Optimal Transport with Cyclic Symmetry

Shoichiro Takeda, Yasunori Akagi, Naoki Marumo, Kenta Niwa

NTT Corporation, 1-1 Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847, Japan
{shoichiro.takeda, yasunori.akagi, naoki.marumo, kenta.niwa}@ntt.com

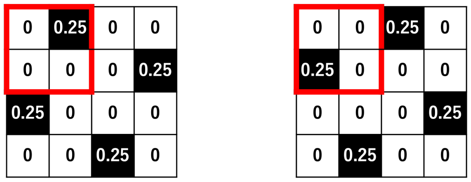
Abstract

We propose novel fast algorithms for optimal transport (OT) utilizing a cyclic symmetry structure of input data. Such OT with cyclic symmetry appears universally in various real-world examples: image processing, urban planning, and graph processing. Our main idea is to reduce OT to a small optimization problem that has significantly fewer variables by utilizing cyclic symmetry and various optimization techniques. On the basis of this reduction, our algorithms solve the small optimization problem instead of the original OT. As a result, our algorithms obtain the optimal solution and the objective function value of the original OT faster than solving the original OT directly. In this paper, our focus is on two crucial OT formulations: the linear programming OT (LOT) and the strongly convex-regularized OT, which includes the well-known entropy-regularized OT (EROT). Experiments show the effectiveness of our algorithms for LOT and EROT in synthetic/real-world data that has a strict/approximate cyclic symmetry structure. Through theoretical and experimental results, this paper successfully introduces the concept of symmetry into the OT research field for the first time.

Introduction

Given two probability vectors and a cost matrix, the discrete optimal transport (OT) problem seeks an optimal solution to minimize the cost of transporting the probability vector toward another one. Its total transportation cost is an effective tool that compares two probability vectors. Therefore, OT has been studied in various research areas, e.g., text embedding (Kusner et al. 2015), image matching (Liu et al. 2020), domain adaptation (Courty et al. 2017), graph comparison (Nikolentzos, Meladianos, and Vazirgiannis 2017), and interpolation (Solomon et al. 2015).

There are many formulations for OT. Kantorovich (1942) was the first to formulate OT as the linear programming problem, and the linear OT (LOT) made great progress toward solving OT. Recently, the strongly convex-regularized OT (SROT) has attracted much attention, especially, the entropy-regularized OT (EROT) (Cuturi 2013; Blondel, Seguy, and Rolet 2018; Peyré and Cuturi 2019; Guo, Ho, and Jordan 2020). SROT is superior to LOT in terms of guaranteeing a unique solution and computational stability.



0	0.25	0	0
0	0	0	0.25
0.25	0	0	0
0	0	0.25	0

0	0	0.25	0
0.25	0	0	0
0	0	0	0.25
0	0.25	0	0

Figure 1: A simple counter-example to the intuitive approach for OT with cyclic symmetry. Given the above two gray images with 90° cyclic symmetry, it seems intuitively sufficient to consider OT for only one of the symmetric areas (i.e., the red rectangular areas). However, when the cost matrix is given by the pixel-wise Euclidean distance matrix, the $(0, 1)$-th entry in the left image is optimally transported toward the $(0, 2)$-th entry beyond the red rectangular area in the right image. Thus, such an intuitive approach gives an incorrect solution, so we must develop novel algorithms that utilize cyclic symmetry appropriately in OT.

Many algorithms have been studied to solve OT. The network simplex algorithm (Ahuja, Magnanti, and Orlin 1993) is a well-known classical algorithm for LOT and has been widely used. The Sinkhorn algorithm (Cuturi 2013) and primal-dual descent algorithms (Dvurechensky, Gasnikov, and Kroshnin 2018; Guo, Ho, and Jordan 2020) have been proposed to solve EROT faster. Recently, algorithms utilizing special structures of input data have been in the spotlight for solving OT faster, e.g., algorithms that utilize the low-rankness of the input data (Tenetov, Wolansky, and Kimmel 2018; Altschuler et al. 2019). Besides, several algorithms utilize the Gibbs kernel structure of the input cost matrix in the Sinkhorn algorithm, such as separability (Solomon et al. 2015; Bonneel, Peyré, and Cuturi 2016) and translation invariance (Getreuer 2013; Peyré and Cuturi 2019).

In this paper, we propose novel fast algorithms for OT utilizing a new special structure, *cyclic symmetry*, of input data. Specifically, we assume n-order cyclic symmetry for the input data; the input d-dimensional probability vector is a concatenation of n copies of an $m(:= d/n)$-dimensional vector, and the input $d \times d$ cost matrix is a block-circulant matrix consisting of n matrices with size $m \times m$ (see Assumption 1). Such OT with cyclic symmetry appears universally in various real-world examples: image processing,

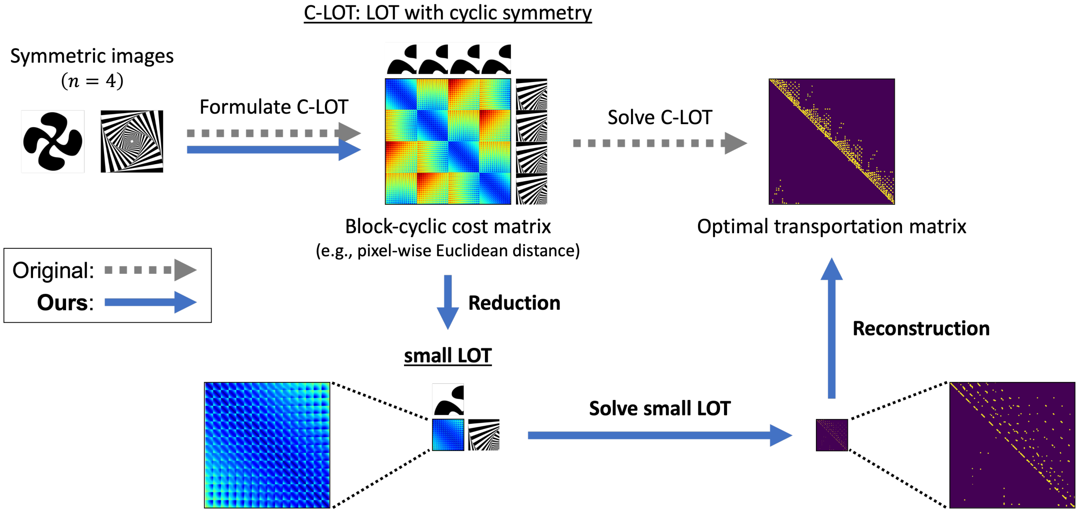


Figure 2: Overview of our algorithm for LOT with cyclic symmetry (C-LOT). This algorithm reduces C-LOT to a small LOT that has significantly fewer variables and solves the small LOT instead, resulting in fast computation. Note that the small cost matrix is not just a part of the original one; it aggregates the original cost matrix on the basis of cyclic symmetry, see (11).

urban planning, and graph processing (for details, see Examples 1 to 3). Intuitively, we can obtain an optimal solution to such a problem faster by solving OT for only one of the symmetric components of the input data and concatenating n copies of the obtained solution. However, this approach cannot work due to ignoring interactions between the symmetric components (see Figure 1). Unlike such an intuitive approach, we propose novel fast algorithms utilizing cyclic symmetry for two crucial OT formulations: LOT and SROT.

First, we propose a fast algorithm for LOT with cyclic symmetry (C-LOT). Figure 2 shows an overview of this algorithm. Our main idea is to reduce C-LOT, which has d^2 variables, to a small LOT, which has only m^2 variables, by utilizing cyclic symmetry. To achieve this reduction, we introduce auxiliary variables considering cyclic symmetry and rewrite C-LOT as a min-min optimization problem. Surprisingly, the inner min problem can be solved analytically, and the min-min problem becomes a small LOT. Therefore, this algorithm solves C-LOT faster by solving the small LOT instead. Using the network simplex algorithm to solve the small LOT, its time complexity bound becomes $O(m^3 \log m \log(m\|\mathbf{C}\|_\infty) + d^2)$ where $\mathbf{C}$ is the cost matrix and $\|\mathbf{C}\|_\infty := \max_{i,j}|C_{ij}|$. This is greatly improved from $O(d^3 \log d \log(d\|\mathbf{C}\|_\infty))$ when solving C-LOT directly.

