWT where C-GnCR scores best. For the 1-SUM, GnCR and H-GnCR outperform the other methods for the first two (BARTH4, BARTH5) and last two (PWT, CAN) datasets, respectively. Results for the $\frac{1}{2}$-SUM objective show that the proposed H-GnCR scores best for the last three datasets, while C-GnCR is best for BARTH4. Again, it is notable that C-GnCR scores very close to the best for the rest datasets. Finally, we can see that all algorithms maintain a reasonable running time as the problem size increases and thus prove to be very scalable.



(a) Original map (b) Spectral_A map (c) Spectral_B map

(d) GnCR map (e) H-GnCR map (f) C-GnCR map

Fig. 6: Original similarity map for the BARTH4 dataset, and reordered versions produced by the two spectral and proposed algorithms.

Figure 6 gives a visual representation of the original BARTH4 matrix and the reordered matrices of the tested algorithms. It can be seen that all methods show similar behavior and successfully reduce the envelope of the corresponding matrix.

4.4 Seriation of Images

In this section we explore seriation on complex patterns, such as images, where their ordering according to their semantic content may be of interest. An optimized linear ordering can reveal whether there are smooth variations across the patterns. Possible applications include browsing collections of photos while preserving scene similarity, exploring patterns of pathology amongst medical images, or sequencing video frames.

In order to test the performance of the proposed methods on images we use two datasets. A set of 40 rotating teapot images (Weinberger and Saul, 2004) captured at 4.5 degrees apart, spanning 180 degrees and categorized in 8 classes, and a set of 585 images from the MSRC2 database[6] categorized in

[6] https://www.microsoft.com/en-us/research/project/image-understanding

	Teapots	MSRC2
Spectral_A	26	520
Spectral_B	28	520
GnCR	8	496
H-GnCR	**7**	488
C-GnCR	8	477
FAQ (2)	8	508
FAQ (1)	8	485
FAQ ($\frac{1}{2}$)	12	462
SA (2)	14	512
SA (1)	8	499
SA ($\frac{1}{2}$)	**7**	**435**
KS	9	480

Table 12: δ_{count} measure for each algorithm (parenthesized numbers indicate the value of p) for the teapots and MSRC2 image datasets.

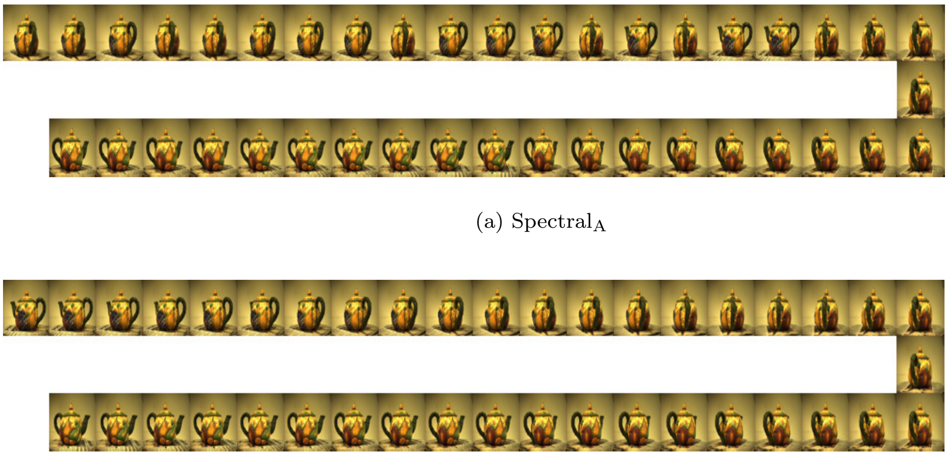
	$\|\tau\|$	PPC
Spectral_A	0.661	0.664
Spectral_B	0.666	0.664
GnCR	0.887	0.905
H-GnCR	**1**	**1**
C-GnCR	0.892	0.909
FAQ (2)	0.884	0.903
FAQ (1)	0.617	0.624
FAQ ($\frac{1}{2}$)	0.066	0.150
SA (2)	0.884	0.903
SA (1)	0.997	0.999
SA ($\frac{1}{2}$)	**1**	**1**
KS	0.764	0.782

Table 13: Kendall's $|\tau|$ and PPC scores between final solution and true underlying ordering, for the Teapots dataset (parenthesized numbers indicate the value of p). Values closer to 1 indicate better ordering agreement.

