Matrix-free interior point methods for point set matching problems∗

Janith Wijesinghe† and Pengwen Chen‡

1,2Applied Mathematics, National Chung Hsing University, Taichung, 402, Taiwan.

September 27, 2022

Abstract

Point sets matching problems can be handled by optimal transport. The mechanism behind it is that optimal transport recovers the point-to-point correspondence associated with the least curl deformation. Optimal transport is a special form of linear programming with dense constraints. Linear programming can be handled by interior point methods, provided that the involved ill-conditioned Hessians can be computed accurately. In interior point methods, the Newton step is computed from the pseudo-inversion of a Schur complement matrix of the Hessian matrix. When the optimal solution is a permutation matrix, the Schur complement matrices along the central path actually undergo a rank-reduction process. We propose the application of matrix balancing to eliminate the solution deviation of the central path from the feasible set. During the decade, matrix balancing has been employed to compute optimal transport under entropy regularization approaches. In this work, we examine the application of sparse support constraints to matrix-balancing based interior point methods, in which the sparse set fulfilling total support is iteratively updated to truncate the domain of the transport plan. Total support condition guarantees the existence of matrix balancing. Experiments justify the effectiveness of the proposed algorithm on point-sets with large cardinality.

Keywords: Optimal mass transport, linear programming, interior point methods, matrix balancing, negative entropy, point-set matching

1 Introduction

Registration aims to match two or more sets of image data altered by geometric transforms, taken at different times, or from different sensors. Point set representing image data is commonly employed in computer vision. The associated point-set matching (registration) problem is to establish a consistent point-to-point correspondence between two point sets and to estimate the spatial alignment transformation. The quality of correspondence plays a crucial role in estimating followup transformations in registration. The iterative closest point (ICP) algorithm is one classic, popular approach to feature-based image registration problems, because of its simplicity [BM92]. For correspondence correctness, the ICP algorithm requires sufficient overlap between the point sets. Its vulnerability in performance also includes the proneness to outliers. To alleviate these difficulties, researchers describe the correspondence by a permutation matrix, which minimizes some “distance” of the point-sets, typically consisting of one regularization term for transformations and one assignment term for correspondence. For instance, Chui and Rangarjan proposed a robust point matching method (RPM), which estimates non-rigid transformation and correspondence simultaneously, where the point-to-point correspondence is enforced by Sinkhorn matrix balancing [CR00]. This can be viewed as one early application of optimal transport in registration. Comprehensive surveys of traditional registration methods can be found in [MV98] and [ZF03].

∗Submitted to the editors DATE. Funding: The research of the author is supported by grant 110-2115-M-005-007- MY3 from the Ministry of Science and Technology, Taiwan.
†Email address: janithuo@gmail.com
‡Corresponding author. Email address: pengwen@nchu.edu.tw.
Two unlabeled point sets can be regarded as two histograms, whose distance can be fast evaluated by various information divergences, for instance, Hellinger distances, Kullback-Leibler divergences and Jensen Shannon divergences. From a perspective of correspondence retrieval, a natural choice is Wasserstein distance (also known as the earth mover’s distance [RTG00b]). Instead of performing a pointwise comparison of the probability distributions, Wasserstein distance quantifies the minimal cost of moving the probability mass from one distribution to the other distribution. In the 1780’s, Monge described a problem of transporting a pile of soil with the least amount of work. In the 1940’s, Kantorovich [Kan42] employed a dual variation principle to convert the original nonlinear problem into a linear programming problem and to study the optimal solutions. A survey of theoretical works on this problem can be found in [Eva97] or [Vil03]. Optimal transport has many applications in image retrieval, image registration, image morphing, shape matching and machine learning, see [WPR85], [Kai98], [RTG00a], [ZYHT07], [RDG09], [PC19], [SS15], [MSKL09], [CLC13], [KPT+17] and [CA14].

In the application of point-set registration, we can incorporate optimal transport in feature based methods to estimate the transforms and the correspondence in the existence of outliers, for instance, Hellinger distances based point set matching model (HD) [Che11]. The HD model can be regarded as an approximation of optimal transport, when the kernel scale tends to infinity. With a finite kernel scale, the measure preserving constraint is relaxed to tolerate the existence of outliers.

The effectiveness of matching algorithms generally depends on the hypothesis of geometric transforms. A fundamental question is, for which class of transformations the underlying point correspondence can be reconstructed correctly? Impressively, when the transformation can be expressed as the gradient of some convex function, the underlying correspondence can be recovered correctly by solving the $L^2$ optimal transport problem. The set of candidates includes scalings, translations, positive definite affine transforms and other curl-free maps. This property makes optimal transport models suitable and robust in certain applications. For instance, [CLC13] applies the optimal mass transport model to match lung vessel branch points, which are extracted from two computed tomography (CT) lung images acquired during breath-holds. Although the physical deformation field is rather large and complex, the correspondence reconstruction is surprisingly almost perfect, which verifies the superiority of the optimal transport model.

Despite of the theoretical advantage of optimal transport, optimal transport is limited by its heavy computational requirement in practical implementations. Standard algorithms include the simplex method and the interior point method [Rob12] [LY16] [Gon12]. Briefly, as one member of linear programming, optimal transport can be solved by various linear programming algorithms. Thanks to second-order convergence in each sub-problem, an interior point method can quickly generate accurate solutions from proper matrix-free algorithms. Nowadays the primal-dual interior method is an efficient interior point method in solving linear programming, when the problem size is moderate; see the textbook [Wri97]. For instance, in the community of machine learning, Wasserstein barycenter is one average of multiple discrete probability measures in terms of Wasserstein distance [YLST21]. Interior point methods have been used to produce accurate solutions for Wasserstein barycenter of multiple images [GWXY19]. In considering the flexibility of handling transformation and correspondence simultaneously, we focus on barrier functions based interior point methods. In addition to log-barrier functions, another popular choice is the negative entropy function, which elegantly converts optimal transport to one matrix balancing task. Recently, matrix balancing algorithms become effective tools to produce one approximation of the optimal transport plan [Cui13] [BCC+15] [KR17] [Sch19]. The major numerical tool is the Sinkhorn balancing algorithm [Sin64] [KS67]. To improve the convergence speed of Sinkhorn algorithm, the $\epsilon$-scaling heuristic and kernel truncation are introduced to reduce the number of iterations and the number of variables [Sch19].

### 1.1 Contributions

This paper is concerned with interior point methods in solving point-set matching problems. The main question is whether we can develop a proper central path for discrete optimal transport approximations with small regularization parameters?

The contribution can be summarized as follows. First, we describe matrix-free conjugate gradient methods in estimating Newton steps along central paths of the interior point methods. It is known that interior point methods can produce very accurate and fast solutions to linear programming. However, for discrete optimal transport, empirical results do not always support this widely known statement. High
accurate solutions relies on the careful handling of ill-conditioned Hessians in Newton direction computation. As the central path heads toward an optimal permutation solution, the rank of the associated Schur complement matrices generally reduces to \( n \), the point cardinality. During the rank-reduction process, it is numerically challenging to maintain the accuracy of Newton iterates, since rounding errors in Hessian matrices could easily drive Newton iterates away from the feasible set, which further deteriorates incorrect Schur complement matrices in the sequential sub-problems and deteriorates the computed duality gap. To alleviate this issue, we employ matrix balancing to project iterates back to the feasible set, and employ early termination as stopping criteria before the degeneracy. Accurate solutions can be obtained under two matrix-balancing based interior point methods: IPMB in Alg. 2.3 and SNNE in section 2.4.

Second, as in the kernel truncation method \cite{Sch19}, sparse support constraints can be imposed to reduce the memory requirement of the interior point methods applied in the large-scale problems. It is crucial that the sparse matrix should have total support to guarantee the existence of matrix balancing. With all the ingredients, we propose Alg. 2.9 to handle large-scale point-set matching problems. The convergence is stated in Theorem 2.

Third, we examine a few Newton methods, which compute the optimal scaling vector for matrix balancing. To reveal the connection between these methods, we introduce one convex function and propose a novel modified Newton method. The well-known Knight-Ruiz fixed point algorithm \cite{KR12} can be viewed as the proposed modified Newton method with constant step size. We provide consistency analysis to show the well-definedness of the algorithm, since the Hessian matrix for Newton steps is exactly the Schur complement matrix in IPMB and shall undergo a rank-reduction process, as the iterations head to an optimal solution. In addition, we provide the computation of Newton decrement, which sheds light on the convergence of Knight-Ruiz fixed point algorithm. For SNNE-sparse, the total support condition actually ensures the boundedness of scaling vectors. Numerical experiments demonstrate fast convergence of these Newton methods with proper warm starts, including the \( \epsilon \)-scaling heuristic.

This paper is organized as follows. In section 2, we describe the application of optimal transport in point-set registration. Discrete optimal transport can be solved by interior point methods with assistance from matrix balancing and early termination procedures. In section 3, we describe a few matrix balancing schemes, including Sinkhorn-Knopp balancing and Knight-Ruiz scheme. Matrix balancing can be achieved through minimizing a convex function. In section 4, we present a few numerical simulations, which demonstrate the effectiveness of the proposed algorithms. In appendix, we present the computation of central path used in IPMB and the procedure of early termination used in IPMB and SNNE.

1.2 Notations

In this paper, let \((x,y)\) denote the inner product between \(x, y\) in \(\mathbb{R}^n\). For a vector \(x \in \mathbb{R}^n\) and a scalar \(\epsilon \in \mathbb{R}\), let \(y = (x > \epsilon)\) denote a zero-one vector, i.e., for \(i = 1, \ldots, n\), set \(y_i = 1\) if \(x_i > \epsilon\), and set \(y_i = 0\) otherwise. For simplicity of notation, the functions exp and log are extended to vector spaces \(\mathbb{R}^n\) by componentwise application to all components: \((\exp(x)), (\log x)) = (\exp x_i), (\log x_i) = \log x_i, i = 1, \ldots, n\). Likewise, let \(x^{-1}\) be the vector whose entries are \(x_i^{-1}\). Let the operator \(\odot\) denote entrywise multiplication, e.g., \(x \odot y \in \mathbb{R}^n\) and \((x \odot y)_i = x_i y_i\). Let \(1_n = [1, 1, \ldots, 1]^\top \in \mathbb{R}^n\) be the vector whose entries are all one. Let \([x; y]\) denote the stacked vector \([x^\top, y^\top]^\top\) for any two vectors \(x, y\). The norm \(|| \cdot \||\) represents the 2-norm. Let \(\mathcal{T}\) be the reshape operator \(x \in \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}\), \(T(x) \in \mathbb{R}^{n \times n}\), \(T(x)_{i, j} = x_{m+i}y_j\) for \(i, j \in \{1, \ldots, n\}\). In addition, for the sake of simplicity, \(x_{i,j}\) stands for \((x)_{i,j}\) if no confusion occurs. Let \(\Pi_n\) denote the set of doubly stochastic matrices, i.e., row stochastic and column stochastic \(T(x)1_n = 1_n = T(x)^\top 1_n\) for each \(T(x) \in \Pi_n\). Finally, let \(A^\dagger\) stand for the pseudo inverse of a matrix \(A\).

2 Optimal transport

2.1 Matching point-sets under deformations

We first review the deformation characterization of optimal transport applied on the point-set matching problems in the previous work \cite{CLC13}. The primary focus of the point set matching is the reconstruction of the correspondence between two unlabeled point-sets \(\{z_i\}_{i=1}^n \subset \Omega\) and \(\{y_i\}_{i=1}^n \subset T(\Omega)\), where \(T\) is
some injective and orientation-preserving deformation on a bounded open connected subset $\Omega$ of $\mathbb{R}^3$. The correspondence can be described by a permutation $\tau$ such that $y_i = T(\tau(i))$ and some optimal condition hold for $\tau$. One natural criterion is the minimization problem:

$$
\min_{\tau} \sum_{i=1}^{n} \|y_i - z_{\tau(i)}\|^2.
$$

(1)

This is a discrete combinatorial optimization problem, because $n!$ possibilities must be evaluated. This difficulty can be alleviated, if we consider the relaxed continuous problem,

$$
\min_{X_{i,j}} \sum_{i=1}^{n} \|y_i - z_j\|^2 X_{i,j},
$$

(2)

subject to the unit mass constraint, $\sum_{i=1}^{n} X_{i,j} = 1 = \sum_{j=1}^{n} X_{i,j}$. $X_{i,j} \geq 0$. The problem is known as the $L^2$ Monge-Kantorovich mass transport problem. The relaxed problem described by Eq. (2) is a convex (in fact, linear) minimization problem, which has an optimal permutation matrix (the existence of this is guaranteed by Birkhoff’s theorem) and can be solved by interior point methods [BV04] or primal-dual algorithms [BDM09] (see chapter 4 in [BDM09]).

In the context of (2), the permutation $\tau$ corresponding to the permutation $X$ is optimal, if and only if $\{(\tau(i), y_i)\}_{i=1}^{n}$ is cyclically monotone. Consider a transform $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ between two point sets $\{z_i\}_{i=1}^{n}, \{y_i\}_{i=1}^{n}$ in $\mathbb{R}^d$ with $y_i = T(z_i)$. When a (unknown) transform between these point sets is the gradient of some convex function, then the correspondence can be recovered correctly by solving mass transport problems. The set of transforms includes scalings, translations, and other curl-free maps. Assume that $T$ is twice continuously differentiable and has the Helmholtz decomposition $T(z) = \nabla \phi(z) + \nabla \times \psi(z)$, where $\phi$ is strongly convex and $\nabla \cdot \psi = 0$. Point correspondence can be reconstructed correctly from optimizing transport objectives, if the transform $T$ between point-sets is the gradient of some convex function. In general, for a point-set with finite cardinality $n$ sampled from $\Omega \subset \mathbb{R}^3$, when the curl of the transform is sufficiently small, less than some upper bound $C/n$ [CLC13], then the underlying correspondence coincides with a minimizer $\{X_{i,j}\}_{i,j=1}^{n}$ of Eq. (2). Here $C$ is a positive constant. Empirical studies show the outstanding performance of optimal transport in recovering the correspondence of point-sets under a small curl deformation. In the following section, we shall investigate the application of interior point methods in solving (2).

