Computational Hardness and Fast Algorithm for Fixed-Support Wasserstein Barycenter

Tianyi Lin[‡] Nhat Ho[⋄] Xi Chen[⋆] Marco Cuturi[◁,▷] Michael I. Jordan[⋄,†]

Department of Industrial Engineering and Operations Research[‡]
Department of Electrical Engineering and Computer Sciences[⋄]
Department of Statistics[†]
University of California, UC Berkeley
Stern School of Business, New York University[⋆]
CREST - ENSAE[◁], Google Brain[▷]

April 30, 2020

Abstract

We study in this paper the fixed-support Wasserstein barycenter problem (FS-WBP), which consists in computing the Wasserstein barycenter of m discrete probability measures supported on a finite metric space of size n. We show first that the constraint matrix arising from the standard linear programming (LP) representation of the FS-WBP is *not totally unimodular* when $m \geq 3$ and $n \geq 3$. This result answers an open question pertaining to the relationship between the FS-WBP and the minimum-cost flow (MCF) problem since it therefore proves that the FS-WBP in the standard LP form is not a MCF problem when $m \geq 3$ and $n \geq 3$. We also develop a provably fast *deterministic* variant of the celebrated iterative Bregman projection (IBP) algorithm, named FastIBP algorithm, with the complexity bound of $\widetilde{O}(mn^{7/3}\varepsilon^{-4/3})$ where $\varepsilon \in (0, 1)$ is the tolerance. This complexity bound is better than the best known complexity bound of $\widetilde{O}(mn^2\varepsilon^{-2})$ from the IBP algorithm in terms of ε, and that of $\widetilde{O}(mn^{5/2}\varepsilon^{-1})$ from other accelerated algorithms in terms of n. Finally, we conduct extensive experiments with both synthetic and real data and demonstrate the favorable performance of the FastIBP algorithm in practice.

1 Introduction

During the past decade, the Wasserstein barycenter problem [Agueh and Carlier, 2011] (WBP) has served as a foundation for numerous applications ranging from economics [Carlier and Ekeland, 2010, Chiappori et al., 2010] and physics [Buttazzo et al., 2012, Cotar et al., 2013, Trouvé and Younes, 2005] to statistics [Munch et al., 2015, Ho et al., 2017, Srivastava et al., 2018], image and shape analysis [Rabin et al., 2011, Bonneel et al., 2015, 2016] and machine learning [Cuturi and Doucet, 2014]. Among all of these applications, a key challenge is to design provably fast algorithms for the WBP and further understand the computational hardness [Peyré and Cuturi, 2019].

The WBP has strong connection with the optimal transport (OT) problem since they both depend on the Wasserstein distance. Compared to the OT problem which computes the Wasserstein distance between two probability measures, the WBP is harder in that it requires to minimize the sum of the Wasserstein distance, and typically considers $m \geq 2$ probability measures. In that sense, its closest relative is the multimarginal optimal transport problem [Gangbo and Swiech, 1998], which also compares m measures. A comprehensive treatment of OT and its applications is beyond the scope of our work; see Villani [2008], Peyré and Cuturi [2019] for an introduction. Since Cuturi [2013] showed that the Sinkhorn algorithm efficiently solves the OT problem, numerous efforts have been made in this direction [Cuturi and Peyré, 2016, Genevay et al., 2016, Altschuler et al., 2017, Dvurechensky et al., 2018, Blanchet et al., 2018, Lin et al., 2019b, Lahn et al., 2019, Quanrud, 2019, Jambulapati et al., 2019, Lin et al., 2019c]. The best-known theoretical complexity bound is $\mathcal{O}(n^2\varepsilon^{-1})$ [Blanchet et al., 2018, Quanrud, 2019, Lahn et al., 2019, Jambulapati et al., 2019] while Sinkhorn and Greenkhorn algorithms [Altschuler et al., 2017, Dvurechensky et al., 2018, Lin et al., 2019b] serve as the baseline approaches in practice. Recently, Lin et al. [2019a] have provided the complexity of approximating the multimarginal OT problem.

There have been much efforts devoted to the development of fast algorithms when considering $m \geq 2$ discrete probability measures [Rabin et al., 2011, Cuturi and Doucet, 2014, Carlier et al., 2015, Bonneel et al., 2015, Benamou et al., 2015, Anderes et al., 2016, Staib et al., 2017, Ye et al., 2017, Borgwardt and Patterson, 2018, Puccetti et al., 2018, Claici et al., 2018, Uribe et al., 2018, Dvurechenskii et al., 2018, Yang et al., 2018, Le et al., 2019, Kroshnin et al., 2019, Guminov et al., 2019, Ge et al., 2019, Borgwardt and Patterson, 2019]. To the best of our knowledge, Rabin et al. [2011] were the first to propose an algorithm to compute Wasserstein barycenters, but did so using the sliced-Wasserstein distance, which was an approximation of the Wasserstein distance and went on to find several other usages. Cuturi and Doucet [2014] proposed to smooth the WBP using an entropic regularization, leading to the simple gradient-descent scheme that was later improved and simplified to yield generalized Sinkhorn-type projections under the name of the iterative Bregman projection (IBP) algorithm [Benamou et al., 2015, Kroshnin et al., 2019]. Several contributions have since been proposed, from semi-dual gradient descent [Cuturi and Peyré, 2016, 2018], accelerated primal-dual gradient descent (APDAGD) [Dvurechenskii et al., 2018, Kroshnin et al., 2019], accelerated IBP [Guminov et al., 2019], stochastic gradient descent [Claici et al., 2018], distributed and parallel gradient descent [Staib et al., 2017, Uribe et al., 2018], alternating direction method of multipliers (ADMM) [Ye et al., 2017, Yang et al., 2018] and interior-point algorithm [Ge et al., 2019]. Very recently, Kroshnin et al. [2019], Guminov et al. [2019] have proposed a novel primal-dual framework, allowing the complexity bound analysis for various algorithms, e.g., IBP, accelerated IBP and APDAGD.

Concerning the computational hardness of the WBP with free support, Anderes et al. [2016] proved that the barycenter of m empirical measures is also an empirical measure with at most the total number of points contained in all of the measures minus $m-1$. When $m=2$ and the measures are bound and the support is fixed, the computation of the barycenter amounts to solving a network flow problem on a directed graph. Borgwardt and Patterson [2019] proved that finding a barycenter of sparse support is NP-hard even under the simple setting when $m=3$. However, their analysis does not work for the WBP with fixed support, namely when the considered m probability measures have the pre-specified support.

Contribution. We revisit the fixed-support Wasserstein barycenter problem (FS-WBP) between m discrete probability measures supported on the prespecified finite set of n points. Our contributions can be summarized as follows:

1. We prove that the FS-WBP in the standard LP form is not a minimum-cost flow (MCF) problem in general. In particular, we show that the constraint matrix arising from the standard LP representation of the FS-WBP is (i) totally unimodular when $m \geq 3$ and $n=2$ and (ii) not totally unimodular when $m \geq 3$ and $n \geq 3$. Our results shed light on *the necessity of problem reformulation*, which has been suggested by Cuturi and Doucet

[2014], Benamou et al. [2015] for using the entropic regularization and IBP algorithm and Ge et al. [2019] for using the block structure and interior-point algorithm.

2. We propose a provably fast *deterministic* variant of the celebrated iterative Bregman projection (IBP) algorithm, named FastIBP algorithm, with the complexity bound of $\widetilde{O}(mn^{7/3}\varepsilon^{-4/3})$ where ε stands for the tolerance. This improves over the complexity bound of $\widetilde{O}(mn^2\varepsilon^{-2})$ from the IBP algorithm [Benamou et al., 2015] in terms of ε and the complexity bound of $\widetilde{O}(mn^{5/2}\varepsilon^{-1})$ from the accelerated IBP and APDAGD algorithms [Kroshnin et al., 2019, Guminov et al., 2019] in terms of n. Extensive experimental results on both synthetic and real data demonstrate the favorable performance of the FastIBP algorithm over the IBP algorithm in practice.

Organization. The remainder of the paper is organized as follows. In Section 2, we provide the basic setup for the entropic regularized FS-WBP and the dual problem. In Section 3, we present the computational hardness results of the FS-WBP in the standard LP form. In Sections 4, we propose and analyze the FastIBP algorithm. Experimental results on both synthetic and real data are presented in Section 5. Finally, we conclude the paper in Section 6.

Notation. We let $[n]$ be the set $\{1, 2, \ldots, n\}$ and $\mathbb{R}^n_+$ be the set of all vectors in $\mathbb{R}^n$ with nonnegative components. $\mathbf{1}_n$ and $\mathbf{0}_n$ are the n-vectors of ones and zeros. Δ^n stands for the probability simplex: $\Delta^n = \{u \in \mathbb{R}^n_+ : \mathbf{1}_n^\top u = 1\}$. For a differentiable function f, we denote ∇f and $\nabla_\lambda f$ for the full gradient of f and its gradient with respect to a variable λ. For $x \in \mathbb{R}^n$ and $1 \leq p \leq \infty$, we write $\|x\|_p$ for its ℓ_p-norm. For $X = (X_{ij}) \in \mathbb{R}^{n\times n}$, the notations $\text{vec}\,(X) \in \mathbb{R}^{n^2}$ and $\det(X)$ stand for the vector representation and the determinant. In addition, $\|X\|_\infty = \max_{1\leq i,j\leq n} |X_{ij}|$, $\|X\|_1 = \sum_{1\leq i,j\leq n} |X_{ij}|$, $r(X) = X\mathbf{1}_n$ and $c(X) = X^\top \mathbf{1}_n$. Let $X, Y \in \mathbb{R}^{n\times n}$, the Frobenius and Kronecker inner product are denoted by $\langle X, Y\rangle := \sum_{1\leq i,j\leq n} X_{ij}Y_{ij}$ and $X \otimes Y \in \mathbb{R}^{n^2\times n^2}$. Given the dimension n and accuracy ε, the notation $a = O(b(n, \varepsilon))$ stands for the upper bound $a \leq C \cdot b(n, \varepsilon)$ where $C > 0$ is independent of n and ε, and $a = \widetilde{O}(b(n, \varepsilon))$ indicates the previous inequality where C depends only the logarithmic factors of n and ε.

2 Preliminaries and Technical Background

In this section, we provide the basic setup of the fixed-support Wasserstein barycenter problem (FS-WBP), starting with the standard linear programming (LP) presentation and entropic regularized formulation with a formal specification of an approximate barycenter.

2.1 Linear programming formulation

For $p \geq 1$, let $\mathcal{P}_p(\Omega)$ be the set of Borel probability measures on Ω with finite p-th moment. The Wasserstein distance of order $p \geq 1$ [Villani, 2008] between $\mu, \nu \in \mathcal{P}_p(\Omega)$ is defined by

$$W_p(\mu, \nu) := \left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{\Omega\times\Omega} d^p(\mathbf{x}, \mathbf{y})\ \pi(d\mathbf{x}, d\mathbf{y})\right)^{1/p},$$

where $d(\cdot, \cdot)$ is a metric on Ω and $\Pi(\mu, \nu)$ is the set of couplings between μ and ν. Given a weight vector $(\omega_1, \omega_2, \ldots, \omega_m) \in \Delta^m$ for $m \geq 2$, the *Wasserstein barycenter* [Agueh and Carlier, 2011] of m probability measures $\{\mu_k\}_{k=1}^m$ is a solution of the following functional minimization problem

$$\min_{\mu\in\mathcal{P}_p(\Omega)} \sum_{k=1}^m \omega_k W_p^p(\mu, \mu_k). \tag{1}$$

Because our goal is to provide computational schemes to approximately solve the WBP, we first adopt the definition of an ε-approximate solution.

Definition 2.1. *The probability measure $\widehat{\mu} \in \mathcal{P}_p(\Omega)$ is called an ε-*approximate barycenter *if* $\sum_{k=1}^m \omega_k W_p^p(\widehat{\mu}, \mu_k) \leq \sum_{k=1}^m \omega_k W_p^p(\mu^\star, \mu_k) + \varepsilon$ *where $\mu^\star$ is an optimal solution to problem* (1).

There are two main settings: (i) *free-support Wasserstein barycenter*, namely, when we optimize both the weights and supports of the barycenter in Eq. (1); (ii) *fixed-support Wasserstein barycenter*, namely, when the supports of the barycenter are similar to those from the probability measures $\{\mu_k\}_{k=1}^m$ and we optimize the weights of the barycenter in Eq. (1).

The free-support WBP problem is notoriously difficult to solve. It can either be solved using a solution to the multimarginal-OT (MOT) problem, as described in detail by Agueh and Carlier [2011], or approximated using alternative optimization techniques. Assuming each measure is supported on n distinct points, the WBP problem can be solved *exactly* by solving first a MOT, to then compute $(n-1)m+1$ barycenters of *points* in Ω (these barycenters are exactly the support of the barycentric measure). Solving a MOT is however equivalent to solving an LP with n^m variables and $(n-1)m+1$ constraints. The other route, alternative optimization, requires setting an initial guess for the barycenter, a discrete measure supported on k weighted points (where k is predefined). One can then proceed by updating the locations of μ (or even add new ones) to decrease the objective function in Eq. (1), before changing their weights. In the Euclidean setting with $p=2$, the free-support WBP is closely related to the clustering problem, and equivalent to k-means when $m=1$ [Cuturi and Doucet, 2014]. Whereas solving the free-support WBP using MOT results in a convex (yet intractable) problem, the alternating mimimization approach is not, in very much the same way that the k-means problem is not, and results in the minimization of a piece-wise quadratic function. *On the other hand, the fixed-support WBP is comparatively easier to solve, and as such as played a role in several real-world applications.* For instance, in imaging sciences, pixels and voxels are supported on a predefined, finite grid. In these applications, the barycenter and μ_k measures share the same support.

In view of this, throughout the remainder of the paper, we let $(\mu_k)_{k=1}^m$ be discrete probability measures and the support points $\{\mathbf{x}_i^k\}_{i\in[n]}$ are fixed. Since $\{\mu_k\}_{k=1}^m$ have the fixed support, they are fully characterized by the weights $\{u^k\}_{k=1}^m$. Accordingly, the support of the barycenter $\{\widehat{\mathbf{x}}_i\}_{i\in[n]}$ is also fixed and can be prespecified by $\{\mathbf{x}_i^k\}_{i\in[n]}$ in many applications. To this end, the FS-WBP between $\{\mu_k\}_{k=1}^m$ has the standard LP representation [Cuturi and Doucet, 2014, Benamou et al., 2015, Peyré and Cuturi, 2019] as follows,

$$
\begin{aligned}
\min_{\{X_i\}_{i=1}^m \subseteq \mathbb{R}_+^{n\times n}} \quad & \sum_{k=1}^m \omega_k \langle C_k, X_k\rangle, \\
\text{s.t.} \quad & r(X_k) = u^k, \ X_k \geq 0, \ \text{for all } k \in [m], \\
& c(X_{k+1}) = c(X_k), \ \text{for all } k \in [m-1],
\end{aligned} \tag{2}
$$

where $\{X_k\}_{k=1}^m$ and $\{C_k\}_{k=1}^m \subseteq \mathbb{R}_+^{n\times\ldots\times n}$ denote a set of *transportation plans* and *nonnegative cost matrices* and $(C_k)_{ij} = d^p(\mathbf{x}_i^k, \widehat{\mathbf{x}}_j)$ for all $k \in [m]$. The fixed-support Wasserstein barycenter $u \in \Delta_n$ is determined by the weight $\sum_{k=1}^m \omega_k c(X_k)$ and the support $(\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2, \ldots, \widehat{\mathbf{x}}_n)$.

From Eq. (2), we see the FS-WBP is a linear programming with $2mn-n$ equality constraints and mn^2 variables. This inspires various works on solving the FS-WBP using classical optimization algorithms [Ge et al., 2019, Yang et al., 2018]. Despite some progresses, the understanding of the structure of FS-WBP has remained limited. While the OT problem [Villani, 2008] is equivalent to a minimum-cost flow (MCF) problem, it remains unknown whether the FS-WBP is a MCF problem even under the simplest setting when $m = 2$.

2.2 Entropic regularized FS-WBP

Using Cuturi's entropic approach to the OT problem [Cuturi, 2013], we define a regularized version of the FS-WBP in Eq. (2), where an entropic regularization is added to the Wasserstein barycenter objective. The resulting formulation is as follows:

$$\begin{aligned} \min_{\{X_i\}_{i=1}^m \subseteq \mathbb{R}^{n\times n}} \quad & \sum_{k=1}^m \omega_k(\langle C_k, X_k\rangle - \eta H(X_k)), \\ \text{s.t.} \quad & r(X_k) = u^k, \ X_k \geq 0, \ \text{for all } k \in [m], \\ & c(X_{k+1}) = c(X_k), \ \text{for all } k \in [m-1], \end{aligned} \tag{3}$$

where $\eta > 0$ is the parameter and $H(X)$ denotes the entropic regularization term:

$$H(X) \ := \ -\langle X, \log(X) - \mathbf{1}_n\mathbf{1}_n^\top\rangle.$$

We refer to Eq. (3) as *entropic regularized FS-WBP*. When η is large, the optimal value of entropic regularized FS-WBP may yield a poor approximation of the cost of the FS-WBP. In order to guarantee a good approximation, we scale the parameter η as a function of the desired accuracy of the approximation.

