An efficient implementable inexact entropic proximal point algorithm for a class of linear programming problems

Hong T. M. Chu,* Ling Liang,† Kim-Chuan Toh,‡ and Lei Yang§

April 26, 2022

Abstract

We introduce a class of specially structured linear programming (LP) problems, which has favorable modeling capability for important application problems in different areas such as optimal transport, discrete tomography and economics. To solve these generally large-scale LP problems efficiently, we design an implementable inexact entropic proximal point algorithm (iEPPA) combined with an easy-to-implement dual block coordinate descent method as a subsolver. Unlike existing entropy-type proximal point algorithms, our iEPPA employs a more practically checkable stopping condition for solving the associated subproblems while achieving provable convergence. Moreover, when solving the capacity constrained multi-marginal optimal transport (CMOT) problem (a special case of our LP problem), our iEPPA is able to bypass the underlying numerical instability issues that often appear in the popular entropic regularization approach, since our algorithm does not require the proximal parameter to be very small in order to obtain an accurate approximate solution. Numerous numerical experiments show that our iEPPA is efficient and robust for solving large-scale CMOT problems. The experiments on the discrete tomography problem also highlight the potential modeling power of our model.

Keywords: Linear programming; proximal point algorithm; entropic proximal term; block coordinate descent; capacity constrained multi-marginal optimal transport.

1 Introduction

In this paper, we introduce a class of specially structured linear programming (LP) problems of the following form:

$$
\begin{aligned}
\min \;\; & \langle C, X \rangle \\
\text{s.t.} \;\; & X \in \Omega := \left\{ X \in \mathbb{R}^{n_1 \times n_2 \times n_3} \; : \; \mathcal{A}^{(i)}(X) = \boldsymbol{b}^{(i)}, \quad i = 1, \ldots, N, \quad 0 \le X \le U \right\},
\end{aligned}
\tag{1.1}
$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{A}^{(i)} : \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{m_i}$ is a given linear mapping defined by

$$
\mathcal{A}^{(i)}(X) := \begin{bmatrix} \langle A_1^{(i)}, X \rangle \\ \vdots \\ \langle A_{m_i}^{(i)}, X \rangle \end{bmatrix}, \quad A_j^{(i)} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \quad 1 \le j \le m_i, \quad 1 \le i \le N,
$$

---

*Department of Mathematics, National University of Singapore (`hongtmchu@u.nus.edu`).

†Department of Mathematics, National University of Singapore (`liang.ling@u.nus.edu`).

‡Department of Mathematics, and Institute of Operations Research and Analytics, National University of Singapore (`mattohkc@nus.edu.sg`). This research is supported in part by the Ministry of Education of Singapore under Academic Research Fund Grant number: R146-000-xxx-xxx.

§Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China (`yanglei.math@gmail.com`).

$\boldsymbol{b}^{(i)} = (b_1^{(i)}, \ldots, b_{m_i}^{(i)})^\top \in \mathbb{R}^{m_i}$ $(i = 1, \ldots, N)$, $C \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $U \in \mathbb{R}_+^{n_1 \times n_2 \times n_3} \cup \{\infty\}^{n_1 \times n_2 \times n_3}$ are given data. Moreover, the linear mappings $\mathcal{A}^{(i)}$ $(i = 1, \ldots, N)$ satisfy Assumption 1 below. As we shall see shortly, problem (1.1) is a generalization of the classical discrete optimal transport problem which has the form: $\min \left\{ \langle C, X \rangle : X \in \mathbb{R}^{n_1 \times n_2}, \sum_{s=1}^{n_2} X_{rs} = a_r, r = 1, \ldots, n_1, \sum_{r=1}^{n_1} X_{rs} = b_s, s = 1, \ldots, n_2, X \geq 0 \right\}$, where $\boldsymbol{a} := (a_1, \ldots, a_{n_1})^\top$ and $\boldsymbol{b} := (b_1, \ldots, b_{n_2})^\top$ are two given marginals[1] in n_1 and n_2-dimensional simplices, and $C \in \mathbb{R}^{n_1 \times n_2}$ is a given cost matrix.

Assumption 1. *For each fixed i $(1 \leq i \leq N)$, $A_j^{(i)}$ only has binary entries (0 or 1) for $j = 1, \ldots, m_i$, and the given constraint tensors $\{ A_j^{(i)} \mid j = 1, \ldots, m_i \}$ satisfy the property that*

$$A_j^{(i)} \circ A_k^{(i)} = 0, \quad if \ \ j \neq k, \ j, k = 1, \ldots, m_i,$$

where "$\circ$" denotes the Hadamard product.

The property stated in Assumption 1 is equivalent to saying that the non-zero patterns of any two distinct constraint tensors $A_j^{(i)}$ and $A_k^{(i)}$ do not overlap in the i-block of the linear constraints $\mathcal{A}^{(i)}(X) = \boldsymbol{b}^{(i)}$. Such structures may look unusual at the first glance, but do appear in a few important application problems, for example, the <u>ca</u>pacity constrained <u>m</u>ulti-marginal <u>o</u>ptimal <u>t</u>ransport (CMOT) problem with three marginals, the discrete tomography problem [1, 7, 42], the disaggregation of industry-by-industry input-output tables in economics [20], and reconstructions of unknown inter-bank liabilities with fixed constraints [18]; see details on first two examples in the next two paragraphs. Moreover, as we shall see later, such special structures allow us to design a highly efficient algorithm to solve the corresponding LPs since they can greatly facilitate the computations of the subproblems involved in our algorithm; see Section 3 and Appendix A for more details.

The discrete 3-marginal CMOT problem is modeled as follows:

$$\min_{X \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \quad \langle C, X \rangle$$

$$\text{s.t.} \quad \begin{aligned} &\sum_{s,t} X_{rst} = a_r, \ r = 1, \ldots, n_1, \quad \sum_{r,t} X_{rst} = b_s, \ s = 1, \ldots, n_2, \\ &\sum_{r,s} X_{rst} = c_t, \ t = 1, \ldots, n_3, \quad 0 \leq X \leq U, \end{aligned} \tag{1.2}$$

where $\boldsymbol{a} := (a_1, \ldots, a_{n_1})^\top \in \Sigma_{n_1}$, $\boldsymbol{b} := (b_1, \ldots, b_{n_2})^\top \in \Sigma_{n_2}$, $\boldsymbol{c} := (c_1, \ldots, c_{n_3})^\top \in \Sigma_{n_3}$ are three given marginals with Σ_{n_i} denoting the n_i-dimensional unit simplex for $i = 1, 2, 3$. When $n_3 = 1$, the above problem readily reduces to the classical optimal transport problem mentioned in the first paragraph, but with an additional upper bound constraint. It is clear that problem (1.2) falls into the form of (1.1) with

$$\begin{aligned} A_j^{(1)} &= \boldsymbol{e}_j^{(1)} \otimes \boldsymbol{1}_{n_2} \otimes \boldsymbol{1}_{n_3}, \quad j = 1, \ldots, n_1, \\ A_j^{(2)} &= \boldsymbol{1}_{n_1} \otimes \boldsymbol{e}_j^{(2)} \otimes \boldsymbol{1}_{n_3}, \quad j = 1, \ldots, n_2, \\ A_j^{(3)} &= \boldsymbol{1}_{n_1} \otimes \boldsymbol{1}_{n_2} \otimes \boldsymbol{e}_j^{(3)}, \quad j = 1, \ldots, n_3, \end{aligned} \tag{1.3}$$

where $\boldsymbol{e}_j^{(i)}$ denotes the jth unit vector in $\mathbb{R}^{n_i}$ $(i = 1, 2, 3)$, $\boldsymbol{1}_{n_i}$ denotes the n_i-dimensional vector of all ones for $i = 1, 2, 3$, and "$\otimes$" denotes the tensor product (see the definition at the end of this section). Problem (1.2) was first proposed and studied by Korman and McCann [24, 25] in the 2-marginal continuous case[2] as an important variant of the classical 2-marginal optimal transport (OT) problem. This variant takes into account limits on the transport capacities[3] via imposing a proper upper bound constraint $X \leq U$, and hence it is better able to model some real-life situations. Moreover, we note that if the constraints in (1.2) are summed over a single index instead of two indices (for example, $\sum_s X_{rst} = a_{rt}$ for $r = 1, \ldots, n_1$, $t = 1, \ldots, n_3$), then the resulting problem can model a multi-commodity flow problem on a bipartite graph, where the commodities are indexed by $t = 1, \ldots, n_3$; see, for example, [23].

---

[1]In the paper, the term 'marginal' refers to a vector obtained by the sum of entries of a matrix/tensor over an index set.
[2]In the paper, the 2-marginal case means that we consider problem (1.2) in the matrix case (namely, $n_3 = 1$).
[3]This consideration can date back to [26], and possibly earlier.

In the 2-dimensional discrete tomography problem studied in [42], one is given the marginals obtained from an $n \times n$ matrix (for simplicity, we discuss the matrix case instead of a third-order tensor) by summing its entries along different directions, for example, 0°, 45°, 90° and 135° directions. In this case, the formulation in [1] would require a sixth-order tensor to model the problem as a 6-marginal optimal transport problem. Unfortunately, this approach leads to an exponential increase in the computational cost because of the curse of dimensionality brought about by the extra dimensions introduced in the higher-order tensor. But using our model in (1.1), the variable remains as a matrix and the projections along the four directions are formulated as four blocks of linear constraints, each represented by a linear mapping $\mathcal{A}^{(i)}(X) = \boldsymbol{b}^{(i)}$ with $\boldsymbol{b}^{(i)}$ being the given ith marginal for $i = 1, \ldots, 4$. Moreover, it is not hard to verify that the constraint matrices associated with each linear mapping $\mathcal{A}^{(i)}$ satisfy Assumption 1. For the construction of a block of linear constraints that represents a projection along a specific direction, we refer the reader to subsection 4.3 and Appendix C.

Note that problem (1.1) has $n_1 n_2 n_3$ box-constrained variables and $\sum_{i=1}^{N} m_i$ linear equality constraints, and thus it is usually a very large-scale LP problem when the dimension of the variable *or* the number of blocks of linear constraints is large. Therefore, classical LP methods such as the simplex method and the interior point method may no longer be efficient enough *or* may consume too much memory when solving this problem. Recently, an entropic regularized approach was proposed in [6] to **approximately** solve problem (1.2) in the 2-marginal case with impressive numerical performance. This approach basically modifies the original LP problem by adding an entropic regularization to the objective, and then applies a certain efficient first-order method to solve the resulting computationally more tractable regularized problem to obtain an approximate solution of the original LP problem. In [6], Dykstra's algorithm[4] with Kullback-Leibler projections (DyKL) is adapted to solve the entropic regularized counterpart of problem (1.2) in the 2-marginal case (see (B.1)). This algorithm can be highly efficient if a crude approximate solution is adequate, in which case the regularization parameter needs not be very small. However, when one decreases the regularization parameter to a small value for obtaining a more accurate solution, the DyKL would encounter the difficulties of numerical instabilities (due to loss of accuracy involving overflow/underflow operations) and slow convergence speed, just as Sinkhorn's algorithm [37] employed in [13] for approximately solving the classical 2-marginal OT problem. Though the former difficulty can partially be alleviated by some stabilization techniques (e.g., applying the *log-sum-exp* operation [32, Section 4.4]) at the expense of losing some computational efficiency, the latter difficulty of slow convergence, however, is unavoidable when the regularization parameter is small, as clearly observed from our numerical experiments in Section 4. In addition, we are not aware of fast algorithms that are specifically designed for solving the more general problem (1.1).

In this paper, we develop an implementable inexact entropic proximal point algorithm (iEPPA) for solving problem (1.1). Our iEPPA falls into the family of Bregman-distance-based PPA [11, 12, 15, 16] and the family of ϕ-divergence-based PPA [3, 17, 21, 22, 38, 39], both of which have been widely studied in the literature, especially in the 1990's starting from the paper [11]. However, we should point out that we have made an essential change to the algorithm by introducing a more practical stopping condition (2.3) for solving the subproblems. Therefore, existing convergence results may not be applicable and the convergence analysis has to be re-established for our iEPPA; see Theorem 1. Moreover, as a byproduct, we actually develop a unified inexact framework for EPPA including Teboulle's framework [39] and Eckstein's framework [16] as special cases. This makes our iEPPA more flexible. To solve the subproblem (2.2), we first derive its dual problem and characterize the properties of its optimal solutions in Section 3. We then apply a block coordinate descent (BCD) method to solve the resulting dual problem and establish the linear convergence by revisiting some classical results for the BCD method in [28, 29, 41]. We also show how the subproblems in the BCD method can be solved efficiently under Assumption 1. In particular, no stabilization technique is needed for the BCD updates since our iEPPA does not require a small proximal parameter in each iteration. This is indeed a key advantage of our iEPPA over the popular entropic regularization approach in [6]. Recently, a similar algorithmic framework studied by Eckstein [16] was also adapted in [43] for solving the classical OT problem with encouraging numerical

---

[4]More details on Dykstra's algorithm and its Bregman extension can be found in [5, 14].

performance. However, the algorithm there was developed under a rather stringent inexact condition, which is nontrivial to verify or implement in practice.

The contributions of this paper are summarized as follows.

1. We introduce a class of specially structured LP problems (1.1), which covers some important existing problems and has favorable modeling capability. For example, it is able to formulate a tomography problem *without* using a high-order tensor. This is in contrast to [1, 7] where a high-order (equals to two plus the number of projection directions) tensor is used to model a 2D tomography problem, and consequently the resulting problem is extremely large-scale and prohibitively expensive to solve in terms of both memory consumption and computational cost. In addition, the third-order tensor model (1.1) and the subsequent algorithms can naturally be extended to higher-order cases if needed.

2. We develop an efficient iEPPA combined with a dual BCD method, namely, iEPPA+BCD, to solve the proposed structured LP problem (1.1). It has the important strength of being able to faithfully solve the original problem *without* requiring the proximal parameter to be very small. As a result, when solving the CMOT problem (1.2), it can bypass the inherent numerical instabilities that often plague the entropic regularization approach. While our iEPPA+BCD framework is not completely new but a novel combination of existing algorithms in the optimization literature, we have nevertheless introduced an essential modification to make the algorithm practically implementable by proposing a computationally checkable stopping condition for finding a sufficiently accurate approximate solution of the subproblem in each iEPPA iteration to ensure the convergence of the overall algorithm.

3. We conduct rigorous numerical experiments to illustrate the efficiency of our iEPPA+BCD framework for solving the CMOT problem (1.2), in comparison to the (stabilized) DyKL and the powerful commercial solver Gurobi. Experiments on the discrete tomography problem also show the favorable modeling power of our model.

The rest of this paper is organized as follows. The iEPPA for solving problem (1.1) and its convergence results are described in Section 2. The dual BCD method for solving the subproblem and its convergence analysis are presented in Section 3. Moreover, the details on the implementable verification of our new inexact condition is also included in Section 3. Extensive numerical results are reported in Section 4, with some concluding remarks given in Section 5.

Notation and Preliminaries The elements of a third-order tensor $X \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are denoted as X_{rst} where $1 \leq r \leq n_1$, $1 \leq s \leq n_2$, $1 \leq t \leq n_3$. For any tensors $X, Y \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we define their inner product as $\langle X, Y \rangle := \sum_{r=1}^{n_1} \sum_{s=1}^{n_2} \sum_{t=1}^{n_3} X_{rst} Y_{rst}$. The Frobenius norm of X is defined by $\|X\|_F := \sqrt{\langle X, X \rangle}$. For any $X, Y \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the Hadamard product of X and Y is defined by $(X \circ Y)_{rst} := X_{rst} Y_{rst}$ for any $1 \leq r \leq n_1$, $1 \leq s \leq n_2$, $1 \leq t \leq n_3$. Similarly, we use "./" to denote the element-wise division operator. We use "$\otimes$" to denote the tensor product of vectors. Specifically, let $\boldsymbol{u}^{(i)} \in \mathbb{R}^{n_i}$ $(i = 1, 2, 3)$ be three arbitrary column vectors. Their tensor product is denoted by $\boldsymbol{u}^{(1)} \otimes \boldsymbol{u}^{(2)} \otimes \boldsymbol{u}^{(3)} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ whose elements are given by $(\boldsymbol{u}^{(1)} \otimes \boldsymbol{u}^{(2)} \otimes \boldsymbol{u}^{(3)})_{rst} := u_r^{(1)} u_s^{(2)} u_t^{(3)}$ for any $1 \leq r \leq n_1$, $1 \leq s \leq n_2$, $1 \leq t \leq n_3$.

Let $\mathbb{E}$ be a finitely dimensional real Euclidean space equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. For an extended-real-valued function $f : \mathbb{E} \to [-\infty, \infty]$, we say that it is *proper* if $f(\boldsymbol{x}) > -\infty$ for all $\boldsymbol{x} \in \mathbb{E}$ and its domain $\mathrm{dom}\, f := \{\boldsymbol{x} \in \mathbb{E} : f(\boldsymbol{x}) < \infty\}$ is nonempty. A proper function f is said to be closed if it is lower semicontinuous. Assume that $f : \mathbb{E} \to (-\infty, \infty]$ is a proper closed convex function. For a given $\nu \geq 0$, the ν-subdifferential of f at $\boldsymbol{x} \in \mathrm{dom}\, f$ is defined by $\partial_\nu f(\boldsymbol{x}) := \{\boldsymbol{d} \in \mathbb{E} : f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{d}, \boldsymbol{y} - \boldsymbol{x} \rangle - \nu, \ \forall \boldsymbol{y} \in \mathbb{E}\}$ and when $\nu = 0$, $\partial_\nu f$ is simply denoted by ∂f, which is referred to as the subdifferential of f. The conjugate function of f is $f^* : \mathbb{E} \to (-\infty, \infty]$ defined by $f^*(\boldsymbol{y}) := \sup\{\langle \boldsymbol{y}, \boldsymbol{x} \rangle - f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{E}\}$. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{E}$, it follows from [34, Theorem 23.5] that

$$\boldsymbol{y} \in \partial f(\boldsymbol{x}) \iff \boldsymbol{x} \in \partial f^*(\boldsymbol{y}). \tag{1.4}$$

4

Moreover, we call a proper closed convex function f essentially smooth if (i) the interior of dom f, denoted by $\operatorname{int} \operatorname{dom} f$, is not empty; (ii) f is differentiable on $\operatorname{int} \operatorname{dom} f$; (iii) $\|\nabla f(x_k)\| \to \infty$ for every sequence $\{x_k\}$ in $\operatorname{int} \operatorname{dom} f$ converging to a boundary point of $\operatorname{int} \operatorname{dom} f$; see [34, page 251].

Finally, we make a blanket assumption throughout this paper.