### 2.2 Discrete optimal transport

To solve (2), introduce a vector $c \in \mathbb{R}^{n^2}$ and its associated (reshaped) matrix $T(c)$ with $T(c)_{i,j} = \|y_i - z_j\|^2$. We can express (2) as the primal problem (transportation): searching for the optimal solution $x \in \mathbb{R}^{n^2}$ in

$$
\min_{\Pi_n} \langle c, x \rangle,
$$

(3)

where $\Pi_n$ is the set

$$
\{\Upsilon(x) : Mx := [\Upsilon(x)_{n+1} \Upsilon(x)^T]_{1,2n} = 1_{2n}, x \geq 0\}.
$$

(4)

The matrix $X = \Upsilon(x)$ represents a coupling matrix $X = [X_{i,j} \geq 0 : i, j = 1, \ldots, n]$, whose entry $X_{i,j}$ describes the amount of mass flowing from bin $i$ toward bin $j$. The problem in (3) is also known as the assignment problem. For each feasible solution $x$, at most $n$ entries can reach the value 1, i.e., $\Upsilon(x)$ is a permutation matrix. By Birkhoff theorem, the extreme points of the set of doubly stochastic matrices are the permutation matrices.

The action of the adjoint operator $M^T$ on a vector $\nu = [\nu^{(1)}; \nu^{(2)}] \in \mathbb{R}^{2n}$ is given by

$$
M^T \nu = \Upsilon^{-1}(\nu^{(1)} 1_{n}^T + 1_n \nu^{(2)}^T).
$$

(5)

Its dual problem to (3) is the maximization problem with respect to a dual variable $\nu \in \mathbb{R}^{2n}$,

$$
\max_{\nu} \{\Upsilon^T 2n \nu : M^T \nu \leq c\}.
$$

(6)
The optimal condition of the primal and dual problem is characterized by the Karush-Kuhn-Tucker (KKT) conditions, i.e., the nonnegativeness of a slack vector in \( s \),

\[
s := c - M^T \nu \geq 0
\]  

holds and \( \mathcal{T}(s)_{i,j} > 0 \) occurs only for those indices \((i, j)\) with \( X_{i,j} = 0 \). The slackness condition actually implies zero duality gap,

\[
\langle c, x \rangle - \langle \nu, \mathbb{1}_{2n} \rangle = \langle c, x \rangle - \langle M^T \nu, x \rangle = \langle s, x \rangle = 0.
\]  

### 2.3 Interior point methods

Here we quickly illustrate the application of interior point methods to \( \mathcal{T} \). More details can be found in textbooks \([BV04]\) and \([LY16]\). In the paper, we consider both log-barrier functions and negative entropy functions as barrier functions in interior point methods. For the sake of presentation, we start with log-barrier functions and discuss the choice of negative entropy in section 2.4.

To reach one optimal solution of \( \mathcal{T} \), path-following methods \([FM68]\) solve the associated logarithmic barrier function with larger and larger values of \( t \). We compute the central point \( x \) starting from the previously computed central point \( x(t_j) \). More details can be found in \([LY16]\). The Newton direction from solving the normal equation in \( \mathcal{T} \),

\[
M(c - M^T \nu)^{-1} - t \mathbb{1} \mathbb{1}^T = 0,
\]

subject to \( M^T \nu = c \).

#### 2.3.1 Matrix-free conjugate gradient methods for central path

For completeness, we illustrate the matrix-free computation of \( x(t) \). The arguments are standard, which can be found, for instance, \([LY16]\). The Newton direction from solving the normal equation in \( \mathcal{T} \),

\[
M(c - M^T \nu)^{-1} - t \mathbb{1}_n = 0,
\]

which provides a measure of closeness to optimality. The optimal solution of \( \mathcal{T} \) can be obtained from a limit of \( x(t) \) as \( t \to \infty \).

\[
x_{k+1} = x_k + \alpha d_k \in \mathcal{T}^{-1}(\Pi_n), \quad \nu_{k+1} = \nu_k + \alpha y_k,
\]
where \( z_k := (d_k, y_k) \) satisfies the linearization of (10)

\[
\nabla f(x_k + d_k, \nu_k + y_k) \approx \nabla f(x_k, \nu_k) + \langle \nabla f^2(x_k, \nu_k), z_k \rangle = 0.
\]

(16)

Introduce the residual vector,

\[
r_k = -(c - (x_k t)^{-1} - M^T \nu_k).
\]

(17)

Since \( Mx_k = \mathbb{I}_{2n} \) holds, (16) gives

\[
\nabla^2 f(x_k, \nu_k) z_k = \begin{pmatrix} t^{-1} \text{diag}(x_k)^{-2}, -M^T \\ -M, 0 \end{pmatrix} \begin{pmatrix} d_k \\ y_k \end{pmatrix} = -\nabla f(x_k, \nu_k) = \begin{pmatrix} r_k \\ 0 \end{pmatrix}.
\]

(18)

In solving (18), we shall avoid forming those big matrices \( M, \text{diag}(x_k)^{-2} \) due to heavy memory loads \( O(n^3), O(n^4) \). The first part of (18) implies

\[
t^{-1} d_k = \text{diag}(x_k^2)(M^T y_k + r_k).
\]

(19)

Together with the second part of (18), we have the normal equation for \( y_k \),

\[
M \text{diag}(x_k^2)(M^T y_k + r_k) = 0, \quad \text{i.e., } y_k = -(M \text{diag}(x_k^2)M^T)^T(M \text{diag}(x_k^2)r_k),
\]

(20)

which indicates that \( y_k \) is the minimizer of

\[
\min_{y_k} \| \text{diag}(x_k)M^T y_k + x_k \circ r_k \|^2.
\]

(21)

Note that the matrix \( M \text{diag}(x_k^2)M^T \) can be regarded as the Schur complement of the first block in the Hessian matrix in (15), after ignoring the scaling factor \( t \). More discussions on the computation of \( d_k \) and \( \nu_k \) are given in the appendix. We can employ Krylov subspace methods, e.g., matrix-free conjugate gradient methods to solve \( y_k \) from (20) and then compute \( d_k \) from (19). Here, we demonstrate the matrix-vector product in the conjugate gradient method in solving \( y_k \). With \( y_k := [y^{(1)}; y^{(2)}] \) and

\[
\begin{pmatrix}
\text{diag}(\mathbb{T}(x_k^2)\mathbb{1}_n), \\
\mathbb{T}(x_k^2)^T, \\
\text{diag}(\mathbb{T}(x_k^2)^T\mathbb{1}_n)
\end{pmatrix},
\]

\( \hat{M}_k := \begin{pmatrix}
\text{diag}(\mathbb{T}(x_k^2)\mathbb{1}_n), \\
\mathbb{T}(x_k^2)^T, \\
\text{diag}(\mathbb{T}(x_k^2)^T\mathbb{1}_n)
\end{pmatrix}, \)

(22)

we implement the matrix-vector product in the conjugate gradient method,

\[
M \text{diag}(x_k^2)M^T y_k = \hat{M}_k y_k = \begin{pmatrix}
y^{(1)} \odot (\mathbb{T}(x_k^2)\mathbb{1}_n) + \mathbb{T}(x_k^2)y^{(2)} \\
y^{(2)} \odot (\mathbb{T}(x_k^2)^T\mathbb{1}_n) + \mathbb{T}(x_k^2)^Ty^{(1)}
\end{pmatrix}.
\]

(23)

To further enhance the convergence speed, we can adopt the preconditioner for the conjugate gradient method, i.e., the application of the diagonal matrix \( S \) on (20),

\[
S = \text{diag}(\hat{M}_k)^{1/2} = \text{diag}([\mathbb{T}(x_k^2)\mathbb{1}_n; \mathbb{T}(x_k^2)^T\mathbb{1}_n]^{1/2}),
\]

(24)

i.e., we solve \( z_k := S y_k \) from the preconditioned system

\[
(S^{-1} \hat{M}_k S^{-1}) z_k = S^{-1} M \text{diag}(x_k^2)(c - (x_k t)^{-1} - M^T \nu_k), \quad \text{and then calculate } y_k = S^{-1} z_k.
\]

(25)

**Remark 2.2** (Rank reduction). In the appendix, we show that each \( x^{(t)} \) is computed based on the Newton direction \( d_k \), whose calculation relies on the application of a projection \( P_k \). The calculation could be inaccurate, if the involved Schur complement \( M \text{diag}(x_k^2)M^T \in \mathbb{R}^{2n \times 2n} \) has rank deficiency due to the limitation of finite precision. More precisely, since the null space of \( M^T \) has dimension 1, the rank of \( M \text{diag}(x_k^2)M^T \) is \( 2n - 1 \) for \( x_k \) with all entries away from 0 (See the appendix). On the other hand, when the optimal solution \( \mathbb{T}(x) \) of (2) is a permutation matrix, \( M \text{diag}(x^2)M^T \), which can be expressed the sum of \( n \) rank one matrices, has rank only \( n \). Hence, as \( x^{(t)} \) tends to \( x \), many entries in \( (x^{(t)})^2 \) (though nonzero) will be rounded to zero in the matrix-vector-product calculation. The inaccuracy is always inevitable for \( t \) sufficiently large, in particular, entries of \( x_k \) tending to zero in a non-uniform speed. In section 4.1, we provide a few simulations, where due to accumulated rounding errors from
rank-deficient Schur complement matrices, the approximation $\mathcal{T}(x^{(t)})$ can be driven away from $\Pi_n$ for large $t$.

To alleviate the rank deficiency issue, we directly employ matrix balancing to obtain one projection of $\mathcal{T}(x^{(t)})$ on $\Pi_n$. The details of matrix balancing are given in section 3. In addition, we shall describe two early termination conditions in Prop. A.5 or Prop. A.6 which can employed to produce an optimal solution.

**Algorithm 2.3** (Interior point method with matrix balancing (IPMB)). Given a starting point $x_{ini}$ in $\Pi_n$ associated with initial $t = t^{(0)} > 0$. Fix $\mu > 1$, and duality gap tolerance $\epsilon > 0$. Let $\epsilon_{MB} > 0$ be some matrix balancing tolerance. Repeat the following steps for $j = 0, 1, 2, \ldots, j_{max}$.

1. **Centering step.** For each $t$, use Newton’s method in Alg. 2.4 to compute $x^{(t)}$ by minimizing (9), starting at $x_{ini}$.

2. If $\|Mx^{(t)} - 1_{2n}\| > \epsilon_{MB}$, perform matrix balancing on $\mathcal{T}(x^{(t)})$ and let $x_{ini}$ be the balanced vector of $x^{(t)}$.

3. **Stopping criterion.** Quit, if (i) the condition $n^2/t < \epsilon$ or (ii) early termination conditions are met for $(x^{(t)}, \nu^{(t)})$. If (ii) is met, report the corresponding KKT point as an optimal solution.

4. Increase $t$ by a factor $\mu$, i.e., $t = t^{(j+1)} := \mu t^{(j)}$.

Introduce a slack variable $s$ in (27). Newton’s method in Alg. 2.4 produces $x^{(t)}$ together with a dual variable $\nu$ to meet (12), i.e., $x^{(t)} \otimes s = t^{-1} 1_{2n}$. The quality of the approximation in the centering step can be evaluated by two parameters $\gamma' \in (0, 1)$ and $\gamma'' > 1$, i.e., (29). Hence, as $t \to \infty$, the duality gap

$$
\epsilon^\top x - \frac{1}{2} s^\top x \leq \frac{\gamma'' n^2}{t}
$$

(26) tends to 0. However, empirically $s^\top x$ cannot reduce to any small amount due to ill-conditioned Hessians in computing Newton direction, which shall be demonstrated in numerical experiments. The implementation of $d_k$ and $\nu_k$ in Newton’s method shall be discussed in section 2.3.1 and appendix A.1. One common stopping criterion called Newton decrement in (28) shall be computed in Remark A.3. The convergence analysis of Newton’s method can be found in section 9.5.3 in [BV04].

**Algorithm 2.4.** [Newton’s method, inner loop] Let tolerance $\epsilon > 0$. Choose some scalars $\gamma' \in (0, 1)$ and $\gamma'' \in (1, \infty)$. Start with a starting point $x = x_0 \in \mathbb{R}^n$ in $\Pi_n$.

Repeat $k = 0, 1, 2, \ldots, k_{max}$.

1. Use matrix-free preconditioned conjugate gradient methods to solve the Newton step $d_k$.

2. **Line search.** Choose step size $\alpha$ to minimize the objective in (9) by backtracking line search. Update $x_{k+1} := x_k + \alpha d_k$.

3. **Stopping criterion.** Estimate the multiplier $\nu_{k+1}$. Let

$$
s_{k+1} := \epsilon - M^\top \nu_{k+1}.
$$

(27) Quit and output $(x_{k+1}, \nu_{k+1})$, if the squared Newton decrement

$$
t \|c \otimes x_{k+1} - t^{-1} \otimes 1_{2n} - \text{diag}(x_{k+1}) M^\top \nu_{k+1} \|^2
$$

(28) is less than $\epsilon$ or the entrywise bounds hold,

$$
\gamma' t^{-1} \leq x_{k+1} \otimes s_{k+1} \leq \gamma'' t^{-1}.
$$

(29)

### 2.3.2 Interior point methods with total support constraints

Even though an optimal solution $x$ could be sparse, interior point methods in solving large-scale problems require memory storage $O(n^2)$ in $x$, which could be prohibited in large-scale point-sets. One effective method handling large data sets is to impose (and dynamically to update) the **sparse support constraint** on $x$, which naturally reduces the storage requirement. Actually, introducing these constraints
to remove those inactive components could improve the quality of solutions $x^{(t)}$. The idea of sparse support sets has been introduced in [Sch19], where the support sets are selected according to the rule depending on some threshold $\epsilon$ to form approximate problems with truncated sparse kernels.

Let $supp(x)$ be the index set of all the positive entries in $x$. We say that the index set $\Sigma$ is one support of $X = T(x)$, if $\Sigma$ consists of all indices of nonzero entries in $X$, i.e., $X_{i,j} = 0$ holds for all $(i, j) \notin \Sigma$. View point-sets $\{y_1\}_{i=1}^{n}$ and $\{z_j\}_{j=1}^{m}$ as vertices of a graph. From graph theory, the set $\Sigma$ can also be regarded as one description of the edge set of a bipartite graph. We say that $X$ is a solution to optimal transport with respect to $\Sigma$ or the bipartite graph associated with $\Sigma$, if $\Sigma$ is a support of $X$ and $X$ is one optimal solution, i.e.,

$$
\min_{X \in \mathbb{R}^{n \times m}} \{ F(X, \Sigma) := \langle T(c), X \rangle : supp(X) \subset \Sigma \}. \tag{30}
$$

Clearly, if a solution $X$ in (30) also satisfies $supp(X) \subset \Sigma' \subset \Sigma$, then $X$ is also a solution to optimal transport with respect to $\Sigma'$ as well. On the contrary, a solution $X$ with respect to $\Sigma$ is not automatically a solution to optimal transport with respect to a superset $\Sigma'$ of $\Sigma$.

