Randomized methods for computing optimal transport without regularization and their convergence analysis

Yue Xie [1 2] **Zhongjian Wang** [3] **Zhiwen Zhang** [1]

Abstract

The optimal transport (OT) problem can be reduced to a linear programming (LP) problem through discretization. In this paper, we introduce the random block coordinate descent (RBCD) methods to directly solve this LP problem. Our approach involves restricting the potentially large-scale optimization problem to small LP subproblems constructed via randomly chosen working sets. By using a random Gauss-Southwell-q rule to select these working sets, we equip the vanilla version of (**RBCD**$_0$) with almost sure convergence and a linear convergence rate to solve general standard LP problems. To further improve the efficiency of the (**RBCD**$_0$) method, we explore the special structure of constraints in the OT problems and propose several approaches for refining the random working set selection and accelerating the vanilla method. Our preliminary numerical experiments demonstrate that the accelerated random block coordinate descent (**ARBCD**) method is comparable to Sinkhorn's algorithm when seeking solutions with relatively high accuracy, and offers the advantage of saving memory.

1. Introduction

Background and motivation The optimal transport problem was first introduced by Monge in 1781, which aims to find the most cost-efficient way to transport mass from a set of sources to a set of sinks. Later, the theory was modernized and revolutionized by Kantorovich in 1942, who found a key link between optimal transport and linear programming. In recent years, optimal transport has become a popular and powerful tool in data science, where it provides a very natural way to compare and interpolate probability distributions (Arjovsky et al., 2017; Lei et al., 2019; Wang et al., 2022; Haker et al., 2004; Perrot et al., 2016). There also exist deep connections between the optimal transport problems with quadratic cost functions and a diverse class of partial differential equations (PDEs) arising in statistical mechanics and fluid mechanics; see e.g. (Brenier, 1991; Benamou & Brenier, 2000; Otto, 2001; Jordan et al., 1998; Villani, 2021).

Recently, a deep particle method is proposed for learning and computing invariant measures of parameterized stochastic dynamical systems (Wang et al., 2022). To achieve this goal, the authors of this paper designed a deep neural network (DNN) to map a uniform distribution (source) to an invariant measure (target), where the Péclet number is an input parameter for the DNN. The network is trained by minimizing the 2-Wasserstein distance (W_2) between the measure of network output μ and target measure ν. They consider a discrete version of W_2 for finitely many samples of μ and ν, which involves a linear program (LP) optimized over doubly stochastic matrices (Sinkhorn, 1964). Motivated by the domain decomposition method (Toselli & Widlund, 2004) in scientific computing, which solves PDE using subroutines that solve problems on subdomains and has the advantage of saving memory (i.e., using the same computational resource, it can compute a larger problem), the authors of (Wang et al., 2022) devised a mini-batch interior point method. This approach involves sampling smaller sub-matrices while preserving row and column sums. It has proven to be highly efficient and integrates seamlessly with the stochastic gradient descent method for overall network training. However, they did not obtain convergence analysis.

The objectives of this paper are twofold. First, we aim to provide rigorous convergence analysis for the mini-batch interior point method presented in (Wang et al., 2022), with minimal modifications. Second, we seek to enhance the

[*]Equal contribution [1]Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China [2]HKU Musketeers Foundation Institute of Data Science, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China [3]Department of Statistics and CCAM, The University of Chicago, Chicago, USA. Correspondence to: Yue Xie <yxie21@hku.hk>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

mini-batch selection strategy, thereby achieving improved and more robust performance in computing optimal transport problems. We recognize that the mini-batch interior point method aligns with the random block coordinate descent (RBCD) method in optimization terminology. Specifically, it applies the block coordinate descent (BCD) method to the LP problem directly, selects the working set randomly, and solves subproblems using the primal-dual interior-point method (Wright, 1997) or any other efficient linear programming solver. Encouraged by the demonstrated efficiency of this approach, we will develop theoretical results for solving LP with RBCD methods and explore various strategies for selecting working sets.

Theorectical contributions In this work, we first introduce an expected Gauss-Southwell-q rule to guide the selection of the working set. It enables almost sure convergence and a linear convergence rate in expectation when solving a general standard LP. Based on this rule, we develop a vanilla RBCD method - **RBCD**$_0$, which selects the working set with complete randomness. Then, we investigate the special linear system present in the LP formulation of OT. Based on the analysis of this linear system, we propose various approaches to refine the working set selection and improve the performance of **RBCD**$_0$. A better estimation of the constant in the linear convergence rate is shown. Moreover, we incorporate an acceleration technique inspired by the momentum concept to improve the algorithm's efficiency.

Numerical experiments We perform numerical experiments to evaluate the performance of the proposed method **ARBCD** (Accelerated RBCD). Synthetic data sets of various shapes/dimensions and invariant measures generated from IPM methods are utilized to create distributions. Our experiments compare **ARBCD** with Sinkhorn's algorithm. Preliminary numerical results show that **ARBCD** is comparable to Sinkhorn's algorithm in computation time when seeking solutions with relatively high accuracy. We also test **ARBCD** on a large-scale OT problem, where Gurobi runs out of memory. This further justifies the memory-saving advantage of **ARBCD**.

Existing algorithms for OT Encouraged by the success in applying Sinkhorn's algorithm to the dual of entropy regularized OT (Cuturi, 2013), researchers have conducted extensive studies in this area, including other types of regularization (Blondel et al., 2018)(Gasnikov et al., 2016), acceleration (Guminov et al., 2021)(Lin et al., 2022) and numerical stability (Schmitzer, 2019). In (Huang et al., 2021), a Riemannian block coordinate descent method is applied to solve projection robust Wasserstein distance. The proposed approach employs entropy regularization, deterministic block coordinate descent, and techniques in Riemannian optimization. Other works that significantly deviate from the entropy regularization framework include (Li et al., 2018), which computes the Schrödinger bridge problem (equivalent to OT with Fisher information regularization), and multiscale strategies such as (Gerber & Maggioni, 2017) and (Liu et al., 2022). The RBCD method employed in this study is a regularization-free method. As a result, it avoids dealing with inaccurate solutions and numerical stability issues introduced by the regularization term. Furthermore, each subproblem in RBCD is a small-size LP, allowing for flexible resolution choices.

A review of previous research on (R)BCD is in Appendix A.

Organization The rest of the paper is organized as follows. In Section 2, we review the basic idea of optimal transport and Wasserstein distance. In Section 3, we introduce the expected Gauss-Southwell-q rule and a vanilla RBCD (**RBCD**$_0$) method for solving general LP problems. In Section 4, we propose several approaches to refine and accelerate the **RBCD**$_0$ method. In Section 5, preliminary numerical results are presented to demonstrate the performance of our proposed method. Finally, concluding remarks are made in Section 6. We keep proofs in the appendix.

Notation. For any matrix X, let $X(i,j)$ denote its element in the ith column and jth row, and let $X(:,j)$ represent its jth row vector. For a vector v, we usually use superscripts to denote its copies (e.g., v^k in kth iteration of an algorithm) and use subscripts to denote its components (e.g., v_i); for a scalar, we usually use subscripts to denote its copies. Occasional inconsistent cases will be declared in context. $\mathrm{mod}(k,n)$ means k modulo n. For any vector v, we define $\mathrm{supp}(v) \triangleq \{i \in \{1,\dots,n\} \mid v_i \neq 0\}$. Given a matrix $X \in \mathbb{R}^{n\times n}$, we define its vectorization as follows:

$$\mathrm{vec}(X) \triangleq (X(:,1)^T, X(:,2)^T, ..., X(:,n)^T)^T.$$

For any positive integer $k \geq 2$, we denote $[1,k] \triangleq \{1,...,k\}$. $\mathbf{1}_{n\times n}$ represents the $n \times n$ matrix of all ones.

2. Optimal transport problems and Wasserstein distance

The Kantorovich formulation of optimal transport can be described as follows,

$$\inf_{\gamma\in\Gamma(\mu,\nu)} \int_{X\times Y} C(x,y)\, \mathrm{d}\gamma(x,y) \tag{1}$$

where $\Gamma(\mu,\nu)$ is the set of all measures on $X \times Y$ whose marginal distribution on X is μ and marginal distribution on Y is ν, $C(x,y)$ is the transportation cost. In this article,

we refer to the Kantorovich formulation when we mention optimal transport.

Wasserstein distances are metrics on probability distributions inspired by the problem of optimal mass transport. They measure the minimal effort required to reconfigure the probability mass of one distribution in order to recover the other distribution. They are ubiquitous in mathematics (Villani, 2021). One can define the p-Wasserstein distance between probability measures μ and ν on a metric space Y with distance function $dist$ by

$$W_p(\mu,\nu) := \left(\inf_{\gamma\in\Gamma(\mu,\nu)} \int_{Y\times Y} dist(\tilde{y},y)^p \, \mathrm{d}\gamma(\tilde{y},y)\right)^{1/p} \tag{2}$$

where $\Gamma(\mu,\nu)$ is the set of probability measures γ on $Y\times Y$ satisfying $\gamma(A\times Y)=\mu(A)$ and $\gamma(Y\times B)=\nu(B)$ for all Borel subsets $A,B\subset Y$. Elements $\gamma\in\Gamma(\mu,\nu)$ are called couplings of the measures μ and ν, i.e., joint distributions on $Y\times Y$ with marginals μ and ν on each axis. p-Wasserstein distance is a special case of optimal transport when $X=Y$ and the cost function $c(\tilde{y},y)=dist(\tilde{y},y)^p$.