Assumption 2. *The feasible set Ω is bounded and $\Omega \cap \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ is nonempty.*

This assumption ensures the well-definedness of problem (1.1) and our method developed in the next section. The boundedness assumption can be satisfied if, for instance, $U \in \mathbb{R}_{+}^{n_1 \times n_2 \times n_3}$ or the constraints are given as in (1.2).

2 An implementable inexact entropic proximal point algorithm

In this section, we develop an implementable inexact entropic proximal point algorithm (iEPPA) for solving problem (1.1). To describe the iterates of the iEPPA, we first rewrite problem (1.1) as follows:

$$\min_X \ \delta_{\Omega^\circ}(X) + \langle C, X \rangle, \quad \text{s.t.} \quad X \geq 0, \tag{2.1}$$

where $\delta_{\Omega^\circ}(\cdot)$ is the indicator function of the set Ω° defined as

$$\Omega^\circ := \big\{ X \in \mathbb{R}^{n_1 \times n_2 \times n_3} \ : \ \mathcal{A}^{(i)}(X) = \boldsymbol{b}^{(i)}, \ i = 1, \ldots, N, \ X \leq U \big\}.$$

Obviously, the set Ω° is formed by removing the non-negative constraint on X from the set Ω and hence $\Omega \subseteq \Omega^\circ$. We also introduce the Boltzmann-Shannon entropy function $\phi(X) = \sum_{rst} X_{rst} \log X_{rst} - X_{rst}$ (with the convention that $0 \log 0 = 0$) and its conjugate function $\phi^*(Y) = \sum_{rst} \exp(Y_{rst})$. Then, the Bregman distance [9] with ϕ as the kernel function, is defined as

$$\mathcal{D}_\phi(X, Y) := \phi(X) - \phi(Y) - \langle \nabla\phi(Y), X - Y \rangle, \quad \forall X \in \mathbb{R}_{+}^{n_1 \times n_2 \times n_3}, \ Y \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}.$$

It is easy to see that $D_\phi(X, Y) \geq 0$ and the equality holds if and only if $X = Y$. Then, the iEPPA for solving (2.1) (hence (1.1)) is presented as Algorithm 1.

Algorithm 1 An implementable inexact entropic proximal point algorithm (iEPPA) for solving (2.1)

Input: Let $\{\varepsilon_k\}_{k=0}^\infty$, $\{\nu_k\}_{k=0}^\infty$, $\{\eta_k\}_{k=0}^\infty$ and $\{\mu_k\}_{k=0}^\infty$ be four sequences of nonnegative scalars. Choose $X^0 = \widetilde{X}^0 \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ arbitrarily. Set $k = 0$.

while the termination criterion is not met, **do**

    Step 1. Find a pair $(X^{k+1}, \widetilde{X}^{k+1})$ by approximately solving the following problem

$$\min_X \ \delta_{\Omega^\circ}(X) + \langle C, X \rangle + \varepsilon_k \, \mathcal{D}_\phi(X, X^k), \tag{2.2}$$

    such that $X^{k+1} \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$, $\widetilde{X}^{k+1} \in \Omega$ and

$$\begin{aligned}
&\Delta^k \in \partial_{\nu_k} \delta_{\Omega^\circ}(\widetilde{X}^{k+1}) + C + \varepsilon_k \left(\nabla\phi(X^{k+1}) - \nabla\phi(X^k) \right) \\
&\text{with} \ \ \|\Delta^k\|_F \leq \eta_k, \ \ \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^{k+1}) \leq \mu_k.
\end{aligned} \tag{2.3}$$

    Step 2. Set $k = k+1$ and go to **Step 1**.

end while

Output: $(X^k, \widetilde{X}^k)$

The reader may have observed that the iEPPA in Algorithm 1 basically solves the original problem (2.1) (hence (1.1)) via approximately solving a sequence of subproblems (2.2) each involving a special entropic Bregman proximal term. Since $\operatorname{dom} \phi = \mathbb{R}_{+}^{n_1 \times n_2 \times n_3}$, the constraint $X \geq 0$ can be removed in

(2.2). Moreover, the Boltzmann-Shannon entropy function $\phi(X) = \sum_{rst} X_{rst} \log X_{rst} - X_{rst}$ is essentially smooth and strictly convex on $\mathbb{R}_+^{n_1 \times n_2 \times n_3}$. This together with Assumption 2 ensures that each subproblem (2.2) is well-defined in the sense that its optimal solution (denoted by $X^{k,*}$) uniquely exists and lies in $\mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$. Indeed, since $\Omega^\circ \cap \operatorname{dom} \phi = \Omega$ is bounded, the objective function in subproblem (2.2) is then level-bounded. Thus, a solution exists [35, Theorem 1.9] and must be unique since ϕ is strictly convex. The essential smoothness of ϕ and Assumption 2 further imply that $X^{k,*}$ can only lie in $\mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$. Notice that our inexact condition (2.3) always holds when $X^{k+1} = \widetilde{X}^{k+1} = X^{k,*}$ and hence it is achievable.

The inexact condition (2.3) is rather general to cover some existing inexact conditions, and more importantly, it makes our iEPPA more practical for solving problem (2.1) (hence (1.1)). When $\nu_k \equiv \eta_k \equiv \mu_k \equiv 0$, X^{k+1} (equals to $\widetilde{X}^{k+1}$) must be the exact optimal solution of subproblem (2.2). In this case, our exact version of iEPPA is indeed a special case of the classical exact generalized PPA (such as the ϕ-divergence-based PPA [3, 17, 21, 22, 38] and the Bregman-distance-based PPA [11, 12, 15]). When $\eta_k \equiv \mu_k \equiv 0$, condition (2.3) reduces to

$$0 \in \partial_{\nu_k} \delta_{\Omega^\circ}(X^{k+1}) + C + \varepsilon_k \big(\nabla \phi(X^{k+1}) - \nabla \phi(X^k) \big), \tag{2.4}$$

which is considered by Teboulle [39] in the ϕ-divergence-based PPA that allows the approximate computations of the subdifferential of δ_{Ω° at X^{k+1}, provided $X^{k+1} \in \Omega^\circ$. When $\nu_k \equiv \mu_k \equiv 0$, condition (2.3) reduces to

$$\Delta^k \in \partial \delta_{\Omega^\circ}(X^{k+1}) + C + \varepsilon_k \big(\nabla \phi(X^{k+1}) - \nabla \phi(X^k) \big) \quad \text{with} \quad \|\Delta^k\|_F \le \eta_k, \tag{2.5}$$

which is considered by Eckstein [16] in the Bregman-distance-based PPA and is typically easier to check than the ν-subdifferential-based condition (2.4). But again, it requires X^{k+1} to be in Ω°. The inexact algorithmic framework based on condition (2.5) has also been adapted in [43] for solving the classical 2-marginal OT problem. However, we should mention that *neither* Teboulle's inexact condition (2.4) *nor* Eckstein's inexact condition (2.5) is easy to implement for solving the subproblem with the complicated constraint that $X \in \Omega^\circ$. Because it is nontrivial to find a point X^{k+1} that *simultaneously* satisfies $X^{k+1} \in \Omega^\circ$ (required by the nonemptyness of $\partial_{\nu_k} \delta_{\Omega^\circ}(X^{k+1})$ or $\partial \delta_{\Omega^\circ}(X^{k+1})$) and $X^{k+1} \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ (required by the essentially smoothness of ϕ). This inadequacy thus motivated us to further relax conditions (2.4) and (2.5) to condition (2.3), in which $\partial_{\nu_k} \delta_{\Omega^\circ}$ and $\nabla \phi$ are allowed to be computed at two slightly different points, respectively. We shall show later in subsection 3.2 that the verification of our inexact condition (2.3) is more practically implementable.

We next establish the convergence of our iEPPA in Algorithm 1. Our analysis is inspired by several existing works (see, for example, [16, 39]), but is more involved due to the flexible inexact condition (2.3). We shall start with some elementary preliminaries. It is known from [10, Section 6.1] that the Boltzmann-Shannon entropy function ϕ has many elegant properties as a Bregman function (see [10, Definition 2.1]). We point out three of them below that are useful in our subsequential analysis. More details on the Bregman function can be found in [4, Section 4].

Property 1. *The following properties hold for $\phi(X) = \sum_{rst} X_{rst} \log X_{rst} - X_{rst}$.*

(i) *For any $X \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$, $\mathcal{D}_\phi(X, \cdot)$ is level-bounded.*

(ii) *If $\{Y^k\} \subseteq \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ converges to some $Y^* \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$, then $\mathcal{D}_\phi(Y^*, Y^k) \to 0$.*

(iii) *(**Convergence consistency**) If $\{X^k\} \subseteq \mathbb{R}_+^{n_1 \times n_2 \times n_3}$ and $\{Y^k\} \subseteq \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ are two sequences such that $\{X^k\}$ is bounded, $Y^k \to Y^*$ and $\mathcal{D}_\phi(X^k, Y^k) \to 0$, then $X^k \to Y^*$.*

We also recall two well-known results.

Lemma 1 (**Three points identity** [12, Lemma 3.1]). *For any $X \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$ and $Y, Z \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$, the following identity holds:*

$$\langle \nabla \phi(Y) - \nabla \phi(Z), \, X - Y \rangle = \mathcal{D}_\phi(X, Z) - \mathcal{D}_\phi(X, Y) - \mathcal{D}_\phi(Y, Z).$$

Lemma 2 ([33, Section 2.2]). *Suppose that $\{a_k\}_{k=0}^\infty \subseteq \mathbb{R}$ and $\{\gamma_k\}_{k=0}^\infty \subseteq \mathbb{R}$ are two sequences such that $\{a_k\}$ is bounded from below, $\sum_{k=0}^\infty \gamma_k < \infty$, and $a_{k+1} \le a_k + \gamma_k$ holds for all k. Then, $\{a_k\}$ is convergent.*

We are now ready to give the main convergence result.

Theorem 1 (**Convergence of the iEPPA**). *Suppose that Assumption 2 holds and $\{\varepsilon_k\}_{k=0}^\infty$, $\{\nu_k\}_{k=0}^\infty$, $\{\eta_k\}_{k=0}^\infty$, $\{\mu_k\}_{k=0}^\infty$ are four sequences of nonnegative scalars. Let $\{X^k\}$ and $\{\widetilde{X}^k\}$ be the sequences generated by the iEPPA in Algorithm 1. If $0 < \underline{\varepsilon} \le \varepsilon_k \le \bar{\varepsilon} < \infty$, $\sum \nu_k < \infty$, $\sum \eta_k < \infty$ and $\sum \mu_k < \infty$, then $\{X^k\}$ and $\{\widetilde{X}^k\}$ converge to a same optimal solution of problem (2.1) (hence problem (1.1)).*

Proof. First, from condition (2.3), there exists a $D^{k+1} \in \partial_{\nu_k} \delta_{\Omega^\circ}(\widetilde{X}^{k+1})$ such that

$$\Delta^k = D^{k+1} + C + \varepsilon_k\big(\nabla\phi(X^{k+1}) - \nabla\phi(X^k)\big).$$

Then, for any $P \in \Omega \subseteq \Omega^\circ$, we see that

$$0 \ge \langle D^{k+1},\, P - \widetilde{X}^{k+1}\rangle - \nu_k = \langle \Delta^k - C - \varepsilon_k\big(\nabla\phi(X^{k+1}) - \nabla\phi(X^k)\big),\, P - \widetilde{X}^{k+1}\rangle - \nu_k,$$

which implies that

$$\langle C,\, \widetilde{X}^{k+1}\rangle \le \langle C,\, P\rangle + \varepsilon_k\langle \nabla\phi(X^{k+1}) - \nabla\phi(X^k),\, P - \widetilde{X}^{k+1}\rangle + \langle \Delta^k,\, \widetilde{X}^{k+1} - P\rangle + \nu_k. \qquad (2.6)$$

Note that

$$\begin{aligned}
&\langle \nabla\phi(X^{k+1}) - \nabla\phi(X^k),\, P - \widetilde{X}^{k+1}\rangle \\
&= \langle \nabla\phi(X^{k+1}) - \nabla\phi(X^k),\, P - X^{k+1}\rangle - \langle \nabla\phi(X^{k+1}) - \nabla\phi(X^k),\, \widetilde{X}^{k+1} - X^{k+1}\rangle \\
&= \mathcal{D}_\phi(P, X^k) - \mathcal{D}_\phi(P, X^{k+1}) - \mathcal{D}_\phi(X^{k+1}, X^k) - \big(\mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k) - \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^{k+1}) - \mathcal{D}_\phi(X^{k+1}, X^k)\big) \\
&= \mathcal{D}_\phi(P, X^k) - \mathcal{D}_\phi(P, X^{k+1}) - \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k) + \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^{k+1}) \\
&\le \mathcal{D}_\phi(P, X^k) - \mathcal{D}_\phi(P, X^{k+1}) - \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k) + \mu_k,
\end{aligned}$$
$$(2.7)$$

where the second equality follows from the three points identity in Lemma 1. Moreover, since $\widetilde{X}^{k+1} \in \Omega$ and $P \in \Omega$, then $\langle \Delta^k,\, \widetilde{X}^{k+1} - P\rangle \le \|\widetilde{X}^{k+1} - P\|_F \|\Delta^k\|_F \le 2\rho\eta_k$, where the last inequality follows from the boundedness of Ω (by Assumption 2) and hence there exists a $\rho > 0$ such that $\|X\|_F \le \rho$ for all $X \in \Omega$. Combining this with (2.6) and (2.7), we have

$$\langle C,\, \widetilde{X}^{k+1}\rangle \le \langle C,\, P\rangle + \varepsilon_k\big(\mathcal{D}_\phi(P, X^k) - \mathcal{D}_\phi(P, X^{k+1}) - \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k)\big) + \varepsilon_k\mu_k + 2\rho\eta_k + \nu_k, \ \ \forall\, P \in \Omega. \ \ (2.8)$$

Now, set $P = \widetilde{X}^k$ in (2.8), we see that

$$\begin{aligned}
\langle C,\, \widetilde{X}^{k+1}\rangle &\le \langle C,\, \widetilde{X}^k\rangle + \varepsilon_k\big(\mathcal{D}_\phi(\widetilde{X}^k, X^k) - \mathcal{D}_\phi(\widetilde{X}^k, X^{k+1}) - \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k)\big) + \varepsilon_k\mu_k + 2\rho\eta_k + \nu_k \\
&\le \langle C,\, \widetilde{X}^k\rangle - \varepsilon_k\big(\mathcal{D}_\phi(\widetilde{X}^k, X^{k+1}) + \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k)\big) + \varepsilon_k(\mu_{k-1} + \mu_k) + 2\rho\eta_k + \nu_k \\
&\le \langle C,\, \widetilde{X}^k\rangle + \varepsilon_k(\mu_{k-1} + \mu_k) + 2\rho\eta_k + \nu_k.
\end{aligned}$$
$$(2.9)$$

Note that $\{\langle C,\, \widetilde{X}^k\rangle\}$ is bounded below since $\widetilde{X}^k$ is in the compact set Ω for all k. Then, since ε_k is nonnegative and bounded from above, $\sum \nu_k < \infty$, $\sum \eta_k < \infty$ and $\sum \mu_k < \infty$, it follows from (2.9) and Lemma 2 that $\{\langle C,\, \widetilde{X}^k\rangle\}$ is convergent. Also, we see from (2.9) that

$$\varepsilon_k\big(\mathcal{D}_\phi(\widetilde{X}^k, X^{k+1}) + \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k)\big) \le \langle C,\, \widetilde{X}^k\rangle - \langle C,\, \widetilde{X}^{k+1}\rangle + \varepsilon_k(\mu_{k-1} + \mu_k) + 2\rho\eta_k + \nu_k.$$

From this, together with $\varepsilon_k \ge \underline{\varepsilon} > 0$, $\nu_k \to 0$, $\eta_k \to 0$, $\mu_k \to 0$ and the fact that $\{\langle C,\, \widetilde{X}^k\rangle\}$ is convergent, we get that

$$\mathcal{D}_\phi(\widetilde{X}^k, X^{k+1}) \to 0 \quad \text{and} \quad \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k) \to 0.$$

Next, let X^* be an arbitrary optimal solution of (2.1) (hence (1.1)). Obviously, $\langle C, X^* \rangle \leq \langle C, \widetilde{X}^{k+1} \rangle$ for all k since $\widetilde{X}^{k+1} \in \Omega$. By setting $P = X^*$ in (2.8), dividing the resulting inequality by ε_k and rearranging terms, we see that

$$
\begin{aligned}
0 \leq\ & \mathcal{D}_\phi(X^*, X^{k+1}) \\
\leq\ & \mathcal{D}_\phi(X^*, X^k) + \varepsilon_k^{-1}\big(\langle C, X^* \rangle - \langle C, \widetilde{X}^{k+1} \rangle\big) - \mathcal{D}_\phi(\widetilde{X}^{k+1}, X^k) + \mu_k + \varepsilon_k^{-1}(2\rho\eta_k + \nu_k) \qquad (2.10) \\
\leq\ & \mathcal{D}_\phi(X^*, X^k) + \mu_k + \varepsilon_k^{-1}(2\rho\eta_k + \nu_k).
\end{aligned}
$$

Thus, we can conclude from the above inequality and Lemma 2 that $\{\mathcal{D}_\phi(X^*, X^k)\}$ is convergent. On the other hand, since $\{\widetilde{X}^k\}$ is bounded (due to $\widetilde{X}^k \in \Omega$), it has at least one cluster point. Suppose that $\widetilde{X}^\infty$ is a cluster point and $\{\widetilde{X}^{k_i}\}$ is a convergent subsequence such that $\lim_{i\to\infty} \widetilde{X}^{k_i} = \widetilde{X}^\infty$. Then, by using (2.8) with $P = X^*$ again, we have for all k_i,

$$
\begin{aligned}
\langle C, \widetilde{X}^{k_i} \rangle \leq\ & \langle C, X^* \rangle + \varepsilon_{k_i-1}\big(\mathcal{D}_\phi(X^*, X^{k_i-1}) - \mathcal{D}_\phi(X^*, X^{k_i}) - \mathcal{D}_\phi(\widetilde{X}^{k_i}, X^{k_i-1})\big) \\
& + \varepsilon_{k_i-1}\mu_{k_i-1} + 2\rho\eta_{k_i-1} + \nu_{k_i-1} \\
\leq\ & \langle C, X^* \rangle + \varepsilon_{k_i-1}\big(\mathcal{D}_\phi(X^*, X^{k_i-1}) - \mathcal{D}_\phi(X^*, X^{k_i})\big) + \varepsilon_{k_i-1}\mu_{k_i-1} + 2\rho\eta_{k_i-1} + \nu_{k_i-1}.
\end{aligned}
$$

Then, passing to the limit and recalling that $\{\mathcal{D}_\phi(X^*, X^k)\}$ is convergent, $0 < \underline{\varepsilon} \leq \varepsilon_k \leq \bar{\varepsilon} < \infty$, $\nu_k \to 0$, $\eta_k \to 0$, $\mu_k \to 0$, we obtain that

$$
\langle C, \widetilde{X}^\infty \rangle \leq \langle C, X^* \rangle.
$$

Note that $\widetilde{X}^\infty \in \Omega$ since Ω is closed. Thus, $\widetilde{X}^\infty$ is an optimal solution of (2.1) (hence (1.1)).

