Exponential Convergence of Sinkhorn Under Regularization Scheduling

Jingbang Chen* Li Chen† Yang P. Liu‡ Richard Peng† Arvind Ramaswami‡

Abstract

In 2013, Cuturi [9] introduced the SINKHORN algorithm for matrix scaling as a method to compute solutions to regularized *optimal transport* problems. In this paper, aiming at a better convergence rate for a high accuracy solution, we work on understanding the SINKHORN algorithm under regularization scheduling, and thus modify it with a mechanism that adaptively doubles the regularization parameter η periodically. We prove that such modified version of SINKHORN has an exponential convergence rate as iteration complexity depending on $\log(1/\varepsilon)$ instead of $\varepsilon^{-O(1)}$ from previous analyses [1, 9] in the *optimal transport* problems with integral supply and demand. Furthermore, with cost and capacity scaling procedures, the general *optimal transport* problem can be solved with a logarithmic dependence on $1/\varepsilon$ as well.

1 Introduction

The *optimal transport* (OT) problem asks to compute the minimum cost needed to send supplies to demands. It is formally described as the following linear program:

$$OPT \stackrel{\text{def}}{=} \min_{\mathbf{X}\in\mathbf{U}(\boldsymbol{r},\boldsymbol{c})} \sum_{i\in[n],j\in[m]} \mathbf{Q}_{ij}\mathbf{X}_{ij}, \tag{1.1}$$

where $\mathbf{U}(\boldsymbol{r},\boldsymbol{c})$ is defined as

$$\left\{\mathbf{X}\in\mathbb{R}_+^{n\times m} : \mathbf{X}\mathbf{1}_m = \boldsymbol{r} \quad \text{and} \quad \mathbf{X}^\top\mathbf{1}_n = \boldsymbol{c}\right\}$$

where $\mathbf{Q}$ is the given cost matrix, and $\boldsymbol{r}\in\mathbb{R}_+^n$ and $\boldsymbol{c}\in\mathbb{R}_+^m$ are the demand and supply vectors. The *optimal transport* problem is widely used in machine learning, particularly in areas such as computer vision [10, 18], natural language processing [20], deep learning [26, 33], clustering [16], unsupervised learning [3], and semi-supervised learning [31].

In 1964, Richard Sinkhorn discovered that for any positive square matrix $\mathbf{A}$, there exists a unique doubly stochastic matrix of the form $\mathbf{X} = \mathrm{diag}(\boldsymbol{a})\mathbf{A}\mathrm{diag}(\boldsymbol{b})$ where $\mathrm{diag}(\boldsymbol{a})$ and $\mathrm{diag}(\boldsymbol{b})$ are diagonal matrices with positive entries [29]. $\mathbf{X}$ can be computed using the SINKHORN algorithm. This algorithm normalizes the rows and columns of the matrix in an alternating fashion [30]. In 2013, Cuturi showed that the matrix scaling method can be used to approximate solutions to the *optimal transport* problem with regularization [9]. Such regularization is achieved by adding an entropy regularizer $\eta^{-1}\sum_{i\in[n],j\in[m]}\mathbf{X}_{ij}(\log\mathbf{X}_{ij}-1)$ to the OT objective function. The idea of solving regularized OT was already introduced in 1980s under the name of gravity models [27].

The convergence rate of the SINKHORN algorithm has been the subject of both theoretical and practical analyses in various settings. For instance, it has been proven to have a $\log(1/\varepsilon)$ convergence bound under the Hilbert projective metric [12]. Since the work of [9], several OT algorithms have been developed using the idea of entropic regularization, which have been efficient in practice [4, 14]. However, there are only a few theoretical guarantees for the optimal transport problem directly. [1] shows that with the appropriate choice of parameters, the standard SINKHORN or GREENKHORN algorithm is a near-linear time approximation algorithm for input data of n dimensions, taking $O(n^2||\mathbf{Q}||_\infty^3(\log n)\varepsilon^{-3})$ runtime to give a solution within $OPT+\varepsilon$. However, the convergence rate may be significantly slower when seeking high-accuracy solutions due to the ε^{-3} factor.

To improve the convergence rate in high-accuracy scenarios, we focus on the selection of the regularization parameter η, which balances the desired accuracy and the iteration complexity of the subroutine. One approach uses a series of $\{\eta_k\}_{k\geq 1}$ instead of a single value. In 2019, Bernhard Schmitzer discussed such scheduling in [28], providing a new analysis of the SINKHORN algorithm with regularization scheduling. In our work, we examine the SINKHORN algorithm under this scheduling and explore incorporating it into an adaptive regularization scheme.

1.1 Our Results In this paper, we show that the SINKHORN algorithm with regularization scheduling has an exponential convergence rate. This means that the

*University of Waterloo. This material is based upon work supported by the National Science Foundation under Grant No. 1846218, and by an Natural Sciences and Engineering Research Council of Canada Discovery Grant. Part of this work was done while authors were at the Georgia Institute of Technology.

†Georgia Institute of Technology.

‡Stanford University.

Algorithm	# of Iterations	Comments
Theorem 1.1	$\widetilde{O}\left(\|\boldsymbol{r}\|_1^2 \log(\|\mathbf{Q}\|_\infty/\varepsilon)\right)$	Integral OT
Theorem 1.2	$\text{poly}(n, m, \log(1/\varepsilon), \log \|\|\mathbf{Q}\|\|_\infty, \log \|\|\boldsymbol{r}\|\|_1)$	General OT
[1]	$\widetilde{O}(\|\mathbf{Q}\|_\infty^3/\varepsilon^3)$	Plain Sinkhorn with $\eta = \log n/\varepsilon$
[12]	$O(\exp(\|\mathbf{Q}\|_\infty \log n/\varepsilon)\log(1/\varepsilon))$	Plain Sinkhorn with $\eta = \log n/\varepsilon$
[23]	$\widetilde{O}(n^5 \log(1/\varepsilon))$	Modified row/column scaling

Table 1: Sinkhorn-based algorithms

number of iterations needed to achieve an ε-additive error desired is $\text{poly}\log(1/\varepsilon)$. Additionally, the algorithm has a runtime of $\text{poly}(n, m, \log(1/\varepsilon))$ using row/column scaling operations. The closest similar result to this is the weakly polynomial time matrix scaling algorithm in [23], which uses a more complicated scaling procedure. We provide a table comparing our result with some previous works in Table 1.