In the discrete case, the definition (2) has a simple intuitive interpretation: given a $\gamma\in\Gamma(\mu,\nu)$ and any pair of locations $(\tilde{y},y)$, the value of $\gamma(\tilde{y},y)$ tells us what proportion of μ mass at $\tilde{y}$ should be transferred to y, in order to reconfigure μ into ν. Computing the effort of moving a unit of mass from $\tilde{y}$ to y by $dist(\tilde{y},y)^p$ yields the interpretation of $W_p(\mu,\nu)$ as the minimal effort required to reconfigure μ mass distribution into that of ν.

In a practical setting (Peyré & Cuturi, 2019), referred to as a point cloud, the closed-form solution of μ and ν may be unknown, instead only n independent and identically distributed (i.i.d.) samples of μ and n i.i.d. samples of ν are available. In further discussion, n refers to the size of the problem. We approximate the probability measures μ and ν by empirical distribution functions:

$$\mu=\frac{1}{n}\sum_{i=1}^n \delta_{\tilde{y}^i} \qquad \text{and} \qquad \nu=\frac{1}{n}\sum_{j=1}^n \delta_{y^j}, \tag{3}$$

where δ_x is the Dirac measure. Any element in $\Gamma(\mu,\nu)$ can clearly be represented by a transition matrix, denoted as $\gamma=(\gamma_{i,j})_{i,j}$ satisfying:

$$\gamma_{i,j}\geq 0; \quad \forall j,\ \sum_{i=1}^n \gamma_{i,j}=\frac{1}{n}; \quad \forall i,\ \sum_{j=1}^n \gamma_{i,j}=\frac{1}{n}. \tag{4}$$

Then $\gamma_{i,j}$ means the mass of $\tilde{y}^i$ that is transferring to y^j.

We denote all matrices in $\mathbb{R}^{n\times n}$ satisfying (4) as Γ^n, then (2) becomes

$$\hat{W}(f) := \left(\inf_{\gamma\in\Gamma^n} \sum_{i,j=1}^{n,n} dist(\tilde{y}^i,y^j)^p\gamma_{i,j}\right)^{1/p}. \tag{5}$$

Remark 2.1. Γ^n is in fact the set of $n\times n$ doubly stochastic matrix (Sinkhorn, 1964) divided by n.

Another practical setting, which is commonly used in fields of computer vision (Peleg et al., 1989; Ling & Okada, 2007), is to compute the Wasserstein distance between two histograms. To compare two grey-scale figures (2D, size $n_0\times n_0$), we first normalize the grey scale such that the values of cells of each picture sum to one. We denote centers of the cell as $\{y^i\}_{i=1}^n$ and $\{\tilde{y}^i\}_{i=1}^n$, then we can use two probability measures to represent the two figures:

$$\mu=\sum_{i=1}^n r_{1,i}\delta_{\tilde{y}^i} \qquad \text{and} \qquad \nu=\sum_{j=1}^n r_{2,j}\delta_{y^j},$$

where $r_{1,i},r_{2,j}\geq 0, \forall 1\leq i,j\leq n$, $\sum_{i=1}^n r_{1,i}=\sum_{j=1}^n r_{2,j}=1$. The discrete Wasserstein distance (5) keeps the same form while the transition matrix follows different constraints:

$$\gamma_{i,j}\geq 0; \forall j,\ \sum_{i=1}^n \gamma_{i,j}=r_{2,j}; \forall i,\ \sum_{j=1}^n \gamma_{i,j}=r_{1,i}. \tag{6}$$

Note that in both settings, the computation of Wasserstein distance is reduced to an LP, i.e.,

$$\begin{aligned} \min \quad & \sum_{1\leq i,j\leq n} C_{i,j}\gamma_{i,j} \\ \text{subject to} \quad & \sum_{j=1}^n \gamma_{i,j}=r_{1,i},\ \sum_{j=1}^n \gamma_{i,j}=r_{2,i},\ \gamma_{i,j}\geq 0, \end{aligned} \tag{7}$$

where $r^1\triangleq(r_{1,1},...,r_{1,n})^T$ and $r^2\triangleq(r_{2,1},...,r_{2,n})^T$ are two probability distributions, and $C_{i,j}=dist(\tilde{x}^i,x^j)^p$. More generally, we let r^1 and r^2 be two nonnegative vectors and $C_{i,j}=C(\tilde{y}^i,y^j)$ be any appropriate transportation cost from $\tilde{y}^i$ to y^j, so (7) also captures the discrete OT.

However, when the number of particles n becomes large, the number of variables (entries of γ) scales like n^2, which leads to costly computation. Therefore, we will discuss random block coordinate descent methods to keep the computational workload in each iteration reasonable.

3. Random block coordinate descent for standard LP

In this section, we first generalize the LP problem (7) to a standard LP (see Eq.(8)). Then we propose a random block

coordinate descent algorithm for resolution. Its almost sure convergence and linear convergence rate in expectation are analyzed.

We consider the following standard LP problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^N} \quad & c^T x \\ \text{subject to} \quad & Ax = b,\ x \geq 0, \end{aligned} \tag{8}$$

where $A \in \mathbb{R}^{M\times N}$, $b \in \mathbb{R}^M$, $c \in \mathbb{R}^N$, hence M is the number of constraint and N is the total degree of freedom. Assume throughout that $M \leq N$. Suppose that $\mathcal{N} \triangleq \{1,\dots,N\}$ and denote $\mathcal{X} \triangleq \{x \in \mathbb{R}^N \mid Ax = b, x \geq 0\}$ as the feasible set. Assume that (8) is finite and has an optimal solution. For any $x \in \mathcal{X}$ and $\mathcal{I} \subseteq \mathcal{N}$, denote

$$\mathcal{D}(x;\mathcal{I}) \triangleq \arg\min_{d\in\mathbb{R}^N} \left\{ c^T d \,\middle|\, \begin{array}{l} x + d \geq 0, Ad = 0, \\ d_i = 0, \forall i \in \mathcal{N}\setminus\mathcal{I} \end{array} \right\}. \tag{9}$$

$$q(x;\mathcal{I}) \triangleq \min_{d\in\mathbb{R}^N} \left\{ c^T d \,\middle|\, \begin{array}{l} x + d \geq 0, Ad = 0, \\ d_i = 0, \forall i \in \mathcal{N}\setminus\mathcal{I} \end{array} \right\}. \tag{10}$$

Namely, $\mathcal{D}(x;\mathcal{I})$ is the optimal solution set of the linear program in (9) and $q(x;\mathcal{I})$ is the optimal function value. We have that $q(x;\mathcal{I}) = c^T d$ for any $d \in \mathcal{D}(x;\mathcal{I})$. Denote $\mathcal{X}^*$ as the optimal solution set of (8). Then the following equations hold for any $x \in \mathcal{X}$:

$$\mathcal{X}^* = x + \mathcal{D}(x;\mathcal{N}), \tag{11}$$

$$q(x;\mathcal{N}) = c^T x^* - c^T x, \quad \forall x^* \in \mathcal{X}^*. \tag{12}$$

Consider the block coordinate descent (BCD) for (8):

$$\begin{aligned} &\text{find } d^k \in \mathcal{D}(x^k, \mathcal{I}_k), \\ &x^{k+1} := x^k + d^k, \end{aligned} \tag{13}$$

where $\mathcal{I}_k \subset \mathcal{N}$ is the working set chosen at iteration k. Next, we describe several approaches to select it.

Gauss-Southwell-q rule Motivated by the *Gauss-Southwell-q* rule introduced in (Tseng & Yun, 2009b), we desire to select $\mathcal{I}_k$ such that

$$q(x^k;\mathcal{I}_k) \leq v q(x^k;\mathcal{N}), \tag{14}$$

for some constant $v \in (0,1]$. Note that by (12), we have

$$q(x^k;\mathcal{N}) = c^T(x^* - x^k), \tag{15}$$

where x^* is an optimal solution of (8). Therefore, (10)-(15) imply that

$$\begin{aligned} & c^T d^k \leq v c^T(x^* - x^k) \\ \overset{(13)}{\Longrightarrow}\ & c^T(x^{k+1} - x^k) \leq v c^T(x^* - x^k) \\ \Longrightarrow\ & c^T(x^{k+1} - x^*) \leq (1-v) c^T(x^k - x^*). \end{aligned} \tag{16}$$

(16) indicates that the gap of function value decays exponentially with rate $1 - v$, as long as we choose $\mathcal{I}_k$ according to the Gauss-Southwell-q rule (14) at each iteration k. A trivial choice of $\mathcal{I}_k$ to satisfy (14) is $\mathcal{N}$ and $v = 1$. However, this choice results in a potential large-scale subproblem in the BCD method (13), contradicting the purpose of using BCD. Instead, we should set an upper bound on $|\mathcal{I}_k|$, namely, a reasonable batch size to balance the computational effort in each iteration and convergence performance of BCD. Next, we discuss the existence of such an $\mathcal{I}_k$ given an upper bound l on $|\mathcal{I}_k|$, which necessitates the following concept.

Definition 3.1. Vector $\bar{d} \in \mathbb{R}^N$ is conformal to $d \in \mathbb{R}^N$ if

$$\operatorname{supp}(\bar{d}) \subseteq \operatorname{supp}(d),\ \bar{d}_i d_i \geq 0, \forall i \in \mathcal{N}.$$

The following Theorem confirms the existence of such an $\mathcal{I}_k$ that satisfies (14).