Definition 2.2. *The probability vector $\widehat{u} \in \Delta^n$ is called an* ε-approximate barycenter *if there exists a feasible solution $(\widehat{X}_1, \widehat{X}_2, \ldots, \widehat{X}_m) \in \mathbb{R}_+^{n\times n} \times \cdots \times \mathbb{R}_+^{n\times n}$ for the FS-WBP in Eq. (2) such that $\widehat{u} = \sum_{k=1}^m \omega_k c(\widehat{X}_k)$ for all $k \in [m]$ and $\sum_{k=1}^m \omega_k\langle C_k, \widehat{X}_k\rangle \leq \sum_{k=1}^m \omega_k\langle C_k, X_k^\star\rangle + \varepsilon$ where $(X_1^\star, X_2^\star, \ldots, X_m^\star)$ is an optimal solution of the FS-WBP in Eq. (2).*

With these notions and definitions in mind, we develop efficient algorithms for approximately solving the FS-WBP where the running time of our algorithms required to obtain an ε-approximate barycenter achieves at least competitive dependence on m, n and ε than other state-of-the-art algorithms [Kroshnin et al., 2019, Guminov et al., 2019].

2.3 Dual entropic regularized FS-WBP

Using the duality theory in convex optimization [Rockafellar, 1970], the dual form of entropic regularized FS-WBP has been derived in Cuturi and Doucet [2014], Kroshnin et al. [2019]. Different from the usual 2-marginal or multimarginal OT [Cuturi and Peyré, 2018, Lin et al., 2019a], *the dual entropic regularized FS-WBP* is a convex optimization problem with an affine constraint set. Let $(B_k)_{ij} = e^{\lambda_{ki} + \tau_{kj} - \eta^{-1}(C_k)_{ij}}$ for all $i, j \in [n]$ and $k \in [m]$, we have

$$\min_{\lambda, \tau \in \mathbb{R}^{mn}} \varphi(\lambda, \tau) := \sum_{k=1}^m \omega_k(\mathbf{1}_n^\top B_k(\lambda_k, \tau_k)\mathbf{1}_n - \lambda_k^\top u^k), \quad \text{s.t.} \ \sum_{k=1}^m \omega_k\tau_k = \mathbf{0}_n. \tag{4}$$

To derive the dual problem with $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ with $\alpha_k \in \mathbb{R}^n$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_{m-1})$ with $\beta_k \in \mathbb{R}^n$, we define the Lagrangian function as follows:

$$\mathcal{L}(\{X_k\}_{k\in[m]}, \alpha, \beta) = \sum_{k=1}^m \omega_k(\langle C_k, X_k\rangle - \eta H(X_k)) - \sum_{k=1}^m \alpha_k^\top(r(X_k) - u^k) - \sum_{k=1}^{m-1} \beta_k^\top(c(X_{k+1}) - c(X_k)).$$

Taking the derivative and setting the resulting equations to zero, we obtain that the optimal solution $\bar{X}_k = \bar{X}_k(\alpha, \beta)$ for all $k \in [m]$ has the following form:

$$(\bar{X}_k)_{ij} = e^{\eta^{-1}(\omega_k^{-1}(\alpha_{ki} + \beta_{k-1,j}) - \beta_{kj} - (C_k)_{ij})} \quad \text{for all } k \in [m],$$

with the convention $\beta_0 = \beta_m = 0$. Performing the change of variable $\lambda_k = (\eta\omega_k)^{-1}\alpha_k$ and $\tau_k = (\eta\omega_k)^{-1}(\beta_{k-1} - \beta_k)$, we have $\sum_{k=1}^m \omega_k \tau_k = \mathbf{0}_n$ and

$$\min_{X_1, \ldots, X_k} \mathcal{L}(X_1, \ldots, X_k, \alpha, \beta) \ = \ -\varphi(\lambda, \tau).$$

Putting these pieces together yields the desired problem formulation in Eq. (4).

Finally, we provides in the following lemma an upper bound for the ℓ_∞-norm of one optimal solution $(\lambda^\star, \tau^\star)$ of the dual entropic regularized FS-WBP.

Lemma 2.1. *For the dual entropic regularized FS-WBP, let* $\bar{C} = \max_{1 \leq k \leq m} \|C_k\|_\infty$ *and* $\bar{u} = \min_{1 \leq k \leq m, 1 \leq j \leq n} u_{kj}$, *there exists an optimal solution* $(\lambda^\star, \tau^\star)$ *such that the following* ℓ_∞*-norm bound holds true:*

$$\|\lambda_k^\star\|_\infty \leq R_\lambda, \quad \|\tau_k^\star\|_\infty \leq R_\tau, \quad \text{for all } k \in [m],$$

where $R_\lambda = 5\eta^{-1}\bar{C} + \log(n) - \log(\bar{u})$ *and* $R_\tau = 4\eta^{-1}\bar{C}$.

Proof. First, we claim that it holds for any optimal solution $(\lambda^\star, \tau^\star)$ to the dual entropic regularized FS-WBP in Eq. (4),

$$\lambda_k^\star \ = \ \log(u^k) - \log(e^{-\eta^{-1}C_k}\text{diag}(e^{\tau_k^\star})\mathbf{1}_n), \tag{5}$$

$$\tau_k^\star \ = \ \sum_{j=1}^m \omega_j \log(e^{-\eta^{-1}C_j}\text{diag}(e^{\lambda_j^\star})\mathbf{1}_n) - \log(e^{-\eta^{-1}C_k}\text{diag}(e^{\lambda_k^\star})\mathbf{1}_n). \tag{6}$$

Indeed, $(\lambda^\star, \tau^\star)$ must satisfy the following KKT condition,

$$\begin{pmatrix} B_k(\lambda_k^\star, \tau_k^\star)\mathbf{1}_n \\ B_k^\top(\lambda_k^\star, \tau_k^\star)\mathbf{1}_n \end{pmatrix} = \begin{pmatrix} u^k \\ \mathbf{z}^\star \end{pmatrix} \text{ for all } k \in [m], \qquad \sum_{k=1}^m \omega_k \tau_k^\star = \mathbf{0}_n. \tag{7}$$

Using the definition of $B_k(\cdot, \cdot)$, we obtain Eq. (5) and $\tau_k^\star = \log(\mathbf{z}^\star) - \log(e^{-\eta^{-1}C}\text{diag}(e^{\lambda_k^\star})\mathbf{1}_n)$. This together with the second equation in Eq. (7) and $\sum_{k=1}^m \omega_k = 1$ yields Eq. (6).

Next we prove the bound with R_τ. Indeed, given that $j \in [m]$, we first show that there exists an optimal solution (λ^j, τ^j) such that

$$\max_{1 \leq i \leq n} (\tau_k^j)_i \geq 0 \geq \min_{1 \leq i \leq n} (\tau_k^j)_i, \quad \text{for all } k \neq j. \tag{8}$$

Letting $(\widehat{\lambda}, \widehat{\tau})$ be an optimal solution of the dual entropic regularized FS-WBP in Eq. (4), the claim holds true if $\widehat{\tau}$ satisfies Eq. (8). Otherwise, we define $m-1$ shift terms given by

$$\Delta\widehat{\tau}_k \ = \ \frac{\max_{1 \leq i \leq n}(\widehat{\tau}_k)_i + \min_{1 \leq i \leq n}(\widehat{\tau}_k)_i}{2} \ \in \ \mathbb{R} \quad \text{for all } k \neq j,$$

and let (λ^j, τ^j) with

$$\begin{array}{lll} \tau_k^j = \widehat{\tau}_k - \Delta\widehat{\tau}_k\mathbf{1}_n, & \lambda_k^j = \widehat{\lambda}_k + \Delta\widehat{\tau}_k\mathbf{1}_n, & \text{for all } k \neq j, \\ \tau_j^j = \widehat{\tau}_j + (\sum_{k \neq j}(\frac{\omega_k}{\omega_j})\Delta\widehat{\tau}_k)\mathbf{1}_n, & \lambda_j^j = \widehat{\lambda}_j - (\sum_{k \neq j}(\frac{\omega_k}{\omega_j})\Delta\widehat{\tau}_k)\mathbf{1}_n. & \end{array}$$

Using this construction, we have $(\lambda_k^j)_i + (\tau_k^j)_{i'} = (\widehat{\lambda}_k)_i + (\widehat{\tau}_k)_{i'}$ for all $i, i' \in [n]$ and all $k \in [m]$. This implies that $B_k(\widehat{\lambda}_k, \widehat{\tau}_k) = B_k(\lambda_k^{k'}, \tau_k^{k'})$ for all $k \in [m]$. Furthermore, we have

$$\sum_{k=1}^m \omega_k \tau_k^j = \sum_{k=1}^m \omega_k \widehat{\tau}_k, \qquad \sum_{k=1}^m \omega_k (\lambda_k^j)^\top u^k = \sum_{k=1}^m \omega_k \widehat{\lambda}_k^\top u^k.$$

Putting these pieces together yields $\varphi(\lambda^j, \tau^j) = \varphi(\widehat{\lambda}, \widehat{\tau})$. In addition, by the definition of (λ^j, τ^j) and $m-1$ shift terms, τ^j satisfies Eq. (8). Therefore, we conclude that (λ^j, τ^j) is an optimal solution that satisfies Eq. (8). Then Kroshnin et al. [2019, Lemma 4] implies that

$$-\eta^{-1}\|C_k\|_\infty + \log(\mathbf{1}_n^\top e^{\lambda_k^j}) \le [\log(e^{-\eta^{-1}C_k}\mathrm{diag}(e^{\lambda_k^j})\mathbf{1}_n)]_{i'} \le \log(\mathbf{1}_n^\top e^{\lambda_k^j}), \quad \text{for all } i' \in [n],$$

Therefore, we have

$$\max_{1\le i\le n} (\tau_k^t)_i - \min_{1\le i\le n} (\tau_k^t)_i \le \eta^{-1}\|C_k\|_\infty + \sum_{i=1}^m \omega_i \eta^{-1}\|C_i\|_\infty, \quad \text{for all } k \in [m]. \tag{9}$$

Combining Eq. (8) and Eq. (9) yields that

$$\|\tau_k^j\|_\infty \ \le \ \eta^{-1}\|C_k\|_\infty + \sum_{i=1}^m \omega_i \eta^{-1}\|C_i\|_\infty \quad \text{for all } k \neq j. \tag{10}$$

Since $\sum_{i=1}^m \omega_i \tau_i^\star = \mathbf{0}_n$, we have

$$\|\tau_j^j\|_\infty \le \omega_j^{-1}\sum_{i\neq j} \omega_i \|\tau_i^\star\|_\infty \le (\eta\omega_j)^{-1}\sum_{i\neq j}\omega_i\|C_i\|_\infty + (\eta\omega_j)^{-1}(1-\omega_j)\sum_{i=1}^m \omega_i\|C_i\|_\infty.$$

Then we proceed to the key part and define the averaging iterate

$$\lambda^\star = \sum_{j=1}^m \omega_j \lambda^j, \qquad \tau^\star = \sum_{j=1}^m \omega_j \tau^j.$$

Since φ is convex and $(\omega_1, \omega_2, \ldots, \omega_m) \in \Delta^m$, we have $\varphi(\lambda^\star, \tau^\star) \le \sum_{j=1}^m \omega_j \varphi(\lambda^j, \tau^j)$ and $\sum_{k=1}^m \omega_k \tau_k^\star = \mathbf{0}_n$. Since (λ^j, τ^j) are optimal solutions for all $j \in [m]$, we conclude that $(\lambda^\star, \tau^\star)$ is an optimal solution and

$$\begin{aligned}
\|\tau_k^\star\|_\infty \ &\le \ \sum_{j=1}^m \omega_j \|\tau_k^j\|_\infty \ = \ \omega_k\|\tau_k^k\|_\infty + \sum_{j\neq k}\omega_j\|\tau_k^j\|_\infty \\
&\le \ \eta^{-1}\sum_{i\neq k}\omega_i\|C_i\|_\infty + \eta^{-1}(1-\omega_k)\sum_{i=1}^m \omega_i\|C_i\|_\infty + \eta^{-1}(1-\omega_k)(\|C_k\|_\infty + \sum_{i=1}^m \omega_i\|C_i\|_\infty) \\
&\le \ \eta^{-1}\|C_k\|_\infty + 3\eta^{-1}\sum_{i=1}^m \omega_i\|C_i\|_\infty \ \le \ 4\eta^{-1}\bar{C}.
\end{aligned}$$

Finally, we prove the bound with R_λ. Indeed, Eq. (5) implies that

$$\begin{aligned}
\max_{1\le i\le n} (\lambda_k^\star)_i \ &\le \ \eta^{-1}\|C_k\|_\infty + \log(n) + \|\tau_k^\star\|_\infty, \\
\min_{1\le i\le n} (\lambda_k^\star)_i \ &\ge \ \log(\bar{u}) - \log(n) - \|\tau_k^\star\|_\infty.
\end{aligned}$$

Therefore, $\|\lambda_k^\star\|_\infty \le \eta^{-1}\|C_k\|_\infty + \log(n) - \log(\bar{u}) + \|\tau_k^\star\|_\infty$ which implies the desired result. □

Remark 2.2. *Lemma 2.1 is analogues to [Lin et al., 2019b, Lemma 3.2] for the dual entropic regularized OT problem. Here we hope to remark that the dual entropic regularized FS-WBP is more complicated and need a novel constructive iterate* $(\lambda^\star, \tau^\star) = \sum_{j=1}^m \omega_j(\lambda^j, \tau^j)$ *which is not required before. Moreover, Lemma 2.1 is important to quantify the progress of the* FastIBP *algorithm where the techniques in Kroshnin et al. [2019] are not applicable.*

The upper bound for the ℓ_∞-norm of the optimal solution of dual entropic regularized FS-WBP in Lemma 2.1 leads to the following straightforward consequence.

Corollary 2.3. *For the dual entropic regularized FS-WBP, there exists an optimal solution* $(\lambda^\star, \tau^\star)$ *such that for all* $k \in [m]$,

$$\|\lambda_k^\star\| \leq \sqrt{n}R_\lambda, \quad \|\tau_k^\star\| \leq \sqrt{n}R_\tau, \quad \text{for all } k \in [m],$$

where $R_\lambda, R_\tau > 0$ *are defined in Lemma 2.1.*

Finally, we observe that φ can be decomposed into the weighted sum of m functions and prove that each component function φ_k has Lipschitz continuous gradient with the constant 4 in the following lemma.

