Operations Research Letters 52 (2024) 107054

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

# Towards optimal running timesfor optimal transport

Jose Blanchet, Arun Jambulapati, Carson Kent *, Aaron Sidford

*Stanford University, 450 Jane Stanford Way Stanford, CA 94305–2004, United States of America*

ARTICLE INFO

*Article history:*
Received 3 August 2021
Received in revised form 7 November 2023
Accepted 13 November 2023
Available online 24 November 2023

*Keywords:*
Optimal transport
Wasserstein distance
Kantorovich problem
Packing LP
Matrix balancing
Maximum cardinality bipartite matching

ABSTRACT

We provide faster algorithms for approximating the optimal transport distance, e.g. earth mover's distance, between two discrete probability distributions on $n$ elements. We present two algorithms which compute couplings between marginal distributions with an expected transportation cost that is within an additive $\epsilon$ of optimal in time $\widetilde{O}(n^2/\epsilon)$; one algorithm is straightforward to parallelize and implementable in depth $\widetilde{O}(1/\epsilon)$. Further, we show that additional improvements on our results must be coupled with breakthroughs in algorithmic graph theory.



## 1. Introduction

In this paper, we consider the *discrete optimal transportation problem*. That is, given two vectors $r$ and $c$ in the $n$-dimensional probability simplex $\Delta^n$, we seek to compute a coupling $X \in \Delta^{n\times n}$ between $r$ and $c$ such that, for a given, non-negative cost function $C : [n] \times [n] \to \mathbb{R}_{\geq 0}$ the expected cost with respect to $X$ is minimized. Due to [24], this problem has a relatively simple expression as a linear program, namely

$$\min_{X\in\mathcal{U}(r,c)} \langle C, X\rangle \text{ where } \mathcal{U}(r,c) := \left\{X \in \mathbb{R}^{n\times n}_{\geq 0} : X\mathbf{1} = r, X^T\mathbf{1} = c\right\}, \tag{1}$$

$\langle\cdot,\cdot\rangle$ is the element-wise inner product, $X$ denotes our *coupling/transportation plan* between $r$ and $c$, and $C \in \mathbb{R}^{n\times n}_{\geq 0}$ is our given cost function expressed as a matrix. In this paper, we focus on computing additive $\epsilon$-optimal solutions to (1), i.e. $\hat{X} \in \mathcal{U}(r,c)$ such that

$$\left\langle C, \widehat{X}\right\rangle \leq \min_{X\in\mathcal{U}(r,c)} \langle C, X\rangle + \epsilon \tag{2}$$

The computation of such solutions, both for discrete distributions $r, c$ and for distributions over more general metric spaces, is playing an increasing role in varied tasks throughout machine learning and statistics. Recent applications in unsupervised learning [8], computer vision [36,11], distributionally-robust optimization [30,10,9], and statistics [37,32] all leverage the ability to compute solutions of (1) or it's continuous analogues. Moreover, these applications have created a need for fast (nearly-linear time) algorithms for (1) in settings where the cost function $C$ is quite general– for instance, in the case where $C$ does not satisfy metric assumptions. Here, we consider nearly-linear time to be any complexity which is of input size $O(n^2)$ after neglecting factors in $\epsilon$ and logarithms in $n$.

As a consequence, recent efforts in the fields of optimization and machine learning [15,5,20,12,17] have focused on establishing nearly-linear time guarantees through the development of new iterative algorithms for (1). This has led to a sequence of increasingly sharper complexity bounds for (1).

In this paper we shed light on the complexity of (1) by giving a pair of *simple* reductions from optimal transport to canonical problems in theoretical computer science, namely *packing linear programming* and *matrix scaling*. Through these reductions we provide new algorithms for (1) with improved asymptotic running times to previous methods. Moreover, we show that these running times cannot be further improved without a major breakthrough in algorithmic graph theory.

### 1.1. Contributions and overview

The contribution of this paper is two-fold. First, we exhibit two separate algorithms for computing an $\epsilon$-approximate solution to

* Corresponding author.
*E-mail addresses:* jose.blanchet@stanford.edu (J. Blanchet), jmblpati@stanford.edu (A. Jambulapati), crkent@stanford.edu (C. Kent), sidford@stanford.edu (A. Sidford).

https://doi.org/10.1016/j.orl.2023.11.007
0167-6377/

(1) in $\widetilde{O}(n^2 \|C\|_{\max}/\epsilon)$ time.[1] This improves upon the following previous best known complexity for this problem of

$$\widetilde{O}\left(\min\left\{\frac{n^{9/4}\sqrt{\|C\|_{\max}}}{\epsilon}, \frac{n^2\|C\|_{\max}}{\epsilon^2}\right\}\right) \quad [17]$$

Additionally, the second of these two algorithms (presented in Section 5), achieves $\widetilde{O}(\|C\|_{\max}/\epsilon)$ parallel depth– an improvement on the previously best known parallel depth (among nearly-linear time algorithms) by a factor of $1/\epsilon$. Beyond differences in the technical machinery underlying these two algorithms, the improved parallelism in the second algorithm is the main complexity theoretic difference between the algorithms presented in Sections 4 and 5.

Each algorithm is derived via a black-box reduction to a different, canonical problem in theoretical computer science that can be solved using powerful iterative methods. The first of our reductions is to a standard *packing linear program* and the second of our reductions is to the *matrix scaling problem*.

**Definition 1** *(Packing Linear Program).* A *packing linear program* is a linear program of the form

$$V_* = \max_{x\in\mathbb{R}^l_{\geq 0}} \left\{d^T x : Ax \leq b\right\} \tag{3}$$

for $b \in \mathbb{R}^m_{\geq 0}$, $d \in \mathbb{R}^l_{\geq 0}$, and $A \in \mathbb{R}^{m\times l}_{\geq 0}$. We say that $x_\epsilon \in \mathbb{R}^m_{\geq 0}$ is an *$\epsilon$-approximate solution* for (3) if $Ax_\epsilon \leq b$ and $d^T x_\epsilon \geq (1-\epsilon)V_*$.

**Definition 2** *(Matrix scaling).* Let $A$ be a non-negative matrix and $r, c \in \mathbb{R}^n_{\geq 0}$ be vectors such that $\sum_{i=1}^n r_i = \sum_{i=1}^n c_i$ and $\|A\|_{\max}$, $\|r\|_{\max}, \|c\|_{\max} \leq 1$. Two non-negative diagonal matrices $X, Y$ are said to *$(r,c)$-scale $A$* if the matrix $B = XAY$ satisfies $B\mathbf{1} = r$ and $B^T\mathbf{1} = c$.

If, instead, $\|B\mathbf{1} - r\|_1 + \left\|B^T\mathbf{1} - c\right\|_1 \leq \epsilon$ we say that $X, Y$ *$\epsilon$-approximately $(r,c)$-scale $A$*. The *matrix scaling problem* is to compute non-negative diagonal matrices $X, Y$ that $\epsilon$-approximately $(r,c)$-scale $A$, provided such matrices exist.

Beyond the gain of a $O(1/\epsilon)$ in complexity, the primary benefit of our results are that these reductions provide a precise characterization of the relationship between the optimal transport problem and breakthroughs in theoretical computer science. In particular, our results link foundational discoveries [2] in positive linear programs to optimal transport and also demonstrate how an orthogonal sequence of techniques from fast Laplacian system solvers and second-order optimization methods [14] provide algorithmic gains for OT.

A secondary contribution of this paper is a precise explanation, in Section 6, of why the above running time of $\widetilde{O}(n^2/\epsilon)$ is bottleneck for the approach of this paper or others [20,5,17,1,7,35]. We provide a reduction from the optimal transport problem to the problem of computing maximum cardinality matching in a bipartite graph (bipartite matching). This shows that any algorithm which improves on the runtime of $\widetilde{O}(n^2/\epsilon)$ yields an algorithm for bipartite matching with an efficient running time that, so far, has only been achieved by using sophisticated techniques of either fast matrix multiplication [21] or dynamic graph data structures [38,13]. This sheds light on the type and sophistication of the tools that are necessary for further improvements (barring another breakthrough and highlights a deep connection between optimal transport and algorithmic graph theory that can be used to understand the limitation of techniques similar to ours.