Theorem 3.2. *Suppose that* $\operatorname{rank}(A) + 1 \leq N$. *Given any* $x \in \mathcal{X}$, $l \in \{\operatorname{rank}(A)+1,\dots,N\}$ *and* $d \in \mathcal{D}(x;\mathcal{N})$. *There exist a set* $\mathcal{I} \in \mathcal{N}$ *satisfying* $|\mathcal{I}| \leq l$ *and a vector* $\bar{d} \in \operatorname{null}(A)$ *conformal to* d *such that*

$$\mathcal{I} = \operatorname{supp}(\bar{d}). \tag{17}$$

$$q(x;\mathcal{I}) \leq \frac{1}{N-l+1} q(x;\mathcal{N}). \tag{18}$$

However, it is not clear how to identify the set $\mathcal{I}$ described in Theorem 3.2 with little computational effort for a general A. Therefore, we introduced the following.

Expected Gauss-Southwell-q rule We introduce randomness in the selection of $\mathcal{I}_k$ to reduce the potential computation burden in identifying an $\mathcal{I}_k$ that satisfies (14). Consider an *expected Gauss-Southwell-q rule*:

$$\mathbb{E}[q(x^k;\mathcal{I}_k) \mid \mathcal{F}_k] \leq v q(x^k;\mathcal{N}), \tag{19}$$

where $v \in (0,1]$ is a constant, and $\mathcal{F}_k \triangleq \{x^0,\dots,x^k\}$ denotes the history of the algorithm. Therefore, using the notations of LP (8) and BCD method (13):

$$\begin{aligned} & \text{(10)(15)(19)} \\ \Longrightarrow\ & \mathbb{E}[c^T d^k \mid \mathcal{F}_k] \leq v c^T(x^* - x^k) \qquad (20) \\ \Longrightarrow\ & \mathbb{E}[c^T(x^{k+1} - x^k) \mid \mathcal{F}_k] \leq v c^T(x^* - x^k) \\ \Longrightarrow\ & \mathbb{E}[c^T(x^{k+1} - x^*) \mid \mathcal{F}_k] \leq (1-v) c^T(x^k - x^*), \qquad (21) \end{aligned}$$

where x^* is an optimal solution of (8). By Lemma 10, page 49 in (Polyak, 1987), $c^T(x^k - x^*) \to 0$ almost surely. Moreover, if we take expectations on both sides of (21),

$$\mathbb{E}[c^T(x^{k+1} - x^*)] \leq (1-v)\mathbb{E}[c^T(x^k - x^*)]$$

$$\implies \mathbb{E}[c^T(x^k - x^*)] \leq (1-v)^k \mathbb{E}[c^T(x^0 - x^*)].$$

i.e., the expectation of function value gap converges to 0 exponentially with a rate $1 - v$.

Vanilla random block coordinate descent Based on the expected Gauss-Southwell-q rule, we formally propose a vanilla random block coordinate descent (**RBCD$_0$**) algorithm (Algorithm 1) to solve the LP (8). Specifically, we choose the working set $\mathcal{I}_k$ with full randomness, that is, randomly choose an index set of cardinality l out of $\mathcal{N}$. Then with probability at least $\frac{1}{\binom{N}{l}}$, the index set will be the same as or cover the working set suggested by Theorem 3.2. As a result, (19) will be satisfied with $v \geq \frac{1}{\binom{N}{l}(N-l+1)}$.

Algorithm 1 Vanilla random block coordinate descent (**RBCD$_0$**)

(Initialization) Choose feasible $x^0 \in \mathbb{R}^N$ and the batch size l such that $\text{rank}(A) + 1 \leq l \leq N$.
for $k = 0, 1, 2, \ldots$ **do**
  Step 1. Choose $\mathcal{I}_k$ uniformly randomly from $\mathcal{N}$ with $|\mathcal{I}_k| = l$.
  Step 2. Find $d^k \in \mathcal{D}(x^k; \mathcal{I}_k)$.
  Step 3. $x^{k+1} := x^k + d^k$.
end for

Based on the previous discussions, Algorithm 1 generates a sequence $\{x^k\}$ such that the value of $c^T x^k$ converges to the optimal with probability 1. Moreover, the expectation of the optimality gap converges to 0 exponentially. It is important to note that $\frac{1}{\binom{N}{l}(N-l+1)}$ is only a loose lower bound of v. This bound can become quite small when N grows large due to the binomial coefficient $\binom{N}{l}$. However, we expect that this lower bound is rarely reached in practice. In the following subsection, we will discuss how to further improve this bound given the structure of the OT problem.

4. Random block coordinate descent and optimal transport

Denote the cost matrix $C \triangleq (C_{i,j})_{n\times n}$ in (7). Then calculating the OT between two measures with finite support (problem (7)) is a special case of (8), where $c = \text{vec}(C)$, and $N = n^2$. Then A has the following structure:

$$A \triangleq \underbrace{\begin{pmatrix} I_n & I_n & \ldots & I_n \\ \mathbf{1}_n^T & & & \\ & \mathbf{1}_n^T & & \\ & & \ddots & \\ & & & \mathbf{1}_n^T \end{pmatrix}}_{n \text{ blocks}}, \tag{22}$$

where I_n is an $n \times n$ identity matrix, $\mathbf{1}_n$ is an n dimensional vector of all 1's (then $M = 2n$). Blank spaces represent 0s. Right hand side b in (8) has the form $b \triangleq ((r^1)^T, (r^2)^T)^T$, where $r^1, r^2 \in \mathbb{R}^n_+$ can be two discrete probability distributions. Now we discuss two approaches to carefully select the support set $\mathcal{I}_k$ at iteration k of the block coordinate descent method (13):

1. *Diagonal band.* Given $3 \leq p \leq n$, denote

$$\mathcal{G} \triangleq \left\{ \begin{array}{c|l} (i,j) & i \in [j, j+p-1] \\ \in \mathbb{Z}^2 & \text{if} \quad j \in [1, n-p+1]; \\ & i \in [1, ..., j+p-n-1] \cup [j, n] \\ & \text{if} \quad j \in [n-p+2, n] \end{array} \right\}$$

and construct matrix $G \in \mathbb{R}^{n\times n}$ such that

$$G(i,j) = \begin{cases} 1, & \text{if} \quad (i,j) \in \mathcal{G}, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Therefore, G has the following structure:

$$p \left\{ \begin{pmatrix} 1 & & & & 1 & \ldots & 1 \\ \vdots & 1 & & & & \ddots & \vdots \\ 1 & \vdots & \ddots & & & & 1 \\ 1 & 1 & & 1 & & & \\ & 1 & \ddots & \vdots & 1 & & \\ & & \ddots & 1 & \vdots & \ddots & \\ & & & 1 & 1 & \ldots & 1 \end{pmatrix} \right\} (p-1)$$
$$n \times n$$

It is like a band of width p across the diagonal, hence the name. Then we may construct $\bar{D}^k \in \mathbb{R}^{n\times n}$ and $\mathcal{I}_k$ as follows:

$$\begin{array}{l} \text{Obtain } \bar{D}^k \text{ by uniformly randomly} \\ \text{permuting all columns and rows of } G. \\ \text{Let} \quad \mathcal{I}_k \triangleq \text{supp}(\text{vec}(\bar{D}^k)). \end{array} \tag{24}$$

Note that $|\mathcal{I}_k| = np$.

2. *Submatrix.* Given $m < n$, obtain $\bar{D}^k$ and $\mathcal{I}_k$ such that

Uniformly randomly pick two sets of m different

numbers out of $[1, n]$: $i_1, ..., i_m$ and $j_1, ..., j_m$.

$$\text{Let} \quad \bar{D}^k(i,j) = \begin{cases} 1 & \text{if} \quad i \in \{i_1, ..., i_m\} \\ & \text{and} \quad j \in \{j_1, ..., j_m\}, \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{Let} \quad \mathcal{I}_k \triangleq \text{supp}(\text{vec}(\bar{D}^k)). \tag{25}$$

In this case, the support of $\bar{D}^k$ is a submatrix of size $m \times m$. Therefore, $|\mathcal{I}_k| = m^2$.

Via the diagonal band approach, we can improve the chance to guess the potential directions along which the transport cost is minimized by a large amount. The speed of the algorithm is enhanced while convergence is maintained.

As for the submatrix approach, we often find it quite efficient in numerical experiments. However, global convergence with a fixed-width submatrix may not be guaranteed. Therefore, we seek to combine these two approaches together.

Algorithm 2 Random block coordinate descent - submatrix and diagonal Band (**RBCD-SDB**)

(Initialization) Choose feasible $X^0 \in \mathbb{R}^{n \times n}$, submatrix row/column dimension m, band width $p \in [3, n]$ and selection parameter $s \in (0, 1]$. Let $x^0 = \text{vec}(X^0)$.
for $k = 0, 1, 2, \ldots$ **do**
  Step 1. With probability s, choose $\mathcal{I}_k$ according to (24); otherwise, choose $\mathcal{I}_k$ according to (25).
  Step 2. Find $d^k \in \mathcal{D}(x^k; \mathcal{I}_k)$.
  Step 3. $x^{k+1} := x^k + d^k$.
end for

Convergence of Alg. 2 is guaranteed by the next theorem.