In addition, from Property 1(i) and the fact that $\{\mathcal{D}_\phi(X^*, X^k)\}$ is convergent, we can conclude that $\{X^k\}$ must be bounded and hence it has at least one cluster point. Suppose that X^∞ is a cluster point and $\{X^{k_j}\}$ is a convergent subsequence such that $\lim_{j\to\infty} X^{k_j} = X^\infty$. Then, from $\mathcal{D}_\phi(\widetilde{X}^{k_j}, X^{k_j}) \leq \mu_{k_j-1} \to 0$, the boundedness of $\{\widetilde{X}^{k_j}\}$ and Property 1(iii), we have that $\lim_{j\to\infty} \widetilde{X}^{k_j} = X^\infty$. Therefore, from what we have proved in the last paragraph, X^∞ is an optimal solution of (2.1) (hence (1.1)), and moreover, by using (2.10) with X^* replaced by X^∞, we can conclude that $\{\mathcal{D}_\phi(X^\infty, X^k)\}$ is convergent. On the other hand, it follows from $\lim_{j\to\infty} X^{k_j} = X^\infty$ and Property 1(ii) that $\mathcal{D}_\phi(X^\infty, X^{k_j}) \to 0$. Consequently, $\{\mathcal{D}_\phi(X^\infty, X^k)\}$ must converge to zero. Now, let $\widehat{X}^\infty$ be any cluster point of $\{X^k\}$ with a subsequence $\{X^{k'_j}\}$ such that $X^{k'_j} \to \widehat{X}^\infty$. Since $\mathcal{D}_\phi(X^\infty, X^k) \to 0$, we have $\mathcal{D}_\phi(X^\infty, X^{k'_j}) \to 0$. Using Property 1(iii) again, we see that $X^\infty = \widehat{X}^\infty$. Since $\widehat{X}^\infty$ is arbitrary, we can conclude that $\lim_{k\to\infty} X^k = X^\infty$. This, together with the boundedness of $\{\widetilde{X}^k\}$, $\mathcal{D}_\phi(\widetilde{X}^k, X^k) \to 0$ and Property 1(iii), implies that $\{\widetilde{X}^k\}$ also converges to X^∞. We then complete the proof. $\square$

From Theorem 1, we see that the convergence of our iEPPA can be easily guaranteed with proper choices of $\{\varepsilon_k\}$, $\{\nu_k\}$, $\{\eta_k\}$ and $\{\mu_k\}$. To make our iEPPA truly implementable, we will illustrate in the next section how to efficiently solve the subproblem (2.2) to find a pair $(X^k, \widetilde{X}^k)$ satisfying condition (2.3) at each iteration (see **Step** 1 in Algorithm 1).

3 A dual block coordinate descent method for solving (2.2)

In this section, we present an efficient method for solving the subproblem (2.2). Specifically, we first derive the dual problem of (2.2), which is conceivably more tractable, and then apply a block coordinate descent (BCD) method for solving it. Note that the subproblem (2.2) has the same form as the entropic regularized counterpart of problem (1.1). Thus, one can also follow [6] to apply the DyKL for solving it. However, our numerical comparisons have shown that the dual BCD method is more efficient than the DyKL for solving (2.2) with a fixed ε_k and hence it can be of independent interest for solving an entropic regularized problem in form of (2.2).[5]

---

[5]In this paper, we omit numerical comparisons between the dual BCD and DyKL to save space, and refer the interested reader to our early arXiv version (arXiv:2011.14312v2).

For notational simplicity, we drop the index k and consider the following generic problem with given $S \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ and $\varepsilon > 0$:

$$\min_{X \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \quad \langle C, X \rangle + \varepsilon \mathcal{D}_\phi(X, S)$$
$$\text{s.t.} \quad \mathcal{A}^{(i)}(X) - \boldsymbol{b}^{(i)} = 0, \quad i = 1, \dots, N, \tag{3.1}$$
$$X \leq U,$$

where $\phi(X) = \sum_{rst} X_{rst} \log X_{rst} - X_{rst}$. By introducing an auxiliary variable $Z \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and substituting ϕ into (3.1), we can equivalently reformulate problem (3.1) as

$$\min_{X, Z \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \quad \langle M, X \rangle + \varepsilon \sum_{r,s,t} X_{rst} (\log X_{rst} - 1) + \delta_+(Z)$$
$$\text{s.t.} \quad \mathcal{A}^{(i)}(X) - \boldsymbol{b}^{(i)} = 0, \quad i = 1, \dots, N, \tag{3.2}$$
$$X + Z = U,$$

where $M := C - \varepsilon \log S$ and $\delta_+(\cdot)$ is the indicator function over the set $\{Z \in \mathbb{R}^{n_1 \times n_2 \times n_3} : Z \geq 0\}$. The Lagrangian function associated with (3.2) is

$$\mathcal{L}(X, Z, \boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(N)}, W) = \left\langle M - \sum_{i=1}^N \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} - W, X \right\rangle + \varepsilon \sum_{r,s,t} X_{rst} (\log X_{rst} - 1)$$
$$+ \delta_+(Z) - \langle W, Z \rangle + \sum_{i=1}^N \langle \boldsymbol{y}^{(i)}, \boldsymbol{b}^{(i)} \rangle + \langle W, U \rangle,$$

where $\boldsymbol{y}^{(i)} \in \mathbb{R}^{m_i}$ ($i = 1, \dots, N$), $W \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are Lagrangian multipliers for (3.2) and $\mathcal{A}^{(i,*)} : \mathbb{R}^{m_i} \to \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the adjoint mapping of $\mathcal{A}^{(i)}$ that is defined by $\mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} := \sum_{j=1}^{m_i} y_j^{(i)} A_j^{(i)}$. Then, the dual problem of (3.2) is given by

$$\max_{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(N)}, W} \left\{ \min_{X, Z} \mathcal{L}(X, Z, \boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(N)}, W) \right\}. \tag{3.3}$$

Observe that

$$\min_X \left\{ \left\langle M - \sum_{i=1}^N \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} - W, X \right\rangle + \varepsilon \sum_{r,s,t} X_{rst} (\log X_{rst} - 1) \right\} = -\varepsilon \left\langle \widetilde{M}, \exp \left(\varepsilon^{-1} \left(W + \sum_{i=1}^N \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} \right) \right) \right\rangle,$$

where $\widetilde{M} := \exp(-M/\varepsilon) = S \circ \exp(-C/\varepsilon)$ and

$$\min_Z \left\{ \delta_+(Z) - \langle W, Z \rangle \right\} = \begin{cases} 0, & \text{if } W \leq 0, \\ -\infty, & \text{otherwise.} \end{cases}$$

Here the notation $\exp(X)$ means that the exponential operation is applied to all entries of X. With these facts and some manipulations, problem (3.3) is then equivalent to

$$\min_{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(N)}, W} \left\{ \begin{array}{l} R(\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(N)}, W) \\ := \varepsilon \langle \widetilde{M}, \exp(\varepsilon^{-1}(W + \sum_{i=1}^N \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)})) \rangle - \sum_{i=1}^N \langle \boldsymbol{y}^{(i)}, \boldsymbol{b}^{(i)} \rangle - \langle W, U \rangle + \delta_-(W) \end{array} \right\}, \tag{3.4}$$

where $\delta_-(\cdot)$ is the indicator function over the set $\{W \in \mathbb{R}^{n_1 \times n_2 \times n_3} : W \leq 0\}$. Now, we see that problem (3.4) is a convex problem with $N+1$ blocks of variables and is conceivably more tractable than the original problem (3.1). Indeed, for this kind of problems containing several blocks of variables, it is desirable to apply the BCD method, which basically minimizes the objective R with respect to $\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(N)}, W$ cyclically at each iteration; see Algorithm 2 for a detailed description.

We will show in the next subsection that the dual BCD in Algorithm 2 is R-linearly convergent and also provides the optimal solution of problem (3.1). Moreover, by using the nice structures imposed on $\mathcal{A}^{(i)}$ ($i = 1, \dots, N$) in Assumption 1 together with some careful manipulations as presented in subsection

Algorithm 2 A dual block coordinate descent method for solving (3.1)

Input: Choose $(\boldsymbol{y}^{(1),0}, \ldots, \boldsymbol{y}^{(N),0}, W^0) \in \operatorname{dom} R$ arbitrarily. Set $\ell = 0$.

while a termination criterion is not met, **do**

    Step 1. compute

$$\boldsymbol{y}^{(i),\ell+1} = \arg\min_{\boldsymbol{y}^{(i)}} R\big(\boldsymbol{y}^{(1),\ell+1}, \ldots, \boldsymbol{y}^{(i-1),\ell+1}, \boldsymbol{y}^{(i)}, \boldsymbol{y}^{(i+1),\ell}, \ldots, \boldsymbol{y}^{(N),\ell}, W^\ell\big), \quad 1 \le i \le N,$$

$$W^{\ell+1} = \arg\min_{W} R\big(\boldsymbol{y}^{(1),\ell+1}, \ldots, \boldsymbol{y}^{(N),\ell+1}, W\big).$$

    Step 2. Set $\ell = \ell + 1$ and go to **Step 1**.

end while

Output: $(\boldsymbol{y}^{(1),\ell}, \ldots, \boldsymbol{y}^{(N),\ell}, W^\ell)$

3.3, one can show that all subproblems in our dual BCD admit closed-form solutions, leading to the following explicit iterative scheme:

$$\boldsymbol{y}^{(i),\ell+1} = \varepsilon \log \boldsymbol{b}^{(i)} - \varepsilon \log \Big(\mathcal{A}^{(i)}\Big(\widetilde{M} \circ \exp\big(\varepsilon^{-1}\textstyle\sum_{q=1}^{i-1} \mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell+1}$$

$$+ \varepsilon^{-1}\textstyle\sum_{q=i+1}^{N} \mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell} \big) \circ \exp\big(\varepsilon^{-1}W^\ell\big)\Big)\Big), \quad 1 \le i \le N, \tag{3.5}$$

$$W^{\ell+1} = \min\Big\{ \varepsilon \log\Big(U./\big(\widetilde{M} \circ \exp\big(\varepsilon^{-1}\textstyle\sum_{q=1}^{N}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell+1}\big)\big)\Big), 0\Big\}.$$

Alternatively, for any $\ell \ge 0$, let $\boldsymbol{\xi}^{(i),\ell} := \exp\big(\varepsilon^{-1}\boldsymbol{y}^{(i),\ell}\big)$ for $i = 1, \ldots, N$ and $\Gamma^\ell := \exp\big(\varepsilon^{-1}W^\ell\big)$, then the iterative scheme (3.5) can be equivalently written as

$$\boldsymbol{\xi}^{(i),\ell+1} = \boldsymbol{b}^{(i)}./\mathcal{A}^{(i)}\Big(\widetilde{M} \circ (\mathcal{A}^{(1,\bullet)}\boldsymbol{\xi}^{(1),\ell+1}) \circ \cdots \circ (\mathcal{A}^{(i-1,\bullet)}\boldsymbol{\xi}^{(i-1),\ell+1})$$

$$\circ (\mathcal{A}^{(i+1,\bullet)}\boldsymbol{\xi}^{(i+1),\ell}) \circ \cdots \circ (\mathcal{A}^{(N,\bullet)}\boldsymbol{\xi}^{(N),\ell}) \circ \Gamma^\ell\Big), \quad 1 \le i \le N, \tag{3.6}$$

$$\Gamma^{\ell+1} = \min\Big\{ U./\big(\widetilde{M} \circ (\mathcal{A}^{(1,\bullet)}\boldsymbol{\xi}^{(1),\ell+1}) \circ \cdots \circ (\mathcal{A}^{(N,\bullet)}\boldsymbol{\xi}^{(N),\ell+1})\big), 1\Big\}.$$

Here, for any $\boldsymbol{z} \in \mathbb{R}^{m_i}$, the tensor $\mathcal{A}^{(i,\bullet)}\boldsymbol{z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as follows:

$$(\mathcal{A}^{(i,\bullet)}\boldsymbol{z})_{rst} = \begin{cases} (\mathcal{A}^{(i,*)}\boldsymbol{z})_{rst}, & \text{if } (r,s,t) \in \mathcal{J}^{(i)}, \\ 1, & \text{otherwise,} \end{cases}$$

where $\mathcal{J}^{(i)}$ is the aggregated non-zero pattern of $\mathcal{A}^{(i,*)}$ defined by

$$\mathcal{J}^{(i)} = \big\{ (r,s,t) \mid (A_j^{(i)})_{rst} \ne 0 \text{ for some } j \in \{1, \ldots, m_i\} \big\}. \tag{3.7}$$

Remark 1. *For the efficient implementation of* (3.6)*, it is more convenient to introduce the following tensors for $1 \le i \le N$:*

$$\widehat{M}^{(i),\ell+1} = \widetilde{M} \circ (\mathcal{A}^{(1,\bullet)}\boldsymbol{\xi}^{(1),\ell+1}) \circ \cdots \circ (\mathcal{A}^{(i-1,\bullet)}\boldsymbol{\xi}^{(i-1),\ell+1}) \circ (\mathcal{A}^{(i+1,\bullet)}\boldsymbol{\xi}^{(i+1),\ell}) \circ \cdots \circ (\mathcal{A}^{(N,\bullet)}\boldsymbol{\xi}^{(N),\ell}) \circ \Gamma^\ell,$$

$$\widehat{M}^{(N+1),\ell+1} = \widetilde{M} \circ (\mathcal{A}^{(1,\bullet)}\boldsymbol{\xi}^{(1),\ell+1}) \circ \cdots \circ (\mathcal{A}^{(N,\bullet)}\boldsymbol{\xi}^{(N),\ell+1}).$$

Then, the ℓ-th cycle of the BCD scheme in (3.6) *can be carried out as follows.*

$$\widehat{M}^{(1),\ell+1} = \Big(\widehat{M}^{(N+1),\ell}./(\mathcal{A}^{(1,\bullet)}\boldsymbol{\xi}^{(1),\ell})\Big) \circ \Gamma^\ell, \qquad\qquad \boldsymbol{\xi}^{(1),\ell+1} = \boldsymbol{b}^{(1)}./\mathcal{A}^{(1)}\big(\widehat{M}^{(1),\ell+1}\big),$$

$$\widehat{M}^{(i),\ell+1} = \Big(\widehat{M}^{(i-1),\ell+1}./(\mathcal{A}^{(i,\bullet)}\boldsymbol{\xi}^{(i),\ell})\Big) \circ (\mathcal{A}^{(i-1,\bullet)}\boldsymbol{\xi}^{(i-1),\ell+1}), \quad \boldsymbol{\xi}^{(i),\ell+1} = \boldsymbol{b}^{(i)}./\mathcal{A}^{(i)}\big(\widehat{M}^{(i),\ell+1}\big), \quad 2 \le i \le N,$$

$$\widehat{M}^{(N+1),\ell+1} = \big(\widehat{M}^{(N),\ell+1}./\Gamma^{(\ell)}\big) \circ (\mathcal{A}^{(N,\bullet)}\boldsymbol{\xi}^{(N),\ell+1}), \qquad \Gamma^{\ell+1} = \min\big\{U./\widehat{M}^{(N+1),\ell+1}, 1\big\}.$$

Note that in the actual implementation of (3.6)*, only a single tensor is used to store $\widehat{M}^{(i),\ell+1}$ for $i = 1, \ldots, N+1$, and it is repeatedly overwritten and updated.*

Note that both iterative schemes (3.5) and (3.6) are simple and easy-to-implement. The main computational complexity for (3.6) is $\mathcal{O}(n_1 n_2 n_3)$. In particular, since the iterative scheme (3.6) only needs elementwise multiplicatons and the simple $\min(\cdot)$ operation, it can be much more efficient than (3.5) in practice. However, like Sinkhorn's algorithm, (3.6) may also suffer from numerical instabilities when ε takes a small value. Hence, in the unlikely event where ε is a small value in our iEPPA, one can use (3.5) instead to carry on all computations in the log domain and perform the *log-sum-exp* (see, e.g., [32, Section 4.4]) technique for avoiding underflow/overflow. In general, the proximal parameter ε_k in our iEPPA does not need to be very small to obtain an accurate solution of the original problem (2.1) (hence (1.1)) in a fairly fast speed. This is also evident from our experiments which indicate that $\varepsilon = 0.05$ is sufficient for obtaining a good performance. Therefore, we can safely use the efficient iterative scheme (3.6) as a subroutine in our iEPPA.

In addition, we notice that there is a close connection between Dykstra's algorithm with Bregman projection (including DyKL as a special case) and (block) coordinate descent methods, although to the best of our knowledge, such a connection has not been stated explicitly until the recent work by Tibshirani [40]. Indeed, one can deduce from [40, Section 5] that the DyKL used in [6] for solving the entropic regularized problem in form of (3.1) (see Appendix B for the DyKL applied to the 2-marginal capacity constrained OT problem) is equivalent to the BCD method applied to the following dual problem

$$\min_{\Lambda_i \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \, i=1,\ldots,N+1} \Phi^* \left(\nabla\Phi(K) - \sum_{i=1}^{N+1} \Lambda_i \right) + \sum_{i=1}^{N+1} \delta^*_{\mathcal{S}_i}(\Lambda_i), \tag{3.8}$$

where $\Phi(X) := \sum_{rst} X_{rst} \log X_{rst}$, $K := S \circ \exp(-C/\varepsilon)$, $\mathcal{S}_i := \{X \in \mathbb{R}^{n_1 \times n_2 \times n_3} : \mathcal{A}^{(i)}(X) = b^{(i)}\}$ for $i = 1, \ldots, N$, $\mathcal{S}_{N+1} := \{X \in \mathbb{R}^{n_1 \times n_2 \times n_3} : X \leq U\}$, and $\delta^*_{\mathcal{S}_i}$ is the conjugate of the indicator function $\delta_{\mathcal{S}_i}$. It is clear that the dual problem (3.8) is different from ours in (3.4). Therefore, the DyKL and our dual BCD are not equivalent to each other. Moreover, our dual BCD (either (3.5) or (3.6)) consumes much less memory. Because our dual variables $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}, W)$ only need $\sum_{i=1}^{N} m_i + n_1 n_2 n_3$ units of memory, while the dual variables $(\Lambda_1, \ldots, \Lambda_{N+1})$ in (3.8) need $(N+1)n_1 n_2 n_3$ units of memory.