For the analysis, we first focus on cases where the demands and supplies are integers bounded by some integer μ. The convergence result is summarized as follows:

Theorem 1.1. (Algorithmic result) *If both the demand vector $\boldsymbol{r}$ and the supply vector $\boldsymbol{c}$ are integral and bounded by μ, i.e. $\mu = \max\{\|\boldsymbol{r}\|_\infty, \|\boldsymbol{c}\|_\infty\}$, Algorithm 1 computes a feasible solution $\mathbf{X}$ to (1.1) with ε-additive error using*

$$O\left(\|\boldsymbol{r}\|_1^2 \log(n\mu) \log\left(\|\mathbf{Q}\|_\infty \|\boldsymbol{r}\|_1/\varepsilon\right)\right)$$

iterations of row/column scaling operations.

Additionally, note that if $\boldsymbol{r}$ is integer and $||r||_1 = O(n)$ (which is relevant in problems like weighted bipartite matching), then Theorem 1.1 gives a stronger guarantee than [23].

We will provide a detailed explanation and proof of our statement in Section 2. Essentially, the proof is based on analyzing the duality gap of the regularized optimal transport problem. Given a good primal-dual solution pair, we show that after doubling the regularization parameter, the duality gap is proportional to $1/\eta$. On the other hand, we also show that a row/column scaling operation reduces the duality gap by roughly $1/\eta$. Both $1/\eta$ terms cancel each other and we can efficiently find a good primal-dual pair w.r.t. to the doubled η.

To achieve $\text{poly}(n, m, \log(1/\varepsilon))$ runtime and to handle non-integral input, we use a cost/capacity scaling scheme commonly used in network flow algorithms (see Appendix C in [8]). The method involves reducing (1.1) to $O(\log(\|\mathbf{Q}\|_\infty)\log(\mu))$ instances each with a dimension of at most $2n^2$ and demand/supply entries at most n^8.

To handle fractional input, we can round each cost, demand, and supply entry to the nearest integral multiple of $\text{poly}(\varepsilon, 1/n, 1/m)$, that is, an integral instance with $\mu = \max\{\|\boldsymbol{r}\|_\infty, \|\boldsymbol{c}\|_\infty\} \cdot \text{poly}(n, m, 1/\varepsilon)$. This allows us to solve the problem in $\text{poly}(n, m, \log(1/\varepsilon))$ time. However, this solution may not be feasible for the original fractional input. But, we can use standard rounding methods to make the solution feasible such as Algorithm 2 in [1]. This process is summarized in the following Lemma.

Theorem 1.2. (Polynomial Runtime via Cost / Capacity Scaling and Rounding) *There is an algorithm that gives a solution $\mathbf{X}$ to (1.1) with ε additive error with $O(\log(\|\mathbf{Q}\|_\infty)\log(\mu))$ calls to Algorithm 1 on integral OT instances with dimension at most n^2 and the total demand/supply at most $O(n^{10})$.*

1.2 Related Work

Optimal Transport Many combinatorial techniques have been introduced to compute the exact solution for certain kinds of OT problems. The Hungarian method invented by Kuhn [19] in 1955 solves the assignment problem (equivalent to OT) in $O(n^3)$ time. In 1991, Gabow and Tarjan gave an $O(n^{2.5}\log(nN))$ time cost/capacity scaling algorithm [13] to solve OT, where N is the largest element in the scaled cost matrix. Using cost/capacity scaling techniques, min-cost flow algorithms such as network simplex also provide exact algorithms for the optimal transport problem in $O(n^3 \log n \log(nN))$ time [11]. There are also studies on

certain kinds of OT problems, such as geometric OT [2] [27]. Additionally, there has been significant recent theoretical work studying the runtime of solving mincost flow, which generalizes OT [5, 6, 8, 21]. These methods rely heavily on second order methods and primitives from graph theory.

Regularization In machine learning, regularization is widely used to resolve various kinds of datasets' heterogeneity [15, 25, 32, 35]. Recently, there have been more works on developing adaptive regularization methods, including deep learning on imbalanced data [7] and learning neural networks [34]. There are also studies on regularization hyperparameter selection [22, 24].

1.3 Notation We use bold lowercase characters such as $\boldsymbol{a}$ to denote vectors. Specially, we use $\mathbf{1}$ or $\mathbf{1}_n$ to denote the all ones vector with proper length. We use bold capital letters (such as $\mathbf{Q}$) as matrices. Specially, we denote the matrix that we are rescaling as $\mathbf{X}$. We denote the inner product of two matrix as $\langle \cdot, \cdot \rangle$, so $\langle \mathbf{X}, \mathbf{Q} \rangle = \sum_{i \in [n], j \in [m]} \mathbf{X}_{ij} \mathbf{Q}_{ij}$. We use the integral vectors $\boldsymbol{r} \in \mathbb{Z}^n$ and $\boldsymbol{c} \in \mathbb{Z}^m$ to denote the desired row and column sums. Note that the matrix $\mathbf{X}$ has row sums $\mathbf{X}\mathbf{1}$ and column sums $\mathbf{X}^\top \mathbf{1}$. We use α_i for $i \in [n]$ and β_j for $j \in [m]$ as the dual variables in our matrix scaling algorithm. As above, η is the regularization parameter.

2 Matrix Scaling with Regularization Scheduling

We propose an algorithm ExpSinkhorn to solve the OT problem to high accuracy. The algorithm maintains a matrix $\mathbf{X}$ to be scaled and a regularization parameter η. It rescales the rows and columns iteratively for this fixed parameter η. When the rows and columns are close enough to scaled, the algorithm doubles η. We ultimately show that this algorithm converges in time depending logarithmically on ε^{-1} (see Theorem 1.1), as opposed to the standard Sinkhorn algorithm requiring time depending polynomially on ε^{-1} to converge [1, 9].

The analysis of our algorithm hinges on understanding the interaction between the ℓ_1 error of the row/column scaling and a *dual objective*. Formally, when the quantities $\|\mathbf{X}\mathbf{1} - \boldsymbol{r}\|_1$ and $\|\mathbf{X}^\top \mathbf{1} - \boldsymbol{c}\|_1$ are small, the algorithm doubles the regularization parameter η. We show that when they are large, then rescaling the rows or columns of $\mathbf{X}$ causes the *dual objective* to significantly improve (see Lemma 2.4). We also prove that when the ℓ_1 errors are small, the duality gap is small (see Lemma 2.3), which bounds the number of iterations (see Lemma 2.5).

We now formally present our matrix scaling algorithm that doubles η over time to give a high accuracy solution to optimal transport.

We will assume throughout this analysis that $\|\boldsymbol{r}\|_\infty, \|\boldsymbol{c}\|_\infty \leq 1$, such that $\mu \boldsymbol{r}, \mu \boldsymbol{c} \in \mathbb{Z}^n$. This is because we scale $\boldsymbol{r}, \boldsymbol{c}$, which are originally in $\mathbb{Z}^n$, down by μ in line 2 of Algorithm 1.