Next, we propose a fast algorithm for SROT with cyclic symmetry (C-SROT). Unlike C-LOT, we cannot reduce C-SROT to a small SROT due to the regularizer. To overcome this issue, we consider the Fenchel dual of C-SROT. By utilizing cyclic symmetry, we show that the Fenchel dual problem has only $2m$ variables, which is significantly fewer than the $2d$ variables in the naive dual of C-SROT. Therefore, this algorithm solves the small Fenchel dual problem by the alternating minimization algorithm (Beck 2017, Chapter 14). Since the number of variables is very few, its time complexity for one iteration will be reduced, resulting in fast computation as a whole. Especially, this algorithm for EROT with cyclic symmetry (C-EROT), which is a subclass of C-SROT, becomes a Sinkhorn-like algorithm. We call it cyclic Sinkhorn algorithm. The interesting point is that the Gibbs kernel in the cyclic Sinkhorn algorithm differs from that in the original Sinkhorn algorithm and is designed by considering cyclic symmetry. Its time complexity bound of each iteration is $O(m^2)$, which is significantly improved from $O(d^2)$ when solving C-EROT by the original Sinkhorn algorithm.

Finally, we propose a two-stage Sinkhorn algorithm for C-EROT with *approximate* cyclic symmetry. In the real world, there are many cases where the input data exhibit only approximate cyclic symmetry due to slight noise and displacement. The cyclic Sinkhorn algorithm cannot be applied to such cases because strict cyclic symmetry of the input data is assumed. To overcome this issue, the two-stage Sinkhorn algorithm first runs the cyclic Sinkhorn algorithm to quickly obtain a strict symmetric solution. It then runs the original Sinkhorn algorithm to modify the solution. As a result, this algorithm obtains the optimal solution to C-EROT with approximate cyclic symmetry faster by utilizing cyclic symmetry at the first stage. In the Experiments section, we experimentally confirmed the fast computation of this algorithm when input data have approximate cyclic symmetry.

In summary, this paper introduces the concept of symmetry into the OT research field for the first time and proposes fast cyclic symmetry-aware algorithms that solve small optimization problems instead of the original OT. We validated the effectiveness of our algorithms in synthetic/real-world data with a strict/approximate cyclic symmetry structure.

Related Work

OT was initially formulated by (Monge 1781). Later (Kantorovich 1942) relaxed it as the linear programming problem, which permits splitting a mass from a single source to multiple targets. The linear OT (LOT) is easier to solve than Monge's form and has made great progress toward solving OT. To solve OT, many algorithms have been proposed. For example, the network simplex algorithm (Ahuja, Magnanti, and Orlin 1993) is one of the classical algorithms for LOT and has been widely used. Recently, algorithms have been proposed to solve OT faster by adding the entropy regularizer (Cuturi 2013; Altschuler, Niles-Weed, and Rigollet 2017; Lin, Ho, and Jordan 2019b; Alaya et al. 2019). The dual form of the entropy-regularized OT can be solved faster by the Sinkhorn algorithm that updates dual variables via matrix-vector products (Sinkhorn 1967). For further acceleration, many improvements to the Sinkhorn algorithm have been proposed. For example, (Altschuler, Niles-Weed, and Rigollet 2017), (Lin, Ho, and Jordan 2019b), and (Alaya et al. 2019) proposed using greedy, randomized, and safe-screening strategies, respectively, to efficiently update the dual variables. Primal-dual algorithms have received much attention (Dvurechensky, Gasnikov, and Kroshnin 2018; Lin, Ho, and Jordan 2019a; Guo, Ho, and Jordan 2020) because they report faster computation than the Sinkhorn algorithm and its variants but are rarely used in practice due to the difficulty of implementation (Pham et al. 2020). This paper focuses on the network simplex algorithm and Sinkhorn algorithm because they are widely used.

As another line of work to solve OT faster, utilizing special structures of input data has been well studied (Solomon et al. 2015; Bonneel, Peyré, and Cuturi 2016; Peyré and Cuturi 2019; Getreuer 2013; Tenetov, Wolansky, and Kimmel 2018; Altschuler et al. 2019). Inspired by the fact that geodesic distance matrices can be low-rank approximated (Shamai et al. 2015), a low-rank approximation for the cost matrix in OT was introduced to reduce the time complexity of the Sinkhorn algorithm (Tenetov, Wolansky, and Kimmel 2018; Altschuler et al. 2019). Several approaches have utilized the Gibbs kernel structures of the cost matrix appearing in the Sinkhorn algorithms. The key to these approaches is to approximate the calculation involving the Gibbs kernel; by utilizing separability (Solomon et al. 2015; Bonneel, Peyré, and Cuturi 2016) or translation invariant (Peyré and Cuturi 2019; Getreuer 2013) of the Gibbs kernel on a fixed uniform grid, the matrix-vector product in the Sinkhorn algorithm can be replaced with convolutions. Thus, it can be computed faster by, e.g., a fast Fourier transform. This paper introduces the utilization of a new special but ubiquitous structure, cyclic symmetry, in OT.

Preliminary

Notations

$\mathbb{R}_{\geq 0}$ denotes the set of non-negative real numbers. $\langle \cdot, \cdot \rangle$ denotes the inner product; that is, for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=0}^{d-1} x_i y_i$, and for matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times d}$, $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i,j=0}^{d-1} X_{ij} Y_{ij}$. The probability simplex is denoted as $\Delta^d := \{\mathbf{x} \in \mathbb{R}^d \mid \sum_{i=0}^{d-1} x_i = 1, x_i \geq 0\}$. $\mathbf{1}_d$ denotes the all-ones vector in $\mathbb{R}^d$. $\mathbf{0}_d$ denotes the all-zeros vector in $\mathbb{R}^d$. $\mathbf{I}_d$ denotes the $d \times d$ identity matrix. $\otimes$ denotes the Kronecker product.

Regularized Optimal Transport (ROT)

We define the regularized OT (ROT) that adds a convex regularizer to the linear OT (LOT) introduced by (Kantorovich 1942). Given two probability vectors $\mathbf{a}, \mathbf{b} \in \Delta^d$ and a cost matrix $\mathbf{C} \in \mathbb{R}_{\geq 0}^{d \times d}$, ROT can be defined as

$$\begin{aligned} &\min_{\mathbf{T} \in \mathbb{R}^{d \times d}} \langle \mathbf{C}, \mathbf{T} \rangle + \sum_{i,j=0}^{d-1} \phi(T_{ij}) \\ &\text{s.t.} \quad \mathbf{T}\mathbf{1}_d = \mathbf{a}, \quad \mathbf{T}^\top \mathbf{1}_d = \mathbf{b}, \end{aligned} \tag{1}$$

where $\mathbf{T}$ is called a transportation matrix and $\phi : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is a convex function, called a regularizer. We assume $\phi(x) = +\infty$ if $x < 0$; this assumption imposes the non-negative constraint on $\mathbf{T}$.

ROT (1) is a generalization of various OT formulations. For example, (1) leads to LOT when ϕ is given by

$$\phi(x) = \begin{cases} 0 & \text{if } x \geq 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{2}$$

Also, (1) leads to the strongly convex-regularized OT (SROT) when ϕ is a *strongly convex* function; a function ϕ is called strongly convex if $\phi - \frac{\mu}{2}\|\cdot\|$ is convex for some $\mu > 0$. As an important subclass of SROT, (1) leads to the entropy-regularized OT (EROT) introduced by (Cuturi 2013) when ϕ is given by

$$\phi(x) = \begin{cases} \lambda x(\log x - 1) & \text{if } x \geq 0, \\ +\infty & \text{otherwise,} \end{cases} \tag{3}$$

where $\lambda > 0$. Note that, unlike the standard OT formulation, (1) imposes the non-negative constraint on $\mathbf{T}$ via the regularizer $\phi(x)$ for two purposes: (i) to unify the description of LOT and SROT, and (ii) to describe the Fenchel dual of (8) concisely and intuitively (see later in Theorem 2).

C-ROT: ROT with Cyclic Symmetry

This section explains our assumption of cyclic symmetry for ROT (1) and real-world examples of this problem.

We assume that $\mathbf{a}, \mathbf{b}, \mathbf{C}$ in (1) have the following n-order cyclic symmetry.

