Improved Complexity Bounds in Wasserstein Barycenter Problem

Darina Dvinskikh
Weierstrass Institute for Applied Analysis and Stochastics,
Institute for Information Transmission Problems RAS,
Moscow Institute of Physics and Technology

Danil Tiapkin
HSE University

Abstract

In this paper, we focus on computational aspects of Wasserstein barycenter problem. We provide two algorithms to compute Wasserstein barycenter of $m$ discrete measures of size $n$ with accuracy $\varepsilon$. The first algorithm, based on mirror prox with some specific norm, meets the complexity of celebrated accelerated iterative Bregman projections (IBP), that is $\tilde{O}(mn^2\sqrt{n}/\varepsilon)$, however, with no limitations unlike (accelerated) IBP, that is numerically unstable when regularization parameter is small. The second algorithm, based on area-convexity and dual extrapolation, improves the previously best-known convergence rates for Wasserstein barycenter problem enjoying $\tilde{O}(mn^2/\varepsilon)$ complexity.

1 Introduction

The theory of optimal transport (OT) provides a natural framework to compare objects that can be modeled as probability measures (images, videos, texts and etc.). Nowadays, OT metrics gains popularity in various fields such as statistics [ESS17, BK12], machine learning [ACB17, SDGP+15], economics and finance [RSF11]. However, the outstanding results of OT come with enormous computations. Indeed, to solve OT problem between two discrete histograms of size $n$, one need to make $\tilde{O}(n^3)$ arithmetic calculations [Tar97, PC18], e.g., by using simplex method or interior point methods. To overcome computational issue, entropic regularization of OT was proposed [Cut13]. It enables the application of Sinkhorn’s algorithm, based on alternating-minimization procedures, with $\tilde{O}(n^2\|C\|_\infty^2/\varepsilon^2)$ convergence rate [DGK18] to approximate the solution of OT with $\varepsilon$-precision. Here $C \in \mathbb{R}^{n \times n}$ is ground cost matrix of transporting a unit of mass between probability measures, and regularization parameter before negative entropy is of order $\varepsilon$. The Sinkhorn’s algorithm can be accelerated to the following rate of convergence $\tilde{O}(n^2\sqrt{n}\|C\|_\infty/\varepsilon)$ [GDG19]. In practice, the accelerated Sinkhorn converges faster than Sinkhorn, and in theory, it has better dependence on $\varepsilon$ but not on $n$. However, entropy-regularized based approaches are numerically unstable when the regularizer parameter $\gamma$ before negative entropy is small (this also means that precision $\varepsilon$ is high as $\gamma$ must be selected proportional to $\varepsilon$ [PC18, KTD+19]). The recent work [JST19] provides optimal method for OT, based on dual averaging [Nes07] and area-convexity [She17], with convergence rate $\tilde{O}(n^2\|C\|_\infty/\varepsilon)$. This method solves OT problem without additional penalization and, moreover, it eliminates the term $\sqrt{n}$ in the bound for accelerated Sinkhorn’s algorithm. The rate $\tilde{O}(n^2\|C\|_\infty/\varepsilon)$ was also obtained in a number of the following works [BJKS18, AZLOW17, CMTV17, LHC+20].

The OT metric finds natural application for defining the mean representative of a set of similar objects. The minimizer of the sum of squared OT distances to all objects in the set is known as Wasserstein barycenter (WB). Regularizing each OT distance in the sum by negative entropy leads to presenting WB problem as Kullback–Leibler projection that can be performed by iterative Bregman projections (IBP) algorithm [BCC+15]. The IBP is the extension of Sinkhorn’s algorithm for $m$ measures, and hence, its complexity is $m$ times more than Sinkhorn complexity, namely $\tilde{O}\left(\min\{n^2\|C\|_\infty^2/\varepsilon^2\}\right)$ [KTD+19]. The analog of accelerated Sinkhorn for WB problem with $m$ measures is accelerated IBP algorithm with complexity $\tilde{O}\left(\min\{mn^2\sqrt{n}\|C\|_\infty/\varepsilon\}\right)$ [GDG19], that is also $m$ times more than accelerated Sinkhorn complexity. Another fast version of the IBP algorithm was recently proposed in [LHC+20], named FastIBP with complexity $\tilde{O}\left(\min\{mn^2\sqrt{n}\|C\|_\infty^2/\varepsilon^{4/3}\}\right)$.

The goal of this paper is providing an algorithm to solve WB problem beating the complexity of existing algo-
1.1 Contribution

Our first contribution is to obtain an algorithm which does not suffer from small value of regularization parameter and at the same time has complexity not worse than celebrated (accelerated) IBP. Thus, we provide the algorithm running with $\tilde{O}(mn^2/\varepsilon)$ convergence rate and which incorporates mirror prox with specific prox-function.

The second contribution is proposing an algorithm that has better complexity in comparison with (accelerated) IBP. Motivated by the work \cite{JST19} proposing optimal way of solving OT problem with better complexity bounds in comparison with (accelerated) Sinkhorn, we develop optimal algorithm for WB problem of $O(mn^2/\varepsilon)$ complexity. Our approach is based on rewriting the WB problem as saddle-point problem, regularizing the problem by specific regularizer which is area-convex and further application of dual extrapolation method.

We notice that the convergence rate obtained by the first algorithm described in this paper is worse than the complexity of our second algorithm, however, in some sense, first algorithm can be seen as an simplified version of the second algorithm and hence, first approach simplifies the understanding of the second approach.

In Table 1 we illustrate our contribution by comparing our algorithms to the most popular algorithms for WB problem.

| Paper        | Approach                  | Complexity                |
|--------------|---------------------------|---------------------------|
| TTD+19       | IBP                       | $\tilde{O}\left(\frac{m^2\|C\|_\infty^2}{\varepsilon}\right)$ |
| CDG+19       | Accelerated IBP           | $\tilde{O}\left(\frac{m^2\|C\|_\infty\sqrt{n}}{\varepsilon}\right)$ |
| LHC+20       | FastIBP                   | $\tilde{O}\left(\frac{m^2\|\nabla C\|_\infty^{4/3}}{\varepsilon}\right)$ |
| This work    | Mirror prox               | $\tilde{O}\left(\frac{m^2\|C\|_\infty}{\varepsilon}\right)$ |
| This work    | Dual extrapolation         | $\tilde{O}\left(\frac{m^2\|C\|_\infty}{\varepsilon}\right)$ |

2 Problem Statement

In this section, we recall the optimal transport (OT) problem, Wasserstein barycenter (WB) problem, and reformulate them as saddle-point problems.