3.1 Convergence results for dual BCD

We next show the convergence results for our dual BCD in Algorithm 2. It is worth noting that the (block) coordinate descent method enjoys a long history for solving the problem (containing (3.4) as a special case) of minimizing a class of convex differentiable functions over a certain closed convex set; see, for example, [28, 29, 41]. Hence, our main convergence results are simply derived by revisiting these classic works. To this end, we first show the existence of optimal solutions of problems (3.1) and (3.4), and their relations in the following proposition whose proof can be found in Appendix A.1.

Proposition 1. *Suppose that Assumption 2 holds. Then, the optimal solutions of problems (3.1) and (3.4) exist. Moreover, for any optimal solution $\left(\bar{\boldsymbol{y}}^{(1)}, \ldots, \bar{\boldsymbol{y}}^{(N)}, \bar{W} \right)$ of problem (3.4),*

$$\bar{X} := \exp \left(\varepsilon^{-1} \left(\sum_{i=1}^{N} \mathcal{A}^{(i,*)} \bar{\boldsymbol{y}}^{(i)} + \bar{W} - M \right) \right) \tag{3.9}$$

is the optimal solution of problem (3.1).

Next we present the main convergence results for our dual BCD based on the theory developed in [28]. To make the paper self-contained, we provide its proof in Appendix A.2.

Theorem 2 (Convergence of dual BCD). *Let $\left\{ \left(\boldsymbol{y}^{(1),\ell}, \ldots, \boldsymbol{y}^{(N),\ell}, W^\ell \right) \right\}$ be the sequence generated by the dual BCD method in Algorithm 2, and let $X^\ell := \exp \left(\varepsilon^{-1} \left(\sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i),\ell} + W^\ell - M \right) \right)$. Then, the following statements hold.*

(i) *$\left\{ \left(\boldsymbol{y}^{(1),\ell}, \ldots, \boldsymbol{y}^{(N),\ell}, W^\ell \right) \right\}$ converges R-linearly to an optimal solution of problem (3.4).*

(ii) *$\{X^\ell\}$ converges R-linearly to an optimal solution of problem (3.1).*

3.2 Implementable verification of condition (2.3)

From the previous subsection, we know that the dual BCD can be efficiently applied for solving the subproblem (2.2) in our iEPPA. In this subsection, we shall discuss how to verify condition (2.3) at a point returned by the dual BCD.

We first assume that there is a mapping $\mathcal{G} : \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_1 \times n_2 \times n_3}$ such that for any $0 \leq X \leq U$, $\mathcal{G}(X) \in \Omega$ and $\|\mathcal{G}(X) - X\|_F \leq c \sum_{i=1}^{N} \|\boldsymbol{r}^{(i)}\|$, where $c > 0$ is a constant depending only on $\mathcal{G}$ and $\boldsymbol{r}^{(i)} := \boldsymbol{b}^{(i)} - \mathcal{A}^{(i)}(X)$ $(i = 1, \ldots, N)$ are residuals. Since Ω is a polyhedron, such a mapping is typically definable in practice. We give three examples as follows.

Example 1. *If the projection of a point X onto Ω (denoted by $P_\Omega(X)$) is easy to compute, one can directly use $\mathcal{G} = P_\Omega$. In this case, from the Hoffman error bound theorem [19], there exists a constant $c > 0$ such that $\|\mathcal{G}(X) - X\|_F \leq c \sum_{i=1}^{N} \|\boldsymbol{r}^{(i)}\|$.*

Example 2. *Suppose that a relative interior point X^{ri} of Ω is available on hand, i.e., $\mathcal{A}^{(i)}(X^{\mathrm{ri}}) = \boldsymbol{b}^{(i)}$, $i = 1, \ldots, N$, and $0 < X^{\mathrm{ri}} < U$. Note that such a point can be obtained for many choices of $\mathcal{A}^{(i)}$ and U. For example, in the 2-marginal COT problem, Ω is formed by $\{X \in \mathbb{R}^{m \times n} : X\boldsymbol{1}_n = \boldsymbol{a},\ X^\top \boldsymbol{1}_m = \boldsymbol{b},\ 0 \leq X \leq U\}$. If $U > \boldsymbol{ab}^\top$, then $\boldsymbol{ab}^\top$ is obviously a relative interior point. Otherwise, one can apply the alternating projection method or its variants to find a point in the intersection of $\{X \in \mathbb{R}^{m \times n} : X\boldsymbol{1}_n = \boldsymbol{a}\}$, $\{X \in \mathbb{R}^{m \times n} : X^\top \boldsymbol{1}_m = \boldsymbol{b}\}$ and $\{X \in \mathbb{R}^{m \times n} : \epsilon \leq X \leq U - \epsilon\}$ with some small $\epsilon > 0$. Having an available relative interior point X^{ri} on hand, we can then perform the following procedure. We first compute the projection of X onto $\overline{\Omega} := \{X \in \mathbb{R}^{m \times n} : \mathcal{A}^{(i)}(X) = \boldsymbol{b}^{(i)},\ i = 1, \ldots, N\}$, which is in general easier than P_Ω. Let $Z := P_{\overline{\Omega}}(X)$ and $V' := Z - X$. It follows from the Hoffman error bound theorem that $\|V'\|_F \leq c' \sum_{i=1}^{N} \|\boldsymbol{r}^{(i)}\|$ with some $c' > 0$. Then, if $0 \leq Z \leq U$, we are done. Otherwise, we employ a pullback strategy to obtain a point $\widetilde{X} = Z + \lambda (X^{\mathrm{ri}} - Z)$ with some $\lambda \in [0, 1]$. It is easy to see that $\mathcal{A}^{(i)}(\widetilde{X}) = \boldsymbol{b}^{(i)}$ for $i = 1, \ldots, N$. By choosing*

$$\lambda = \max \left\{ \max_{(r,s,t) \in \mathcal{J}_1} \left\{ \frac{Z_{rst} - U_{rst}}{Z_{rst} - X_{rst}^{\mathrm{ri}}} \right\}, \max_{(r,s,t) \in \mathcal{J}_2} \left\{ \frac{-Z_{rst}}{X_{rst}^{\mathrm{ri}} - Z_{rst}} \right\} \right\},$$

where $\mathcal{J}_1 := \{(r, s, t) : Z_{rst} > U_{rst}\}$ and $\mathcal{J}_2 := \{(r, s, t) : Z_{rst} < 0\}$, we can also ensure that $0 \leq \widetilde{X} \leq U$ and hence $\widetilde{X} \in \Omega$. Moreover, note from $Z = X + V'$ and $0 \leq X \leq U$ that

$$\lambda = \max \left\{ \max_{(r,s,t) \in \mathcal{J}_1} \left\{ \frac{X_{rst} + V'_{rst} - U_{rst}}{Z_{rst} - X_{rst}^{\mathrm{ri}}} \right\}, \max_{(r,s,t) \in \mathcal{J}_2} \left\{ \frac{-X_{rst} - V'_{rst}}{X_{rst}^{\mathrm{ri}} - Z_{rst}} \right\} \right\}$$

$$\leq \max \left\{ \max_{(r,s,t) \in \mathcal{J}_1} \left\{ \frac{V'_{rst}}{U_{rst} - X_{rst}^{\mathrm{ri}}} \right\}, \max_{(r,s,t) \in \mathcal{J}_2} \left\{ \frac{-V'_{rst}}{X_{rst}^{\mathrm{ri}}} \right\} \right\} \leq c'' \|V'\|_F,$$

where $c'' > 0$ is a constant depending only on U and X^{ri}. Then, we have

$$\|\widetilde{X} - X\|_F = \|Z + \lambda (X^{\mathrm{ri}} - Z) - X\|_F \leq \lambda \|X^{\mathrm{ri}} - Z\|_F + \|Z - X\|_F$$

$$\leq \lambda(\|X^{\mathrm{ri}} - X\|_F + \|Z - X\|_F) + \|Z - X\|_F \leq \lambda \|U\|_F + (1 + \lambda)\|V'\|_F$$

$$\leq c'' \|U\|_F \|V'\|_F + 2\|V'\|_F \leq (c'' \|U\|_F + 2)\, c' \sum_{i=1}^{N} \|\boldsymbol{r}^{(i)}\|.$$

Therefore, the above procedure can be used as $\mathcal{G}$, i.e., $\mathcal{G}(X) = \widetilde{X}$.

Example 3. *For the 3-marginal CMOT problem (1.2), we consider two cases.*

- *When no upper bound U is imposed or U is a trivial upper bound (e.g., U is a matrix of all ones), a highly efficient rounding procedure [27, Algorithm 2] (an extension of [2, Algorithm 2] to the multi-marginal case) can be readily used as $\mathcal{G}$, whose main computational complexity is $\mathcal{O}(n_1 n_2 n_3)$.*

- *When the upper bound U is nontrivial, one can perform as follows. Similar to Example 2, let X^{ri} be a relative interior point of Ω. We first apply the rounding procedure [27, Algorithm 2] on X to*

obtain a point Z in the set $\{X \in \mathbb{R}^{n_1 \times n_2 \times n_3} : \sum_{s,t} X_{rst} = a_r,\ r = 1, \ldots, n_1,\ \sum_{r,t} X_{rst} = b_s,\ s = 1, \ldots, n_2,\ \sum_{r,s} X_{rst} = c_t,\ t = 1, \ldots, n_3,\ X \geq 0\}$. If $Z \leq U$, we are done; otherwise, we employ a pullback strategy as Example 2 to obtain a point $\widetilde{X} = Z + \lambda (X^{\mathrm{ri}} - Z)$ with some $\lambda \in [0, 1]$. Thus, such a procedure can be used as $\mathcal{G}$.

Now, suppose that we have a dual point $\left(\boldsymbol{y}^{(1),\ell+1}, \ldots, \boldsymbol{y}^{(N),\ell+1}, W^{\ell+1}\right)$ given by our dual BCD at the ℓ-th iteration, and computed a primal point by

$$X^{\ell+1} := \exp\left(\varepsilon^{-1}\left(\sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i),\ell+1} + W^{\ell+1} - M\right)\right) \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}.$$

From the optimality condition of W-subproblem in Algorithm 2 and $\nabla\phi(X) = \log X$, one can verify that

$$\begin{cases} 0 = C + \varepsilon(\log X^{\ell+1} - \log S) - W^{\ell+1} - \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i),\ell+1}, & \text{(3.10a)} \\ W^{\ell+1} \leq 0,\ U - X^{\ell+1} \geq 0,\ \langle W^{\ell+1}, U - X^{\ell+1}\rangle = 0, & \text{(3.10b)} \\ \boldsymbol{r}^{(i),\ell+1} = \boldsymbol{b}^{(i)} - \mathcal{A}^{(i)}(X^{\ell+1}),\ 1 \leq i \leq N, & \text{(3.10c)} \end{cases}$$

where $\boldsymbol{r}^{(1),\ell+1}, \ldots, \boldsymbol{r}^{(N),\ell+1}$ are the residuals at the ℓ-th iteration. Clearly, when $\boldsymbol{r}^{(i),\ell+1} = 0$ for $i = 1, \ldots, N$, $X^{\ell+1}$ is an exact optimal solution. However, $\boldsymbol{r}^{(1),\ell+1}, \ldots, \boldsymbol{r}^{(N),\ell+1}$ are generally nonzero vectors.

Next, we perform the procedure $\mathcal{G}$ on $X^{\ell+1}$ to obtain that $\widetilde{X}^{\ell+1} = \mathcal{G}(X^{\ell+1}) \in \Omega \subseteq \Omega^{\circ}$ and $\|V^{\ell+1}\|_F \leq c\sum_{i=1}^{N}\|\boldsymbol{r}^{(i),\ell+1}\|$ with $V^{\ell+1} := \widetilde{X}^{\ell+1} - X^{\ell+1}$. Moreover, for any $Y \in \Omega^{\circ}$, we have

$$\begin{aligned} &\langle -W^{\ell+1} - \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i),\ell+1}, Y - \widetilde{X}^{\ell+1}\rangle = -\langle W^{\ell+1}, Y - \widetilde{X}^{\ell+1}\rangle \\ &= -\langle W^{\ell+1}, Y - U\rangle - \langle W^{\ell+1}, U - \widetilde{X}^{\ell+1}\rangle \leq -\langle W^{\ell+1}, U - X^{\ell+1} - V^{\ell+1}\rangle \\ &= \langle W^{\ell+1}, V^{\ell+1}\rangle \leq \|W^{\ell+1}\|_F \|V^{\ell+1}\|_F \leq c\|W^{\ell+1}\|_F \sum_{i=1}^{N}\|\boldsymbol{r}^{(i),\ell+1}\| \leq c\tilde{c}\sum_{i=1}^{N}\|\boldsymbol{r}^{(i),\ell+1}\|, \end{aligned} \tag{3.11}$$

where the first equality follows because $Y, \widetilde{X}^{\ell+1} \in \Omega^{\circ}$ and hence $\langle\sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i),\ell+1}, Y - \widetilde{X}^{\ell+1}\rangle = 0$, the first inequality follows from $W^{\ell+1} \leq 0$ and $Y - U \leq 0$, the third equality follows from (3.10b) and the last inequality follows because $\{W^{\ell}\}$ is convergent (by Theorem 2(i)) and hence must be bounded from the above by some constant $\tilde{c} > 0$. Thus, letting $\nu_{\ell} := c\tilde{c}\sum_{i=1}^{N}\|\boldsymbol{r}^{(i),\ell+1}\|$, we can obtain from (3.11) that $-W^{\ell+1} - \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i),\ell+1} \in \partial_{\nu_{\ell}} \delta_{\Omega^{\circ}}(\widetilde{X}^{\ell+1})$. This together with (3.10a) implies that

$$0 \in \partial_{\nu_{\ell}} \delta_{\Omega^{\circ}}(\widetilde{X}^{\ell+1}) + C + \varepsilon\left(\nabla\phi(X^{\ell+1}) - \nabla\phi(S)\right).$$

From this relation, we see that condition (2.3) is verifiable at the candidate $(X^{\ell+1}, \widetilde{X}^{\ell+1})$ and no error occurs on the left-hand-side in this case, i.e., $\Delta^{\ell} = 0$. Then, our inexact condition (2.3) can be satisfied when both the primal feasibility accuracy $\sum_{i=1}^{N}\|\boldsymbol{r}^{(i),\ell+1}\|$ and the Bregman distance $\mathcal{D}_{\phi}(\widetilde{X}^{\ell+1}, X^{\ell+1})$ are smaller than the specified tolerance parameters. In practical implementations, since the construction of $\widetilde{X}^{\ell+1}$ and the computation of the Bregman distance will incur additional overhead, one will not compute $\widetilde{X}^{\ell+1}$ at the *early* stage of the dual BCD iteration since it is not needed in the algorithm. Specifically, one may start to compute $\widetilde{X}^{\ell+1}$ for checking the Bregman distance $\mathcal{D}_{\phi}(\widetilde{X}^{\ell+1}, X^{\ell+1})$ *only* when the primal feasibility accuracy has decreased to a sufficiently small level. In this way, the overhead incurred will be reduced.

In contrast, for *either* Teboulle's inexact condition (2.4) *or* Eckstein's inexact condition (2.5), even though a feasible point $\widetilde{X}^{\ell+1} \in \Omega$ can be constructed successfully, one still *cannot* verify condition (2.4) or (2.5) at $\widetilde{X}^{\ell+1}$ because $\widetilde{X}^{\ell+1}$ may not lie in $\mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ and hence $\nabla\phi$ may not be well defined at $\widetilde{X}^{\ell+1}$. Thus, such existing inexact conditions may not be easy to verify, even if one is willing to do the expensive computation. In this regard, our inexact condition (2.3) is more advantageous.

3.3 Computation of solutions of subproblems in dual BCD

In this subsection, we provide more details on how to solve the subproblems efficiently in our dual BCD method via the special structures imposed on $\mathcal{A}^{(i)}$ $(i = 1, \ldots, N)$ in Assumption 1. Recall the iterative

scheme in Algorithm 2, for any $1 \le i \le N$, $\boldsymbol{y}^{(i),\ell+1}$ is computed by solving an unconstrained minimization problem:

$$\min_{\boldsymbol{y}^{(i)}} \left\{ \begin{array}{l} \varepsilon\Big\langle \widetilde{M}, \exp\Big(\varepsilon^{-1}\big(\mathcal{A}^{(i,*)}\boldsymbol{y}^{(i)} + \sum_{q=1}^{i-1}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell+1} + \sum_{q=i+1}^{N}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell} + W^\ell\big)\Big)\Big\rangle \\ - \langle \boldsymbol{y}^{(i)}, \boldsymbol{b}^{(i)}\rangle \end{array} \right\}. \tag{3.12}$$

To solve this problem, we give the following auxiliary proposition.

Proposition 2. *Suppose that Assumption 1 holds. Then, for any tensor $M \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and any vector $\boldsymbol{y}^{(i)} \in \mathbb{R}^{m_i}$, we have*

$$\big\langle M, \exp\big(\varepsilon^{-1}\mathcal{A}^{(i,*)}\boldsymbol{y}^{(i)}\big)\big\rangle = \big\langle \mathcal{A}^{(i)}(M), \exp\big(\varepsilon^{-1}\boldsymbol{y}^{(i)}\big)\big\rangle + \sum_{(r,s,t)\notin \mathcal{J}^{(i)}} M_{rst},$$

where $\mathcal{J}^{(i)}$ is defined in (3.7).