To reach one optimal transport approximation, we generate a sequence of supports

$$
\{\Sigma_0, \ldots, \Sigma_{\xi}, \ldots\} \tag{31}
$$

in Alg. 2.7 so that for each $\xi \in \{0, 1, 2, \ldots\}$, $X_{\xi}$ is an optimal transport solution with respect to the support $\Sigma_{\xi}$. Motivated by KKT conditions in (7), we update one sparse support $\Sigma_{\xi+1}$ by the multiplier vector $\nu_\xi = [\nu^{(1)}, \nu^{(2)}]$ associated with $X_{\xi}$, so that $\Sigma_{\xi+1}$ satisfies the total support condition (see Definition 1) and contains the index set,

$$
\{(i, j) : s_{i,j} \leq \epsilon \} \subset \Sigma_{\xi+1}, \tag{32}
$$

where $\epsilon$ is some positive parameter to ensure the sparsity of the support and

$$
s_{i,j} := c_{i,j} - (\nu^{(1)}(i) + \nu^{(2)}(j)). \tag{33}
$$

**Definition 1.** Let $X$ be an $n \times n$ matrix and $\sigma$ be a permutation of $\{1, 2, \ldots, n\}$. Then the sequence $X_{1,\sigma(1)}, X_{2,\sigma(2)}, \ldots, X_{n,\sigma(n)}$ is a diagonal of $X$ (corresponding to $\sigma$). Then a nonnegative square matrix $X$ is said to have support if $X$ contains one positive diagonal. Also, $X$ has total support if $X \neq 0$ and if every positive entry of $X$ lies on a positive diagonal [KS67]. We say that an index set $\Sigma$ satisfies total support condition, if the matrix $1_{\Sigma}$ has total support.

When $1_{\Sigma}$ has no support, then $1_{\Sigma}$ cannot be scaled to a doubly stochastic matrix. Actually, by Birkhoff theorem, any doubly stochastic matrix is convex combination of permutation matrices. Since the support of one nonnegative matrix remains invariant under the product of positive diagonal matrices, having total support is one necessary condition for matrix balancing. Hence, we have to assume that the sparse support set $\Sigma$ in (30) satisfies the total support condition. Indeed, the following states that total support is the crucial condition to ensure the existence of a doubly stochastic matrix from a sparse nonnegative matrix $X$.

**Theorem 1.** [KS67] Let $X$ be a nonnegative squared matrix. A necessary and sufficient condition that $B = \text{diag}(y)X\text{diag}(z)$ is double stochastic for two positive vectors $y, z$ is that $X$ has total support.

For instance, let $X = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & 4 \end{pmatrix}$. Since the entry 2 is not contained in a positive diagonal, $X$ cannot be scaled to a doubly stochastic matrix. However, when $X = \begin{pmatrix} 1 & .05 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & 4 \end{pmatrix}$, the entry 2 is contained in the positive diagonal $[0.05, 2, 4]$ and thus the matrix $X$ can be balanced. On the other hand,
let \( X = \begin{pmatrix} 1 & \epsilon \\ 1 & 1 \end{pmatrix} \) and \( t = \epsilon^{-1/2} \) with \( \epsilon > 0 \). Even though \( X \) can be scaled to a doubly stochastic matrix

\[
\text{diag}([1, t]) \begin{pmatrix} 1 & \epsilon \\ 1 & 1 \end{pmatrix} \text{diag}((1 + t)^{-1}[1, t]) = \begin{pmatrix} t & \frac{t^2}{1+t} \\ \frac{t^2}{1+t} & \frac{t}{1+t} \end{pmatrix},
\]

the relative magnitude of entries of the scaling vectors tend to \( \infty \) as \( \epsilon \to 0 \).

Let \( \Sigma'' \) denote the set,

\[
\{(i, j) : c_{i,j} = (\nu^{(1)}(i) + \nu^{(2)}(j)) \leq \epsilon \}. \tag{35}
\]

The following provides one simple construction of a total support set \( \Sigma \) containing the prescribed index set \( \Sigma'' \).

**Proposition 2.5.** Let \( \Sigma'' \) be some prescribed index set. Let \( \sigma \) be a permutation of \( \{1, 2, \ldots, n\} \) and let \( \Sigma' := \{(i, \sigma(i)) : i = 1, \ldots, n\} \). Then the union set

\[
\Sigma := \Sigma' \cup \Sigma'' \cup \{(\sigma^{-1}(j), \sigma(i)) : (i, j) \in \Sigma''\}. \tag{36}
\]

has total support.

**Proof.** We shall point out one diagonal in \( \Sigma' \) for each \((i, j) \in \Sigma''\). Since \( \sigma \) is a permutation, then \( \{(k, \sigma(k)) : k = 1, 2, 3, \ldots, n\} \) is one diagonal. Express the diagonal sequence as \( \{(i, \sigma(i)), (\sigma^{-1}(j), j), \Sigma'\} \), i.e., \( \Sigma' \) is the set consisting the remaining \( n-2 \) indices. Note that \( \Sigma' \) does not consist of any entries in row- \( i \), row-\( \sigma^{-1}(j) \), column- \( j \) and column- \( \sigma(i) \). By definition of \( \Sigma' \), we have a diagonal \( \{(i, j), (\sigma^{-1}(j), \sigma(i)), \Sigma'\} \) for \((i, j) \).

**Remark 2.6** (The choice of \( \sigma \)). Here we provide one characterization of the union set in \( \Sigma'' \). Let \( Q \) be the permutation \( I_{\Sigma'} \), i.e., \( Q_{i, \sigma(i)} = 1 \) for all \( i = 1, \ldots, n \). Let \( \Sigma'' = \{((\sigma^{-1}(j), \sigma(i)) : (i, j) \in \Sigma''\} \), which can be described by matrix products,

\[
\mathbb{1}_{\Sigma''} = Q \mathbb{1}_{\Sigma''} Q. \tag{37}
\]

Empirically, to make matrix-balancing of \( X \) in \( \Sigma'' \) effortless, the permutation \( \sigma \) should be chosen dynamically, so that the corresponding entries \( \{X_{i, \sigma(i)} : i = 1, \ldots, n\} \) are as large as possible, away from zero. Here, for simplicity, we consider the fixed choice: let \( \sigma \) to be the identity and then \( \Sigma \) in \( \Sigma'' \) is the index set corresponding to the positive entries of \( I + \mathbb{1}_{\Sigma''} + \mathbb{1}_{\Sigma''} \).

In summary, we can employ the following framework to estimate one approximation of \( \Sigma'' \). The following theorem provides the convergence property.

**Algorithm 2.7** (Interior point methods with support constraints). *Input: the assignment matrix \( c \) and a tolerance parameter \( \epsilon > 0 \).

- **Initialization:** Generate \( \Sigma_0 \) and some nonnegative matrix \( X_{\text{ini}} \) with total support and \( \text{supp}(X_{\text{ini}}) \subset \Sigma_0 \).
- **For** \( \xi = 0, 1, 2, 3, \ldots, \xi_{\text{max}} \), **repeat** the following steps.
  - Solve one optimal matrix \( X_{\xi} \) together with multiplier vector \( \nu_{\xi} \) from \( \Sigma_\xi \) with respect to the support constraint \( \Sigma_\xi \),

\[
\min_{X} \{\langle T(c), X \rangle : \text{supp}(X) \subset \Sigma_\xi, M T^{-1}(X) = \mathbb{1}_{2n}\}. \tag{38}
\]
  - Use \( \nu_{\xi} \) and \( X_{\xi} \) to form the index set \( \Sigma'' \) in \( \Sigma_{\xi} \) and to update the set \( \Sigma_{\xi+1} \) with total support according to \( \Sigma'' \).

**Theorem 2.** Fix \( \epsilon > 0 \). Suppose \( X_{\xi} \) is an optimal solution of \( \Sigma_\xi \) subject to a support set \( \Sigma_\xi \) for each \( \xi = 0, 1, 2, 3, \ldots \), where the index set \( \Sigma_{\xi+1} \) has total support and contains the indices in \( \Sigma_\xi \). Then \( X_{\xi} \) converges to an optimal solution of \( \Sigma'' \).
Proof. Since $\Sigma_\xi$ has total support, then the feasible set is nonempty thanks to Theorem [1]. Hence, an optimal solution $X_\xi$ exists for (38). The optimal condition is that for some vector $\nu_\xi$, introducing

$$s := c - M^T \nu_\xi$$

we have (i) $\mathbb{T}(s)_{i,j} \geq 0$ for all $(i,j) \in \Sigma_\xi$; (ii) $\mathbb{T}(s)_{i,j} > 0$ only occurs for those $X_{i,j} = 0$ with $(i,j) \in \Sigma_\xi$.

From (39) and the definition of $\Sigma_\xi$, we have for $\xi = 1, 2, 3, \ldots$,

$$\mathbb{F}(X_\xi, \Sigma_\xi) = \mathbb{F}(X_\xi, \Sigma_{\xi+1}) \geq \mathbb{F}(X_{\xi+1}, \Sigma_{\xi+1}) = \mathbb{F}(X_{\xi+1}, \Sigma_{\xi+2}) \geq \ldots,$$

i.e., the objective $\mathbb{F}(X_\xi, \Sigma_\xi)$ decreases monotonically. Since the number of permutations is finite, the algorithm terminates at some integer $\xi$. When $\Sigma_{\xi+1} = \Sigma_\xi$ occurs, we have (i) $0 \leq \mathbb{T}(s)_{i,j} \leq \epsilon$ for all $(i,j) \in \Sigma_\xi$; (ii) $s_{i,j} > \epsilon$ holds for all $(i,j) \notin \Sigma_\xi$; (iii) $\mathbb{T}(s)_{i,j} > 0$ only occurs for those $X_{i,j} = 0$. Hence, the KKT condition in (38) holds, which implies that $X_\xi$ is one optimal solution. 

**Remark 2.8.** Fix $t_{\max} > 0$. Introducing log-barrier functions, we can approximate $X_\xi$ in (38) from solving the following sub-problems with $t$ tending to $t_{\max}$,

$$\min_X \{ \mathbb{T}(c, X) - t^{-1}(\mathbb{1}_{\Sigma_\xi}, \log X) : \text{supp}(X) = \Sigma_\xi, M T^{-1}(X) = \mathbb{1}_{2n} \}.$$  

(41)

Since $\Sigma_\xi$ has total support, the feasible set is nonempty. Examine the problem in (41) with some $t \in (0, t_{\max}]$. The optimal condition is

$$t X_\xi = \mathbb{1}_{\Sigma_\xi} \odot \mathbb{T}(c - M^T \nu_\xi)^{-1},$$

(42)

where $\nu_\xi$ is a multiplier vector corresponding to the constraint $M T^{-1}(X_\xi) = \mathbb{1}_{2n}$. Hence, $\mathbb{T}(s)_{i,j} = t^{-1} X_{i,j}^{-1} = (c - M^T \nu_\xi)_{i,j} > 0$ hold for all $(i,j) \in \Sigma_\xi$. Now update $\Sigma_\xi$ to $\Sigma_{\xi+1}$ as in Alg. 2.7. Theoretically, we can not ensure the monotonic decrease of the objective value $\mathbb{F}$. However, with sufficiently large $t_{\max}$, $X = X_\xi$ could be empirically a good approximation of a minimizer of $\mathbb{F}(X, \Sigma_\xi)$. Under the circumstance, suppose that $\Sigma_{\xi} = \Sigma_{\xi+1}$ holds. Then

- $s_{i,j} > \epsilon$ for all $(i,j) \notin \Sigma_\xi$ holds, which implies $X_{i,j} = 0$;
- for all $(i,j) \in \Sigma_\xi$, we have

$$\mathbb{T}(s)_{i,j} = t^{-1} X_{i,j}^{-1} \leq \epsilon, \ i.e., X_{i,j} \geq (\epsilon t)^{-1}. (43)$$

In summary, $\mathbb{T}(s)_{i,j} \geq 0$ holds for all entries $(i,j)$ and $\mathbb{T}(s)_{i,j} > \epsilon$ occurs only when $X_{i,j} = 0$. Since an optimal point requires that $\mathbb{T}(s)_{i,j} > 0$ occurs only when $X_{i,j} = 0$, then $X_\xi$ can be regarded as an approximate optimal point. In addition, since $MX_\xi = \mathbb{1}_{2n}$, we have

$$c^T \mathbb{T}^{-1}(X_\xi) - \mathbb{1}_{2n}^T \nu_\xi = \langle s, \mathbb{T}^{-1}(X_\xi) \rangle \leq t^{-1} |\Sigma_\xi|,$$

(44)

### 2.4 Optimal transport by matrix balancing

Optimal transport can be approximated by the entropic regularized optimal transport problem [Cut13, CPSV18]. The entropic regularized optimal transport problem can be formed by replacing the logarithmic barrier function in interior point methods with the negative entropy function $-x \log x - x$. Indeed, we obtain a regularized problem with $t > 0$,

$$\min_{Mx = \mathbb{1}_{2n}} \{ \langle c, x \rangle + t^{-1} \langle \mathbb{1}_{n^2}, x \odot \log x - x \rangle \}.$$  

(45)

The strict convexity of $x \odot \log x$ implies the uniqueness of the minimizer in $\mathbb{R}_{+}^{n^2}$. The first-order optimal condition suggests that the optimal solution can be computed by matrix scaling algorithms. Introducing a multiplier vector $\nu$ for the constraint, we have the problem

$$\min_{Mx = \mathbb{1}_{2n}} \{ \langle c, x \rangle + t^{-1} \langle \mathbb{1}_{n^2}, x \odot \log x - x \rangle - \langle \nu, Mx - \mathbb{1}_{2n} \rangle \}.$$  

(46)
The gradient computation gives the optimal condition of $x$,

$$c + t^{-1} \log x - M^T \nu = 0, \quad \text{i.e., } x = \exp(-t(c - M^T \nu)). \quad (47)$$

The multiplier vector $\nu := [\nu^{(1)}; \nu^{(2)}]$ in (47) can be determined in matrix balancing of $\exp(-tT(c))$. Indeed, since $T(M^T \nu) = \nu^{(1)}1_n^T + 1_n\nu^{(2)}$, then (47) yields that $T(x) \in \Pi_n$ is obtained under proper scaling matrices,

$$T(x) = T(\exp(-t(c - M^T \nu))) = \text{diag}(\exp(t\nu^{(1)}))\exp(-tT(c))\text{diag}(\exp(t\nu^{(2)})) \in \Pi_n. \quad (48)$$

Various Newton methods can be employed to perform matrix balancing in (48). The details of matrix balancing algorithms will be presented in next section.

