ON THE CONVERGENCE OF PROJECTED ALTERNATING MAXIMIZATION FOR EQUITABLE AND OPTIMAL TRANSPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies the equitable and optimal transport (EOT) problem, which has many applications such as fair division problems and optimal transport with multiple agents etc. In the discrete distributions case, the EOT problem can be formulated as a linear program (LP). Since this LP is prohibitively large for general LP solvers, Scetbon *et al.* Scetbon et al. (2021) suggests to perturb the problem by adding an entropy regularization. They proposed a projected alternating maximization algorithm (PAM) to solve the dual of the entropy regularized EOT. In this paper, we provide the first convergence analysis of PAM. A novel rounding procedure is proposed to help construct the primal solution for the original EOT problem. We also propose a variant of PAM by incorporating the extrapolation technique that can numerically improve the performance of PAM. Results in this paper may shed lights on block coordinate (gradient) descent methods for general optimization problems.

1 INTRODUCTION

Optimal transport (OT) is a classical problem that recently finds many emerging applications in machine learning and artificial intelligence, including generative models Arjovsky et al. (2017), representation learning Ozair et al. (2019), reinforcement learning Bellemare et al. (2017) and word embeddings Alvarez-Melis et al. (2019) etc. More recently, Scetbon *et al.* Scetbon et al. (2021) proposed an equitable and optimal transport (EOT) problem that targets to fairly distribute the workload of OT when there are multiple agents. In this problem formulation, there are multiple agents working together to move mass from measures μ to ν and each agent has its unique cost function. A very important issue that needs to be considered here is the fairness, which aims at finding transportation plans such that the workloads among all the agents are equal to each other. This can be achieved by minimizing the largest transportation cost among all agents, which leads to a convex-concave saddle point problem. The EOT problem has wide applications in economics and machine learning, such as fair division or the cake-cutting problem Moulin (2003); Brandt et al. (2016), multi-type resource allocation Mackin & Xia (2015), internet minimal transportation time and sequential optimal transport Scetbon et al. (2021).

We now describe the EOT problem formally. Given two discrete probability measures $\mu_n = \sum_{i=1}^n a_i \delta_{x_i}$ and $\nu_n = \sum_{i=1}^n b_i \delta_{y_i}$, the EOT studies the problem of transporting mass from μ to ν by N agents. Here, $\{x_1, x_2, ..., x_n\} \subseteq \mathbb{R}^d$ and $\{y_1, y_2, ..., y_n\} \subset \mathbb{R}^d$ are the support points of each measure and $a = [a_1, a_2, ..., a_n]^\top \in \Delta^n$, $b = [b_1, b_2, ..., b_n]^\top \in \Delta^n$ are corresponding weights for each measure, where Δ^n denotes the probability simplex in $\mathbb{R}^n$. Moreover, throughout this paper, we assume vector $b > 0$. For each agent k, we denote its unique cost function as $c^k(x, y), k \in [N] = \{1, \ldots, N\}$ and its cost matrix as C^k, where $C^k_{i,j} = c^k(x_i, y_j)$. Moreover, we define the following coupling decomposition set

$$\Pi^N_{a,b} := \left\{ \boldsymbol{\pi} = (\pi^k)_{k\in[N]} \,\middle|\, r\left(\sum_k \pi^k\right) = a, \quad c\left(\sum_k \pi^k\right) = b, \quad \pi^k_{ij} \geq 0, \forall i, j \in [n] \right\},$$

where $r(\pi) = \pi \mathbf{1}, c(\pi) = \pi^\top \mathbf{1}$ are the row sum and column sum of matrix π respectively. Mathematically, the EOT problem can be formulated as

$$\min_{\boldsymbol{\pi} \in \Pi_{a,b}^N} \max_{1 \le k \le N} \langle \pi^k, C^k \rangle. \tag{1}$$

When $N = 1$, (1) reduces to the standard OT problem. Note that (1) minimizes the point-wise maximum of a finite collection of functions. It is easy to see that (1) is equivalent to the following constrained problem:

$$\min_{\boldsymbol{\pi} \in \Pi_{a,b}^N} \max_{\lambda \in \Delta_+^N} \ell(\boldsymbol{\pi}, \lambda) := \sum_{k=1}^N \lambda_k \langle \pi^k, C^k \rangle. \tag{2}$$

The following proposition shows an important property of EOT: at the optimum of the minimax EOT formulation (2), the transportation costs of the agents are equal to each other.

Proposition 1 *(Scetbon et al., 2021, Proposition 1) Assume that all cost matrices $C^k, k \in [N]$ have the same sign. Let $\boldsymbol{\pi}^* \in \Pi_{a,b}^N$ be the optimal solution of (2). It holds that*

$$\langle (\pi^*)^i, C^i \rangle = \langle (\pi^*)^j, C^j \rangle, \quad \forall i, j \in [N]. \tag{3}$$

Note that Proposition 1 requires all cost matrices to have the same sign. When the cost matrices are all non-negative, (2) solves the transportation problem with multiple agents. When the cost matrices are all non-positive, the cost matrices are interpreted as the utility functions and (2) solves the fair division problem Moulin (2003).