Lemma 2.4. *The following statement holds true,* $\varphi(\lambda, \tau) = \sum_{k=1}^m \varphi_k(\lambda_k, \tau_k)$ *where* $\varphi_k(\lambda_k, \tau_k) = \mathbf{1}_n^\top B_k(\lambda_k, \tau_k)\mathbf{1}_n - \lambda_k^\top u^k$ *for all* $k \in [m]$*. Moreover, each* φ_k *has Lipschitz continuous gradient in* ℓ_2*-norm and the Lipschitz constant is upper bounded by* 4*. Formally,*

$$\|\nabla\varphi_k(\lambda, \tau) - \nabla\varphi_k(\lambda', \tau')\| \leq 4\left\|\begin{pmatrix}\lambda \\ \tau\end{pmatrix} - \begin{pmatrix}\lambda' \\ \tau'\end{pmatrix}\right\| \quad \text{for all } k \in [m].$$

which is equivalent to

$$\varphi(\lambda', \tau') - \varphi(\lambda, \tau) \leq \begin{pmatrix}\lambda' - \lambda \\ \tau' - \tau\end{pmatrix}^\top \nabla\varphi(\lambda, \tau) + 2\left(\sum_{k=1}^m \omega_k \left\|\begin{pmatrix}\lambda_k' - \lambda_k \\ \tau_k' - \tau_k\end{pmatrix}\right\|^2\right).$$

Proof. The first statement directly follows from the definition of φ in Eq. (4). For the second statement, we provide the explicit form of the gradient of φ_k as follows,

$$\nabla\varphi_k(\lambda, \tau) = \begin{pmatrix}B_k(\lambda, \tau)\mathbf{1}_n - u^k \\ B_k^\top(\lambda, \tau)\mathbf{1}_n\end{pmatrix}.$$

Equivalently, we have

$$\|\nabla\varphi_k(\lambda', \tau') - \nabla\varphi_k(\lambda, \tau)\| = \left\|\begin{pmatrix}B_k(\lambda', \tau')\mathbf{1}_n - B_k(\lambda, \tau)\mathbf{1}_n \\ B_k(\lambda', \tau')^\top\mathbf{1}_n - B_k(\lambda, \tau)^\top\mathbf{1}_n\end{pmatrix}\right\|.$$

Now we construct the following entropic regularized OT problem,

$$\min_{X \in \mathbb{R}^{n\times n}} \langle C_k, X\rangle - \eta H(X), \quad \text{s.t.} X\mathbf{1}_n = u, \ X^\top\mathbf{1}_n = v.$$

where u and v are two probability vectors in Δ^n. This belongs to the minimization problem $\min_{x\in\mathbb{R}^{n^2}}\{f(x) : Ax = b\}$. Following the derivation in Dvurechensky et al. [2018], Lin et al. [2019b], the dual function has the following form,

$$\tilde{f}(\alpha, \beta) \ = \ \eta\left(\sum_{i,j=1}^n e^{-\frac{(C_k)_{ij} - \alpha_i - \beta_j}{\eta} - 1}\right) - \langle\alpha, u\rangle - \langle\beta, v\rangle + \eta.$$

Since f is strongly convex on $\{x : Ax = b\}$ in ℓ_1-norm and $\|A\|_{1\to 2} = 2$, the gradient of $\tilde{f}$ is Lipschitz continuous with the constant $4\eta^{-1}$ [Nesterov, 2018]. This implies that

$$\left\|\begin{pmatrix}\text{diag}(e^{\alpha'})e^{-\eta^{-1}C-\mathbf{1}_n\mathbf{1}_n^\top}\text{diag}(e^{\beta'})\mathbf{1}_n - \text{diag}(e^{\alpha})e^{-\eta^{-1}C-\mathbf{1}_n\mathbf{1}_n^\top}\text{diag}(e^{\beta})\mathbf{1}_n \\ \text{diag}(e^{\beta'})e^{-\eta^{-1}C-\mathbf{1}_n\mathbf{1}_n^\top}\text{diag}(e^{\alpha'})\mathbf{1}_n - \text{diag}(e^{\beta})e^{-\eta^{-1}C-\mathbf{1}_n\mathbf{1}_n^\top}\text{diag}(e^{\alpha})\mathbf{1}_n\end{pmatrix}\right\| \leq 4\eta^{-1}\left\|\begin{pmatrix}\alpha' - \alpha \\ \beta' - \beta\end{pmatrix}\right\|.$$

Performing the change of variable $\lambda = \eta^{-1}\alpha - (1/2)\mathbf{1}_n$ and $\tau = \eta^{-1}\beta - (1/2)\mathbf{1}_n$ together with the definition of $B_k(\cdot, \cdot)$, we have

$$\left\|\begin{pmatrix}B_k(\lambda', \tau')\mathbf{1}_n - B_k(\lambda, \tau)\mathbf{1}_n \\ B_k(\lambda', \tau')^\top\mathbf{1}_n - B_k(\lambda, \tau)^\top\mathbf{1}_n\end{pmatrix}\right\| \;\leq\; 4\left\|\begin{pmatrix}\lambda' - \lambda \\ \tau' - \tau\end{pmatrix}\right\|.$$

This completes the proof. □

3 Computational Hardness

In this section, we analyze the computational hardness of the fixed-support Wasserstein barycenter problem (FS-WBP) in Eq. (2). After introducing some well-known characterization theorems in combinatorial optimization, we show that the FS-WBP in Eq. (2) is a minimum-cost flow (MCF) problem when $m = 2$ and $n \geq 3$ but is not when $m \geq 3$ and $n \geq 3$.

3.1 Combinatorial techniques

We present some classical results in combinatorial optimization and graph theory. The first one is celebrated Ghouila-Houri's characterization theorem [Ghouila-Houri, 1962].

Definition 3.1. *A totally unimodular (TU) matrix is one for which every square submatrix has determinant -1, 0 or 1.*

Proposition 3.1 (Ghouila-Houri)**.** *A $\{-1, 0, 1\}$-valued matrix $A \in \mathbb{R}^{m\times n}$ is TU if and only if for each set $I \subseteq [m]$ there is a partition I_1, I_2 of I such that*

$$\sum_{i\in I_1} a_{ij} - \sum_{i\in I_2} a_{ij} \in \{-1, 0, 1\}, \quad \forall j \in [n].$$

The second result [Berge, 2001, Theorem 1, Chapter 15] shows that the incidence matrices of directed graphs and 2-colorable undirected graphs are TU.

Proposition 3.2. *Let A be a $\{-1, 0, 1\}$-valued matrix. Then A is TU if each column contains at most two nonzero entries and all rows are partitioned into two sets I_1 and I_2 such that: If two nonzero entries of a column have the same sign, they are in different sets. If these two entries have different signs, they are in the same set.*

Finally, we characterize the constraint matrix arising in a MCF problem.

Definition 3.2. *The MCF problem finds the cheapest possible way of sending a certain amount of flow through a flow network. Formally,*

$$\begin{array}{ll}\min & \sum_{(u,v)\in E} f(u,v)\cdot a(u,v) \\ \text{s.t.} & f(u,v) \geq 0, \ \ \text{for all } (u,v) \in E, \\ & f(u,v) \leq c(u,v) \ \ \text{for all } (u,v) \in E, \\ & f(u,v) = -f(v,u) \ \ \text{for all } (u,v) \in E, \\ & \sum_{(u,w)\in E \text{ or } (w,u)\in E} f(u,w) = 0, \\ & \sum_{w\in V} f(s,w) = d \ \text{ and } \ \sum_{w\in V} f(w,t) = d.\end{array}$$

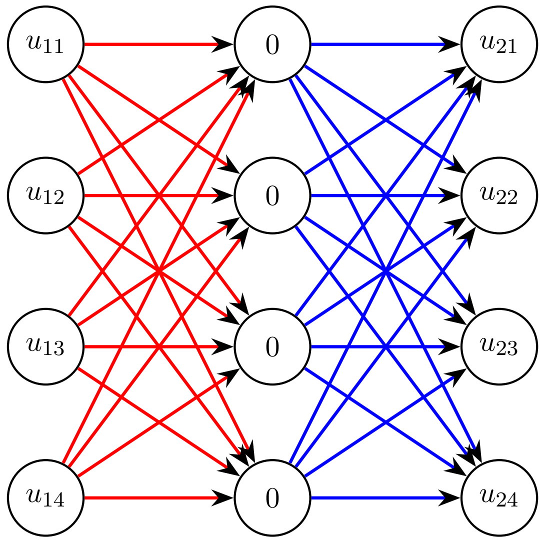


Figure 1: Represent the FS-WBP in Eq. (2) as a MCF problem when $(m, n) = (2, 4)$.

The flow network $G = (V, E)$ is a directed graph $G = (V, E)$ with a source vertex $s \in V$ and a sink vertex $t \in V$, where each edge $(u, v) \in E$ has capacity $c(u, v) > 0$, flow $f(u, v) \geq 0$ and cost $a(u, v)$, with most MCF algorithms supporting edges with negative costs. The cost of sending this flow along an edge (u, v) is $f(u, v) \cdot a(u, v)$. The problem requires an amount of flow d to be sent from source s to sink t. The definition of the problem is to minimize the total cost of the flow over all edges.

Proposition 3.3. *The constraint matrix arising in a MCF problem is TU and its rows are categorized into a single set using Proposition 3.2.*

Proof. The standard LP representation of the MCF problem is

$$\min_{x \in \mathbb{R}^{|E|}} \ c^\top x, \quad \text{s.t. } Ax = b, \ l \leq x \leq u.$$

where $x \in \mathbb{R}^{|E|}$ with x_j being the flow through arc j, $b \in \mathbb{R}^{|V|}$ with b_i being external supply at node i and $\mathbf{1}^\top b = 0$, c_j is unit cost of flow through arc j, l_j and u_j are lower and upper bounds on flow through arc j and $A \in \mathbb{R}^{|V| \times |E|}$ is the arc-node incidence matrix with entries

$$A_{ij} = \begin{cases} -1 & \text{if arc } j \text{ starts at node } i \\ 1 & \text{if arc } j \text{ ends at node } i \\ 0 & \text{otherwise} \end{cases}.$$

Since each arc has two endpoints, the constraint matrix A is a $\{-1, 0, 1\}$-valued matrix in which each column contains two nonzero entries 1 and -1. Using Proposition 3.2, we obtain that A is TU and the rows of A are categorized into a single set. □

3.2 Main result

We present our main results on the computational hardness of the FS-WBP in Eq. (2). First, we show that the FS-WBP in this LP form is a MCF problem when $m = 2$ and $n \geq 2$. This result has been briefly discussed in [Anderes et al., 2016, Page 400] and we provide the details for the sake of completeness.

The FS-WBP is a MCF problem when $m = 2$ and $n \geq 2$; see Figure 1 for the graph when $(m, n) = (2, 4)$. In particular, it is a transportation problem with n warehouse, n transshipment centers and n retail outlets. Each u_{1i} is the amount of supply provided by ith warehouse and each u_{2j} is the amount of demand requested by jth retail outlet. $(X_1)_{ij}$ is the flow sent from ith warehouse to jth transshipment center and $(X_2)_{ij}$ is the flow sent from ith transshipment center to jth retail outlet. $(C_1)_{ij}$ and $(C_2)_{ij}$ refer to the unit cost of corresponding flow. To this end, the Wasserstein barycenter $u \in \mathbb{R}^n$ is a flow vector with u_i being the flow through ith transshipment center.

Proceed to the setting $m \geq 3$, it is unclear whether the FS-WBP in Eq. (2) has the graph representation as shown before. Instead, we turn to the algebraic techniques and provide an explicit form as follows,

$$\begin{aligned} \min \quad & \sum_{k=1}^{m} \langle C_k, X_k \rangle \qquad (11) \\ \text{s.t.} \quad & \left(\begin{array}{ccccc} -E & \cdots & \cdots & \cdots & \cdots \\ \vdots & E & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & (-1)^{m-1}E & \vdots \\ \vdots & \ddots & \ddots & \ddots & (-1)^m E \\ \hdashline G & -G & \ddots & \ddots & \vdots \\ \vdots & -G & G & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \cdots & \cdots & \cdots & (-1)^m G & (-1)^{m+1} G \end{array}\right) \begin{pmatrix} \text{vec}\,(X_1) \\ \text{vec}\,(X_2) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \text{vec}\,(X_m) \end{pmatrix} = \begin{pmatrix} -u^1 \\ u^2 \\ \vdots \\ (-1)^{m-1} u^{m-1} \\ (-1)^m u^m \\ \mathbf{0}_n \\ \vdots \\ \mathbf{0}_n \end{pmatrix}, \end{aligned}$$

where $E = I_n \otimes \mathbf{1}_n^\top \in \mathbb{R}^{n \times n^2}$ and $G = \mathbf{1}_n^\top \otimes I_n \in \mathbb{R}^{n \times n^2}$. Each column of the constraint matrix arising in Eq. (11) has either two or three nonzero entries in $\{-1, 0, 1\}$. In the following theorem, we study the structure of the constraint matrix when $m \geq 3$ and $n = 2$.

Theorem 3.4. *The constraint matrix arising in Eq.* (11) *is TU when* $m \geq 3$ *and* $n = 2$*.*

Proof. When $n = 2$, the constraint matrix A has $E = I_2 \otimes \mathbf{1}_2^\top$ and $G = \mathbf{1}_2^\top \otimes I_2$. The matrix $A \in \mathbb{R}^{(4m-2) \times 4m}$ is a $\{-1, 0, 1\}$-valued matrix with several redundant rows and each column has at most three nonzero entries in $\{-1, 0, 1\}$. Now we simplify the matrix A by removing a specific set of redundant rows. In particular, we observe that

$$\sum_{i \in \{1,2,3,4,2m+1,2m+2\}} a_{ij} = 0, \quad \forall j \in [4m],$$

which implies that the $(2m + 2)$th row is redundant. Similarly, we have

$$\sum_{i \in \{3,4,5,6,2m+3,2m+4\}} a_{ij} = 0, \quad \forall j \in [4m],$$

which implies that the $(2m + 3)$th row is redundant. Using this argument, we remove $m - 1$ rows from the last $2m - 2$ rows. The resulting matrix $\bar{A} \in \mathbb{R}^{(3m-1) \times 4m}$ has very nice structure such that each column has only two nonzero entries 1 and -1; see the following matrix when m is odd:

$$\bar{A} \;=\; \left(\begin{array}{ccccc} -E & \cdots & \cdots & \cdots & \cdots \\ \vdots & E & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & (-1)^{m-1}E & \vdots \\ \vdots & \ddots & \ddots & \ddots & (-1)^{m}E \\ \hdashline \mathbf{1}_2^\top \otimes e_1 & -\mathbf{1}_2^\top \otimes e_1 & \ddots & \ddots & \vdots \\ \vdots & -\mathbf{1}_2^\top \otimes e_2 & \mathbf{1}_2^\top \otimes e_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \cdots & \cdots & \cdots & (-1)^m \mathbf{1}_2^\top \otimes e_2 & (-1)^{m+1}\mathbf{1}_2^\top \otimes e_2 \end{array}\right).$$

where e_1 and e_2 are respectively the first and second standard basis (row) vectors in $\mathbb{R}^2$. Furthermore, the rows of $\bar{A}$ are categorized into a single set so that the criterion in Proposition 3.2 holds true (the dashed line in the formulation of $\bar{A}$ serves as a partition of this single set into two sets). Using Proposition 3.2, we conclude that $\bar{A}$ is TU. □

We first provide an illustrative counterexample for showing that the FS-WBP in Eq. (11) is not a MCF problem when $m = 3$ and $n = 3$.

Example 3.1. *When $m = 3$ and $n = 3$, the constraint matrix is*

$$A \;=\; \left(\begin{array}{ccc} -I_3 \otimes \boldsymbol{1}_3^\top & \boldsymbol{0}_{3\times 9} & \boldsymbol{0}_{3\times 9} \\ \boldsymbol{0}_{3\times 9} & I_3 \otimes \boldsymbol{1}_3^\top & \boldsymbol{0}_{3\times 9} \\ \boldsymbol{0}_{3\times 9} & \boldsymbol{0}_{3\times 9} & -I_3 \otimes \boldsymbol{1}_3^\top \\ \hdashline \boldsymbol{1}_3^\top \otimes I_3 & -\boldsymbol{1}_3^\top \otimes I_3 & \boldsymbol{0}_{3\times 9} \\ \boldsymbol{0}_{3\times 9} & -\boldsymbol{1}_3^\top \otimes I_3 & \boldsymbol{1}_3^\top \otimes I_3 \end{array}\right). \tag{12}$$

Setting the set $I = \{1, 4, 7, 10, 11, 13, 15\}$ and letting e_1, e_2 and e_3 be the first, second and third standard basis row vectors in $\mathbb{R}^n$, the resulting matrix with the rows in I is

$$R = \left(\begin{array}{ccc} -e_1 \otimes \boldsymbol{1}_3^\top & \boldsymbol{0}_{1\times 9} & \boldsymbol{0}_{1\times 9} \\ \boldsymbol{0}_{1\times 9} & e_1 \otimes \boldsymbol{1}_3^\top & \boldsymbol{0}_{1\times 9} \\ \boldsymbol{0}_{1\times 9} & \boldsymbol{0}_{1\times 9} & -e_1 \otimes \boldsymbol{1}_3^\top \\ \hdashline \boldsymbol{1}_3^\top \otimes e_1 & -\boldsymbol{1}_3^\top \otimes e_1 & \boldsymbol{0}_{1\times 9} \\ \boldsymbol{1}_3^\top \otimes e_2 & -\boldsymbol{1}_3^\top \otimes e_2 & \boldsymbol{0}_{1\times 9} \\ \boldsymbol{0}_{1\times 9} & -\boldsymbol{1}_3^\top \otimes e_1 & \boldsymbol{1}_3^\top \otimes e_1 \\ \boldsymbol{0}_{1\times 9} & -\boldsymbol{1}_3^\top \otimes e_3 & \boldsymbol{1}_3^\top \otimes e_3 \end{array}\right).$$

Instead of considering all columns of R, it suffices to show that no partition of I guarantees for any $j \in \{1, 2, 11, 12, 13, 19, 21\}$ that

$$\sum_{i \in I_1} R_{ij} - \sum_{i \in I_2} R_{ij} \in \{-1, 0, 1\}.$$

We write the submatrix of R with these columns as

$$
\bar{R} = \begin{pmatrix}
-1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1
\end{pmatrix}
$$

and perform the following steps:

1. *We claim that rows 1, 2, 4, 5 and 7 are in the same set I_1. Indeed, columns 1 and 2 imply that rows 1, 4 and 5 are in the same set. Column 3 and 4 imply that rows 2, 5 and 7 are in the same set. Putting these pieces together yields the desired claim.*

2. *We consider the set that the row 6 belongs to and claim a contradiction. Indeed, row 6 can not be in I_1 since column 5 implies that rows 4 and 6 are not in the same set. However, row 6 must be in I_1 since columns 6 and 7 imply that rows 3, 6 and 7 are in the same set. Putting these pieces together yields the desired contradiction.*

Using Propositions 3.1 and 3.3, we conclude that A is not TU and problem (11) is not a MCF problem when $m = 3$ and $n = 3$.