Given two histograms $p,q \in \Delta_n$ and ground cost $C \in \mathbb{R}^{n \times n}$, the OT problem is formulated as follows

$$W(p, q) = \min_{X \in \mathcal{U}(p, q)} \langle C, X \rangle,$$

where $X$ is a transport plan from transport polytope $\mathcal{U} = \{ X \in \mathbb{R}^{n \times n}, X1 = p, X^\top 1 = q \}$. Let $d$ be vectorized cost matrix $C$, $b = \begin{pmatrix} p \\ q \end{pmatrix}$ and $A = \{0, 1\}^{2n \times n^2}$ be the incidence matrix. As $\sum_{i,j=1}^n X_{ij} = 1$, we following by the paper \cite{JST19}, rewrite

$$\min_{x \in \Delta_n, y \in \{-1,1\}^{2n}} \{d^\top x + 2\|d\|_\infty( y^\top Ax - b^\top y)\}.$$ 

Given histograms $q_1, q_2, \ldots, q_m \in \Delta_n$, the WB of these measures is the solution of the following problem

$$p^* = \arg \min_{p \in \Delta_n} \frac{1}{m} \sum_{i=1}^m W(p, q_i).$$

Then introducing $b_i = \begin{pmatrix} p \\ q_i \end{pmatrix}$, we rewrite

$$\min_{p \in \Delta_n} \frac{1}{m} \sum_{i=1}^m \min_{x_i \in \Delta_n, y_i \in \{-1,1\}^{2n}} \max_{y \in \mathcal{Y}} \left\{d_i^\top x_i + 2\|d\|_\infty(y^\top Ax_i - b_i^\top y_i)\right\}.$$ 

Next, we define spaces $\mathcal{X} = \prod_m \Delta_n \times \Delta_n$ and $\mathcal{Y} = \prod_m \{-1,1\}^{2mn}$, where $\prod_m \Delta_n$ is the short form of $\Delta_n \times \ldots \times \Delta_n$. Then we rewrite problem

$$\min_{x \in \Delta_n, y \in \{-1,1\}^{2n}} \{d^\top x + 2\|d\|_\infty( y^\top Ax - b^\top y)\}.$$

Paper Organisation The structure of the paper is the following. In Sect. 2 we reformulate WB problem as a saddle-point problem. Sections 3 and 4 present our first and second algorithms for solving WB problem respectively.

Notation Let $\Delta_n = \{a \in \mathbb{R}_+^n \mid \sum_{i=1}^n a_i = 1\}$ be the probability simplex. We use bold symbol for column vector $x = [x_1, \ldots, x_n]^\top \in \mathbb{R}^mn$, where $x_1, \ldots, x_n \in \mathbb{R}^n$. Then we refer to the $i$-th component of vector $x$ as $x_i \in \mathbb{R}^n$ and to the $j$-th component of vector $x_i$ as $[x_i]_j$. When used on vectors, functions such as log or exp are always applied element-wise. For two vectors $x,y$ of the same size, $x/y$ and $x \circ y$ stand for the element-wise product and element-wise division respectively. For some norm $\| \cdot \|$, we define the dual norm $\| \cdot \|_*$ in a usual way $\|s\|_* = \max_{x \in \mathcal{X}} \{ \langle x, s \rangle : \|x\| \leq 1 \}$. We denote by $I_n$ the identity matrix and by $0_{n \times n}$ zeros matrix. For prox-function $d(x)$, we define corresponding Bregman divergence $B(x,y) = d(x) - d(y) - \langle \nabla d(y), x - y \rangle$. 

2 Problem Statement

In this section, we recall the optimal transport (OT) problem, Wasserstein barycenter (WB) problem, and reformulate them as saddle-point problems.

Given two histograms $p,q \in \Delta_n$ and ground cost $C \in \mathbb{R}^{n \times n}$, the OT problem is formulated as follows

$$W(p, q) = \min_{X \in \mathcal{U}(p, q)} \langle C, X \rangle,$$

where $X$ is a transport plan from transport polytope $\mathcal{U} = \{ X \in \mathbb{R}^{n \times n}, X1 = p, X^\top 1 = q \}$. Let $d$ be vectorized cost matrix $C$, $b = \begin{pmatrix} p \\ q \end{pmatrix}$ and $A = \{0, 1\}^{2n \times n^2}$ be the incidence matrix. As $\sum_{i,j=1}^n X_{ij} = 1$, we following by the paper \cite{JST19}, rewrite

$$\min_{x \in \Delta_n, y \in \{-1,1\}^{2n}} \{d^\top x + 2\|d\|_\infty( y^\top Ax - b^\top y)\}.$$ 

Given histograms $q_1, q_2, \ldots, q_m \in \Delta_n$, the WB of these measures is the solution of the following problem

$$p^* = \arg \min_{p \in \Delta_n} \frac{1}{m} \sum_{i=1}^m W(p, q_i).$$

Then introducing $b_i = \begin{pmatrix} p \\ q_i \end{pmatrix}$, we rewrite

$$\min_{p \in \Delta_n} \frac{1}{m} \sum_{i=1}^m \min_{x_i \in \Delta_n, y_i \in \{-1,1\}^{2n}} \max_{y \in \mathcal{Y}} \left\{d_i^\top x_i + 2\|d\|_\infty(y^\top Ax_i - b_i^\top y_i)\right\}.$$ 

Next, we define spaces $\mathcal{X} = \prod_m \Delta_n \times \Delta_n$ and $\mathcal{Y} = \prod_m \{-1,1\}^{2mn}$, where $\prod_m \Delta_n$ is the short form of $\Delta_n \times \ldots \times \Delta_n$. Then we rewrite problem
for column vectors \( x = (x_1^T, \ldots, x_m^T, p)^T \in \mathcal{X} \) and \( y = (y_1, \ldots, y_m)^T \in \mathcal{Y} \)

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} F(x, y) = \frac{1}{m} \left\{ d^T x + 2\|d\|_\infty \left(y^T Ax - c^T y\right) \right\}, \tag{4}
\]

where \( d = (d^T, \ldots, d^T, 0_n^T) \), \( c = (0_n^T, q_1^T, \ldots, 0_n^T, q_m^T) \) and \( A \in \{-1, 0, 1\}^{2mn \times (mn+n)} \) is almost block-diagonal matrix

\[
A = \begin{pmatrix}
A & 0_{2n \times n^2} & \cdots & 0_{2n \times n^2} & -I_n & 0_{n \times n} \\
0_{2n \times n^2} & A & \cdots & 0_{2n \times n^2} & -I_n & 0_{n \times n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0_{2n \times n^2} & 0_{2n \times n^2} & \cdots & A & -I_n & 0_{n \times n}
\end{pmatrix}
\]

As objective \( F(x, y) \) in (4) is convex in \( x \) and concave in \( y \), problem (4) is saddle-point problem.

### 3 Mirror Prox for Wasserstein Barycenter

In this section, we present our first algorithm which does not improve the complexity of state-of-the-art methods for WB problem but has no limitations that other Sinkhorn based algorithms have. Moreover, this method contributes to a better understanding of our second approach. To present the results, we define the following setups using throughout this section.

#### 3.1 Setups

We endow space \( \mathcal{Y} \triangleq [-1,1]^{2nm} \) with the standard Euclidean setup: the Euclidean norm \( \| \cdot \|_2 \), projection \( d_y^2(y) = \frac{1}{2} \|y\|_2^2 \) and the corresponding Bregman divergence \( B_y(y, \hat{y}) = \frac{1}{2} \|y - \hat{y}\|_2^2 \).

For space \( \mathcal{X} \triangleq \prod_{i=1}^n \Delta_{n,i} \times \Delta_{n,0} \), we choose the following specific norm \( \|x\|_X = \sqrt{\sum_{i=1}^n \|x_i\|^2 + m\|p\|^2} \) for \( x = (x_1, \ldots, x_m, p)^T \), where \( \| \cdot \|_1 \) is the Manhattan norm (for \( a \in \mathbb{R}^n, \|a\|_1 = \sum_{i=1}^n |a_i| \)). We endow \( \mathcal{X} \) with projection \( dx(x) = \sum_{i=1}^m (x_i, \ln x_i) + m(p, \ln p) \) and corresponding Bregman divergence

\[
B_x(x, \bar{x}) = \sum_{i=1}^m (x_i, \ln x_i/\bar{x}_i) - \sum_{i=1}^m 1^T(x_i - \bar{x}_i) \\
+ m(p, \ln(p/\bar{p})) - m1^T(p - \bar{p}).
\]

We also define \( R_{\bar{x}}^2 = \sup_{x \in \mathcal{X}} dx(x) - \min_{x \in \mathcal{X}} dx(x) \) and \( R_{\bar{y}}^2 = \sup_{y \in \mathcal{Y}} dy(y) - \min_{y \in \mathcal{Y}} dy(y) \).