The discrete OT is a linear programming (LP) problem (in fact, an assignment problem) with a complexity of $O(n^3 \log n)$ Tarjan (1997). Due to this cubic dependence on the dimension n, it is challenging to solve large-scale OT in practice. A widely adopted compromise is to add an entropy regularizer to the OT problem Cuturi (2013). The resulting problem is strongly convex and smooth, and its dual problem can be efficiently solved by the celebrated Sinkhorn's algorithm Sinkhorn & Knopp (1967); Cuturi (2013). This strategy is now widely used in the OT community due to its computational advantages as well as improved sample complexity Genevay et al. (2019). Similar ideas were also used for computing the Wasserstein barycenter Benamou et al. (2015), projection robust Wasserstein distance Paty & Cuturi (2019); Lin et al. (2020); Huang et al. (2021a), projection robust Wasserstein barycenter Huang et al. (2021b). Motivated by these previous works, Scetbon *et al.* Scetbon et al. (2021) proposed to add an entropy regularizer to (2), and designed a projected alternating maximization algorithm (PAM) to solve its dual problem. However, the convergence of PAM has not been studied. Scetbon *et al.* Scetbon et al. (2021) also proposed an accelerated projected gradient ascent algorithm (APGA) for solving a different form of the dual problem of the entropy regularized EOT. Since the objective function of this new dual form has Lipschitz continuous gradient, APGA is essentially the Nesterov's accelerated gradient method and thus its convergence rate is known. However, numerical experiments conducted in Scetbon et al. (2021) indicate that APGA performs worse than PAM. We will discuss the reasons in details later.

Our Contributions. There are mainly three issues with the PAM and APGA algorithms in Scetbon et al. (2021), and we will address all of them in this paper. Our results may shed lights on designing new block coordinate descent algorithms. Our main contributions are given below.

(i) The PAM algorithm in Scetbon et al. (2021) only returns the dual variables. How to find the primal solution of (2), i.e., the optimal transport plans $\boldsymbol{\pi}$, was not discussed in Scetbon et al. (2021). In this paper, we propose a novel rounding procedure to find the primal solution. Our rounding procedure is different from the one widely used in the literature Altschuler et al. (2017).

(ii) We provide the first convergence analysis of the PAM algorithm, and analyze its iteration complexity for finding an ϵ-optimal solution to the EOT problem (2). In particular, we show that it takes at most $O(Nn^2\epsilon^{-2})$ arithmetic operations to find an ϵ-optimal solution to (2). This matches the rate of the Sinkhorn's algorithm for computing the Wasserstein distance Dvurechensky et al. (2018).

(iii) We propose a variant of PAM that incorporates the extrapolation technique as used in Nesterov's accelerated gradient method. We name this variant as PAM with Extrapolation (PAME). The iteration complexity of PAME is also analyzed. Though we are not able to prove a better complexity over PAM at this moment, we find that PAME performs much better than PAM numerically.

Notation. For vectors a and b with the same dimension, $a./b$ denotes their entry-wise division. We denote $c_\infty := \max_k \|C^k\|_\infty$. Throughout this paper, we assume vector $b > 0$, and we denote $\iota := \min_j \log(b_j)$. We use $\mathbf{1}_n$ to denote the n-dimensional vector whose entries are all equal to one. We use $\mathbb{I}_{\mathcal{X}}(x)$ to denote the indicator function of set $\mathcal{X}$, i.e., $\mathbb{I}_{\mathcal{X}}(x) = 0$ if $x \in \mathcal{X}$, and $\mathbb{I}_{\mathcal{X}}(x) = \infty$ otherwise. We denote $c^t = c(\sum_{k=1}^N \pi^k(f^{t+1}, g^t, \lambda^t))$. For integer $N > 0$, we denote $[N] := \{1, \ldots, N\}$. We also denote $\boldsymbol{\pi}(f, g, \lambda) = [\pi^k(f, g, \lambda)]_{k\in[N]}$.

2 Projected Alternating Maximization Algorithm

The PAM algorithm proposed in Scetbon et al. (2021) aims to solve the entropy regularized EOT problem, which is given by

$$\min_{\boldsymbol{\pi}\in\Pi_{a,b}^N} \max_{\lambda\in\Delta_+^N} \ell_\eta(\boldsymbol{\pi}, \lambda) := \sum_{k=1}^N p_\eta^k(\pi^k, \lambda) \tag{4}$$

where $\eta > 0$ is a regularization parameter, $p_\eta^k(\pi^k, \lambda) := \lambda_k\langle\pi^k, C^k\rangle - \eta H(\pi^k)$, and the entropy function H is defined as $H(\pi) = -\sum_{i,j} \pi_{i,j}(\log \pi_{i,j} - 1)$. Note that (4) is a strongly-convex-concave minimax problem whose constraint sets are convex and bounded, and thus the Sion's minimax theorem Sion (1958) guarantees that

$$\min_{\boldsymbol{\pi}\in\Pi_{a,b}^N} \max_{\lambda\in\Delta_+^N} \ell_\eta(\boldsymbol{\pi}, \lambda) = \max_{\lambda\in\Delta_+^N} \min_{\boldsymbol{\pi}\in\Pi_{a,b}^N} \ell_\eta(\boldsymbol{\pi}, \lambda). \tag{5}$$