The analysis is based on looking at the dual program of the optimal transport objective:

$$\max_{\alpha_i + \beta_j \leq \mathbf{Q}_{ij} \forall i \in [n], j \in [m]} \sum_{i \in [n]} \boldsymbol{r}_i \alpha_i + \sum_{j \in [m]} \boldsymbol{c}_j \beta_j.$$

The value of this program is also OPT, the same as the value of the optimal transport objective $\min_{\mathbf{X} \geq 0, \mathbf{X} \in \mathbf{U}(\boldsymbol{r}, \boldsymbol{c})} \langle \mathbf{X}, \mathbf{Q} \rangle$ by linear programming duality.

Thus, as long as we can guarantee that the α_i, β_j parameters in Algorithm 1 always satisfy $\alpha_i + \beta_j \leq \mathbf{Q}_{ij}$, then the dual potential $D := \sum_{i \in [n]} \boldsymbol{r}_i \alpha_i + \sum_{j \in [m]} \boldsymbol{c}_j \beta_j \leq OPT$ at all times. We will show these by induction.

Lemma 2.1. (Algorithm invariants) *At all times during an execution of Algorithm 1, we have that $\mathbf{X}_{ij} \leq 1$ for all $i \in [n], j \in [m]$ and $\sum_{i,j} \mathbf{X}_{ij} \leq \|\boldsymbol{r}\|_1$. Hence $\alpha_i + \beta_j \leq \mathbf{Q}_{ij}$ at all times.*

Proof. The "hence" part follows because $\mathbf{X}_{ij} = \exp(\eta(\alpha_i + \beta_j - \mathbf{Q}_{ij}))$, so if $\mathbf{X}_{ij} \leq 1$ then $\alpha_i + \beta_j \leq \mathbf{Q}_{ij}$. Thus, in this proof we focus on showing the claims about $\mathbf{X}$.

We will proceed by induction. We first check that all conditions hold at the start of the algorithm. For the initial choices of η, α_i, β_j we have that

$$\begin{aligned} \mathbf{X}_{ij} &= \exp(10 \|\mathbf{Q}\|_\infty^{-1} \log n (-2 \|\mathbf{Q}\|_\infty + \mathbf{Q}_{ij})) \\ &\leq \exp(-10 \log n) \leq n^{-10}. \end{aligned}$$

Hence, $\sum_{ij} \mathbf{X}_{ij} \leq n^{-8} \leq 1 \leq \|\boldsymbol{r}\|_1$, and $\mathbf{X}_{ij} \leq 1$ for all i, j.

Now, we check that the condition continues to hold after we double η in line 17. Let $\mathbf{X}^{\text{new}}$ be the new matrix after η is doubled. Clearly, $\mathbf{X}_{ij}^{\text{new}} = \mathbf{X}_{ij}^2 \leq \mathbf{X}_{ij}$ because $\mathbf{X}_{ij} \leq 1$ by induction. So $\sum_{ij} \mathbf{X}_{ij}^{\text{new}} \leq \sum_{ij} \mathbf{X}_{ij} \leq \|\boldsymbol{r}\|_1$ by induction, and $\mathbf{X}_{ij}^{\text{new}} = \mathbf{X}_{ij}^2 \leq \mathbf{X}_{ij} \leq 1$.

Finally, we check the conditions after a rescaling step in lines 11, 14. By symmetry, we consider a row rescaling step in line 11. After such a step, we know that $\sum_{j \in [m]} \mathbf{X}_{ij} = \boldsymbol{r}_i$ for all $i \in [n]$. Because $\|\boldsymbol{r}\|_\infty \leq 1$, we deduce that $\mathbf{X}_{ij} \leq 1$ for all i, j and $\sum_{ij} \mathbf{X}_{ij} \leq \|\boldsymbol{r}\|_1$ as desired. The same argument applies to a column rescaling in line 14, if we note that $\|\boldsymbol{c}\|_1 = \|\boldsymbol{r}\|_1$. ∎

Because the dual potentials α_i, β_j are feasible, we know that the dual potential is upper bounded.

Corollary 2.1. (Dual potential upper bound) *During an execution of Algorithm 1, α_i, β_j satisfy $D := \sum_{i \in [n]} \boldsymbol{r}_i \alpha_i + \sum_{j \in [m]} \boldsymbol{c}_j \beta_j \leq OPT$ at all times.*

Algorithm 1: ExpSinkhorn$(\mathbf{Q}, \boldsymbol{r}, \boldsymbol{c}, \epsilon)$ - Solves the optimal transport problem.

Input: A $n \times m$ cost matrix $\mathbf{Q}$.

Output: A $n \times m$ matrix $\mathbf{X}_{ij} \geq 0$ such that $\mathbf{X} \in \mathbf{U}(\boldsymbol{r}, \boldsymbol{c})$ and $\langle \mathbf{X}, \mathbf{Q} \rangle \leq OPT + \epsilon$, where

$$OPT \stackrel{\text{def}}{=} \min_{\mathbf{X}_{ij} \geq 0, \mathbf{X} \in \mathbf{U}(\boldsymbol{r}, \boldsymbol{c})} \langle \mathbf{X}, \mathbf{Q} \rangle.$$

1 $\mu \leftarrow \max\{\max_{i \in [n]} r_i, \max_{j \in [m]} c_j\}$.
2 $\boldsymbol{r} \leftarrow \boldsymbol{r}/\mu, \boldsymbol{c} \leftarrow \boldsymbol{c}/\mu$ ▷ Scale $\boldsymbol{r}, \boldsymbol{c}$ to have $\|\boldsymbol{r}\|_\infty \leq 1, \|\boldsymbol{c}\|_\infty \leq 1$.
3 $\eta \leftarrow 10\|\mathbf{Q}\|_\infty^{-1} \log(n\mu)$. ▷ Starting value of η.
4 $\alpha_i \leftarrow -\|\mathbf{Q}\|_\infty, \beta_j \leftarrow -\|\mathbf{Q}\|_\infty$ for $i \in [n]$, $j \in [m]$. ▷ Dual variable initialization
5 **while** $\eta \leq 4\mu\varepsilon^{-1}\|\boldsymbol{r}\|_1 \log(n\mu)$ **do**
6 $\mathbf{X}_{ij} \leftarrow \exp(\eta(\alpha_i + \beta_j - \mathbf{Q}_{ij}))$. ▷ Initialize matrix to be scaled.
7 **for** $k \geq 0$ **do**
8 $\boldsymbol{a} \leftarrow \mathbf{X}\mathbf{1}$. ▷ Row sums
9 $\boldsymbol{b} \leftarrow \mathbf{X}^T\mathbf{1}$. ▷ Column sums
10 **if** $\|\boldsymbol{a} - \boldsymbol{r}\|_1 > 1/(2\mu)$ **then**
11 $\mathbf{X}_{ij} \leftarrow (\boldsymbol{a}_i/\boldsymbol{r}_i)^{-1}\mathbf{X}_{ij}$ for $1 \leq i \leq n$, $1 \leq j \leq m$ ▷ Row scaling
12 $\alpha_i \leftarrow \alpha_i - \eta^{-1}\log(\boldsymbol{a}_i/\boldsymbol{r}_i)$ for $1 \leq i \leq n$ ▷ Row dual adjustment
13 **else if** $\|\boldsymbol{b} - \boldsymbol{c}\|_1 > 1/(2\mu)$ **then**
14 $\mathbf{X}_{ij} \leftarrow (\boldsymbol{b}_j/\boldsymbol{c}_j)^{-1}\mathbf{X}_{ij}$ for $1 \leq i \leq n$, $1 \leq j \leq m$ ▷ Column scaling
15 $\beta_j \leftarrow \beta_j - \eta^{-1}\log(\boldsymbol{b}_j/\boldsymbol{c}_j)$ for $1 \leq j \leq m$ ▷ Row dual adjustment
16 **else**
17 $\eta \leftarrow 2\eta$ and return to line 5.
18 $\mathbf{X} \leftarrow \mu\mathbf{X}$ ▷ Scale $\mathbf{X}$ back up.
19 Repair the demands routed by $\mathbf{X}$ and return $\mathbf{X}$.