Proof. Note from Assumption 1 that $A_j^{(i)}$ only has binary entries (0 or 1) for all $j = 1, \ldots, m_i$, and the non-zero patterns of $\big\{A_j^{(i)} \mid j = 1, \ldots, m_i\big\}$ do not overlap with each other. Thus, one can see that

$$\big(\exp\big(\varepsilon^{-1}\mathcal{A}^{(i,*)}\boldsymbol{y}^{(i)}\big)\big)_{rst} = \big(\exp\big(\varepsilon^{-1}\sum_{j=1}^{m_i}A_j^{(i)}y_j^{(i)}\big)\big)_{rst}$$

$$= \left\{ \begin{array}{ll} \sum_{j=1}^{m_i}\exp\big(\varepsilon^{-1}y_j^{(i)}\big)\big(A_j^{(i)}\big)_{rst}, & \text{if } (r,s,t) \in \mathcal{J}^{(i)}, \\ \exp(0) = 1, & \text{otherwise.} \end{array} \right.$$

Let $\kappa = \sum_{(r,s,t)\notin \mathcal{J}^{(i)}} M_{rst}$. We then have that

$$\big\langle M, \exp\big(\varepsilon^{-1}\mathcal{A}^{(i,*)}\boldsymbol{y}^{(i)}\big)\big\rangle = \kappa + \sum_{(r,s,t)\in \mathcal{J}^{(i)}}\sum_{j=1}^{m_i}\exp\big(\varepsilon^{-1}y_j^{(i)}\big)\big(A_j^{(i)}\big)_{rst}M_{rst}$$

$$= \kappa + \sum_{j=1}^{m_i}\exp\big(\varepsilon^{-1}y_j^{(i)}\big)\left(\sum_{(r,s,t)\in \mathcal{J}^{(i)}}\big(A_j^{(i)}\big)_{rst}M_{rst}\right) = \kappa + \sum_{j=1}^{m_i}\exp\big(\varepsilon^{-1}y_j^{(i)}\big)\langle A_j^{(i)}, M\rangle$$

$$= \kappa + \sum_{j=1}^{m_i}\exp\big(\varepsilon^{-1}y_j^{(i)}\big)\big(\mathcal{A}^{(i)}(M)\big)_j.$$

This completes the proof. $\square$

Using Proposition 2, we can reformulate the first term in the objective function of (3.12) as below:

$$\Big\langle \widetilde{M}, \exp\Big(\varepsilon^{-1}\big(\mathcal{A}^{(i,*)}\boldsymbol{y}^{(i)} + \sum_{q=1}^{i-1}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell+1} + \sum_{q=i+1}^{N}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell} + W^\ell\big)\Big)\Big\rangle$$

$$= \Big\langle \widetilde{M} \circ \exp\Big(\varepsilon^{-1}\big(\sum_{q=1}^{i-1}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell+1} + \sum_{q=i+1}^{N}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell} + W^\ell\big)\Big), \exp\big(\varepsilon^{-1}\mathcal{A}^{(i,*)}\boldsymbol{y}^{(i)}\big)\Big\rangle$$

$$= \big\langle \widetilde{M}^{(i),\ell} \circ \exp\big(\varepsilon^{-1}W^\ell\big), \exp\big(\varepsilon^{-1}\mathcal{A}^{(i,*)}\boldsymbol{y}^{(i)}\big)\big\rangle = \big\langle \mathcal{A}^{(i)}\big(\widetilde{M}^{(i),\ell} \circ \exp\big(\varepsilon^{-1}W^\ell\big)\big), \exp\big(\varepsilon^{-1}\boldsymbol{y}^{(i)}\big)\big\rangle + \Upsilon,$$

where $\widetilde{M}^{(i),\ell} := \widetilde{M} \circ \exp\big(\varepsilon^{-1}\sum_{q=1}^{i-1}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell+1} + \varepsilon^{-1}\sum_{q=i+1}^{N}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell}\big)$ and Υ is a constant independent of $\boldsymbol{y}^{(i)}$. Thus, $\boldsymbol{y}^{(i),\ell+1}$ can be simply computed by

$$\boldsymbol{y}^{(i),\ell+1} = \arg\min_{\boldsymbol{y}^{(i)}} \Big\{ \varepsilon\big\langle \mathcal{A}^{(i)}\big(\widetilde{M}^{(i),\ell} \circ \exp\big(\varepsilon^{-1}W^\ell\big)\big), \exp\big(\varepsilon^{-1}\boldsymbol{y}^{(i)}\big)\big\rangle - \langle \boldsymbol{y}^{(i)}, \boldsymbol{b}^{(i)}\rangle \Big\}$$

$$= \varepsilon\log\boldsymbol{b}^{(i)} - \varepsilon\log\big(\mathcal{A}^{(i)}\big(\widetilde{M}^{(i),\ell} \circ \exp\big(\varepsilon^{-1}W^\ell\big)\big)\big).$$

After obtaining $\boldsymbol{y}^{(i),\ell+1}$, $i = 1, \ldots, N$, we then update $W^{\ell+1}$ by solving the following problem:

$$W^{\ell+1} = \arg\min_{W} \Big\{ \varepsilon\big\langle \widetilde{M}, \exp\big(\varepsilon^{-1}\sum_{q=1}^{N}\mathcal{A}^{(q,*)}\boldsymbol{y}^{(q),\ell+1} + \varepsilon^{-1}W\big)\big\rangle - \langle W, U\rangle + \delta_-(W) \Big\},$$

$$= \min\big\{\varepsilon\log\big(U./(\widetilde{M} \circ Y^{\ell+1})\big), 0\big\},$$

14

where $Y^{\ell+1} = \exp\left(\sum_{q=1}^{N} \mathcal{A}^{(q,*)} \boldsymbol{y}^{(q),\ell+1}\right)$.

From the above discussions, one can see that the binary coefficient entries and the non-overlapping pattern imposed on $\mathcal{A}^{(i)}$ ($i = 1, \ldots, N$) are vital for the efficient computation of solutions of the subproblems, and as we have mentioned after Assumption 1, such special structures do appear in application problems such as the CMOT problem and the discrete tomography problem.

4 Numerical experiments

In this section, we conduct numerical experiments to evaluate the performance of our iEPPA in Algorithm 1, which employs the dual BCD in Algorithm 2 as a subroutine, for solving the 2-marginal and 3-marginal CMOT problems (1.2). More details on applying the dual BCD for solving (3.1) with the constraints in (1.2) can be found in Appendix A.3. For the 2-marginal case, we compare our iEPPA with DyKL adapted in [6] (see also in Appendix B) and the commercial solver Gurobi. For the 3-marginal case, we only compare our iEPPA with Gurobi. Moreover, we conduct experiments by applying our model (1.1) for solving the discrete tomography problem [42]. All experiments are run in MATLAB R2021a on a workstation with Intel Xeon processor E5-2680v3@2.50GHz (with 12 cores and 24 threads) and 128GB of RAM, equipped with Linux OS.

It is easy to show that the dual problem of (1.1) is

$$\max_{\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(N)},W} \left\{ \sum_{i=1}^{N} \langle \boldsymbol{b}^{(i)}, \boldsymbol{y}^{(i)} \rangle + \langle U, W \rangle \; : \; \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} + W \leq C, \; W \leq 0 \right\}, \qquad (4.1)$$

and the Karush-Kuhn-Tucker (KKT) system for (1.1) and (4.1) is

$$\mathcal{A}^{(i)}(X) - \boldsymbol{b}^{(i)} = 0, \quad i = 1, \ldots, N, \quad \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} + W \leq C, \quad 0 \leq X \leq U,$$
$$\langle X, \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} + W - C \rangle = 0, \quad \langle W, U - X \rangle = 0, \quad W \leq 0. \qquad (4.2)$$

where $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}$ and W are the Lagrangian multipliers (or dual variables). It is well known (for example, from [8, Section 4.3]) that when both the primal problem (1.1) and the dual problem (4.1) are feasible, $(\widehat{X}, \widehat{\boldsymbol{y}}^{(1)}, \ldots, \widehat{\boldsymbol{y}}^{(N)}, \widehat{W})$ satisfies the KKT system (4.2) if and only if $\widehat{X}$ solves the primal problem (1.1) and $(\widehat{\boldsymbol{y}}^{(1)}, \ldots, \widehat{\boldsymbol{y}}^{(N)}, \widehat{W})$ solves the dual problem (4.1), respectively. Then, based on the KKT system (4.2), we define the relative KKT residual for any $(X, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}, W)$ as follows:

$$\Delta_{\mathrm{kkt}}\left(:= \Delta_{\mathrm{kkt}}(X, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}, W) \right) := \max\left\{ \Delta_i \; : \; 1 \leq i \leq 7 \right\},$$

where

$$\Delta_1(X) := \frac{\left(\sum_{i=1}^{N} \|\mathcal{A}^{(i)}(X) - \boldsymbol{b}^{(i)}\|^2 \right)^{1/2}}{1 + \left(\sum_{i=1}^{N} \|\boldsymbol{b}^{(i)}\|^2 \right)^{1/2}}, \quad \Delta_2(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}, W) := \frac{\left\| \max\{ \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} + W - C, 0 \} \right\|_F}{1 + \|C\|_F},$$

$$\Delta_3(X) := \frac{\|\min\{X, 0\}\|_F}{1 + \|X\|_F}, \quad \Delta_4(X) := \frac{\|\min\{U - X, 0\}\|_F}{1 + \|U\|_F}, \quad \Delta_5(W) := \frac{\|\max\{W, 0\}\|_F}{1 + \|W\|_F},$$

$$\Delta_6(X, W) := \frac{|\langle W, U - X \rangle|}{1 + \|U\|_F}, \quad \Delta_7(X, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}, W) := \frac{|\langle X, \sum_{i=1}^{N} \mathcal{A}^{(i,*)} \boldsymbol{y}^{(i)} + W - C \rangle|}{1 + \|C\|_F}.$$

Obviously, $(X, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}, W)$ is a solution of the KKT system (4.2) if and only if $\Delta_{\mathrm{kkt}} = 0$. We then use Δ_{kkt} to set up the stopping criterion for the iEPPA. Specifically, we terminate the iEPPA when

$$\Delta_{\mathrm{kkt}}\left(X^{k+1}, \boldsymbol{y}^{(1),k+1}, \ldots, \boldsymbol{y}^{(N),k+1}, W^{k+1} \right) < 10^{-5},$$

where X^{k+1} and $(\boldsymbol{y}^{(1),k+1}, \ldots, \boldsymbol{y}^{(N),k+1}, W^{k+1})$ are the approximate optimal solutions of the subproblem (2.2) and its corresponding dual problem, respectively, at the k-th iteration. The maximum number of iterations for the iEPPA is set to be 500.

The performance of our iEPPA naturally depends on the efficiency of the dual BCD for solving the subproblem (2.2). Therefore, the choice of the proximal parameter ε and the stopping criterion for the subproblem at each iteration are vital for implementing the iEPPA. Note that a smaller regularization parameter ε would lead to a more difficult subproblem, and moreover, a very small ε may also cause numerical instabilities due to the loss of accuracy involving overflow/underflow operations. In all our numerical experiments, we simply fix $\varepsilon = 0.05$. With this choice, we would not encounter any numerical instability and can safely use the iterative scheme (A.7) to solve the subproblem efficiently.

As discussed in subsection 3.2, our inexact condition (2.3) is verifiable and can be satisfied as long as both the primal feasibility accuracy $\sum_{i=1}^{N} \|\boldsymbol{r}^{(i),k,\ell+1}\|$ ($\boldsymbol{r}^{(i),k,\ell+1} := \boldsymbol{b}^{(i)} - \mathcal{A}^{(i)}(X^{k,\ell+1})$, $i = 1, \ldots, N$) and the Bregman distance $\mathcal{D}_\phi(\widetilde{X}^{k,\ell+1}, X^{k,\ell+1})$ are sufficiently small, where $X^{k,\ell+1}$ is obtained by substituting $(\boldsymbol{y}^{(1),k,\ell+1}, \ldots, \boldsymbol{y}^{(N),k,\ell+1}, W^{k,\ell+1})$ into (3.9) at the ℓ-th dual BCD iteration within the k-th outer iteration, and $\widetilde{X}^{k,\ell+1}$ can be constructed by $\widetilde{X}^{k,\ell+1} := \mathcal{G}(X^{k,\ell+1})$ with a proper procedure $\mathcal{G}$ (see Example 3 in subsection 3.2). Note that, by such a construction, we have $\|\widetilde{X}^{k,\ell+1} - X^{k,\ell+1}\|_F \le c \sum_{i=1}^{N} \|\boldsymbol{r}^{(i),k,\ell+1}\|$ for some constant $c > 0$. Thus, when the primal feasibility accuracy $\sum_{i=1}^{N} \|\boldsymbol{r}^{(i),k,\ell+1}\|$ is small, the Bregman distance $\mathcal{D}_\phi(\widetilde{X}^{k,\ell+1}, X^{k,\ell+1})$ is also likely to be small, as always observed from our experiments. Since constructing $\widetilde{X}^{k,\ell+1}$ and calculating the Bregman distance $\mathcal{D}_\phi(\widetilde{X}^{k,\ell+1}, X^{k,\ell+1})$ explicitly is more costly than calculating the primal feasibility accuracy, such a phenomenon then allows us to employ an economical way to check the condition $\mathcal{D}_\phi(\widetilde{X}^{k,\ell+1}, X^{k,\ell+1}) \le \mu_k$. Specifically, in our implementation, we *first* compute the relative primal feasibility accuracy $\Delta_1(X^{k,\ell+1})$, and *only start* to check $\mathcal{D}_\phi(\widetilde{X}^{k,\ell+1}, X^{k,\ell+1}) \le \mu_k$ when $\Delta_1(X^{k,\ell+1}) \le \widetilde{\mu}_k$ with $\{\widetilde{\mu}_k\}$ being a given summable positive sequence. This together with proper choices of $\{\widetilde{\mu}_k\}$ would help us to avoid the explicit construction of $\widetilde{X}^{k,\ell+1}$ and the computation of the Bregman distance as much as possible to save cost until the later stage of the dual BCD method, while enforcing our inexact condition (2.3) to guarantee the convergence of the iEPPA. In the following experiments, we set $\mu_k = \max\left\{(k+1)^{-1.1}, 10^{-6}\right\}$ and $\widetilde{\mu}_k = \max\left\{10^{-4} \times \left(\frac{2}{3}\right)^k, 10^{-6}\right\}$ for $k \ge 0$. As we shall see later, such a simple checking strategy is enough to obtain a good practical performance.

It is well known that Gurobi is one of the most powerful and reliable solvers for solving LPs. Therefore, we use the solution obtained by Gurobi as a benchmark to evaluate the quality of solutions obtained by other methods. In our experiments, we use Gurobi (version 9.5.1 with an academic license) by only choosing the barrier method and disabling the presolving phase as well as the cross-over strategy so that Gurobi has the best performance. The reasons for choosing the aforementioned settings are three-fold. First, as observed from our experiments, other methods (such as the primal/dual simplex method) embedded in Gurobi are in general not as efficient as the barrier method. Second, we observe that when the presolving phase is enabled, Gurobi appears to be rather unstable and may fail to give a reasonably accurate solution for large-scale CMOT problems. Third, the cross-over strategy is usually too costly in our tests.

4.1 Experiments on synthetic data for 2-marginal CMOT

In this subsection, we consider the CMOT problem (1.2) in the 2-marginal case and generate simulated examples to test each algorithm. For each example, we first generate two discrete probability distributions denoted by

$$D_1 := \left\{(a_r, \boldsymbol{p}_r) \in \mathbb{R}_+ \times \mathbb{R}^3 \ : \ r = 1, \ldots, n_1\right\} \quad \text{and} \quad D_2 := \left\{(b_s, \boldsymbol{q}_s) \in \mathbb{R}_+ \times \mathbb{R}^3 \ : \ s = 1, \ldots, n_2\right\}.$$

Here, $\boldsymbol{a} := (a_1, \ldots, a_{n_1})^\top$ and $\boldsymbol{b} := (b_1, \ldots, b_{n_2})^\top$ are probabilities/weights generated from the standard uniform distribution on the open interval $(0, 1)$, and further normalized such that $\sum_{r}^{n_1} a_r = \sum_{s}^{n_2} b_s = 1$. Moreover, $\{\boldsymbol{p}_r\}$ and $\{\boldsymbol{q}_s\}$ are the support points whose entries are drawn from a Gaussian mixture distribution. With these support points, the cost matrix C is generated by $C_{rs} = \|\boldsymbol{p}_r - \boldsymbol{q}_s\|^2$ for $1 \le r \le n_1$ and $1 \le s \le n_2$ and normalized by dividing (element-wise) by its maximal entry.

We next describe how to generate an upper bound matrix $U \in \mathbb{R}_{++}^{n_1 \times n_2}$. Note that if most of the entries of U are too large (e.g., U is a matrix of all ones), then such an upper bound matrix can be

redundant. Conversely, if most of the entries of U are too small, then the feasible set of (1.2) can be empty and Assumption 2 fails to hold. Hence, a randomly generated upper bound matrix U is usually unsatisfactory for our testing purpose. Thanks to the special structure of the constraints in (1.2), one can easily see that $P := \boldsymbol{a}\boldsymbol{b}^\top$ must lie in the set $\{X \in \mathbb{R}^{n_1 \times n_2} : X\mathbf{1}_{n_2} = \boldsymbol{a}, \ X^\top\mathbf{1}_{n_1} = \boldsymbol{b}, \ X \geq 0\}$. We then set $U := 2P = 2\boldsymbol{a}\boldsymbol{b}^\top$ as the upper bound matrix. With this setting, Assumption 2 can be satisfied and our numerical results also indicate that such an upper bound matrix is generally not redundant.

4.1.1   Comparisons between Gurobi, iEPPA and DyKL

In this part of experiments, we evaluate the performances of Gurobi, iEPPA and DyKL. For the DyKL, the entropic regularization parameter ε is chosen from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ in our numerical tests. For $\varepsilon \in \{10^{-1}, 10^{-2}\}$, we follow [6, Section 5.2] to implement the DyKL directly, while for $\varepsilon \in \{10^{-3}, 10^{-4}\}$, we adapt the *log-sum-exp* trick (see, for example, [32, Section 4.4]) to stabilize the DyKL (see Appendix B for the implementations of the DyKL). We terminate the DyKL when $\Delta_1(X^{k+1}) < 10^{-5}$, where X^{k+1} is generated by the DyKL at the k-th iteration. Moreover, the maximum number of iterations for the DyKL is set to be 20000.

Table 1 presents the computational results for different choices of (n_1, n_2). In this table, "normalized obj" denotes the normalized objective function value defined as $|\mathcal{F}^k - \mathcal{F}_g|/(1 + |\mathcal{F}_g|)$, where $\mathcal{F}_g$ denotes the objective value returned by Gurobi and $\mathcal{F}^k$ is the approximate objective function value obtained by each algorithm; "feasibility" denotes the primal feasibility accuracy, namely, $\max\{\Delta_1, \Delta_3, \Delta_4\}$; "time" denotes the total computational time (in seconds) and "iter" denotes the number of iterations. For our iEPPA, we also record the total number of dual BCD iterations. For instance, the item "14(418)" means that iEPPA took 14 outer iterations with a total of 418 dual BCD iterations.