Finally, we prove that the FS-WBP in Eq. (11) is not a MCF when $m \geq 3$ and $n \geq 3$. The basic idea is to extend the construction in Example 3.1 to the general case.

Theorem 3.5. *The FS-WBP in Eq. (11) is not a MCF problem when $m \geq 3$ and $n \geq 3$.*

Proof. We use the proof by contradiction. In particular, assume that problem (11) is a MCF problem when $m \geq 3$ and $n \geq 3$, Proposition 3.3 implies that the constraint matrix A is TU. Since A is a $\{-1, 0, 1\}$-valued matrix, Proposition 3.1 further implies that for each set $I \subseteq [2mn - n]$ there is a partition I_1, I_2 of I such that

$$
\sum_{i \in I_1} a_{ij} - \sum_{i \in I_2} a_{ij} \in \{-1, 0, 1\}, \quad \forall j \in [mn^2]. \tag{13}
$$

In what follows, for any given $m \geq 3$ and $n \geq 3$, we construct a set of rows I such that no partition of I guarantees that Eq. (13) holds true. For the ease of presentation, we rewrite the matrix $A \in \mathbb{R}^{(2mn-n) \times mn^2}$ as follows,

$$
A = \left(\begin{array}{ccccc}
-I_n \otimes \mathbf{1}_n^\top & \cdots & \cdots & \cdots & \cdots \\
\vdots & I_n \otimes \mathbf{1}_n^\top & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & (-1)^{m-1} I_n \otimes \mathbf{1}_n^\top & \vdots \\
\vdots & \ddots & \ddots & \ddots & (-1)^m I_n \otimes \mathbf{1}_n^\top \\
\hdashline
\mathbf{1}_n^\top \otimes I_n & -\mathbf{1}_n^\top \otimes I_n & \ddots & \ddots & \vdots \\
\vdots & -\mathbf{1}_n^\top \otimes I_n & \mathbf{1}_n^\top \otimes I_n & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\cdots & \cdots & \cdots & (-1)^m \mathbf{1}_n^\top \otimes I_n & (-1)^{m+1} \mathbf{1}_n^\top \otimes I_n
\end{array}\right).
$$

Setting the set $I = \{1, n+1, 2n+1, 3n+1, 3n+2, 4n+1, 4n+3\}$ and letting e_1, e_2 and e_3 be the first, second and third standard basis row vectors in $\mathbb{R}^n$, the resulting matrix with the rows in I is

$$R = \begin{pmatrix} -e_1 \otimes \mathbf{1}_n^\top & \mathbf{0}_{1\times n^2} & \mathbf{0}_{1\times n^2} & \mathbf{0}_{1\times n^2} & \cdots & \mathbf{0}_{1\times n^2} \\ \mathbf{0}_{1\times n^2} & e_1 \otimes \mathbf{1}_n^\top & \mathbf{0}_{1\times n^2} & \mathbf{0}_{1\times n^2} & \cdots & \mathbf{0}_{1\times n^2} \\ \mathbf{0}_{1\times n^2} & \mathbf{0}_{1\times n^2} & -e_1 \otimes \mathbf{1}_n^\top & \mathbf{0}_{1\times n^2} & \cdots & \mathbf{0}_{1\times n^2} \\ \hdashline \mathbf{1}_n^\top \otimes e_1 & -\mathbf{1}_n^\top \otimes e_1 & \mathbf{0}_{1\times n^2} & \mathbf{0}_{1\times n^2} & \cdots & \mathbf{0}_{1\times n^2} \\ \mathbf{1}_n^\top \otimes e_2 & -\mathbf{1}_n^\top \otimes e_2 & \mathbf{0}_{1\times n^2} & \mathbf{0}_{1\times n^2} & \cdots & \mathbf{0}_{1\times n^2} \\ \mathbf{0}_{1\times n^2} & -\mathbf{1}_n^\top \otimes e_1 & \mathbf{1}_n^\top \otimes e_1 & \mathbf{0}_{1\times n^2} & \cdots & \mathbf{0}_{1\times n^2} \\ \mathbf{0}_{1\times n^2} & -\mathbf{1}_n^\top \otimes e_3 & \mathbf{1}_n^\top \otimes e_3 & \mathbf{0}_{1\times n^2} & \cdots & \mathbf{0}_{1\times n^2} \end{pmatrix}.$$

Instead of considering all columns of R, it suffices to show that no partition of I guarantees

$$\sum_{i\in I_1} R_{ij} - \sum_{i\in I_2} R_{ij} \in \{-1, 0, 1\},$$

for all $j \in \{1, 2, n^2+2, n^2+3, n^2+n+1, 2n^2+1, 2n^2+3\}$. We write the submatrix of R with these columns as

$$\bar{R} = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix}.$$

Applying the same argument used in Example 3.1, we obtain from Propositions 3.1 and 3.3 that A is not TU when $m \geq 3$ and $n \geq 3$, which is a contradiction. As a consequence, the conclusion of the theorem follows. □

Remark 3.6. *Theorem 3.5 resolves an open question and partially explains why the direct application of network flow algorithms to the FS-WBP in Eq. (11) is inefficient in practice. However, this does not eliminate the possibility that the FS-WBP is equivalent to some other LP with good complexity. For example, Ge et al. [2019] have recently explored the block structure of the FS-WBP in Eq. (11) and identified an equivalent LP formulation which is suitable for the interior-point algorithm.*

Remark 3.7. *Theorem 3.5 supports the problem reformulation of the FS-WBP which forms the basis for various algorithms, e.g., Benamou et al. [2015], Cuturi and Peyré [2016, 2018], Claici et al. [2018], Kroshnin et al. [2019], Ge et al. [2019], Guminov et al. [2019] for example.*

4 Fast Iterative Bregman Projection

In this section, we present a fast *deterministic* variant of the iterative Bregman projection (IBP) algorithm, named FastIBP algorithm, and prove that it achieves the complexity bound of $\widetilde{O}(mn^{7/3}\varepsilon^{-4/3})$. This improves over $\widetilde{O}(mn^2\varepsilon^{-2})$ from iterative Bregman projection algorithm [Benamou et al., 2015] in terms of ε and $\widetilde{O}(mn^{5/2}\varepsilon^{-1})$ from the APDAGD and accelerated Sinkhorn algorithms [Kroshnin et al., 2019, Guminov et al., 2019] in terms of n.

Algorithm 1: FASTIBP($\{C_k, u^k\}_{k\in[m]}, \varepsilon$)

Initialization: $t = 0$, $\theta_0 = 1$ and $\check{\lambda}^0 = \tilde{\lambda}^0 = \check{\tau}^0 = \tilde{\tau}^0 = \mathbf{0}_{mn}$.

while $E_t > \varepsilon$ **do**

Step 1: Compute $\begin{pmatrix} \bar{\lambda}^t \\ \bar{\tau}^t \end{pmatrix} = (1-\theta_t)\begin{pmatrix} \check{\lambda}^t \\ \check{\tau}^t \end{pmatrix} + \theta_t \begin{pmatrix} \tilde{\lambda}^t \\ \tilde{\tau}^t \end{pmatrix}$.

Step 2: Compute $r_k = r(B_k(\bar{\lambda}_k^t, \bar{\tau}_k^t))$ and $c_k = c(B_k(\bar{\lambda}_k^t, \bar{\tau}_k^t))$ for all $k \in [m]$ and perform

$$\begin{aligned} \tilde{\lambda}_k^{t+1} &= \tilde{\lambda}_k^t - (r_k - u^k)/(4\theta_t), \quad \text{for all } k \in [m], \\ \tilde{\tau}^{t+1} &= \operatorname*{argmin}_{\sum_{k=1}^m \omega_k \tau_k = \mathbf{0}_n} \sum_{k=1}^m \omega_k [(\tau_k - \bar{\tau}_k^t)^\top c_k + 2\theta_t \|\tau_k - \tilde{\tau}_k^t\|^2]. \end{aligned}$$

Step 3: Compute $\begin{pmatrix} \widehat{\lambda}^t \\ \widehat{\tau}^t \end{pmatrix} = \begin{pmatrix} \bar{\lambda}^t \\ \bar{\tau}^t \end{pmatrix} + \theta_t \begin{pmatrix} \tilde{\lambda}^{t+1} \\ \tilde{\tau}^{t+1} \end{pmatrix} - \theta_t \begin{pmatrix} \tilde{\lambda}^t \\ \tilde{\tau}^t \end{pmatrix}$.

Step 4: Compute $\begin{pmatrix} \acute{\lambda}^t \\ \acute{\tau}^t \end{pmatrix} = \operatorname{argmin}\left\{\varphi(\lambda, \tau) \mid \begin{pmatrix} \lambda \\ \tau \end{pmatrix} \in \left\{\begin{pmatrix} \check{\lambda}^t \\ \check{\tau}^t \end{pmatrix}, \begin{pmatrix} \widehat{\lambda}^t \\ \widehat{\tau}^t \end{pmatrix}\right\}\right\}$.

Step 5a: Compute $c_k = c(B_k(\acute{\lambda}_k^t, \acute{\tau}_k^t))$ for all $k \in [m]$.

Step 5b: Compute $\grave{\tau}_k^t = \acute{\tau}_k^t + \sum_{k=1}^m \omega_k \log(c_k) - \log(c_k)$ for all $k \in [m]$ and $\grave{\lambda}^{t+1} = \acute{\lambda}^t$.

Step 6a: Compute $r_k = r(B_k(\grave{\lambda}_k^t, \grave{\tau}_k^t))$ for all $k \in [m]$.

Step 6b: Compute $\lambda_k^t = \grave{\lambda}_k^t + \log(u^k) - \log(r_k)$ for all $k \in [m]$ and $\tau^t = \grave{\tau}^t$.

Step 7a: Compute $c_k = c(B_k(\lambda_k^t, \tau_k^t))$ for all $k \in [m]$.

Step 7b: Compute $\check{\tau}_k^{t+1} = \tau_k^t + \sum_{k=1}^m \omega_k \log(c_k) - \log(c_k)$ for all $k \in [m]$ and $\check{\lambda}^{t+1} = \lambda^t$.

Step 8: Compute $\theta_{t+1} = \theta_t(\sqrt{\theta_t^2 + 4} - \theta_t)/2$.

Step 9: Increment by $t = t + 1$.

end while

Output: $(B_1(\lambda_1^t, \tau_1^t), B_2(\lambda_2^t, \tau_2^t), \ldots, B_m(\lambda_m^t, \tau_m^t))$.

4.1 Algorithmic scheme

To facilitate the later discussion, we present the FASTIBP algorithm in pseudocode form in Algorithm 1 and its application to entropic regularized FS-WBP in Algorithm 2. Note that $(B_1(\lambda_1^t, \tau_1^t), \ldots, B_m(\lambda_m^t, \tau_m^t))$ stand for the primal variables while (λ^t, τ^t) are the dual variables for the entropic regularized FS-WBP.

The FASTIBP algorithm is a *deterministic* variant of the iterative Bregman projection (IBP) algorithm [Benamou et al., 2015]. While the IBP algorithm can be interpreted as a dual coordinate descent, the acceleration achieved by the FASTIBP algorithm mostly depends on the refined characterization of per-iteration progress using the scheme with momentum; see **Step 1-3** and **Step 8**. To the best of our knowledge, this scheme has been well studied by [Nesterov, 2012, 2013, Fercoq and Richtárik, 2015, Nesterov and Stich, 2017] yet *first* introduced to accelerate the optimal transport algorithms.

Furthermore, **Step 4** guarantees that $\{\varphi(\check{\lambda}^t, \check{\tau}^t)\}_{t\geq 0}$ is monotonically decreasing and **Step 7** ensures the sufficient large progress from (λ_k^t, τ_k^t) to $(\check{\lambda}^{t+1}, \check{\tau}^{t+1})$. Moreover, **Step 5a-5b** are performed such that $\tau_k^t = \grave{\tau}_k^t$ satisfies the bounded difference property: $\max_{1\leq i\leq n}(\tau_k^t)_i - \min_{1\leq i\leq n}(\tau_k^t)_i \leq R_\tau/2$ while **Step 6a-6b** guarantees that the marginal conditions hold true: $r(B_k(\lambda_k^t, \tau_k^t)) = u^k$ for all $k \in [m]$. We remark that **Step 4-7** are *specialized* to the FS-WBP in Eq. (2) and have *not* been appeared in the coordinate descent literature before.

Algorithm 2: Finding Wasserstein barycenter by the FastIBP algorithm

Input: $\eta = \varepsilon/(4\log(n))$ and $\bar{\varepsilon} = \varepsilon/(4\max_{1\leq k\leq m}\|C_k\|_\infty)$.
Step 1: Compute $(\tilde{u}^1, \ldots, \tilde{u}^m) = (1 - \bar{\varepsilon}/4)(u^1, \ldots, u^m) + (\bar{\varepsilon}/4n)(\mathbf{1}_n, \ldots, \mathbf{1}_n)$.
Step 2: Compute $(\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_m) = \text{FastIBP}(\{C_k, \tilde{u}^k\}_{k\in[m]}, \bar{\varepsilon}/2)$.
Step 3: Round $(\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_m)$ to $(\widehat{X}_1, \widehat{X}_2, \ldots, \widehat{X}_m)$ using Kroshnin et al. [2019, Algorithm 4] such that $(\widehat{X}_1, \widehat{X}_2, \ldots, \widehat{X}_m)$ is feasible to the FS-WBP in Eq. (2).
Step 4: Compute $\widehat{u} = \sum_{k=1}^m \omega_k \widehat{X}_k^\top \mathbf{1}_n$
Output: $\widehat{u}$.

The optimality conditions of primal entropic regularized WBP in Eq. (3) and dual entropic regularized WBP in Eq. (4) are

$$\begin{array}{rcll} \mathbf{0}_n & = & r(B_k(\lambda_k, \tau_k)) - u^k, & \text{for all } k \in [m], \\ \mathbf{0}_n & = & c(B_k(\lambda_k, \tau_k)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i, \tau_i)), & \text{for all } k \in [m], \\ \mathbf{0}_n & = & \sum_{k=1}^m \omega_k \tau_k. & \end{array}$$

Since the FastIBP algorithm guarantees that $\sum_{k=1}^m \omega_k \tau_k^t = \mathbf{0}_n$ and $r(B_k(\lambda_k^t, \tau_k^t)) = u^k$ for all $k \in [m]$, we can solve simultaneously primal and dual entropic regularized FS-WBP with an adaptive stopping criterion which does not require to calculate any objective value. The criterion depends on the following quantity to measure the residue at each iteration:

$$E_t \ := \ \sum_{k=1}^m \omega_k \|c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\|_1. \tag{14}$$

Remark 4.1. *We provide some comments on the* FastIBP *algorithm. First, each iteration updates $O(mn^2)$ entries which is similar to the IBP algorithm. Updating $\tilde{\lambda}$ and $\check{\lambda}$ can be efficiently implemented in distributed manner and each of m machine updates $O(n^2)$ entries at each iteration. Second, the computation of $4m$ marginals can be performed using implementation tricks. Indeed, this can be done effectively by using $r(e^{-\eta^{-1}C_k})$ and $c(e^{-\eta^{-1}C_k})$ for all $k \in [m]$, which are computed and stored at the beginning of the algorithm.*

4.2 Convergence analysis

We present several technical lemmas which are important to analyzing the FastIBP algorithm. The first lemma provides the inductive formula and the upper bound for θ_t.