Theorem 4.1. *Consider* (8)(22). *Then sequence* $\{x^k\}$ *and* $\{\mathcal{I}_k\}$ *generated by Algorithm 2 satisfies the expected Gauss-Southwell-q rule* (19), *with* $v \geq \frac{sn(p-2)}{(n^2-3)(n!)^2}$. *Therefore,* $c^T(x^k - x^*) \to 0$ *almost surely and* $\mathbb{E}[c^T(x^k - x^*)]$ *converges to* 0 *exponentially with rate* $1 - v$.

Remark 4.2. It can be shown that if n is large enough and p is chosen between $O(log(n))$ and $O(n)$, then the lower bound for constant v derived in Theorem 4.1 is better than the one estimated for Algorithm 1, i.e., $\frac{1}{\binom{N}{l}(N-l+1)}$. In fact, we have the following results.

Lemma 4.3. *Suppose that* $\bar{K} \geq 2$ *and* $\eta > 0$ *satisfies*

$$\frac{2\bar{K} - 3}{2(\bar{K} - 1)} + \log\left(\frac{\bar{K}}{2}\right) > 2/\eta,$$

and n *satisfies*

$$n \geq \frac{4}{\left(\frac{2\bar{K}-3}{2(\bar{K}-1)} + \log\left(\frac{\bar{K}}{2}\right)\right)\eta - 2}, \frac{n}{\log(n)} \geq \eta\bar{K}, n \geq \frac{2}{s}.$$

Then for any $p \in [\eta \log(n), \frac{n}{\bar{K}}]$, *and* $p \geq 3$, *we have* $\frac{sn(p-2)}{(n^2-3)(n!)^2} \geq \frac{1}{\binom{n^2}{np}(n^2-np+1)}$.

Let $n \geq 30$, $\eta = 1$, $\bar{K} = 8$, $s \geq 0.1$. Then according to Lemma 4.3, for $log(n) \leq p \leq n/8$, the lower bound $\frac{sn(p-2)}{(n^2-3)(n!)^2}$ is larger. We believe that this is a fairly reasonable range of p when n grows large. This lower bound is improved because we have knowledge of the structure of matrix A in the OT problems. In addition, it is possible to further sharpen the current convergence rate and we will address this in our future work.

Accelerated random block coordinate descent Algorithm 3 is an accelerated random block coordinate descent (**ARBCD**) algorithm. It selects the working set $\mathcal{I}_k$ in a different way from Algorithm 2 intermittently for acceleration. At times, we build $\mathcal{I}_k$ based on the iterates generated by the algorithm in the past, i.e., $x^{end} - x^{start}$. This vector reflects the progress achieved by running the **RBCD-SDB** for a few iterations. It predicts the direction in which the algorithm potentially makes further improvements. Such a choice is analogous to the momentum concept and often employed acceleration techniques in optimization, such as in the heavy ball method and Nesterov acceleration. Algorithm 3 has a similar convergence rate as Algorithm 2 (note that acceleration iteration happens occasionally). However, we expect that the acceleration technique leads to a better performance than Algorithm 1 and 2.

5. Numerical experiments

In this section, we conduct numerical experiments on various examples of OT problems[1]. We focus on **ARBCD**, which is the best among Algorithm 1 - 3. In Section 5.1, we compare **ARBCD** with **Sinkhorn**. A large-scale OT problem is solved using **ARBCD** in Section 5.2.

5.1. Comparison between ARBCD and the Sinkhorn's algorithm

Experiment settings We generated 8 pairs of distributions/patterns based on synthetic and real datasets. Descriptions are as follows. Note that we use histogram settings (c.f. Section 2) for datasets 1 and 2, and point cloud settings (c.f. Section 2) for other datasets. We use norm square cost function: $c(x, y) = \|x - y\|^2$.

Dataset 1: Uniform distribution to standard normal distri-

[1]All experiments are conducted using Matlab R2021b on Dell OptiPlex 7090 with CPU: Intel(R) Core(TM) i9-10900 @ 2.80GHz (20 CPUs), ~2.8GHz and RAM: 65536Mb. Data and codes are uploaded to `https://github.com/gxybrh/RBCDforOT`.

Algorithm 3 Accelerated random block coordinate descent (**ARBCD**)

(Initialization) Choose feasible $X^0 \in \mathbb{R}^{n\times n}$, submatrix row/column dimension m, band width $p \in [3, n]$, selection parameter $s \in [0, 1]$, and acceleration interval T. Let $x^0 = \mathrm{vec}(X^0)$, $x^{start} = x^{end} = x^0$. Binary variable acc.
for $k = 0, 1, 2, \ldots$ **do**
  Step 1. Choose $\mathcal{I}_k$ as following.
  if $\mathrm{mod}(k+1, T) \neq 0$ or $|\,\mathrm{supp}(x^{end} - x^{start})| \leq m^2$ **then**
    $acc =$ `false`. With probability s, choose $\mathcal{I}_k$ according to (24); otherwise, choose $\mathcal{I}_k$ according to (25).
  else
    $acc =$ `true`. Choose $\mathcal{I}_k$ uniformly randomly from $\mathrm{supp}(x^{end} - x^{start})$ so that $|\mathcal{I}_k| = m^2$.
  end if
  Step 2. Find $d^k \in \mathcal{D}(x^k; \mathcal{I}_k)$.
  Step 3. Update $x^{k+1} := x^k + d^k$;
  Step 4. Update $x^{end} = x^{k+1}$.
  if $acc =$ `true`. **then**
    Update $x^{start} = x^{k+1}$.
  end if
end for

bution over $[-1, 1]$. $n = 200$.
Dataset 2: Uniform distribution to a randomly shuffled[2] standard normal distribution over $[-1, 1]$. $n = 1000$.
Dataset 3: Uniform distribution over $[-\pi, \pi]^2$ to an empirical invariant measure generated from IPM methods. $n = 1000$.
Dataset 4: Distribution of $\sqrt{\Sigma}u$ to distribution of $2\sqrt{\Sigma}v - (1; 1; 1)$, where $\Sigma = \begin{pmatrix} 1 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 1 \end{pmatrix}$, u and v conform uniform distributions on $[0, 1]^3$ and are independent. $n = 1000$.
Dataset 5: Similar to Dataset 4, with $\Sigma = \begin{pmatrix} 1 & 0.8 & 0.64 \\ 0.8 & 1 & 0.8 \\ 0.64 & 0.8 & 1 \end{pmatrix}$. $n = 1000$.
Dataset 6: Distribution of Σu to distribution of Σv, where $\Sigma = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix}^T$, u conforms a uniform distribution on $[0, 2\pi]^2$ and v conforms a uniform distribution on $[-1, 1]^2$. $n = 1000$.

[2]We randomly shuffled the weights of the normal distribution histogram.

Dataset 7: Distribution of $\underbrace{(1; 1; \ldots; 1)^T}_{10} u$ to distribution of $(1; 2; 3; \ldots; 10)^T v + (1; 1; \ldots; 1)^T$, where u conforms uniform distribution over $[0, 2\pi]$ and v conforms uniform distribution over $[-1, 1]$. $n = 1000$.
Dataset 8: Distibution of a "cylinder" to a "spiral", see Figure 1. $n = 1000$.
In all cases, we normalize the cost matrix C such that its maximal element is 1. For all cases, we use the `linprog` in `Matlab` to find a solution with high precision (dual-simplex, constraint tolerance 1e-9, optimality tolerance 1e-10). We refer readers to the github repository for more details.

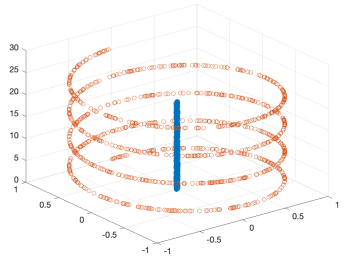

Figure 1. Visualization of dataset 8

Methods Implementation of **Sinkhorn** and **ARBCD** are specified as follows.
Sinkhorn. The algorithm proposed in (Cuturi, 2013) to compute Wasserstein distance. Let γ be the coefficient of the entropy term. We let $\gamma = \epsilon/(4 \log n)$ as suggested in (Dvurechensky et al., 2018). We consider the settings $\epsilon = 10^{-4}, 10^{-3}, 0.01, 0.1$. Iterations of **Sinkhorn** are projected onto the feasible region using a rounding procedure: Algorithm 2 in (Altschuler et al., 2017). Note that this projection step is added only for evaluation purposes because Sinkhorn does not provide feasible solutions if early stopped. It does not affect Sinkhorn's main steps or Sinkhorn's convergence at all. A similar approach is used for evaluation in (Jambulapati et al., 2019). In addition, we take all the updates to log space and use the LogSumExp function to avoid numerical instability issues. We stop **Sinkhorn** after 3×10^5 iter. when $n = 200$ and 10^5 iterations if $n = 1000$.
ARBCD. Algorithm 3: Accelerated random block coordinate descent. Let $m = 40$ when $n = 200$ and $m = 100$ when $n = 1000$. Let $p = \lfloor m^2/n \rfloor$, $s = 0.1$ and $T = 10$. Stop the algorithm after 10000 iterations. To be fair, we also project the solution in each iteration onto the feasible region via the rounding procedure. LP subproblems are solved via `linprog` in `Matlab` with high precision (dual-simplex, constraint tolerance 1e-9).