Proof. By Lemma 2.1 we know that $\alpha_i + \beta_j \leq \mathbf{Q}_{ij}$ at all times. As noted above, by linear programming duality

$$D \leq \max_{\alpha_i + \beta_j \leq \mathbf{Q}_{ij} \forall i \in [n], j \in [m]} \sum_{i \in [n]} \boldsymbol{r}_i \alpha_i + \sum_{j \in [m]} \boldsymbol{c}_j \beta_j = OPT.$$

□

The remainder of the analysis requires the following claims. First, we show that the duality gap $OPT - D$ is small when $\|\boldsymbol{r} - \boldsymbol{a}\|_1 \leq 1/(2\mu)$ and $\|\boldsymbol{c} - \boldsymbol{b}\|_1 \leq 1/(2\mu)$ trigger, i.e. line 17. When these do not hold, we show that a rescaling step in lines 12 or 15 causes D to significantly increase. Finally, we will show how to round our approximately scaled solution $\mathbf{X}$ to a feasible point.

Towards this, we show the following useful helper lemma which intuitively shows that an approximately feasible $\mathbf{X}$ "contains" half of a truly feasible solution.

Lemma 2.2. (Containing a feasible solution) *Let $\boldsymbol{r}, \boldsymbol{c}$ be vectors with $\|\boldsymbol{r}\|_1, \|\boldsymbol{c}\|_1 \leq 1$ and $\mu\boldsymbol{r}, \mu\boldsymbol{c} \in \mathbb{Z}^n$. If $\mathbf{X} \geq 0$ satisfies $\mathbf{X}\mathbf{1} = \boldsymbol{r}$ and $\|\mathbf{X}^\top\mathbf{1} - \boldsymbol{c}\|_1 \leq 1/(2\mu)$, then there is a vector $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times m}$ with $0 \leq \widehat{\mathbf{X}}_{ij} \leq \mathbf{X}_{ij}$ for all $i \in [n], j \in [m]$ and $\widehat{\mathbf{X}}\mathbf{1} = \boldsymbol{r}/2$ and $\widehat{\mathbf{X}}^\top\mathbf{1} = \boldsymbol{c}/2$.*

Additionally, such an $\widehat{\mathbf{X}}$ can be found by running any maximum flow algorithm.

Clearly we may swap the roles of $\boldsymbol{r}, \boldsymbol{c}$ above. We state only one case in Lemma 2.2 for brevity.

Proof. Let $\alpha \geq 0$ be maximal so that there exists a $0 \leq \widehat{\mathbf{X}} \leq \mathbf{X}$ such that $\widehat{\mathbf{X}}\mathbf{1} = \alpha\boldsymbol{r}$ and $\widehat{\mathbf{X}}^\top\mathbf{1} = \alpha\boldsymbol{c}$. Let $\mathbf{Y}$ satisfy $\mathbf{Y}\mathbf{1} = \alpha\boldsymbol{r}$ and $\mathbf{Y}^\top\mathbf{1} = \alpha\boldsymbol{c}$. We wish to show that $\alpha \geq 1/2$.

Assume $\alpha < 1/2$ for contradiction, and let $\mathbf{X}^{(1)} = \mathbf{X} - \mathbf{Y}$, so that $\mathbf{X}^{(1)}\mathbf{1} = (1-\alpha)\boldsymbol{r}$ and $\|(\mathbf{X}^{(1)})^\top\mathbf{1} - (1-\alpha)\boldsymbol{c}\|_1 \leq 1/(2\mu)$. Multiplying the previous equations by $(1-\alpha)^{-1}\mu$ on both sides yields that

$$\overline{\mathbf{X}}\mathbf{1} = \mu\boldsymbol{r} \quad \text{and} \quad \left\|\overline{\mathbf{X}}^\top\mathbf{1} - \mu\boldsymbol{c}\right\|_1 \leq \frac{1}{2(1-\alpha)} < 1, \tag{2.2}$$

where $\overline{\mathbf{X}} := (1-\alpha)^{-1}\mu\mathbf{X}^{(1)}$. Note that if there exists $0 \leq \mathbf{Z} \leq \overline{\mathbf{X}}$ such that $0 \leq \mathbf{Z} \leq \overline{\mathbf{X}}$ and $\delta > 0$ with $\mathbf{Z}\mathbf{1} = \delta\boldsymbol{r}$ and $\mathbf{Z}^T\mathbf{1} = \delta\boldsymbol{c}$, then letting $\mathbf{W} = (1-\alpha)\mu^{-1}\mathbf{Z} + \mathbf{Y}$ gives that $\mathbf{W} \leq \mathbf{X}^{(1)} + \mathbf{Y} \leq \mathbf{X}$, and $\mathbf{W}\mathbf{1} = (\alpha + (1-\alpha)\mu^{-1}\delta)\boldsymbol{r}, \mathbf{W}^T\mathbf{1} = (\alpha + (1-\alpha)\mu^{-1}\delta)\boldsymbol{c}$, contradicting the maximality of α. Thus, it suffices to use the fact that both $\mu\boldsymbol{r}$ and $\mu\boldsymbol{c}$ are integral vectors to

construct $0 \le \mathbf{Z} \le \overline{\mathbf{X}}$ and $\delta > 0$ such that $\mathbf{Z}\mathbf{1} = \delta \boldsymbol{r}$ and $\mathbf{Z}^T\mathbf{1} = \delta \boldsymbol{c}$.