Now we consider the dual problem of $\min_{\boldsymbol{\pi}\in\Pi_{a,b}^N} \ell_\eta(\boldsymbol{\pi}, \lambda)$. First, we add a redundant constraint $\sum_{k,i,j} \pi_{i,j}^k = 1$ and consider the dual of

$$\min_{\boldsymbol{\pi}\in\Pi_{a,b}^N, \sum_{k,i,j}\pi_{i,j}^k=1} \ell_\eta(\boldsymbol{\pi}, \lambda). \tag{6}$$

The reason for adding this redundant constraint is to guarantee that the dual objective function is Lipschitz smooth. It is easy to verify that the dual problem of (6) is given by

$$\max_{f,g} \min_{\substack{\sum_{k,i,j}\pi_{i,j}^k=1,\\ \boldsymbol{\pi}\in(\mathbb{R}_+^{n\times n})^N}} \sum_{k=1}^N \lambda_k\langle\pi^k, C^k\rangle - \eta H(\boldsymbol{\pi}) + f^\top\left(a - r\left(\sum_k \pi^k\right)\right) + g^\top\left(b - c\left(\sum_k (\pi^k)^\top\right)\right), \tag{7}$$

where f and g are the dual variables and $H(\boldsymbol{\pi}) = \sum_k H(\pi^k)$. It is noted that problem (7) admits the following solution:

$$\pi^k(f, g, \lambda) = \frac{\zeta^k(f, g, \lambda)}{\sum_k \|\zeta^k(f, g, \lambda)\|_1}, \quad \forall k \in [N], \tag{8}$$

where

$$\zeta^k(f, g, \lambda) = \exp\left(\frac{f\mathbf{1}_n^\top + \mathbf{1}_n g^\top - \lambda_k C^k}{\eta}\right), \quad \forall k \in [N]. \tag{9}$$

By plugging (8) into (7), we obtain the following dual problem of (6):

$$\max_{f\in\mathbb{R}^n,\ g\in\mathbb{R}^n} \langle f, a\rangle + \langle g, b\rangle - \eta \log\left(\sum_{k=1}^N \|\zeta^k(f, g, \lambda)\|_1\right) - \eta. \tag{10}$$

Plugging (10) into (5), we know that the entropy regularized EOT problem (4) is equavalent to a pure maximization problem:

$$\max_{f\in\mathbb{R}^n,\ g\in\mathbb{R}^n,\ \lambda\in\Delta^N} F(f, g, \lambda) := \langle f, a\rangle + \langle g, b\rangle - \eta \log\left(\sum_{k=1}^N \|\zeta^k(f, g, \lambda)\|_1\right) - \eta. \tag{11}$$

Function $F(f, g, \lambda)$ is a smooth concave function with three block variables (f, g, λ). We use (f^*, g^*, λ^*) to denote an optimal solution of (11), and we denote $F^* = F(f^*, g^*, \lambda^*)$. The PAM

algorithm proposed in Scetbon et al. (2021) is essentially a block coordinate descent (BCD) algorithm for solving (11). More specifically, the PAM updates the three block variables by the following scheme:

$$f^{t+1} \in \underset{f}{\operatorname{argmax}}\, F(f, g^t, \lambda^t), \tag{12a}$$

$$g^{t+1} \in \underset{g}{\operatorname{argmax}}\, F(f^{t+1}, g, \lambda^t), \tag{12b}$$

$$\lambda^{t+1} := \operatorname{Proj}_{\Delta^N}\left(\lambda^t + \tau \nabla_\lambda F(f^{t+1}, g^{t+1}, \lambda^t)\right). \tag{12c}$$

Each iteration of PAM consists of two exact maximization steps followed by one projected gradient step. Importantly, the two exact maximization problems (12a)-(12b) have numerous optimal solutions, and we choose to use the following ones:

$$f^{t+1} = f^t + \eta \log\left(\frac{a}{r\left(\sum_{k=1}^N \zeta^k(f^t, g^t, \lambda^t)\right)}\right), \tag{13}$$

$$g^{t+1} = g^t + \eta \log\left(\frac{b}{c\left(\sum_{k=1}^N \zeta^k(f^{t+1}, g^t, \lambda^t)\right)}\right). \tag{14}$$

Furthermore, the optimiality conditions of (12a)-(12b) imply that

$$a - \frac{r\left(\sum_{k=1}^N \zeta^k(f^{t+1}, g^t, \lambda^t)\right)}{\sum_k \|\zeta^k(f^{t+1}, g^t, \lambda^t)\|_1} = 0, \quad b - \frac{c\left(\sum_{k=1}^N \zeta^k(f^{t+1}, g^{t+1}, \lambda^t)\right)}{\sum_k \|\zeta^k(f^{t+1}, g^{t+1}, \lambda^t)\|_1} = 0, \quad \forall t. \tag{15}$$

However, we need to point out that the PAM (12) only returns the dual variables (f^t, g^t, λ^t). One can compute the primal variable $\boldsymbol{\pi}$ using (8), but it is not necessarily a feasible solution. That is, $\boldsymbol{\pi}$ computed from (8) does not satisfy $\boldsymbol{\pi} \in \Pi_{a,b}^N$. How to obtain an optimal primal solution from the dual variables was not discussed in Scetbon et al. (2021). For the OT problem, i.e., $N = 1$, a rounding procedure for returning a feasible primal solution has been proposed in Altschuler et al. (2017). However, this rounding procedure cannot be applied to the EOT problem directly. In the next section, we propose a new rounding procedure for returning a primal solution based on the dual solution (f^t, g^t, λ^t). This new rounding procedure involves a dedicated way to compute the margins.