20 distinct classes. We represent images as bag-of-visual-words (Csurka et al, 2004), that is histograms of quantized local descriptors densely sampled using overlapping matches of each image (Tuytelaars, 2010). In this setup, we use the SIFT (Lowe, 2004) vector descriptors[7] and image patches of 12 pixels long overlapping every 6 pixels. For the bag-of-visual-words representation we use k-means with a cluster size of 500. Then, to derive the similarity matrix we use the exponentiated χ^2 distance, as in Quadrianto et al (2010). For comparison, we also include the algorithm kernelized sorting (KS) (Quadrianto et al, 2010) that can align a set of images according to a given template, which in this case is an one dimensional grid.

Numerical results in Table 12 rely on the δ_{count} in order to evaluate how closely images from the same category are placed. We use different p-SUM objectives to obtain more local ordering solutions. For the Teapots dataset we can see that H-GnCR and SA($\frac{1}{2}$) achieve the optimum δ_{count} value. It is also

[7] SIFT descriptors are extracted using the VL_FEAT toolbox: `http://www.vlfeat.org`.



(a) Spectral$_{\mathrm{A}}$

(b) H-GnCR

Fig. 7: Image sequence of a teapot captured in different angles, seriated using Spectral$_{\mathrm{A}}$ and H-GnCR.

notable that all three proposed methods maintain a very low value across the different objectives, while this is not the case for all SA and FAQ versions. For the MSRC2 dataset, SA($\frac{1}{2}$) scores the best δ_{count}, while C-GnCR achieves the third best.

Figure 7 shows the seriated teapots for the spectral (Barnard et al, 1993) and H-GnCR methods, while Figures 8 and 9 the results on MSRC2 for C-GnCR and SA($\frac{1}{2}$), respectively. For the teapot experiment we can see that H-GnCR finds an ordering that reflects the smooth variation across the patterns, while spectral fails to do so. For MSRC2, it is noticeable that images with similar content are frequently placed close to each other along the linear ordering, i.e., categories of trees, animals, cars, planes, faces, flowers, books, etc. Although a perfect reconstruction of the original order cannot be achieved in this case, both methods seem to do a good job seriating images with animals, trees and books, while the SA($\frac{1}{2}$) method performs better in seriating images with faces.

We additionally evaluate the ability of the proposed methods to find a solution that is close to the true underlying ordering of the rotating teapots. Table 13 presents for all methods the corresponding absolute Kendall's tau (Critchlow, 2012) which measures the rank correlation between two orderings, and PPC (Goulermas et al, 2016) which measures the agreement in terms of positional proximities. We can see that H-GnCR and SA($\frac{1}{2}$) achieve the optimum scores. Moreover, all three proposed methods maintain very low proximity scores across the different objectives.