Let E be the support of $\overline{\mathbf{X}}$, i.e. $E := \{(i,j) : \overline{\mathbf{X}}_{ij} > 0\}$. For a subset $S \subseteq [n]$, let $N(S) := \{t : \exists s \in S, (s,t) \in E\}$, i.e. the neighborhood of S. By Hall's marriage theorem (for weighted sources and sinks), the subset E supports a flow between $\mu\boldsymbol{r}$ and $\mu\boldsymbol{c}$ as long as for all subsets $S \subseteq [n]$, we have that $\sum_{s\in S}(\mu\boldsymbol{r})_s \le \sum_{t\in N(S)}(\mu\boldsymbol{c})_t$. By the guarantee in (2.2) we know that

$$\begin{aligned}\sum_{s\in S}(\mu\boldsymbol{r})_s &= \sum_{(s,t)\in E}\overline{\mathbf{X}}_{st} \le \sum_{t\in N(S)}\sum_{s\in[n]}\overline{\mathbf{X}}_{st} \\ &\le \sum_{t\in N(S)}(\mu\boldsymbol{c})_t + \|\overline{\mathbf{X}}^\top\mathbf{1} - \mu\boldsymbol{c}\|_1 \\ &< \sum_{t\in N(S)}(\mu\boldsymbol{c})_t + 1.\end{aligned}$$

Because $\sum_{s\in S}(\mu\boldsymbol{r})_s$ and $\sum_{t\in N(S)}(\mu\boldsymbol{c})_t$ are both integral quantities, the previous equation implies that $\sum_{s\in S}(\mu\boldsymbol{r})_s \le \sum_{t\in N(S)}(\mu\boldsymbol{c})_t$ as desired. This shows that there is some $0 \le \mathbf{Z} \le \overline{\mathbf{X}}$ and strictly positive $\delta > 0$ such that $\mathbf{Z}\mathbf{1} = \delta\boldsymbol{r}$ and $\mathbf{Z}^T\mathbf{1} = \delta\boldsymbol{c}$. This completes the proof. ∎

The above lemma lets us bound the duality gap right before we double η, i.e. when line 17 occurs.

LEMMA 2.3. (DUALITY GAP) *Let $D = \sum_{i\in[n]}\alpha_i\boldsymbol{r}_i + \sum_{j\in[m]}\beta_j\boldsymbol{c}_j$. During an execution of Algorithm 1 when line 17 occurs, we have that $OPT - D \le 2\eta^{-1}\|\boldsymbol{r}\|_1\log(n\mu)$.*

Proof. We only handle the case where $\mathbf{X}^\top\mathbf{1} = \boldsymbol{r}$, as the other case is symmetric (recall that $\|\boldsymbol{r}\|_1 = \|\boldsymbol{c}\|_1$). Hence $\mathbf{X}^\top\mathbf{1} = \boldsymbol{r}$.

By Jensen's inequality, we know that

$$\begin{aligned}\sum_{i\in[n],j\in[m]}\mathbf{X}_{ij}\log\mathbf{X}_{ij} &= -\sum_{i\in[n]}\boldsymbol{r}_i\sum_{j\in[m]}\frac{\mathbf{X}_{ij}}{\boldsymbol{r}_i}\log(1/\mathbf{X}_{ij}) \\ &\ge -\sum_{i\in[n]}\boldsymbol{r}_i\log\left(\sum_{j\in[m]}\frac{\mathbf{X}_{ij}}{\boldsymbol{r}_i}\frac{1}{\mathbf{X}_{ij}}\right) \\ &= -\sum_{i\in[n]}\boldsymbol{r}_i\log(m/\boldsymbol{r}_i) \\ &\ge -\|\boldsymbol{r}\|_1\log(n\mu),\end{aligned}$$

because $\boldsymbol{r}_i \ge \mu^{-1}$ for all i, because $\mu\boldsymbol{r} \in \mathbb{Z}^n$ by assumption. Let $\widehat{\mathbf{X}}$ be as constructed in Lemma 2.2. Because $\mathbf{X}_{ij} \le 1$ for all i,j by Lemma 2.1 (so $\log\mathbf{X}_{ij} \le 0$), we can write

$$\begin{aligned}\sum_{i\in[n],j\in[m]}\mathbf{X}_{ij}\log\mathbf{X}_{ij} &\le \sum_{i\in[n],j\in[m]}\widehat{\mathbf{X}}_{ij}\log\mathbf{X}_{ij} \\ &= \eta\sum_{i\in[n],j\in[m]}\widehat{\mathbf{X}}_{ij}(\alpha_i + \beta_j - \mathbf{Q}_{ij}) \\ &= \eta(D/2 - \langle\widehat{\mathbf{X}}, \mathbf{Q}\rangle) \\ &\le \eta(D/2 - OPT/2),\end{aligned}$$

where the final inequality follows because $\widehat{\mathbf{X}}\mathbf{1} = \boldsymbol{r}/2$ and $\widehat{\mathbf{X}}^\top\mathbf{1} = \boldsymbol{c}/2$, hence $\langle\widehat{\mathbf{X}}, \mathbf{Q}\rangle \ge OPT/2$ by the minimality of OPT. Combining the previous two expressions completes the proof. ∎

Now, we prove that if line 17 does not occur, then the dual solution increases significantly.

LEMMA 2.4. (DUAL INCREASE) *Let $\boldsymbol{a} = \mathbf{X}\mathbf{1}$, and consider updating α as in line 12. Then the dual $D := \sum_{i\in[n]}\alpha_i\boldsymbol{r}_i + \sum_{j\in[m]}\beta_j\boldsymbol{c}_j$ increases by at least*

$$\eta^{-1}/10 \cdot \min\{\mu^{-1}, \|\boldsymbol{r}\|_1^{-1}\|\boldsymbol{a} - \boldsymbol{r}\|_1^2\}.$$