Lemma 4.2. *Let $\{\theta_t\}_{t\geq 0}$ be the iterates generated by the* FastIBP *algorithm. Then we have $0 < \theta_t \leq 2/(t+2)$ and $\theta_t^{-2} = (1 - \theta_{t+1})\theta_{t+1}^{-2}$ for all $t \geq 0$.*

Proof. By the definition of θ_t, we have

$$\left(\frac{\theta_{t+1}}{\theta_t}\right)^2 = \frac{1}{4}\left(\sqrt{\theta_t^2 + 4} - \theta_t\right)^2 = 1 + \frac{\theta_t}{2}\left(\theta_t - \sqrt{\theta_t^2 + 4}\right) = 1 - \theta_{t+1},$$

which implies the desired inductive formula and $\theta_t > 0$ for all $t \geq 0$. Then we proceed to prove that $0 < \theta_t \leq 2/(t+2)$ for all $t \geq 0$ using the induction. Indeed, the claim trivially holds when $t = 0$ as $\theta_0 = 1$. Assume that the hypothesis holds for $t \leq t_0$, i.e., $\theta_{t_0} \leq 2/(t_0 + 2)$, we have

$$\theta_{t_0+1} \ = \ \frac{2}{1 + \sqrt{1 + \frac{4}{\theta_{t_0}^2}}} \ \leq \ \frac{2}{t_0 + 3}.$$

This completes the proof of the lemma. $\square$

The second lemma shows that all the iterates generated by the FastIBP algorithm are feasible to the dual entropic regularized FS-WBP for all $t \geq 1$.

Lemma 4.3. *Let $\{(\check{\lambda}^t, \check{\tau}^t)\}_{t\geq 0}$, $\{(\tilde{\lambda}^t, \tilde{\tau}^t)\}_{t\geq 0}$, $\{(\bar{\lambda}^t, \bar{\tau}^t)\}_{t\geq 0}$, $\{(\widehat{\lambda}^t, \widehat{\tau}^t)\}_{t\geq 0}$, $\{(\acute{\lambda}^t, \acute{\tau}^t)\}_{t\geq 0}$, and $\{(\lambda^t, \tau^t)\}_{t\geq 0}$ be the iterates generated by the* FastIBP *algorithm. Then, we have*

$$\sum_{k=1}^m \omega_k \check{\tau}_k^t = \sum_{k=1}^m \omega_k \tilde{\tau}_k^t = \sum_{k=1}^m \omega_k \bar{\tau}_k^t = \sum_{k=1}^m \omega_k \widehat{\tau}_k^t = \sum_{k=1}^m \omega_k \acute{\tau}_k^t = \sum_{k=1}^m \omega_k \grave{\tau}_k^t = \sum_{k=1}^m \omega_k \tau_k^t = \mathbf{0}_n \quad \text{for all } t \geq 0.$$

Proof. We first verify Lemma 4.3 when $t = 0$. Indeed,

$$\sum_{k=1}^m \omega_k \check{\tau}_k^0 \ = \ \sum_{k=1}^m \omega_k \tilde{\tau}_k^0 \ = \ \mathbf{0}_n.$$

By the definition, $\bar{\tau}^0$ is a convex combination of $\check{\tau}^0$ and $\tilde{\tau}^0$ and $\widehat{\tau}^0$ is a linear combination of $\bar{\tau}^0$, $\tilde{\tau}^1$ and $\tilde{\tau}^0$. Thus, we have

$$\sum_{k=1}^m \omega_k \bar{\tau}_k^0 \ = \ \sum_{k=1}^m \omega_k \widehat{\tau}_k^0 \ = \ \mathbf{0}_n.$$

This also implies that $\sum_{k=1}^m \omega_k \acute{\tau}_k^0 = \mathbf{0}_n$. Using the update formula for $\grave{\tau}^0$, τ^0 and $\check{\tau}^1$, we have

$$\sum_{k=1}^m \omega_k \grave{\tau}_k^0 \ = \ \sum_{k=1}^m \omega_k \tau_k^0 \ = \ \sum_{k=1}^m \omega_k \check{\tau}_k^1 \ = \ \mathbf{0}_n.$$

Besides that, the update formula for $\tilde{\tau}^1$ implies $\sum_{k=1}^m \omega_k \tilde{\tau}_k^1 = \mathbf{0}_n$. Repeating this argument, we obtain the desired equality in the conclusion of Lemma 4.3 for all $t \geq 0$. $\square$

The third lemma shows that the iterates $\{\tau^t\}_{t\geq 0}$ generated by the FastIBP algorithm satisfies the bounded difference property: $\max_{1\leq i\leq n}(\tau_k^t)_i - \min_{1\leq i\leq n}(\tau_k^t)_i \leq R_\tau/2$.

Lemma 4.4. *Let $\{(\lambda^t, \tau^t)\}_{t\geq 0}$ be the iterates generated by the* FastIBP *algorithm. Then the following statement holds true:*

$$\max_{1\leq i\leq n} (\tau_k^t)_i - \min_{1\leq i\leq n} (\tau_k^t)_i \ \leq \ R_\tau/2,$$

where $R_\tau > 0$ is defined in Lemma 2.1.

Proof. We observe that $\tau_k^t = \grave{\tau}_k^t$ for all $k \in [m]$. By the update formula for $\grave{\tau}_k^t$, we have

$$\grave{\tau}_k^t \ = \ \acute{\tau}_k^t + \sum_{i=1}^m \omega_i \log(c_i) - \log(c_k) \ = \ \sum_{i=1}^m \omega_i \log(e^{-\eta^{-1}C_i}\text{diag}(e^{\acute{\lambda}_i^t})\mathbf{1}_n) - \log(e^{-\eta^{-1}C_k}\text{diag}(e^{\acute{\lambda}_k^t})\mathbf{1}_n).$$

After the simple calculation, we have

$$-\eta^{-1}\|C_k\|_\infty + \mathbf{1}_n^\top e^{\acute{\lambda}_k^t} \ \leq \ \log(e^{-\eta^{-1}C_k}\text{diag}(e^{\acute{\lambda}_k^t})\mathbf{1}_n)]_j \ \leq \ \mathbf{1}_n^\top e^{\acute{\lambda}_k^t}.$$

Therefore, the following inequality holds true for all $k \in [m]$,

$$\max_{1\le i\le n} (\tau_k^t)_i - \min_{1\le i\le n} (\tau_k^t)_i \ \le \ \eta^{-1}\|C_k\|_\infty + \eta^{-1}\left(\sum_{i=1}^m \omega_i \|C_i\|_\infty\right) \ = \ 2\eta^{-1}(\max_{1\le k\le m} \|C_k\|_\infty).$$

This together with the definition of R_τ yields the desired inequality. □

The final lemma presents a key descent inequality for the FastIBP algorithm.

Lemma 4.5. *Let $\{(\check{\lambda}^t, \check{\tau}^t)\}_{t\ge 0}$ be the iterates generated by the* FastIBP *algorithm and let $(\lambda^\star, \tau^\star)$ be an optimal solution in Lemma 2.1. Then the following statement holds true:*

$$\varphi(\check{\lambda}^{t+1}, \check{\tau}^{t+1}) - (1-\theta_t)\varphi(\check{\lambda}^t, \check{\tau}^t) - \theta_t\varphi(\lambda^\star, \tau^\star) \le 2\theta_t^2\left(\sum_{k=1}^m \omega_k \left(\left\|\begin{pmatrix}\lambda_k^\star - \tilde{\lambda}_k^t \\ \tau_k^\star - \tilde{\tau}_k^t\end{pmatrix}\right\|^2 - \left\|\begin{pmatrix}\lambda_k^\star - \tilde{\lambda}_k^{t+1} \\ \tau_k^\star - \tilde{\tau}_k^{t+1}\end{pmatrix}\right\|^2\right)\right).$$

Proof. Using Lemma 2.4 with $(\lambda', \tau') = (\widehat{\lambda}^{t+1}, \widehat{\tau}^{t+1})$ and $(\lambda, \tau) = (\bar{\lambda}^t, \bar{\tau}^t)$, we have

$$\varphi(\widehat{\lambda}^{t+1}, \widehat{\tau}^{t+1}) \ \le \ \varphi(\bar{\lambda}^t, \bar{\tau}^t) + \theta_t \begin{pmatrix}\tilde{\lambda}^{t+1} - \tilde{\lambda}^t \\ \tilde{\tau}^{t+1} - \tilde{\tau}^t\end{pmatrix}^\top \nabla\varphi(\bar{\lambda}^t, \bar{\tau}^t) + 2\theta_t^2\left(\sum_{k=1}^m \omega_k \left\|\begin{pmatrix}\tilde{\lambda}_k^{t+1} - \tilde{\lambda}_k^t \\ \tilde{\tau}_k^{t+1} - \tilde{\tau}_k^t\end{pmatrix}\right\|^2\right).$$

After some simple calculations, we find that

$$\begin{aligned}\varphi(\bar{\lambda}^t, \bar{\tau}^t) \ &= \ (1-\theta_t)\varphi(\bar{\lambda}^t, \bar{\tau}^t) + \theta_t\varphi(\bar{\lambda}^t, \bar{\tau}^t), \\ \begin{pmatrix}\tilde{\lambda}^{t+1} - \tilde{\lambda}^t \\ \tilde{\tau}^{t+1} - \tilde{\tau}^t\end{pmatrix}^\top \nabla\varphi(\bar{\lambda}^t, \bar{\tau}^t) \ &= \ -\begin{pmatrix}\tilde{\lambda}^t - \bar{\lambda}^t \\ \tilde{\tau}^t - \bar{\tau}^t\end{pmatrix}^\top \nabla\varphi(\bar{\lambda}^t, \bar{\tau}^t) + \begin{pmatrix}\tilde{\lambda}^{t+1} - \bar{\lambda}^t \\ \tilde{\tau}^{t+1} - \bar{\tau}^t\end{pmatrix}^\top \nabla\varphi(\bar{\lambda}^t, \bar{\tau}^t).\end{aligned}$$

Putting these pieces together yields that

$$\begin{aligned}\varphi(\widehat{\lambda}^{t+1}, \widehat{\tau}^{t+1}) \ \le \ & \underbrace{(1-\theta_t)\varphi(\bar{\lambda}^t, \bar{\tau}^t) - \theta_t \begin{pmatrix}\tilde{\lambda}^t - \bar{\lambda}^t \\ \tilde{\tau}^t - \bar{\tau}^t\end{pmatrix}^\top \nabla\varphi(\bar{\lambda}^t, \bar{\tau}^t)}_{\text{I}} \qquad (15) \\ & + \theta_t\left(\underbrace{\varphi(\bar{\lambda}^t, \bar{\tau}^t) + \begin{pmatrix}\tilde{\lambda}^{t+1} - \bar{\lambda}^t \\ \tilde{\tau}^{t+1} - \bar{\tau}^t\end{pmatrix}^\top \nabla\varphi(\bar{\lambda}^t, \bar{\tau}^t) + 2\theta_t\left(\sum_{k=1}^m \omega_k \left\|\begin{pmatrix}\tilde{\lambda}^{t+1} - \tilde{\lambda}^t \\ \tilde{\tau}^{t+1} - \tilde{\tau}^t\end{pmatrix}\right\|^2\right)}_{\text{II}}\right).\end{aligned}$$

For the term I in equation (15), we derive from the definition of $(\bar{\lambda}^t, \bar{\tau}^t)$ that

$$-\theta_t\begin{pmatrix}\tilde{\lambda}^t - \bar{\lambda}^t \\ \tilde{\tau}^t - \bar{\tau}^t\end{pmatrix} \ = \ \theta_t\begin{pmatrix}\bar{\lambda}^t \\ \bar{\tau}^t\end{pmatrix} + (1-\theta_t)\begin{pmatrix}\check{\lambda}^t \\ \check{\tau}^t\end{pmatrix} - \begin{pmatrix}\bar{\lambda}^t \\ \bar{\tau}^t\end{pmatrix} \ = \ (1-\theta_t)\begin{pmatrix}\check{\lambda}^t - \bar{\lambda}^t \\ \check{\tau}^t - \bar{\tau}^t\end{pmatrix}.$$

Using this equality and the convexity of φ, we have

$$\text{I} \ = \ (1-\theta_t)\left(\varphi(\bar{\lambda}^t, \bar{\tau}^t) + \begin{pmatrix}\check{\lambda}^t - \bar{\lambda}^t \\ \check{\tau}^t - \bar{\tau}^t\end{pmatrix}^\top \nabla\varphi(\bar{\lambda}^t, \bar{\tau}^t)\right) \ \le \ (1-\theta_t)\varphi(\check{\lambda}^t, \check{\tau}^t). \qquad (16)$$

For the term II in equation (15), the definition of $(\tilde{\lambda}^{t+1}, \tilde{\tau}^{t+1})$ implies that

$$\begin{pmatrix} \lambda - \tilde{\lambda}^{t+1} \\ \tau - \tilde{\tau}^{t+1} \end{pmatrix}^\top \left(\nabla \varphi(\bar{\lambda}^t, \bar{\tau}^t) + 4\theta_t \begin{pmatrix} \omega_1(\tilde{\lambda}_1^{t+1} - \tilde{\lambda}_1^t) \\ \vdots \\ \omega_m(\tilde{\lambda}_m^{t+1} - \tilde{\lambda}_m^t) \\ \omega_1(\tilde{\tau}_1^{t+1} - \tilde{\tau}_1^t) \\ \vdots \\ \omega_m(\tilde{\tau}_m^{t+1} - \tilde{\tau}_m^t) \end{pmatrix} \right) \geq 0, \quad \text{for all } (\lambda, \tau) \in \mathbb{R}^{mn} \times \mathcal{P}.$$

Letting $(\lambda, \tau) = (\lambda^\star, \tau^\star)$ and rearranging the resulting inequality yields that

$$\begin{aligned} & \begin{pmatrix} \tilde{\lambda}^{t+1} - \bar{\lambda}^t \\ \tilde{\tau}^{t+1} - \bar{\tau}^t \end{pmatrix}^\top \nabla \varphi(\bar{\lambda}^t, \bar{\tau}^t) + 2\theta_t \left(\sum_{k=1}^m \omega_k \left\| \begin{pmatrix} \tilde{\lambda}_k^{t+1} - \tilde{\lambda}_k^t \\ \tilde{\tau}_k^{t+1} - \tilde{\tau}_k^t \end{pmatrix} \right\|^2 \right) \\ \leq \ & \begin{pmatrix} \lambda^\star - \bar{\lambda}^t \\ \tau^\star - \bar{\tau}^t \end{pmatrix}^\top \nabla \varphi(\bar{\lambda}^t, \bar{\tau}^t) + 2\theta_t \left(\sum_{k=1}^m \omega_k \left(\left\| \begin{pmatrix} \lambda_k^\star - \tilde{\lambda}_k^t \\ \tau_k^\star - \tilde{\tau}_k^t \end{pmatrix} \right\|^2 - \left\| \begin{pmatrix} \lambda_k^\star - \tilde{\lambda}_k^{t+1} \\ \tau_k^\star - \tilde{\tau}_k^{t+1} \end{pmatrix} \right\|^2 \right) \right). \end{aligned}$$

Using the convexity of φ again, we have

$$\begin{pmatrix} \lambda^\star - \bar{\lambda}^t \\ \tau^\star - \bar{\tau}^t \end{pmatrix}^\top \nabla \varphi(\bar{\lambda}^t, \bar{\tau}^t) \ \leq \ \varphi(\lambda^\star, \tau^\star) - \varphi(\bar{\lambda}^t, \bar{\tau}^t).$$

Putting these pieces together yields that

$$\mathrm{I} \ \leq \ \varphi(\lambda^\star, \tau^\star) + 2\theta_t \left(\sum_{k=1}^m \omega_k \left(\left\| \begin{pmatrix} \lambda_k^\star - \tilde{\lambda}_k^t \\ \tau_k^\star - \tilde{\tau}_k^t \end{pmatrix} \right\|^2 - \left\| \begin{pmatrix} \lambda_k^\star - \tilde{\lambda}_k^{t+1} \\ \tau_k^\star - \tilde{\tau}_k^{t+1} \end{pmatrix} \right\|^2 \right) \right). \tag{17}$$

Plugging Eq. (16) and Eq. (17) into Eq. (15) yields that

$$\varphi(\widehat{\lambda}^{t+1}, \widehat{\tau}^{t+1}) \leq (1-\theta_t)\varphi(\check{\lambda}^t, \check{\tau}^t) + \theta_t \varphi(\lambda^\star, \tau^\star) + 2\theta_t^2 \left(\sum_{k=1}^m \omega_k \left(\left\| \begin{pmatrix} \lambda_k^\star - \tilde{\lambda}_k^t \\ \tau_k^\star - \tilde{\tau}_k^t \end{pmatrix} \right\|^2 - \left\| \begin{pmatrix} \lambda_k^\star - \tilde{\lambda}_k^{t+1} \\ \tau_k^\star - \tilde{\tau}_k^{t+1} \end{pmatrix} \right\|^2 \right) \right).$$

Since $(\check{\lambda}^{t+1}, \check{\tau}^{t+1})$ is obtained by an exact coordinate update from (λ^t, τ^t), we have $\varphi(\lambda^t, \tau^t) \geq \varphi(\check{\lambda}^{t+1}, \check{\tau}^{t+1})$. Using the similar argument, we have $\varphi(\acute{\lambda}^t, \acute{\tau}^t) \geq \varphi(\grave{\lambda}^t, \dot{\tau}^t) \geq \varphi(\lambda^t, \tau^t)$. By the definition of $(\acute{\lambda}^t, \acute{\tau}^t)$, we have $\varphi(\widehat{\lambda}^t, \widehat{\tau}^t) \geq \varphi(\acute{\lambda}^t, \acute{\tau}^t)$. Putting these pieces together yields the desired inequality. □

4.3 Main result

We present an upper bound for the iteration numbers required by the FastIBP algorithm.