2.1 The Rounding Procedure and the Margins

Given $a \in \Delta^n$, $b \in \Delta^n$, and $\boldsymbol{\pi} = \{\pi^k\}_{k\in[N]}$ satisfying $r(\sum_k \pi^k) = a$, we construct vectors $a^k, b^k \in \mathbb{R}^n, k \in [N]$ from the procedure

$$(a^k, b^k)_{k\in[N]} = \text{Margins}(\boldsymbol{\pi}, a, b). \tag{16}$$

The details of this procedure is given below. First, we set $a^k = r(\pi^k)$, which immediately implies $\sum_{k=1}^N a^k = a$. We then construct b^k such that the following properties hold (these properties are required in our convergence analysis later):

(i) $b^k \geq 0$;

(ii) $\sum_{k=1}^N b^k = b$;

(iii) $\sum_{i=1}^n a_i^k = \sum_{j=1}^n b_j^k, \quad \forall k \in [N]$;

(iv) For any fixed $j \in [n]$, the quantities $b_j^k - [c(\pi^k)]_j$ have the same sign for all $k \in [N]$. That is, for any k and k', we have

$$(b_j^k - [c(\pi^k)]_j) \cdot (b_j^{k'} - [c(\pi^{k'})]_j) \geq 0, \tag{17}$$

which provides the following identity that is useful in our convergence analysis later:

$$\begin{aligned}\sum_{k=1}^N \|b^k - c(\pi^k)\|_1 &= \sum_{k=1}^N \sum_{j=1}^n |b_j^k - [c(\pi^k)]_j| = \sum_{j=1}^n \left|\sum_{k=1}^N (b_j^k - [c(\pi^k)]_j)\right| \\ &= \sum_{j=1}^n \left|b_j - \left[c\left(\sum_{k=1}^N \pi^k\right)\right]_j\right| = \left\|b - c\left(\sum_{k=1}^N \pi^k\right)\right\|_1.\end{aligned} \tag{18}$$

The procedure on constructing $(b^k)_{k\in[N]}$ satisfying these four properties is provided in Appendix A.

After $(a^k, b^k)_{k\in[N]}$ are constructed from (16) with $\boldsymbol{\pi} = \boldsymbol{\pi}(f^T, g^{T-1}, \lambda^{T-1})$, we adopt the rounding procedure proposed in Altschuler et al. (2017) to output a primal feasible solution $(\hat{\pi}^k)_{k\in[N]}$. The rounding procedure is described in Algorithm 2.

With this new procedure for rounding and computing the margins a^k, b^k, we now formally describe our PAM algorithm in Algorithm 1. Note that the algorithm is terminated when the following criteria are met:

$$\|c^{t-1} - b\|_1 \leq \epsilon/(6(6c_\infty - \eta\iota)), \tag{19a}$$

$$\left\|\lambda^t - \lambda^{t-1}\right\|_2 \leq \eta\epsilon/(18c_\infty^2), \tag{19b}$$

$$\tilde{F}(f^t, g^{t-1}, \lambda^{t-1}) \leq \epsilon/6, \tag{19c}$$

where $\tilde{F}(f, g, \lambda)$ is the suboptimality defined as: $\tilde{F}(f, g, \lambda) = F(f^*, g^*, \lambda^*) - F(f, g, \lambda)$.

Algorithm 1 Projected Alternating Maximization Algorithm

1: **Input:** Cost matrices $\{C^k\}_{1\leq k\leq N}$, vectors $a, b \in \Delta_+^n$ with $b > 0$, accuracy ϵ. $f^0 = g^0 = [1, ..., 1]^\top$, $\lambda^0 = [1/N, ..., 1/N]^\top \in \Delta_+^N$. $t = 0$
2: Choose parameters as

$$\eta = \min\left\{\frac{\epsilon}{3(\log(n^2N) + 1)}, c_\infty\right\}, \quad \tau = \eta/c_\infty^2. \tag{20}$$

3: **while** (19) is not met **do**
4: Compute f^{t+1} by (13)
5: Compute g^{t+1} by (14)
6: Compute λ^{t+1} by (12c)
7: $t \leftarrow t + 1$
8: **end while**
9: Assume stopping condition (19) is satisfied at the T-th iteration. Compute $(a^k, b^k)_{k\in[N]} = \text{Margins}(\boldsymbol{\pi}(f^T, g^{T-1}, \lambda^{T-1}), a, b)$ as in Section 2.1.
10: **Output:** $(\hat{\pi}, \hat{\lambda})$ where $\hat{\pi}^k = \text{Round}(\pi^k(f^T, g^{T-1}, \lambda^{T-1}), a^k, b^k)$, $\forall k \in [N]$, $\hat{\lambda} = \lambda^{T-1}$.

Algorithm 2 Round(π, a, b)

1: **Input:** $\pi \in \mathbb{R}^{n\times n}$, $a \in \mathbb{R}_+^n$, $b \in \mathbb{R}_+^n$.
2: $X = \text{Diag}\,(x)$ with $x_i = \frac{a_i}{r(\pi)_i} \wedge 1$
3: $\pi' = X\pi$
4: $Y = \text{Diag}\,(y)$ with $y_j = \frac{b_j}{c(\pi')_j} \wedge 1$
5: $\pi'' = \pi' Y$
6: $err_a = a - r(\pi''), err_b = b - c(\pi'')$
7: **Output:** $\pi'' + err_a err_b^\top / \|err_a\|_1$.