Proof. Note the following numerical bound: $-\log(1-t) \ge t + \min\{1/10, t^2/3\}$ for all $t < 1$. By the formula in line 12, the dual increases by

$$\begin{aligned}-\eta^{-1}\sum_{i\in[n]}\boldsymbol{r}_i\log(\boldsymbol{a}_i/\boldsymbol{r}_i) &= \eta^{-1}\sum_{i\in[n]}\boldsymbol{r}_i(-\log(1-(1-\boldsymbol{a}_i/\boldsymbol{r}_i))) \\ &\ge \eta^{-1}\sum_{i\in[n]}\boldsymbol{r}_i\cdot((1-\boldsymbol{a}_i/\boldsymbol{r}_i) \\ &\quad + \min\{(1-\boldsymbol{a}_i/\boldsymbol{r}_i)^2/3, 1/10)\} \\ &= \eta^{-1}\sum_{i\in[n]}\boldsymbol{r}_i\cdot\min\{(1-\boldsymbol{a}_i/\boldsymbol{r}_i)^2/3 \\ &\quad , 1/10\},\end{aligned}$$

because $\|\boldsymbol{a}\|_1 = \sum_{i,j}\mathbf{X}_{ij} \le \|\boldsymbol{r}\|_1$ by Lemma 2.1. If any of the min's in the previous expression evaluate to $1/10$, then the expression is clearly at least $\eta^{-1}\boldsymbol{r}_i/10 \ge \eta^{-1}/10 \cdot \mu^{-1}$, because $\mu\boldsymbol{r}$ is integral. Otherwise, by the Cauchy-Schwarz inequality,

$$\begin{aligned}\eta^{-1}\sum_{i\in[n]}\boldsymbol{r}_i(1-\boldsymbol{a}_i/\boldsymbol{r}_i)^2/3 &= \eta^{-1}/3\cdot\sum_{i\in[n]}(\boldsymbol{a}_i-\boldsymbol{r}_i)^2/\boldsymbol{r}_i \\ &\ge \eta^{-1}/3\cdot\frac{\|\boldsymbol{a}-\boldsymbol{r}\|_1^2}{\|\boldsymbol{r}\|_1},\end{aligned}$$

as desired. This completes the proof. ∎

We can now bound the total number of iterations of the algorithm.

LEMMA 2.5. (ITERATION COUNT) *For integral vectors* $\boldsymbol{r}, \boldsymbol{c} \in \mathbb{Z}^n$, *and* $\mu := \max\{\|\boldsymbol{r}\|_\infty, \|\boldsymbol{c}\|_\infty\}$ *an execution of Algorithm 1 uses at most* $O(\|\boldsymbol{r}\|_1^2 \log(n\mu) \log(\varepsilon^{-1}\|\mathbf{Q}\|_\infty \mu))$ *iterations.*

Proof. After doubling η the duality gap is at most $4\eta^{-1}\|\boldsymbol{r}\|_1 \log(n\mu)$ by Lemma 2.3. If $\|\boldsymbol{a} - \boldsymbol{r}\|_1 \geq 1/(2\mu)$, then the dual increase is at least $\eta^{-1}/10 \cdot \min\{\mu^{-1}, \|\boldsymbol{r}\|_1^{-1}\|\boldsymbol{a} - \boldsymbol{r}\|_1^2\} \geq 1/40 \cdot \eta^{-1}\|\boldsymbol{r}\|_1^{-1}\mu^{-2}$. Hence the number of iterations during a doubling phase is bounded by $\frac{4\eta^{-1}\|\boldsymbol{r}\|_1 \log(n\mu)}{1/40 \cdot \eta^{-1}\|\boldsymbol{r}\|_1^{-1}\mu^{-2}} = O((\mu\|\boldsymbol{r}\|_1)^2 \log(n\mu))$. Additionally, the total number of doubling phases is bounded by $\log((4\mu\varepsilon^{-1}\|\boldsymbol{r}\|_1 \log(n\mu))/(10\|\mathbf{Q}\|_\infty^{-1} \log(n\mu)))$. Thus, the lemma follows (recall that the $\boldsymbol{r}$ in the Lemma statement is really $\mu\boldsymbol{r}$ after scaling). ∎

Finally, we show how to recover a feasible solution from $\mathbf{X}$, and complete the proof of Theorem 1.1.

Proof. [Proof of Theorem 1.1] The iteration complexity bound follows from Lemma 2.5, so it suffices to explain how to round our final solution $\mathbf{X}$ to an accurate solution $\mathbf{Y}$.

To construct $\mathbf{Y}$, let $\widehat{\mathbf{X}}$ be as in Lemma 2.2, and let $\mathbf{Y} = 2\widehat{\mathbf{X}}$. By definition, we know that $\mathbf{Y}\mathbf{1} = 2\widehat{\mathbf{X}}\mathbf{1} = \boldsymbol{r}$, and similarly $\mathbf{Y}^\top\mathbf{1} = \boldsymbol{c}$. To bound the optimality gap of $\mathbf{Y}$, note by the equations in the proof of Lemma 2.3 that $-\|\boldsymbol{r}\|_1 \log(n\mu) \leq \eta(D/2 - \langle\widehat{\mathbf{X}}, \mathbf{Q}\rangle)$, so

$$\begin{aligned}\langle\widehat{\mathbf{X}}, \mathbf{Q}\rangle &\leq \eta^{-1}\|\boldsymbol{r}\|_1 \log(n\mu) + D/2 \\ &\leq \eta^{-1}\|\boldsymbol{r}\|_1 \log(n\mu) + OPT/2,\end{aligned}$$

as $D \leq OPT$ by Corollary 2.1. Hence

$$\begin{aligned}\langle\mathbf{Y}, \mathbf{Q}\rangle = 2\langle\widehat{\mathbf{X}}, \mathbf{Q}\rangle &\leq 2\eta^{-1}\|\boldsymbol{r}\|_1 \log(n\mu) + OPT \\ &\leq OPT + \mu^{-1}\epsilon\end{aligned}$$

by the ending choice of η. Because Algorithm 1 scales everything down by μ, the error in terms of the original objective is ϵ, as desired. $\mathbf{Y}$ can be computed efficiently by calling maximum flow. ∎

3 Reducing to Polynomially Bounded Instances via Scaling

In this section, we will present cost and capacity scaling procedures that reduce solving integral OT to instances with polynomially bounded entries and prove Theorem 1.2.

The following proof can be extended to the case where $n \neq m$ in the OT problem to obtain a $\text{poly}(n, m, \log\frac{1}{\epsilon})$ time algorithm. However, one may find such a proof confusing to read, since m, in addition to being the size of the demand vector, also denotes the number of edges in a min-cost circulation instance. Thus, for ease of exposition, we present the proof for $n = m$.

Instead of OT, we consider the problem of finding minimum cost circulation (MCC) on directed graphs. In the problem of minimum cost circulation, we are given a directed graph $G = (V, E)$ with integral edge costs $\boldsymbol{c} \in \pm[C]^E$ and integral capacities $\boldsymbol{u} \in [U]^E$. The goal is to find a circulation $\boldsymbol{f}$ viewed as a vector over the set of edges E of minimum cost. It is formulated as the following linear program:

$$\min_{\mathbf{B}^\top\boldsymbol{f}=0, 0\leq\boldsymbol{f}\leq\boldsymbol{u}} \boldsymbol{c}^\top\boldsymbol{f} \tag{3.3}$$

where $\mathbf{B}$ is the edge-vertex incidence matrix of G. We use $T_{MCC}(n, m, C, U)$ to be the time to find an integral solution that minimizes (3.3), given a graph with n vertices and m edges. We also define $T_{OT}(n, C, U)$ to denote the time for solving (1.1) for n-dimensional $\boldsymbol{r}, \boldsymbol{c}$ within $1/\text{poly}(n)$-additive error where C is the maximum absolute value of costs and U is the maximum demand or supply entries.