Theorem 4.6. *Let $\{(\lambda^t, \tau^t)\}_{t \geq 0}$ be the iterates generated by the* FastIBP *algorithm. Then the number of iterations required to reach the stopping criterion $E_t \leq \varepsilon$ satisfies*

$$t \ \leq \ 1 + 10 \left(\frac{n(R_\lambda^2 + R_\tau^2)}{\varepsilon^2} \right)^{1/3},$$

where $R_\lambda, R_\tau > 0$ are defined in Lemma 2.1.

Proof. First, let $\delta_t = \varphi(\check{\lambda}^t, \check{\tau}^t) - \varphi(\lambda^\star, \tau^\star)$, we show that

$$\delta_t \;\leq\; \frac{8n(R_\lambda^2 + R_\tau^2)}{(t+1)^2}. \tag{18}$$

Indeed, by Lemma 4.2 and 4.5, we have

$$\left(\frac{1-\theta_{t+1}}{\theta_{t+1}^2}\right)\delta_{t+1} - \left(\frac{1-\theta_t}{\theta_t^2}\right)\delta_t \;\leq\; 2\left(\sum_{k=1}^m \omega_k \left(\left\|\begin{pmatrix}\lambda_k^\star - \tilde{\lambda}_k^t \\ \tau_k^\star - \tilde{\tau}_k^t\end{pmatrix}\right\|^2 - \left\|\begin{pmatrix}\lambda_k^\star - \tilde{\lambda}_k^{t+1} \\ \tau_k^\star - \tilde{\tau}_k^{t+1}\end{pmatrix}\right\|^2\right)\right).$$

By unrolling the recurrence and using $\theta_0 = 1$ and $\tilde{\lambda}_0 = \tilde{\tau}_0 = \mathbf{0}_{mn}$, we have

$$\begin{aligned}\left(\frac{1-\theta_t}{\theta_t^2}\right)\delta_t + 2\left(\sum_{k=1}^m \omega_k \left\|\begin{pmatrix}\lambda_k^\star - \tilde{\lambda}_k^t \\ \tau_k^\star - \tilde{\tau}_k^t\end{pmatrix}\right\|^2\right) \;&\leq\; \left(\frac{1-\theta_0}{\theta_0^2}\right)\delta_0 + 2\left(\sum_{k=1}^m \omega_k \left\|\begin{pmatrix}\lambda_k^\star - \tilde{\lambda}_k^0 \\ \tau_k^\star - \tilde{\tau}_k^0\end{pmatrix}\right\|^2\right) \\ &\leq\; 2\left(\sum_{k=1}^m \omega_k \left\|\begin{pmatrix}\lambda_k^\star \\ \tau_k^\star\end{pmatrix}\right\|^2\right) \overset{\text{Corollary 2.3}}{\leq} 2n(R_\lambda^2 + R_\tau^2).\end{aligned}$$

For $t \geq 1$, Lemma 4.2 implies that $\theta_{t-1}^{-2} = (1-\theta_t)\theta_t^{-2}$. Therefore, we conclude that

$$\delta_t \;\leq\; 2\theta_{t-1}^2 n(R_\lambda^2 + R_\tau^2).$$

This together with the fact that $0 < \theta_{t-1} \leq 2/(t+1)$ yields the desired inequality.

Furthermore, we show that

$$\delta_t - \delta_{t+1} \;\geq\; \frac{E_t^2}{11}. \tag{19}$$

Indeed, by the definition of Δ_t, we have

$$\delta_t - \delta_{t+1} \;=\; \varphi(\check{\lambda}^t, \check{\tau}^t) - \varphi(\check{\lambda}^{t+1}, \check{\tau}^{t+1}) \;\geq\; \varphi(\lambda^t, \tau^t) - \varphi(\check{\lambda}^{t+1}, \check{\tau}^{t+1}).$$

By the definition of φ and the update formula of $(\check{\lambda}^{t+1}, \check{\tau}^{t+1})$, we have

$$\begin{aligned}\varphi(\lambda^t, \tau^t) - \varphi(\check{\lambda}^{t+1}, \check{\tau}^{t+1}) \;&=\; \sum_{k=1}^m \omega_k(\mathbf{1}_n^\top B_k(\lambda_k^t, \tau_k^t)\mathbf{1}_n - \mathbf{1}_n^\top B_k(\check{\lambda}_k^{t+1}, \check{\tau}_k^{t+1})\mathbf{1}_n) \\ &=\; \mathbf{1}_n^\top\left(\sum_{k=1}^m \omega_k c(B_k(\lambda_k^t, \tau_k^t)) - e^{\sum_{k=1}^m \omega_k \log(c(B_k(\lambda_k^t, \tau_k^t)))}\right).\end{aligned}$$

Since $r(B_k(\lambda_k^t, \tau_k^t)) = u^k \in \Delta_n$ for all $k \in [m]$, we have $\mathbf{1}_n^\top c(B_k(\lambda_k^t, \tau_k^t)) = 1$. By applying the arguments in Kroshnin et al. [2019, Lemma 6], we have

$$\varphi(\lambda^t, \tau^t) - \varphi(\check{\lambda}^{t+1}, \check{\tau}^{t+1}) \;\geq\; \frac{1}{11}\sum_{k=1}^m \omega_k \|c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\|_1^2.$$

Using the Cauchy-Schwarz inequality together with $\sum_{k=1}^m \omega_k = 1$, we have

$$E_t^2 \;\leq\; \sum_{k=1}^m \omega_k \|c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i, \tau_i))\|_1^2.$$

Putting these pieces together yields the desired inequality.

Finally, we derive from Eq. (18) and (19) and the non-negativeness of δ_t that

$$\sum_{i=t}^{+\infty} E_i^2 \;\leq\; 11\left(\sum_{i=t}^{+\infty}(\delta_i - \delta_{i+1})\right) \;\leq\; 11\delta_t \;\leq\; \frac{88n(R_\lambda^2 + R_\tau^2)}{(t+1)^2}$$

Let $T > 0$ satisfy $E_T \leq \varepsilon$, we have $E_t > \varepsilon$ for all $t \in [T]$. Without loss of generality, we assume T is even. Then the following statement holds true:

$$\varepsilon^2 \;\leq\; \frac{704n(R_\lambda^2 + R_\tau^2)}{T^3}.$$

Rearranging the above inequality yields the desired inequality. □

Equipped with the result of Theorem 4.6, we are ready to present the complexity bound of Algorithm 2 for approximating the FS-WBP in Eq. (2).

Theorem 4.7. *The* FASTIBP *algorithm for approximately solving the FS-WBP in Eq.* (2) *(Algorithm 2) returns an ε-approximate barycenter $\widehat{u} \in \mathbb{R}^n$ within*

$$O\left(mn^{7/3}\left(\frac{(\max_{1\leq k\leq m}\|C_k\|_\infty)\sqrt{\log(n)}}{\varepsilon}\right)^{4/3}\right)$$

arithmetic operations.

Proof. Consider the iterate $(\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_m)$ be generated by the FASTIBP algorithm (cf. Algorithm 1), the rounding scheme (cf. Kroshnin et al. [2019, Algorithm 4]) returns the feasible solution $(\widehat{X}_1, \widehat{X}_2, \ldots, \widehat{X}_m)$ to the FS-WBP in Eq. (2) and $c(\widehat{X}_k)$ are the same for all $k \in [m]$.

To show that $\widehat{u} = \sum_{k=1}^m \omega_k c(\widehat{X}_k)$ is an ε-approximate barycenter (cf. Definition 2.2), it suffices to show that

$$\sum_{k=1}^m \omega_k\langle C_k, \widehat{X}_k\rangle \;\leq\; \sum_{k=1}^m \omega_k\langle C_k, X_k^\star\rangle + \varepsilon, \tag{20}$$

where $(X_1^\star, X_2^\star, \ldots, X_m^\star)$ is a set of optimal transportation plan between m measures $\{u^k\}_{k\in[m]}$ and the barycenter of the FS-WBP.

First, we derive from the scheme of Kroshnin et al. [2019, Algorithm 4] that the following inequality holds for all $k \in [m]$,

$$\|\widehat{X}_k - \widetilde{X}_k\|_1 \;\leq\; \|c(\widetilde{X}_k) - \sum_{i=1}^m \omega_i c(\widetilde{X}_i)\|_1.$$

This together with the Hölder's inequality implies that

$$\sum_{k=1}^m \omega_k\langle C_k, \widehat{X}_k - \widetilde{X}_k\rangle \;\leq\; \left(\max_{1\leq k\leq m}\|C_k\|_\infty\right)\left(\sum_{k=1}^m \omega_k\|c(\widetilde{X}_k) - \sum_{i=1}^m \omega_i c(\widetilde{X}_i)\|_1\right). \tag{21}$$

Furthermore, we have

$$\begin{aligned}\sum_{k=1}^m \omega_k\langle C_k, \widetilde{X}_k - X_k^\star\rangle \;&=\; \sum_{k=1}^m \omega_k(\langle C_k, \widetilde{X}_k\rangle - \eta H(\widetilde{X}_k)) - \sum_{k=1}^m \omega_k(\langle C_k, X_k^\star\rangle - \eta H(X_k^\star)) \\ &\quad + \sum_{k=1}^m \omega_k\eta H(\widetilde{X}_k) - \sum_{k=1}^m \omega_k\eta H(X_k^\star).\end{aligned}$$

Since $0 \leq H(X) \leq 2\log(n)$ for any $X \in \mathbb{R}_+^{n\times n}$ satisfying that $\|X\|_1 = 1$ [Cover and Thomas, 2012] and $\sum_{k=1}^m \omega_k = 1$, we have

$$\sum_{k=1}^m \omega_k \langle C_k, \widetilde{X}_k - X_k^\star\rangle \leq 2\eta\log(n) + \sum_{k=1}^m \omega_k(\langle C_k, \widetilde{X}_k\rangle - \eta H(\widetilde{X}_k)) - \sum_{k=1}^m \omega_k(\langle C_k, X_k^\star\rangle - \eta H(X_k^\star)).$$

Let $(X_1^\eta, X_2^\eta, \ldots, X_m^\eta)$ be a set of optimal transportation plans to the entropic regularized FS-WBP in Eq. (3), we have

$$\sum_{k=1}^m \omega_k(\langle C_k, X_k^\eta\rangle - \eta H(X_k^\eta)) \ \leq \ \sum_{k=1}^m \omega_k(\langle C_k, X_k^\star\rangle - \eta H(X_k^\star)).$$

By the optimality of $(X_1^\eta, X_2^\eta, \ldots, X_m^\eta)$, we have

$$\sum_{k=1}^m \omega_k(\langle C_k, X_k^\eta\rangle - \eta H(X_k^\eta)) \ = \ -\eta\left(\min_{\lambda\in\mathbb{R}^{mn}, \tau\in\mathcal{P}} \varphi(\lambda, \tau)\right) \ \geq \ -\eta\varphi(\lambda^t, \tau^t).$$

Since $(\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_m)$ is generated by the FastIBP algorithm, we have $\widetilde{X}_k = B_k(\lambda_k^t, \tau_k^t)$ for all $k \in [m]$ where (λ^t, τ^t) are the dual iterates. Then

$$\begin{aligned}
\sum_{k=1}^m \omega_k(\langle C_k, \widetilde{X}_k\rangle - \eta H(\widetilde{X}_k)) \ &= \ \sum_{k=1}^m \omega_k(\langle C_k, B_k(\lambda_k^t, \tau_k^t)\rangle - \eta H(B_k(\lambda_k^t, \tau_k^t))) \\
&= \ -\eta\left(\sum_{k=1}^m \omega_k(\mathbf{1}_n^\top B_k(\lambda_k^t, \tau_k^t)\mathbf{1}_n - (\lambda_k^t)^\top u^k)\right) + \eta\sum_{k=1}^m \omega_k(\tau_k^t)^\top c(B_k(\lambda_k^t, \tau_k^t)) \\
&= \ -\eta\varphi(\lambda^t, \tau^t) + \eta\left(\sum_{k=1}^m \omega_k(\tau_k^t)^\top\left(c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\right)\right).
\end{aligned}$$

Putting these pieces together yields that

$$\sum_{k=1}^m \omega_k\langle C_k, \widetilde{X}_k - X_k^\star\rangle \ \leq \ 2\eta\log(n) + \eta\left(\sum_{k=1}^m \omega_k(\tau_k^t)^\top\left(c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\right)\right).$$

Since $\mathbf{1}_n^\top c(B_k(\lambda_k^t, \tau_k^t)) = 1$ for all $k \in [m]$, we have

$$\begin{aligned}
&\left(\sum_{k=1}^m \omega_k(\tau_k^t)^\top\left(c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\right)\right) \\
= \ &\left(\sum_{k=1}^m \omega_k\left(\tau_k^t - \frac{\max_{1\leq i\leq n}(\tau_k^t)_i + \min_{1\leq i\leq n}(\tau_k^t)_i}{2}\mathbf{1}_n\right)^\top\left(c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\right)\right) \\
\leq \ &\left\|\tau_k^t - \frac{\max_{1\leq i\leq n}(\tau_k^t)_i + \min_{1\leq i\leq n}(\tau_k^t)_i}{2}\mathbf{1}_n\right\|_\infty\left(\sum_{k=1}^m \omega_k\|c(\widetilde{X}_k) - \sum_{i=1}^m \omega_i c(\widetilde{X}_i)\|_1\right).
\end{aligned}$$

Using Lemma 4.4, we have

$$\left\|\tau_k^t - \frac{\max_{1\leq i\leq n}(\tau_k^t)_i + \min_{1\leq i\leq n}(\tau_k^t)_i}{2}\mathbf{1}_n\right\|_\infty \ \leq \ \frac{R_\tau}{2}.$$

Putting these pieces together yields that

$$\sum_{k=1}^m \omega_k \langle C_k, \widetilde{X}_k - X_k^\star \rangle \;\leq\; 2\eta \log(n) + \frac{\eta R_\tau}{2} \left(\sum_{k=1}^m \omega_k \| c(\widetilde{X}_k) - \sum_{i=1}^m \omega_i c(\widetilde{X}_i) \|_1 \right). \tag{22}$$

Recall that $E_t = \sum_{k=1}^m \omega_k \|c(\widetilde{X}_k) - \sum_{i=1}^m \omega_i c(\widetilde{X}_i)\|_1$ and $R_\tau = 4\eta^{-1}(\max_{1 \leq k \leq m} \|C_k\|_\infty)$. Then Eq. (21) and Eq. (22) together imply that

$$\sum_{k=1}^m \omega_k \langle C_k, \widehat{X}_k - X_k^\star \rangle \;\leq\; 2\eta \log(n) + 3 \left(\max_{1 \leq k \leq m} \|C_k\|_\infty \right) E_t.$$

This together with $E_t \leq \bar{\varepsilon}/2$ and the choice of η and $\bar{\varepsilon}$ implies Eq. (20) as desired.