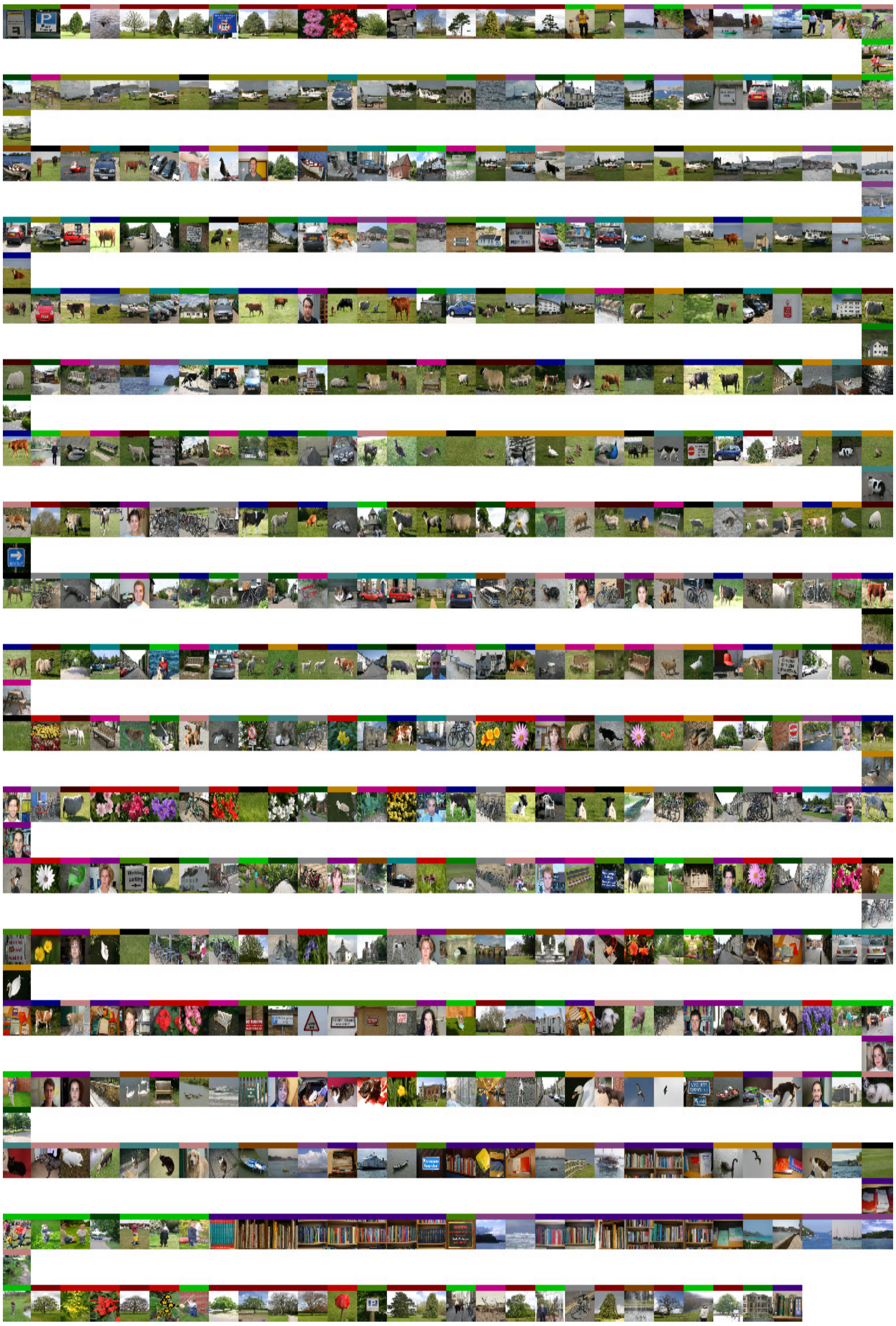


Fig. 8: Seriated images from the MSRC2 dataset with C-GnCR. Colorbars at the top of each image correspond to different categories.