Assumption 1. There exists a divisor n of d, and the probability vectors $\mathbf{a}, \mathbf{b}$ in (1) have a periodic structure:

$$\mathbf{a} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha} \\ \vdots \\ \boldsymbol{\alpha} \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\beta} \\ \vdots \\ \boldsymbol{\beta} \end{pmatrix}, \tag{4}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}_{\geq 0}^m$ and $m := \frac{d}{n}$ is an integer. Also, the cost matrix $\mathbf{C}$ in (1) has a block-circulant structure:

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_{n-1} \\ \mathbf{C}_{n-1} & \mathbf{C}_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{C}_1 \\ \mathbf{C}_1 & \cdots & \mathbf{C}_{n-1} & \mathbf{C}_0 \end{pmatrix}, \tag{5}$$

where $\mathbf{C}_0, \ldots, \mathbf{C}_{n-1} \in \mathbb{R}_{\geq 0}^{m \times m}$.

In this paper, we call ROT (1) with Assumption 1 *Cyclic ROT (C-ROT)*. This problem appears universally in various real-world examples given below.

Example 1 (Image with Cyclic Symmetry)**.** Cyclic symmetry in images has been a central image research topic. Especially, because image data are represented in a rectangle form, mirror or 90° rotational symmetry has been utilized for various tasks; mirror symmetry has been utilized for the face recognition (Zhao et al. 2003) and rendering (Wu, Rupprecht, and Vedaldi 2023), and 90° rotational symmetry in medical and galaxy images has been utilized for the image segmentation (Pang et al. 2022) and morphology prediction (Dieleman, Willett, and Dambre 2015). Thus, we here consider ROT between images with cyclic symmetry, $\mathbf{A}$ and $\mathbf{B} \in \mathbb{R}_{\geq 0}^{h \times w}$. For images with mirror symmetry, we assume mirror symmetry along the vertical axis;

$$A_{ij} = A_{i,w-j-1}, \quad B_{ij} = B_{i,w-j-1},$$

for $0 \leq i < h$ and $0 \leq j < w$. We vectorize these images by appropriately ordering pixels as follows:

$$\begin{aligned}
\mathbf{a} &= \left(A_{\pi(0)}, A_{\pi(1)}, \ldots, A_{\pi(hw-1)}\right)^\top, \qquad (6)\\
\mathbf{b} &= \left(B_{\pi(0)}, A_{\pi(1)}, \ldots, B_{\pi(hw-1)}\right)^\top,\\
\pi(k) &= \begin{cases} (k \bmod h, \lfloor k/h \rfloor) & 0 \leq k < \frac{hw}{2} \\ \left(k \bmod h, \frac{3w}{2} - \lfloor k/h \rfloor - 1\right) & \frac{hw}{2} \leq k < hw \end{cases}.
\end{aligned}$$

By defining $\mathbf{C}$ as the Manhattan, Euclidean, or Chebyshev distance matrix between pixel positions, $\mathbf{a}, \mathbf{b}, \mathbf{C}$ satisfy Assumption 1; thus, C-ROT for $n = 2$ will appear. Similarly, by appropriately ordering pixels for $\mathbf{a}, \mathbf{b}$ in the case of 90° rotational symmetry, C-ROT for $n = 4$ will appear.

Example 2 (Urban Planning with Cyclic Symmetry)**.** ROT has straightforward applications in logistics and economy (Kantorovich 1942; Guillaume 2012). Imagine a situation where planners design the structure of a city, this structure is simply given by two probability distributions: the distributions of residents $\mathbf{a}$ and services $\mathbf{b}$. In this context, the objective function value of ROT enables us to measure how close residents and services are and evaluate the city's efficiency. Several city structures, such as Howard's garden city (Howard 1965), assume that residents and services are equally located along cyclic symmetry to improve quality of life. In such structures, $\mathbf{a}, \mathbf{b}$ and $\mathbf{C}$, where C_{ij} is given by the Euclidean distance between each resident a_i and service b_j, satisfy Assumption 1; thus, C-ROT will appear.

Example 3 (Graph with Cyclic Symmetry)**.** Graphs are commonly used to model real-world data. For example, chemical molecules and crystal structures can be modeled using graphs (Bonchev 1991; Xie and Grossman 2018), and their graph structures often exhibit cyclic symmetry (Jaffé and Orchin 2002; Ladd 2014). To compare two graphs, computing their distance has been well-studied and OT-based approaches have been proposed (Nikolentzos, Meladianos, and Vazirgiannis 2017; Petric Maretic et al. 2019; Togninalli et al. 2019). We here consider ROT for computing a distance between two graphs, G_1 and G_2, whose structures exhibit cyclic symmetry. Following (Togninalli et al. 2019), we define $a_i = b_j = \frac{1}{d}$, where d is the number of nodes, to ensure the same amount of outgoing/incoming flow from/to a node and C_{ij} as the Hamming distance between the Weisfeiler-Lehman hash features of each node pair in G_1 and G_2, namely $C_{ij} = \mathrm{hamming}(\mathbf{f}_{1,i}, \mathbf{f}_{2,j})$ where $\mathbf{f}_{k,l}$ is the l-th node hash feature in G_k. In this situation, $\mathbf{a}, \mathbf{b}$, and $\mathbf{C}$ satisfy Assumption 1, and thus C-ROT for two graphs will appear.

Fast Algorithms for C-ROT

In this section, we propose fast algorithms for C-ROT.

Block-Cyclic Structure of Optimal Solution

We first show the following lemma.

Lemma 1. Under Assumption 1, there exists an optimal solution to (1) that has the following structure:

$$\mathbf{T} = \begin{pmatrix} \mathbf{T}_0 & \mathbf{T}_1 & \cdots & \mathbf{T}_{n-1} \\ \mathbf{T}_{n-1} & \mathbf{T}_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{T}_1 \\ \mathbf{T}_1 & \cdots & \mathbf{T}_{n-1} & \mathbf{T}_0 \end{pmatrix}, \qquad (7)$$

where $\mathbf{T}_0, \ldots, \mathbf{T}_{n-1} \in \mathbb{R}_{\geq 0}^{m \times m}$.

Proof. Let $\mathbf{T}'$ be an optimal solution to (1). We define $\mathbf{T}^*$ as follows:

$$\mathbf{T}^* := \frac{1}{n} \sum_{k=0}^{n-1} \mathbf{P}^k \mathbf{T}' \left(\mathbf{P}^k\right)^\top,$$

where

$$\mathbf{P}^k := \begin{pmatrix} \mathbf{0}_{n-1}^\top & 1 \\ \mathbf{I}_{n-1} & \mathbf{0}_{n-1} \end{pmatrix}^k \otimes \mathbf{I}_m$$

is the block-circulant permutation matrix.

First, we will show that $\mathbf{T}^*$ is a feasible solution to (1). $\mathbf{T}^*$ satisfies the constraints of row summation because

$$\begin{aligned}
\mathbf{T}^* \mathbf{1}_d &= \frac{1}{n} \sum_{k=0}^{n-1} \mathbf{P}^k \mathbf{T}' \left(\mathbf{P}^k\right)^\top \mathbf{1}_d = \frac{1}{n} \sum_{k=0}^{n-1} \mathbf{P}^k \mathbf{T}' \mathbf{1}_d \\
&= \frac{1}{n} \sum_{k=0}^{n-1} \mathbf{P}^k \mathbf{a} = \frac{1}{n} \sum_{k=0}^{n-1} \mathbf{a} = \mathbf{a}.
\end{aligned}$$

Similarly, we can show that $\mathbf{T}^*$ satisfies the constraints of column summation (i.e., $(\mathbf{T}^*)^\top \mathbf{1}_d = \mathbf{b}$). Thus, $\mathbf{T}^*$ is a feasible solution to (1).

Next, we will show that $\mathbf{T}^*$ is an optimal solution to (1). For this purpose, we check the objective function value of (1) when $\mathbf{T} = \mathbf{T}^*$. Let $f(\mathbf{T})$ be the objective function of (1), we get

$$\begin{aligned}
f(\mathbf{T}^*) &= f\left(\frac{1}{n} \sum_{k=0}^{n-1} \mathbf{P}^k \mathbf{T}' \left(\mathbf{P}^k\right)^\top\right) \\
&\leq \frac{1}{n} \sum_{k=0}^{n-1} f\left(\mathbf{P}^k \mathbf{T}' \left(\mathbf{P}^k\right)^\top\right) \\
&= \frac{1}{n} \sum_{k=0}^{n-1} f(\mathbf{T}') = f(\mathbf{T}').
\end{aligned}$$

Note that, we use the convexity of f and Jensen's inequality in the inequality relationship of the above equation. Thus, $\mathbf{T}^*$ is an optimal solution to (1).