Complexity bound estimation. We first bound the number of iterations required by the FastIBP algorithm (cf. Algorithm 1) to reach $E_t \leq \bar{\varepsilon}/2$. Indeed, Theorem 4.6 implies that

$$t \;\leq\; 1 + 20 \left(\frac{n(R_\lambda^2 + R_\tau^2)}{\bar{\varepsilon}^2} \right)^{1/3} \;\leq\; 20\sqrt[3]{n} \left(\frac{R_\lambda + R_\tau}{\bar{\varepsilon}} \right)^{2/3}.$$

For the simplicity, we let $\bar{C} = \max_{1 \leq k \leq m} \|C_k\|_\infty$. Using the definition of R_λ and R_τ in Lemma 2.1, the construction of $\{\tilde{u}^k\}_{k \in [m]}$ and the choice of η and $\bar{\varepsilon}$, we have

$$\begin{aligned} t \;&\leq\; 1 + 20\sqrt[3]{n} \left(\frac{4\bar{C}}{\varepsilon} \left(\frac{36 \log(n) \bar{C}}{\varepsilon} + \log(n) - \log\left(\frac{16 n \bar{C}}{\varepsilon} \right) \right) \right)^{2/3} \\ &= \; O\left(\sqrt[3]{n} \left(\frac{\bar{C}\sqrt{\log(n)}}{\varepsilon} \right)^{4/3} \right). \end{aligned}$$

Recall that each iteration of the FastIBP algorithm requires $\mathcal{O}(mn^2)$ arithmetic operations, the total arithmetic operations required by the FastIBP algorithm as the subroutine in Algorithm 2 is bounded by

$$O\left(mn^{7/3} \left(\frac{\bar{C}\sqrt{\log(n)}}{\varepsilon} \right)^{4/3} \right).$$

Computing a collection of vectors $\{\tilde{u}^k\}_{k \in [m]}$ needs $\mathcal{O}(mn)$ arithmetic operations while the rounding scheme in Kroshnin et al. [2019, Algorithm 4] requires $\mathcal{O}(mn^2)$ arithmetic operations. Putting these pieces together yields that the desired complexity bound of Algorithm 2. □

Remark 4.8. *First, we notice that $(\widehat{X}_1, \widehat{X}_2, \ldots, \widehat{X}_m)$ are one set of approximate optimal transportation plans between m measures $\{u^k\}_{k \in [m]}$ and an ε-approximate barycenter $\widehat{u}$. These matrices are equivalent to those constructed by using [Altschuler et al., 2017, Algorithm 2]. We also remark that the approximate barycenter $\widehat{u}$ can be constructed by only using $(\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_m)$; see [Kroshnin et al., 2019, Section 2.2] for the details.*

5 Experiments

In this section, we conduct extensive numerical experiments to evaluate the FastIBP algorithm for computing the fixed-support Wasserstein barycenters, i.e., solving Eq. (2). In all our experiments, we consider the Wasserstein distance with ℓ_2-norm, i.e., second order Wasserstein distance. We also compare our FastIBP algorithm with the commercial software Gurobi and the existing state-of-the-art algorithms, including the iterative Bregman projection (IBP) algorithm [Benamou et al., 2015] and the modified Bregman ADMM (BADMM) [Ye et al., 2017]. All the experiments are run in MATLAB R2016a on a MacBook Pro with a Intel i5 2.6GHz Intel Core i5 (this processor has 2 cores and 4 threads) and 16GB memory, equipped with Mac OS High Sierra 10.13.6.

5.1 Implementation details

For the FastIBP algorithm, the regularization parameter η is chosen from $\{0.01, 0.001\}$ in our experiments. We follow Benamou et al. [2015, Remark 3] to implement the algorithm and terminate it when

$$\begin{aligned}
\frac{\sum_{k=1}^m \omega_k \|c(B_k(\lambda_k^t, \tau_k^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\|}{1 + \sum_{k=1}^m \omega_k \|c(B_k(\lambda_k^t, \tau_k^t))\| + \|\sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\|} &\leq \text{Tol}_{\mathsf{fibp}}, \\
\frac{\sum_{k=1}^m \omega_k \|r(B_k(\lambda_k^t, \tau_k^t)) - u^k\|}{1 + \sum_{k=1}^m \omega_k \|r(B_k(\lambda_k^t, \tau_k^t))\| + \sum_{k=1}^m \omega_k \|u^k\|} &\leq \text{Tol}_{\mathsf{fibp}}, \\
\frac{\|\sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t)) - \sum_{i=1}^m \omega_i c(B_i(\lambda_i^{t-1}, \tau_i^{t-1}))\|}{1 + \|\sum_{i=1}^m \omega_i c(B_i(\lambda_i^t, \tau_i^t))\| + \|\sum_{i=1}^m \omega_i c(B_i(\lambda_i^{t-1}, \tau_i^{t-1}))\|} &\leq \text{Tol}_{\mathsf{fibp}}, \\
\frac{\sum_{k=1}^m \omega_k \|B_k(\lambda_k^t, \tau_k^t) - B_k(\lambda_k^{t-1}, \tau_k^{t-1})\|_F}{1 + \sum_{k=1}^m \omega_k \|B_k(\lambda_k^t, \tau_k^t)\|_F + \sum_{k=1}^m \omega_k \|B_k(\lambda_k^{t-1}, \tau_k^{t-1})\|_F} &\leq \text{Tol}_{\mathsf{fibp}}, \\
\frac{\sum_{k=1}^m \omega_k \|\lambda_k^t - \lambda_k^{t-1}\|}{1 + \sum_{k=1}^m \omega_k \|\lambda_k^t\| + \sum_{k=1}^m \omega_k \|\lambda_k^{t-1}\|} &\leq \text{Tol}_{\mathsf{fibp}}, \\
\frac{\sum_{k=1}^m \omega_k \|\tau_k^t - \tau_k^{t-1}\|}{1 + \sum_{k=1}^m \omega_k \|\tau_k^t\| + \sum_{k=1}^m \omega_k \|\tau_k^{t-1}\|} &\leq \text{Tol}_{\mathsf{fibp}}.
\end{aligned}$$

These inequalities guarantee that (i) the infeasibility violations for marginal constraints, (ii) the iterative gap between approximate barycenters, and (iii) the iterative gap between dual variables are relatively small. Computing all the above residuals is expensive. Thus, in our implementations, we only compute them and check the termination criteria at every 20 iterations when $\eta = 0.01$ and every 200 iteration when $\eta = 0.001$. We set $\text{Tol}_{\mathsf{fibp}} = 10^{-6}$ and $\text{MaxIter}_{\mathsf{fibp}} = 10000$ on synthetic data and $\text{Tol}_{\mathsf{fibp}} = 10^{-10}$ on MNIST images.

For IBP and BADMM, we use the Matlab code[1] implemented by Ye et al. [2017] and terminate them with the refined stopping criterion provided by Yang et al. [2018]. The regularization parameter η for the IBP algorithm is still chosen from $\{0.01, 0.001\}$. For synthetic data, we set $\text{Tol}_{\mathsf{badmm}} = 10^{-5}$ and $\text{Tol}_{\mathsf{ibp}} = 10^{-6}$ with $\text{MaxIter}_{\mathsf{badmm}} = 5000$ and $\text{MaxIter}_{\mathsf{ibp}} = 10000$. For MNIST images, we set $\text{Tol}_{\mathsf{ibp}} = 10^{-10}$.

For the linear programming algorithm, we apply Gurobi 9.0.2 (Gurobi Optimization, 2019) (with an academic license) to solve the FS-WBP in Eq. (2). Since Gurobi can provide high quality solutions when the problem of medium size, we use the solution obtained by Gurobi as

[1] Available in https://github.com/bobye/WBC_Matlab

a benchmark to evaluate the qualities of solution obtained by different algorithms on synthetic data. In our experiments, we force Gurobi to *only run the dual simplex algorithm* and use other parameters in the default settings.

For the evaluation metrics, "**normalized obj**" stands for the normalized objective value which is defined by

$$\textbf{normalized obj} := \frac{|\sum_{k=1}^m \omega_k \langle C_k, \widehat{X}_k\rangle - \sum_{k=1}^m \omega_k \langle C_k, X_k^{\mathbf{g}}\rangle|}{|\sum_{k=1}^m \omega_k \langle C_k, X_k^{\mathbf{g}}\rangle|}$$

where $(\widehat{X}_1, \ldots, \widehat{X}_m)$ is the solution obtained by each algorithm and $(X_1^{\mathbf{g}}, \ldots, X_m^{\mathbf{g}})$ denotes the solution obtained by Gurobi. Additionally, "**feasibility**" denotes the the deviation of the terminating solution from the feasible set[2]; see Yang et al. [2018, Section 5.1]. Furthermore, "**iteration**" denotes the number of iterations. Finally, "**time (in seconds)**" denotes the computational time.

In what follows, we present our experimental results. In Section 5.2, we evaluate all the candidate algorithms on synthetic data and compare their computational performance in terms of accuracy and speed. In Section 5.3, we compare our algorithm with IBP on the MNIST dataset to visualize the quality of approximate barycenters obtained by each algorithm. For the simplicity of the presentation, in our figures "g" stands for Gurobi; "b" stands for BADMM; "i1" and "i2" stand for the IBP algorithm with $\eta = 0.01$ and $\eta = 0.001$; "f1" and "f2" stand for the FastIBP algorithm with $\eta = 0.01$ and $\eta = 0.001$.

5.2 Experiments on synthetic data

In this section, we generate a set of discrete probability distributions $\{\mu_k\}_{k=1}^m$ with $\mu_k = \{(u_i^k, \mathbf{x}_i) \in \mathbb{R}_+ \times \mathbb{R}^d \mid i \in [n]\}$ and $\sum_{i=1}^n u_i^k = 1$. The fixed-support Wasserstein barycenter $\widehat{\mu} = \{(\widehat{u}_i, \mathbf{x}_i) \in \mathbb{R}_+ \times \mathbb{R}^d \mid i \in [n]\}$ where $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ are known and shared with $\{\mu_k\}_{k=1}^m$. In our experiment, we set $d = 3$ and choose different values of (m, n). Then, given each tuple (m, n), we randomly generate a trial as follows.

First, we generate the support points $(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_n^k)$ whose entries are drawn from a Gaussian mixture distribution via the Matlab commands provided by Yang et al. [2018]:

```
gm_num = 5; gm_mean = [-20; -10; 0; 10; 20];
sigma = zeros(1, 1, gm_num); sigma(1, 1, :) = 5*ones(gm_num, 1);
gm_weights = rand(gm_num, 1); gm_weights = gm_weights/sum(gm_weights);
distrib = gmdistribution(gm_mean, sigma, gm_weights);
```

For each $k \in [m]$, we generate the weight vector $(u_1^k, u_2^k, \ldots, u_n^k)$ whose entries are drawn from the uniform distribution on the interval (0, 1), and normalize it such that $\sum_{i=1}^n u_i^k = 1$. After generating all $\{\mu_k\}_{k=1}^m$, we use the k-means[3] method to choose n points from $\{\mathbf{x}_i^k \mid i \in [n], k \in [m]\}$ to be the support points of the barycenter. Finally, we generate the weight vector $(\omega_1, \omega_2, \ldots, \omega_m)$ whose entries are drawn from the uniform distribution on the interval $(0, 1)$, and normalize it such that $\sum_{k=1}^m \omega_k = 1$.

We present some representative numerical results in Figure 2. Given $n = 100$, we evaluate the performance of FastIBP, IBP, BADMM algorithms, and commercial software Gurobi by

[2] Since we do not put the iterative gap between dual variables in "**feasibility**" and the FS-WBP is relatively easier than general WBP, our results for BADMM and IBP are consistently smaller than that presented by Ye et al. [2017], Yang et al. [2018], Ge et al. [2019].

[3] In our experiments, we call the Matlab function kmeans, which is built in machine learning toolbox.

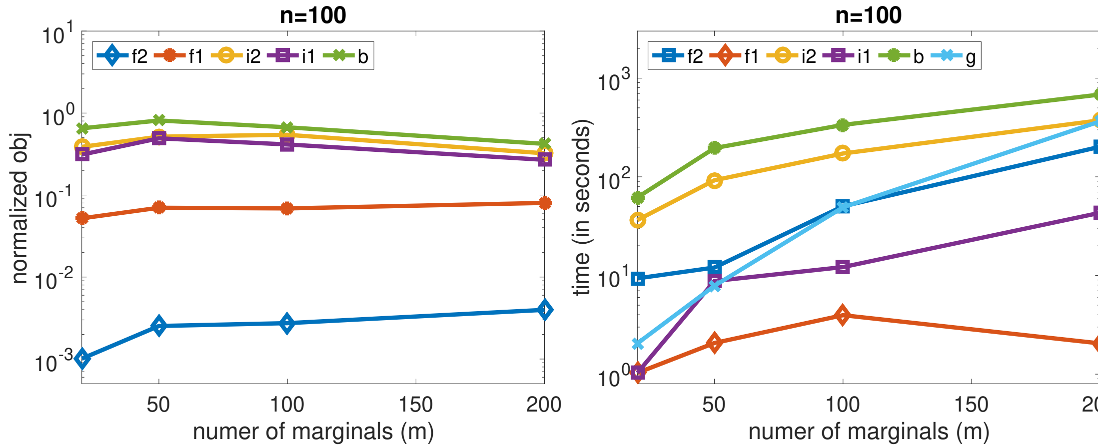


Figure 2: The left and right figures present the average normalized objective value and computational time (in seconds) of all the algorithms from 10 independent trials.

varying $m \in \{20, 50, 100, 200\}$. As indicated in Figure 2, the FastIBP algorithm performs better than BADMM and IBP algorithms in the sense that it consistently returns an objective value closer to that of Gurobi in less computational time. More specifically, IBP achieves high feasibility accuracy and converges very fast when $\eta = 0.01$, but suffers from a crude solution with poor objective value; BADMM takes much more time with unsatisfactory objective value, and is not provably convergent in theory; Gurobi is highly optimized and can solve the problem of relatively small size very efficiently. However, when the problem size becomes larger, Gurobi would take much more time. As an example, for the case where $(m, n) = (200, 100)$, we see that Gurobi is about 10 times slower than the FastIBP algorithm with $\eta = 0.001$ while keeping relatively small normalized objective value. To further facilitate the readers, we present the averaged results from 10 independent trials with FastIBP, IBP, BADMM algorithms, and commercial software Gurobi in Table 1.

To further compare the performances of Gurobi and the FastIBP algorithm, we conduct one more experiment with $n = 50$ and the varying number of marginals $m \in \{200, 500, 1000, 2000\}$. We keep $\text{Tol}_{\textsf{fibp}} = 10^{-6}$ but without setting the maximum iteration number. Figure 3 shows the running time taken by the two algorithms across a wide range of m and each value is an average over 5 independent trials. From the results, we observe that the FastIBP algorithm is competitive with Gurobi in terms of objective value and feasibility violation. For the computational time, the FastIBP algorithm increases linearly with respect to the number of marginals, while Gurobi increases much more rapidly. Yang et al. [2018], Ge et al. [2019] have presented similar results for Gurobi before but we hope to remark two main differences: (i) the feasibility violation in our paper is better; (ii) the computational time in our paper grows faster. This is because we force Gurobi to run the dual simplex algorithm, which iterates over the feasible solutions but can be more computationally expensive than the interior-point algorithm. This also demonstrates that the structure of the FS-WBP is not favorable to traditional LP algorithms.