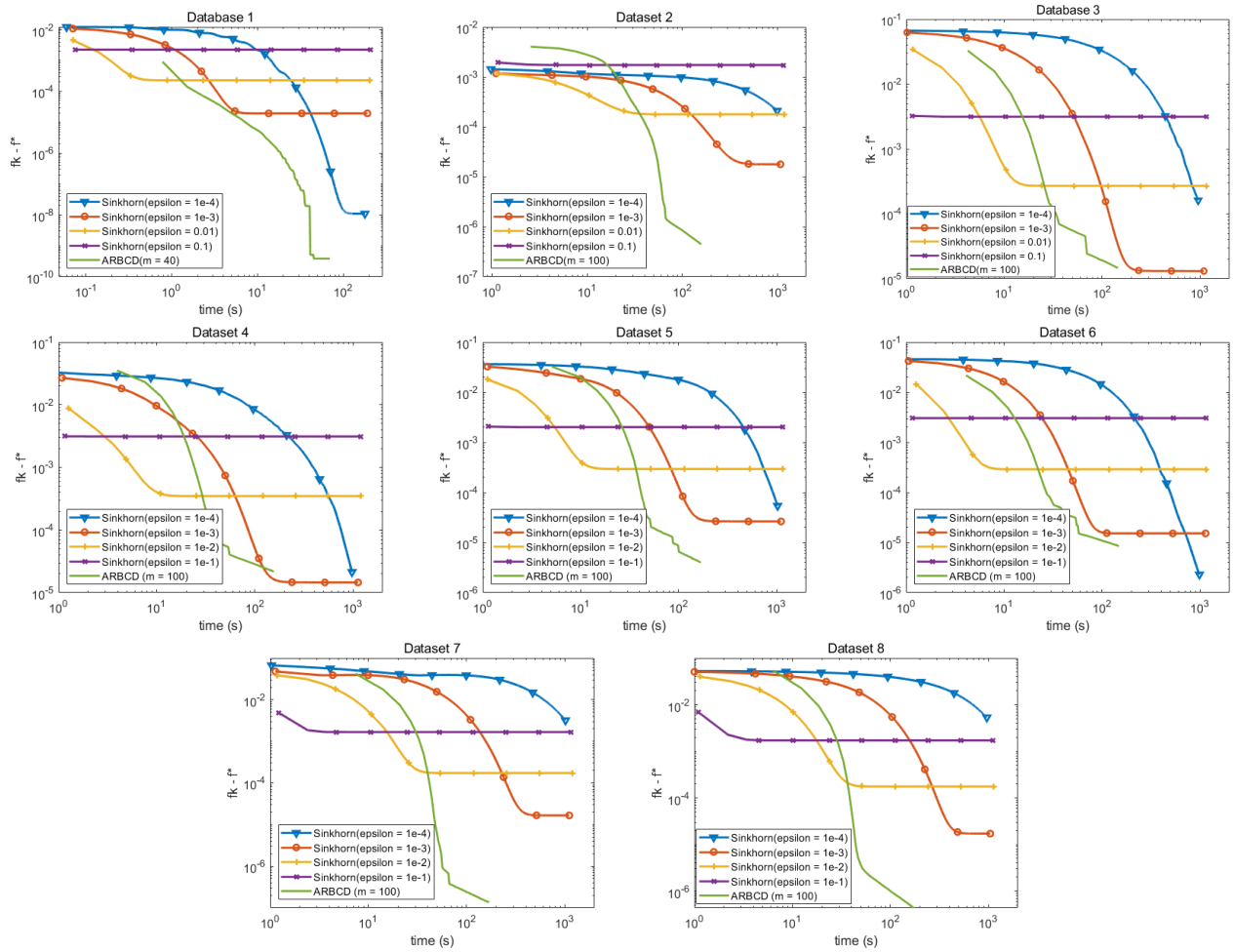

Figure 2. Comparison of algorithms to compute Wasserstein distance

X-axis is the wall-clock time in seconds. Y-axis is the optimality gap $f_k - f^* = c^T x^k - c^T x^*$. This figure shows the trajectory/progress of Algorithm 3: **ARBCD** and **Sinkhorn** with different settings when computing the Wasserstein distance between eight pairs of probability. **ARBCD** is run 5 times in each experiment and the curves showcase the average behavior.

Comments on Figure 2 We can observe the following from Figure 2: although **Sinkhorn** with larger ϵ may converge fast, the solution accuracy is also lower. In fact, this is true for all Sinkhorn-based algorithms because the optimization problem is not exact - it has an extra entropy term. Therefore, the larger γ or ϵ is chosen, the less accurate the solution becomes. On the other hand, when ϵ is set smaller, the convergence of **Sinkhorn** becomes slower. As can be seen from the plots, when $\epsilon = 0.1$ or 0.01, **Sinkhorn** converges faster than **ARBCD**; when $\epsilon = 10^{-3}$, **Sinkhorn** is comparable to **ARBCD**; when $\epsilon = 10^{-4}$, **Sinkhorn** is slower than **ARBCD**. In conclusion, if relatively higher precision is desired, **ARBCD** is comparable with **Sinkhorn**. Moreover, note that here we solve the subproblems in **ARBCD** using `Matlab` built-in solver `linprog`. **ARBCD** can be faster if more efficient subproblem solvers are applied.

5.2. Test on a large-scale OT problem

In this subsection, we generate a pair of 1-dim probability distributions with large discrete support sets ($n = 12800$). For the first distribution, locations of the discrete support $(x^i, i = 1, \ldots, n)$ are evenly aligned between $[-1, 1]$, and their weights/probability are uniformly distributed (i.e., $1/n$). For the other distribution, locations of the discrete support are determined as $\tilde{x}^i = x^{\sigma(i)} + u^i$, where $\sigma(i)$ is a random permutation of $i = 1, \ldots, n$, and u^i is a random variable that conforms to a uniform distribution over $[-0.5, 0.5]$. Weights/probability are determined as $w_i = \frac{\phi(\tilde{x}^i)}{\sum_{i=1}^n \phi(\tilde{x}^i)}$, where $\phi(x)$ is the pdf of the standard normal. The benchmark optimal solution is quickly computed via a closed-form formula for 1-d OT problem ($f^* = 6.236 \times 10^{-3}$). For **ARBCD**, we use the setting $m = \lceil \sqrt{10n} \rceil$, $p = \lfloor m^2/n \rfloor$,

$s = 0.1$ and $T = 10$.

Comments on Figure 3 The figure showcases the average behavior of **ARBCD** within 10000 iterations. It is able to locate a solution such that $(f_k - f^*)/f^* \leq 0.1$. The convergence is linear by observing the trajectory. We also want to point out that Gurobi 10.01 (academic license) runs out of memory on the desktop we use for numerical experiments. Indeed, memory saving is one of the merits that motivate us to consider RBCD methods.

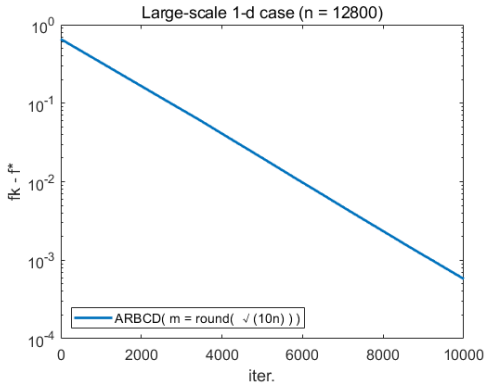

Figure 3. Solving large-scale problem via **ARBCD**
We apply **ARBCD** to solve the large-scale 1-d problem ($n = 12800$). y-axis shows the optimality gap $f_k - f^*$ and x-axis records the iteration number. **ARBCD** is repeated for 3 times and average results are reported.

6. Conclusion

In this paper, we investigate the RBCD method to solve LP problems, including OT problems. In particular, an expected Gauss-Southwell-q rule is proposed to select the working set $\mathcal{I}_k$ at iteration k. It guarantees almost sure convergence and linear convergence rate and is satisfied by all algorithms proposed in this work. We first develop a vanilla RBCD, called **RBCD**$_0$, to solve general LP problems. Then, by examining the structure of the matrix A in the linear system of OT, we refine the working set selection. We use two approaches - diagonal band and submatrix - for constructing $\mathcal{I}_k$ and employ an acceleration technique inspired by the momentum concept to improve the performance of **RBCD**$_0$. In our numerical experiments, we run **ARBCD** against Sinkhorn's algorithm and on a large-scale OT problem. The results show the advantages of our method in finding relatively accurate solutions to OT problems and saving memory. For future work, we plan to extend our method to handle continuous measures and further improve it through parallelization and multiscale strategies, among other approaches.

References

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

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In **International Conference on Machine Learning**, pp. 214–223. PMLR, 2017.

Beck, A. The 2-coordinate descent method for solving double-sided simplex constrained minimization problems. **Journal of Optimization Theory and Applications**, 162 (3):892–919, 2014.

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

Benamou, J. and Brenier, Y. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. **Numerische Mathematik**, 84(3):375–393, 2000.

Berahas, A. S., Bollapragada, R., and Nocedal, J. An investigation of Newton-sketch and subsampled Newton methods. **Optimization Methods and Software**, 35(4): 661–680, 2020.

Blondel, M., Seguy, V., and Rolet, A. Smooth and sparse optimal transport. In **International Conference on Artificial Intelligence and Statistics**, pp. 880–889. PMLR, 2018.

Brenier, Y. Polar factorization and monotone rearrangement of vector-valued functions. **Communications on Pure and Applied Mathematics**, 44(4):375–417, 1991.

Chen, C., He, B., Ye, Y., and Yuan, X. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. **Mathematical Programming**, 155(1):57–79, 2016.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. **Advances in Neural Information Processing Systems**, 26, 2013.