The quality of the approximation improves under large $t$. However, the problem with large $t$ becomes generally very ill-conditioned and is hard to solve. Similar to the IPMB algorithm, with $\eta > 1$, to alleviate the ill-condition issue, we solve $\nu$ in a sequence of problems associated with $t = t(0)$ initially and with $t \rightarrow \eta t$ iteratively to reach $t_{\text{max}}$. This method is known as $\epsilon$-scaling heuristic in [Sch19] with $1/t$ replaced with $\epsilon \rightarrow 0$. We shall call the interior point method in solving (45) with $t \rightarrow \infty$ as the Sinkhorn-Newton-Newton-negative-entropy method (SNNE).

- Initialize $t = t_0$ and $\nu = \nu_{ini}$. Repeat the following two steps until $t = t_{\max}$.
  - Employ matrix balancing algorithms to update $\nu$, i.e., $\exp(-tT(c - M\nu))$ is doubly stochastic.
  - If $t < t_{\max}$, update $t \rightarrow t\eta$.

### 2.4.1 Sparse support

Similar to IPMB in Alg. 2.7 for large point-set problems, we can incorporate the sparse support estimation into the aforementioned SNNE algorithm. Observe that thanks to (48), the Lagrange dual of (45) is given by

$$\max_{\nu} \left\{ -t^{-1}(\exp(t\nu^{(1)}), \exp(-tT(c))\exp(t\nu^{(2)})) + \langle \nu, 1_{2n} \rangle \right\}. \quad (49)$$

Now introduce a sparse support set $\Sigma$ for $x$. Then $x$ satisfies

$$x = \Pi^{-1}(1_{\Sigma}) \circ \exp(-t(c - M^T \nu)). \quad (50)$$

Since $1_{\Sigma^c} \exp(-tT(c - M^T \nu)) \geq 0$ holds for the complement set $\Sigma^c := \Pi_n \setminus \Sigma$, then (49) can be approximated by an upper bound

$$\max_{\nu} \{-t^{-1}(1_{\Sigma}, \exp(-tT(c - M^T \nu))) + \langle \nu, 1_{2n} \rangle\}. \quad (51)$$

To reach a tight approximation to (49), we choose the support set $\Sigma$ in (51), consisting of those $(i, j)$ corresponding to small entries in $(c - M^T \nu)_{i,j}$. The index selection from $c - M^T \nu$ is exactly the sparse kernel truncation rule proposed in [Sch19].

Let $x$ be given in (47), i.e., the solution of (45) with support $\Sigma$. From the computation in (47), the solution $(x, \nu)$ gives the duality gap

$$t^{-1} n \log n \geq c^T x - \nu^T 1_{2n} = -t^{-1} \langle x, \log x \rangle \geq 0, \quad (52)$$

where we use Jensen inequality on $- \log x$ and $x \log x$ to establish the lower and upper bounds. Hence, $x$ is a good approximate solution of (3) under sufficiently large $t$. With $s = c - M^T \nu$, the optimal condition in (47) can be expressed as $s = -t^{-1} \log x \geq 0$. Fixing $\gamma' \in (0, 1)$ and $\gamma'' \in (1, \infty)$, we can compute an approximate solution $x$ with

$$\gamma' t^{-1} (-x \odot \log x) \leq s \odot x \leq \gamma'' t^{-1} (-x \odot \log x) \quad (53)$$

As $t \rightarrow \infty$, we reach the KKT condition in (7),

$$0 \leq x \odot s = -t^{-1} x \odot \log x \leq (\epsilon t)^{-1} \rightarrow 0. \quad (54)$$
Similar to IPMB, an optimal solution to optimal transport can also be reached with the aid of early termination in Prop. A.5 and Prop. A.6.

As pointed in Theorem 1, the support set must satisfy total support condition to ensure the existence of scaling vectors \( \nu^{(1)} \) and \( \nu^{(2)} \). The following algorithm is exactly Alg. 2.7 where each \( X_\xi \) is approximated by matrix balancing.

**Algorithm 2.9 (SNNE with sparse support).** Input: parameters \( \epsilon > 0, \xi_{\text{max}} > 0, t_{\text{max}} > 0 \) and the assignment matrix \( c \). Generate one initial support set \( \Sigma_0 \) fulfilling the total support condition. For \( \xi = 0, 1, 2, 3, \ldots, \xi_{\text{max}}, \) repeat the following steps.

- Use matrix balancing in section 3.3 to solve \( (52) \) as one approximation \( X_\xi \) from the optimal transport with respect to the support constraint \( \Sigma_\xi \).

\[
\min_X \{ \{I(c), X \} : \text{supp}(X) \subset \Sigma_\xi, M\Sigma^{-1}(X) = 1_{2n} \}. \tag{55}
\]

Indeed, the major task is to apply the SNNE algorithm to get the multiplier \( \nu_\xi = [\nu^{(1)}; \nu^{(2)}] \) associated to the support set, i.e., matrix balancing on \( \exp(-tc) \odot I_{\Sigma_\xi} \) with \( t \) tending to \( t_{\text{max}} \).

- Let \( \Sigma_{\xi+1} \) be one total support set, containing the entries in \( (72) \). That is, choose

\[
\Sigma'' = \{(i, j) : c(i, j) - \nu^{(1)}(i) - \nu^{(2)}(j) < \epsilon \} \tag{56}
\]

and let \( \Sigma_{\xi+1} \) be the union set described in \( (36) \) of nonzero entries.

**Remark 2.10 (SNNE-sparse).** We make three remarks. First, as \( t_{\text{max}} \) is sufficiently large, \( x \) in \( (55) \) is a good approximation to \( (52) \). Suppose in addition, \( \Sigma_\xi = \Sigma_{\xi+1} \). By the arguments in Remark 2.8 \( X_\xi \) can be regarded as an approximate optimal point. Second, in general a sparse support set \( \Sigma'' \) in \( (56) \) does not have total support, which easily yields the failure of matrix balancing. Empirically, we can choose \( \sigma \) to be the identity. Then the union set \( \Sigma_{\xi+1} \) is the support of the matrix

\[
I + 1_{\Sigma} + 1^T_{\Sigma}. \tag{57}
\]

Third, sometimes it is not easy to choose a proper parameter \( \epsilon > 0 \) to meet desired sparsity. One practicable manner is to select a parameter \( k > 0 \) and let \( \Sigma \) consist of those \( (i, j) \) corresponding to (at most) \( k \) smallest entries \( (c - M^T \nu)_{i,j} \) for each row and each column. In this manner, \( \Sigma_\xi \) consists of at most \( (2k + 1)n \) entries. We shall call this method SNNE-sparse. In section 4.3 we shall present numerical experiments under a proper value \( k \) to demonstrate the effectiveness. Surely, theoretically we cannot ensure the convergence to an optimal solution.

### 3 Matrix balancing

Let \( A \) denote a positive matrix in \( \mathbb{R}^{n \times n} \). Matrix balancing \([\text{Sin}44]\) aims to find a pair of positive scaling vectors \( r, c \in \mathbb{R}^n \), so that the matrix balancing projection

\[
A' := \text{diag}(r)A\text{diag}(c)
\]

is doubly stochastic, i.e.,

\[
A'1_n = \text{diag}(r)Ac = \text{diag}(r)\text{diag}(c)1_n = 1_n, \tag{58}
\]

\[
A'^T1_n = \text{diag}(c)A^Tr = (\text{diag}(r)\text{diag}(c))^T1_n = 1_n. \tag{59}
\]

The existence of \( r, c \) is proved in \([\text{Sin}44], [\text{KS}67]\) for any positive matrix and any nonnegative matrix, respectively. Literature of matrix scaling methods and its various applications in scientific computing, statistics and engineering can be found in the extensive survey \([\text{Ide}16]\) and the references therein. In general, the prescribed row sums and column sums do not have to be restricted to \( 1_n \). See \([\text{KLRS}08]\) and \([\text{AZLOW}17]\). Here, we only give a brief introduction related to our application. Matrix scaling serves two distinct functions in this paper: a projection to ensure the iterations \( x^{(t)} \in \Pi_n \); the construction
of an optimal solution in \cite{48} for negative entropy barrier functions. In the section, we shall list a few
matrix scaling algorithms and their variant.

### 3.1 Sinkhorn-Knopp balancing (SK) and Knight-Ruiz (KR) method

For the conditions in \cite{58,59}, the Sinkhorn-Knopp balancing (SK) (also known as the RAS or bipropor-
tional problem \cite{Bac70}) is one well-known method to carry out matrix balancing on $A$, consisting of iterates $\{(r_k, c_k) : k = 1, 2, 3, \ldots\}$,

$$c_{k+1} = (A^T r_k)^{-1}, \quad r_{k+1} = (A c_k)^{-1}.$$  \hfill (60)

We can express (60) in a symmetric manner \cite{Kni08}. Form one symmetric matrix $\tilde{A}$ from $A$,

$$\tilde{A} := \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}. \hfill (61)$$

Let $\zeta_k := [r_k; c_k]$ be a sequence of the scaling vectors. When

$$c_1 = (A^T r_1)^{-1}, \hfill (62)$$

the SK algorithm in (61) can be expressed in a compact form,

$$\zeta_{k+1} = (\tilde{A}\zeta_k)^{-1},$$

whose limit $\zeta = \lim_{k \to \infty} \zeta_k$ is actually a root of

$$g(\zeta) := \zeta \odot (\tilde{A}\zeta) - 1_{2 n} = 0.$$  \hfill (63)

\textbf{Remark 3.1 (\((r, c)\)-balancing).} In this paper, we focus on the application of point-set matching problems and thus consider the matrix balancing with $(1_n, 1_n)$-balancing, i.e., the row sum and the column sum both 1. In literatures, e.g., section 3 in \cite{Ide16}, SK algorithms can be applied to reach a matrix with row sum $r$ and column sum $c$, where $(r, c)$ is not necessarily restricted to $(1_n, 1_n)$.

To solve the roots of $g(\zeta) = 0$, Knight and Ruiz \cite{KR12} proposed one Newton method,

$$\zeta_{k+1} = \zeta_k - (\text{diag}(\zeta_k) \tilde{A} + \text{diag}(\tilde{A}\zeta_k))^\dagger (\zeta_k \odot (\tilde{A}\zeta_k) - 1_{2n}) \hfill (64)$$

$$= \zeta_k \odot \left\{ 1 - (A_k + \text{diag}(A_k 1_{2n}))^\dagger (A_k 1_{2n} - 1_{2n}) \right\} \hfill (65)$$

to alleviate slow convergence of SK, where $A_k := \text{diag}(\zeta_k) \tilde{A} \text{diag}(\zeta_k)$ is used. Compared with the SK algorithm, the Newton approach exhibits fast convergence. However, as mentioned in \cite{KR12}, the global convergence property of (64) is theoretically unclear.

### 3.2 Negative entropy (NE) based matrix balancing

We describe one algorithm proposed in \cite{CMTV17,BCLW17}, which implements Newton’s method for matrix balancing in \cite{48}. To simplify the notation, consider $t = 1$ in (48) and let

$$A = \exp(- t \Gamma(c)) \in R^{n \times n}. \hfill (66)$$

Introduce the symmetric $2n \times 2n$-matrix $\tilde{A}$ as in \cite{61}. Write the scaling vector $\exp(\nu)$ of $\tilde{A}$ with $\nu := [y; z]$ and $y \in R^n$ and $z \in R^n$. Matrix balancing on $A$ can be solved by the convex optimization (i.e., the problem in (49)),

$$\min_{\nu \in R^n} \left\{ f(\nu) = \frac{1}{2} \langle \exp(\nu), \tilde{A} \exp(\nu) \rangle - \langle 1_{2n}, \nu \rangle \right\}. \hfill (67)$$
Indeed, reformulate (67) as follows:

\[
\mathbf{f}(\nu) = \langle \exp(y), \exp(-\mathbf{1}(c)) \exp(z) \rangle - \langle \mathbf{1}_n, y \rangle - \langle \mathbf{1}_n, z \rangle
\]  

(68)

(\begin{align*}
&= \langle \mathbf{1}_n, \exp(-\mathbf{1}(c - M^T \nu)) \mathbf{1}_n \rangle - \langle \mathbf{1}_n, y \rangle - \langle \mathbf{1}_n, z \rangle \\
&= \langle \mathbf{1}_n, \mathbf{A}(\nu) \mathbf{1}_n \rangle - \langle \mathbf{1}_2n, \nu \rangle, 
\end{align*} 

(69)

(70)

where \( \mathbf{A}(\nu) \) denotes the scaled matrix of \( \mathbf{A} \).

\[
\mathbf{A}(\nu) := \exp(-\mathbf{1}(c - M^T \nu)) = \text{diag}(\exp(y)) \text{Adiac}(\exp(z)), \quad \text{and} \quad \mathbf{T}(M^T \nu) = y \mathbf{1}_n^T + \mathbf{1}_n z^T. 
\]  

(71)

First, a scaling vector \( \nu \) with \( \nabla \mathbf{f}(\nu) = 0 \) yields the double stochastic matrix \( \mathbf{A}(\nu) \). Indeed,

\[
\nabla \mathbf{f}(\nu) = M \exp(-\mathbf{T}(c - M^T \nu)) \mathbf{1}_n - \mathbf{1}_{2n} = \left( \begin{array}{c} \exp(y) \odot (A \exp(z)) \\ \exp(z) \odot (A^T \exp(y)) \end{array} \right) - \mathbf{1}_{2n} 
\]  

(72)

(\begin{align*}
&= \begin{pmatrix} \mathbf{A}(\nu) - I_n \mathbf{1}_n \mathbf{1}_n^T \end{pmatrix} I_n. 
\end{align*} 

(73)

Second, the Hessian computation verifies the convexity of \( \mathbf{f} \). Computation shows

\[
\nabla^2 \mathbf{f}(\nu) = \begin{pmatrix} \text{diag}(\exp(y) \odot (A \exp(z)) & \text{diag}(\exp(y) \text{Adiac}(\exp(z))) \\ \text{diag}(\exp(z) \odot (A^T \exp(y))) & \text{diag}(\exp(y) \odot (A^T \exp(y))) \end{pmatrix} 
\]  

(74)

and

\[
\begin{pmatrix} \text{diag}(\exp(y) \odot (A \exp(z)) & \text{diag}(\exp(y) \text{Adiac}(\exp(z))) \\ \text{diag}(\exp(z) \odot (A^T \exp(y))) & \text{diag}(\exp(y) \odot (A^T \exp(y))) \end{pmatrix} 
\]  

(75)

\[
\begin{pmatrix} \text{diag}(\mathbf{A}(\nu) \mathbf{1}_n) & \mathbf{A}(\nu) \mathbf{1}_n^T \\ \mathbf{A}(\nu) \mathbf{1}_n^T & \text{diag}(\mathbf{A}(\nu) \mathbf{1}_n^T) \end{pmatrix}. 
\]  

(76)

We can employ Newton’s method with step size given by backtracking line search to compute a minimizer of the problem in (67). See section 9.5.3 [BV04]. Note that from (48), the Hessian \( \nabla^2 \mathbf{f} \) is symmetric. Indeed, since \( \mathbf{A}(\nu) \) is symmetric, the Schur complement matrix \( \mathbf{M} \) in (22). The Hessian matrix will undergo the rank-reduction, when \( x \) heads to an optimal permutation. The following shows the consistency analysis and provides the computation of the associated Newton decrement.