As a road-map for the reader, after covering previous work in Section 2 and preliminaries in Section 3, in Section 4 we give a reduction from (1) to a packing linear program (LP) and then show how a recently-developed fast solver for packing LPs [3] can be applied to yield our desired sequential run-time. In Section 5 we give a reduction from (1) to matrix scaling and then provide our second algorithm, which obtains both our stated run-time and stated parallel depth. The surprising fact that we can recover the same overall complexity via these very different approaches then motivates Section 6 where we demonstrate the difficulty of further improvements with a reduction from maximum cardinality bipartite matching.

*Concurrent and subsequent work* During the final revision process for this work, a paper [33] offering partially overlapping results was published to ArXiv. This concurrent work constitutes an independent research effort. The result which is shared by [33] and this work is the serial, randomized running time for (1) that is obtained in Sections 4 and 5 of this paper and Theorem 2 of [33]. Indeed, a reduction to packing LPs which is similar to the one given in Section 4 appears in [33]. [33] also appeals to further results concerning packing LP solvers and an additional reduction from (1) to mixed packing and covering LPs in order to provide deterministic and parallel running times for (1) which do not appear in this paper– see Theorem 2 in [33].

Since, the release of [33], the parallel complexity for (1), which appears in Section 5, was added to highlight the difference between the reduction of Section 5 and the reductions obtained in [33]. Indeed, in the case of parallel, randomized running time, the result of Section 5 improves upon [33] by a factor of $1/\epsilon$. Beyond this, edits were made only to improve the presentation and clarify the relationship of this paper to [33] and subsequent work, see Section 7.

Further, since the initial release of this paper there have been advances in obtaining parallel first-order methods matching our best results [22] and obtaining faster running times using more sophisticated optimization and algorithmic graph theory tools [38, 13]. See Section 7 for further discussion.

## 2. Previous work

In this paper, we focus on the case of obtaining nearly-linear running time results for (1). While we could consider solving (1) as a general linear program, any approaches involving the fastest known methods (e.g. [26] via Laplacian system solvers or [27] for generic solvers) would be insufficient for our stated goal since they currently have running time at least $\Omega(n^{2.5})$ for (1).

Outside of such generic solvers and within the scope of previous algorithms which achieve nearly-linear running time (or better) for (1), contemporary literature comprises two veins. The first vein, encompasses those algorithms which impose further conditions on the costs of (1) in order to create a fast computational method for a more restricted subclass of applications. Examples in this line of work are [1,7,35], where nearly-linear run-times are obtained, but at the expense of assuming that the cost matrix $C$ is induced by a metric– or, in the latter case, by a low dimensional $l_p$ metric. For the purposes of this paper, we will only make positivity/boundedness assumptions on our costs (as metric or related assumptions on $C$ can often be violated in practice). Thus, this line of inquiry is less relevant for our efforts.

The second vein of results, however, is more directly related to the algorithm that we will present in Section 5 and stems from the use of entropy-regularization to solve (1). Beginning with the work of [15], this line of research [20,5,12,17] essentially centers

[1] Throughout, we use $\widetilde{O}$ to hide logarithmic factors in $n$ and $\epsilon$ and use $\|C\|_{\max}$ to denote the largest entry of $C$.

**Table 1**
**Running times for computing $\epsilon$-optimal solutions of (1):** In the table, $\widetilde{O}$ hides polylogarithmic factors in $\epsilon, n$. All results except for the interior point method also explicitly hide linear dependence on the norm of the cost matrix $\|C\|_\infty$.

| Algorithm | Running Time | Paper |
|---|---|---|
| Interior Point | $\widetilde{O}\left(n^{2.5}\right)$ | [26] |
| Sinkhorn/RAS | $\widetilde{O}\left(\frac{n^2}{\epsilon^2}\right)$ | [17] |
| APDAGD | $\widetilde{O}\left(\min\left\{\frac{n^{9/4}}{\epsilon}, \frac{n^2}{\epsilon^2}\right\}\right)$ | [17] |
| Box-constrained Newton and Packing LP Reductions | $\widetilde{O}\left(\frac{n^2}{\epsilon}\right)$ | **This paper** |

around applying a particular iterative technique, such as alternating minimiziation (Sinkhorn/RAS) or an accelerated first order method (APDAGD), to solve the dual of an entropy-regularized version of (1). As shown in Table 1, this leads to different approaches for solving (1) in nearly-linear time. It is worth noting that the procedure which appears in Section 5 is tangentially alluded to in [17], but no derivation or concrete running times were given.

## 3. Preliminaries

In this section, we define notation and several, canonical assumptions concerning (1) that will be relevant for the subsequent reductions.

First, we denote the set of non-negative real numbers by $\mathbb{R}_{\geq 0}$, the set of integers $\{1, \ldots, n\}$ by $[n]$, and the $n$ dimensional probability simplex by $\Delta^n = \{x \in \mathbb{R}_{\geq 0} : \sum_{i\in[n]} x_i = 1\}$. Correspondingly, let $\Delta^{n\times n} = \{x \in \mathbb{R}^{n\times n}_{\geq 0} : \mathbf{1}^T X \mathbf{1} = 1\}$ where $\mathbf{1}$ is the all ones vector. Given a set $S \subseteq [n]$ and $r \in \Delta^n$ define $r_{|S}$ to be the conditional distribution induced by $r$ given $S$. Denote the product distribution of $r, c \in \Delta^n$ by $r \otimes c \in \Delta^{n\times n}$.

For $A \in \mathbb{R}^{n\times n}$, we define $\|A\|_{\max}$ to be maximum modulus of any element of $A$. Further, we denote the entry-wise exponential of $A$ by $e^A$ and for $A \in \mathbb{R}^{n\times n}_{\geq 0}$ define

$$H(A) \stackrel{\text{def}}{=} -\sum_{i,j=1}^{n} A_{i,j}\left(\log A_{i,j} - 1\right)$$

to be the (entry-wise) matrix entropy. For two matrices $A, B \in \mathbb{R}^{n\times n}$ we denote the Frobenius inner product by $\langle A, B\rangle = \sum_{i,j\in[n]} A_{i,j}B_{i,j}$.

We will refer to the linear program (1) as the optimal transport problem, Kantorovitch problem, or primal. As is standard, the cost matrix $C \in \mathbb{R}^{n\times n}_{\geq 0}$ has also been assumed to be non-negative and the marginals have been taken to be strictly positive ($r, c > 0$). Note, while we have implicitly assumed that the marginals $r, c \in \Delta^n$ have the same dimension, this has been done for the sake of exposition and the complexities will suitably generalize for $r$ and $c$ of differing dimensions– i.e. our running times will become $\widetilde{O}(mn/\epsilon)$ for $r$ of dimension $m$ and $c$ of dimension $n$.