Finally, for $l = 0, \dots, n-1$, we get

$$\begin{aligned}\mathbf{P}^l\mathbf{T}^*(\mathbf{P}^l)^\top &= \frac{1}{n}\sum_{k=0}^{n-1}\mathbf{P}^{k+l}\mathbf{T}'(\mathbf{P}^{k+l})^\top \\ &= \frac{1}{n}\sum_{k=0}^{n-1}\mathbf{P}^{k}\mathbf{T}'(\mathbf{P}^{k})^\top = \mathbf{T}^*.\end{aligned}$$

Therefore, $\mathbf{T}^*$ has a block-circulant structure of (7). □

From Assumption 1 and Lemma 1, $\mathbf{C}$ and $\mathbf{T}$ have the same block-circulant structure. Plugging (5) and (7) into C-ROT (1) yields the following optimization problem:

$$\begin{aligned}\min_{\mathbf{T}_0,\dots,\mathbf{T}_{n-1}\in\mathbb{R}^{m\times m}} &\sum_{k=0}^{n-1}\langle\mathbf{C}_k,\mathbf{T}_k\rangle + \sum_{k=0}^{n-1}\sum_{i,j=0}^{m-1}\phi(T_{ijk}) \\ \text{s.t.}\quad &\sum_{k=0}^{n-1}\mathbf{T}_k\mathbf{1}_m = \boldsymbol{\alpha}, \quad \sum_{k=0}^{n-1}\mathbf{T}_k^\top\mathbf{1}_m = \boldsymbol{\beta},\end{aligned} \tag{8}$$

where T_{ijk} is the (i,j)-th entry of $\mathbf{T}_k$. Note that the optimal objective function value of (8) is exactly $\frac{1}{n}$ of that of (1).

Fast Algorithm for C-LOT

We here propose a fast algorithm for cyclic LOT (C-LOT), which is the special case of C-ROT (1) where ϕ is given by (2). From (8), C-LOT (1) can be rewritten as

$$\begin{aligned}\min_{\mathbf{T}_0,\dots,\mathbf{T}_{n-1}\in\mathbb{R}_{\geq 0}^{m\times m}} &\sum_{k=0}^{n-1}\langle\mathbf{C}_k,\mathbf{T}_k\rangle \\ \text{s.t.}\quad &\sum_{k=0}^{n-1}\mathbf{T}_k\mathbf{1}_m = \boldsymbol{\alpha}, \quad \sum_{k=0}^{n-1}\mathbf{T}_k^\top\mathbf{1}_m = \boldsymbol{\beta}.\end{aligned} \tag{9}$$

By introducing auxiliary variables $\mathbf{S} := \sum_{k=0}^{n-1}\mathbf{T}_k$ and rewriting (9) for $\mathbf{S}$, we can show the following theorem.

Theorem 1. We consider a small LOT

$$\min_{\mathbf{S}\in\mathbb{R}_{\geq 0}^{m\times m}} \langle\mathbf{G},\mathbf{S}\rangle \quad \text{s.t.} \quad \mathbf{S}\mathbf{1}_m = \boldsymbol{\alpha}, \quad \mathbf{S}^\top\mathbf{1}_m = \boldsymbol{\beta}, \tag{10}$$

where

$$G_{ij} := \min_{0\leq k\leq n-1} C_{ijk}. \tag{11}$$

Let $\mathbf{S}^*$ be an optimal solution to (10). Then, $(\mathbf{T}_k^*)_{k=0,\dots,n-1}$ defined by

$$T_{ijk}^* = \begin{cases} S_{ij}^* & \text{if } k = \min\left(\operatorname*{argmin}_{0\leq k\leq n-1} C_{ijk}\right), \\ 0 & \text{otherwise}\end{cases} \tag{12}$$

is an optimal solution to (9). Also, the optimal objective function value of (9) is the same as that of (10).

Note that $\mathrm{argmin}_{0\leq k\leq n-1}C_{ijk}$ in (12) will return a set of indices if the same minimum value exists in several indices, and we can choose any one but the smallest index by $\min$.

Algorithm 1: Fast Algorithm for C-LOT

```
Require: $\mathbf{a}, \mathbf{b} \in \Delta^d$ and $\mathbf{C} \in \mathbb{R}_{\geq 0}^{d\times d}$ under Assumption 1.
1: Compute $\mathbf{G}$ whose entry is given by (11).
2: Compute the optimal solution $\mathbf{S}^*$ to (10).
3: for $i, j, k$ do
4:     Compute $T_{ijk}$ by the relationship (12)
5: end for
6: Compute $\mathbf{T}$ by Lemma 1 with $(T_{ijk})$
7: return $\mathbf{T}$
```

Proof. We fix $\mathbf{S} := \sum_{k=0}^{n-1}\mathbf{T}_k$ in (9). The matrix $\mathbf{S}$ satisfies $\mathbf{S}\mathbf{1}_m = \boldsymbol{\alpha}, \mathbf{S}^\top\mathbf{1}_m = \boldsymbol{\beta}$ and we can rewrite (9) as

$$\min_{\substack{\mathbf{S}\in\mathbb{R}_{\geq 0}^{m\times m},\\ \mathbf{S}\mathbf{1}_m=\boldsymbol{\alpha},\, \mathbf{S}^\top\mathbf{1}_m=\boldsymbol{\beta}}} \left(\min_{\substack{\mathbf{T}_0,\dots,\mathbf{T}_{n-1}\in\mathbb{R}_{\geq 0}^{m\times m},\\ \sum_{k=0}^{n-1}\mathbf{T}_k=\mathbf{S}}} \sum_{k=0}^{n-1}\langle\mathbf{C}_k,\mathbf{T}_k\rangle \right).$$

The inner problem is equivalent to individual problems for each (i,j)-th entry of $\mathbf{T}_0, \dots, \mathbf{T}_{n-1}$, that is we can consider

$$\min_{T_{ij0},\dots,T_{ij(n-1)}\in\mathbb{R}_{\geq 0}} \sum_{k=0}^{n-1} C_{ijk}T_{ijk} \quad \text{s.t.} \quad \sum_{k=0}^{n-1}T_{ijk} = S_{ij},$$

for each (i,j) independently. This problem can be solved analytically; the optimal solution is given by (12), and the optimal objective function value is $\langle\mathbf{G},\mathbf{S}\rangle$. Next, we solve the outer optimization problem for $\mathbf{S}$. Because $\mathbf{S} \in \mathbb{R}_{\geq 0}^{m\times m}, \mathbf{S}\mathbf{1}_m = \boldsymbol{\alpha}, \mathbf{S}^\top\mathbf{1}_m = \boldsymbol{\beta}$ and the objective function is $\langle\mathbf{G},\mathbf{S}\rangle$, this optimization problem is identical with (10). □

Theorem 1 indicates that C-LOT (1) can be reduced to the small LOT (10), which has significantly fewer m^2 variables than $d^2 = m^2n^2$ variables of the original C-LOT (1). Therefore, we will obtain the optimal solution to C-LOT (1) by solving the small LOT (10) instead. The proposed algorithm is summarized in Algorithm 1. Also, Figure 2 shows the overview of this algorithm.

We evaluate the time complexity of Algorithm 1. The time complexity depends on the algorithm to solve the small LOT (10). We here use the network simplex algorithm, the most popular algorithm to solve LOT, to evaluate the time complexity. Tarjan (1997) showed that the time complexity of the network simplex algorithm to solve LOT (1) with the regularizer (2) is $O(d^3\log d\log(d\|\mathbf{C}\|_\infty))$, where $\|\mathbf{C}\|_\infty := \max_{i,j}|C_{ij}|$. This enables the time complexity of line 2 in Algorithm 1 to be bounded by $O(m^3\log m\log(m\|\mathbf{C}\|_\infty))$. Because line 1 and lines 3–7 can be conducted in $O(d^2)$ time for inputting $\mathbf{C}$ and outputting $\mathbf{T}$ respectively, the total time complexity of Algorithm 1 is $O(m^3\log m\log(m\|\mathbf{C}\|_\infty) + d^2)$. This is significantly improved from $O(d^3\log d\log(d\|\mathbf{C}\|_\infty))$ when solving C-LOT (1) directly. Note that we can consider that inputting only $\mathbf{C}_0, \dots, \mathbf{C}_{n-1}$ (in line 1) and outputting only $\mathbf{T}_0, \dots, \mathbf{T}_{n-1}$ (in lines 3–7) are sufficient because $\mathbf{T}$ consists of $\mathbf{T}_0, \dots, \mathbf{T}_{n-1}$. In this case, those lines can be conducted in $O(m^2n)$ time, and thus the total time complexity of Algorithm 1 will be $O(m^3\log m\log(m\|\mathbf{C}\|_\infty) + m^2n)$.