**Proposition 3.2.** Suppose the matrix \( \mathbf{A} \) is nonnegative and has support. Then the system

\[
\nabla^2 \mathbf{f}(\nu_k) w = -\nabla \mathbf{f}(\nu_k) 
\]  

(77)

is consistent for some vector \( w \in \mathbb{R}^{2n} \). In addition, for nonzero \( \nabla \mathbf{f}(\nu_k) \), let \( u = - (\nabla^2 \mathbf{f}(\nu_k))^\dagger \nabla \mathbf{f}(\nu_k) \).

Then we have the squared **Newton decrement**

\[
\langle u, \nabla^2 \mathbf{f}(\nu_k) w \rangle = \langle \nabla \mathbf{f}(\nu_k), (\nabla^2 \mathbf{f}(\nu_k))^\dagger \nabla \mathbf{f}(\nu_k) \rangle > 0. 
\]  

(78)

**Proof.** For each vector \( w = [w^{(1)}; w^{(2)}] \in \mathbb{R}^{2n} \) with \( w^{(1)} \in \mathbb{R}^n, w^{(2)} \in \mathbb{R}^n \), the Hessian \( \nabla^2 \mathbf{f}(\nu) \) is symmetric diagonally dominant [CMTV17;AZLOW17], thus positive semi-definite,

\[
\langle w, \nabla^2 \mathbf{f}(\nu) w \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} e^{y_i} e^{z_j} (w^{(1)}_i + w^{(2)}_j)^2 \geq 0, 
\]  

(79)

which shows the convexity of \( \mathbf{f} \). For each vector \( w \) in the null space of \( \nabla^2 \mathbf{f} \), from (79), \( w \) satisfies \( \langle w, \nabla^2 \mathbf{f}(\nu) w \rangle = 0 \), which implies

\[
w^{(1)}_i + w^{(2)}_j = 0 \text{ for all } A_{i,j} > 0. 
\]  

(80)

Since \( A \) has support, then \( \{A_{i,\sigma(i)} : i = 1, 2, \ldots, n\} \) are all positive for some permutation \( \sigma \). Since
Thus, SK balancing decreases the objective in (67). Second, suppose a minimizer \( \tilde{\nu} \) exists. The introduction can shed light on convergence of Knight-Ruiz algorithm. Define here, we provide another Newton method for matrix balancing, similar to the previous scheme in (85).

Thanks to (78), when \( \nu = \zeta > 0 \), then any vector \( w \) in the null space of \( \nabla^2 f \) satisfies \( w^{(1)} = -w_{\sigma(i)} \) and has the form

\[
    w := [w^{(1)}, w^{(2)}] = [w_1, w_2, \ldots, w_n, -w_{\sigma^{-1}(1)}, \ldots, -w_{\sigma^{-1}(n)}]^T. 
\]

(81)

Clearly, \( \langle w^{(1)}, \mathbb{1}_n \rangle + \langle w^{(2)}, \mathbb{1}_n \rangle = 0 \) holds. Thus, we have the orthogonality between \( -\nabla f(\nu_k) \) and the null space of \( \nabla^2 f(\nu) \). Indeed,

\[
    \langle w, \nabla f(\nu_k) \rangle = w^{(1)}^T (A(\nu_k) - I_n) \mathbb{1}_n + w^{(2)}^T (A(\nu_k) - I_n)^T \mathbb{1}_n 
\]

\[
    = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i^{(1)} A_{i,j} e^{u_i} e^{z_j} + \sum_{i=1}^{n} \sum_{j=1}^{n} w_j^{(2)} A_{i,j} e^{u_i} e^{z_j} 
\]

\[
    = \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i^{(1)} + w_j^{(2)}) A_{i,j} e^{u_i} e^{z_j} = 0, 
\]

(82)

(83)

(84)

where we used (80). Hence, \( -\nabla f(\nu_k) \) lies in the range of \( \nabla^2 f \), which verifies that the system in (77) is consistent. Finally, we obtain (78) according to the positive semi-definite property in (79) and the following observation. When \( \nabla f(\nu_k) \) is orthogonal to the null space of \( \nabla^2 f(\nu_k) \), then \( \nabla f(\nu_k) \) is orthogonal to the null space of \( \nabla^2 f(\nu_k) \).

Since \( \nabla^2 f(\nu_k)(\nu_{k+1} - \nu_k) = -\nabla f(\nu_k) \) is consistent, the Newton iterations

\[
    \nu_{k+1} = \nu_k - \alpha (\nabla^2 f(\nu_k))^{-1} \nabla f(\nu_k). 
\]

(85)

can be employed to find \( \nu \) with \( \nabla f(\nu) = 0 \), e.g., the conjugate gradient method [BCLW17]. Note that the squared Newton decrement in (78) can be interpreted as the directional derivative of \( f \) in the direction of \( u \),

\[
    -\langle u, \nabla^2 f(\nu_k) u \rangle = \langle \nabla f(\nu), u \rangle = \left. \frac{d}{d\alpha} f(\nu + \alpha u) \right|_{\alpha=0}. 
\]

(86)

(87)

(88)

(89)

3.3 Logarithmic barrier functions (LB) based matrix balancing

Here, we provide another Newton method for matrix balancing, similar to the previous scheme in (85). The introduction can shed light on convergence of Knight-Ruiz algorithm. Define \( \tilde{A} \) as in (61). Consider the minimization of \( g \),

\[
    \min_{\zeta > 0} \{ g(\zeta) := \frac{1}{2} \zeta^T \tilde{A} \zeta - \frac{1}{2} \log \zeta \}. 
\]

(87)

The objective function in (87) is identical to the function in (67), except for the variable \( \nu \) replaced with \( \log \zeta \). In [MO68], the function \( g \) is employed to show the existence of matrix-scaling on a fully indecomposable matrix. In [KK92], authors proposed one path-following Newton algorithm, minimizing a sequence of sub-problems to scale a symmetric positive semi-definite matrix \( \tilde{A} \), so that convergence requirement of Newton iterates can be met in each sub-problem. Here, we propose a modified Newton method for the computation of matrix balancing for one positive matrix \( A \).

Compute the gradient and the Hessian of \( g \),

\[
    \nabla g = \tilde{A} \zeta - \zeta^{-1}, \quad \nabla^2 g(\zeta) = \tilde{A} + \text{diag}(\zeta^{-2}),
\]

(88)

respectively. First, from (88), the Sinkhorn-Knopp balancing is the coordinate descent iteration of \( g(\zeta) \) with \( \zeta = [\zeta^{(1)}, \zeta^{(2)}] \),

\[
    \zeta^{(1)}_{k+1} \leftarrow \text{arg min}_{\zeta^{(1)}} g([\zeta^{(1)}_{k}, \zeta^{(2)}]), \quad \zeta^{(2)}_{k+1} \leftarrow \text{arg min}_{\zeta^{(2)}} g([\zeta^{(1)}_{k+1}, \zeta^{(2)}])
\]

Thus, SK balancing decreases the objective \( g \) in (87) (similarly, SK balancing decreases the objective \( f \) in (67)). Second, suppose a minimizer \( \zeta \) is an interior point in \( \mathbb{R}^n_+ \). Clearly, \( \zeta \) is a root to (63),

15
globally convergent. To alleviate this difficulty, we introduce one modified Newton iteration with some Remark 3.3 (KR method is a special case with $\alpha_k$ in (94)). Similar to

\[ \zeta_{k+1} = \zeta_k - \alpha_k \nabla^2 g(\zeta_k)^{-1} \nabla g(\zeta_k) = \zeta_k - \alpha_k (\tilde{A} + \text{diag}(\zeta_k^{-2}))^{-1}(\tilde{A}\zeta_k - \zeta_k) \]

for $k = 1, 2, 3, \ldots$. With $A_k := \text{diag}(\zeta_k) \tilde{A} \text{diag}(\zeta_k)$. 

Since the matrix $I_{2n} + A_k$ in (92) is not necessarily positive definite, the iteration in (92) is not globally convergent. To alleviate this difficulty, we introduce one modified Newton iteration with some proper positive diagonal matrices $C_k \in \mathbb{R}^{2n \times 2n}$, (replacing $I_{2n}$ in (92) with $C_k$)

\[ \zeta_{k+1} = \zeta_k \circ \{I_{2n} - \alpha_k (C_k + A_k)\} (A_k - I_{2n})'_2n \].

We dynamically select $C_k$ in (94) according to the rule

\[ C_k = \text{diag}(A_k I_{2n}). \]

Remark 3.3 (KR method is a special case with $\alpha_k = 1$). The whole algorithm in (94) is one kind of compromise between the steepest descent method and Newton’s method. The objective values decrease monotonically. As $A_k$ tends to be a doubly stochastic matrix, we have $C_k = \text{diag}(A_k I_{2n}) \rightarrow I_{2n}$ and the method reduces to Newton’s method in (93). Note that the algorithm in (94) with $\alpha_k = 1$ coincides with the algorithm proposed by Knight and Ruiz in (64).

The following proposition shows the well definedness of $(C_k + A_k)'_2n(A_k - I_{2n})'_2n$ in (94). Similar to Prop. 3.2, we can calculate the directional derivative of $g$ in the direction of $u := \zeta_k \circ (C_k + A_k)'_2n(A_k - I_{2n})'_2n$ with $\|u\| > 0$, which sheds some light on the convergence of this Newton method,

\[ \frac{d}{d\alpha} g(\zeta_k + \alpha u)|_{\alpha = 0} = \langle \nabla g(\zeta_k), u \rangle = - \langle (A_k - I_{2n})'_2n, (C_k + A_k)'_2n(A_k - I_{2n})'_2n \rangle < 0. \]

In the following, we shall verify the calculation in (97). For notation simplicity, we drop the subscript $k$.

**Proposition 3.4.** Suppose the matrix $A$ is nonnegative and has support. Let $\tilde{A}$ be given in (61). Let $C + \tilde{A}$ is symmetric and positive semi-definite and the system

\[ (C + \tilde{A}) y = (\tilde{A} - I_{2n})'_2n \]

is consistent. For each vector $(\tilde{A} - I_{2n})'_2n \neq 0$, the directional derivative of $g$ in the direction of $u := \zeta \circ (C + \tilde{A})' (\tilde{A} - I_{2n})'_2n$ is

\[ \frac{d}{d\alpha} g(\zeta + \alpha u)|_{\alpha = 0} = - \langle (\tilde{A} - I_{2n})'_2n, (C + \tilde{A})' (\tilde{A} - I_{2n})'_2n \rangle < 0. \]

In addition, for any positive vector $\zeta \in \mathbb{R}^{2n}$ and for any null vector $w$ of $C + \tilde{A}$, the function $g$ takes a constant value, as $\zeta \rightarrow \zeta \circ \exp(w)$, i.e.,

\[ g(\zeta \circ \exp(w)) = g(\zeta). \]

**Proof.** By Gershgorin circle theorem, the symmetric matrix $C + \tilde{A} \geq 0$ is diagonally dominant and thus
is a positive semi-definite matrix. Actually, for each vector \( w = [w^{(1)}; w^{(2)}] \in \mathbb{R}^{2n} \),

\[
\langle w, (C + \tilde{A})w \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j}(w_j^{(1)} + w_j^{(2)})^2 \geq 0.
\]  \(101\)

Hence, for each null vector \( w \) of \( C + \tilde{A} \), then

\[
w_j^{(1)} + w_j^{(2)} = 0 \quad \text{for all } j.
\]  \(102\)

Since \( A \) has support, then for some permutation \( \sigma \), we have \( A_{i,\sigma(i)} > 0 \) for \( i = 1, \ldots, n \). Hence, \(102\) implies

\[
\sum_{i=1}^{n} w_i^{(1)} + \sum_{j=1}^{n} w_j^{(2)} = \sum_{i=1}^{n} w_i^{(1)} + \sum_{i=1}^{n} w_{\sigma(i)}^{(2)} = 0.
\]  \(103\)

Next, we show that \((\tilde{A} - I_{2n})I_{2n}\) lies in the range of \((C + \tilde{A})\). Indeed, \((\tilde{A} - I_{2n})I_{2n}\) is orthogonal to the null space of \((C + \tilde{A})^\top = C + \tilde{A} \), i.e., using \(102\) and \(103\), for each null vector \( w \), we have

\[
\langle w, (\tilde{A} - I_{2n})I_{2n} \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i^{(1)} + w_j^{(2)})A_{i,j} - (\sum_{i=1}^{n} w_i^{(1)} + \sum_{j=1}^{n} w_j^{(2)}) = 0.
\]  \(104\)

The above orthogonality arguments also implies \(97\). Finally, according to \(102\) and \(103\),

\[
g(\zeta \odot \exp(w)) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} s_i^{(1)} s_j^{(2)} \exp(w_i^{(1)} + w_j^{(2)}) - \langle I_{2n}, \log \zeta \rangle - \langle I_{2n}, w \rangle
\]  \(105\)

\[
= \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} s_i^{(1)} s_j^{(2)} - \langle I_{2n}, \log \zeta \rangle = g(\zeta).
\]  \(106\)

Lastly, we examine the boundedness of the sequence \( \{\zeta_k : k \} \) under total support assumption on \( A \).