## 4. Solving by packing LP algorithms

In this section, we give a procedure for computing an $\epsilon$-optimal solution to the optimal transport problem in $\widetilde{O}\left(n^2\|C\|_{\max}/\epsilon\right)$ time. To obtain our reduction, consider solving the linear program:

$$\max_{X\in\mathcal{K}(r,c)} \langle B, X\rangle$$
$$\mathcal{K}(r,c) := \left\{X \in \mathbb{R}^{n\times n}_{+} : X\mathbf{1} \leq r, X^T\mathbf{1} \leq c\right\} \quad B := \|C\|_{\max}\mathbf{1}\mathbf{1}^T - C \tag{4}$$

In other words, we turn the minimization problem (1) into a maximization problem by reversing the sign of $C$ while adding a constant of $\|C\|_{\max}$ to the constraint matrix to keep the new cost matrix, $B$, non-negative. This allows us to just solve under upper bound constraints, rather than both upper and lower bound constraints, on the row and column sums of $X$. Indeed, the new objective encourages using $X$ to make the row and column constraints tight while still minimizing the original cost. Furthermore, since $B$ is an entry-wise, uniform perturbation of $C$ by $\|C\|_{\max}$, (4) will maintain the same set of optimal solutions as (1) while only perturbing the objective function by an additive $\|C\|_{\max}$ term– since $\left\langle X, \mathbf{1}\mathbf{1}^T\right\rangle = \mathbf{1}^T X \mathbf{1} = 1$.

Formally, we first show how to round solutions of (4) to solutions of (1).

**Lemma 1.** *Suppose $X \in \mathbb{R}^{n\times n}_{\geq 0}$ satisfies $X\mathbf{1} \leq r$ and $X^T\mathbf{1} \leq c$. Then, there exists a matrix $D \in \mathbb{R}^{n\times n}_{\geq 0}$ (which can be trivially computed in $O(n^2)$ time) such that $Y = X + D$ satisfies $Y\mathbf{1} = r$ and $Y^T\mathbf{1} = c$.*

**Proof.** Define $e_r := r - X\mathbf{1}$ and $e_c := c - X^T\mathbf{1}$ and observe $e_r, e_c \geq 0$ coordinate-wise and that

$$\|e_r\|_1 = \mathbf{1}^T(r - X\mathbf{1}) = 1 - \mathbf{1}^T X\mathbf{1} = (c^T - \mathbf{1}^T X)\mathbf{1} = \|e_c\|_1$$

Hence, set $D := \frac{1}{\|e_c\|_1} e_r e_c^T$ where, by convention, $D = 0$ if $\|e_c\|_1 = 0$.

It is easy to verify that if $\|e_c\|_1 = 0$, then $Y = X + D$ has the prescribed marginals (row and column sums). Thus, assume that $\|e_c\|_1 \neq 0$. Then,

$$Y\mathbf{1} = \left(X + \frac{1}{\|e_c\|_1} e_r e_c^T\right)\mathbf{1} = X\mathbf{1} + e_r = r$$

and, similarly, $Y^T\mathbf{1} = c$. □

Using this lemma, the main result quickly follows:

**Theorem 1.** *Suppose there exists an oracle $\mathcal{O}$ which computes an $\epsilon'$-approximate solution (see Definition 1) to the packing LP (4) in time $O(\mathcal{T}(m, l, 1/\epsilon'))$. Then, there is an algorithm which computes an $\epsilon$-approximate solution to the optimal transport problem (1) in time*

$$O\left(n^2 + \mathcal{T}\left(n, n, \frac{\|C\|_{\max}}{\epsilon}\right)\right)$$

**Proof.** Let $X_{\epsilon'}$ be the $\epsilon'$-approximate solution obtained by running $\mathcal{O}$ on (4) with approximation parameter $\epsilon' = \epsilon/\|C\|_{\max}$. By Lemma 1, we can compute a $D \in \mathbb{R}^{n\times n}_{\geq 0}$ in $O(n^2)$ time such that $Y = X_{\epsilon'} + D$ is feasible for (1). Hence, denoting the optimal solution to the original transportation problem (1) by $X_*$, we have

$$\langle B, Y\rangle \geq \langle B, X_{\epsilon'}\rangle \geq (1 - \epsilon')\langle B, X_*\rangle$$

where we have used the definition of $\epsilon'$-optimality for $X_{\epsilon'}$ and the fact that $Y \geq X_{\epsilon'}$ entry-wise. Expanding this relationship in $B$ and using the fact that $\mathbf{1}^T Y\mathbf{1} = 1$ and $\mathbf{1}^T X_*\mathbf{1} = 1$, we obtain

$$\|C\|_{\max} - \langle C, Y\rangle \geq \|C\|_{\max} - \langle C, X_*\rangle - \epsilon'\langle B, X_*\rangle$$

Upon rearrangement this yields

$$\langle C, Y\rangle \leq \langle C, X_*\rangle + \epsilon'\langle B, X_*\rangle$$

As $\|B\|_{\max} \leq \|C\|_{\max}$ and $\epsilon' = \epsilon/\|C\|_{\max}$, Hölder's inequality implies that

$$\langle C, Y\rangle \leq \langle C, X_*\rangle + \epsilon$$

Hence, $Y$ is an $\epsilon$-approximate solution of the optimal transportation problem (1). Moreover, it quickly follows that the total time of this procedure is $O\left(n^2 + \mathcal{T}(n, n, \|C\|_{\max}/\epsilon)\right)$. □

Using this reduction, we can now obtain our desired run-time for (1), simply by solving (4) using the current best packing algorithm.

**Theorem 2** *([3]). Given a packing linear program (3), there exists an algorithm that computes an $\epsilon$-approximate solution to (3) in time $\widetilde{O}(m + l + \mathrm{nnz}(A)/\epsilon)$ with high probability.*

With Theorem 2 providing the oracle in Theorem 1, we immediately obtain the following corollary

**Corollary 1.** *There exists an algorithm which computes an $\epsilon$-approximate solution to (1) in time $\widetilde{O}(n^2 \|C\|_{\max}/\epsilon)$ with high probability.*

## 5. Solving by matrix scaling and box-constrained Newton

In this section, we provide another $\widetilde{O}(n^2 \|C\|_\infty/\epsilon)$-time algorithm for computing an $\epsilon$-optimal solution to the optimal transport problem. In comparison to the results of Section 4, the algorithm presented in this section constitutes a different link between the optimal transport problem and recent advances in theoretical computer science (in particular to constrained optimization techniques for matrix balancing). Further, the algorithm not only obtains the sequential run-time of Section 4 but also improves upon it in terms of parallel complexity; the algorithm achieves the fastest known, parallel complexity ($\widetilde{O}(\|C\|_\infty/\epsilon)$ depth) for solving (1) (while preserving total work). Indeed, the approach of Section 4 does not achieve a similar result due to the polynomial depth of [3] in problem dimension. Obtaining an efficient parallel packing algorithm that would yield both the sequential run-time and parallel depth claimed in this section is a key outstanding open problem in positive linear programming. In contrast, the algorithm presented in the section uses certain graph theoretic advances which, potentially make the algorithm more complex and specialized to optimal transport, and enable the improved parallel complexity.

At present, the results of this section and Section 4 provide results of theoretical import. Such results, however, do not necessarily clarify how the empirical performance of the algorithms in these sections would compare. While interesting, such a comparison is outside the scope of this work and, to the authors' knowledge, could be hindered by the lack of an existing numerical implementation for the constrained-Newton step in [14]. Alongside such a comparison, it would be natural to compare these methods to other, canonical techniques for solving optimal transport (such as the Sinkhorn method [15]). However, improving upon the practical performance of state-of-the-art optimal transport methods is key line of work related to this paper that would constitute a distinct, significant, and notable contribution.

As a first step, we will note the following reduction to the matrix scaling problem which appears in prior work [15,5,17]. The optimal transport problem naturally yields an entropy-regularized version

$$\min_{X \in \mathcal{U}(r,c)} \langle C, X\rangle - \eta H(X) \tag{5}$$

whose optimal value of (5) is called the Sinkhorn cost [15]. The namesake refers to the fact that the dual of (5) is equivalent to the problem

$$\min_{x,y\in\mathbb{R}^n} \psi(x,y) \stackrel{\text{def}}{=} \mathbf{1}^T B_{C/\eta}(x,y)\mathbf{1} - r^T x - c^T y \text{ where}$$
$$\left(B_{C/\eta}(x,y)\right)_{ij} \stackrel{\text{def}}{=} e^{x_i + y_j - C_{ij}/\eta} \tag{6}$$

More generally, we will write

$$\min_{x,y\in\mathbb{R}^n} \psi_{A,r,c}(x,y) \stackrel{\text{def}}{=} \mathbf{1}^T M_A(x,y)\mathbf{1} - r^T x - c^T y \text{ where}$$
$$\left(M_A(x,y)\right)_{ij} \stackrel{\text{def}}{=} A_{ij} e^{x_i + y_j} \tag{7}$$

for any non-negative matrix $A \in \mathbb{R}^{n\times n}$ and positive vectors $r, c \in R^n_*$. An optimal solution of (7) gives diagonal matrices which $(r,c)$-scale $A$.