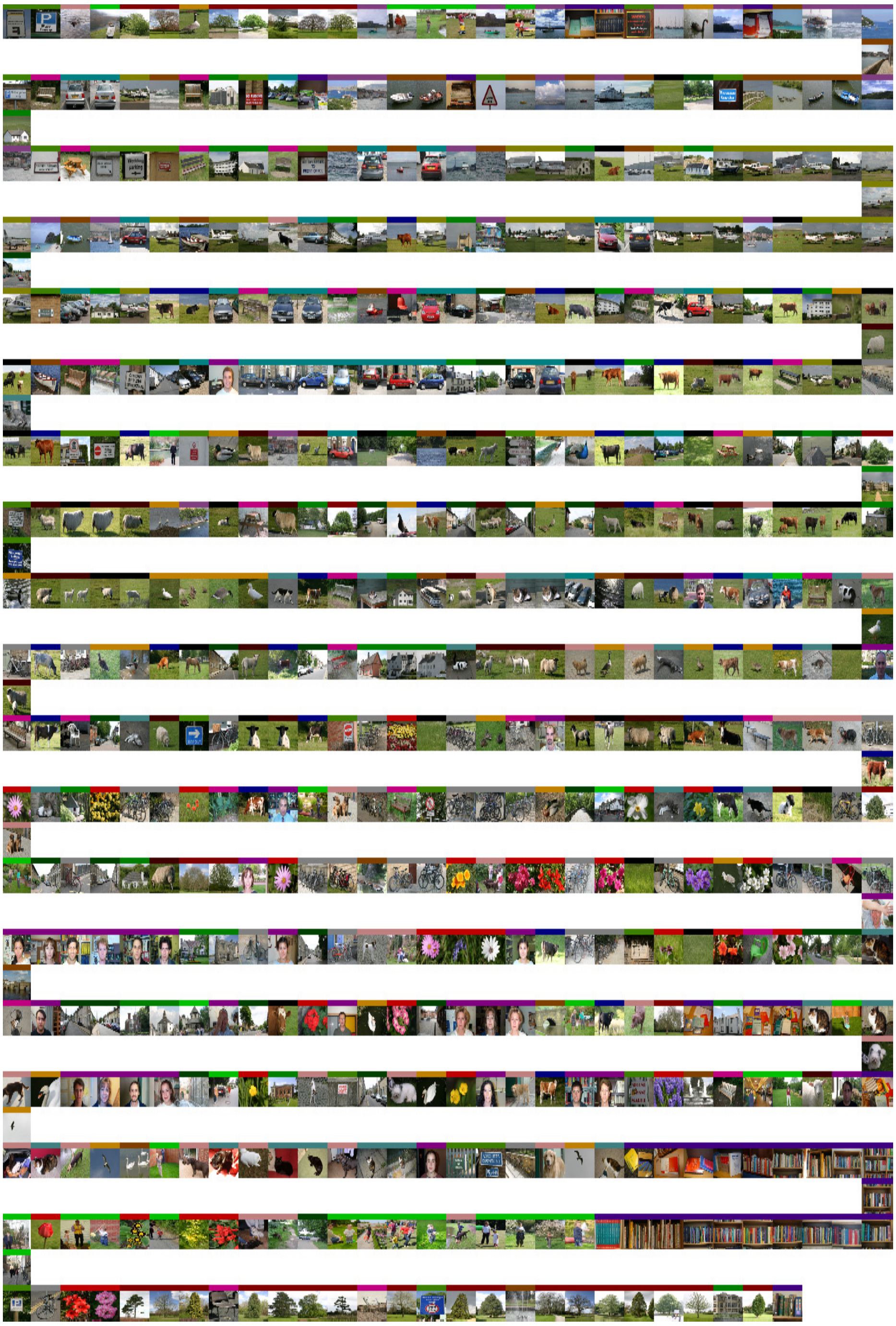


Fig. 9: Seriated images from the MSRC2 dataset with SA($\frac{1}{2}$). Colorbars at the top of each image correspond to different categories.

5 Discussion and Conclusions

In this work we have introduced a new set of algorithms for a continuous relaxation of various versions of the p-SUM problem, based on a graduated non-convexity procedure with a first-order optimization method that is performed directly in terms of a permutation vector. To the best of our knowledge, it is the first time continuation-based algorithms are used for approximating a wide range of instances of the p-SUM. A clear advantage of vector gradient-based search when solving large problems is that they are very efficient and naturally scalable.

The experimental results from the previous sections contain some interesting observations regarding the usefulness of the proposed methods for the problem of seriation. In the first set of experiments we examined the utility of the three proposed algorithms in a set of real and artificial datasets. Results show that all proposed algorithms maintain a good performance in a wide range of datasets. SA seems to be the main competitor in this experimental setup, but this is a much slower method (running times are usually greater than 1,000 seconds for problem sizes over $n = 500$). It is also notable that the two convex relaxation approaches (vRCR and vRCR$_2$) do not outperform the two spectral methods when no auxiliary information is present. Similar performance behavior is observed for the WRE measure as well. Moreover, we verified the consistency of the proposed methods with the aid of pre-Robinsonian datasets, where results show that the proposed methods effectively solve the noiseless seriation problem and perform closely to the spectral method. This further supports the benefit of graduated non-convexity as a method to track solutions close to the global optimum.