Fast Algorithm for C-SROT

We propose a fast algorithm for cyclic SROT (C-SROT) which is the special case of C-ROT (1) where ϕ is a strongly convex regularizer. Note that because ϕ defined by (2) is not strongly convex, we cannot apply this algorithm to C-LOT.

The following theorem follows from Fenchel's duality theorem and optimality conditions in convex analysis (see, e.g., (Rockafellar 1970, Section 31)).

Theorem 2. The Fenchel dual of the problem (8) is

$$\max_{\mathbf{w},\mathbf{z}\in\mathbb{R}^m} \langle \mathbf{w}, \boldsymbol{\alpha}\rangle + \langle \mathbf{z}, \boldsymbol{\beta}\rangle - \sum_{k=0}^{n-1}\sum_{i,j=0}^{m-1} \phi^\star(w_i + z_j - C_{ijk}), \tag{13}$$

where $\phi^\star : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is the Fenchel conjugate of ϕ defined by $\phi^\star(y) := \sup\{yx - \phi(x) \mid x \in \mathbb{R}\}$. Also, the optimal solutions to the problem (8), $\mathbf{T}_k^*$, and to the problem (13), $\mathbf{w}^*$ and $\mathbf{z}^*$, have the following relationship:

$$T_{ijk}^* = (\phi^\star)'(w_i^* + z_j^* - C_{ijk}). \tag{14}$$

Proof. We rewrite (8) with Lagrange multipliers $\mathbf{w}$ and $\mathbf{z}$ for the two equality constraints as follows:

$$\min_{\mathbf{T}_0,\ldots,\mathbf{T}_{n-1}\in\mathbb{R}^{m\times m}} \max_{\mathbf{w},\mathbf{z}\in\mathbb{R}^m} L(\mathbf{T}_0,\ldots,\mathbf{T}_{n-1},\mathbf{w},\mathbf{z}) \tag{15}$$

where

$$\begin{aligned}
&L(\mathbf{T}_0,\ldots,\mathbf{T}_{n-1},\mathbf{w},\mathbf{z})\\
&:= \sum_{k=0}^{n-1}\langle \mathbf{C}_k, \mathbf{T}_k\rangle + \sum_{k=0}^{n-1}\sum_{i,j=0}^{m-1}\phi(T_{ijk})\\
&\quad + \left\langle \mathbf{w}, \boldsymbol{\alpha} - \sum_{k=0}^{n-1}\mathbf{T}_k\mathbf{1}_m\right\rangle + \left\langle \mathbf{z}, \boldsymbol{\beta} - \sum_{k=0}^{n-1}\mathbf{T}_k^\top\mathbf{1}_m\right\rangle\\
&= \langle \mathbf{w}, \boldsymbol{\alpha}\rangle + \langle \mathbf{z}, \boldsymbol{\beta}\rangle\\
&\quad + \sum_{k=0}^{n-1}\big\langle \mathbf{C}_k - \mathbf{w}\mathbf{1}^\top - \mathbf{1}\mathbf{z}^\top, \mathbf{T}_k\big\rangle + \sum_{k=0}^{n-1}\sum_{i,j=0}^{m-1}\phi(T_{ijk}).
\end{aligned}$$

Note that Problem (8) is convex and the constraints are linear and that Slater's constraint qualification holds. Hence, the strong duality holds (see, e.g., (Boyd and Vandenberghe 2004, Section 5.2.3)), and we can swap the $\min$- and $\max$-operations in (15):

$$\begin{aligned}
&\max_{\mathbf{w},\mathbf{z}\in\mathbb{R}^m}\min_{\mathbf{T}_0,\ldots,\mathbf{T}_{n-1}\in\mathbb{R}^{m\times m}} L(\mathbf{T}_0,\ldots,\mathbf{T}_{n-1},\mathbf{w},\mathbf{z})\\
&= \max_{\mathbf{w},\mathbf{z}\in\mathbb{R}^m}\langle \mathbf{w}, \boldsymbol{\alpha}\rangle + \langle \mathbf{z}, \boldsymbol{\beta}\rangle\\
&\qquad - \sum_{k=0}^{n-1}\sum_{i,j=0}^{m-1}\phi^\star(w_i + z_j - C_{ijk}).
\end{aligned}$$

Thus, the Fenchel dual of the problem (8) is given by (13), and one of the optimality conditions is

$$T_{ijk}^* = (\phi^\star)'(w_i + z_j - C_{ijk}). \qquad \square$$

Note that $\phi^\star$ is a smooth and differentiable convex function because ϕ is strongly convex. Theorem 2 indicates that we will obtain the optimal solution to C-SROT (1) by solving the dual problem (13) instead because we can reconstruct it by the relationship (14) and Lemma 1.

We here propose to apply the alternating minimization algorithm (Beck 2017, Chapter 14) to (13); we iteratively optimize the objective function of (13) with respect to $\mathbf{w}$ while fixing $\mathbf{z}$, and vice versa. When we fix $\mathbf{z}$, the partial derivative of the objective function with respect to w_i is

$$\alpha_i - \sum_{k=0}^{n-1}\sum_{j=0}^{m-1}(\phi^\star)'(w_i + z_j - C_{ijk}), \tag{16}$$

and w_i is optimal if (16) equals to 0. Because (16) monotonically decreases with respect to w_i, we can find such w_i easily by, e.g., the well-known Newton's method. This also applies to the optimization with respect to $\mathbf{z}$ while fixing $\mathbf{w}$. The alternating minimization algorithm for a smooth convex function is guaranteed to attain fast convergence (see (Beck and Tetruashvili 2013) for more details).

The distinguishing feature of this algorithm is treating a few dual variables. If the alternating minimization algorithm is used for the dual problem of (1) without considering cyclic symmetry, the number of dual variables is $2d = 2mn$. In contrast, our algorithm treats only $2m$ dual variables, which is significantly reduced to $\frac{1}{n}$. Therefore, the computational time required for one iteration in the alternating minimization will be considerably reduced.

Fast Algorithm for C-EROT

We here propose a fast algorithm for cyclic EROT (C-EROT), which is the crucial special case of C-ROT (1) where ϕ is given by (3). Because (3) is strongly convex, we can apply the cyclic-aware alternating minimization algorithm introduced in the previous subsection to C-EROT.

Because $\phi^\star(y) = \lambda\exp(\frac{y}{\lambda})$, (16) can be written as

$$\alpha_i - \exp\Big(\frac{w_i}{\lambda}\Big)\sum_{j=0}^{m-1}K_{ij}\exp\Big(\frac{z_j}{\lambda}\Big), \tag{17}$$

where

$$K_{ij} := \sum_{k=0}^{n-1}\exp\Big(-\frac{C_{ijk}}{\lambda}\Big). \tag{18}$$

From (17), we can get optimal w_i in closed form:

$$w_i = \lambda\left(\log\alpha_i - \log\left(\sum_{j=0}^{m-1}K_{ij}\exp\Big(\frac{z_j}{\lambda}\Big)\right)\right). \tag{19}$$

We can rewrite (19) and describe the optimal q_j as follows:

$$p_i = \frac{\alpha_i}{\sum_{j=0}^{m-1}K_{ij}q_j}, \quad q_j = \frac{\beta_j}{\sum_{i=0}^{m-1}K_{ij}p_i},$$

where $p_i := \exp\big(\frac{w_i}{\lambda}\big)$, $q_j := \exp\big(\frac{z_j}{\lambda}\big)$. This algorithm resembles the Sinkhorn algorithm (Cuturi 2013); we call it

Algorithm 2: Cyclic Sinkhorn Algorithm for C-EROT

```
Require: $\mathbf{a}, \mathbf{b} \in \Delta^d$, $\mathbf{C} \in \mathbb{R}_{\geq 0}^{d \times d}$ under Assumption 1 and $\lambda > 0$.
 1: Compute $\mathbf{K}$ whose entry is given by (18).
 2: Initialize $\mathbf{q} \leftarrow \mathbf{1}_m$.
 3: repeat
 4:     $\mathbf{p} \leftarrow \boldsymbol{\alpha} \oslash (\mathbf{K}\mathbf{q})$    ▷ $\oslash$ denotes elementwise division
 5:     $\mathbf{q} \leftarrow \boldsymbol{\beta} \oslash (\mathbf{K}^\top \mathbf{p})$
 6: until convergence
 7: for $i, j, k$ do
 8:     $T_{ijk} \leftarrow p_i q_j \exp\left(-\frac{C_{ijk}}{\lambda}\right)$
 9: end for
10: Compute $\mathbf{T}$ by Lemma 1 with $(T_{ijk})$
11: return $\mathbf{T}$
```

cyclic Sinkhorn algorithm. Note that the optimal solution $\mathbf{T}$ to C-EROT (1) can be easily reconstructed from the optimal $\mathbf{w}$ and $\mathbf{z}$ by (14) and Lemma 1. The proposed algorithm is summarized in Algorithm 2.