It is known that solving (6) is sufficient to solve the optimal transport problem in the following sense.

**Lemma 2** *(See proof of Theorem 1 in [5]). Let $\widehat{x}, \widehat{y}$ be solutions which satisfy*

$$\left\| B_{C/\eta}(\widehat{x},\widehat{y})\mathbf{1} - r\right\|_1 + \left\| B_{C/\eta}(\widehat{x},\widehat{y})^T\mathbf{1} - c\right\|_1 \le \epsilon$$

*i.e. $\|\nabla\psi(\widehat{x},\widehat{y})\|_1 \le \epsilon$. Then, there exists a projection $\widehat{X}$ of $B_{C/\eta}(\widehat{x},\widehat{y})$ onto $\mathcal{U}(r,c)$ that can be computed in linear-time and work (i.e. $O(n^2)$) and $\widetilde{O}(1)$ depth such that*

$$\left\langle C, \widehat{X}\right\rangle \le \min_{X\in\mathcal{U}(r,c)} \langle C, X\rangle + 2\eta \log n + 4\epsilon \|C\|_\infty$$

Moreover, using Lemma 2 and the following fact, the main reduction of this section is almost immediate.

**Lemma 3.** *Given an instance of (1), there exist a pair of modified, input distributions $\widetilde{r}, \widetilde{c}$ such that $\widetilde{r}_i, \widetilde{c}_i \ge \frac{\epsilon}{2\|C\|_\infty n}$ for all $i \in [n]$ and the solution*

$$\widetilde{X}_* = \min_{X\in\mathcal{U}(\widetilde{r},\widetilde{c})} \langle C, X\rangle \tag{8}$$

*can be extended to an $\epsilon$-approximate solution $\widehat{X}$ of (1) in $O(n^2)$ time/work and $\widetilde{O}(1)$ depth.*

**Proof.** Let

$$S_r = \left\{ i \in [n] : r_i \ge \frac{\epsilon}{2\|C\|_\infty n}\right\} \quad \text{and}$$
$$S_c = \left\{ i \in [n] : c_i \ge \frac{\epsilon}{2\|C\|_\infty n}\right\}$$

and set $\widetilde{r}$ and $\widetilde{c}$ to be the corresponding marginal distributions of $r_{|S_r} \otimes c_{|S_c} \in \Delta^{n\times n}$. Let $\widetilde{X}_*$ be the solution of (8) for such marginals $\widetilde{r}, \widetilde{c}$, denote

$$\mu = \sum_{i\in S_r, j\in S_c} r_i c_j \le 1$$

and set $E = S_r \times S_C \in [n]\times[n]$. For the optimal solution $X_*$ of (1) with marginals $r, c$ and let $X_*^E$ be the distribution induced by conditioning $X_*$ on the set $E$.

The optimality of $\widetilde{X}_*$ implies that

$$\left\langle C, \widetilde{X}_*\right\rangle \le \left\langle C, X_*^E\right\rangle \le \frac{1}{\mu}\langle C, X_*\rangle$$

Further, if we let $\widehat{X}$ be the coupling such that

$$\widehat{X}_{ij} = \begin{cases} \mu\widetilde{X}_{ij} & \text{if } i \in S_r, j \in S_c \\ r_i c_j & \text{otherwise} \end{cases}$$

it is easy to see that $\hat{X}$ has marginals $r$ and $c$ and, by construction of $S_r$ and $S_c$, satisfies

$$\left\langle C, \widehat{X}\right\rangle \le \mu\left\langle C, \widetilde{X}_*\right\rangle + \epsilon \le \langle C, X_*\rangle + \epsilon$$

Clearly, $\widetilde{r}, \widetilde{c}$ and $\hat{X}$ can be constructed in $O(n^2)$ time/work and $\widetilde{O}(1)$ depth. □

**Theorem 3.** *Suppose there exists an oracle $\mathcal{O}$ which computes an $\epsilon'$-approximate solution (see Definition 2) to the matrix scaling problem in time $O\left(\mathcal{T}\left(n, 1/\epsilon', \nu, \xi\right)\right)$ where $\nu = \max_{i,j} 1/A_{ij}$, $\xi = \max_{i\in[n]}(1/\min(r_i, c_i))$, and we let $\mathcal{T}\left(n, 1/\epsilon', \nu, \xi\right) = \infty$ when $\nu = \infty$ or $\xi = \infty$. Then, there is an algorithm which computes an $\epsilon$-approximate solution to the optimal transport problem (1) in time*

$$O\left(n^2 + \mathcal{T}\left(n, \frac{16\|C\|_\infty}{\epsilon}, n^{8\|C\|_\infty/\epsilon}, \frac{4\|C\|_\infty n}{\epsilon}\right)\right)$$

**Proof.** By Lemma 3, we can assume without loss of generality that $\xi \le (4\|C\|_\infty n)/\epsilon$. Set $\eta = \epsilon/(4\log n)$. From [29] and the fact that $e^{-C_{i,j}/\eta}, r_i, c_i > 0$ we know that $e^{-C/\eta}$ is $(r,c)$-scalable. Thus, by running $\mathcal{O}$ on the matrix $e^{-C/\eta}$ with $\epsilon' = 8\|C\|_\infty/\epsilon$, we can produce an approximate $(r,c)$-scaling $B = Xe^{-C/\eta}Y$ such that

$$\|B\mathbf{1} - r\|_1 + \left\|B^T\mathbf{1} - c\right\|_1 \le \epsilon'$$

By Lemma 2, this scaling can be rounded in $O(n^2)$ time to produce a $\hat{X}$ with

$$\left\langle C, \widehat{X}\right\rangle \le \min_{X\in\mathcal{U}(r,c)} \langle C, X\rangle + 2\eta\log n + 4\epsilon'\|C\|_\infty \le \min_{X\in\mathcal{U}(r,c)} \langle C, X\rangle + \epsilon$$

Since

$$\nu = \max_{i,j} \frac{1}{e^{-C_{ij}/\eta}} \le \exp\left(\frac{4\|C\|_\infty \log n}{\epsilon}\right) = n^{4\|C\|_\infty/\epsilon}$$

It follows that this procedure takes

$$O\left(n^2 + \mathcal{T}\left(n, \frac{8\|C\|_\infty}{\epsilon}, n^{4\|C\|_\infty/\epsilon}\right)\right)$$

total time. □

**Corollary 2.** *Suppose there exists an oracle $\mathcal{O}$ which computes an $\epsilon'$-approximate solution to the matrix scaling problem in parallel in $O\left(\mathcal{T}_w\left(n, 1/\epsilon', \nu, \xi\right)\right)$ total work and $\widetilde{O}\left(\mathcal{T}_d\left(n, 1/\epsilon', \nu, \xi\right)\right)$ depth. Then, there is an algorithm which computes an $\epsilon$-approximate solution to the optimal transport problem (1) in*

$$O\left(n^2 + \mathcal{T}_w\left(n, \frac{16\|C\|_\infty}{\epsilon}, n^{8\|C\|_\infty/\epsilon}, \frac{4\|C\|_\infty n}{\epsilon}\right)\right)$$