#### Definition 3.1

\( f(x, y) \) is \((L_{xx}, L_{xy}, L_{yx}, \bar{L}_{yy})\)-smooth if for any \( x, x' \in \mathcal{X} \) and \( y, y' \in \mathcal{Y} \),

\[
\|\nabla_x f(x, y) - \nabla_x f(x', y)\|_{\mathcal{X}} \leq L_{xx}\|x - x'\|_{\mathcal{X}},
\]

\[
\|\nabla_x f(x, y) - \nabla_x f(x, y')\|_{\mathcal{X}} \leq L_{xy}\|y - y'\|_{\mathcal{Y}},
\]

\[
\|\nabla_y f(x, y) - \nabla_y f(x, y')\|_{\mathcal{Y}} \leq L_{yx}\|y - y'\|_{\mathcal{Y}},
\]

\[
\|\nabla_y f(x, y) - \nabla_y f(x', y)\|_{\mathcal{Y}} \leq \bar{L}_{yy}\|x - x'\|_{\mathcal{X}}.
\]

#### 3.2 Implementation and Complexity Bound

In this section, as problem (1) is saddle-point problem, we evaluate the quality of the algorithm, that outputs a pair of solutions \((\tilde{x}, \tilde{y}) \in (\mathcal{X}, \mathcal{Y})\), through the so-called duality gap

\[
\max_{y \in \mathcal{Y}} F(\bar{x}, y) - \min_{x \in \mathcal{X}} F(x, \bar{y}) \leq \varepsilon, \tag{5}
\]

Our first algorithm is based on the mirror prox (MP) for saddle-point problems on space \( Z \triangleq \mathcal{X} \times \mathcal{Y} \) with prox-function \( d_Z(z) = a_1 dx(x) + a_2 dy(y) \) and corresponding Bregman divergence \( B_Z(z, \bar{z}) = a_1 B_x(x, \bar{x}) + a_2 B_y(y, \bar{y}) \), where \( a_1 = \frac{1}{N} \), \( a_2 = \frac{1}{N} \).

\[
\left( u_{k+1}^{X}, v_{k+1}^{Y} \right) = \arg \min_{z \in Z} \left\{ \eta G(z) \right\} \tag{6}
\]

\[
G(z, y) = \left( \begin{array}{c}
\nabla_x F(x, y) \\
-\nabla_y F(x, y)
\end{array} \right) = \frac{1}{m} \left( \begin{array}{c}
(d + 2\|d\|_\infty A^T y) \\
2\|d\|_\infty (c - Ax)
\end{array} \right).
\]

If \( F(x, y) \) is \((L_{xx}, L_{xy}, L_{yx}, L_{yy})\)-smooth, then to satisfy (5) with \( \tilde{x} = \frac{1}{N} \sum_{k=1}^N u_k^{X}, \tilde{y} = \frac{1}{N} \sum_{k=1}^N v_k^{Y}, \) we need to make

\[
N = 4L_{xy} R_x R_y / \varepsilon
\]

iterations of MP with

\[
\eta = 1/(2 \max \{L_{xx} R_x^2, L_{xy} R_x R_y, L_{yx} R_y R_x, L_{yy} R_y^2 \}).
\]

#### Lemma 3.2

**Objective \( F(x, y) \) in (4) is \((L_{xx}, L_{xy}, L_{yx}, L_{yy})\)-smooth with \( L_{xx} = L_{yy} = 0 \) and \( L_{yx} = L_{xy} = 2\|d\|_\infty / m \).**

#### Proof

Let us consider bilinear function \( f(x, y) \triangleq y^T Ax \). By Definition 3.1 it can be shown that \( f(x, y) \) is \((0, 1, 0, 0)\)-smooth. As linear terms does not contribute to the smoothness property, objective \( F(x, y) \) in (4) is \((0, 2\|d\|_\infty / m, 2\|d\|_\infty / m, 0, 0)\)-smooth.
Input: measures \(q_1, \ldots, q_m\), linearized cost matrix \(d\), incidence matrix \(A\), step \(\eta\), \(p^0 = \frac{1}{n}1_n\), \(x_1^0 = \ldots = x_m^0 = \frac{1}{n}1_n\), \(y_1^0 = \ldots = y_m^0 = 0_m\).

1. \(\alpha = 2|d||\infty|n\eta\), \(\beta = 6|d||\infty|\eta\ln n/m\), \(\gamma = 3\eta m\ln n\).
2. for \(k = 0, 1, 2, \ldots, N - 1\) do
3. for \(i = 1, 2, \ldots, m\) do
4. \(v_i^{k+1} = y_i^k + \alpha (Ax_i^k - (p^k)_i)\),
   Project \(v_i^{k+1}\) onto \([-1, 1]^{2n}\)
5. \(u_{i+1}^k = \sum_{l=1}^{n^2} [x_l^k, i] \exp \left\{ -\gamma (d_1 + 2 \|d\|_\infty A^T y_i^k)\right\}\)
6. end for
7. \(s_{k+1} = p^0 \sum_{i=1}^m \exp \{\beta \sum_{i=1}^m |v_i^{k+1}|_{1-n} + \alpha \}\)
8. for \(i = 1, 2, \ldots, m\) do
9. \(u_{i+1}^{k+1} = u_{i+1}^k + \alpha (Ax_i^k - (s_{k+1}^{k+1}))\)
10. Project \(u_{i+1}^{k+1}\) onto \([-1, 1]^{2n}\)
11. end for
12. \(p^{k+1} = p^0 \sum_{i=1}^m \sum_{i=1}^m \sum_{i=1}^m [x_l^k, i] \exp \{\beta \sum_{i=1}^m |v_i^{k+1}|_{1-n} + \alpha \}\)
13. end for

Output: \(\tilde{u} = \begin{pmatrix} \tilde{u}_1 \\ \vdots \\ \tilde{u}_m \\ \tilde{s} \end{pmatrix}, \tilde{v} = \begin{pmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_m \\ \tilde{m} \end{pmatrix}\), where \(\tilde{u}_i = \frac{1}{N} \sum_{k=0}^{N-1} u_i^k\), \(\tilde{v}_i = \frac{1}{N} \sum_{k=0}^{N-1} v_i^k\), \(i = 1, \ldots, m\), \(\tilde{s} = \frac{1}{N} \sum_{k=0}^{N-1} s_k^k\).

The total complexity of Algorithm 1 is \(O(\frac{mn^2}{\varepsilon} \sqrt{n\ln n \|d\|_\infty})\).

**Proof.** By Lemma 3.2 \(F(x, y)\) is \((0, 2|d||\infty|/m, 2|d||\infty|/m, 0, 0, 0, 0, 0, 0\))-smooth. Then the bound on duality gap follows from direct substitution of the expression for \(R_X, R_Y\) and \(L_{XX}, L_{XY}, L_{YX}, L_{YY}\) in (6) and (7).