We evaluate the time complexity of Algorithm 2. The time complexity depends on the matrix-vector product iterations in lines 4 and 5 in Algorithm 2 to solve the Fenchel dual problem (13). In the original Sinkhorn algorithm, the time complexity of each iteration is $O(d^2) = O(m^2 n^2)$ (Cuturi 2013). In contrast, in our cyclic Sinkhorn algorithm, the time complexity of each iteration is $O(m^2)$; thus, our algorithm solves C-EROT significantly faster than the original Sinkhorn algorithm.

Two-Stage Algorithm for C-EROT with Approximate Cyclic Symmetry

There are many real-world cases in which input data show only *approximate* cyclic symmetry. In Example 1, $\mathbf{C}$ satisfies Assumption 1 strictly when using the pixel-wise Euclidean distance, but input distributions $\mathbf{a}, \mathbf{b}$ (namely, images) often satisfy Assumption 1 only approximately due to slight noise and displacement. Thus, the above-proposed algorithms cannot be applied to such cases because they assume to satisfy Assumption 1 strictly. To overcome this issue, we here propose a fast *two-stage Sinkhorn algorithm* for C-EROT with approximate cyclic symmetry. Because EROT is commonly used thanks to its differentiability and computational efficiency (Peyré and Cuturi 2019; Guo, Ho, and Jordan 2020), we focused on C-EROT here. The two-stage Sinkhorn algorithm first runs the cyclic Sinkhorn algorithm (Algorithm 2) to quickly obtain a strict symmetric solution. It then runs the original Sinkhorn algorithm (Cuturi 2013) to modify the solution. Therefore, this algorithm obtains the optimal solution to C-EROT with approximate cyclic symmetry faster by utilizing cyclic symmetry at the first stage. The proposed algorithm is described in Algorithm 3.

If satisfying Assumption 1 strictly, the time complexity of this algorithm is the same as that of the cyclic Sinkhorn algorithm. If not, it will be complex due to mixing the two Sinkhorn algorithms at Stages 1 and 2. This analysis is for future research, but we experimentally confirmed that this algorithm shows fast computation when input data have approximate cyclic symmetry in the next section.

Algorithm 3: Two-Stage Sinkhorn Algorithm for C-EROT with Approximate Cyclic Symmetry

```
Require: $\mathbf{a}, \mathbf{b} \in \Delta^d$, $\mathbf{C} \in \mathbb{R}_{\geq 0}^{d \times d}$ and $\lambda > 0$.
    // Stage1: Cyclic Sinkhorn algorithm
for $i = 0, \ldots, m-1$ do
    $\alpha_i = \frac{1}{n} \sum_{k=0}^{n-1} a_{i+mk}$    ▷ the average of $n$-divided $\mathbf{a}$
    $\beta_i = \frac{1}{n} \sum_{k=0}^{n-1} b_{i+mk}$    ▷ the average of $n$-divided $\mathbf{b}$
end for
Compute $\mathbf{K}$ whose entry is given by (18).
Initialize $\widehat{\mathbf{q}} \leftarrow \mathbf{1}_m$.
repeat
    $\widehat{\mathbf{p}} \leftarrow \boldsymbol{\alpha} \oslash (\mathbf{K}\widehat{\mathbf{q}})$    ▷ $\oslash$ denotes elementwise division
    $\widehat{\mathbf{q}} \leftarrow \boldsymbol{\beta} \oslash (\mathbf{K}^\top \widehat{\mathbf{p}})$
until convergence
    // Stage2: Sinkhorn algorithm (Cuturi 2013)
Initialize $\mathbf{p}, \mathbf{q}$ as the $n$ concatenated $\widehat{\mathbf{p}}, \widehat{\mathbf{q}}$, respectively.
Compute $K_{ij} = \exp\left(-\frac{C_{ij}}{\lambda}\right)$.
repeat
    $\mathbf{p} \leftarrow \mathbf{a} \oslash (\mathbf{K}\mathbf{q})$
    $\mathbf{q} \leftarrow \mathbf{b} \oslash (\mathbf{K}^\top \mathbf{p})$
until convergence
return $\mathbf{T} \leftarrow \mathrm{diag}(\mathbf{p})\mathbf{K}\mathrm{diag}(\mathbf{q})$
```

Experiments

To validate the effectiveness of our algorithms, we conducted experiments on synthetic/real-world data that satisfy Assumption 1 strictly/approximately. In all experiments, we evaluated whether our algorithms, which solve small optimization problems instead of the original OT, show the same results as the original OT but with faster computation; for details, we checked the average and standard deviation of the objective function values, marginal constraint errors defined by $||\mathbf{T}^\top \mathbf{1}_d - \mathbf{b}||_2$, and the computation time when using different algorithms. The computation time was recorded between inputting the data and outputting the optimal solution. These experiments were performed on a Windows laptop with Intel Core i7-10750H CPU, 32 GB memory. All the codes were implemented in Python.

Synthetic Data with Strict Cyclic Symmetry

We created 20 synthetic random data for each of the two dimensions, $d \in \{5000, 10000\}$, that satisfy Assumption 1 strictly in $n = 50$; for details, we created $\mathbf{a}$ and $\mathbf{b}$ by concatenating n copies of m-dimensional uniform distribution with the half-open interval $[0.0, 1.0)$ like (4) and normalized $\mathbf{a}$ and $\mathbf{b}$ so that the sum of each is 1. For $\mathbf{C}$, we first sampled $\mathbf{C}_0, \ldots, \mathbf{C}_{n-1}$ by $m \times m$-dimensional Gaussian distribution with the mean 3.0 and the standard deviation 5.0. We then add the absolute minimum value, namely $|\min_{k=0,\ldots,n-1} (\min_{i,j=0,\ldots,m-1} C_{ijk})|$, to the all entries to ensure their non-negativity. After that, we created $\mathbf{C}$ by concatenating n copies of $\mathbf{C}_0, \ldots, \mathbf{C}_{n-1}$ like (5). For valida-

Algorithm	n	$d = 5000$			$d = 10000$		
		Obj. value	Marginal error	Time (sec.)	Obj. value	Marginal error	Time (sec.)
Network Simplex	–	6.034 ± 0.824	0.000 ± 0.000	6.523 ± 1.013	6.526 ± 0.917	0.000 ± 0.000	33.660 ± 3.238
Cyclic Network Simplex	2	6.034 ± 0.824	0.000 ± 0.000	1.477 ± 0.235	6.526 ± 0.917	0.000 ± 0.000	7.084 ± 0.728
	5	6.034 ± 0.824	0.000 ± 0.000	0.300 ± 0.030	6.526 ± 0.917	0.000 ± 0.000	1.391 ± 0.155
	10	6.034 ± 0.824	0.000 ± 0.000	0.136 ± 0.026	6.526 ± 0.917	0.000 ± 0.000	0.618 ± 0.073
	25	6.034 ± 0.824	0.000 ± 0.000	0.080 ± 0.019	6.526 ± 0.917	0.000 ± 0.000	0.381 ± 0.044
	50	6.034 ± 0.824	0.000 ± 0.000	0.056 ± 0.015	6.526 ± 0.917	0.000 ± 0.000	0.329 ± 0.034
Sinkhorn	–	6.233 ± 0.821	0.000 ± 0.000	3.271 ± 1.445	6.745 ± 0.916	0.000 ± 0.000	14.589 ± 4.213
Cyclic Sinkhorn	2	6.233 ± 0.821	0.000 ± 0.000	0.918 ± 0.463	6.745 ± 0.916	0.000 ± 0.000	3.973 ± 0.922
	5	6.233 ± 0.821	0.000 ± 0.000	0.207 ± 0.170	6.745 ± 0.916	0.000 ± 0.000	1.262 ± 0.324
	10	6.233 ± 0.821	0.000 ± 0.000	0.116 ± 0.036	6.745 ± 0.916	0.000 ± 0.000	0.636 ± 0.259
	25	6.233 ± 0.821	0.000 ± 0.000	0.093 ± 0.036	6.745 ± 0.916	0.000 ± 0.000	0.381 ± 0.127
	50	6.233 ± 0.821	0.000 ± 0.000	0.067 ± 0.034	6.745 ± 0.916	0.000 ± 0.000	0.320 ± 0.053

Table 1: Experimental results in the synthetic data. "Obj. value" indicates objective function value.