*work and*

$$\widetilde{O}\left(\mathcal{T}_d\left(n, \frac{16\|C\|_\infty}{\epsilon}, n^{8\|C\|_\infty/\epsilon}, \frac{4\|C\|_\infty n}{\epsilon}\right)\right)$$

*depth.*

Given this reduction between matrix scaling and optimal transport, it remains for us to provide concrete bounds for $\mathcal{T}\left(n, 1/\epsilon', \nu, \xi\right)$ in order to show our desired run-time. To this end, consider the following guarantee given by a currently best algorithm for the matrix scaling problem[2]

**Theorem 4** *(See Theorem 9 in [14]). Suppose that there exists a point $z^*_\epsilon = (x^*_\epsilon, y^*_\epsilon)$ for which $\psi_{A,r,c}(x^*_\epsilon, y^*_\epsilon) - \psi^* \le \epsilon^2/(3n)$ and $\left\|z^*_\epsilon\right\|_\infty \le B$, where $\psi^* = \min_{x,y\in\mathbb{R}^n} \psi_{A,r,c}(x,y)$. Then, there exists a Newton-type algorithm which, with high probability, computes an $\widehat{x}, \widehat{y}$ such that*

$$\|M_A(\widehat{x}, \widehat{y})\mathbf{1} - r\|_2^2 + \left\|M_A(\widehat{x}, \widehat{y})^T\mathbf{1} - c\right\|_2^2 \le \epsilon$$

*in $\widetilde{O}\left(n^2 B\log^2(s_A)\right)$ time– where $s_A$ is the sum of the entries in $A$.*

[2] It should be remarked that similar results to [14] were obtained independently by [4]. We focus on the guarantee stated in [14] since it is more amenable for our use.

The following parallel complexity for the Newton-type algorithm of Theorem 4 is nearly trivial, but not explicitly stated in [14]. Hence, we provide a proof for completeness.

**Theorem 5.** *Suppose that there exists a point $z^*_\epsilon = (x^*_\epsilon, y^*_\epsilon)$ for which $\psi_{A,r,c}(x^*_\epsilon, y^*_\epsilon) - \psi^* \le \epsilon^2/(3n)$ and $\left\|z^*_\epsilon\right\|_\infty \le B$, where $\psi^* = \min_{x,y\in\mathbb{R}^n} \psi_{A,r,c}(x,y)$. Then, there exists a Newton-type algorithm which, with high probability, computes an $\widehat{x}, \widehat{y}$ such that*

$$\|M_A(\widehat{x}, \widehat{y})\mathbf{1} - r\|_2^2 + \left\|M_A(\widehat{x}, \widehat{y})^T\mathbf{1} - c\right\|_2^2 \le \epsilon$$

*in $\widetilde{O}\left(n^2 B\log^2(s_A)\right)$ total work and $\widetilde{O}\left(B\log^2(s_A)\right)$ depth.*

**Proof.** From the proof of Theorem 3.4 in [14], observe that the Newton-type algorithm of Theorem 4 performs $\widetilde{O}\left(B\log^2(s_A)\right)$ sequential (box-constrained) Newton steps on the function

$$f(x,y) = \psi_{A,r,c}(x,y) + \frac{\epsilon^2}{36n^2e^B}\left(\sum_{i\in[n]}\left(e^{x_i} + e^{-x_i} + e^{y_i} + e^{-y_i}\right)\right)$$

Hence, it suffices to show that each Newton iteration can be implemented in $\widetilde{O}\left(n^2\right)$ total work and $\widetilde{O}(1)$ depth.

From the proof of Theorem 5.11 in [14], each Newton-step consists of constructing a vertex sparsifier chain $\left(M^{(1)}, \ldots, M^{(d)}; F_1, \ldots, F_{d-1}\right)$ (see Definition 5.9 in [28]) for the Hessian $\nabla^2 f(x^{(k)}, y^{(k)})$ at the current Newton iterate $x^{(k)}, y^{(k)}$ and then applying the procedure OptimizeChain (see Figure 5.2 in [14]) to $\left(M^{(1)}, \ldots, M^{(d)}; F_1, \ldots, F_{d-1}\right)$ and the gradient $\nabla f(x^{(k)}, y^{(k)})$. Trivially, the Hessian and gradient of $f$ can be computed in $O(n^2)$ work and $\widetilde{O}(1)$ depth. Further, by Theorem 5.10 in [28], we know that a vertex sparsifier chain $\left(M^{(1)}, \ldots, M^{(d)}; F_1, \ldots, F_{d-1}\right)$ of length $d = O(\log n)$ and total sparsity $O(n)$ can be constructed for the Hessian in $O(n^2)$ work and $\widetilde{O}(1)$ depth. Thus, it need only be shown that OptimizeChain can be implemented in $\widetilde{O}(n^2)$ total work and $\widetilde{O}(1)$ depth.

The procedure OptimizeChain applies the subroutines ApproxMapping (see Figure 5.1 in [14]) and FastSolve (see Lemma 5.3 in [14]) to the members $\left(M^{(t)}, F_t\right)$ of the vertex sparsifier chain. The approximate voltage extension subroutine ApproxMapping computes $O(\log(1/\epsilon))$ matrix-vector multiplications using $M^{(t)}$ and disjoint sub-matrices of $\nabla^2 f(x^{(k)}, y^{(k)})$ induced by the vertices $F_t$. Hence, ApproxMapping can be applied to all of the $O(\log n)$ members of the vertex sparsifier in $\widetilde{O}(n^2)$ total work and $\widetilde{O}(1)$ depth.

Further, for each $M^{(t)}$, FastSolve performs $O(1)$ iterations of projected gradient descent on a quadratic function in $M^{(t)}$; where the projection is onto an $\ell_\infty$ ball. Since the gradient of any quadratic in $M^{(t)}$ can be calculated in time equal to the sparsity of $M^{(t)}$ and projection onto an $\ell_\infty$ ball can be implemented simply by truncating coordinates, it follows that FastSolve can be applied to all the members of $\left(M^{(1)}, \ldots, M^{(d)}; F_1, \ldots, F_{d-1}\right)$ in $O(n)$ total work and $\widetilde{O}(1)$ depth. Thus, OptimizeChain can be implemented in $\widetilde{O}(n^2)$ total work and $\widetilde{O}(1)$ depth. □

One would like to immediately apply Theorems 4 and 5 to give the oracles for Theorem 3 and Corollary 2. Unfortunately, there is a mismatch between the $l_1$ guarantee required by Definition 2 and the $l_2$ guarantee in Theorem 4 for which we need the following lemma.

**Lemma 4.** *Suppose that there exists a point $z^*_\epsilon = (x^*_\epsilon, y^*_\epsilon)$ for which $\psi_{A,r,c}(x^*_\epsilon, y^*_\epsilon) - \psi^* \le \epsilon^4/\left(3n^3\right)$ and $\left\|z^*_\epsilon\right\|_\infty \le B$, where $\psi^* = \min_{x,y\in\mathbb{R}^n} \psi_{A,r,c}(x,y)$. Then, there exists a Newton-type algorithm which computes an $\widehat{x}, \widehat{y}$ such that*

$$\|M_A(\widehat{x},\widehat{y})\mathbf{1}-r\|_1+\left\|M_A(\widehat{x},\widehat{y})^T\mathbf{1}-c\right\|_1\leq\epsilon$$

*in time/total work* $\widetilde{O}\left(n^2B\log^2(s_A)\right)$ *and with* $\widetilde{O}\left(B\log^2(s_A)\right)$ *depth.*

**Proof.** Let $\delta=\epsilon^2/(2n)$ be the error tolerance used in Theorem 4 and Theorem 5. Then, by Cauchy-Schwartz and the inequality $(a+b)^2\leq 2(a^2+b^2)$ we have

$$\begin{aligned}&\left(\left\|B_{C/\eta}(\widehat{x},\widehat{y})\mathbf{1}-r\right\|_1+\left\|B_{C/\eta}(\widehat{x},\widehat{y})^T\mathbf{1}-c\right\|_1\right)^2\\&\leq n\left(\left\|B_{C/\eta}(\widehat{x},\widehat{y})\mathbf{1}-r\right\|_2+\left\|B_{C/\eta}(\widehat{x},\widehat{y})^T\mathbf{1}-c\right\|_2\right)^2\\&\leq\epsilon^2\end{aligned}$$

Hence, for such a $\delta$, the algorithms of Theorems 4 and 5 have the same sequential and parallel complexities, respectively, and produce a $\hat{x},\hat{y}$ satisfying

$$\left\|B_{C/\eta}(\widehat{x},\widehat{y})\mathbf{1}-r\right\|_1+\left\|B_{C/\eta}(\widehat{x},\widehat{y})^T\mathbf{1}-c\right\|_1\leq\epsilon\quad\square$$

The final step before combining Theorem 3, Corollary 2, and Lemma 4 is to bound the constant $B$ in Lemma 4 in terms of $\nu=\max_{i,j}1/A_{ij}$ and $\xi=\max_i 1/\min(r_i,c_i)$.