The complexity of one iteration of Alg. 1 is \(O(\frac{mn^2}{\varepsilon})\) as the number of non-zero elements in matrix \(A\) is \(2n^2\) and \(m\) is the number of vector-components in \(y\) and \(x\). Multiplying this by the number of iterations \(N\), we get the last statement of the theorem.

Using the definition of \(d\) we may reformulate complexity results of Theorem 3.3 in terms of ground cost matrix \(C\) as \(O(\frac{mn^2}{\varepsilon} \sqrt{n\ln n \|C\|_\infty})\).

## 4 Dual Extrapolation with area-convexity for Wasserstein Barycenter

In this section, we present our second algorithm that improves the complexity bounds for WB problem.

### 4.1 General framework

Recall \(Z \triangleq \mathcal{X} \times \mathcal{Y}\) as a space of pairs \((x, y), x \in \mathcal{X}, y \in \mathcal{Y}\). Using this space, we can redefine our functions of pairs as a functions of a single argument from \(Z\), such as a gradient operator.

Now we use the main framework proposed in [She17] and developed in [IST19]. The key idea is to use a wider family of regularizers then strongly convex ones in Dual Extrapolation algorithm for bilinear saddle-point problems. This family is called area-convex regularizers and can be defined in the following way:

**Definition 4.1.** Regularizer \(r\) is called \(\kappa\)-area convex with respect to \(G\) for any points \(a, b, c \in Z\)

\[\kappa \left( r(a) + r(b) + r(c) - 3r \left( \frac{a + b + c}{3} \right) \right) \geq (G(a) - G(b), b - c)\]

We will use only differentiable regularizer and it gives us a possibility to define a proximal operator using \(r(z)\) as a prox-function and use a Dual Extrapolation algorithm. Define \(\tilde{z} = \arg\min_{z \in Z} r(z)\).

In this condition, we have the following converge guarantees in terms of a number of iterations for any gradient operator \(G\) for bilinear saddle-point problems:
Lemma 4.2 (Corollary 1 in [JST19]). Suppose \( r \) is \( \kappa \)-area convex with respect to \( G \). Further, suppose for some \( u, \Theta \geq r(u) - r(\tilde{z}) \), then the output \( w \) of Dual Extrapolation algorithm with the proximal steps implemented with \( \varepsilon' \) additive error satisfies

\[
\langle G(w), w - u \rangle \leq \frac{2\kappa \Theta}{N} + \varepsilon'
\]

If we choose \( \Theta = \sup_{z \in \mathcal{Z}} r(z) - r(\tilde{z}) \), we obtain the converge guarantees in terms of a duality gap \( 5 \).

Algorithm 2 Dual Extrapolation with area-convex \( r \):

Input: area-convexity coefficient \( \kappa \), regularizer \( r \), gradient operator \( G \), number of iterations \( N \).

1. \( s^0 = 0, \tilde{z} = \arg \min_{z \in \mathcal{Z}} r(z) \)
2. for \( k = 0, 1, 2, \ldots, N - 1 \) do
3. \( z^k = \text{prox}_{s^k}(s^k) \)
4. \( w^k = \text{prox}_{s^k}(s^k + \frac{1}{\kappa} G(z^k)) \)
5. \( s^{k+1} = s^k + \frac{1}{\kappa} G(w^k) \)
6. end for

Output: \( \bar{w} = \frac{1}{N} \sum_{k=0}^{N-1} w^k \)

4.2 Complexity bounds

To define an area-convex regularizer, we should examine the structure of the our particular problem. Firstly, we separate matrix \( A \) into two parts \( A = (A|\mathcal{E}) \):

\[
\hat{A} = \begin{pmatrix}
A & 0_{2n \times n^2} & \cdots & 0_{2n \times n^2} \\
0_{2n \times n^2} & A & \cdots & 0_{2n \times n^2} \\
\vdots & \vdots & \ddots & \vdots \\
0_{2n \times n^2} & 0_{2n \times n^2} & \cdots & A
\end{pmatrix}
\]

where \( A \in \{0,1\}^{2n \times n^2} \) and \( \hat{A} \) has \( m \) blocks, and

\[
\mathcal{E}^\top = \begin{pmatrix}
(-I_n, 0_{n \times n}) & (-I_n, 0_{n \times n}) & \cdots & (-I_n, 0_{n \times n}) \\
-B^k_{\tilde{z}} & -B^k_{\tilde{z}} & \cdots & -B^k_{\tilde{z}}
\end{pmatrix}.
\]

Then we can define our regularizer as a generalization of the regularizer from [JST19]:

\[
r(x, y) = \frac{2\|d\|_\infty}{m} \left( 10 \sum_{i=1}^{m} \langle x_i, \log x_i \rangle + 5m(p, \log p) \right. \\
\left. + \hat{x}^\top \hat{A}^\top (y)^2 - p^\top \mathcal{E}^\top (y)^2 \right),
\]

where \( \log x \) and \( (x)^2 \) are entry-wise, and \( \hat{x} = (x_1, \ldots, x_m)^\top \).

For this regularizer area-convexity can be proven:

Theorem 4.3. \( r \) is \( 3 \)-area-convex with respect to the gradient operator \( G \).

To compute the range of the regularizer, we can rewrite it in the following homogeneous manner:

\[
r(x, y) = \frac{2\|d\|_\infty}{m} \left( \sum_{i=1}^{m} \left[ 10\langle x_i, \log x_i \rangle + (Ax_i, (y_i)^2) \right] \\
+ \sum_{i=1}^{m} \left[ 5(p, \log p) + (B\varepsilon p, (y_i)^2) \right] \right).
\]

Hence, using properties of sets \( \mathcal{X} \) and \( \mathcal{Y} \), we obtain the following bound on range of regularizer:

\[
\Theta = \sup_{z \in \mathcal{Z}} r(z) - \inf_{z \in \mathcal{Z}} r(z) = 40 \log n \|d\|_\infty + 6\|d\|_\infty.
\]

The only question is how to compute a proximal step effective. Formally, we are solving the following type of problem:

\[
H(x, y) = \langle v, x \rangle + \langle u, y \rangle + r(x, y).
\]

It can be done using a simple alternating minimization scheme as in the case of [JST19].

Theorem 4.4. Alternating minimization (AM) scheme for the proximal steps required by Dual Extrapolation algorithm with \( \varepsilon/2 \) accuracy with \( \kappa = 3 \) can obtain \( \varepsilon/2 \) additive error in

\[
24\log \left( \frac{88\|d\|_\infty}{\varepsilon^2} + \frac{4}{\varepsilon} \right) \Theta + \frac{36\|d\|_\infty}{\varepsilon} \end{equation}
\]

iterations and in \( O(mn^2 \log \gamma) \) time, where \( \gamma = \varepsilon^{-1} \|d\|_\infty \log n \).

The complete algorithm is presented in Algorithm 3. We will refer to this algorithm in the further algorithms as AM(\( M, v, u \)).

The division in the algorithm between two vectors is entrywise.

The proof of correctness of this procedure can be found in supplementary material. It consists of three main parts: the required details from the proof of [JST19] to obtain a linear convergence, bound on the time for each substep, and the bound on the initial error for our setup of proximal steps.

Overall, for the particular WB problem (4), we obtain the required complexity bound by combination of Lemma 4.2, Theorem 4.3 and Theorem 4.4. The final algorithm is presented as Algorithm 3.