Dvurechensky, P., Gasnikov, A., and Kroshnin, A. 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.

Gasnikov, A. V., Gasnikova, E., Nesterov, Y. E., and Chernov, A. Efficient numerical methods for entropy-linear programming problems. **Computational Mathematics and Mathematical Physics**, 56(4):514–524, 2016.

Gerber, S. and Maggioni, M. Multiscale strategies for computing optimal transport. **Journal of Machine Learning Research**, 18, 08 2017.

Guminov, S., Dvurechensky, P., Tupitsa, N., and Gasnikov, A. On a combination of alternating minimization and Nesterov's momentum. In **International Conference on Machine Learning**, pp. 3886–3898. PMLR, 2021.

Gurbuzbalaban, M., Ozdaglar, A., Parrilo, P. A., and Vanli, N. When cyclic coordinate descent outperforms randomized coordinate descent. **Advances in Neural Information Processing Systems**, 30, 2017.

Haker, S., Zhu, L., Tannenbaum, A., and Angenent, S. Optimal mass transport for registration and warping. **International Journal of Computer Vision**, 60(3):225–240, 2004.

He, B. and Yuan, X. On the $\mathcal{O}(1/n)$ convergence rate of the Douglas–Rachford alternating direction method. **SIAM Journal on Numerical Analysis**, 50(2):700–709, 2012.

Huang, M., Ma, S., and Lai, L. A riemannian block coordinate descent method for computing the projection robust wasserstein distance. In **International Conference on Machine Learning**, pp. 4446–4455. PMLR, 2021.

Jambulapati, A., Sidford, A., and Tian, K. A direct $\tilde{O}(1/\epsilon)$ iteration parallel algorithm for optimal transport. **Advances in Neural Information Processing Systems**, 32, 2019.

Jordan, R., Kinderlehrer, D., and Otto, F. The variational formulation of the Fokker–Planck equation. **SIAM Journal on Mathematical Analysis**, 29(1):1–17, 1998.

Lei, N., Su, K., Cui, L., Yau, S.-T., and Gu, X. D. A geometric view of optimal transportation and generative model. **Computer Aided Geometric Design**, 68:1–21, 2019.

Li, W., Yin, P., and Osher, S. Computations of optimal transport distance with Fisher information regularization. **Journal of Scientific Computing**, 75(3):1581–1595, 2018.

Lin, T., Ho, N., and Jordan, M. I. On the efficiency of entropic regularized algorithms for optimal transport. **Journal of Machine Learning Research**, 23(137):1–42, 2022.

Ling, H. and Okada, K. An efficient earth mover's distance algorithm for robust histogram comparison. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 29(5):840–853, 2007.

Liu, Y., Wen, Z., and Yin, W. A multiscale semi-smooth newton method for optimal transport. **Journal of Scientific Computing**, 91(2):39, 2022.

Lu, Z. and Xiao, L. On the complexity analysis of randomized block-coordinate descent methods. **Mathematical Programming**, 152(1):615–642, 2015.

Necoara, I. and Clipici, D. Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds. **SIAM Journal on Optimization**, 26(1):197–226, 2016.

Necoara, I. and Takáč, M. Randomized sketch descent methods for non-separable linearly constrained optimization. **IMA Journal of Numerical Analysis**, 41(2):1056–1092, 2021.

Necoara, I., Nesterov, Y., and Glineur, F. Random block coordinate descent methods for linearly constrained optimization over networks. **Journal of Optimization Theory and Applications**, 173(1):227–254, 2017.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. **SIAM Journal on Optimization**, 22(2):341–362, 2012.

Otto, F. The geometry of dissipative evolution equations: the porous medium equation. **Taylor & Francis**, 2001.

Peleg, S., Werman, M., and Rom, H. A unified approach to the change of resolution: Space and gray-level. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 11(7):739–742, 1989.

Perrot, M., Courty, N., Flamary, R., and Habrard, A. Mapping estimation for discrete optimal transport. **Advances in Neural Information Processing Systems**, 29, 2016.

Peyré, G. and Cuturi, M. Computational optimal transport. **Foundations and Trends in Machine Learning**, 11(5-6):355–607, 2019.

Polyak, B. T. Introduction to optimization. **Optimization Software, Inc., Publications Division, New York**, 1987.

Qu, Z., Richtárik, P., Takác, M., and Fercoq, O. SDNA: stochastic dual Newton ascent for empirical risk minimization. In **International Conference on Machine Learning**, pp. 1823–1832. PMLR, 2016.

Richtárik, P. and Takáč, M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. **Mathematical Programming**, 144 (1):1–38, 2014.

Richtárik, P. and Takáč, M. Parallel coordinate descent methods for big data optimization. **Mathematical Programming**, 156(1):433–484, 2016.

Schmitzer, B. Stabilized sparse scaling algorithms for entropy regularized transport problems. **SIAM Journal on Scientific Computing**, 41(3):A1443–A1481, 2019.

Sinkhorn, R. A relationship between arbitrary positive matrices and doubly stochastic matrices. **The annals of mathematical statistics**, 35(2):876–879, 1964.

Sun, R. and Ye, Y. Worst-case complexity of cyclic coordinate descent: $\mathcal{O}(n^2)$ gap with randomized version. **Mathematical Programming**, 185(1):487–520, 2021.

Toselli, A. and Widlund, O. **Domain decomposition methods-algorithms and theory**, volume 34. Springer Science & Business Media, 2004.

Tseng, P. and Yun, S. Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. **Journal of Optimization Theory and Applications**, 140(3):513–535, 2009a.

Tseng, P. and Yun, S. A coordinate gradient descent method for nonsmooth separable minimization. **Mathematical Programming**, 117(1):387–423, 2009b.

Tseng, P. and Yun, S. A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. **Computational Optimization and Applications**, 47(2):179–206, 2010.

Villani, C. **Topics in optimal transportation**, volume 58. American Math. Soc., 2021.

Wang, Z., Xin, J., and Zhang, Z. DeepParticle: learning invariant measure by a deep neural network minimizing Wasserstein distance on data generated from an interacting particle method. **Journal of Computational Physics**, pp. 111309, 2022.

Wright, S. **Primal-dual interior-point methods**. SIAM, 1997.

Xie, Y. and Shanbhag, U. V. SI-ADMM: A stochastic inexact ADMM framework for stochastic convex programs. **IEEE Transactions on Automatic Control**, 65(6):2355–2370, 2019.

Xie, Y. and Shanbhag, U. V. Tractable ADMM schemes for computing KKT points and local minimizers for ℓ_0-minimization problems. **Computational Optimization and Applications**, 78(1):43–85, 2021.

A. Previous research on (R)BCD.

BCD and RBCD are well-studied for essentially unconstrained smooth optimization (sometimes allow separable constraints or nonsmooth separable objective functions): (Beck & Tetruashvili, 2013; Gurbuzbalaban et al., 2017; Sun & Ye, 2021) investigate BCD with cyclic coordinate search; (Nesterov, 2012; Lu & Xiao, 2015; Richtárik & Takáč, 2014) study RBCD to address problems with possibly nonsmooth separable objective functions; other related works include theoretical speedup of RBCD ((Richtárik & Takáč, 2016; Necoara & Clipici, 2016)), second-order sketching ((Qu et al., 2016; Berahas et al., 2020)). However, much less is known for their convergence properties when applied to problems with nonseparable nonsmooth functions as summands or coupled constraints. To our best knowledge, no one has ever considered using the RBCD to solve general LP before and the related theoretical guarantees are absent. In (Necoara et al., 2017), the authors studied the RBCD method to tackle problems with a convex smooth objective and coupled linear equality constraints $x_1 + x_2 + \ldots + x_N = 0$; a similar algorithm named random sketch descent method (Necoara & Takáč, 2021) is investigated to solve problems with a general smooth objective and general coupled linear equality constraints $Ax = b$. However, after adding the simple bound constraints $x \geq 0$, the analysis in (Necoara et al., 2017; Necoara & Takáč, 2021) may not work anymore, nor can it be easily generalized. Beck (Beck, 2014) studied a greedy coordinate descent method but focus on a single linear equality constraint and bound constraints. In Paul Tseng and his collaborators' work (Tseng & Yun, 2009a;b; 2010), a block coordinate gradient descent method is proposed to solve linearly constrained optimization problems including general LP. In these works, a Gauss-Southwell-q rule is proposed to guide the selection of the working set in each iteration. Therefore, the working set selected in a deterministic fashion can only be decided after solving a quadratic program with a similar problem size as the original one. In contrast, our proposed mini-batch interior point/RBCD method approach selects the working set through a combination of randomness and low computational cost. Another research direction that addresses separable functions, linearly coupled constraints, and additional separable constraints involves using the alternating direction method of multipliers (ADMM) (Chen et al., 2016; He & Yuan, 2012; Xie & Shanbhag, 2019; 2021). This method updates blocks of primal variables in a Gauss-Seidal fashion and incorporates multiplier updates as well.