**Lemma 5.** *Suppose that* $A$ *and* $r,c$ *are strictly positive in* (7) *and satisfy the hypotheses of Definition 2, then there exists an optimal solution* $z^*=(x^*,y^*)$ *such that* $\|z^*\|_\infty\leq 2\log(n\nu\xi)$ *where* $\nu,\xi$ *are as defined in Theorem 3.*

**Proof.** From [29] and the fact that $A$ and $r,c$ are strictly positive, there exists an optimal solution $z^*=(x^*,y^*)$. It is easy to see that for any $\alpha\in\mathbb{R}$, $(x^*+\alpha\mathbf{1},y^*-\alpha\mathbf{1})$ is also optimal. Hence, without loss of generality, we can assume that $z^*$ is an optimal solution such that $\min_{i\in[n]}\{x_i^*\}=0$.

Let $m$ be such that $x_m^*=0$. For such a $z^*$, notice that first-order optimality conditions imply that

$$\begin{aligned}&\frac{e^{\max_i\{y_i^*\}}}{\nu}\leq e^{x_m}\sum_{i\in[n]}e^{y_i}A_{m,i}=r_m\leq 1\quad\text{and}\\&r_m=e^{x_m}\sum_{i\in[n]}e^{y_i}A_{m,i}\leq ne^{\max_i\{y_i^*\}}\end{aligned}$$

where we have used that fact that $A_{i,j},r_i\leq 1$ for all $i,j$. This gives that $\max_i\{y_i^*\}\leq\log(\nu)$ and $-\max_i\{y_i^*\}\leq\log(n\xi)$. Additionally, for $k=\arg\max_i\{x_i^*\}$ and $t=\arg\min_i\{y_i^*\}$ we have

$$\begin{aligned}&\frac{e^{x_k+\max_i\{y_i^*\}}}{\nu}\leq e^{x_k}\sum_{i\in[n]}e^{y_i}A_{k,i}=r_k\leq 1\quad\text{and}\\&c_t=e^{y_t}\sum_{i\in[n]}e^{x_i}A_{i,t}\leq ne^{y_t+x_k}\end{aligned}$$

This yields $\max_i\{x_i^*\}\leq\log(n\nu\xi)$ and $-\min_i\{y_i^*\}\leq 2\log(n\nu\xi)$. Putting all these bounds together, it follows that $\left\|z^*\right\|_\infty\leq 2\log(n\nu\xi)$. $\square$

Using Lemma 5, we can now prove our final result.

**Theorem 6.** *Consider an instance of the optimal transport problem* (1)*. There exists an algorithm which computes an* $\epsilon$*-approximate solution with high probability in time*

$$\widetilde{O}\left(\frac{n^2\|C\|_\infty}{\epsilon}\right)$$

*and in parallel with* $\widetilde{O}\left(n^2\|C\|_\infty/\epsilon\right)$ *total work and* $\widetilde{O}\left(\|C\|_\infty/\epsilon\right)$ *depth.*

**Proof.** Consider the Newton-type algorithm of Lemma 4. By Lemma 5, when $A,r,c$ are strictly positive and satisfy the hypotheses of the matrix scaling problem, it follows that $B=O(\log(n\nu\xi))$ and $s_A=O(n^2)$– where $\nu$ and $\xi$ are as defined in Theorem 3. Hence, in this case, the algorithm runs in

$$\widetilde{O}\left(n^2\log(n\nu\xi)\right)\quad\text{time/total work and }\widetilde{O}\left(\log(n\nu\xi)\right)\quad\text{depth}$$

This gives an oracle satisfying the requirements of Theorem 3 and Corollary 2 where, respectively,

$$\widetilde{\mathcal{T}}\left(n,\frac{1}{\epsilon},\nu,\xi\right)\Big)=\widetilde{\mathcal{T}}_w\left(n,\frac{1}{\epsilon},\nu,\xi\right)\Big)=\widetilde{O}\left(n^2\log(n\nu\xi)\right)$$

and

$$\widetilde{\mathcal{T}}_d\left(n,\frac{1}{\epsilon},\nu,\xi\right)\Big)=\widetilde{O}\left(\log(n\nu\xi)\right)$$

Plugging in for $\nu$ and $\xi$, it follows that

$$\widetilde{\mathcal{T}}\left(n,\frac{16\|C\|_\infty}{\epsilon},n^{8\|C\|_\infty/\epsilon},\frac{4\|C\|_\infty n}{\epsilon}\right)\Big)=\widetilde{O}\left(\frac{n^2\|C\|_\infty}{\epsilon}\right)$$

and

$$\widetilde{\mathcal{T}}_d\left(n,\frac{16\|C\|_\infty}{\epsilon},n^{8\|C\|_\infty/\epsilon},\frac{4\|C\|_\infty n}{\epsilon}\right)\Big)=\widetilde{O}\left(\frac{\|C\|_\infty}{\epsilon}\right)$$

giving the result. $\square$

## 6. Reduction for bipartite matching

In this section, we show that further improvements to the $\widetilde{O}(n^2\|C\|_\infty/\epsilon)$ complexity achieved in previous sections must depend on breakthroughs for a long-standing open problem in algorithmic graph theory. Specifically, we show that any additional improvement in the complexity of solving (1) yields a $o(n^{2.5})$ algorithm for *maximum cardinality bipartite matching*. Currently, the only known algorithms that achieve such a complexity are based on sophisticated techniques, e.g. fast matrix multiplication [21]or dynamic graph data structures [38,13].

Note that the identification of this relationship between optimal transport and maximum cardinality bipartite matching does not imply that further complexity improvements for optimal transport are impossible. Indeed, [38] have shown that a complex, breakthrough application of interior point methods and dynamic graph algorithms to solve maximum cardinality bipartite matching in nearly-linear time can also yield a $\widetilde{O}(n^2)$ time algorithm for optimal transport. Rather, this section highlights that additional improvements to our results must be significantly more sophisticated– appeals to standard, iterative/black-box algorithms ([20,5,17,1,7,35,22,6,25,34]) are unlikely to offer additional improvements to our results without further assumptions.