Table 1: Numerical results on synthetic data for 2-marginal CMOT. In the table, "g" stands for Gurobi; "e" stands for iEPPA; "d1", "d2", "d3", "d4" stand for DyKL with $\varepsilon = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}, respectively.

n_1	n_2	g	e	d1	d2	d3	d4	g	e	d1	d2	d3	d4
				normalized obj						feasibility			
4000	2000	0	5.7e-05	1.5e-02	3.7e-04	2.5e-04	2.5e-04	2.1e-13	9.9e-07	8.0e-06	9.8e-06	1.0e-05	1.0e-05
4000	4000	0	6.2e-05	1.6e-02	3.8e-04	1.9e-03	1.9e-03	8.6e-16	9.9e-07	8.8e-06	9.9e-06	9.9e-06	1.0e-05
4000	8000	0	7.2e-05	1.5e-02	5.3e-04	2.9e-04	3.0e-04	2.3e-14	1.0e-06	9.9e-06	9.8e-06	1.0e-05	1.0e-05
5000	2500	0	5.4e-05	1.5e-02	2.6e-04	5.1e-04	5.1e-04	8.6e-14	9.8e-07	7.4e-06	9.8e-06	1.0e-05	1.0e-05
5000	5000	0	6.3e-05	1.6e-02	3.9e-04	1.6e-04	1.7e-04	1.5e-13	9.9e-07	7.1e-06	9.6e-06	1.0e-05	1.0e-05
5000	10000	0	5.4e-05	1.6e-02	3.5e-04	5.7e-04	5.8e-04	1.7e-15	9.8e-07	7.9e-06	9.9e-06	1.0e-05	1.0e-05
6000	3000	0	5.2e-05	1.6e-02	3.2e-04	7.7e-04	7.7e-04	2.0e-14	9.9e-07	8.8e-06	9.8e-06	1.0e-05	1.0e-05
6000	6000	0	5.5e-05	1.5e-02	3.6e-04	1.0e-03	1.0e-03	4.2e-14	9.9e-07	9.7e-06	1.0e-05	1.0e-05	1.0e-05
6000	12000	0	5.1e-05	1.6e-02	3.2e-04	5.2e-04	5.4e-04	3.5e-14	9.8e-07	8.2e-06	9.9e-06	1.0e-05	1.0e-05
7000	3500	0	6.2e-05	1.5e-02	4.4e-04	3.1e-04	3.2e-04	4.6e-14	9.8e-07	6.8e-06	9.7e-06	1.0e-05	1.0e-05
7000	7000	0	6.7e-05	1.7e-02	4.3e-04	8.8e-04	9.1e-04	5.2e-15	9.8e-07	9.7e-06	9.9e-06	1.0e-05	1.0e-05
7000	14000	0	4.1e-05	1.5e-02	2.2e-04	4.5e-04	4.6e-04	9.1e-14	9.9e-07	7.1e-06	1.0e-05	1.0e-05	1.0e-05
				iter						time (in seconds)			
4000	2000	-	14(418)	13	187	1120	11228	125.3	28.8	1.4	20.1	282.8	2714.2
4000	4000	-	14(346)	11	155	588	5911	169.4	61.4	3.4	48.9	361.1	3632.6
4000	8000	-	15(281)	10	135	857	8617	708.8	108.6	6.2	82.9	1038.1	10507.3
5000	2500	-	12(300)	13	162	951	9546	172.5	37.7	3.2	40.6	464.8	4614.3
5000	5000	-	14(292)	11	134	957	9549	527.3	74.2	5.3	65.1	912.3	9104.1
5000	10000	-	14(258)	10	128	737	7425	1198.4	137.1	9.0	112.7	1370.8	13865.0
6000	3000	-	13(370)	11	176	783	7873	249.9	60.7	3.8	58.8	538.8	5425.3
6000	6000	-	14(412)	13	233	891	9133	731.4	131.0	8.7	150.7	1213.0	12291.6
6000	12000	-	14(243)	10	129	780	7828	1686.5	190.5	12.9	157.8	2043.1	20847.3
7000	3500	-	15(315)	9	133	823	8316	352.5	77.1	4.3	61.5	771.7	7735.1
7000	7000	-	14(229)	8	107	543	5424	986.8	127.6	7.4	93.3	990.1	9971.4
7000	14000	-	13(277)	13	177	1109	11146	3239.9	266.2	22.6	292.3	3892.9	39098.4

From Table 1, one can observe that our iEPPA performs better than the DyKL in the sense that the

iEPPA always returns a better approximate objective function value (using Gurobi as the benchmark) with a comparable feasibility accuracy in much less CPU time. The accuracy for the normalized objective function value returned by the iEPPA is always at the level of 10^{-5}, while the accuracy of the DyKL is usually at the level of 10^{-4}. In particular, decreasing the value of ε from 10^{-2} to 10^{-4} in the DyKL does not improve the accuracy for the objective function value significantly, but is more time-consuming (this phenomenon is detailed more in Remark 2). Therefore, the DyKL and its stabilized variants may not be efficient for computing a relatively high precision solution of the original LP problem. Moreover, for large-scale problems, Gurobi is rather time-consuming and memory-consuming. As an example, for the case where $(n_1, n_2) = (7000, 14000)$ in Table 1, a large-scale LP containing 9.8×10^7 box-constrained variables and 21000 equality constraints was solved. In this case, we observe that Gurobi is at least 10 times slower than our iEPPA, and it also needs about 55GB of RAM whereas our iEPPA only requires 15GB of RAM.

Remark 2. *For the DyKL, the accuracy of the solution in terms of the normalized objective function value is supposed to become better when the regularization parameter ε becomes smaller. However, we only observe such a phenomenon when ε is decreased from 10^{-1} to 10^{-2}. When $\varepsilon \in \left\{ 10^{-2}, 10^{-3}, 10^{-4} \right\}$, the accuracy remains almost the same. The reason is that the DyKL actually suffers from very slow convergence speed when ε is small and hence the stopping tolerance $Tol_d = 10^{-5}$ is not sufficient for the DyKL to obtain a good approximate solution. Indeed, when we set $Tol_d = 10^{-7}$ and test the DyKL with $\varepsilon = 10^{-3}$ on the case with $(n_1, n_2) = (4000, 2000)$ (same as the first instance in Table 1), the returned normalized objective function value is 2.7×10^{-6}, which is much smaller than the accuracy (2.5×10^{-4}) reported in Table 1. However, the computational time also increases dramatically.*

4.1.2 Comparisons between Gurobi and iEPPA

To further evaluate the performance of our iEPPA, we conduct more experiments on synthetic data with support points generated by the same Gaussian mixture distribution as in the previous set of experiments. In the following experiments, we set $n_1 = n_2 = n$ and vary n from 1000 to 9000. The computational results are presented in Figure 1. From the results, we see that the "nobj" of iEPPA is always at the level of 10^{-5}, which means that the objective function value returned by iEPPA is always close to that of Gurobi. Moreover, the computational time of iEPPA increases almost linearly with respect to n, while the computational time taken by Gurobi grows much more rapidly than iEPPA. This is because when the problem size becomes large, the barrier method used in Gurobi may not be efficient enough and may also consume too much memory which may not be affordable on an ordinary PC. In contrast, our iEPPA scales well in the sense that its computational time and memory consumption only grow at a low rate. Thus, it can be more favorable for solving the large-scale CMOT problem up to a moderate accuracy.

4.2 Experiments on synthetic data for 3-marginal CMOT

In this subsection, we consider the standard CMOT problem (1.2) in the 3-marginal case. Here, in view of the inferior performance of the DyKL presented in the last section, we only generate synthetic instances to evaluate the performance of our iEPPA against Gurobi to save space. Specially, we randomly generate three discrete probability distributions: $D_1 = \left\{ (a_r, \boldsymbol{p}_r) \in \mathbb{R}_+ \times \mathbb{R}^3 : r = 1, \ldots, n_1 \right\}$, $D_2 = \left\{ (b_s, \boldsymbol{q}_s) \in \mathbb{R}_+ \times \mathbb{R}^3 : s = 1, \ldots, n_2 \right\}$ and $D_3 = \left\{ (c_t, \boldsymbol{r}_t) \in \mathbb{R}_+ \times \mathbb{R}^3 : t = 1, \ldots, n_3 \right\}$. Similar to the 2-marginal case in subsection 4.1, the marginals $\boldsymbol{a} := (a_1, \ldots, a_{n_1})^\top$, $\boldsymbol{b} := (b_1, \ldots, b_{n_2})^\top$ and $\boldsymbol{c} := (c_1, \ldots, c_{n_3})^\top$ are all generated independently from a uniformly distribution on the interval $(0, 1)$, respectively. Again, the marginals are normalized so that $\sum_{r=1}^{n_1} a_r = \sum_{s=1}^{n_2} b_s = \sum_{t=1}^{n_3} c_t = 1$. Moreover, the support points are generated independently from a Gaussian mixture distribution. Given these support points, we then compute the cost tensor C as follows:

$$C_{rst} := \|\boldsymbol{p}_r - \boldsymbol{q}_s\|^2 + \|\boldsymbol{q}_s - \boldsymbol{r}_t\|^2 + \|\boldsymbol{r}_t - \boldsymbol{p}_r\|^2, \quad \forall\, 1 \le r \le n_1,\ 1 \le s \le n_2,\ 1 \le t \le n_3.$$

We also normalize C by dividing it by its maximal entry. To generate a reasonable upper bound U, we adapt the same strategy as in the 2-marginal case to set $U := 2\,(\boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c})$. Figure 2 presents comparisons

problem		nobj		feas	
id	n	g	e	g	e
1	1000	0	6.2e-05	5.3e-14	1.0e-06
2	2000	0	5.3e-05	2.1e-14	9.8e-07
3	3000	0	5.6e-05	4.3e-14	9.8e-07
4	4000	0	6.2e-05	8.6e-16	9.9e-07
5	5000	0	6.3e-05	1.5e-13	9.9e-07
6	6000	0	5.5e-05	4.2e-14	9.9e-07
7	7000	0	6.7e-05	5.2e-15	9.8e-07
8	8000	0	4.7e-05	1.7e-14	9.8e-07
9	9000	0	5.7e-05	4.2e-14	1.0e-06

Figure 1: Comparisons between Gurobi and iEPPA for 2-marginal CMOT. In the table, "nobj" denotes the normalized objective function value, "feas" denotes the primal feasibility accuracy, "g" stands for Gurobi and "e" stands for iEPPA.

between iEPPA and Gurobi, where we set $n_1 = n_2 = n_3 = n$ and vary n from 50 to 500. Similar to the 2-marginal case in subsection 4.1.2, our iEPPA has better scalability with respect to the problem size for solving problems to a moderate accuracy.

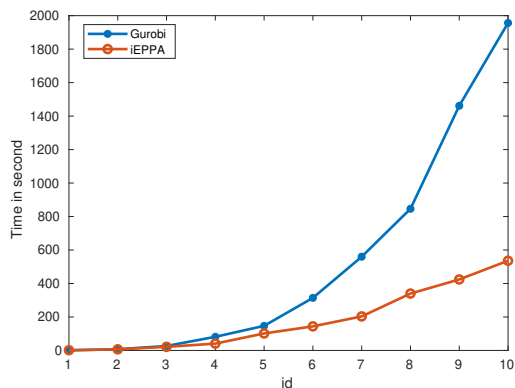| problem		nobj		feas	
id	n	g	e	g	e
1	50	0	4.6e-05	1.1e-12	1.0e-06
2	100	0	5.0e-05	7.0e-13	1.0e-06
3	150	0	5.0e-05	1.1e-12	9.8e-07
4	200	0	5.0e-05	1.3e-12	9.9e-07
5	250	0	4.9e-05	4.5e-13	9.9e-07
6	300	0	5.2e-05	3.2e-13	9.9e-07
7	350	0	4.9e-05	1.2e-12	9.9e-07
8	400	0	5.1e-05	1.1e-12	1.0e-06
9	450	0	5.3e-05	1.4e-12	9.9e-07
10	500	0	5.7e-05	5.3e-13	9.7e-07

Figure 2: Comparisons between Gurobi and iEPPA for 3-marginal CMOT with $n_1 = n_2 = n_3 = n$ and $n \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$.

4.3 Experiments on an application to discrete tomography

In this subsection, we conduct experiments on discrete tomography to illustrate the modeling capability of our model (1.1). We should mention that the purpose here is to present a preliminary investigation on the potential of using our model together with the iEPPA+BCD framework for solving the discrete tomography problem. A thorough numerical investigation is beyond the scope of this paper and will be left as a future research topic.

Let X be a 2D image of size $n \times n$ and $\vec{v}$ be a given direction that takes the form $(1, p)$, $(1, -p)$, $(p, 1)$ or $(p, -1)$ with p being a nonnegative integer. In our experiments, a tomographic projection on the image X along $\vec{v}$ is constructed as follows: we view the image X as a 2D grid of size $n \times n$, first pick all lines which are parallel to the direction $\vec{v}$ on this grid, then sum the entries on each line to form a vector. Such a projection then corresponds to a block of linear equality constraints of the form $\boldsymbol{b}^{(i)} = \mathcal{A}^{(i)}(X)$ in our model. Figure 3 shows the constructions of the tomographic projection along directions $(1, 0)$ and

19

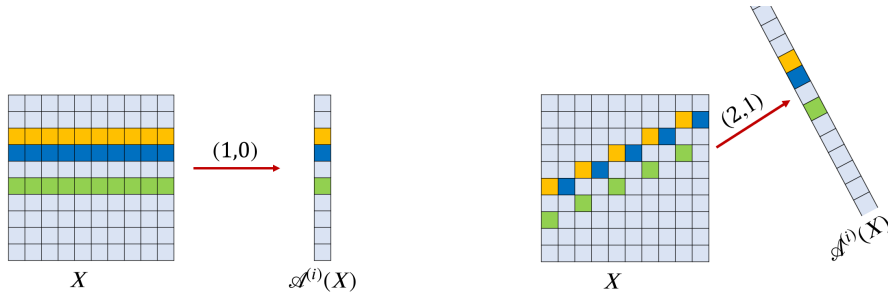$(2, 1)$, respectively. More details on the construction can be found in Appendix C.

Figure 3: Examples of the operator $\mathcal{A}^{(i)}$ for direction $\vec{v} = (1, 0)$ (**left**) and direction $\vec{v} = (2, 1)$ (**right**).

In the following experiments, we will use five ground-truth images of size 256×256 (namely, $n = 256$), as shown in the first row of Figure 5. For each of them, we compute N tomographic projections (which correspond to the linear mappings $\mathcal{A}^{(i)}$, $i = 1, \dots, N$ in our model) on this image along different directions to obtain $\boldsymbol{b}^{(i)}$, $i = 1, \dots, N$. Then, our goal is to recover the original image from the collection of projections $\{\boldsymbol{b}^{(i)}\}_{i=1}^N$ via applying our iEPPA+BCD framework for solving problem (1.1). Moreover, in our experiments, we set the entries of the cost matrix $C \in \mathbb{R}^{n \times n}$ to be $C_{rs} = |r - s|^2$ for $1 \leq r, s \leq n$ and normalize C by dividing (element-wise) it by its maximal entry.[6] We do not use any upper bound matrix U in the experiments. Moreover, to quantify the reconstruction quality between the recovered solution X^k and the ground-truth image X, we evaluate the peak signal-to-noise ratio that is defined as $\texttt{PSNR} = 10 \cdot \log_{10}\left(n^2 \cdot \max\{X_{rs} \,:\, 1 \leq r, s \leq n\}^2 / \|X^k - X\|_F^2\right)$.

Figure 4 presents the $\texttt{PSNR}$ values of the reconstructed images for $N \in \{10, 20, \dots, 90\}$. For a better visualization, we also show the reconstructed images corresponding to $N \in \{20, 50, 80\}$ in Figure 5. From the results, we observe that our model (1.1) can faithfully recover the ground-truth image and the quality of the reconstructed image gradually improves when more projections are used. Thus, to improve the quality of the reconstructed image, a straightforward way is to increase the number of projections. Fortunately, for our approach of using model (1.1) and the iEPPA+BCD, imposing more projections would not increase the computational cost dramatically since the main computational unit (which is one BCD iteration) only depends on N linearly. Specifically, for one more projection, we only need to add one more block of constraints in our model (1.1) and then correspondingly add one more block of dual variables in the dual BCD method.
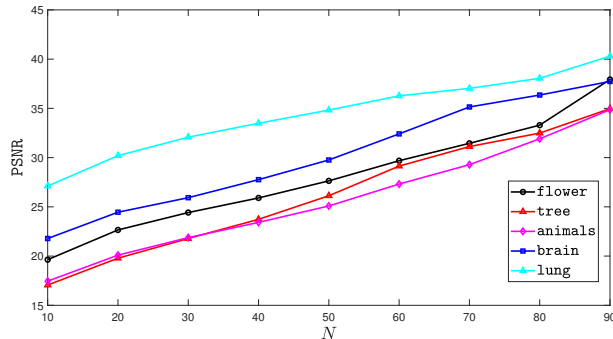

Figure 4: $\texttt{PSNR}$ values of the reconstructed images for $N \in \{10, 20, \dots, 90\}$.

---

[6]The setting of the cost matrix C may depend on the prior knowledge on the distribution of features in an image. Here, we simply set the entry of C to be $C_{rs} = |r - s|^2$, $\forall 1 \leq r, s \leq n$, and normalize it for the preliminary testing purpose. More study on the choice of C will be left in the future.
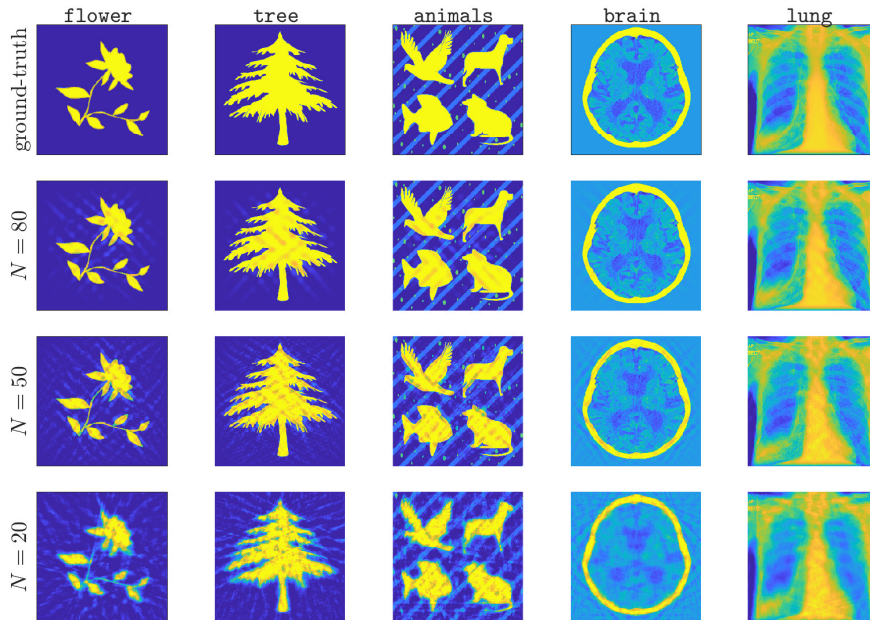
Figure 5: **The first row**: ground-truth images of size 256×256. Here, `flower`, `tree` and `animals` are artificial images, while `brain` and `chest` are taken from `https://radiopaedia.org/images/9219097` and `https://radiopaedia.org/cases/loculated-pneumothorax`, respectively. **The second, third and fourth rows**: reconstructed images using 80, 50 and 20 projections.