B. Proof of Theorem 3.2

Proof. If $d = 0$, then let $\bar{d} = 0$ and $\mathcal{I} = \emptyset$. We have $q(x;\mathcal{I}) = q(x;\mathcal{N}) = 0$. Therefore, both (17) and (18) are satisfied. If $d \neq 0$ and $|\operatorname{supp}(d)| \leq l$, then let $\bar{d} = d$. Thus, $\mathcal{I} = \operatorname{supp}(\bar{d})$ satisfies $|\mathcal{I}| \leq l$ and $q(x;\mathcal{I}) = q(x;\mathcal{N})$. If $|\operatorname{supp}(d)| > l$, then similar to the discussion in Proposition 6.1 in (Tseng & Yun, 2009a), we have that

$$d = d^{(1)} + \ldots + d^{(r)},$$

for some $r \leq |\operatorname{supp}(d)| - l + 1$ and some nonzero $d^{(s)} \in \operatorname{null}(A)$ conformal to d with $|\operatorname{supp}(d^{(s)})| \leq l$, $s = 1, ..., r$. Since $|\operatorname{supp}(d)| \leq N$, we have $r \leq N - l + 1$. Since $Ad^{(s)} = 0$ and $x_i + d_i^{(s)} \geq x_i + d_i \geq 0, \forall s = 1, ..., r$ and $\forall i \in \{i \mid d_i < 0\}$, we have that $x + d^{(s)} \in \mathcal{X}, \forall s = 1, ..., r$. Therefore,

$$q(x;\mathcal{N}) = c^T d = \sum_{s=1}^{r} c^T d^{(s)} \geq r \min_{s=1,\ldots,r} \{c^T d^{(s)}\}.$$

Denote $\bar{s} \in \operatorname{argmin}_{s=1,\ldots,r}\{c^T d^{(s)}\}$ and let $\mathcal{I} = \operatorname{supp}(d^{(\bar{s})})$, then $|\mathcal{I}| \leq l$ and

$$q(x;\mathcal{N}) \geq r c^T d^{(\bar{s})} \geq r q(x;\mathcal{I}) \geq (N - l + 1) q(x;\mathcal{I}).$$

Therefore (17) and (18) hold for this $\mathcal{I}$ and $\bar{d} = d^{(\bar{s})}$. ∎

C. Theory of the linear system in OT

Property of matrix A in (22) A nonzero $d \in \mathbb{R}^N$ is an *elementary vector* of $\operatorname{null}(A)$ if $d \in \operatorname{null}(A)$ and there is no nonzero $d' \in \operatorname{null}(A)$ that is conformal to d and $\operatorname{supp}(d') \neq \operatorname{supp}(d)$. According to the definition in (22), we say that a nonzero matrix X is an *elementary matrix* of $\operatorname{null}(A)$ if $\operatorname{vec}(X)$ is an elementary vector of $\operatorname{null}(A)$. For simplicity, a matrix M^1 being conformal to M^2 means $\operatorname{vec}(M^1)$ being conformal to $\operatorname{vec}(M^2)$ for the rest of this paper. Now we define a set $\mathcal{E}_A$:

$X \in \mathcal{E}_A \subseteq \mathbb{R}^{n \times n} \iff X \neq 0$, and after row and column permutations, X is a multiple of one of the following matrices:

$$E^2 = \begin{pmatrix} 1 & -1 & & & \\ -1 & 1 & & & \\ & & 0 & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix}_{n\times n}, E^3 = \begin{pmatrix} 1 & & -1 & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix}_{n\times n}, ...,$$

$$E^{n-1} = \begin{pmatrix} 1 & & & & -1 \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & & \ddots & \\ & & & -1 & 1 \\ & & & & & 0 \end{pmatrix}_{n\times n}, E^n = \begin{pmatrix} 1 & & & & -1 \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & & \ddots & \\ & & & -1 & 1 \end{pmatrix}_{n\times n}.$$

First, we state a Lemma about $\mathcal{E}_A$, the proof of which is trivial and thus omitted.

Lemma C.1. *Every matrix in $\mathcal{E}_A$ is an elementary matrix of* $\mathrm{null}(A)$.

We show that $\mathcal{E}_A$ characterizes all the elementary matrices.

Theorem C.2. *Given any $D \in \mathbb{R}^{n\times n}$, if* $\mathrm{vec}(D) \in \mathrm{null}(A)$*, then D has a conformal realization, namely:*

$$D = D^{(1)} + D^{(2)} + \ldots + D^{(s)}, \tag{26}$$

where $D^{(1)}, \ldots, D^{(s)}$ are elementary matrices of $\mathrm{null}(A)$ *and $D^{(i)}$ is conformal to D, for all $i = 1, \ldots, s$. In particular, $D^{(i)} \in \mathcal{E}_A$, $\forall i = 1, ..., s$. Therefore, $\mathcal{E}_A$ includes all the elementary matrices of* $\mathrm{null}(A)$.

Proof. First, we show that for any nonzero D such that $\mathrm{vec}(D) \in \mathrm{null}(A)$, there exists $X \in \mathcal{E}_A$ such that X is conformal to D. We prove this by contradiction and induction.

Suppose that no $X \in \mathcal{E}_A$ is conformal to D. Note that $\mathrm{vec}(D) \in \mathrm{null}(A)$ is equivalent to $\sum_{i=1}^m D(i, \bar{j}) = \sum_{j=1}^n D(\bar{i}, j) = 0, \forall \bar{i}, \bar{j}$. WLOG, suppose that $D(1,1) \neq 0$ since we can permute row/column to let $D(1,1) \neq 0$. Further, suppose that $D(1,1) > 0$ since we can otherwise prove the same statement for $-D$. Since $\mathrm{vec}(D) \in \mathrm{null}(A)$, the first column of D must have one negative element. Suppose $D(2,1) < 0$ WLOG. The second row of D must have one positive element, so suppose $D(2,2) > 0$ WLOG. Since no $X \in \mathcal{E}_A$ is conformal to D, we must have $D(1,2) \geq 0$. Therefore, the 2×2 principal matrix of D has the following sign arrangement (after appropriate row/column permutations),

$$\begin{pmatrix} + & +/0 \\ - & + \end{pmatrix},$$

where we use $+$, $+/0$, $-$, and $-/0$ to indicate that the corresponding entry is positive, nonnegative, negative, and nonpositive respectively. If $n = 2$, then the above pattern is impossible, leading to a contradiction. Suppose that $n \geq 3$. For math induction, we assume that after appropriate row/column permutations, the $k \times k$ principal matrix of D has the following sign arrangement ($2 \leq k \leq n-1$),

$$\begin{pmatrix} + & +/0 & +/0 & \ldots & +/0 \\ - & + & +/0 & \ddots & \vdots \\ -/0 & - & + & \ddots & +/0 \\ \vdots & \ddots & \ddots & \ddots & +/0 \\ -/0 & \ldots & -/0 & - & + \end{pmatrix}, \tag{27}$$

i.e., $D(i,j) \geq 0, \forall i \leq j \leq k$; $D_{ij} \leq 0, \forall j < i \leq k$; $D(i,i) > 0, \forall 1 \leq i \leq k$; $D(i+1,i) < 0, \forall 1 \leq i \leq k-1$.

kth column of D needs to have at least one negative element, so suppose $D(k+1,k) < 0$ WLOG. No $X \in \mathcal{E}_A$ is conformal to D, so $D(k+1,i) \leq 0, \forall i = 1, ..., k-1$. Otherwise, let i_0 be the largest index $1, \cdots, k-1$ such that $D(k+1, i_0) > 0$.

Then the submatrix $D(i_0+1:k+1, i_0:k)$ takes the form,

$$\begin{pmatrix} - & + & +/0 & \dots & +/0 \\ -/0 & - & + & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & +/0 \\ -/0 & \dots & -/0 & - & + \\ + & -/0 & \dots & -/0 & - \end{pmatrix}. \tag{28}$$

Moving the first column of (28) to the last (i.e., for D, move the i_0th column and insert it between k and $k+1$th column) and shift the resulting submatrix to the upper left corner through permutation operations, we can see E_{k-i_0+1} is conformal to it.

$(k+1)$th row of D needs to have at least one positive element, so suppose $D(k+1, k+1) > 0$ WLOG. Similar argument shows if there is no $X \in \mathcal{E}_A$ is conformal to D, so $D(i, k+1) \geq 0, \forall i = 1, ..., k$.

Therefore, the $(k+1)\times(k+1)$ principal matrix of D has exactly the same sign pattern as indicated by (27), after appropriate row/column permutations. Note that this is true when $k+1 = n$. However, D itself cannot have the sign pattern as (27) after row/column permutations since the summation of each column/row of D is 0. Contradiction.

Suppose that $X^{(1)} \in \mathcal{E}_A$ and $X^{(1)}$ is conformal to D. Then $X^{(1)}$ can be scaled properly by $\alpha_1 > 0$ such that $|\operatorname{supp}(D - \alpha_1 X^{(1)})| < |\operatorname{supp}(D)|$ and $D - \alpha_1 X^{(1)}$ is conformal to D. Denote $D^{(1)} \triangleq \alpha_1 X^{(1)}$ and $\bar{D}^{(1)} = D - D^{(1)}$. $\bar{D}^{(1)}$ is the new D and we repeat this process. Eventually, we have that the conformal realization (26) holds since $|\operatorname{supp}(D)| \leq n^2$. If D is an elementary matrix, by the conformal realization of D as in (26), D must have the same support with all $D^{(i)} \in \mathcal{E}_A$, $i = 1, ..., s$. Therefore, by definition of $\mathcal{E}_A$, D must be a multiple of the special matrix in the description of $\mathcal{E}_A$ after a certain row/column permutation, and itself is in $\mathcal{E}_A$. Thus $\mathcal{E}_A$ describes all the elementary matrices of $\operatorname{null}(A)$. ∎

Working set selection By analyzing the structure of elementary matrices of $\operatorname{null}(A)$, we will have a better idea of potential directions along which the transport cost is minimized by a large amount. This is supported by the following theorem, where we continue using notations introduced in Section 3.