When \( A \) has total support, then \( \zeta_k = [\zeta_k^{(1)}, \zeta_k^{(2)}] \) from \(94\) generates a bounded matrix \( s_k^{(1)}, s_k^{(2)} \in \mathbb{R}^{n \times n} \), whose entries stay away from 0. For notation simplicity, we drop the subscript \( k \).

**Proposition 3.5.** Suppose \( A \in \mathbb{R}^{n \times n} \) has total support. Let \( \Sigma := \{(i, j) : A_{i,j} > 0 \} \). Let \( \delta \) be a positive lower bound for \( \{A_{i,j} : (i, j) \in \Sigma\} \). Fix some \( c_0 \in \mathbb{R} \). Let \( \zeta = [\zeta^{(1)}, \zeta^{(2)}] \) be a positive vector in the \( c_0 \)-sublevel set of \( g \), i.e., \( g(\zeta) \leq c_0 \). Then \( \{\zeta^{(1)} s_j^{(2)} : (i, j) \in \Sigma\} \) are bounded below by

\[
\exp(-c_0 + (n-1)(1 + \log \delta))
\]  \(107\)

and bounded above by

\[
\zeta_j^{(1)} \zeta_j^{(2)} \leq \max(\delta^{-1}(c_0 - (n-1)(1 + \log \delta)), 1).
\]  \(108\)

**Proof.** By assumption, \( A \) has total support. Fix one entry \( A_{i_1,j_1} > 0 \). Then \( (i_1, j_1) \) lies on some diagonal \( \{(i, \sigma(i)) : i \in \{1, 2, \ldots, n\}\} \). Then

\[
\sum_{i=1}^{n} \{A_{i,\sigma(i)} s_i^{(1)} s_{\sigma(i)}^{(2)} - \log(s_i^{(1)} s_{\sigma(i)}^{(2)})\} \leq g(\zeta) = \langle \zeta, A\zeta \rangle - \langle I_{2n}, \log \zeta \rangle \leq c_0
\]  \(109\)

Note that by convexity, the following inequality holds for each \( a > 0 \),

\[
\min_{x \geq 0} (ax - \log x) \geq 1 + \log a.
\]  \(110\)
Applying \(110\) to the right hand side of \(109\) corresponding to \(i \neq i_1\), we have
\[
\sum_{i=1}^{n} \{A_{i, \sigma(i)} \zeta_{i}^{(1)} \zeta_{i}^{(2)} - \log(\zeta_{i}^{(1)} \zeta_{i}^{(2)})\} \geq A_{i_1, j_1} \zeta_{i_1}^{(1)} \zeta_{j_1}^{(2)} - \log(\zeta_{i_1}^{(1)} \zeta_{j_1}^{(2)}) + (n-1)(1 + \log \delta). \tag{111}
\]
Together with \(109\), dropping \(A_{i, \sigma(i)} \zeta_{i}^{(1)} \zeta_{i}^{(2)}\) in \(111\), we have \(107\). Likewise, for an upper bound, when \(\zeta_{i_1}^{(1)} \zeta_{j_1}^{(2)} \geq 1\), we can drop \(- \log(\zeta_{i_1}^{(1)} \zeta_{j_1}^{(2)})\) in \(111\), which yields the upper bound in \(108\).

The following remark indicates that optimal scaling vectors of each sub-problem in SNNE-sparse are bounded, if the matrix \(A\) has total support.

**Remark 3.6.** By the definition of \(C_k\), the vector \([1_n; -1_n]\) lies in the null space of \(C_k + A_k\) and thus \(g(\zeta \otimes \exp(s[1_n; -1_n])) = g(\zeta)\) holds for any \(s \in \mathbb{R}\). By the proper scaling vector \(\exp(s[1_n; -1_n])\), we may assume that \(\zeta^{(1)}\) is bounded below by \(\epsilon > 0\). Then \(107\) indicates that \(\zeta^{(2)}\) is bounded below by
\[
\epsilon^{-1} \exp(-c_0 + (n-1)(1 + \log \delta)). \tag{112}
\]
Together with the upper bound in \(108\), we have a bounded sequence \(\{\zeta_k\}\) from the total support of \(A\). Since the bounds are independent of \(k\), these bounds also work for the optimal scaling vectors, i.e., the minimizer in \(87\).

**Remark 3.7.** We implement \(94\) as follows. Compute \(A_k\) and \(C_k\) from \(95, 95\) and use conjugate gradient to solve \(y\) from the consistent system,
\[
(C_k + A_k) y = (A_k - I_{2n}) z_{2n}. \tag{113}
\]
Update \(\zeta_{k+1} = \zeta_k \odot (1 - \alpha_k y)\) for a proper \(\alpha_k\), ensuring the decrease of \(g\).

## 4 Numerical Simulations

We provide three experiments in the section: (i) Comparison experiments of interior point methods in solving discrete optimal transport; (ii) Comparison of matrix balancing schemes; (iii) Application of sparse support algorithms on large data-sets.

### 4.1 Barrier functions and early termination

First, we demonstrate the improvement of early termination for IPMB described in section A.2 in solving the discrete optimal transport problem in \(3\).

Let \(x_{\text{opt}}\) be an optimal minimizer of \(3\). At the \(k\)-th iteration, the performance metric is given by the relative error of \(x_k\),
\[
|c^T x_{\text{opt}}|^{-1} |c^T x_k - c^T x_{\text{opt}}|,
\]
where \(x_k\) is the optimal solution of each sub-problem (given in \(1\) and \(59\), respectively) associated with \(t_k = t_0 k^{\mu}, \mu = 2\). To illustrate the performance of IPMB, we consider Alg. \(2.3\) with adjustments:

(a) IP-alone. Apply Alg. \(2.3\) but disable matrix balancing and early termination. Here we choose \(\epsilon = 10^{-15}\) and thus the stopping condition \(n^t t^{-1} < \epsilon\) is not met within 30 iterations.

(b) IPMB. Apply Alg. \(2.3\) but disable early termination conditions in the stopping criterion.

(c) IP+ET. Apply Alg. \(2.3\) with early termination 1 or early termination 2, but disable the second step: matrix balancing.

(d) IPMB+ET. Apply Alg. \(2.3\)

Here, early termination 1 and 2 shall be described in \(148\) and \(158\), respectively. We conduct three experiments to demonstrate the importance of matrix balancing. Here, we also compare these approaches with one matlab command “fmincon”, which directly uses the primal-dual algorithm to solve the sequence of constrained optimization problems in \(6\).
• Consider the assignment matrix $c$ from the $L^2$ displacement of lung data sets named H6012, which has been considered in [CLC13]. The data size is $n = 254$. The relative errors of case (a-d) are reported in the top figures of Fig. 1. When $t_k$ gets large, without matrix balancing the iterates of interior points can deviate from $\Pi_n$. For simplicity, we apply Sinkhorn-Knopp matrix scaling in (b,d). In the right zoom-in figure, the objective curve “IP alone” is deviated from the optimal objective value, which shows the importance of matrix balancing. The stopping criterion of the Sinkhorn matrix balancing

$$\|X_1 - 1_n\|_1 + \|X^T 1_n - 1_n\|_1 \leq \epsilon,$$

where the tolerance $\epsilon = 10^{-3}$ is used. In addition, we apply the early-termination-1 (ET1) to retrieve one optimal permutation before the rank degeneracy of Schur complement matrices. Clearly, with the aid of ET1, the optimal solution is reached within 5 iterations, i.e., $t_5 = 2^5 = 32$. Note that we do not get the exact optimal solution in the case (a) IP-alone due to the rounding error. The bottom figures of Fig. 1 show one comparison with the matlab command ‘fmincon’. Due to the memory requirement, the command cannot handle the whole point-set. Here, we subsample 50 pair of points to form the assignment matrix $c$. The result indicates certain numerical error under the application of ‘fmincon’, even the size of the point set is only $n = 50$.

• Let $\text{magic}(n)$ be the magic matrix with size $n \times n$. Next consider the assignment matrix $c$ from a magic matrix of size $20 \times 20$. The early-termination-1 in (138) cannot function well, since $x^{(c)}$ does not tend to a permutation matrix. See Fig. 3. Rounding errors in (a) can be alleviated by the application of matrix balancing, shown in the result of (b). With the aid of early-termination-2 (ET2), i.e., (155), the optimal solution can be reached within 12 iterations. See (c) and (d). Similar to Fig. 1, some small error still occurs in the direct application of the matlab command “fmincon”.

• To further illustrate the importance of matrix balancing, consider the assignment matrix $c$ from the $L^2$ displacement of one-dimensional non-uniform point sets $\{y_i = i/n : i = 1, \ldots, n\}$ and $\{z_i = (i/n)^2 : i = 1, \ldots, n\}$. Let $n = 100$. The optimal solution $x_{\text{opt}}$ is given by $\cal T(I_n)$. Use IP-alone, IPMB and IPMB+ET to solve (3). Figure 2 show the results of the following error metrics, including (i) the feasibility error,

$$\|\cal T(x_{t_k}) 1_n - 1_n\|_1 + \|\cal T(x_{t_k})^T 1_n - 1_n\|_1;$$

(ii) the 1-norm error, $\|x_{t_k} - x_{\text{opt}}\|_1$; (iii) the relative error given in (114). Observe that although the relative error of IP-alone is small, the 1-norm error is non-negligible, which is caused by its large feasibility error and thus incorrect Schur complement matrices. Obviously, the application of matrix balancing reduces the undesired 1-norm error. In addition, the early termination-1 condition is met around $k = 25$. The approach IP+ET does not work either, since the feasibility error is rather large and the early termination-1 condition cannot be satisfied. Lastly, the feasibility error of IPMB is determined by the tolerance parameter in matrix balancing.

• Lastly, we directly compare the performance between IPMB and Sinkhorn-Newton(SNNE). Results on the data set H6012 are shown in Fig. 4. Here, we employ slow SK matrix balancing in IPMB, thus 5 iterations in IPMB consume almost the same amount of time as 21 iterations in SNNE. Both IPMB and SNNE can be used to solve optimal transport, but there exist minor differences between their numerical performances. Exponential functions used in SNNE can easily cause the Hessians singular or ill-conditioned for large $t_k$ and thus rounding error could become a serious issue in SNNE without proper stopping criterion. In this case, early termination works well to report the optimal solution within 5 iterations of IPMB; instead SNNE takes 21 iterations to reach an early termination. The right figure of Fig. 4 demonstrates the decay of duality gap $\epsilon^T x - 1_{2n}^T \nu$ in the case without early-termination, which is consistent with (13) and (62). Clearly, IPMB yields a smaller duality gap than SNNE.
Figure 1: The left figures show the relative error versus the iterations of IPMB. The right figures are the zoom-in of the left figures. Here $c$ is generated from H6012. Top row figures show the result of the whole point-sets. Bottom row figures show the result of $n = 50$, i.e., subsampling 50 pairs of points from 254 pairs of points.
Figure 2: The top figures shows the feasibility error and the 1-norm error. The bottom right figure is the zoom-in of the bottom left figure, which shows the relative error in matching non-uniform point-sets.

Figure 3: The left figure shows the relative error versus the iterations of IPMB with $c = \text{magic}(20)$. The right figure is the zoom-in of the left figure.

Figure 4: The left figure shows the relative error versus CPU time of IPMB and SNNE on H6012. The right figure shows the duality gap.
4.2 Matrix balancing

We demonstrate matrix balancing algorithms on ill-conditioned matrices, \( A = \exp(-\text{magic}(20)/20) \) of size \( 20 \times 20 \), \( A = \exp(-\text{magic}(50)/20) \) of size \( 50 \times 50 \), and \( A = \exp(-\text{magic}(200)/50) \) of size \( 200 \times 200 \). See Fig. 5 for the pattern visualisation of the matrices \( \text{magic}(20) \), \( \text{magic}(50) \) and \( \text{magic}(200) \).

We compare four matrix balancing methods on \( A \), including

- Sinkhorn-Knopp algorithm (SK) in (60);
- three Newton based matrix balancing algorithms:
  - Knight-Ruiz method (KR) in (64);
  - Negative entropy method (NE) in (85);
  - Logarithmic barrier method (LB) in (94).

By \( A_k \) defined in (93), the performance metric is the sum of 1-norm

\[
\text{Error} := \|A_k \mathbb{1}_n - \mathbb{1}_n\|_1 + \|A_k^T \mathbb{1}_n - \mathbb{1}_n\|_1.
\]  

(117)

- First, we start with the initial vector \( \mathbb{1}_n \) for \( r_1 \) and generate \( c_1 \) in (62). Typically, Sinkhorn-Knopp algorithm requires a lot of iterations for ill-conditioned \( A \). Direct performance comparison shows that all Newton based methods produce very accurate doubly stochastic matrices within a few iterations. In particular, KR converges quickly in the case \( n = 20 \) and \( n = 200 \), even though the global convergence of KR is unclear. See the left subfigure of Fig. 6. The winner between LB and NE methods depends on the matrix to be balanced. However, floating-point exponential functions are more expensive than floating-point multiplications and divisions. The method LB generally takes less CPU time than the method NE.

The case \( n = 50 \) is a challenging problem. Optimal scaling vectors \( r, c \) have norm both greater than \( 10^{12} \), since \( \exp(-\text{magic}(50)/20) \) nearly do not have total support. Under the circumstance, all Newton methods give relatively slow convergence, comparing with the other two balancing tasks. Note that KR still gives the fastest convergence in the perspective of CPU time.

- Second, we use 10 SK iterations to generate a warm start fed to three Newton methods, respectively. Typically, a warm start generated from a few Sinkhorn-Knopp iterations, once falling into the the attractive basin, can accelerate the convergence of these Newton based methods. See the right subfigure of Fig. 6.