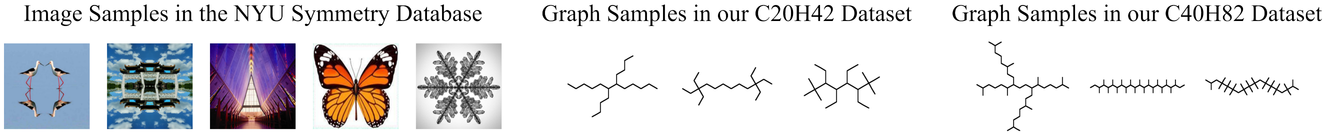


Image Samples in the NYU Symmetry Database
Graph Samples in our C20H42 Dataset
Graph Samples in our C40H82 Dataset

Figure 3: Real-world image and graph samples with cyclic symmetry ($n = 2$) that are used in the experiments of real-world cases. The above-shown chemical graph images are obtained by PubChem (Kim et al. 2022).

tion, we compared the network simplex algorithm (Ahuja, Magnanti, and Orlin 1993), Algorithm 1 using the network simplex algorithm in line 2 (we call it *cyclic network simplex algorithm*), the Sinkhorn algorithm (Cuturi 2013), and the cyclic Sinkhorn algorithm (Algorithm 2). We set $\lambda = 0.5$ for the regularizer (3). Because these synthetic data also satisfy Assumption 1 for all n that are divisors of 50, namely $n \in \{2, 5, 10, 25, 50\}$, we conducted experiments for each n; larger n leads to smaller problems that output the same result. The network simplex algorithm was implemented using LEMON (Dezső, Jüttner, and Kovács 2011).

Table 1 lists the results. The network simplex algorithm and cyclic one had the same optimal objective function values, but the latter showed faster computation times as n becomes larger. This was also the case when using the Sinkhorn algorithm and the cyclic one. Moreover, we experimentally confirmed that the marginal errors of the cyclic algorithms are 0. These results support our theoretically exact reformulations in Theorems 1 and 2 and the effectiveness of our proposed algorithms; higher cyclic symmetry (i.e., larger n) results in faster computation time.

Real-World Image Data with Approximate Cyclic Symmetry

We tested our algorithms on the real-world case of mirror symmetry ($n = 2$) in Example 1 with the NYU Symmetry Database (Cicconet et al. 2017). In this database, we selected 20 images with mirror symmetry along the vertical axis (see Figure 3). These images were converted to gray-scale, resized to be 64×64 or 96×96 pixels, and normalized so that the sum of the intensity is 1. We then obtained $\mathbf{a}, \mathbf{b}$ by (6) and $\mathbf{C}$ by the pixel-wise Euclidean distance. Because EROT is commonly used in real applications (Peyré and Cuturi 2019), we focused on C-EROT here and compared the Sinkhorn algorithm (Cuturi 2013), the cyclic one (Algorithm 2), and the two-stage one (Algorithm 3) over $190 (= {}_{20}\mathrm{C}_2)$ image pairs, for validation. Note that, in the two-stage Sinkhorn algorithm, we stopped Stage 1 before the end of convergence to prevent the solution far from the optimal one for real-world images; we first run the cyclic Sinkhorn algorithm until the marginal error $||\left(\mathrm{diag}(\widehat{\mathbf{p}})\mathbf{K}\mathrm{diag}(\widehat{\mathbf{q}})\right)^\top \mathbf{1}_m - \boldsymbol{\beta}||_2$ is below 1.0×10^{-3} and then run the Sinkhorn algorithm until the difference between its objective function value and the value obtained by directly solving C-EROT with the Sinkhorn algorithm is below 1.0×10^{-4}. We set $\lambda = 0.5$ for the regularizer (3).

Table 2 lists the results. The cyclic Sinkhorn algorithm showed the fastest computation time. However, because this algorithm assumes to satisfy Assumption 1 strictly, its objective function value differed from that of the original Sinkhorn algorithm, and marginal error occurred. In contrast, the two-stage Sinkhorn algorithm showed the same objective function value as that of the original one and no marginal error but with faster computation time than using the original one. These results indicate that the cyclic Sinkhorn algorithm can be a good choice for real-world data because of its fastest computation time if users tolerate the objective function value difference and the marginal error. If not, the two-stage Sinkhorn algorithm is promising for real-world data, which solves C-EROT with approximate cyclic symmetry faster than the original Sinkhorn algorithm.

Algorithm	$(h,w)=(64,64),\quad d=4096$			$(h,w)=(96,96),\quad d=9216$		
	Obj. value	Marginal error	Time (sec.)	Obj. value	Marginal error	Time (sec.)
Sinkhorn	4.320 ± 2.056	0.000 ± 0.000	16.610 ± 6.502	6.296 ± 3.100	0.000 ± 0.000	117.152 ± 53.442
Cyclic Sinkhorn	4.289 ± 2.048	0.001 ± 0.001	3.837 ± 1.286	6.250 ± 3.089	0.001 ± 0.001	26.087 ± 11.985
Two-Stage Sinkhorn	4.320 ± 2.056	0.000 ± 0.000	13.877 ± 6.244	6.296 ± 3.100	0.000 ± 0.000	91.790 ± 43.000

Table 2: Experimental results in the real-world image data.

Algorithm	C20H42 dataset, $\quad d=62$			C40H82 dataset, $\quad d=122$		
	Obj. value	Marginal error	Time (msec.)	Obj. value	Marginal error	Time (msec.)
Sinkhorn	0.598 ± 0.186	0.000 ± 0.000	1.534 ± 0.840	0.640 ± 0.220	0.000 ± 0.000	4.204 ± 4.331
Cyclic Sinkhorn	0.598 ± 0.186	0.000 ± 0.000	1.338 ± 0.619	0.640 ± 0.220	0.000 ± 0.000	1.712 ± 1.297

Table 3: Experimental results in the real-world C20H42 and C40H82 alkanes graph data.

Real-World Graph Data with Strict Cyclic Symmetry

We also tested our algorithms on another real-world case of 2-order cyclic symmetry in Example 3. No well-maintained graph dataset with symmetry exists, so we made two new chemical graph datasets about C20H42 and C40H82 alkanes, which consist of 15 and 10 chemical graphs of different structures with 2-order cyclic symmetry, respectively (see Figure 3). In this test, we set d as the number of atoms and $n = 2$. Following (Togninalli et al. 2019), we defined $\mathbf{a}, \mathbf{b}$ and $\mathbf{C}$ as those in Example 3. Because these graph data are real-world but with strict 2-order cyclic symmetry, we focused on C-EROT and compared only the Sinkhorn algorithm and the cyclic one (Algorithm 2), excluding the two-stage one (Algorithm 3), for validation. This validation was performed over $105(= {}_{15}\mathrm{C}_2)$ graph pairs in C20H42 dataset and $45(= {}_{10}\mathrm{C}_2)$ graph pairs in C40H82 dataset, respectively. We set $\lambda = 0.1$ for the regularizer (3).

Table 3 lists the graph comparison results. Our cyclic Sinkhorn algorithm had the same objective function values and marginal errors as the original one but showed faster computation times. These results support the effectiveness of our algorithm on real-world graph data.