Theorem 4.5. Dual Extrapolation algorithm after

\[
N = \frac{12\Theta}{\varepsilon} = \frac{480 \log n \|d\|_\infty + 72\|d\|_\infty}{\varepsilon}
\]

iterations outputs a pair \( (\bar{w}, \bar{w}) \in (\mathcal{X}, \mathcal{Y}) \) such that the duality gap \( 7 \) becomes less then \( \varepsilon \), and it can be done in \( O(mn^2 \|d\|_\infty \varepsilon^{-1}) \) time.
Proof. The required number of iterations to obtain $\epsilon$ precision follows from the choice of 3-area-convex regularizer $r$ (follows from Lemma 4.3 and Lemma 4.2). For each step we need to do two proximal steps, that can be done in $O(mn^2 \log \gamma)$ time by Theorem 4.4. As a result, we have an algorithm with $O(mn^2 \|d\|_{\infty} \epsilon^{-1} \log n \log \gamma) = O(mn^2 \|d\|_{\infty} \epsilon^{-1})$ time complexity.

In terms of the initial cost matrix $C$, we obtain $O(mn^2 \|C\|_{\infty} \epsilon^{-1})$ complexity.

5 Numerical experiments

In this section, we illustrate the work of the proposed algorithms in practice on the MNIST dataset and Gaussian distributions.

**MNIST.** In the paper, we mentioned that when high-precision $\epsilon$ of calculating WB is desired, iterative Bregman projections (IBP) algorithm with regularizing parameter $\gamma$ is numerically unstable (as $\gamma$ must be selected proportional to $\epsilon$) unlike Algorithm 1 from Sec. 3 and Algorithm 4 from Sec. 4. Now we support this statement by computing the barycenter of handwritten digits 5’s from the MNIST dataset. Figure 1 illustrates the results obtained by the proposed algorithms in comparison with iterative Bregman projections (IBP) with small values of regularizing parameter ($\gamma = 1e^{-3}$, $1e^{-5}$).

Figure 1: Comparison of the barycenters computed by Algorithm 1 (Mirror Prox for WB), Algorithm 4 (Dual extrapolation for WB) and IBP with different values of regularizing parameter

**Gaussian measures.** To compare the convergence of the proposed algorithms we randomly generated 10 Gaussian measures with mean from $[-5, 5]$, variance from $[0.5, 1]$, and studied the convergence of calculated barycenters to the theoretical true barycenter [DD20]. Figure 2 demonstrates this convergence. Despite the fact that Algorithm 4 has better complexity bound, Algorithm 3 has better convergence in practice.
slope ration $-1$ for the convergence of Algorithm 1 in log-scale perfectly fits theoretical dependence of working time (iteration number $N$) on the desired accuracy $\varepsilon$ ($N \sim \varepsilon^{-1}$ from Theorem 3.3). For Algorithm 4 we do not have so obvious agreement with the theory but this is due to the necessity of solving practically computationally costly subproblems.

Figure 2: Convergence of Algorithm 1 (Mirror Prox for WB) and Algorithm 4 (Dual extrapolation for WB)

Next, we illustrate the approximation of the true Gaussian barycenter by Algorithm 1 and IBP algorithm with regularizing parameter $\gamma$. The smaller $\gamma$, the closer regularized IBP barycenter is to the true barycenter. Due to numerical instability of IBP with small regularized parameter, the smallest possible value of $\gamma$ for this problem is $\gamma = 0.0005$. Figure 3 demonstrates better approximation of the true Gaussian barycenter by Algorithm 1 than IBP algorithm.

We also tried to accelerate Algorithm 1 by choosing the constants by adaptive way for Mirror Prox algorithm [BL19, SGD+18], however, this did not give significant improvement in convergence.

6 Conclusion

In this work, we provided two algorithms that have theoretical and practical interests. The main theoretical value is obtaining $\sqrt{n}$ faster algorithm for approximating Wasserstein barycenter of discrete measures with support $n$. Making the same arguments as the authors of [BJKS18] proving the unimprovable complexity bound for OT, we expect that our obtained complexity bound for WB is unimprovable in that sense that allows to solve long-standing open problem. The main practical value is the opportunity to calculate Wasserstein barycenter with high desired precision that is not possible by entropy-regularized based approaches. However in practice, this problem is very time consuming.

Acknowledgements

The research of Sec. 3 is supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) No. 075-00337-20-03, project No. 0714-2020-0005. The research of Sec. 4 is funded by RFBR 19-31-51001 Sirius. The work of Sec. 5 is supported by the Russian Science Foundation (project 18-71-10108). The work of D. Tiapkin was fulfilled in Sirius, Sochi [https://ssopt.org] (August, 2020), the work was initiated by A. Gasnikov.

References

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. arXiv:1701.07875, 2017.

[AZLOW17] Z. Allen-Zhu, Y. Li, R. Oliveira, and A. Wigderson. Much faster algorithms
Improved Complexity Bounds in Wasserstein Barycenter Problem

for matrix scaling. In 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pages 890–901, Oct 2017. [EJS17]

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

[BCC+15]

Jose Blanchet, Arun Jambulapati, Carson Kent, and Aaron Sidford. Towards optimal running times for optimal transport. arXiv preprint arXiv:1810.07717, 2018.

[BJKS18]

Francis Bach and Kfir Y Levy. A universal algorithm for variational inequalities adaptive to smoothness and noise. arXiv preprint arXiv:1902.01637, 2019.

[BK+12]

M. B. Cohen, A. Madry, D. Tsipras, and A. Vladu. Matrix scaling and balancing via box constrained newton’s method and interior point methods. In 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pages 902–913, Oct 2017. [CMTV17]

[Cut13]

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In Advances in Neural Information Processing Systems, pages 2292–2300, 2013.

[DD20]

Julie Delon and Agnès Desolneux. A wasserstein-type distance in the space of gaussian mixture models. SIAM Journal on Imaging Sciences, 13(2):936–970, 2020.

[DGK18]

Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn’s algorithm. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80, pages 1367–1376, 2018. arXiv:1802.04367.

[DGK19]

Sergey Guminov, Pavel Dvurechensky, and Alexander Gasnikov. Accelerated alternating minimization. arXiv preprint arXiv:1906.03622, 2019.

[ESS17]

Arun Jambulapati, Aaron Sidford, and Kevin Tian. A direct tilde \(\{O\}(1/\epsilon)\) iteration parallel algorithm for optimal transport. In Advances in Neural Information Processing Systems, pages 11359–11370, 2019.

[JST19]

Alexey Kroshnin, Nazarii Tupitsa, Darina Dvinskikh, Pavel Dvurechensky, Alexander Gasnikov, and Cesar Uribe. On the complexity of approximating Wasserstein barycenters. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97, pages 3530–3540, 2019. arXiv:1901.08686.

[KTD+19]

Tianyi Lin, Nhat Ho, Xi Chen, Marco Cuturi, and Michael I Jordan. Fixed-support wasserstein barycenters: Computational hardness and fast algorithm. 2020.

[LHC20]

Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. Mathematical Programming, 109(2-3):319–344, 2007.

[Nes07]

Gabriel Peyré and Marco Cuturi. Computational optimal transport. arXiv:1803.00567, 2018.