2.2 Connections with BCD and BCGD Methods

We now discsuss the connections between PAM and the block coordinate descent (BCD) method and the block coordinate gradient descent (BCGD) method. For the ease of presentation, we now assume that we are dealing with the following general convex optimization problem with m block variables:

$$\min_{x_i\in\mathcal{X}_i, i=1,\ldots,m} J(x_1, x_2, \ldots, x_m), \tag{21}$$

where $\mathcal{X}_i \subset \mathbb{R}^{d_i}$ and J is convex and differentiable. The BCD method for solving (21) iterates as follows:

$$x_i^{t+1} = \underset{x_i\in\mathcal{X}_i}{\text{argmin}}\ J(x_1^{t+1}, x_2^{t+1}, \ldots, x_{i-1}^{t+1}, x_i, x_{i+1}^t, \ldots, x_m^t), \tag{22}$$

and it assumes that these subproblems are easy to solve. The BCGD method for solving (21) iterates as follows:

$$x_i^{t+1} = \underset{x_i \in \mathcal{X}_i}{\operatorname{argmin}} \langle \nabla_{x_i} J(x_1^{t+1}, x_2^{t+1}, \ldots, x_{i-1}^{t+1}, x_i, x_{i+1}^t, \ldots, x_m^t), x_i - x_i^t \rangle + \frac{1}{2\tau} \|x_i - x_i^t\|_2^2, \quad (23)$$

where $\tau > 0$ is the step size. The PAM (12) is a hybrid of BCD (22) and BCGD (23), in the sense that some block variables are updated by exactly solving a maximization problem (the f and g steps), and some other block variables are updated by taking a gradient step (the λ step). Though this hybrid idea has been studied in the literature Hong et al. (2017); Xu & Yin (2013), their convergence analysis requires the blocks corresponding to exact minimization to be strongly convex. However, in our problem (11), the negative of the objective function is merely convex. Hence we need to develop new convergence proofs to analyze the convergence of PAM (Algorithm 1). How to extend our convergence results of PAM (Algorithm 1) to more general settings is a very interesting topic for future study.

3 Convergence Analysis of PAM

In this section, we analyze the iteration complexity of Algorithm 1 for obtaining an ϵ-optimal solution to the original EOT problem (2). The ϵ-optimal solution to (2) is defined as follows.

Definition 2 (see, e.g., Nemirovski (2005)) *We call $(\hat{\boldsymbol{\pi}}, \hat{\lambda}) \in \Pi_{a,b}^N \times \Delta^N$ an ϵ-optimal solution to the EOT problem (2) if the following inequality holds:*

$$\max_{\lambda \in \Delta^N} \ell(\hat{\boldsymbol{\pi}}, \lambda) - \min_{\boldsymbol{\pi} \in \Pi_{a,b}^N} \ell(\boldsymbol{\pi}, \hat{\lambda}) \leq \epsilon.$$

Note that the left hand side of the inequality is the duality gap of (2).

3.1 Main Result

We now present our main theorem, which gives the iteration complexity of PAM such that (55) is satisfied, and as a result of Lemma 13, an ϵ-optimal solution to the original EOT problem (2) is obtained.

Theorem 3 *Define $\epsilon' = \epsilon/(6c_\infty - \eta\iota)$, and set T as*

$$T = 5 + \frac{36}{\eta\sqrt{\gamma_0}\epsilon'} + \frac{648c_\infty^2}{\eta\epsilon} + \frac{28}{\eta\gamma_0\epsilon} = O\left(c_\infty^2 \epsilon^{-2}\right), \quad (24)$$

where $\gamma_0 = \min\left\{\frac{1}{(2c_\infty - \eta\iota)^2}, \frac{1}{9c_\infty^2}\right\}$ is a constant and we know $\gamma_0 = O(c_\infty^{-2})$. The output pair of Algorithm 1 is an ϵ-optimal solution of the EOT problem (2).

Proof. See Appendix B. □

4 Projected Alternating Maximization with Extrapolation

In this section, we discuss how to accelerate the PAM algorithm (Algorithm 1). It can be shown that the gradient of F in (11) is Lipschitz continuous[1]. Therefore, Scetbon *et al.* Scetbon et al. (2021) proposed to adopt Nesterov's accelerated gradient method Nesterov (2004) to solve (11). Their algorithm, named APGA (Accelerated Projected Gradient Ascent algorithm), iterates as follows:

$$(v, w, z)^\top \leftarrow (f^{t-1}, g^{t-1}, \lambda^{t-1})^\top + \frac{t-2}{t+1}\left((f^{t-1}, g^{t-1}, \lambda^{t-1})^\top - (f^{t-2}, g^{t-2}, \lambda^{t-2})^\top\right) \quad (25a)$$

$$(f^t, g^t)^\top \leftarrow (v, w)^\top + \frac{1}{L}\nabla_{(f,g)} F(v, w, z) \quad (25b)$$

$$(\lambda^t)^\top \leftarrow \text{Proj}_{\Delta^N}\left(z + \frac{1}{L}\nabla_\lambda F(v, w, z)\right), \quad (25c)$$