In order to prove this reduction, consider an instance of the maximum cardinality bipartite matching problem where we have an undirected, bipartite graph $G=(V,E)$ such that $V$ is the union of disjoint sets of vertices $L$ and $R$ (each of size $n$) and all edges go exclusively between $L$ and $R$, i.e. $E\subseteq L\times R$. Our goal is to compute a matching, $F\subseteq E$ with

$$\deg_F(i)\stackrel{\text{def}}{=}|\{j\in V\mid\{i,j\}\in F\}|\leq 1,\quad\forall i\in V$$

which maximizes $|F|$. Consider the following lemma

**Lemma 6.** *Given an oracle for computing an $\epsilon$-approximate solution to the optimal transportation problem (1) (under the assumption $\|C\|_\infty = O(1)$) in time $T(n,\epsilon)$, one can compute a maximum cardinality matching $F$ in time $O(T(n,\epsilon)+n^3\epsilon)$.*

**Proof.** We reduce an instance of the bipartite matching problem to optimal transport as follows. Without loss of generality, let $L=[n]$ and $R=[n]$ and let $r=c=\frac{1}{n}\mathbf{1}$. Furthermore, define a cost matrix $C\in\mathbb{R}^{n\times n}$ with $C_{ij}=0$ if $\{i,j\}\in E$ and $C_{ij}=1$ otherwise.

Now, suppose we solve the optimal transport problem corresponding to these inputs to $\epsilon$-accuracy. Define $OPT_T$ to be the optimal value of this transportation problem and let $OPT_M$ to be the optimal value of the maximum cardinality matching in our graph. Clearly, we have computed an $X$ with $X\mathbf{1}=X^T\mathbf{1}=\frac{1}{n}\mathbf{1}$ and such that $\langle C,X\rangle\le OPT_T+\epsilon$. Furthermore, notice that by taking the maximum matching in our graph adding an arbitrary matching between it's unmatched vertices, we can create a perfect matching $Y\in[0,1]^{n\times n}$ such that $\frac{1}{n}Y$ is feasible for our optimal transportation problem and we have $\langle C,Y\rangle=1-OPT_M/n$. Hence $\epsilon$-optimality of $X$ implies that

$$\langle C,X\rangle\le 1+\epsilon-\frac{OPT_M}{n}$$

Hence, as $Z=nX$ is a fractional perfect matching in our graph, this result immediately implies that our oracle for solving optimal transport gives us a fractional perfect matching $Z$ where $\langle C,Z\rangle\le(1+\epsilon)n-OPT_M$. By removing all flow in $Z$ along edges $(i,j)$ where $C_{ij}=1$ (i.e. edges which are non-existent in our original graph) and then rounding the corresponding fractional matching to an actual matching [23] (which can be done in nearly-linear time) we obtain an actual matching $\hat{Z}$ such that

$$\left\langle C,\hat{Z}\right\rangle\le(1+\epsilon)n-OPT_M$$

Hence, $\hat{Z}$ is a matching which has at least $OPT_M-n\epsilon$ edges. Thus, by running augmenting paths [18] on $\hat{Z}$ in $O(n^3\epsilon)$ time (since $G$ is dense) we can find the remaining $n\epsilon$ edges in the maximum matching. This yields an algorithm with complexity

$$O\left(T(n,\epsilon)+n^3\epsilon\right)$$

for finding a maximum matching in a dense graph. □

Using Lemma 6, we see that, if $T(n,\epsilon)=\widetilde{O}(n^2/\epsilon)$, picking $\epsilon=1/\sqrt{n}$ gives a $\widetilde{O}\left(n^{2.5}\right)$ algorithm for matching. For any smaller $T(n,\epsilon)$ (more than log factors of course) an appropriate choice of $\epsilon$ would give a $o(n^{2.5})$ algorithm for maximum cardinality bipartite matching.

## 7. Conclusions and additional subsequent work

In this work, we have demonstrated how to obtain nearly-linear run times for the optimal transportation problem (2) which improve upon the best, previously-known complexities for this problem. Further, we have provided the first parallel method for this problem with $\tilde{O}(\|C\|_\infty/\epsilon)$-depth and nearly linear work- the primary achievement of the alternate approach provided in Section 5. Broadly, these improvements provide utility by linking optimal transport with algorithmic advances in theoretical computer science. Further, our reduction from maximum cardinality bipartite matching shows why, without further assumptions, runtime improvements to our results must be coupled with breakthroughs on a long-standing problem in computer science.

Since the initial release of this work, additional, follow-up work by several of the authors of this paper [22] has replicated the runtimes of this paper (both sequential and parallel). The results of [22] are obtained using an different set of improvements for solving bilinear, minimax optimization problems and apply techniques that are notably different from those in this paper. Ultimately, [22] provides a first-order, iterative scheme with the same complexity requirements as this work.

Additionally, [38] has obtained an improvement to the complexity requirements of this work and [22] for solving the optimal transport problem– achieving an $\widetilde{O}(n^2)$ complexity. This improvement is achieved through a sophisticated use interior point methods and dynamic graph algorithms that is tailored for solving maximum cardinality bipartite matching. Such a result is consistent with the finding of Section 6– that more sophisticated techniques arising from algorithmic graph theory are likely necessary for further improvements to this work. Further, in recent work, [13] provides an additional use of interior point methods and dynamic graph algorithms to solve minimum-cost flow problems that obtain an $O(n^{2+o(1)})$ algorithm for the optimal transport problem– via a similar reduction to the matrix balancing problem presented in Section 5. We note, however that the $\tilde{O}(\|C\|_\infty/\epsilon)$-depth, nearly-linear work algorithm of Section 5 remains the state-of-the-art depth for nearly linear work algorithms– [38] has at least $O(\sqrt{n})$ depth via lower-bounds on self-concordance [31].

Further, a number of papers have considered obtaining better computational performance for more structured problem instances and in high-accuracy regimes. For example, [6,34] have demonstrated that, when the cost function is approximable via kernel decompositions, run times that are sublinear in input size, $o(n^2)$, are achievable for optimal transport– even using standard Sinkhorn-based approaches. These results have been complemented by efforts such as [25,16] which have shown that, when high accuracy solutions to (2) are desired ($\epsilon\ll 1$), classical, combinatorial techniques from network problems can provide methods which, practically, are highly computationally efficient.

Finally, we conclude with some notes on practical considerations in implementing the methods Section 4 and Section 5. While implementing the algorithms for solving packing LPs (the computational task used in Section 4) seems straightforward there is no readily-available, numerical implementation for the algorithm of Section 5. One obstacle in obtaining an efficient implementation of this algorithm is that the Newton-step for the box-constrained Newton algorithm of Section 5, requires the construction of a spectral sparsifier (see Theorem 5.7 of [14]). Therefore, future efforts to perform a numerical study of the method in Section 5 would involve developing a practically efficient method for constructing such spectral sparsifiers compatible with the Newton-step. Recent work [19] has made progress on this for the, arguably, simpler problem of Laplacian system solving; building upon this work for the implementing the method in Section 5 is outside the scope of the paper but an interesting direction for future work.

## Data availability

No data was used for the research described in the article.

## Acknowledgements

J. Blanchet gratefully acknowledges support the Air Force Office of Scientific Research under award number FA9550-20-1-0397, and support from NSF 1915967, 2118199, 2229012, 2312204. A. Sidford was supported in part by a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, a PayPal research award, and a Sloan Research Fellowship.

## References


[1] P.K. Agarwal, R. Sharathkumar, Approximation algorithms for bipartite matching with metric and geometric costs, in: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14, ACM, New York, NY, USA, 2014, pp. 555–564, http://doi.acm.org/10.1145/2591796.2591844.

[2] Z. Allen-Zhu, L. Orecchia, Nearly-linear time positive lp solver with faster convergence rate, in: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, STOC '15, ACM, New York, NY, USA, 2015, pp. 229–236, http://doi.acm.org/10.1145/2746539.2746573.

[3] Z. Allen-Zhu, L. Orecchia, Nearly linear-time packing and covering lp solvers, Math. Program. (Feb 2018), https://doi.org/10.1007/s10107-018-1244-x.

[4] Z. Allen-Zhu, Y. Li, R. Oliveira, A. Wigderson, Much faster algorithms for matrix scaling, in: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), 2017, pp. 890–901.

[5] J. Altschuler, J. Weed, P. Rigollet, Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017, pp. 1964–1974.