[PC18]

Svetlozar T Rachev, Stoyan V Stoyanov, and Frank J Fabozzi. A probability metrics approach to financial risk measures. John Wiley & Sons, 2011.
Fedor Stonyakin, Alexander Gasnikov, Pavel Dvurechensky, Mohammad Alkousa, and Alexander Titov. Generalized mirror prox for monotone variational inequalities: Universality and inexact oracle. *arXiv preprint arXiv:1806.05140*, 2018.

Jonah Sherman. Area-convexity, $l_\infty$ regularization, and undirected multicommodity flow. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 452–460, 2017.

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

7.1 Proof of Theorem 4.3

Proof of Theorem 4.3. Firstly, define some notation connected to block-diagonal matrices. Assume that \( D \) is a block diagonal matrix of size \( ak \times bk \)

\[
D = \begin{pmatrix}
B_1 & 0 & \cdots & 0 \\
0 & B_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & B_k
\end{pmatrix},
\]

where matrices \( B_i \) of size \( a \times b \). We refer to \( i \)-th block of \( D \) as \( D_{(i)} = B_i \). Also we define \( D_{[i]} \) as a matrix \( D \) with all blocks zeroes except the \( i \)-th one. Equivalent, we can write \( D_{[i]} = \delta^{(k)}_{ii} \otimes D_{(i)} \), where \( \delta^{(k)}_{ij} \) is a matrix of size \( k \times k \) with 1 on the position \( i, j \) position and 0 in any other, and \( \otimes \) is a Kronecker product of matrices.

We will use a second-order criteria proposed in [JST19]. We will show that

\[
\begin{pmatrix}
\nabla^2 r(z) & -J \\
J & \nabla^2 r(z)
\end{pmatrix} \succeq 0,
\]

where

\[
J = \frac{2\|d\|_\infty}{m} \begin{pmatrix}
0 & A^T \\
-A & 0
\end{pmatrix} = \frac{2\|d\|_\infty}{n} \begin{pmatrix}
0 & 0 & \hat{A}^T \\
0 & 0 & \hat{\mathcal{E}}^T \\
-\hat{A} & -\hat{\mathcal{E}} & 0
\end{pmatrix}
\]

is a Jacobian matrix for \( F(x, y) \).

A good idea to remove a positive multiplicative term \( 2\|d\|_\infty m^{-1} \) to simplify the statement. Define \( r'(z) = 1/(2\|d\|_\infty m^{-1})r(z) \) and \( J' = 1/(2\|d\|_\infty m^{-1})J \). Hence we only should show that

\[
P = \begin{pmatrix}
\nabla^2 r'(z) & -J' \\
J' & \nabla^2 r'(z)
\end{pmatrix} = \frac{m}{2\|d\|_\infty} \begin{pmatrix}
\nabla^2 r(z) & -J \\
J & \nabla^2 r(z)
\end{pmatrix} \succeq 0
\]

Then we can rewrite \( r' \) in the following manner:

\[
r'(x, y) = \sum_{i=1}^{m} \left[ 10(x_i, \log x_i) + \langle Ax_i, (y_i)^2 \rangle \right] + 5m(p, \log p) - p^T \mathcal{E}^T (y^2) = \sum_{i=1}^{m} \left[ 10(x_i, \log x_i) + \langle Ax_i, (y_i)^2 \rangle \right] + \sum_{i=1}^{m} \left[ 5(p, \log p) + \langle B\mathcal{E} p, (y_i)^2 \rangle \right]
\]

In this case, we can easily calculate the hessian of \( r' \), divide it into blocks:

\[
\nabla^2 r'(z) = \begin{pmatrix}
\nabla^2_{x,x} r'(z) & \nabla^2_{x,y} r'(z) & \nabla^2_{x,p} r'(z) \\
\nabla^2_{y,x} r'(z) & \nabla^2_{y,y} r'(z) & \nabla^2_{y,p} r'(z) \\
\n\end{pmatrix} = \begin{pmatrix}
10 \text{diag}((\hat{x})^{-1}) & 0_{mn^2 \times n} & 2\hat{A}^T \text{diag}(y) \\
0_{n \times mn^2} & 5m \text{diag}((p)^{-1}) & -2\mathcal{E}^T \text{diag}(y) \\
2 \text{diag}(y)\hat{A} & -2 \text{diag}(y)\mathcal{E} & 2 \text{diag}(\hat{A}\hat{x}) - 2 \text{diag}(\mathcal{E}p)
\end{pmatrix},
\]

where \( \text{diag}(v) \) for a vector \( v \in \mathbb{R}^n \) produces a diagonal matrix with \( v \) on diagonal and \( v^{-1} \) is an entry-wise operation on vector.

Remark that matrices \( \text{diag}((\hat{x})^{-1}) \), \( \hat{A}^T \text{diag}(y) \), \( \text{diag}(\hat{A}\hat{x}) \) have a block-diagonal structure with \( m \) blocks. Define the following matrices:

\[
B_i(z) = \begin{pmatrix}
10 \text{diag}((\hat{x})^{-1})_{[i]} & 0_{mn^2 \times n} & 2(\hat{A}^T \text{diag}(y))_{[i]} \\
0_{n \times mn^2} & 0_{n \times n} & 0_{n \times 2mn} \\
2(\text{diag}(y)\hat{A})_{[i]} & 0_{2mn \times n} & 2 \text{diag}(\hat{A}\hat{x})_{[i]}
\end{pmatrix}
\]
and

\[
R(\mathbf{z}) = \begin{pmatrix}
0_{mn^2 \times mn^2} & 0_{mn^2 \times n} & 0_{mn^2 \times 2mn} \\
0_{n \times mn^2} & 5m \text{ diag}((p)^{-1}) & -2\mathbf{E}^T \text{ diag}(\mathbf{y}) \\
0_{2mn^2 \times mn^2} & -2 \text{ diag}(\mathbf{y})\mathbf{E} & -2\text{ diag}((\mathbf{E}p))
\end{pmatrix}.
\]

Using these matrices, the decomposition of Hessian can be observed: \( \nabla^2 r'(\mathbf{z}) = \sum_{i=1}^{m} B_i(\mathbf{z}) + R(\mathbf{z}) \).

Remark that the matrix \( J' \) has the same block decomposition:

\[
C_i = \begin{pmatrix}
0 & 0 & (\hat{A}^\top)_{[i]} \\
0 & 0 & 0 \\
(\hat{A})_{[i]} & 0 & 0
\end{pmatrix}, \quad S = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & \mathbf{E}^\top \\
0 & -\mathbf{E} & 0
\end{pmatrix}.
\]

And we clearly have \( J' = \sum_{i=1}^{m} C_i + S \). Using these two decompositions, we have the following:

\[
P = \sum_{i=1}^{m} \begin{pmatrix}
B_i(\mathbf{z}) & -C_i \\
C_i & B_i(\mathbf{z})
\end{pmatrix} + \begin{pmatrix}
R(\mathbf{z}) & -S \\
S & R(\mathbf{z})
\end{pmatrix}.
\]

It can be observed that each matrix \( P_i \) is almost a corresponding matrix for the area-convex regularizer for the optimal transportation problem with variables \( x_i, y_i \) in \([JST19]\), except the rows and columns of zeros. Moreover, it was proven that these matrices are positive semi-definite. Hence, only the remaining term is need to be examined.