[1] In Lemma 6 we proved that $\nabla_\lambda F$ is Lipschitz continuous. The Lipschitz continuity of $\nabla_f F$ and $\nabla_g F$ can be proved similarly.

where L is the Lipschitz constant of ∇F. Note that APGA treats the problem (11) as a generic convex and smooth problem, and does not take advantage of the special structures of (11). In particular, f and g are updated using gradient ascent steps. This is in contrast to PAM in which f and g are obtained by exact maximizations, which is expected to improve the function value of F more significantly. In the following, we will design an accelerated algorithm that utilizes this property. Our method is called PAME (PAM with Extrapolation) and it incorporates the extrapolation technique to the gradient step for updating λ, and f and g are still updated using exact maximizations. We note that currently we are not able to prove a better complexity for PAME. Our iteration complexity result in Theorem 20 is in the same order as that of PAM, but numerically we have observed great improvement of PAME over PAM. It is an interesting future topic to study other accelerations to PAM that can provably achieve improved complexity.

A typical iteration of our PAME algorithm is given below:

$$f^{t+1} = f^t + \eta \log\left(\frac{a}{r\left(\sum_k \zeta^k(f^t, g^t, \lambda^t)\right)}\right), \tag{26a}$$

$$g^{t+1} = g^t + \eta \log\left(\frac{b}{c\left(\sum_k \zeta^k(f^{t+1}, g^t, \lambda^t)\right)}\right), \tag{26b}$$

$$y^{t+1} = \text{Proj}_{\Delta^N}\left(\lambda^t + (1-\theta)(\lambda^t - \lambda^{t-1})\right), \tag{26c}$$

$$\lambda^{t+1} = \text{Proj}_{\Delta^N}\left(y^{t+1} + \tau \nabla_\lambda F(f^{t+1}, g^{t+1}, y^{t+1})\right). \tag{26d}$$

Here $\theta \in (0,1)$ is a given parameter for the extrapolation step. We see that steps (26a)-(26b) are the same as (13)-(14) and they are solutions to the exact maximizations (12a)-(12b). Steps (26c)-(26c) give extrapolation to the gradient step for λ, similar to Nesterov's accelerated gradient method. Note that PAME (26) solves the dual entropy-regularized EOT problem (11). We use the same rounding procedure in Section 2.1 to generate a primal solution to the original EOT problem (1). The complete PAME algorithm is described in Algorithm 3. Note that the algorithm is terminated when the following criteria are met:

$$\|c^{t-1} - b\|_1 \leq \epsilon/(6(6c_\infty - \eta\iota), \tag{27a}$$

$$\|\lambda^{t-1} - \lambda^{t-2}\|_2 \leq \eta\epsilon/(60(1-\theta)c_\infty^2), \tag{27b}$$

$$\|\lambda^t - y^t\|_2 \leq \eta\epsilon/(42c_\infty^2), \tag{27c}$$

$$\tilde{F}(f^t, g^{t-1}, \lambda^{t-1}) \leq \epsilon/6. \tag{27d}$$

Algorithm 3 Projected Alternating Maximization with Extrapolation Algorithm

Input: Cost matrices $\{C^k\}_{1\leq k\leq N}$, accuracy ϵ, $\theta \in (0,1)$.
Initialization: $f^0 = g^0 = [1, ..., 1]^\top$, $\lambda^0 = [1/N, ..., 1/N]^\top \in \Delta^N$.
Choose parameters as

$$\eta = \min\left\{\frac{\epsilon}{3(\log(n^2N)+1)}, c_\infty\right\}, \quad \tau = \frac{\eta}{2c_\infty^2}. \tag{28}$$

while (27) is not met **do**
  Compute f^{t+1} by (26a)
  Compute g^{t+1} by (26b)
  Compute y^{t+1} by (26c)
  Compute λ^{t+1} by (26d)
  $t \leftarrow t+1$
end while
Assume the stop condition (27) is satisfied at the T-th iteration. Compute $(a^k, b^k)_{k\in[N]} = \text{Margins}(\boldsymbol{\pi}(f^T, g^{T-1}, \lambda^{T-1}), a, b)$ as in Section 2.1.
Output: $(\hat{\pi}, \hat{\lambda})$ where $\hat{\pi}^k = \text{Round}(\pi^k(f^T, g^{T-1}, \lambda^{T-1}), a^k, b^k)$, $\forall k \in [N]$, $\hat{\lambda} = \lambda^{T-1}$.