Remark 3. *To recover an $n \times n$ image by our model* (1.1) *(which is an LP) with N available projections, the corresponding (sparse) coefficient matrix of the equality constraint has at least the size of $Nn \times n^2$. When n and N are large, such a large-scale problem can cause some LP solvers (e.g., Gurobi) to suffer from insufficient memory issues as well as high computational cost on an ordinary PC. We note that another model (based on knowing a prior distribution) that aims to recover objects from a few tomographic projections is suggested in [1, 7]. In their framework, suppose that a 2D object with N projections is available. Then, the decision variable for the corresponding multi-marginal optimal transport problem will be a tensor of the order $2 + N$ in the formulation given in [1]. Hence, it is difficult to implement the model efficiently. In addition, when the order of the tensor is large, the aforementioned model will invariably encounter memory issues. Moreover, our approach do not require any prior knowledge on the image to be recovered, which is another key feature that makes our modeling framework even more attractive.*

5   Concluding remarks

In this paper, we propose a class of linear programming (LP) problems that can be employed to efficiently model several application problems such as discrete tomography and disaggregation of input-output tables in economics. We then develop an implementable inexact entropic proximal point algorithm (iEPPA) for solving these specially structured LPs. To solve the subproblems that contain a special entropic proximal term, we adapt an easy-to-implement dual block coordinate descent (BCD) method to solve the associated more tractable dual subproblem. The convergence of our iEPPA and the R-linear convergence of the dual BCD method are also established. In particular, we develop a new practically verifiable inexact stopping condition for solving the iEPPA subproblem that has some computational advantages over those in the existing methods. Extensive numerical experiments have been conducted to demonstrate the high efficiency and robustness of our iEPPA+BCD framework for solving the capacity constrained multi-marginal optimal transport problem. We also illustrate the potential modeling power of the proposed

model by applying it to discrete tomography problems. Finally, we are aware of the classical works [30, 31] that applied the EPPA with specialized subsolvers for solving the two-stage and multi-stage stochastic network problems. It may be possible to extend our iEPPA+BCD framework for solving such special classes of LP problems. We will leave it as a future research topic.

Acknowledgments

Appendix A   More details on the dual BCD

A.1   Proof of Proposition 1

First, problem (3.1) is equivalent to $\min_X \left\{ \delta_{\Omega^\circ}(X) + \langle C, X \rangle + \varepsilon \mathcal{D}_\phi(X, S) \right\}$. Since $\operatorname{dom} \phi = \mathbb{R}_+^{n_1 \times n_2 \times n_3}$ and thus $\Omega^\circ \cap \operatorname{dom} \phi = \Omega$ is nonempty (by Assumption 2) and bounded, then the objective function in the above problem is level bounded. Thus, a solution exists [35, Theorem 1.9] and must be unique since ϕ is strictly convex. The essential smoothness of ϕ further implies that the optimal solution can only lie in $\mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$. Hence, the optimal solution of problem (3.2) also exists. Let $(\bar{X}, \bar{Z}) \in \mathbb{R}_{++}^{n_1 \times n_2 \times n_3} \times \mathbb{R}_{++}^{n_1 \times n_2 \times n_3}$ be an optimal solution of problem (3.2). Since all constraint functions in (3.2) are affine and the set $\left\{ (X, Z) \in \mathbb{R}^{n_1 \times n_2 \times n_3} \times \mathbb{R}^{n_1 \times n_2 \times n_3} : Z \geq 0 \right\}$ is a convex polyhedron, then it follows from [36, Theorem 3.25] that there exist $\bar{y}^{(i)} \in \mathbb{R}^{m_i}$, $1 \leq i \leq N$ and $\bar{W} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ such that

$$
\begin{cases}
0 = M - \bar{W} - \sum_{i=1}^N \mathcal{A}^{(i,*)} \bar{y}^{(i)} + \varepsilon \log \bar{X}, & \text{(A.1a)} \\
0 \in -\bar{W} + \partial \delta_+(\bar{Z}), & \text{(A.1b)} \\
0 = \boldsymbol{b}^{(i)} - \mathcal{A}^{(i)}(\bar{X}), \quad 1 \leq i \leq N, & \text{(A.1c)} \\
0 = U - \bar{X} - \bar{Z}. & \text{(A.1d)}
\end{cases}
$$

Note from (A.1a) that $\bar{X} = \exp\left(\varepsilon^{-1} \left(\sum_{i=1}^N \mathcal{A}^{(i,*)} \bar{y}^{(i)} + \bar{W} - M \right) \right)$. Then, substituting this and (A.1d) into (A.1b) and (A.1c), recalling (1.4) and the fact that $\partial \delta_+^* = \partial \delta_-$, one can see that

$$
\begin{cases}
0 = \boldsymbol{b}^{(i)} - \mathcal{A}^{(i)}\left(\exp\left(\varepsilon^{-1}\left(\sum_{i=1}^N \mathcal{A}^{(i,*)} \bar{y}^{(i)} + \bar{W} - M \right) \right) \right), & i = 1, \dots, N, \\
0 \in \exp\left(\varepsilon^{-1}\left(\sum_{i=1}^N \mathcal{A}^{(i,*)} \bar{y}^{(i)} + \bar{W} - M \right) \right) - U + \partial \delta_-(\bar{W}).
\end{cases} \tag{A.2}
$$

This together with [36, Theorem 3.5] implies that $\left(\bar{y}^{(1)}, \dots, \bar{y}^{(N)}, \bar{W} \right)$ is an optimal solution of the dual problem (3.4) and hence the optimal solution of problem (3.4) exists.

Moreover, for any optimal solution $\left(\hat{y}^{(1)}, \dots, \hat{y}^{(N)}, \widehat{W} \right)$ of problem (3.4), it follows from [36, Theorem 3.5] that it satisfies the system (A.2) in place of $\left(\bar{y}^{(1)}, \dots, \bar{y}^{(N)}, \bar{W} \right)$. Let $\widehat{X} := \exp\left(\varepsilon^{-1}\left(\sum_{i=1}^N \mathcal{A}^{(i,*)} \hat{y}^{(i)} + \widehat{W} - M \right) \right)$ and $\widehat{Z} := U - \widehat{X}$. By (1.4) and the fact that $\partial \delta_-^* = \partial \delta_+$, it holds that $\left(\widehat{X}, \widehat{Z}, \hat{y}^{(1)}, \dots, \hat{y}^{(N)}, \widehat{W} \right)$ satisfies the system (A.1a)–(A.1d). Thus it follows from [36, Theorem 3.27] that $\left(\widehat{X}, \widehat{Z} \right)$ is an optimal solution of problem (3.2) and hence $\widehat{X}$ is an optimal solution of problem (3.1). This completes the proof.

A.2   Proof of Theorem 2

For the ease of applying the convergence results developed in [28], we first express problem (3.4) in the following compact form:

$$
\min_{\boldsymbol{\chi}} \ \Psi(E\boldsymbol{\chi}) + \langle \boldsymbol{q}, \boldsymbol{\chi} \rangle \quad \text{s.t.} \quad \boldsymbol{\chi} \in \Xi, \tag{A.3}
$$

where $\Psi : \mathbb{R}^{n_1 n_2 n_3} \to \mathbb{R}$ is defined by $\Psi(\boldsymbol{y}) := \varepsilon \sum_i^{n_1 n_2 n_3} \exp((y_i - z_i)/\varepsilon)$, $\boldsymbol{z} := \mathrm{vec}(M) \in \mathbb{R}^{n_1 n_2 n_3}$, $\boldsymbol{q} := -\big[\boldsymbol{b}^{(1)}; \ldots; \boldsymbol{b}^{(N)}; \mathrm{vec}(U)\big] \in \mathbb{R}^{\sum_{i=1}^N m_i + n_1 n_2 n_3}$, $\boldsymbol{\chi} := \big[\boldsymbol{y}^{(1)}; \ldots; \boldsymbol{y}^{(N)}; \mathrm{vec}(W)\big] \in \mathbb{R}^{\sum_{i=1}^N m_i + n_1 n_2 n_3}$, $\Xi := \big\{\boldsymbol{\chi} := \big[\boldsymbol{y}^{(1)}; \ldots; \boldsymbol{y}^{(N)}; \mathrm{vec}(W)\big] : W \leq 0\big\}$ and

$$E := \big[\mathrm{vec}(A_1^{(1)}), \ldots, \mathrm{vec}(A_{m_1}^{(1)}), \ldots, \mathrm{vec}(A_1^{(N)}), \ldots, \mathrm{vec}(A_{m_N}^{(N)}), I_{n_1 n_2 n_3}\big] \in \mathbb{R}^{n_1 n_2 n_3 \times (\sum_{i=1}^N m_i + n_1 n_2 n_3)}.$$

One can easily verify that $\mathrm{dom}\,\Psi = \mathbb{R}^{n_1 n_2 n_3}$ is open and Ψ is strictly convex and twice continuously differentiable on $\mathrm{dom}\,\Psi$.

Moreover, the optimal solution set of problem (A.3) is nonempty (by Proposition 1) and our dual BCD in Algorithm 2 indeed falls into the algorithmic framework in [28] for solving the problem in form of (A.3). Also, note from [28, Lemma 3.3] that the set $\big\{E\boldsymbol{\chi} : \Psi(E\boldsymbol{\chi}) + \langle \boldsymbol{q}, \boldsymbol{\chi} \rangle \leq \alpha, \ \boldsymbol{\chi} \in \Xi\big\}$ is compact for any $\alpha \in \mathbb{R}$. Then, one can easily verify that $\nabla^2 \Psi(E\boldsymbol{\chi}^*)$ is positive definite for any optimal solution $\boldsymbol{\chi}^*$ of problem (A.3). Based on these facts, we can readily apply [28, Theorem 2.1] to obtain statement (i), i.e., $\boldsymbol{\chi}^t := (\boldsymbol{y}^{(1),t}, \ldots, \boldsymbol{y}^{(N),t}, W^t) \to \boldsymbol{\chi}^*$ R-linearly.

We next prove statement (ii). Let $\big\{\big(\hat{\boldsymbol{y}}^{(1)}, \ldots, \hat{\boldsymbol{y}}^{(N)}, \widehat{W}\big)\big\}$ be the limit of $\big\{\big(\boldsymbol{y}^{(1),\ell}, \ldots, \boldsymbol{y}^{(N),\ell}, W^\ell\big)\big\}$. Then, one can see from statement (i) that $\big\{\big(\hat{\boldsymbol{y}}^{(1)}, \ldots, \hat{\boldsymbol{y}}^{(N)}, \widehat{W}\big)\big\}$ is an optimal solution of problem (3.4) and further see from Proposition 1 that $\widehat{X} := \exp\big(\big(\sum_{i=1}^N \mathcal{A}^{(i,*)} \hat{\boldsymbol{y}}^{(i)} + \widehat{W} - M\big)/\varepsilon\big)$ is an optimal solution of problem (3.1). Define the mapping $\mathcal{H} : \mathbb{R}^{\sum_{i=1}^N m_i + n_1 n_2 n_3} \to \mathbb{R}^{n_1 n_2 n_3}$ by $\mathcal{H}(\boldsymbol{\chi}) := \exp((E\boldsymbol{\chi} - \boldsymbol{m})/\varepsilon)$, whose Jacobian matrix is given by $\mathrm{J}\mathcal{H}(\boldsymbol{\chi}) = \varepsilon^{-1}\mathrm{Diag}\big[\exp((E\boldsymbol{\chi} - \boldsymbol{m})/\varepsilon)\big]E$. Then, we see that $\boldsymbol{x}^\ell := \mathrm{vec}(X^\ell) = \mathcal{H}(\boldsymbol{\chi}^\ell)$ and $\hat{\boldsymbol{x}} := \mathrm{vec}(\widehat{X}) = \mathcal{H}(\hat{\boldsymbol{\chi}})$, where $\boldsymbol{\chi}^\ell := \big[\boldsymbol{y}^{(1),\ell}; \ldots; \boldsymbol{y}^{(N),\ell}; \mathrm{vec}(W^\ell)\big]$ and $\hat{\boldsymbol{\chi}} := \big[\hat{\boldsymbol{y}}^{(1)}; \ldots; \hat{\boldsymbol{y}}^{(N)}; \mathrm{vec}(\widehat{W})\big]$. Moreover, we have

$$
\begin{aligned}
\big\|\boldsymbol{x}^\ell - \hat{\boldsymbol{x}}\big\| &= \big\|\mathcal{H}(\boldsymbol{\chi}^\ell) - \mathcal{H}(\hat{\boldsymbol{\chi}})\big\| = \big\|\big(\textstyle\int_0^1 \mathrm{J}\mathcal{H}(\hat{\boldsymbol{\chi}} + \tau(\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}}))\,\mathrm{d}\tau\big) \cdot (\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}})\big\| \\
&\leq \big\|\textstyle\int_0^1 \mathrm{J}\mathcal{H}(\hat{\boldsymbol{\chi}} + \tau(\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}}))\,\mathrm{d}\tau\big\| \cdot \big\|\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}}\big\| \leq \textstyle\int_0^1 \big\|\mathrm{J}\mathcal{H}(\hat{\boldsymbol{\chi}} + \tau(\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}}))\big\|\,\mathrm{d}\tau \cdot \big\|\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}}\big\|,
\end{aligned}
\tag{A.4}
$$

where the second equality follows from the mean-value theorem. Note that

$$\Psi(E\hat{\boldsymbol{\chi}}) + \langle \boldsymbol{q}, \hat{\boldsymbol{\chi}} \rangle \leq \Psi(E\boldsymbol{\chi}^\ell) + \langle \boldsymbol{q}, \boldsymbol{\chi}^\ell \rangle \leq \Psi(E\boldsymbol{\chi}^0) + \langle \boldsymbol{q}, \boldsymbol{\chi}^0 \rangle, \quad \forall \ell \geq 0.$$

It then follows from [28, Lemma 3.3] that $\{E\boldsymbol{\chi}^\ell\}$ is bounded. With this fact, one can easily verify that $\big\|\mathrm{J}\mathcal{H}(\hat{\boldsymbol{\chi}} + \tau(\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}}))\big\|$ is uniformly bounded from above by some constant L, i.e., $\big\|\mathrm{J}\mathcal{H}(\hat{\boldsymbol{\chi}} + \tau(\boldsymbol{\chi}^\ell - \hat{\boldsymbol{\chi}}))\big\| \leq L$ for all $\ell \geq 0$ and $\tau \in [0, 1]$. This together with (A.4) and statement (i) prove statement (ii).

A.3 The dual BCD for the CMOT problem

As a special case of problem (1.1), the 3-marginal capacity constrained optimal transport problem (1.2) (taking the linear mappings defined in (1.3)) has attracted particular attention. In this section, we write down the concrete iterative scheme of the dual BCD in Algorithm 2 for solving (3.1) with the constraints in (1.2). We use $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}, W$ to denote Lagrangian multipliers with respect to the following four constraints

$$
\begin{aligned}
&\textstyle\sum_{s,t} X_{rst} = a_r, \ r = 1, \ldots, n_1, \quad \textstyle\sum_{r,t} X_{rst} = b_s, \ s = 1, \ldots, n_2, \\
&\textstyle\sum_{r,s} X_{rst} = c_t, \ t = 1, \ldots, n_3, \quad X \leq U,
\end{aligned}
$$

respectively. By using similar arguments as in Section 3, one obtains the dual subproblem:

$$
\begin{aligned}
\min_{\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}, W} R\big(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}, W\big) := {}&\varepsilon \textstyle\sum_{r,s,t} \exp\big((f_r + g_s + h_t + W_{rst} - M_{rst})/\varepsilon\big) - \langle \boldsymbol{f}, \boldsymbol{a} \rangle \\
&- \langle \boldsymbol{g}, \boldsymbol{b} \rangle - \langle \boldsymbol{h}, \boldsymbol{c} \rangle - \langle W, U \rangle + \delta_-(W),
\end{aligned}
\tag{A.5}
$$

where $M := C - \varepsilon \log S$. We then apply the BCD method for solving (A.5). Specifically, start from any $(\boldsymbol{f}^0, \boldsymbol{g}^0, \boldsymbol{h}^0, W^0) \in \mathrm{dom}\,R$, at the ℓ-th iteration, compute

$$
\begin{aligned}
\boldsymbol{f}^{\ell+1} &= \arg\min_{\boldsymbol{f}} R\big(\boldsymbol{f}, \boldsymbol{g}^\ell, \boldsymbol{h}^\ell, W^\ell\big), & \boldsymbol{g}^{\ell+1} &= \arg\min_{\boldsymbol{g}} R\big(\boldsymbol{f}^{\ell+1}, \boldsymbol{g}, \boldsymbol{h}^\ell, W^\ell\big), \\
\boldsymbol{h}^{\ell+1} &= \arg\min_{\boldsymbol{h}} R\big(\boldsymbol{f}^{\ell+1}, \boldsymbol{g}^{\ell+1}, \boldsymbol{h}, W^\ell\big), & W^{\ell+1} &= \arg\min_{W} R\big(\boldsymbol{f}^{\ell+1}, \boldsymbol{g}^{\ell+1}, \boldsymbol{h}^{\ell+1}, W\big).
\end{aligned}
$$