To explore the suitability of the methods for solving different p-SUM problems, in Section 4.2 we used a synthetic dataset with class label information that enabled us to calculate a local ordering measure and compare algorithms that solve general p-SUM instances. Results show that as we reduce the value of p, the seriation results are more localized. The proposed methods for the 1-SUM and the $\frac{1}{2}$-SUM perform slightly worse in terms of δ_{count}, due to the fact that they rely on smooth approximations of the objective functions. H-GnCR outperforms the FAQ method in terms of 2-SUM$_{sup}$. This can be explained since one single misplacement of objects that are very far apart, could result into a poor seriation quality when assessed globally. Additional experiments on the Facebook dataset demonstrate the effects of solving the p-SUM for $p < 2$. Results in terms of Hamiltonian path show that as we reduce p, more localized orderings are obtained. It can also be seen that the proposed method for the 2-SUM outperforms FAQ and SA, but this is not the case for the ones designed for the 1-SUM and $\frac{1}{2}$-SUM, again due to their approximated objective functions. The scalability of the proposed methods was tested on four large scale sparse matrices in the context of the envelope reduction problem. For this reason, we compared with the two spectral methods which perform well on such problems. Experiments reveal that all three proposed methods are very scalable. In terms of envelope reduction quality GnCR and H-GnCR achieve the best envelope size values, a fact that further supports their close connection to the envelope reduction problem. C-GnCR is slightly worse, but maintains a performance very close to the best in all datasets. For the bandwidth measure, each of the proposed methods achieves best for one of the first three datasets, while for the CAN dataset Spectral_B outperforms them. Results therefore suggest that the proposed methods are suitable for envelope reduction. Finally, the proposed methods were applied to image seriation. With regard to the Teapots dataset, results show that all proposed methods show good performance, with H-GnCR performing best. Additionally, all proposed methods appear to be able to find solutions that are very close to the true underlying ordering. For the MSRC2 dataset we see that C-GnCR performs well in terms of keeping close images of the same category, although it does not outperform $\text{SA}(\frac{1}{2})$ which scores best. In general, the problem of image seriation using extracted features is a challenging one and is highly dependent on the type and quality of the features, such as the SIFT descriptors.

Overall, the results demonstrate the practical benefit of solving the p-SUM for different values of p. The proposed algorithms show a competitive performance and strong scalability to problem sizes unattainable by other methods, a fact that makes them suitable for highlighting patterns of global or local similarities on data in various real-world applications, such as bioinformatics, data mining, image analysis, data visualization, etc.

Acknowledgements The work was supported by a studentship from the School of Electrical Engineering, Electronics and Computer Science, at the University of Liverpool, U.K. The authors would like to thank the anonymous reviewers for their very constructive and helpful comments.

References

Alter O, Brown PO, Botstein D (2000) Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Sciences 97(18):10,101–10,106

Anstreicher KM (2003) Recent advances in the solution of quadratic assignment problems. Mathematical Programming 97(1-2):27–42

Atkins JE, Boman EG, Hendrickson B (1998) A spectral algorithm for seriation and the consecutive ones problem. SIAM Journal on Computing 28:297–310

Barnard ST, Pothen A, Simon HD (1993) A spectral algorithm for envelope reduction of sparse matrices. In: Supercomputing '93, ACM/IEEE, pp 493–502

Basak J (2008) A least square kernel machine with box constraints. In: 19th International Conference on Pattern Recognition, pp 1–4

Baudat G, Anouar F (2001) Kernel-based methods and function approximation. In: Proceedings of the International Joint Conference on Neural Networks, 2001, vol 2, pp 1244–1249

Bazaraa MS, Sherali HD (1982) On the use of exact and heuristic cutting plane methods for the quadratic assignment problem. Journal of the Operational Research Society 33(11):991–1003

Bertsekas DP (1995) Nonlinear Programming. Athena Scientific

Blake A (1983) The least-disturbance principle and weak constraints. Pattern Recognition Letters 1(5):393 – 399

Brusco MJ, Stahl S (2000) Using quadratic assignment methods to generate initial permutations for least-squares unidimensional scaling of symmetric proximity matrices. Journal of Classification 17(2):197–223

Brusco MJ, Stahl S (2001) Compact integer-programming models for extracting subsets of stimuli from confusion matrices. Psychometrika 66(3):405–419

Bubeck S (2015) Convex optimization: Algorithms and complexity. Foundations and Trends in Machine Learning 8(3-4):231–357

Burkard R, Çela E (1999) Linear assignment problems and extensions. In: Du DZ, Pardalos PM (eds) Handbook of Combinatorial Optimization, Kluwer, pp 75–149

Burkard RE, Çela E, Pardalos PM, Pitsoulis LS (1999) The quadratic assignment problem. In: Handbook of Combinatorial Optimization: Volume 1–3, Springer US, Boston, MA, pp 1713–1809

Çela E (2013) The quadratic assignment problem: theory and algorithms, vol 1. Springer Science & Business Media

Christofides N, Benavent E (1989) An exact algorithm for the quadratic assignment problem on a tree. Operations Research 37(5):760–768

Critchlow DE (2012) Metric methods for analyzing partially ranked data, vol 34. Springer Science & Business Media

Csurka G, Dance C, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, Prague, vol 1, pp 1–22