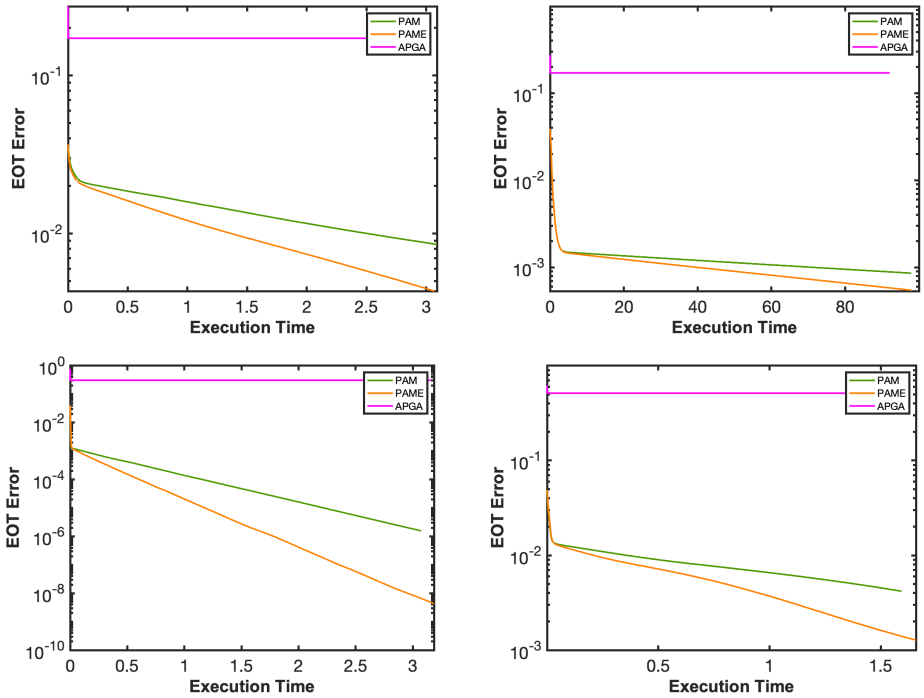

Figure 1: Computational time comparison between PAM, PAME and APGA algorithms on Gaussian distributions. **Upper Left**: $N = 10, n = 100, \eta = 0.1$, **Upper Right**: $N = 10, n = 500, \eta = 0.1$, **Bottom Left**: $N = 10, n = 100, \eta = 0.5$, **Bottom Right**: $N = 5, n = 100, \eta = 0.1$.

4.1 Convergence Analysis of PAME Algorithm

In this section, we analyze the iteration complexity of PAME (Algorithm 3) for obtaining an ϵ-optimal solution to the original EOT problem (1). The proof for PAME is different from that of PAM, and here we need to analyze the behavior of the following Hamiltonian, inspired by Jin *et al.* Jin et al. (2018).

$$E(f, g, \lambda^1, \lambda^2) = F(f, g, \lambda^1) - \frac{1}{2\tau}\|\lambda^1 - \lambda^2\|_2^2. \tag{29}$$

Theorem 4 *Define $\epsilon' = \epsilon/(6c_\infty - \eta\iota)$, and set T to be*

$$T = 8 + \frac{48}{\eta\sqrt{\gamma_1}\epsilon'} + \frac{\left(3600(1-\theta)^2 + 882\right) c_\infty^2 s}{\eta\epsilon^2} + \frac{48}{\eta\gamma_1\epsilon} = O\left(c_\infty^2\epsilon^{-2}\right), \tag{30}$$

where $\gamma_1 = \min\left\{\frac{1}{(2c_\infty - \eta\iota)^2}, \frac{2(2\theta-\theta^2)}{(7-5\theta)^2 c_\infty^2}, \frac{1}{49c_\infty^2}\right\}$ and we know $\gamma_1 = O(c_\infty^{-2})$. At least one of the iterations in Algorithm 3, after rounding, is an ϵ-saddle point of the EOT problem (2).

Proof. See Appendix C. □

5 Numerical Experiments

We compare the performance of PAME with PAM and APGA (25) Scetbon et al. (2021) on a synthetic dataset: the Gaussian distributions. We also conduct numerical comparison on another synthetic dataset: the fragmented hypercube dataset, and the results will be given in the Appendix.

Gaussian Distribution: Consider the case when two sets of discrete support $\{x_i\}_{i\in[n]}, \{y_j\}_{j\in[n]}$ are independently sampled from Gaussian distributions

$$\mathcal{N}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 10 & 1 \\ 1 & 10 \end{pmatrix}\right) \text{ and } \mathcal{N}\left(\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & -0.2 \\ -0.2 & 1 \end{pmatrix}\right) \tag{31}$$

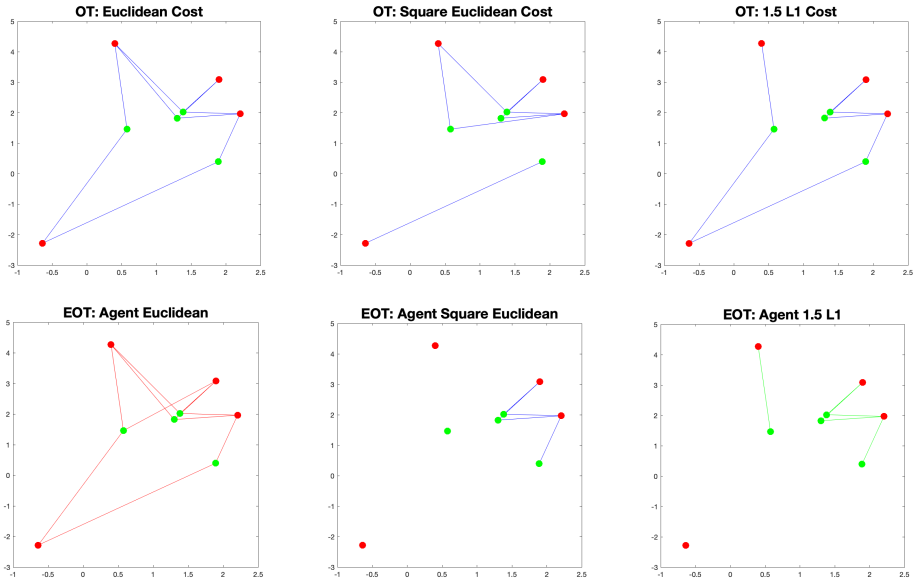

Figure 2: Optimal couplings of standard OT (first row) and EOT (second row). OT Square Euclidean Cost: 5.935; OT Euclidean Cost: 2.158; OT $L_1^{1.5}$ Cost: 5.030; EOT Cost: 0.906.