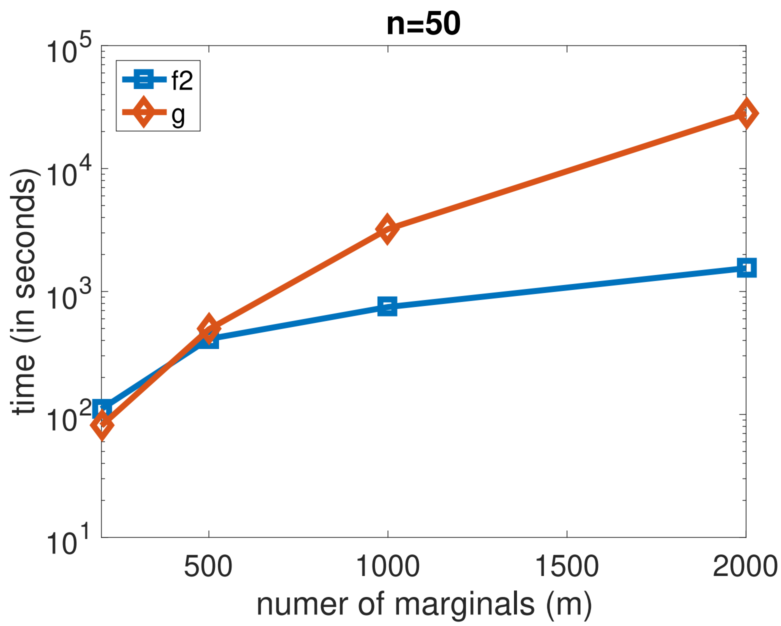
5.3 Experiments on MNIST

To better visualize the quality of approximate barycenters obtained by each algorithm, we follow Cuturi and Doucet [2014] on the MNIST[4] dataset [LeCun et al., 1998]. We randomly

[4] Available in http://yann.lecun.com/exdb/mnist/

Table 1: Numerical results on synthetic data where each distribution has different dense weights and share the same support points.

m	n	g	b	i1	i2	f1	f2
				normalized obj			
20	50	-	5.49e-01	1.89e-01	1.75e-01	6.42e-02	1.62e-03
20	100	-	6.51e-01	3.11e-01	3.87e-01	5.20e-02	1.01e-03
20	200	-	7.42e-01	4.73e-01	6.42e-01	5.24e-02	2.18e-03
50	50	-	4.28e-01	1.45e-01	1.39e-01	5.97e-02	2.18e-03
50	100	-	8.12e-01	4.92e-01	5.15e-01	7.00e-02	2.53e-03
50	200	-	6.97e-01	4.76e-01	6.46e-01	6.29e-02	4.06e-03
100	50	-	3.94e-01	1.91e-01	1.21e-01	9.35e-02	2.94e-03
100	100	-	6.70e-01	4.14e-01	5.43e-01	6.87e-02	2.74e-03
100	200	-	8.25e-01	6.23e-01	6.61e-01	6.32e-02	5.82e-03
200	50	-	2.57e-01	8.11e-02	4.48e-02	5.77e-02	2.84e-03
200	100	-	4.21e-01	2.69e-01	3.23e-01	8.02e-02	3.98e-03
200	200	-	7.31e-01	5.51e-01	7.25e-01	5.92e-02	5.40e-03
				feasibility			
20	50	4.32e-16	4.04e-07	8.79e-07	9.26e-07	7.96e-07	5.54e-07
20	100	6.73e-16	2.01e-07	8.27e-07	7.20e-07	3.04e-07	5.73e-07
20	200	1.05e-15	1.04e-07	6.49e-07	8.06e-07	1.44e-07	5.98e-07
50	50	7.37e-16	6.25e-07	9.63e-07	7.66e-07	2.68e-08	7.63e-07
50	100	1.59e-15	2.19e-07	9.59e-07	3.08e-05	6.91e-08	1.36e-08
50	200	6.58e-07	1.42e-07	8.89e-07	8.43e-07	7.99e-08	3.02e-07
100	50	1.15e-15	7.28e-07	8.94e-07	7.75e-07	3.94e-07	5.89e-07
100	100	7.88e-16	3.81e-07	9.61e-07	8.16e-07	4.51e-09	5.91e-07
100	200	2.37e-15	1.40e-07	9.93e-07	9.83e-07	2.29e-07	3.84e-10
200	50	2.34e-07	5.51e-07	3.91e-07	9.95e-07	2.75e-07	8.39e-07
200	100	1.51e-15	4.97e-07	9.86e-07	9.39e-07	9.91e-09	7.81e-07
200	200	2.41e-15	2.40e-07	9.66e-07	8.83e-06	6.80e-07	5.12e-07
				iteration			
20	50	3585	5000	240	3200	120	1800
20	100	6572	5000	280	9800	140	1200
20	200	13051	5000	220	6600	120	4400
50	50	11352	5000	580	9800	60	3000
50	100	15567	5000	920	10000	60	600
50	200	59218	5000	460	9000	60	1400
100	50	24415	5000	1160	5000	60	2600
100	100	56266	5000	640	10000	100	1200
100	200	95256	5000	720	6000	80	200
200	50	55953	5000	80	8800	60	4800
200	100	145137	5000	1040	10000	20	2600
200	200	373216	5000	1400	10000	40	400
				time (in seconds)			
20	50	0.61	15.80	0.27	3.17	0.31	4.05
20	100	2.05	61.76	1.03	36.43	1.03	9.35
20	200	9.05	278.02	3.57	91.07	3.84	159.43
50	50	3.39	39.93	1.50	21.53	0.37	14.17
50	100	7.87	195.55	8.79	92.30	2.07	12.04
50	200	64.19	743.65	18.89	375.97	5.01	113.29
100	50	14.79	73.43	5.54	21.42	0.68	27.44
100	100	49.09	335.90	12.17	172.35	3.96	50.30
100	200	163.47	1410.32	60.01	493.02	13.16	31.46
200	50	62.48	167.54	1.01	77.64	1.37	90.10
200	100	364.77	682.24	43.21	373.95	2.05	201.76
200	200	1217.02	2922.03	211.61	1475.05	12.69	110.50



m	g	f2
	normalized obj	
200	-	2.84e-03
500	-	6.67e-03
1000	-	2.74e-03
2000	-	3.52e-03
	feasibility	
200	2.34e-07	8.39e-07
500	1.61e-07	9.69e-07
1000	1.20e-15	9.52e-07
2000	1.50e-07	9.82e-07
	iteration	
200	55953	4800
500	148695	8000
1000	342266	7000
2000	799431	9200

Figure 3: Preliminary results with Gurobi and the FastIBP algorithm ($\eta = 0.001$).

select 15 images for each digit (1∼9) and resize each image to ζ times of its original size of 28×28, where ζ is drawn uniformly at random from $[0.5, 2]$. We randomly put each resized image in a larger 56×56 blank image and normalize the resulting image so that all pixel values add up to 1. Each image can be viewed as a discrete distribution supported on grids. Additionally, we set the weight vector $(\omega_1, \omega_2, \ldots, \omega_m)$ such that $\omega_k = 1/m$ for all $k \in [m]$.

We apply the FastIBP algorithm ($\eta = 0.001$) to compute the Wasserstein barycenter of the resulting images for each digit on the MNIST dataset and compare it to IBP ($\eta = 0.001$). We exclude BADMM since Yang et al. [2018, Figure 3] and Ge et al. [2019, Table 1] have shown that IBP outperforms BADMM on the MNIST dataset. The size of barycenter is set to 56×56. For a fair comparison, we do not implement convolutional technique [Solomon et al., 2015] and its stabilized version [Schmitzer, 2019, Section 4.1.2], which can be used to substantially improve IBP with small η. The approximate barycenters obtained by the FastIBP and IBP algorithms are presented in Table 2. It can be seen that the FastIBP algorithm provides a slightly "smoother" approximate barycenter than IBP when $\eta = 0.001$ is set for both. This demonstrates the good quality of the solution obtained by our algorithm.

6 Conclusions

In this paper, we study the computational hardness for solving the fixed-support Wasserstein barycenter problem (FS-WBP) and proves that the FS-WBP in the standard linear programming form is not a minimum-cost flow (MCF) problem when $m \geq 3$ and $n \geq 3$. Our results suggest that the direct application of network flow algorithms to the FS-WBP in standard LP form is inefficient, shedding the light on the practical performance of various existing algorithms, which are developed based on problem reformulation of the FS-WBP. Moreover, we propose a *deterministic* variant of iterative Bregman projection (IBP) algorithm, namely FastIBP, and prove that the complexity bound is $\widetilde{O}(mn^{7/3}\varepsilon^{-4/3})$. This bound is better than the complexity bound of $\widetilde{O}(mn^2\varepsilon^{-2})$ from the IBP algorithm in terms of ε, and that of $\widetilde{O}(mn^{5/2}\varepsilon^{-1})$ from other accelerated algorithms in terms of n. Careful experimental results on

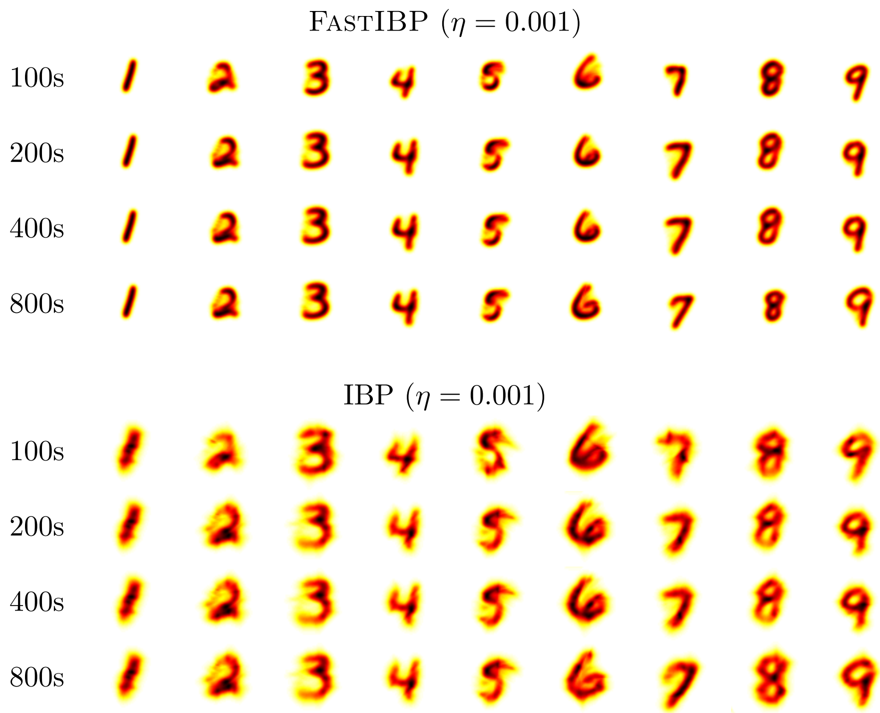


Table 2: Approximate barycenters obtained by running the FastIBP algorithm and IBP for 100s, 200s, 400s and 800s.

synthetic and real datasets demonstrate the favorable performance of the FastIBP algorithm in practice.

7 Acknowledgements

We would like to thank Lei Yang for very helpful discussion with the experiments on MNIST datasets. This work was supported in part by the Mathematical Data Science program of the Office of Naval Research under grant number N00014-18-1-2764.

References

M. Agueh and G. Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. (Not cited.)

J. Altschuler, J. Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *NIPS*, pages 1964–1974, 2017. (Not cited.)

E. Anderes, S. Borgwardt, and J. Miller. Discrete wasserstein barycenters: Optimal transport for discrete data. *Mathematical Methods of Operations Research*, 84(2):389–409, 2016. (Not cited.)

J-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2): A1111–A1138, 2015. (Not cited.)

C. Berge. *The Theory of Graphs*. Courier Corporation, 2001. (Not cited.)

J. Blanchet, A. Jambulapati, C. Kent, and A. Sidford. Towards optimal running times for optimal transport. *ArXiv Preprint: 1810.07717*, 2018. (Not cited.)

N. Bonneel, J. Rabin, G. Peyré, and H. Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. (Not cited.)

N. Bonneel, G. Peyré, and M. Cuturi. Wasserstein barycentric coordinates: histogram regression using optimal transport. *ACM Transactions on Graphics*, 35(4):71:1–71:10, 2016. (Not cited.)

S. Borgwardt and S. Patterson. Improved linear programs for discrete barycenters. *ArXiv Preprint: 1803.11313*, 2018. (Not cited.)

S. Borgwardt and S. Patterson. On the computational complexity of finding a sparse Wasserstein barycenter. *ArXiv Preprint: 1910.07568*, 2019. (Not cited.)

G. Buttazzo, L. De Pascale, and P. Gori-Giorgi. Optimal-transport formulation of electronic density-functional theory. *Physical Review A*, 85(6):062502, 2012. (Not cited.)

G. Carlier and I. Ekeland. Matching for teams. *Economic theory*, 42(2):397–418, 2010. (Not cited.)

G. Carlier, A. Oberman, and E. Oudet. Numerical methods for matching for teams and wasserstein barycenters. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6): 1621–1642, 2015. (Not cited.)

P-A. Chiappori, R. J. McCann, and L. P. Nesheim. Hedonic price equilibria, stable matching, and optimal transport: equivalence, topology, and uniqueness. *Economic Theory*, 42(2): 317–354, 2010. (Not cited.)

S. Claici, E. Chien, and J. Solomon. Stochastic Wasserstein barycenters. In *ICML*, pages 998–1007, 2018. (Not cited.)

C. Cotar, G. Friesecke, and C. Klüppelberg. Density functional theory and optimal transportation with coulomb cost. *Communications on Pure and Applied Mathematics*, 66(4):548–599, 2013. (Not cited.)

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012. (Not cited.)

M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *NIPS*, pages 2292–2300, 2013. (Not cited.)

M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *ICML*, pages 685–693, 2014. (Not cited.)

M. Cuturi and G. Peyré. A smoothed dual approach for variational Wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016. (Not cited.)

M. Cuturi and G. Peyré. Semidual regularized optimal transport. *SIAM Review*, 60(4):941–965, 2018. (Not cited.)

P. Dvurechenskii, D. Dvinskikh, A. Gasnikov, C. Uribe, and A. Nedich. Decentralize and randomize: faster algorithm for Wasserstein barycenters. In *NeurIPS*, pages 10783–10793, 2018. (Not cited.)

P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: complexity by accelerated gradient descent is better than by Sinkhorns algorithm. In *ICML*, pages 1366–1375, 2018. (Not cited.)

O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015. (Not cited.)

W. Gangbo and A. Swiech. Optimal maps for the multidimensional Monge-Kantorovich problem. *Communications on Pure and Applied Mathematics*, 51(1):23–45, 1998. (Not cited.)

D. Ge, H. Wang, Z. Xiong, and Y. Ye. Interior-point methods strike back: solving the Wasserstein barycenter problem. In *NeurIPS*, pages 6891–6902, 2019. (Not cited.)

A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In *NIPS*, pages 3440–3448, 2016. (Not cited.)

A. Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. *Comptes Redus Hebdomadaires des Séances de l'Académie des Sciences (Paris)*, 254:1192–1194, 1962. (Not cited.)

S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov. Accelerated alternating minimization, accelerated Sinkhorns algorithm and accelerated iterative Bregman projections. *ArXiv Preprint: 1906.03622*, 2019. (Not cited.)

N. Ho, X. L. Nguyen, M. Yurochkin, H. H. Bui, V. Huynh, and D. Phung. Multilevel clustering via Wasserstein means. In *ICML*, pages 1501–1509. JMLR. org, 2017. (Not cited.)

A. Jambulapati, A. Sidford, and K. Tian. A direct $\tilde{O}(1/\epsilon)$ iteration parallel algorithm for optimal transport. *ArXiv Preprint: 1906.00618*, 2019. (Not cited.)

A. Kroshnin, N. Tupitsa, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and C. Uribe. On the complexity of approximating Wasserstein barycenters. In *ICML*, pages 3530–3540, 2019. (Not cited.)

N. Lahn, D. Mulchandani, and S. Raghvendra. A graph theoretic additive approximation of optimal transport. *ArXiv Preprint: 1905.11830*, 2019. (Not cited.)

T. Le, V. Huynh, N. Ho, D. Phung, and M. Yamada. On scalable variant of Wasserstein barycenter. *ArXiv Preprint: 1910.04483*, 2019. (Not cited.)

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. (Not cited.)

T. Lin, N. Ho, M. Cuturi, and M. I. Jordan. On the complexity of approximating multimarginal optimal transport. *ArXiv Preprint: 1910.00152*, 2019a. (Not cited.)

T. Lin, N. Ho, and M. Jordan. On efficient optimal transport: an analysis of greedy and accelerated mirror descent algorithms. In *ICML*, pages 3982–3991, 2019b. (Not cited.)

T. Lin, N. Ho, and M. I. Jordan. On the acceleration of the Sinkhorn and Greenkhorn algorithms for optimal transport. *ArXiv Preprint: 1906.01437*, 2019c. (Not cited.)

E. Munch, K. Turner, P. Bendich, S. Mukherjee, J. Mattingly, J. Harer, et al. Probabilistic fréchet means for time varying persistence diagrams. *Electronic Journal of Statistics*, 9(1): 1173–1204, 2015. (Not cited.)

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. (Not cited.)

Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. (Not cited.)

Y. Nesterov. *Lectures on Convex Optimization*, volume 137. Springer, 2018. (Not cited.)

Y. Nesterov and S. U. Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017. (Not cited.)

G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. (Not cited.)

G. Puccetti, L. Rüschendorf, and S. Vanduffel. On the computation of Wasserstein barycenters. *Available at SSRN 3276147*, 2018. (Not cited.)

K. Quanrud. Approximating optimal transport with linear programs. In *SOSA*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. (Not cited.)

J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011. (Not cited.)

R. T. Rockafellar. *Convex Analysis*, volume 28. Princeton University Press, 1970. (Not cited.)

B. Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019. (Not cited.)

J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015. (Not cited.)

S. Srivastava, C. Li, and D. B. Dunson. Scalable Bayes via barycenter in Wasserstein space. *Journal of Machine Learning Research*, 19(1):312–346, 2018. (Not cited.)

M. Staib, S. Claici, J. M. Solomon, and S. Jegelka. Parallel streaming Wasserstein barycenters. In *NIPS*, pages 2647–2658, 2017. (Not cited.)

A. Trouvé and L. Younes. Local geometry of deformable templates. *SIAM journal on mathematical analysis*, 37(1):17–59, 2005. (Not cited.)

C. A. Uribe, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and A. Nedić. Distributed computation of Wasserstein barycenters over networks. In *CDC*, pages 6544–6549. IEEE, 2018. (Not cited.)

C. Villani. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008. (Not cited.)

L. Yang, J. Li, D. Sun, and K. Toh. A fast globally linearly convergent algorithm for the computation of Wasserstein barycenters. *Arxiv Preprint: 1809.04249*, 2018. (Not cited.)

J. Ye, P. Wu, J. Z. Wang, and J. Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017. (Not cited.)