Theorem C.3. *Consider the linear program (8) where $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$ are defined as in (22) ($M = 2n$, $N = n^2$). Given any $X \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{n\times n}$ such that $\operatorname{vec}(X) \in \mathcal{X}$, and $\operatorname{vec}(D) \in \mathcal{D}(\operatorname{vec}(X); \mathcal{N})$. There exists an elementary matrix $\bar{D}$ of $\operatorname{null}(A)$ conformal to D such that for any set $\mathcal{I} \in \mathcal{N}$ satisfying*

$$\mathcal{I} \supseteq \operatorname{supp}(\operatorname{vec}(\bar{D})),$$

We have

$$q(\operatorname{vec}(X); \mathcal{I}) \leq \left(\frac{1}{n^2-3}\right) q(\operatorname{vec}(X); \mathcal{N}). \tag{29}$$

Proof. Since $\operatorname{vec}(D) \in \mathcal{D}(\operatorname{vec}(X); \mathcal{N})$, $\operatorname{vec}(D) \in \operatorname{null}(A)$. Then based on Theorem C.2, we have the conformal realization:

$$D = D^{(1)} + D^{(2)} + ... + D^{(s)}.$$

Moreover, proof of Theorem C.2 indicates that we can construct this realization with $s \leq n^2 - 3$, because the support of $D^{(i)}$ has cardinality at least 4. Then similar to discussion in Theorem 3.2, we may find $\bar{s} \in \{1, \dots, s\}$ such that $\bar{D} = D^{(\bar{s})}$, $\mathcal{I} \supseteq \operatorname{supp}(\operatorname{vec}(D^{(\bar{s})}))$, and

$$q(\operatorname{vec}(X); \mathcal{N}) \geq (n^2 - 3) q(\operatorname{vec}(X); \mathcal{I}).$$

∎

D. Proof of Theorem 4.1

Proof. Given x^k, Theorem C.3 guarantees that there exists $D^k \in \mathcal{E}_A$ such that if $\mathcal{I}_k \supseteq \mathrm{supp}(\mathrm{vec}(D^k))$, then (29) holds for $\mathcal{I} = \mathcal{I}_k$ and $\mathrm{vec}(X) = x^k$, i.e.,

$$q(x^k; \mathcal{I}_k) \leq \left(\frac{1}{n^2-3}\right) q(x^k; \mathcal{N}). \tag{30}$$

Next, we will estimate the probability that $\mathcal{I}_k \supseteq \mathrm{supp}(\mathrm{vec}(D^k))$ holds.

First, consider the case when $\mathcal{I}_k$ is chosen according to (24). Suppose that after row/column permutations and scaling of D^k, we obtain E^t, $2 \leq t \leq n$. Then after appropriate row and column swapping, D^k can be written as

$$t \left\{ \begin{pmatrix} 0 & & & & & & & & \\ * & * & & & & & & & \\ * & & * & & & & & & \\ & * & & \ddots & & & & & \\ & & * & & * & & & & \\ & & & \ddots & & * & & & \\ & & & & * & * & 0 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 0 \end{pmatrix}_{n \times n} \right. . \tag{31}$$

That is, elements $(2,1)$ and $(3,1)$ are nonzeros; elements (j,j) and $(\mathrm{mod}(j+2,n),j)$ are nonzeros, for all $j = 2, ..., t-1$; elements (t,t) and $(\mathrm{mod}(t+1,n),t)$ are nonzeros; all other elements are zeros. Obviously, support of this matrix is covered by the support of G in (23). Moreover, by moving the whole support in matrix (31) downwards or to the bottom right corner, we can create at least $n(p-2)-1$ more different matrices whose support are all covered by G. These $n(p-2)$ matrices can be obtained by permuting rows and columns of D^k in $n(p-2)$ in different ways. Therefore, the probability that $\mathcal{I}_k$ will cover the support of D^k is at least $\frac{n(p-2)}{(n!)^2}$.

Second, consider the case when $\mathcal{I}_k$ is chosen according to (25). Suppose that after row/column permutations and scaling of D^k, we obtain E^t, $2 \leq t \leq m$. Then $\mathcal{I}_k$ will cover the support of D^k with probability

$$\frac{\binom{n-t}{m-t}^2}{\binom{n}{m}^2} = \left(\frac{(n-t)!/((m-t)!(n-m)!)}{n!/(m!(n-m)!)}\right)^2 = \left(\frac{m!/(m-t)!}{n!/(n-t)!}\right)^2.$$

Then suppose we get E^t, $m+1 \leq t \leq n$ after row/column permutations and rescaling of D^k. Then $\mathcal{I}_k$ will cover the support of D^k with probability 0.

In general, the probability that $\mathcal{I}_k$ cover the support of D^k can be bounded below by $\frac{sn(p-2)}{(n!)^2}$, and we have that

$$\begin{aligned} \mathbb{E}[q(x^k; \mathcal{I}_k) \mid x^k] &= \sum_{\mathrm{supp}(\mathrm{vec}(D^k)) \subseteq \mathcal{I}} q(x^k; \mathcal{I}) P(\mathcal{I}_k = \mathcal{I}) + \sum_{\mathrm{supp}(\mathrm{vec}(D^k)) \nsubseteq \mathcal{I}} q(x^k; \mathcal{I}) P(\mathcal{I}_k = \mathcal{I}) \\ &\leq \left(\frac{1}{n^2-3}\right) q(x^k; \mathcal{N}) P(\mathrm{supp}(\mathrm{vec}(D^k)) \subseteq \mathcal{I}_k) + 0 \\ &\leq \left(\frac{sn(p-2)}{(n^2-3)(n!)^2}\right) q(x^k; \mathcal{N}) \end{aligned}$$

Therefore, the expected Gauss-Southwell-q rule (19) holds with v at least $\frac{sn(p-2)}{(n^2-3)(n!)^2}$. ∎

E. Proof of Lemma 4.3

Proof. Suppose that

$$\log((n^2)!/(n^2-np)!) \geq 2\log(n!) + \log(np)! \tag{32}$$

Then

$$\begin{aligned}
& (n^2)!/(n^2-np)!)/(np)! && \geq (n!)^2 \\
\Longrightarrow \quad & \binom{n^2}{np}/(n!)^2 && \geq 1 \\
\Longrightarrow \quad & \tfrac{\binom{n^2}{np}}{(n!)^2} \cdot \tfrac{sn(p-2)(n^2-np+1)}{n^2-3} && \geq 1 \\
\Longrightarrow \quad & \tfrac{sn(p-2)}{(n^2-3)(n!)^2} && \geq \tfrac{1}{\binom{n^2}{np}(n^2-np+1)},
\end{aligned}$$

where the third inequality holds because $p \leq n/2$ and $n \geq 2/s$. So we only need to prove (32). Note that

$$\begin{aligned}
\log \frac{(n^2)!}{(n^2-np)!} &= \sum_{x=n^2-np+1}^{n^2} \log(x) \geq \int_{n^2-np}^{n^2} (\log x) dx \\
&= n^2 \log(n^2) - n^2 - \big((n^2-np)\log(n^2-np) - n^2 + np\big) \\
&= n^2 \log(n^2) - (n^2-np)\log(n^2-np) - np \\
&\overset{(p=n/K)}{=} 2np \log n + \frac{K-1}{K} \cdot n^2 \cdot \log \frac{K}{K-1} - np \\
&\geq 2np \log n + \frac{2K-3}{2K-2} \cdot np - np. \qquad (33)
\end{aligned}$$

The last inequality holds because $\log(1+x) \geq x - x^2/2$ for $x \in (0,1)$ and $p = n/K$. Meanwhile, right hand side of (32) satisfies the following:

$$\begin{aligned}
& 2\log(n!) + \log(np)! \\
& \leq 2(n+1)\log(n+1) - 2n + (np+1)\log(np+1) - np \\
& \leq 2(n+1)(\log n + \log 2) - 2n + (np+1)(\log(np) + \log 2) - np \\
& = (np+2n+3)\log n + (np+1)\log p + 2(n+1)\log 2 - 2n - np + (\log 2)(np+1) \\
& \overset{(p=\frac{n}{K})}{=} 2np\log n + (2n+4)\log n + (\log 4)n + (\log 2)np + \log 8 - 2n - (1+\log K)np - \log K \\
& \overset{(K \geq \bar{K} \geq 2, n \geq p\bar{K} \geq 6)}{\leq} 2np\log n + (2n+4)\log n + (\log 2)np - (1+\log K)np \qquad (34)
\end{aligned}$$

In order to show (32), we only need to confirm (34) $\leq$ (33). By observation, this is equivalent to

$$\begin{aligned}
& \left(\tfrac{2K-3}{2K-2} + \log\left(\tfrac{K}{2}\right)\right) np && \geq (2n+4)\log n \\
\overset{(p \geq \eta \log n, \bar{K} \leq K)}{\Longleftarrow} \quad & \left(\tfrac{2\bar{K}-3}{2\bar{K}-2} + \log\left(\tfrac{\bar{K}}{2}\right)\right) \eta n && \geq 2n+4 \\
\Longleftrightarrow \quad & \tfrac{4}{\left(\frac{2\bar{K}-3}{2\bar{K}-2} + \log\left(\frac{\bar{K}}{2}\right)\right)\eta - 2} && \leq n.
\end{aligned}$$

The last inequality is assumed. ∎