Davis TA, Hu Y (2011) The University of Florida Sparse Matrix Collection. ACM Transactions on Mathematical Software 38:1:1 – 1:25, URL `http://www.cise.ufl.edu/research/sparse/matrices`

Dheeru D, Karra Taniskidou E (2017) UCI machine learning repository. URL `http://archive.ics.uci.edu/ml`

Ding C, He X (2004) Linearized cluster assignment via spectral ordering. In: Proceedings of the 21st International Conference on Machine Learning, pp 30–37

Earle D, Hurley CB (2015) Advances in dendrogram seriation for application to visualization. Journal of Computational and Graphical Statistics 24(1):1–25

Evangelopoulos X, Brockmeier AJ, Mu T, Goulermas JY (2017) A graduated non-convexity relaxation for large scale seriation. In: Proceedings of SIAM International Conference on Data Mining, 2017

Fiedler M (1973) Algebraic connectivity of graphs. Czechoslovak Mathematical Journal 23(2):298–305

Fogel F, Jenatton R, Bach F, d'Aspremont A (2013) Convex relaxations for permutation problems. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in Neural Information Processing Systems 26, Curran Associates, Inc., pp 1016–1024

Fogel F, Jenatton R, Bach F, d'Aspremont A (2015) Convex relaxations for permutation problems. SIAM Journal on Matrix Analysis and Applications 36(4):1465–1488

Fountoulakis K, Gondzio J (2016) A second-order method for strongly convex ℓ_1-regularization problems. Mathematical Programming 156(1-2):189–219

Frank M, Wolfe P (1956) An algorithm for quadratic programming. Naval Research Logistics Quarterly 3(1-2):95–110

Fulkerson D, Gross O (1965) Incidence matrices and interval graphs. Pacific Journal of Mathematics 15(3):835–855

George A, Liu JW (1981) Computer Solution of Large Sparse Positive Definite Systems. Prentice Hall Professional Technical Reference

George A, Pothen A (1994) An analysis of spectral envelope reduction via quadratic assignment problems. SIAM Journal of Matrix Analysis and Application 18:706–732

Glover F, Laguna M (1997) Tabu Search. Kluwer Academic Publishers, Norwell, MA, USA

Goemans MX (2015) Smallest compact formulation for the permutahedron. Mathematical Programming 153(1):5–11

González-Recio O, Forni S (2011) Genome-wide prediction of discrete traits using bayesian regressions and machine learning. Genetics Selection Evolution 43(1):7

Goulermas JY, Kostopoulos A, Mu T (2016) A new measure for analyzing and fusing sequences of objects. IEEE Transactions on Pattern Analysis and Machine Intelligence

38(5):833–848

Hahsler M (2017) An experimental comparison of seriation methods for one-mode two-way data. European Journal of Operational Research 257(1):133–143

Hahsler M, Hornik K, Buchta C (2008) Getting things in order: an introduction to the R package seriation. Journal of Statistical Software 25(3):1–34

Hardy GH, Littlewood JE, Pólya G (1952) Inequalities. Cambridge University Press

Hartley RI, Zisserman A (2004) Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press

Havens TC, Bezdek JC (2012) An efficient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm. IEEE Transactions on Knowledge and Data Engineering 24(5):813–822

Helmberg C, Rendl F, Mohar B, Poljak S (1995) A spectral approach to bandwidth and separator problems in graphs. Linear and Multilinear Algebra 39(1-2):73–90

Hodson FR (1968) The La Tène cemetery at Münsingen-Rain: catalogue and relative chronology, vol 5. Stämpfli

Huber PJ (1992) Robust estimation of a location parameter. In: Breakthroughs in Statistics: Methodology and Distribution, Springer New York, NY, pp 492–518

Jaggi M (2013) Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In: Proceedings of the 30th International Conference on Machine Learning, PMLR, Atlanta, Georgia, USA, vol 28, pp 427–435

Juvan M, Mohar B (1992) Optimal linear labelings and eigenvalues of graphs. Discrete Applied Mathematics 36(2):153–168

Juvan M, Mohar B (1993) Laplace eigenvalues and bandwidth-type invariants of graphs. Journal of Graph Theory 17(3):393–407

Kendall DG (1971) Abundance matrices and seriation in archaeology. Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 17(2):104–112

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol 4, pp 1942–1948

Lacoste-Julien S (2016) Convergence Rate of Frank-Wolfe for Non-Convex Objectives. ArXiv e-prints 1607.00345