We first show that OT can be reduced to MCC.

LEMMA 3.1. *Given an integral instance of* (1.1), *we have*

$$T_{OT}(n, \|\mathbf{Q}\|_\infty, \mu) = O(n^2) + T_{MCC}(2n, n^2, \|\mathbf{Q}\|_\infty, \mu).$$

Proof. First, we can construct in $O(n^2)$ time a integral matrix $\mathbf{X}^{(0)}$ such that $\mathbf{X}^{(0)}\mathbf{1} = \boldsymbol{r}$ and $\mathbf{X}^{(0)\top}\mathbf{1} = \boldsymbol{c}$. Solving (1.1) is equivalent to finding $\boldsymbol{\Delta}$ that minimizes

$$\min_{\boldsymbol{\Delta}} \sum_{i,j} \mathbf{Q}_{ij}\boldsymbol{\Delta}_{ij}, \text{ such that } \mathbf{X}^{(0)} + \boldsymbol{\Delta} \geq 0, \boldsymbol{\Delta}\mathbf{1} = 0, \text{ and } \boldsymbol{\Delta}^\top\mathbf{1} = 0$$

This corresponds to an MCC problem on a complete bipartite graph with n vertices on each side. The direction and capacity of each edge between the i-th vertex on the left and the j-th vertex on the right depend on the value of $\mathbf{X}^{(0)}_{ij}$. ∎

Next, we show that one can reduce solving (3.3) to few instances where the largest cost in absolute value is $O(n)$. This is done via a revisit of the cost scaling scheme that appears in [8].

LEMMA 3.2. (COST SCALING, LEMMA C.3 [8]) *We have*

$$T_{MCC}(n, m, C, U) = O((T_{MCC}(n, m, 10n, U) + m) \log C)$$

Proof. In Lemma C.8 of [8], we only need the rounded cost differs from the real cost by at most $\varepsilon/2$. Therefore, we only need to round edge costs to the nearest integral multiple of $\varepsilon/2$ within the range $[-\varepsilon, \varepsilon n]$. Thus, the new rounded costs are within $\pm(\varepsilon/2) \cdot [10n]$. ∎

Given the largest cost in absolute value is $O(n)$, we can further reduce (3.3) to few instances whose capacity is poly(n). This is also done via a revisit of the capacity scaling scheme of [8].

LEMMA 3.3. (CAPACITY SCALING, LEMMA C.10 [8]) *We have*

$$T_{MCC}(n,m,10n,U) = O((T_{MCC}(n,m,O(n),O(m^2n^4)) + m)\log U)$$

Proof. In Lemma C.11 of [8], the cycle found via solving unit-capacitated MCC has an approximation ratio $10mn^2$ instead of m^{12} because the cost is bounded by $10n$ instead of m^{10}. Thus, the rounded capacities are integers at most $O((mn^2)^2)$. ∎

Finally, we show that MCC can be solved using the SINKHORN algorithm with regularization scheduling. In particular, we reduce any integral MCC to an integral OT instance. Using the algorithm from Theorem 1.1, we can compute a feasible solution within $OPT + 1/\text{poly}(n)$. Then, we can round the solution to a feasible integral solution without increasing the cost in n^2-time via a cycle cancellation procedure from [17]. The reduction is summarized as follows:

LEMMA 3.4. (SOLVING MCC VIA OT) *We have*

$$T_{MCC}(n,m,C,U) = T_{OT}(\max\{n,m\}, mUC, mU).$$

In addition, the total demand/supply of the reduced OT instance is mU as well.

Proof. Given an instance of (3.3), we construct an integral OT instance as follows: We define the row and column space indexed by V and E respectively. For any $u \in V$, we define its demand $\boldsymbol{r}_u$ to be the weighted incoming degree $\boldsymbol{r}_u = \deg^{in}(u) = \sum_{e=(u,v)} \boldsymbol{u}(e)$. For any edge $e \in E$, we define its supply $\boldsymbol{c}_e$ to be its capacity $\boldsymbol{c}_e = \boldsymbol{u}(e)$. Clearly, both the demand and supply vectors $\boldsymbol{r}$ and $\boldsymbol{c}$ are integers at most $m \cdot U$. The cost matrix $\mathbf{Q} \in \mathbb{R}^{V \times E}$ is defined as follows:

$$\mathbf{Q}_{ue} = \begin{cases} \boldsymbol{c}(e) & \text{if } e = (u,v) \\ 0 & \text{if } e = (v,u) \\ m \cdot U \cdot C & \text{otherwise} \end{cases}$$

Next, we show that solving the OT w.r.t. $\boldsymbol{r}, \boldsymbol{c}$, and $\mathbf{Q}$ we construct is equivalent to solving the given MCC instance. Given any integral OT solution $\mathbf{X}$, we define the flow $\boldsymbol{f}$ as follows:

$$\boldsymbol{f}_e = \mathbf{X}_{ue} \geq 0, \forall e = (u,v)$$

We have $\boldsymbol{c}^\top \boldsymbol{f} = \sum_{u,e} \mathbf{Q}_{ue}\mathbf{X}_{ue}$. To see that $\boldsymbol{f}$ is a circulation, let us look at the net flow at any vertex u

$$\begin{aligned} \boldsymbol{f}^{net}(u) &= \sum_{e=(v,u)} \boldsymbol{f}_e - \sum_{e=(u,v)} \boldsymbol{f}_e \\ &= \sum_{e=(v,u)} \mathbf{X}_{ve} - \sum_{e=(u,v)} \mathbf{X}_{ue} \\ &= \sum_{e=(v,u)} (\boldsymbol{u}(e) - \mathbf{X}_{ue}) - \sum_{e=(u,v)} \mathbf{X}_{ue} \\ &= \deg^{in}(u) - \sum_{e:u\in e} \mathbf{X}_{ue} = \deg^{in}(u) - \boldsymbol{r}_u = 0 \end{aligned}$$

where the 3_{rd} equality comes from that the supply on edge e in the OT instance is exactly $\boldsymbol{u}(e)$, i.e. $\mathbf{X}_{ue} + \mathbf{X}_{ve} = \boldsymbol{c}_e = \boldsymbol{u}(e)$. In addition, $\mathbf{X}_{ue} = 0$ whenever $\mathbf{Q}_{ue} = mUC$ because $\mathbf{X}$ is an optimal solution.