- Third, we further examine other warm start choices. As in \( \epsilon \)-scaling method, we can employ \( \epsilon \)-scaling on \( A = \exp(-\text{magic}(50)/20) \). Let \( A' \) denote the component-wise exponential with \( \epsilon \in (0,1) \) close to 0. Fig. 7 demonstrates that four phases, where four matrix balancing methods are applied on \( A^{1/8}, A^{1/4}, A^{1/2} \) and \( A \), respectively. In phase 1, we use the scaling vector which balances \( A^{1/16} \) as a warm start to balance \( A^{1/8} \). Later, in phase \( k = 2, 3, 4 \), we use the scaling vector (from the
NE method) which balances $A^{(2^k-5)}$ as a warm start to balance $A^{(2^k-4)}$. For phase $k = 1, 2, 3, 4$, the geometric mean of the norm of the scaling vectors is

$$
\frac{\|r\|^{1/2}}{\|c\|^{1/2}} = 2.31 \times 10^1, \ 8.05 \times 10^2, \ 1.61 \times 10^6, \ 1.08 \times 10^{13},
$$

respectively.\(^1\) From Remark 3.6, the norm growth of scaling vectors indicates that the matrices to be balanced nearly do not have total support. Results show that the warm start from phase 3 speeds up Newton methods applied on $A$. Hence, compared with the second row of Fig. 6, optimal transport can be handled effectively by solving a sequence of subproblems with smaller parameter $t$ in SNNE. Lastly, for this balancing task, the LB method takes more CPU time than the NE method, mainly in selecting proper step size.

### 4.3 Rigid-motion estimation

One big advantage of SNNE over IPMB is that SNNE updates multiplier vectors solely along the increase of $t$, i.e., no need to store/pass $x$ between sub-problems. Thus, the memory requirement in SNNE can be

\(^1\)As one reference, $\|r\|^{1/2}/\|c\|^{1/2}$ is only 1.663 and 137.8 for two easy problems $A = \exp(-\text{magic}(20)/20)$ and $A = \exp(-\text{magic}(200)/20)$, respectively.
Figure 7: The performance metric in (117) evaluates the balancing of $\exp(-2^{k-4} \cdot \text{magic}(50)/20)$ with a warm start from matrix balancing of $\exp(-2^{k-5} \cdot \text{magic}(50)/20)$ via NE method, where $k = 1, 2, 3, 4$.

much less than that in IPMB, when we face a large-scale problem. The following experiment demonstrates that SNNE is suitable in handling large-scale problems.

To demonstrate the capability in handling large point-sets, we present a rigid motion experiment on a three-dimensional teapot point cloud consisting of 41472 points. We subsample 1000/2500/5000 point-sets $\{y_1, \ldots, y_n\}$ from the teapot point cloud. For simplicity, $\{y_i\}$ is shifted so that $\sum_{i=1}^n y_i = 0$. Select one orthogonal matrix $Q \in \mathbb{R}^{3 \times 3}$, and generate another set of point-sets, $\{z_i = Qy_i : i = 1, \ldots, n\}$, as shown in Figure 8. Introducing a user-defined parameter $\xi > 0$, we estimate $Q \in \mathbb{R}^{3 \times 3}$ from the minimization

$$\min_Q \min_x \{ F(Q,x) := \langle c(Q), x \rangle + \xi \|Q - I\|_2 \} ,$$

where the assignment $c(Q)$ is a function of $Q$ with $T(\{c(Q)\})_{i,j} = \|y_i - Qz_j\|^2$. The algorithm consists of repeating the estimations of $Q$ and $x$:

- Fix $Q$. Use SNNE Alg. 2.9 to estimate $x$ with $T(x) \in \Pi_n$.
- Fix $x$. Solve $Q$ from the least squares problem,

$$\min_Q \{ F(Q,x) = \sum_{i,j=1}^n \langle (y_i - Qz_j), x_{i,j}(y_i - Qz_j) \rangle + \xi \langle Q - I, Q - I \rangle \} .$$

From the SVD property, an optimal matrix is $Q = UV^\top$, where $U, V$ are unitary matrices in the SVD,

$$UDV^\top = \sum_{i,j=1}^n \{ x_{i,j}y_i z_j^\top + \xi I \} .$$

Starting with $Q = I$ and an initial support constraint $\Sigma_0$, we employ SNNE-sparse, i.e., Alg. 2.9 to compute one approximate multiplier vector $\nu_\xi$ and $\Sigma_\xi$ for $\xi = 1, 2, 3, \ldots$. To have a better control on sparsity of $\Sigma_\xi$ in SNNE, as mentioned in Remark 2.10 we can select a sparse parameter $k = 20$ to ensure
an upper bound \((2k+1)n\) for the cardinality of \(\Sigma_e\). The performance metric is given by

\[
\text{error} := \langle \mathbf{c}(Q), x \rangle \geq 0.
\]  (122)

Note that \(\text{error} = 0\) if and only if \(y_i = Qz_j\) holds for all \(x_{i,j} > 0\). As reported in Fig. 8, SNNE reconstructs \(x\) and \(Q\) successfully under the aid of sparse support constraints in the cases \(n = 1000\), \(n = 2500\) and \(n = 5000\). As expected, when \(n\) increases, the computational time increases accordingly.

**Remark 4.1 (Multi-scale similarity).** Actually the multiplier vectors corresponding to different cardinality \(n\) resemble each other. For instance, consider the application of SNNE on the problem with \(n = 2500\) and \(n = 5000\), respectively, \(\min_{T(\mathbf{c}) \in \Pi_n} \langle \mathbf{c}(x), T(\mathbf{c})_{i,j} = \|y_i - z_j\|_2^2, i, j = 1, \ldots, n \rangle\). For the case \(n = 2500\), let \(Y' = \{y_1, \ldots, y_{2500}\}\) and \(Z' = \{z_1, \ldots, z_{2500}\}\). Let \([\nu^{(1)'}', \nu^{(2)'}']\) be the multiplier vector in (49). For the case \(n = 5000\), let \(Y = \{y_1, \ldots, y_{5000}\}\) and \(Z = \{z_1, \ldots, z_{5000}\}\). Let \([\nu^{(1)}, \nu^{(2)}]\) be the multiplier vector in (46). The color distribution in the top figures showing \((\nu^{(1)'}', \nu^{(2)'}')\) resembles the color distribution in the bottom figures showing \((\nu^{(1)}, \nu^{(2)})\). Indeed, \(\nu^{(1)} \approx \nu^{(1)'} + 160\) and \(\nu^{(2)} \approx \nu^{(2)'} - 160\). (Here the shift is caused by the one-dimension null space of \(M\).) Hence, we can employ \((\nu^{(1)'}', \nu^{(2)'}')\) to produce a warm start \((\nu^{(1)}, \nu^{(2)})\) (satisfying KKT conditions in (7)) to initialize \(\nu\) (which initializes \(\Sigma\)) in the problem with \(n = 5000\). That is,
Figure 10: Computational time vs. error metric in (122) under SNNE, SNNE-sparse and linprog.

- let \( \nu^{(1)}_{\text{ini}} \) be computed as follows: for \( j = 1, \ldots, 5000 \)
  \[
  \nu^{(1)}_{\text{ini}}(j) = \max_k \{ \| y_j - z_k \|^2 - \nu^{(2)}_{\text{ini}}(k) : z_k \in Z' \}. \tag{123}
  \]

- Let \( \nu^{(2)}_{\text{ini}} \) be computed as follows: for \( k = 1, \ldots, 5000 \)
  \[
  \nu^{(2)}_{\text{ini}}(k) = \max_j \{ \| y_j - z_k \|^2 - \nu^{(1)}_{\text{ini}}(j) : y_j \in Y \}. \tag{124}
  \]

Lastly, we provide one comparison between SNNE, SNNE-sparse and the command “linprog” in matlab, which uses a primal-dual method to solve linear programming. The CPU time comparison is reported in Fig. 10. We test these algorithms on 500 points and 800 points sampled from the teapoint point set. Apply the rigid-motion estimation procedure. The matlab command “linprog” performs the best in the case \( n = 500 \) and solves linear programming in the case \( n = 800 \) within 6 minutes. However, due to the memory limitation, “linprog” cannot handle point sets, when the cardinality exceeds 1000. With the aid of sparse support, SNNE-sparse \((k = 20)\) yields faster convergence than SNNE. As shown in Fig. 9, SNNE-sparse still can be applied to point-sets with cardinality 5000.

4.4 Discussion

The assignment problem discussed in the paper can be handled by other methods, e.g., the dual simplex method. See chapter 4.5 in [LY16]. According to our numerical experiences, one powerful commercial software Gurobi can use the dual simplex method to reach a solution to point-sets matching problems within a minute, if the cardinality is less than 500. (For the case with cardinality larger than 2500, the memory shortage can still be a problem to Gurobi.) Since the software is operated on a different computing platform, we do not include these comparison experiments.

In this paper, we study interior point methods, IPMB and SNNE incorporated with matrix balancing, early termination and sparse support constraints, to solve point-set matching problems. Matrix balancing is introduced to overcome the deviation caused by the rank reduction of Schur complement matrices in Newton step computation, as shown in Fig. 2. In addition, with the aid of support constraints with total support, we can further reduce the storage demand in interior point methods, so that large-scale problems can be handled properly. Both IPMB and SNNE have their pros and cons. In Figure 4, IPMB does perform slightly better than SNNE in the medium-size problems. However, SNNE has a big advantage: SNNE solely updates multiplier vectors along the increase of \( t \), i.e., no need to store/pass \( x \) between each sub-problem. From the viewpoint of implementation, SNNE-sparse is a relatively convenient tool in solving large-scale point-set matching problems, even though the complexity analysis on SNNE is unclear.

26
Figure 11: Top: the point sets $Y', Z'$ with $n = 2500$, respectively. Bottom: the point sets $Y, Z$ with $n = 5000$, respectively. The color on $Y, Y'$ illustrates the values $\nu^{(1)}$ and $\nu^{(1)'}$. The color on $Z, Z'$ illustrates the values $\nu^{(2)}$ and $\nu^{(2)'}$.

4.5 Data availability

The teapot dataset can be retrieved from the matlab 3-D point cloud file, “pcread(’teapot.ply’)”. The lung branch points of subject H6012 is available from the corresponding author upon request.

4.6 Acknowledgements

The authors are grateful to the referees for their very careful reading. Their comments and suggestions have led to significant improvements of the presentation of the paper.

A Appendix

A.1 Computation of $d_k$ and $\nu_k$

Before we solve (18), note that without any support constraint, the null space of $M\text{diag}(x)^2M^\top$ has dimension 1.

Proposition A.1. Consider a positive vector $x \in \mathbb{R}^{n^2}$ and a matrix $M$ in [I]. Then $M\text{diag}(x)$ has rank $2n - 1$ and

$$\text{null}(M\text{diag}(x)^2M^\top) = \text{null}(M^\top) = \text{span}\{[1_n; -1_n]\}.$$  \hfill (125)

In addition, for each $r \in \mathbb{R}^{n^2}$ and $x \in \Gamma^{-1}(\Pi_n)$, the system

$$M\text{diag}(x^2)M^\top u = M\text{diag}(x^2)r$$  \hfill (126)

is consistent.

Proof. Suppose $M\text{diag}(x^2)M^\top u = 0$ for some $u \in \mathbb{R}^{2n}$. Then

$$0 = \langle u, M\text{diag}(x)^2M^\top u \rangle = \|\text{diag}(x)M^\top u\|^2$$  \hfill (127)
implies \( \text{diag}(x)M^\top u = 0 \), i.e., \( M^\top u = 0 \). Hence, \( \text{null}(M\text{diag}(x)2M^\top) \subseteq \text{null}(M^\top) \). Besides, write \( u = [v; w] \) with some vectors \( v \in \mathbb{R}^n \) and \( w \in \mathbb{R}^n \). Since \( M^\top u = 0 = \mathbf{1}_n w^\top + v\mathbf{1}_n \), then \( u_i + w_j = 0 \) for all \( i, j = 1, \ldots, n \), i.e., \( u_i = u_i = -w_j \) for all \( i, j \). This establishes

\[
\text{null}(M\text{diag}(x)2M^\top) \subseteq \text{null}(M^\top) \subseteq \text{span}\{[1_n; -1_n]\}.
\]

On the other hand, consider a vector in the form \( u = c[1_n; -1_n] \) with \( c \in \mathbb{R} \). Then \( M^\top u = c(1_n2_n^\top - 1_n1_n^\top) = 0 \) and \( u \in \text{null}(M\text{diag}(x)2M^\top) \). This completes the proof of the first part. Finally, note that (126) is the associated normal equation to the least squares problem

\[
\min_u \|\text{diag}(x)M^\top u - \text{diag}(x)v\|^2.
\] (128)

Hence, (126) is consistent.

The following gives the computation of \( d_k \) for (18). As pointed out in (42), the optimal condition is that \( 1_{2n} \odot (c \odot x - t^{-1}1_{2n}) \) is orthogonal to the null space of \( M\text{diag}(x)1_{2n} \), when restricted to \( \Sigma \). Prop. A.2 states the reduction of the null-space component of \( 1_{2n} \odot (c \odot x - t^{-1}) \) in the Newton direction \( d_k \) in (18). Notice that the entries in \( d_k \) are zero outside the support \( \Sigma \) and thus \( x_{k+1} \) satisfies the support constraint \( \Sigma \) automatically. The proof can be given by the direct computation. Empirically, we shall use conjugate gradient method described in section 2.4.1 to implement (130) and (134) to update \( d_k \) and \( \nu_{k+1} \).

**Proposition A.2.** For each \( x_k \), introduce the projection on the range of \( \text{diag}(x_k)M^\top \),

\[
P_k := \text{diag}(x_k)M^\top(M\text{diag}(x_k^2)M^\top)\text{diag}(x_k)).
\] (129)

The Newton direction \( d_k \) in (18) is determined by the projection of \( c \odot x_k - t^{-1}1_{2n} \) on the null space of \( M\text{diag}(x_k) \), i.e., for entries inside \( \text{supp}(d_k) \subset \text{supp}(x_k) \), we have

\[
t^{-1}d_k = -x_k \odot \{I - P_k\}(c \odot x_k - t^{-1}1_{2n})
\] (130)

\[
= -\text{diag}(x_k)^2(c - t^{-1}x_k^{-1}) + \text{diag}(x_k)^2M^\top(M\text{diag}(x_k^2)M^\top)\text{diag}(x_k)^2(c - t^{-1}x_k^{-1}).
\] (131)

**Proof.** From (19,20), the Newton direction \( d_k \) can be determined from

\[
t^{-1}x_k^{-1} \odot d_k = x_k \odot r_k - x_k \odot M^\top(M\text{diag}(x_k^2)M^\top)\text{diag}(x_k^2)\odot r_k
\] (132)

\[
= (I - P_k)(x_k \odot r_k) = -(I - P_k)(c_k \odot x_k - t^{-1}1_{2n}),
\] (133)

where we drop the term \( x_k \odot (M^\top \nu_k) \), since it lies in the null space of \( I - P_k \).