respectively. The base cost matrix C^{base} is computed by $C_{i,j}^{base} = \|x_i - y_j\|_2^2$. Assume we have N agents. The cost matrix of each agent can be obtained by adding Gaussian noise sampled from $\mathcal{N}(0, 10)$ to each element of the base cost. For instance, for the k-th agent with a cost matrix C^k, we have $C_{i,j}^k = |C_{i,j}^{base} + \mathcal{N}(0, 10)|$.

We then set $a = b = [1/n, ..., 1/n]$ for all experiments. For all algorithms, we set $\tau = \frac{5\eta}{c_\infty^2}$ and we set $\theta = 0.1$ for the PAME algorithm. We consider the EOT error as a measure of optimality. The EOT error at iteration t is defined by

$$Error = |\ell(\pi(f^t, g^t, \lambda^t), \lambda^t) - \ell^*|, \tag{32}$$

where ℓ^* is the approximated optimal value of EOT (2) obtained by running the PAM algorithm for 20000 iterations. Figures 1 plots the EOT error against the execution time for Gaussian distributions. We run each algorithm for 2000 iterations for different parameter settings. In all cases, the PAME and PAM perform significantly better than APGA, and PAME also shows significant improvement over PAM.

Figure 2 shows the optimal couplings obtained from the standard OT and EOT of two Gaussian distributions under three different metrics: the Euclidean cost ($\|\cdot\|_2$), the square Euclidean cost ($\|\cdot\|_2^2$) and the $L_1^{1.5}$ norm ($\|\cdot\|_1^{1.5}$) respectively. We set $n = 4, \eta = 0.05$ and generate samples independently according to (31). For the EOT problem, we consider three agents with cost matrices computed by the three metrics mentioned above. Note that the entropy regularized models lead to a dense transportation plan and Figure 2 only plots the couplings with a probability larger than 10^{-3}. We see that all the agents have the same total cost in the EOT model, and as expected, the cost is smaller than the other three OT costs obtained by using the same metric. We provide additional results in Appendix D.

6 Conclusion

In this paper, we provided the first convergence analysis of the PAM algorithm for solving the EOT problem. Specifically, we have shown that it takes at most $O(\epsilon^{-2})$ iterations for the PAM algorithm to find an ϵ-saddle point. We proposed a PAME algorithm which incorporates the extrapolation technique to PAM. The PAME shows significant numerical improvement over PAM. Results in this paper might shed lights on designing new BCD type algorithms.

REFERENCES

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in neural information processing systems*, pp. 1964–1974, 2017.

David Alvarez-Melis, Stefanie Jegelka, and Tommi S Jaakkola. Towards optimal transport with global invariances. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1870–1879. PMLR, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458, 2017.

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.

Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pp. 2292–2300, 2013.

Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In *International Conference on Machine Learning*, pp. 1367–1376. PMLR, 2018.

Aude Genevay, Lénaic Chizat, Francis Bach, Marco Cuturi, and Gabriel Peyré. Sample complexity of sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1574–1583, 2019.

M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo. Iteration complexity analysis of block coordinate descent method. *Mathematical Programming Series A*, 163(1):85–114, 2017.

Minhui Huang, Shiqian Ma, and Lifeng Lai. A Riemannian block coordinate descent method for computing the projection robust Wasserstein distance. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 4446–4455. PMLR, 2021a.

Minhui Huang, Shiqian Ma, and Lifeng Lai. Projection robust Wasserstein barycenters. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 4456–4465. PMLR, 2021b.

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pp. 1042–1085. PMLR, 2018.

Tianyi Lin, Chenyou Fan, Nhat Ho, Marco Cuturi, and Michael Jordan. Projection robust Wasserstein distance and Riemannian optimization. In *NeurIPS*, volume 33, 2020.

Erika Mackin and Lirong Xia. Allocating indivisible items in categorized domains. *arXiv preprint arXiv:1504.05932*, 2015.

Hervé Moulin. *Fair division and collective welfare*. MIT press, 2003.

A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2005.

Y. E. Nesterov. *Introductory lectures on convex optimization: A basic course*. Applied Optimization. Kluwer Academic Publishers, Boston, MA, 2004. ISBN 1-4020-7553-7.

Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. In *Advances in Neural Information Processing Systems*, pp. 15604–15614, 2019.

François-Pierre Paty and Marco Cuturi. Subspace robust Wasserstein distances. In *International Conference on Machine Learning*, pp. 5072–5081, 2019.

Meyer Scetbon, Laurent Meunier, Jamal Atif, and Marco Cuturi. Equitable and optimal transport with multiple agents. In *International Conference on Artificial Intelligence and Statistics*, pp. 2035–2043. PMLR, 2021.

R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21:343–348, 1967.

Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

Maurice Sion. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958.

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

Y. Xu and W. Yin. A block coordinate descent method for regularized multi-convex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 63(3):1758–1789, 2013.