[6] J. Altschuler, F.R. Bach, A. Rudi, J. Niles-Weed, Massively scalable Sinkhorn distances via the Nyström method, in: NeurIPS, 2019, pp. 4429–4439, http://dblp.uni-trier.de/db/conf/nips/nips2019.html#AltschulerBRN19.

[7] A. Andoni, A. Nikolov, K. Onak, G. Yaroslavtsev, Parallel algorithms for geometric graph problems, in: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14, ACM, New York, NY, USA, 2014, pp. 574–583, http://doi.acm.org/10.1145/2591796.2591805.

[8] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 70, PMLR, International Convention Centre, Sydney, Australia, 2017, pp. 214–223, http://proceedings.mlr.press/v70/arjovsky17a.html.

[9] J. Blanchet, Y. Kang, Distributionally robust groupwise regularization estimator, in: M.-L. Zhang, Y.-K. Noh (Eds.), Proceedings of the Ninth Asian Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 77, PMLR, 2017, pp. 97–112, http://proceedings.mlr.press/v77/blanchet17a.html.

[10] J. Blanchet, Y. Kang, F. Zhang, K. Murthy, Data-driven optimal transport cost selection for distributionally robust optimization (05 2017).

[11] N. Bonneel, M. van de Panne, S. Paris, W. Heidrich, Displacement interpolation using lagrangian mass transport, ACM Trans. Graph. 30 (6) (2011) 158:1–158:12, https://doi.org/10.1145/2070781.2024192, http://doi.acm.org/10.1145/2070781.2024192.

[12] D. Chakrabarty, S. Khanna, Better and simpler error analysis of the Sinkhorn-Knopp algorithm for matrix scaling, in: R. Seidel (Ed.), 1st Symposium on Simplicity in Algorithms (SOSA 2018), in: OpenAccess Series in Informatics (OASIcs), vol. 61, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018, pp. 4:1–4:11, http://drops.dagstuhl.de/opus/volltexte/2018/8304.

[13] L. Chen, R. Kyng, Y.P. Liu, R. Peng, M.P. Gutenberg, S. Sachdeva, Maximum flow and minimum-cost flow in almost-linear time, arXiv:2203.00671, 2022.

[14] M.B. Cohen, A. Madry, D. Tsipras, A. Vladu, Matrix scaling and balancing via box constrained Newton's method and interior point methods, in: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), Oct 2017.

[15] M. Cuturi, Sinkhorn distances: lightspeed computation of optimal transport, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., USA, 2013, pp. 2292–2300, http://dl.acm.org/citation.cfm?id=2999792.2999868.

[16] Y. Dong, Y. Gao, R. Peng, I. Razenshteyn, S. Sawlani, A study of performance of optimal transport, arXiv preprint, arXiv:2005.01182, 2020.

[17] P. Dvurechensky, A. Gasnikov, A. Kroshnin, Computational optimal transport: complexity by accelerated gradient descent is better than by Sinkhorn's algorithm, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 80, PMLR, Stockholmsmässan, Stockholm, Sweden, 2018, pp. 1366–1375, http://proceedings.mlr.press/v80/dvurechensky18a.html.

[18] D. Fulkerson, An out-of-kilter method for minimal-cost flow problems, J. Soc. Ind. Appl. Math. 9 (1) (1961) 18–27, https://doi.org/10.1137/0109002.

[19] Y. Gao, R. Kyng, D.A. Spielman, Robust and practical solution of laplacian equations by approximate elimination, CoRR, arXiv:2303.00709 [abs], 2023, https://doi.org/10.48550/ARXIV.2303.00709.

[20] A. Genevay, M. Cuturi, G. Peyré, F. Bach, Stochastic optimization for large-scale optimal transport, in: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 29, Curran Associates, Inc., 2016, pp. 3440–3448, http://papers.nips.cc/paper/6566-stochastic-optimization-for-large-scale-optimal-transport.pdf.

[21] O.H. Ibarra, S. Moran, Deterministic and probabilistic algorithms for maximum bipartite matching via fast matrix multiplication, Inf. Process. Lett. 13 (1) (1981) 12–15, https://doi.org/10.1016/0020-0190(81)90142-3.

[22] A. Jambulapati, A. Sidford, K. Tian, A direct $\tilde{O}(1/epsilon)$ iteration parallel algorithm for optimal transport, in: Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, https://proceedings.neurips.cc/paper/2019/file/024d2d699e6c1a82c9ba986386f4d824-Paper.pdf.

[23] D. Kang, J. Payor, Flow rounding, CoRR, arXiv:1507.08139 [abs], 2015, http://arxiv.org/abs/1507.08139.

[24] L. Kantorovitch, On the translocation of masses, Manag. Sci. 5 (1) (1958) 1–4, https://doi.org/10.1287/mnsc.5.1.1.

[25] N. Lahn, D. Mulchandani, S. Raghvendra, A graph theoretic additive approximation of optimal transport, in: Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, https://proceedings.neurips.cc/paper/2019/file/9b07f50145902e945a1cc629f729c213-Paper.pdf.

[26] Y.T. Lee, A. Sidford, Path finding methods for linear programming: solving linear programs in square root rank iterations and faster algorithms for maximum flow, in: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, 2014, pp. 424–433.

[27] Y.T. Lee, A. Sidford, Efficient inverse maintenance and faster algorithms for linear programming, in: Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS), FOCS '15, IEEE Computer Society, Washington, DC, USA, 2015, pp. 230–249.

[28] Y.T. Lee, R. Peng, D.A. Spielman, Sparsified Cholesky solvers for sdd linear systems, arXiv:1506.08204, 2015.

[29] N. Linial, A. Samorodnitsky, A. Wigderson, A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents, in: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98, ACM, New York, NY, USA, 1998, pp. 644–652, http://doi.acm.org/10.1145/276698.276880.

[30] P. Mohajerin Esfahani, D. Kuhn, Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations, Math. Program. 171 (1) (2018) 115–166, https://doi.org/10.1007/s10107-017-1172-1.

[31] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Applied Optimization, Springer US, 2013, https://books.google.com/books?id=2-ElBQAAQBAJ.

[32] V.M. Panaretos, Y. Zemel, Amplitude and phase variation of point processes, Ann. Stat. 44 (2) (2016) 771–812, https://doi.org/10.1214/15-AOS1387.

[33] K. Quanrud, Approximating optimal transport with linear programs, in: J.T. Fineman, M. Mitzenmacher (Eds.), 2nd Symposium on Simplicity in Algorithms (SOSA 2019), in: OpenAccess Series in Informatics (OASIcs), vol. 69, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018, pp. 6:1–6:9, http://drops.dagstuhl.de/opus/volltexte/2018/10032.

[34] M. Scetbon, M. Cuturi, Linear time Sinkhorn divergences using positive features, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 13468–13480, https://proceedings.neurips.cc/paper/2020/file/9bde76f262285bb1eaeb7b40c758b53e-Paper.pdf.

[35] R. Sharathkumar, P.K. Agarwal, A near-linear time epsilon-approximation algorithm for geometric bipartite matching, in: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, STOC '12, ACM, New York, NY, USA, 2012, pp. 385–394, http://doi.acm.org/10.1145/2213977.2214014.

[36] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, L. Guibas, Convolutional Wasserstein distances: efficient optimal transportation on geometric domains, ACM Trans. Graph. 34 (4) (2015) 66, https://doi.org/10.1145/2766963, http://doi.acm.org/10.1145/2766963.

[37] G.J. Székely, M.L. Rizzo, Testing for equal distributions in high dimensions, in: InterStat, 2004.

[38] J. van den Brand, Y. Lee, D. Nanongkai, R. Peng, T. Saranurak, A. Sidford, Z. Song, D. Wang, Bipartite matching in nearly-linear time on moderately dense graphs, in: 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, Los Alamitos, CA, USA, 2020, pp. 919–930, https://doi.ieeecomputersociety.org/10.1109/FOCS46700.2020.00090.