Lacoste-Julien S, Jaggi M (2015) On the global linear convergence of frank-wolfe optimization variants. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, MIT Press, Cambridge, MA, USA, pp 496–504

Laurent M, Seminaroti M (2015) The quadratic assignment problem is easy for Robinsonian matrices with toeplitz structure. Operations Research Letters 43(1):103–109

Leskovec J, Krevl A (2014) SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data

Liiv I (2010) Seriation and matrix reordering methods: An historical overview. Statistical Analysis and Data Mining 3(2):70–91

Lim CH, Wright S (2014) Beyond the Birkhoff polytope: Convex relaxations for vector permutation problems. In: Advances in Neural Information Processing Systems, pp 2168–2176

Lim CH, Wright S (2016a) A box-constrained approach for hard permutation problems. In: Proceedings of the 33rd International Conference on Machine Learning, pp 2454–2463

Lim CH, Wright SJ (2014) Sorting Network Relaxations for Vector Permutation Problems. ArXiv e-prints 1407.6609

Lim CH, Wright SJ (2016b) Efficient bregman projections onto the permutahedron and related polytopes. In: Gretton A, Robert CC (eds) Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, PMLR, Cadiz, Spain, Proceedings of Machine Learning Research, vol 51, pp 1205–1213

Liu ZY, Qiao H (2014) GNCCP: Graduated nonconvexity and concavity procedure. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(6):1258–1267

Loiola EM, de Abreu NMM, Boaventura-Netto PO, Hahn P, Querido T (2007) A survey for the quadratic assignment problem. European Journal of Operational Research 176(2):657–690

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2):91–110

Lyzinski V, Fishkind DE, Fiori M, Vogelstein JT, Priebe CE, Sapiro G (2016) Graph matching: Relax at your own risk. IEEE Transactions on Pattern Analysis Machine Intelligence 38(1):60–73

Mavroeidis D, Bingham E (2010) Enhancing the stability and efficiency of spectral ordering with partial supervision and feature selection. Knowledge and Information Systems 23(2):243–265

McAuley J, Leskovec J (2012) Learning to discover social circles in ego networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, Curran Associates Inc., USA, pp 539–547

Mühlenbein H (1989) Parallel genetic algorithms population genetics and combinatorial optimization. In: Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 416–421

Quadrianto N, Smola AJ, Song L, Tuytelaars T (2010) Kernelized sorting. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(10):1809–1821

Rangarajan A, Chellappa R (1990) Generalized graduated nonconvexity algorithm for maximum a posteriori image estimation. In: Proceedings of the 10th International Conference on Pattern Recognition, pp 127–133 vol.2

Recanati A, Brüls T, d'Aspremont A (2017) A spectral algorithm for fast de novo layout of uncorrected long nanopore reads. Bioinformatics 33(20):3188–3194

Recanati A, Servant N, Vert JP, d'Aspremont A (2018) Robust Seriation and Applications to Cancer Genomics. ArXiv e-prints 1806.00664

Robinson WS (1951) A method for chronologically ordering archaeological deposits. American Antiquity 16(4):293–301

Tien YJ, Lee YS, Wu HM, Chen CH (2008) Methods for simultaneously identifying coherent local clusters with smooth global patterns in gene expression profiles. BMC Bioinformatics 9(1):155

Tsafrir D, Tsafrir I, Ein-Dor L, Zuk O, Notterman DA, Domany E (2005) Sorting points into neighborhoods (SPIN): data analysis and visualization by ordering distance matrices. Bioinformatics 21(10):2301–2308

Tuytelaars T (2010) Dense interest points. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 2281–2288

Vogelstein JT, Conroy JM, Lyzinski V, Podrazik LJ, Kratzer SG, Harley ET, Fishkind DE, Vogelstein RJ, Priebe CE (2015) Fast approximate quadratic programming for graph matching. PLOS ONE 10(4):1–17

Weinberger KQ, Saul LK (2004) Unsupervised learning of image manifolds by semidefinite programming. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, pp 988–995

Xia Y (2010) An efficient continuation method for quadratic assignment problems. Computers & Operations Research 37(6):1027–1032

Yang XS (2008) Nature-Inspired Metaheuristic Algorithms. Luniver Press

Zaslavskiy M, Bach F, Vert JP (2009) A path following algorithm for the graph matching problem. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(12):2227–2242