**Remark A.3.** Here we illustrate the computation of \( \nu_{k+1} \). To avoid the accumulated rounding error in \( \nu_{k+1} \) from (17), we compute \( \nu_{k+1} \) directly from the least squares estimate

\[
\min_{\nu} \|c \odot x_k - t^{-1}1_{2n} - \text{diag}(x_k)M^\top \nu\|^2 = (M\text{diag}(x_k^2)M^\top)^\dagger M\text{diag}(x_k)^2(cx_k - t^{-1}1_{2n}).
\] (134)

Notice that the expression in (134) is consistent with (13) in the case of step size \( \alpha = 1 \). Indeed, from (20),

\[
M\text{diag}(x_k^2)M^\top y_k = M\text{diag}(x_k^2)(c - (x_k t)^{-1} - M^\top \nu_k).
\] (135)

When the step size \( \alpha = 1 \) and \( r_{k+1} = y_k + y_k \), we have the result identical to (134),

\[
M\text{diag}(x_k^2)M^\top \nu_{k+1} = M\text{diag}(x_k^2)(c - (x_k t)^{-1}).
\] (136)

**Remark A.4.** One common stopping criterion of Newton’s method is that the squared Newton decrement is sufficiently close to zero. The squared Newton decrement for the equality constrained problem in (19) is

\[
-\frac{d}{ds}f(x_k + sd_k)|_{s=0} = -(r_k, (\nabla^2 f)^\dagger (r_k)) = \|c \odot x_k - t^{-1}1_{2n} - \text{diag}(x_k)M^\top \nu_k\|^2.
\] (137)
Once $x^{(k)}$ is close to $x^{(t)}$, convergence to $x^{(t)}$ can be extremely rapid. The convergence analysis with finite $t$ can be found in section 10.2 [BV02].

### A.2 Early termination

The following rounding procedure could quickly provide a KKT candidate point before the degeneracy of Schur complement matrices occurs. Consider the case, where the optimal solution to (3) exists uniquely, $x \in \Pi_n$. Suppose that one diagonal in $\mathbb{T}(x^{(t)})$ dominates other diagonals for some $t$. Then we have early termination of the interior point method, i.e., a permutation matrix can be identified as one optimal solution from $\mathbb{T}(x^{(t)})$. For simplicity, the following discussion does not involve support constraints.

**Proposition A.5.** Let $\gamma' \in (0, 1)$ and $\gamma'' \in (1, \infty)$. Let $(x^{(t)}, \nu^{(t)})$ be one approximate KKT point to (12) for some $t > 0$ with (29). Let $X := \mathbb{T}(x^{(t)})$. Suppose that for some permutation $\mathcal{J} : \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$,

\[
X_{i,j} \leq \frac{\gamma'}{\gamma} X_{i,\mathcal{J}(i)} \text{ for all } j \neq \mathcal{J}(i),
\]

(138)

Let $\nu := [\nu^{(1)}; \nu^{(2)}] \in \mathbb{R}^{2n}$ be given by

\[
\nu^{(1)}(i) := c_{i,\mathcal{J}(i)} - \nu^{(2)}(\mathcal{J}(i)), \quad \nu^{(2)}(j) := \nu^{(2)}(j).
\]

(139)

Let $\tilde{X}$ be one permutation

\[
\tilde{X}_{i,\mathcal{J}(i)} = 1 \text{ and } \tilde{X}_{i,j} = 0, \quad j \neq \mathcal{J}(i),
\]

(140)

and $\tilde{x} := \mathbb{T}^{-1}(\tilde{X})$. Then $(\tilde{x}, \nu)$ is one KKT point to (7).

**Proof.** The KKT condition in (29) ensures that for all $i, j = 1, \ldots, n$,

\[
\epsilon_{i,j} := tX_{i,j}(c_{i,j} - (M^T\nu_{i,j})) \in (\gamma', \gamma'').
\]

(141)

In particular, for $j \neq \mathcal{J}(i)$,

\[
c_{i,\mathcal{J}(i)} - \nu^{(1)}(i) - \nu^{(2)}(\mathcal{J}(i)) \leq \gamma'' (tX_{i,\mathcal{J}(i)})^{-1}.
\]

(142)

We shall prove that (1) holds under this $\nu$. Let $s := -M^T\nu$. From the definition in (139), it suffices to show $s_{i,j} \geq 0$ for all entries with $j \neq \mathcal{J}(i)$. From (151) and (139), we have

\[
s_{i,j} := c_{i,j} - (M^T\nu)_{i,j} = c_{i,j} - \nu^{(1)}(i) - \nu^{(2)}(j)
\]

(143)

\[
\geq c_{i,j} - \nu^{(2)}(\mathcal{J}(i)) + \nu^{(2)}(\mathcal{J}(i)) - \nu^{(2)}(j) - \nu^{(1)}(i) + \nu^{(1)}(i)
\]

(144)

\[
\geq (tX_{i,j})^{-1}\epsilon_{i,j} - \gamma'' X_{i,j} / (tX_{i,\mathcal{J}(i)})^{-1}
\]

(145)

\[
\geq (tX_{i,j})^{-1}(\epsilon_{i,j} - \gamma') \geq 0,
\]

(146)

where we used the assumption in (138) and (142). \qed

For some degenerate cases of $c$, an optimal solution can be a non-permutation. The computed matrix $X := \mathbb{T}(x^{(t)})$ could be far away from a permutation for every $t$. Under this circumstance, we employ the following procedure to reach a KKT solution. Note that from (29), $X_{i,j}$ is approximately $t(c_{i,j} - \nu^{(2)}(\mathcal{J}(i)) - \nu^{(2)}(j))$. We can divide all the indices $(i, j)$ into two classes $\mathcal{K}$ and $\mathcal{K}^c$, depending on whether $X_{i,j} > \epsilon$ holds or not.

**Proposition A.6.** Let $\gamma' \in (0, 1)$ and $\gamma'' \in (1, \infty)$. Let $(x^{(t)}, \nu^{(t)})$ be one approximate KKT point to (12) for some $t > 0$ with (29). Let $X := \mathbb{T}(x^{(t)})$. Fix some $\epsilon \in (0, 1/n)$, and let $\mathcal{K}$ and $\mathcal{K}^c$ denote the index sets,

\[
\mathcal{K} := \{(i, j) : X_{i,j} > \epsilon\}, \quad \mathcal{K}^c := \{(i, j) : X_{i,j} \leq \epsilon\}.
\]

(147)

Let $\mathcal{J}$ be one permutation such that the relation holds for each $(i, j) \in \mathcal{K}^c$,

\[
X_{i,j}(X_{i,\mathcal{J}(i)}^{-1} + X_{\mathcal{J}^{-1}(j),j}^{-1}) \leq (\gamma'')^{-1}\gamma',
\]

(148)
If $J$ exists, then let $\hat{x}$ be the rounded vector $\hat{x} = x^{(t)} \odot (x^{(t)} > c)$ and $\delta := [\delta^{(1)}; \delta^{(2)}]$ be the solution to the least squares problem,

$$
\arg \min_{\delta} \| \text{diag}(\hat{x}) M^T \delta - \hat{x} \odot (c - M^T \nu_t) \|^2 = (M \text{diag}(\hat{x}^2) M^T)^\dagger M \text{diag}(\hat{x}^2) (c - M^T \nu_t).
$$

(149)

In addition suppose for $(i,j) \in K$,

$$
c_i, j - \nu^{(1)}(i) - \nu^{(2)}(j) = \delta^{(1)}(i) + \delta^{(2)}(j)
$$

(150)

holds. Let $\nu := [\nu^{(1)}; \nu^{(2)}] = \nu_t + \delta$. Then $(\hat{x}', \nu)$ is one KKT point to $\overline{J}$, where $\hat{x}'$ is the application of matrix balancing on $\hat{x}$.

**Proof.** We shall prove that $\overline{J}$ holds under this $\nu$. Let $s := c - M^T \nu$. From the definition in (139), it suffices to show $s_{i,j} = c_{i,j} - \nu^{(1)}(i) - \nu^{(2)}(j) > 0$ for all entries with $(i,j) \in K^c$.

The KKT condition in (29) ensures that for all $i,j = 1, \ldots, n$,

$$
e_{i,j} := t X_{i,j} (c_{i,j} - (M^T \nu_t)_{i,j}) \in (\gamma', \gamma'').
$$

(151)

Hence,

$$
\delta^{(1)}(i) + \delta^{(2)}(J(i)) \leq \gamma''(t X_{i,J(i)})^{-1}, \quad \delta^{(1)}(J^{-1}(j)) + \delta^{(2)}(j) \leq \gamma''(t X_{J^{-1}(j),j})^{-1}.
$$

(152)

From (151) and (139), we have for $(i,j) \notin K$,

$$
s_{i,j} := c_{i,j} - (M^T \nu)_i = c_{i,j} - \nu^{(1)}(i) - \nu^{(2)}(j)
$$

(153)

$$
= c_{i,j} - \delta^{(1)}(i) - \delta^{(2)}(j) - \nu^{(1)}(i) - \nu^{(2)}(j)
$$

(154)

$$
- \delta^{(2)}(J(i)) - \delta^{(1)}(J^{-1}(j)) + \delta^{(2)}(J(i)) + \delta^{(1)}(J^{-1}(j))
$$

(155)

$$
\geq (t X_{i,j})^{-1} \left\{ \epsilon_{i,j} - \gamma'' \frac{X_{i,j}}{X_{i,J(i)}} - \gamma'' \frac{X_{i,j}}{X_{J^{-1}(j),j}} + \gamma' \frac{X_{i,j}}{X_{J^{-1}(j),j}} \right\}
$$

(156)

$$
\geq (t X_{i,j})^{-1} (\epsilon_{i,j} - \gamma') \geq 0,
$$

(157)

where we used the assumption in (148).

**Remark A.7.** In summary, fixing $\epsilon < 1/n$, we can determine $K$ and $K^c$ in (147). Early termination in the degenerate case is ensured under two conditions: (1) Determine one diagonal $J$ from $K$ so that (148) holds for all entries in $K^c$. Identification of $J$ can be complicated in some cases. In empirical studies, we can employ one simpler condition: For each $(i,j) \in K^c$,

$$
X_{i,j} (X_{i,j}^{-1}) \leq (2\gamma'')^{-1} \gamma', \quad X_{j,i} (X_{j,i}^{-1}) \leq (2\gamma'')^{-1} \gamma'.
$$

(158)

holds for all $(i,j') \in K$ and $(i',j) \in K$. In short, the masses in $K$ outweigh the masses in $K^c$ with a factor $2\gamma''/\gamma'$. (2) Compute $\delta$ from (149) and examine that (150) holds (or within a small tolerance) for all entries in $K$.

**References**

[AZLOW17] Zeyuan Allen-Zhu, Yuanzhi Li, R. Oliveira, and A. Wigderson. Much faster algorithms for matrix scaling. 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pages 890–901, 2017.

[Bac70] M. Bacharach. Biproportional Matrices and Input-Output Change. Cambridge University Department of Applied Economics. 16 Monographs. Cambridge University Press, 1970.

[BCC+15] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. SIAM Journal on Scientific Computing, 37(2):A1111–A1138, 2015.

[BCLW17] C. Brauer, Christian Clason, Dirk A. Lorenz, and Benedikt Wirth. A sinkhorn-newton method for entropic optimal transport. arXiv: Optimization and Control, 2017.
Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K. Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017.

Philip A. Knight and Daniel Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 33(3):1029–1047, 2012.

Johan Karlsson and Axel Ringh. Generalized sinkhorn iterations for regularizing inverse problems using optimal mass transport. *SIAM J. Imaging Sci.*, 10(4):1935–1962, 2017.

Paul Knopp and Richard Sinkhorn. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343 – 348, 1967.

David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer International Publishing, 2016.

Albert W. Marshall and Ingram Olkin. Scaling of matrices to achieve specified row and column sums. *Numerische Mathematik*, 12(1):83–90, 1968.

O. Museyko, M. Stiglmayr, K. Klamroth, and G. Leugering. On the application of the Monge-Kantorovich problem to image registration. *SIAM J. Imaging Sciences*, 2(4):1068–1097, 2009.

J.B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1 – 36, 1998.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

J. Rabin, J. Delon, and Y. Gousseau. A statistical approach to the matching of local features. *SIAM J. Imaging sciences*, 2:931–958, 2009.

Robert Robere. Interior point methods and linear programming. 2012.

Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vis.*, 40(2):99–121, 2000.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.

Bernhard Schmitzer and Christoph Schnörr. Globally optimal joint image segmentation and shape matching based on wasserstein modes. *Journal of Mathematical Imaging and Vision*, 52(3):436–458, 2015.

C. Villani. *Topics in Optimal Transportation*. Graduate Studies in Mathematics, AMS, 2003.

C. Villani. *Optimal transport: Old and New*. Springer Verlag (Grundlehren der mathematischen Wissenschaften), 2008.

M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multi-dimensional histograms. *Comp. Vis. Graphics Image Proc.*, 32:328–336, 1985.

Stephen J. Wright. *Primal-dual interior-point methods*. Society for Industrial and Applied Mathematics, 1997.
[YLST21] Lei Yang, Jia Li, Defeng Sun, and Kim-Chuan Toh. A fast globally linearly convergent algorithm for the computation of wasserstein barycenters. *J. Mach. Learn. Res.*, 22:21:1–21:37, 2021.

[ZF03] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vis. Compu.*, 21:977–1000, 2003.

[ZYHT07] L. Zhu, Y. Yang, S. Haker, and A. Tannenbaum. An image morphing technique based on optimal mass preserving mapping. *IEEE Image Processing*, 16(6):1481 – 1495, 2007.