Firstly, write the action of non-zero corner of \( R(\mathbf{z}) \), called \( \hat{R}(\mathbf{z}) \), as a quadratic form:

\[
Q_{\hat{R}(\mathbf{z})}(u, v) = (u^\top, v^\top)\hat{R}(\mathbf{z}) (u \atop v) = (u^\top, v^\top) \begin{pmatrix} 5m \text{ diag}((p)^{-1}) & -2\mathbf{E}^T \text{ diag}(\mathbf{y}) \\ -2 \text{ diag}(\mathbf{y})\mathbf{E} & -2\text{ diag}((\mathbf{E}p)) \end{pmatrix} (u \atop v).
\]

The we can use the trick induced by the structure of the matrix \( \mathbf{E} \) to compute the quadratic form. The trick is about to rewrite \( m \) in the following way: \( m = \|\mathbf{E}_{i,j}\|_1 = -\sum_{i=1}^{2mn} \mathbf{E}_{ij}, \forall j \in [n] \).

Then, we can calculate the quadratic form:

\[
Q_{\hat{R}(\mathbf{z})}(u, v) = \sum_{i,j} (-\mathbf{E}_{ij}) \begin{pmatrix} 5u_i^2 \overline{p_j} + 4u_i v_i y_i + 2v_i^2 p_j \end{pmatrix}
\]

Secondly, write the action of non-zero corner of \( S \), called \( \hat{S} \), as a bilinear form

\[
B_{\hat{S}}((a, b), (u, v)) = (x^\top, y^\top) \begin{pmatrix} 0 & \mathbf{E}^\top \\ -\mathbf{E} & 0 \end{pmatrix} (u \atop v) = \sum_{i,j} \mathbf{E}_{ij} (a_j v_i - u_j b_i),
\]

and, as a result, we have the complete analytic expression for the quadratic form induced by the remaining term of \( P \):

\[
((a^\top, b^\top), (u^\top, v^\top)) \begin{pmatrix} \hat{R}(\mathbf{z}) & -\hat{S} \\ \hat{S} & \hat{R}(\mathbf{z}) \end{pmatrix} \begin{pmatrix} a \\ b \\ u \\ v \end{pmatrix} = \sum_{i,j} (-\mathbf{E}_{ij}) \begin{pmatrix} 5u_i^2 \overline{p_j} + 4a_j b_i y_i + 2v_i^2 p_j + 2a_j v_i - 2u_j b_i + \frac{5u_i^2}{p_j} + 4u_j v_i y_i + 2v_i^2 p_j \\
(2a_j y_i + b_i p_j)^2 + (2u_j y_i + v_i p_j)^2 \\
(a_j + v_i p_j)^2 + (u_j + b_i p_j)^2 + (1 - (y_i)^2)(a_j^2 + u_j^2) \end{pmatrix} \geq 0
\]

The final inequality follows from the range of \( y_i \in [-1, 1] \) and finishes the proof. \( \square \)
7.2 Proof of Theorem 4.4

To prove this theorem we will use results from [JST19] about their Alternating minimization scheme. First of all, we need to obtain a linear convergence and we can do it by adapting an argument of [JST19] Lemma 6 to our setup.

**Lemma 7.1.** For some $x^{k+1}, y_k$, let $X_k = \{ x \mid x \geq \frac{1}{2} x^{k+1} \}$ where inequality is entrywise, and let $Y_k$ be the entire domain of $y$ (i.e., $Y$). Then for any $x' \in X_k, y', y'' \in Y_k$,

$$\nabla^2 r(x', y') \succeq \frac{1}{12} \nabla^2_{yy} r(x^{k+1}, y'')$$

**Proof.** The only thing that differs in the analysis is a diagonal approximation then does not depends on $y$. Hence, we only need to show that for any $y$:

$$D(x) \preceq \nabla^2 r(x, y) \preceq 6D(x)$$

where $D(x)$ is the diagonal approximation

$$D(x) = \begin{pmatrix} 2 \text{diag}((\hat{x})^{-1}) & 0_{mn^2 \times n} & 0_{mn^2 \times 2mn} \\ 0_{n \times mn^2} & m \text{diag}((p)^{-1}) & 0_{n \times 2mn} \\ 0_{2mn \times mn^2} & 0_{2mn \times n} & \text{diag}(\hat{A} \hat{x}) - \text{diag}(E_p) \end{pmatrix}.$$

It is easy to see that $D(x)$ has the same block structure as $\nabla^2 r(x, y)$ and we can prove our inequalities for each block separately. But remark that all blocks connected to $\hat{x}$ is blocks that appears in optimal transport problem and the required inequalities were proven in [JST19]. Hence, we only need to show that

$$\hat{D}_p(x, y) \preceq \hat{R}(x, y) \preceq 6\hat{D}_p(x),$$

where

$$\hat{D}_p(x) = \begin{pmatrix} m \text{diag}((p)^{-1}) & 0_{n \times 2mn} \\ 0_{2mn \times mn} & -\text{diag}(E_p) \end{pmatrix},$$

and $\hat{R}$ was defined in the proof of Theorem 4.3.

Also, in the proof of Theorem 4.3 we show that

$$Q_{\hat{R}(x)}(u, v) = \sum_{i,j} (-E_{ij}) \left( \frac{5u_j^2}{p_j} + 4u_jv_iy_i + 2v_i^2p_j \right).$$

Using the same idea, we can write the action of quadratic form induced by $\hat{D}_p$:

$$Q_{\hat{D}_p(x)}(u, v) = \sum_{i,j} (-E_{ij}) \left( \frac{u_i^2}{p_j} + v_i^2p_j \right).$$

Using the fact that $y_i \in [-1, 1]$, we can obtain the required by the following inequalities and finish the proof:

$$\frac{u_j^2}{p_j} + v_i^2p_j \leq \frac{5u_j^2}{p_j} + 4u_jv_iy_i + 2v_i^2p_j \leq \frac{6u_j^2}{p_j} + 6v_i^2p_j$$

By the exactly same arguments, we obtain the linear rate of converge for our Alternating Minimization (AM) scheme. We need to show last two points:

- Bound the complexity of each iteration;
- Bound the initial range;

**Lemma 7.2.** For $H(x, y)$, defined in [9], we can implement the steps
1. \(x^{k+1} = \arg\min_{x \in X} H(x, y^k)\);

2. \(y^{k+1} = \arg\min_{y \in Y} H(x^{k+1}, y)\),

in time \(O(mn^2)\).

**Proof.** First of all, divide a vector \(v\) from the definition of function \([9]\) into \(m + 1\) part and vector \(u\) into \(m\) parts. We have the following function to optimize by some regrouping and rewriting a regularizer in homogeneous manner:

\[
H(x, y) = \frac{2\|d\|_{\infty}}{m} \sum_{i=1}^{m} \left( \frac{m}{2\|d\|_{\infty}} (v_i, x_i) + \langle (y_i)^2, Ax_i \rangle + 10(x_i, \log x_i) \right) + \frac{m}{2\|d\|_{\infty}} \langle u_i, y_i \rangle + \langle B_{\varepsilon} p, (y_i)^2 \rangle + 10\|d\|_{\infty} \langle p, \log p \rangle + \langle v_{m+1}, p \rangle
\]

Remark that each \(x_i\) is independent from others and we can compute \(x_i^{(k+1)}\) apart as a solutions of the following optimization problems:

\[
x_i^{(k+1)} = \arg\min_{x \in \Delta^n} \left( \frac{m}{20\|d\|_{\infty}} v_i + \frac{1}{10} A^\top(y_i)^2, x \right) + \langle x, \log x \rangle,
\]

and the solution of this type of problems is well-known and proportional to \(\exp(-\gamma_i)\). The multiplication on the matrix \(A\) and \(A^\top\) can be computed in \(O(n^2)\) time, because these matrices consists of \(O(n^2)\) non-zero entries, and all these steps can be performed in \(O(mn^2)\).