After some manipulations, one can obtain the following explicit iterative scheme:

$$
\begin{aligned}
\boldsymbol{f}^{\ell+1} &= \varepsilon \log(\boldsymbol{a}) - \varepsilon \log \left(\left[\sum_{s,t} \exp\left((g_s^\ell + h_t^\ell + W_{rst}^\ell - M_{rst})/\varepsilon \right) \right]_{r=1}^{n_1} \right), \\
\boldsymbol{g}^{\ell+1} &= \varepsilon \log(\boldsymbol{b}) - \varepsilon \log \left(\left[\sum_{r,t} \exp\left((f_r^{\ell+1} + h_t^\ell + W_{rst}^\ell - M_{rst})/\varepsilon \right) \right]_{s=1}^{n_2} \right), \\
\boldsymbol{h}^{\ell+1} &= \varepsilon \log(\boldsymbol{c}) - \varepsilon \log \left(\left[\sum_{r,s} \exp\left((f_r^{\ell+1} + g_s^{\ell+1} + W_{rst}^\ell - M_{rst})/\varepsilon \right) \right]_{t=1}^{n_3} \right), \\
W^{\ell+1} &= \min \left\{ \varepsilon \log U + M - \boldsymbol{f}^{\ell+1} \otimes \mathbf{1}_{n_2} \otimes \mathbf{1}_{n_3} - \mathbf{1}_{n_1} \otimes \boldsymbol{g}^{\ell+1} \otimes \mathbf{1}_{n_3} - \mathbf{1}_{n_1} \otimes \mathbf{1}_{n_2} \otimes \boldsymbol{h}^{\ell+1}, 0 \right\},
\end{aligned}
\tag{A.6}
$$

where $\mathbf{1}_{n_i}$ denotes the n_i-dimensional vector of all ones for $i = 1, 2, 3$. Moreover, let $\widetilde{M} := \exp(-M/\varepsilon)$, $\tilde{\boldsymbol{f}}^\ell := \exp(\boldsymbol{f}^\ell/\varepsilon)$, $\tilde{\boldsymbol{g}}^\ell := \exp(\boldsymbol{g}^\ell/\varepsilon)$, $\tilde{\boldsymbol{h}}^\ell := \exp(\boldsymbol{h}^\ell/\varepsilon)$ and $\widetilde{W}^\ell := \exp(W^\ell/\varepsilon)$. Then, we can equivalently rewrite the iterative scheme (A.6) as

$$
\begin{aligned}
\tilde{\boldsymbol{f}}^{\ell+1} &= \boldsymbol{a} ./ \left(\left[\sum_{s,t} (\widetilde{W}^\ell \circ \widetilde{M})_{rst} \, \tilde{g}_s^\ell \, \tilde{h}_t^\ell \right]_{r=1}^{n_1} \right), \quad \tilde{\boldsymbol{g}}^{\ell+1} = \boldsymbol{b} ./ \left(\left[\sum_{r,t} (\widetilde{W}^\ell \circ \widetilde{M})_{rst} \, \tilde{f}_r^{\ell+1} \, \tilde{h}_t^\ell \right]_{s=1}^{n_2} \right), \\
\tilde{\boldsymbol{h}}^{\ell+1} &= \boldsymbol{c} ./ \left(\left[\sum_{r,s} (\widetilde{W}^\ell \circ \widetilde{M})_{rst} \, \tilde{f}_r^{\ell+1} \, \tilde{g}_s^{\ell+1} \right]_{t=1}^{n_3} \right), \quad \widetilde{W}^{\ell+1} = \min \left\{ (U./\widetilde{M}) ./ (\tilde{\boldsymbol{f}}^{\ell+1} \otimes \tilde{\boldsymbol{g}}^{\ell+1} \otimes \tilde{\boldsymbol{h}}^{\ell+1}), 1 \right\}.
\end{aligned}
\tag{A.7}
$$

In our numerical experiments conducted in Section 4, we always adopt the iterative scheme (A.7) since the proximal parameter ε in our iEPPA does not need to take a small value.

Appendix B Dykstra's algorithm with KL projections

Dykstra's algorithm with <u>K</u>ullback-<u>L</u>eibler projections (DyKL) [5] is adapted in [6] to solve the following entropic regularized capacity constrained optimal transport problem:

$$
\begin{aligned}
\min_{X \in \mathbb{R}^{m \times n}} \quad & \langle C, X \rangle + \varepsilon \sum_{s=1}^m \sum_{r=1}^n X_{rs} (\log X_{rs} - 1) \\
\text{s.t.} \quad & X \mathbf{1}_n = \boldsymbol{a}, \quad X^\top \mathbf{1}_m = \boldsymbol{b}, \quad 0 \le X \le U,
\end{aligned}
\tag{B.1}
$$

where $C \in \mathbb{R}_+^{m \times n}$, $U \in \mathbb{R}_{++}^{m \times n}$, $\boldsymbol{a} := (a_1, \ldots, a_m)^\top \in \Sigma_m$, $\boldsymbol{b} := (b_1, \ldots, b_n)^\top \in \Sigma_n$. Recall the definition of the Kullback-Leibler (KL) divergence between $X \in \mathbb{R}_+^{m \times n}$ and $Y \in \mathbb{R}_{++}^{m \times n}$ is given as follows:

$$
\mathbf{KL}(X, Y) = \sum_{r,s} \left(x_{rs} \log (x_{rs}/y_{rs}) - x_{rs} + y_{rs} \right).
$$

Moreover, given a convex set $\mathcal{S} \subseteq \mathbb{R}^{m \times n}$ and $Y \in \mathbb{R}_{++}^{m \times n}$, the projection associated with the KL divergence (called KL projection) is defined as $\mathrm{Proj}_{\mathcal{S}}^{\mathbf{KL}}(Y) := \arg \min_{X \in \mathcal{S}} \mathbf{KL}(X, Y)$. Thus, problem (B.1) can be equivalently reformulated as

$$
\min_{X \in \mathbb{R}^{m \times n}} \quad \mathbf{KL}(X, K) \qquad \text{s.t.} \qquad X \in \mathcal{S}_1 \cap \mathcal{S}_2 \cap \mathcal{S}_3,
$$

where $K := e^{-C/\varepsilon}$ is the kernel matrix, $\mathcal{S}_1 := \{X \in \mathbb{R}^{m \times n} : X \mathbf{1}_n = \boldsymbol{a}\}$, $\mathcal{S}_2 := \{X \in \mathbb{R}^{m \times n} : X^\top \mathbf{1}_m = \boldsymbol{b}\}$ and $\mathcal{S}_3 := \{X \in \mathbb{R}^{m \times n} : X \le U\}$. Then, the DyKL is presented as follows: let $X^0 = K$, $Q_1^0 = Q_2^0 = Q_3^0 = \mathbf{1}_m \mathbf{1}_n^\top$, then for $k \ge 0$, compute

$$
\begin{aligned}
\Pi_0^{k+1} &= X^k, \\
\Pi_1^{k+1} &= \mathrm{Proj}_{\mathcal{S}_1}^{\mathbf{KL}}(\Pi_0^{k+1} \circ Q_1^k), \quad Q_1^{k+1} = Q_1^k \circ (\Pi_0^{k+1} ./ \Pi_1^{k+1}), \\
\Pi_2^{k+1} &= \mathrm{Proj}_{\mathcal{S}_2}^{\mathbf{KL}}(\Pi_1^{k+1} \circ Q_2^k), \quad Q_2^{k+1} = Q_2^k \circ (\Pi_1^{k+1} ./ \Pi_2^{k+1}), \\
\Pi_3^{k+1} &= \mathrm{Proj}_{\mathcal{S}_3}^{\mathbf{KL}}(\Pi_2^{k+1} \circ Q_3^k), \quad Q_3^{k+1} = Q_3^k \circ (\Pi_2^{k+1} ./ \Pi_3^{k+1}), \\
X^{k+1} &= \Pi_3^{k+1},
\end{aligned}
\tag{B.2}
$$

where ∘ denotes the Hadamard product. Note that the above iterative scheme is a slightly different but an equivalent form of the DyKL used in [6]. We adapt it here because it is more explicit and convenient for comparison. Moreover, by simple calculations, one can verify that

$$\Pi_1^{k+1} = \text{Proj}_{\mathcal{S}_1}^{\mathbf{KL}}(\Pi_0^{k+1} \circ Q_1^k) = \text{Diag}\Big(\boldsymbol{a}./\big((\Pi_0^{k+1} \circ Q_1^k)\mathbf{1}_n\big)\Big)\big(\Pi_0^{k+1} \circ Q_1^k\big),$$

$$\Pi_2^{k+1} = \text{Proj}_{\mathcal{S}_2}^{\mathbf{KL}}(\Pi_1^{k+1} \circ Q_2^k) = \big(\Pi_1^{k+1} \circ Q_2^k\big)\text{Diag}\Big(\boldsymbol{b}./\big((\Pi_1^{k+1} \circ Q_2^k)^\top \mathbf{1}_m\big)\Big),$$

$$\Pi_3^{k+1} = \text{Proj}_{\mathcal{S}_3}^{\mathbf{KL}}(\Pi_2^{k+1} \circ Q_3^k) = \min\big\{\Pi_2^{k+1} \circ Q_3^k,\, U\big\}.$$

It is worth noting that the DyKL in (B.2) may suffer from severe numerical issues when ε takes a small value. Thus, one may need to carry out the computations of Π_i^k and Q_i^k $(i = 1, 2, 3)$ in the log domain to alleviate the numerical instability. Specifically, by taking logarithm on both sides of above equations and letting $\widetilde{X}^k := \varepsilon \log X^k$, $\widetilde{\Pi}_i^k := \varepsilon \log \Pi_i^k$, $\widetilde{Q}_i^k := \varepsilon \log Q_i^k$, $\tilde{\boldsymbol{a}} := \varepsilon \log \boldsymbol{a}$, $\widetilde{U} := \varepsilon \log U$, we obtain after some manipulations that

$$\widetilde{\Pi}_0^{k+1} = \widetilde{X}^k,$$

$$\widetilde{\Pi}_1^{k+1} = \Big[\tilde{\boldsymbol{a}} - \varepsilon \log\Big(\big[\exp\big((\widetilde{\Pi}_0^{k+1} + \widetilde{Q}_1^k)/\varepsilon\big)\big]\mathbf{1}_n\Big)\Big]\mathbf{1}_n^\top + \widetilde{\Pi}_0^{k+1} + \widetilde{Q}_1^k, \qquad \widetilde{Q}_1^{k+1} = \widetilde{Q}_1^k + \widetilde{\Pi}_0^{k+1} - \widetilde{\Pi}_1^{k+1},$$

$$\widetilde{\Pi}_2^{k+1} = \mathbf{1}_m\Big[\tilde{\boldsymbol{b}} - \varepsilon \log\Big(\big[\exp\big((\widetilde{\Pi}_1^{k+1} + \widetilde{Q}_2^k)/\varepsilon\big)\big]^\top \mathbf{1}_m\Big)\Big]^\top + \widetilde{\Pi}_1^{k+1} + \widetilde{Q}_2^k, \quad \widetilde{Q}_2^{k+1} = \widetilde{Q}_2^k + \widetilde{\Pi}_1^{k+1} - \widetilde{\Pi}_2^{k+1},$$

$$\widetilde{\Pi}_3^{k+1} = \min\big\{\widetilde{\Pi}_2^{k+1} + \widetilde{Q}_3^k,\, \widetilde{U}\big\}, \qquad\qquad\qquad\qquad\qquad\qquad\qquad \widetilde{Q}_3^{k+1} = \widetilde{Q}_3^k + \widetilde{\Pi}_2^{k+1} - \widetilde{\Pi}_3^{k+1},$$

$$\widetilde{X}^{k+1} = \widetilde{\Pi}_3^{k+1}.$$

$$\text{(B.3)}$$

In this stabilization framework, the initialization is set to $\widetilde{X}^0 = -C$ and $\widetilde{Q}_1^0 = \widetilde{Q}_2^0 = \widetilde{Q}_3^0 = 0$. When checking the primal feasibility accuracy, we recover X^{k+1} by setting $X^{k+1} = \exp(\widetilde{X}^{k+1}/\varepsilon)$.

Appendix C Construction of a tomographic projection

Let p be a nonnegative integer. We consider the following four directions

$$\vec{v} = (1,\, p),\ (1,\, -p),\ (p,\, 1) \text{ and } (p,\, -1).$$

Note that when $p \in \{0, 1\}$, we only have two directions. The process to find the projection $\mathcal{A}^{(i)}(X)$ along a given direction $\vec{v}$ is described as follows (see Figure 6 for a concrete example):

1. Plot the entries of X as points on the integer grid $\{1, \ldots, n\} \times \{1, \ldots, n\}$.

2. For each point, draw a line v_j parallelling to $\vec{v}$, identify all other points for which v_j passes through.

3. Take the sum of the entries of X for all points on v_j to define $(\mathcal{A}^{(i)}(X))_j$.

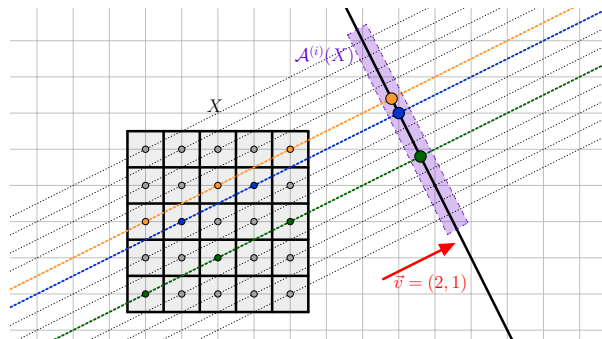4. Repeat this process until all $\{\mathsf{v}_j\}$ covers the whole grid, i.e., covers all entries of X.

Figure 6: Construction of the projection operator along $\vec{v} = (2, 1)$ for a 5×5 matrix X.

Declarations

Data availability Not applicable.

References

[1] I. Abraham, R. Abraham, M. Bergounioux, and G. Carlier. Tomographic reconstruction from a few views: A multi-marginal optimal transport approach. *Appl. Math. Optim.*, 75(1):55–73, 2017.

[2] J. Altschuler, J. Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[3] A. Auslender and M. Haddou. An interior-proximal method for convex linearly constrained problems and its extension to variational inequalities. *Math. Program.*, 71(1):77–100, 1995.

[4] H.H. Bauschke and J.M. Borwein. Legendre functions and the method of random Bregman projections. *J. Convex Anal.*, 4(1):27–67, 1997.

[5] H.H. Bauschke and A.S. Lewis. Dykstras algorithm with Bregman projections: A convergence proof. *Optimization*, 48(4):409–427, 2000.

[6] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM J. Sci. Comput.*, 37(2):A1111–A1138, 2015.

[7] M. Bergounioux, I. Abraham, R. Abraham, G. Carlier, E. Le Pennec, and E. Trélat. Variational methods for tomographic reconstruction with few views. *Milan J. Math.*, 86(2):157–200, 2018.

[8] D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific Belmont, MA, 1997.

[9] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. Math. Phys.*, 7(3):200–217, 1967.

[10] Y. Censor and A. Lent. An iterative row-action method for interval convex programming. *J. Optim. Theory Appl.*, 34(3):321–353, 1981.

[11] Y. Censor and S.A. Zenios. Proximal minimization algorithm with D-functions. *J. Optim. Theory Appl.*, 73(3):451–464, 1992.

[12] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM J. Optim.*, 3(3):538–543, 1993.

[13] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

[14] R.L. Dykstra. An algorithm for restricted least squares regression. *J. Am. Stat. Assoc.*, 78(384):837–842, 1983.

[15] J. Eckstein. Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming. *Math. Oper. Res.*, 18(1):202–226, 1993.

[16] J. Eckstein. Approximate iterations in Bregman-function-based proximal algorithms. *Math. Prog.*, 83(1-3):113–123, 1998.

[17] P.P.B Eggermont. Multiplicative iterative algorithms for convex programming. *Linear Algebra Appl.*, 130:25–42, 1990.

[18] A. Grandy and L. Veraart. Bayesian methodology for systemic risk assessment in financial networks. *Manage. Sci.*, 63:3999–4446, 2017.

[19] A.J. Hoffman. On approximate solutions of systems of linear inequalities. *J. Res. Natl. Bur. Stand.*, 49(4):263–265, 1952.

[20] V. Holy and K. Safr. Disaggregating input-output tables by the multidimensional RAS method. *arXiv preprint arXiv:1704.07814v2*, 2019.

[21] A.N. Iusem, B.F. Svaiter, and M. Teboulle. Entropy-like proximal methods in convex programming. *Math. Oper. Res.*, 19(4):790–814, 1994.

[22] A.N. Iusem and M. Teboulle. Convergence rate analysis of nonquadratic proximal methods for convex and linear programming. *Math. Oper. Res.*, 20(3):657–677, 1995.

[23] J. Kennington and M. Shalaby. An effective subgradient procedure for minimal cost multicommodity flow problems. *Manage. Sci.*, 23(9):994–1004, 1977.

[24] J. Korman and R.J. McCann. Insights into capacity-constrained optimal transport. *Proc. Natl. Acad. Sci.*, 110(25):10064–10067, 2013.

[25] J. Korman and R.J. McCann. Optimal transportation with capacity constraints. *Trans. Am. Math. Soc.*, 367(3):1501–1521, 2015.

[26] V.L. Levin. The problem of mass transfer in a topological space and probability measures with given marginal measures on the product of two spaces. *Dokl. Akad. Nauk SSSR*, 276(5):1059–1064, 1984.

[27] T. Lin, N. Ho, M. Cuturi, and M.I. Jordan. On the complexity of approximating multimarginal optimal transport. *To appear in J. Mach. Learn. Res.*, 2022.

[28] Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *J. Optim. Theory Appl.*, 72(1):7–35, 1992.

[29] Z.-Q. Luo and P. Tseng. On the convergence rate of dual ascent methods for linearly constrained convex minimization. *Math. Oper. Res.*, 18(4):846–867, 1993.

[30] S. S Nielsen and S. A Zenios. Massively parallel proximal algorithms for solving linear stochastic network programs. *The Int. J. Supercomput Appl.*, 7(4):349–364, 1993.

[31] S. S Nielsen and S. A Zenios. Solving multistage stochastic network programs on massively parallel computers. *Math. Program.*, 73(3):227–250, 1996.

[32] G. Peyré and M. Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11(5-6):355–607, 2019.

[33] B. T. Polyak. *Introduction to optimization.* Optimization Software Inc., New York, 1987.

[34] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, 1970.

[35] R. T. Rockafellar and R. J-B. Wets. *Variational Analysis.* Springer, 1998.

[36] A. Ruszczyński. *Nonlinear Optimization.* Princeton University Press, Princeton, 2006.

[37] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *Am. Math. Mon.*, 74(4):402–405, 1967.

[38] M. Teboulle. Entropic proximal mappings with applications to nonlinear programming. *Math. Oper. Res.*, 17(3):670–690, 1992.

[39] M. Teboulle. Convergence of proximal-like algorithms. *SIAM J. Optim.*, 7(4):1069–1083, 1997.

[40] R.J. Tibshirani. Dykstra's algorithm, ADMM, and coordinate descent: Connections, insights, and extensions. In *Advances in Neural Information Processing Systems*, pages 517–528, 2017.

[41] P. Tseng. Dual coordinate ascent methods for non-strictly convex minimization. *Math. Program.*, 59(1-3):231–247, 1993.

[42] S. Weber, C. Schnörr, T. Schüle, and J. Hornegger. Binary tomography by iterating linear programs. In *Geometric Properties for Incomplete Data*, pages 183–197, 2006.

[43] Y. Xie, X. Wang, R. Wang, and H. Zha. A fast proximal point method for computing exact Wasserstein distance. In *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, pages 433–453, 2020.