Discussions and Limitations

Through this paper, we confirmed that our algorithms can solve C-ROT faster. For further progress, we discuss the following future issues. (I) In Assumption 1, we assume knowing the cyclic order n in advance. Because cyclic symmetry arises naturally from the physical structure of input data, this assumption is reasonable in some real-world cases. However, we must improve our algorithms for unknown-order cyclic symmetry. (II) It is unknown whether our algorithms can be generalized for other symmetries, e.g., dihedral symmetry (Gatermann and Parrilo 2004). Further development of our algorithms for general symmetries remains as future work. (III) The main contribution of this paper is showing the utilization of cyclic symmetry in OT with theoretical proofs, but we must test our algorithms in various real-world data for further development. (IV) Our study is orthogonal to the existing OT methods; our algorithms can be combined with many existing OT methods, including the state-of-the-art methods like the primal-dual descent methods (Dvurechensky, Gasnikov, and Kroshnin 2018; Guo, Ho, and Jordan 2020). As the first step, this paper combined the most widely adopted OT methods (the network simplex and Sinkhorn methods) with our algorithms to evaluate the computational improvements but a detailed analysis of the combination of our algorithms with the state-of-the-art methods remains as future work. (V) Approaches that combine special structures, such as low rankness or separability, with symmetry have excellent potential. However, it is not so trivial to combine them well because our symmetry approach may break such special structures. We must explore this potential in the future.

Conclusion

We proposed novel fast algorithms for OT with cyclic symmetry. We showed that such OT can be reduced to a smaller optimization problem that has significantly fewer variables as higher cyclic symmetry exists in the input data. Our algorithms solve the small problem instead of the original OT and achieve fast computation. Through experiments, we confirmed the effectiveness of our algorithms in synthetic/real-world data with strict/approximate cyclic symmetry. This paper cultivates a new research direction, OT with symmetry, and paves the way for future research.

References

Ahuja, R. K.; Magnanti, T. L.; and Orlin, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc.

Alaya, M. Z.; Berar, M.; Gasso, G.; and Rakotomamonjy, A. 2019. Screening Sinkhorn Algorithm for Regularized Optimal Transport. In *Advances in Neural Information Processing Systems*.

Altschuler, J.; Bach, F.; Rudi, A.; and Niles-Weed, J. 2019. Massively scalable Sinkhorn distances via the Nyström method. In *Advances in Neural Information Processing Systems*.

Altschuler, J.; Niles-Weed, J.; and Rigollet, P. 2017. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*.

Beck, A. 2017. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics.

Beck, A.; and Tetruashvili, L. 2013. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4): 2037–2060.

Blondel, M.; Seguy, V.; and Rolet, A. 2018. Smooth and Sparse Optimal Transport. In *International Conference on Artificial Intelligence and Statistics*.

Bonchev, D. 1991. *Chemical graph theory: introduction and fundamentals*, volume 1. CRC Press.

Bonneel, N.; Peyré, G.; and Cuturi, M. 2016. Wasserstein Barycentric Coordinates: Histogram Regression Using Optimal Transport. *ACM Transactions on Graphics*, 35(4).

Boyd, S.; and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Cicconet, M.; Birodkar, V.; Lund, M.; Werman, M.; and Geiger, D. 2017. A convolutional approach to reflection symmetry. *Pattern Recognition Letters*, 95: 44–50.

Courty, N.; Flamary, R.; Tuia, D.; and Rakotomamonjy, A. 2017. Optimal Transport for Domain Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9): 1853–1865.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*.

Dezső, B.; Jüttner, A.; and Kovács, P. 2011. LEMON – an Open Source C++ Graph Template Library. *Electronic Notes in Theoretical Computer Science*, 264(5): 23–45. Second Workshop on Generative Technologies.

Dieleman, S.; Willett, K. W.; and Dambre, J. 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2): 1441–1459.

Dvurechensky, P.; Gasnikov, A.; and Kroshnin, A. 2018. Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn's Algorithm. In *International Conference on Machine Learning*.

Gatermann, K.; and Parrilo, P. A. 2004. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra*, 192(1-3): 95–128.

Getreuer, P. 2013. A Survey of Gaussian Convolution Algorithms. *Image Processing On Line*, 3: 286–310.

Guillaume, C. 2012. Optimal transportation and economic applications. *Lecture Notes*.

Guo, W.; Ho, N.; and Jordan, M. 2020. Fast Algorithms for Computational Optimal Transport and Wasserstein Barycenter. In *International Conference on Artificial Intelligence and Statistics*.

Howard, E. 1965. *Garden Cities of To-Morrow*, volume 23. Mit Press.

Jaffé, H. H.; and Orchin, M. 2002. *Symmetry in chemistry*. Courier Corporation.

Kantorovich, L. 1942. On the transfer of masses (in Russian). In *Doklady Akademii Nauk*, volume 37, 227.

Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; and Bolton, E. E. 2022. PubChem 2023 update. *Nucleic Acids Research*, 51(D1): D1373–D1380.

Kusner, M.; Sun, Y.; Kolkin, N.; and Weinberger, K. 2015. From Word Embeddings To Document Distances. In *International Conference on Machine Learning*.

Ladd, M. F. C. 2014. *Symmetry of crystals and molecules*. Oxford University Press, USA.

Lin, T.; Ho, N.; and Jordan, M. 2019a. On Efficient Optimal Transport: An Analysis of Greedy and Accelerated Mirror Descent Algorithms. In *International Conference on Machine Learning*.

Lin, T.; Ho, N.; and Jordan, M. I. 2019b. On the Acceleration of the Sinkhorn and Greenkhorn Algorithms for Optimal Transport. In *ArXiv Preprint: 1906.01437v1*.

Liu, Y.; Zhu, L.; Yamada, M.; and Yang, Y. 2020. Semantic Correspondence as an Optimal Transport Problem. In *Computer Vision and Pattern Recognition*.

Monge, G. 1781. *Mémoire sur la théorie des déblais et des remblais*. De l'Imprimerie Royale.

Nikolentzos, G.; Meladianos, P.; and Vazirgiannis, M. 2017. Matching Node Embeddings for Graph Similarity. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.

Pang, S.; Du, A.; Orgun, M. A.; Wang, Y.; Sheng, Q. Z.; Wang, S.; Huang, X.; and Yu, Z. 2022. Beyond CNNs: Exploiting Further Inherent Symmetries in Medical Image Segmentation. *IEEE Transactions on Cybernetics*, 1–12.

Petric Maretic, H.; El Gheche, M.; Chierchia, G.; and Frossard, P. 2019. GOT: An Optimal Transport framework for Graph comparison. In *Advances in Neural Information Processing Systems*.

Peyré, G.; and Cuturi, M. 2019. *Computational Optimal Transport: With Applications to Data Science*. Now Publishers.

Pham, K.; Le, K.; Ho, N.; Pham, T.; and Bui, H. 2020. On Unbalanced Optimal Transport: An Analysis of Sinkhorn Algorithm. In *International Conference on Machine Learning*.

Rockafellar, R. T. 1970. *Convex Analysis*, volume 18 of *Princeton Mathematical Series*. New Jersey: Princeton University Press.

Shamai, G.; Aflalo, Y.; Zibulevsky, M.; and Kimmel, R. 2015. Classical Scaling Revisited. In *International Conference on Computer Vision*.

Sinkhorn, R. 1967. Diagonal equivalence to matrices with prescribed row and column sums. In *The American Mathematical Monthly*, 402–405.

Solomon, J.; de Goes, F.; Peyré, G.; Cuturi, M.; Butscher, A.; Nguyen, A.; Du, T.; and Guibas, L. 2015. Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains. *ACM Transactions on Graphics*, 34(4).

Tarjan, R. E. 1997. Dynamic trees as search trees via euler tours, applied to the network simplex algorithm. *Mathematical Programming*, 78(2): 169–177.

Tenetov, E.; Wolansky, G.; and Kimmel, R. 2018. Fast Entropic Regularized Optimal Transport Using Semidiscrete Cost Approximation. *SIAM Journal on Scientific Computing*, 40(5): A3400–A3422.

Togninalli, M.; Ghisu, E.; Llinares-López, F.; Rieck, B.; and Borgwardt, K. 2019. Wasserstein Weisfeiler–Lehman Graph Kernels. In *NeurIPS*, 6436–6446.

Wu, S.; Rupprecht, C.; and Vedaldi, A. 2023. Unsupervised Learning of Probably Symmetric Deformable 3D Objects From Images in the Wild (Invited Paper). *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4): 5268–5281.

Xie, T.; and Grossman, J. C. 2018. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.*, 120: 145301.

Zhao, W.; Chellappa, R.; Phillips, P. J.; and Rosenfeld, A. 2003. Face Recognition: A Literature Survey. *ACM Comput. Surv.*, 35(4): 399–458.