Also we need to compute an optimal \(p\) by the same idea:

\[
p^{k+1} = \arg\min_{p \in \Delta^n} \left( \frac{1}{10\|d\|_{\infty}} v_{m+1} - \frac{1}{5m} \varepsilon \langle y^k \rangle^2, p \right) + \langle p, \log p \rangle.
\]

As in the previous case, an optimal \(p^{k+1}\) is proportional to \(\exp(-\gamma_{m+1})\) and it can be computed in \(O(mn^2)\) time.

For the computation of \(y_i^{(k+1)}\) remark that each \([y_i^{(k+1)}]_j\) can be computed separately as a solution of the following 1-D optimization problem:

\[
[y_i^{(k+1)}]_j = \arg\min_{y \in [-1, 1]} \left( \frac{m}{2\|d\|_{\infty}} [u_i]_j \cdot y + ([Ax_i^{(k+1)}]_j + [B_{\varepsilon} p^{(k+1)}]_j) \cdot y^2 \right)
\]

It could be easily solved in constant time if we know \(Ax_i^{(k+1)}\) and \(B_{\varepsilon} p^{(k+1)} = (p^\top, 0_n)\):

\[
[y_i^{(k+1)}]_j = \begin{cases} -1, & \alpha \leq -1 \\ 1, & \alpha \geq 1 \\ \alpha, & \alpha \in [-1, 1] \end{cases}, \text{ where } \alpha = -\frac{m}{4\|d\|_{\infty}} \frac{[u_i]_j}{[Ax_i]_j + [B_{\varepsilon} p]_j}.
\]

Hence, we can make all calculations in \(O(mn^2)\). \(\square\)

**Proof of Theorem 4.4.** To proof the final result, we need to remind the proximal operator for \(r\):

\[
\text{prox}_r^\varepsilon(v) = \arg\min_{z \in Z} \langle v, z \rangle + B_r(\varepsilon, z) = \arg\min_{z \in Z} \langle v - \nabla r(\zeta), z \rangle + r(z).
\]
Remark, that it is equivalent to the next view, separate over \(x\) and \(y\):

\[
\text{prox}_{x, y}^\varepsilon (v) = \arg \min_{x \in X, y \in Y} (v_x - \nabla_x r(\bar{x}, \bar{y}), x) + (v_y - \nabla_y r(\bar{x}, \bar{y}), y) + r(x, y)
\]  

(10)

We have precisely the type of problems that can be solved using AM scheme described above in linear time, moreover, each step reduces error by \(1/2\) factor (similar as [JST10]).

The only thing we need to bound is an initial error. For this goal we should bound the norm of the gradient and the argument of the proximal function in all calls during the algorithm.

Firstly, divide gradient operator \(G(z) = (G_x(z) \top, G_y(z) \top) \top\) into two parts and bound uniformly \(\ell_\infty\) and \(\ell_1\) norms of each part respectively:

\[
\|G_x(z)\|_\infty = \frac{1}{m} \|d\|_\infty + 2\|d\|_\infty A \top \|y\|_\infty \leq \frac{\|d\|_\infty}{m} + \frac{2\|d\|_\infty}{m} m \leq 3\|d\|_\infty;
\]

\[
\|G_x(z)\|_1 = \frac{1}{m} \|2\|d\|_\infty (c - Ax)\|_1 \leq \frac{2\|d\|_\infty}{m} (\|c\|_1 + \|Ax\|_1) \leq 8\|d\|_\infty
\]

In the inequality in the first row we used the fact \(m \geq 1\) for simplicity and in the second one we use the fact that matrix \(A\) and vector \(x_i\) are non-negative, hence, \(\|Ax_i\|_1 = \langle 1_n, Ax_i \rangle = 2\langle 1_n, x_i \rangle = 2\), where \(1_n\) is a vector consists of ones.

Then we can use the fact that the argument of the first prox-operator \(s^k = (s^k_x, s^k_y)\) is a sum of \(k\) gradients multiplied by \(1/\kappa\), computed in different points. In the second operator we also add gradient operator, multiplied by \(1/\kappa\). Since \(k \leq 2\kappa \Theta \cdot (\varepsilon/2)^{-1}\), we have by triangle inequality:

\[
\|s^k_x\|_\infty \leq \frac{k}{2\kappa} \cdot 3\|d\|_\infty \leq \frac{6\Theta\|d\|_\infty}{\varepsilon}
\]

\[
\|s^k_y\|_1 \leq \frac{k}{2\kappa} \cdot 8\|d\|_\infty \leq \frac{16\Theta\|d\|_\infty}{\varepsilon}
\]

Then, all our arguments of the proximal operator during the running time can be bounded in the following way (for \(\kappa = 3\)):

\[
\|v_x\|_\infty \leq \frac{6\Theta\|d\|_\infty}{\varepsilon} + \|d\|_\infty
\]

\[
\|v_y\|_1 \leq \frac{16\Theta\|d\|_\infty}{\varepsilon} + \frac{8}{3}\|d\|_\infty
\]

Then fix \(x^*\) and \(y^*\) as minimizers for the proximal operator \([\text{10}]\) and remind the bound for \(\Theta \leq 40\log n\|d\|_\infty + 6\|d\|_\infty\). Also we can compute \(\|\nabla_x r(\bar{x}, \bar{y})\|_\infty \leq 20\|d\|_\infty (2\log n + 1)\) and \(\|\nabla_y r(\bar{x}, \bar{y})\|_1 = 1\).

Then we can write a suboptimality gap \(\delta_0\) for our algorithm for any initial \(x^0\) and \(y^0\):

\[
\delta_0 = (v_x - \nabla_x r(\bar{x}, \bar{y}), x^0 - x^*) + (v_y - \nabla_y r(\bar{x}, \bar{y}), y^0 - y^*) + r(x^0, y^0) - r(x^*, y^*)
\]

\[
\leq \|v_x - \nabla_x r(\bar{x}, \bar{y})\|_\infty \|x^0 - x^*\|_1 + \|v_y - \nabla_y r(\bar{x}, \bar{y})\|_1 \|y^0 - y^*\|_\infty + \Theta
\]

\[
\leq 2\|d\|_\infty \cdot \left( \frac{6\Theta}{\varepsilon} + 20\log n + 10 \right) + \|d\|_\infty + 2 \cdot \frac{16\Theta\|d\|_\infty}{\varepsilon} + 8 \|d\|_\infty + \Theta
\]

\[
\leq \left( \frac{44\|d\|_\infty}{\varepsilon} + 2 \right) \Theta + 18\|d\|_\infty.
\]

Then we can compute the total number of iterations to obtain \(\varepsilon/2\) desired accuracy:

\[
N = \log_{24/23} \frac{2\delta_0}{\varepsilon} \leq 24\log \left( \frac{\left( \frac{88\|d\|_\infty}{\varepsilon^2} + \frac{4}{\varepsilon} \right) \Theta + \frac{36\|d\|_\infty}{\varepsilon} }{\varepsilon} \right) = O(\log \gamma),
\]

where \(\gamma = \|d\|^{-1} \log n\), as desired. Each iteration can be done in \(O(mn^2)\) time and we obtain the required complexity.