On the other hand, given any feasible circulation $\boldsymbol{f}$ to the MCC instance, we can construct $\mathbf{X}$, a feasible OT solution of identical cost as follows:

$$\mathbf{X}_{ue} = \begin{cases} \boldsymbol{f}_e & \text{if } e = (u,v) \\ \boldsymbol{u}(e) - \boldsymbol{f}_e & \text{if } e = (v,u) \\ 0 & \text{otherwise} \end{cases}$$

Using a similar argument as above, we know that $\boldsymbol{c}^\top \boldsymbol{f} = \sum_{u,e} \mathbf{Q}_{ue}\mathbf{X}_{ue}$, $\mathbf{X}\mathbf{1} = \boldsymbol{r}$, and $\mathbf{X}^\top \mathbf{1} = \boldsymbol{c}$.

Thus, to solve the MCC, we can apply Theorem 1.1 to solve the OT instance with $1/\text{poly}(n)$-additive error in

$$\widetilde{O}\left(\underbrace{\left(\sum_u \boldsymbol{d}^H(u)\right)^2}_{\#\text{ of iterations}} \cdot \underbrace{mn}_{\text{cost per iteration}}\right) = \widetilde{O}((Um)^2 mn)\text{-time.}$$

Then, we round the fractional solution to an integral one without additional error in $O(m^2)$-time (see Section 5 of [17]). Integrity ensures that any integral solution within $OPT + 1/\text{poly}(n)$ is an exact optimal solution. ∎

Given all these Lemmas, we are now ready to prove Theorem 1.2.

Proof. [Proof of Theorem 1.2] Given an integral OT instance, combining Lemma 3.1, Lemma 3.2, Lemma

3.3, and Lemma 3.4 solves the instance in time

$$\begin{aligned}
T_{OT}(n, \|\mathbf{Q}\|_\infty, \mu) &\underset{\text{Lemma 3.1}}{=} O(n^2) + T_{MCC}(2n, n^2, \|\mathbf{Q}\|_\infty, \mu) \\
&\underset{\text{Lemma 3.2}}{=} O\left(n^2 + T_{MCC}\left(2n, n^2,\right.\right. \\
&\quad \left.\left. O(n), \mu\right) \log\left(\|\mathbf{Q}\|_\infty\right)\right) \\
&\underset{\text{Lemma 3.3}}{=} O\left(n^2 + T_{MCC}\left(2n, n^2,\right.\right. \\
&\quad \left.\left. O(n), O(n^8)\right) \log\left(\|\mathbf{Q}\|_\infty\right) \log(\mu)\right) \\
&\underset{\text{Lemma 3.4}}{=} O\left(n^2 + n^4 + T_{OT}\left(n^2,\right.\right. \\
&\quad \left.\left. O(n^{11}), O(n^{10})\right) \log\left(\|\mathbf{Q}\|_\infty\right) \log(\mu)\right).
\end{aligned}$$

This concludes the proof. □

References

[1] Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration". In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 1–3).

[2] David Alvarez-Melis and Nicolo Fusi. "Geometric dataset distances via optimal transport". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21428–21439 (cit. on p. 3).

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223 (cit. on p. 1).

[4] Jean-David Benamou et al. "Iterative Bregman projections for regularized transportation problems". In: *SIAM Journal on Scientific Computing* 37.2 (2015), A1111–A1138 (cit. on p. 1).

[5] Jan van den Brand et al. "Bipartite Matching in Nearly-linear Time on Moderately Dense Graphs". In: *FOCS*. IEEE, 2020, pp. 919–930 (cit. on p. 3).

[6] Jan van den Brand et al. "Minimum cost flows, MDPs, and ℓ_1-regression in nearly linear time for dense instances". In: *STOC*. ACM, 2021, pp. 859–869 (cit. on p. 3).

[7] Kaidi Cao et al. "Heteroskedastic and imbalanced deep learning with adaptive regularization". In: *arXiv preprint arXiv:2006.15766* (2020) (cit. on p. 3).

[8] Li Chen et al. "Maximum flow and minimum-cost flow in almost-linear time". In: *arXiv preprint arXiv:2203.00671* (2022) (cit. on pp. 2, 3, 6, 7).

[9] Marco Cuturi. *Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances*. 2013. arXiv: 1306.0895 [stat.ML] (cit. on pp. 1, 3).

[10] Bharath Bhushan Damodaran et al. "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 447–463 (cit. on p. 1).

[11] Yihe Dong et al. "A Study of Performance of Optimal Transport". In: *ArXiv* abs/2005.01182 (2020) (cit. on p. 2).

[12] Joel Franklin and Jens Lorenz. "On the scaling of multidimensional matrices". In: *Linear Algebra and its applications* 114 (1989), pp. 717–735 (cit. on pp. 1, 2).

[13] Harold N. Gabow and Robert E. Tarjan. "Faster Scaling Algorithms for General Graph Matching Problems". In: *J. ACM* 38.4 (Oct. 1991), pp. 815–853. ISSN: 0004-5411. DOI: 10.1145/115234.115366. URL: https://doi.org/10.1145/115234.115366 (cit. on p. 2).

[14] Aude Genevay et al. "Stochastic optimization for large-scale optimal transport". In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 1).

[15] Ian Goodfellow, Y Bengio, and A Courville. "Regularization for deep learning". In: *Deep learning* (2016), pp. 216–261 (cit. on p. 3).

[16] Nhat Ho et al. "Multilevel clustering via Wasserstein means". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1501–1509 (cit. on p. 1).

[17] Donggu Kang and James Payor. "Flow rounding". In: *arXiv preprint arXiv:1507.08139* (2015) (cit. on p. 7).

[18] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. "Style Transfer by Relaxed Optimal Transport and Self-Similarity". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on p. 1).

[19] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97 (cit. on p. 2).

[20] Matt Kusner et al. "From word embeddings to document distances". In: *International conference on machine learning*. PMLR. 2015, pp. 957–966 (cit. on p. 1).

[21] Yin Tat Lee and Aaron Sidford. "Solving linear programs with Sqrt (rank) linear system solves". In: *arXiv preprint arXiv:1910.08033* (2019) (cit. on p. 3).

[22] Chi-Tat Leung and Tommy WS Chow. "Adaptive regularization parameter selection method for enhancing generalization capability of neural networks". In: *Artificial Intelligence* 107.2 (1999), pp. 347–356 (cit. on p. 3).

[23] Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. "A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents". In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998, pp. 644–652 (cit. on p. 2).

[24] Jelena Luketina et al. "Scalable gradient-based tuning of continuous regularization hyperparameters". In: *International conference on machine learning*. PMLR. 2016, pp. 2952–2960 (cit. on p. 3).

[25] Behnam Neyshabur. "Implicit regularization in deep learning". In: *arXiv preprint arXiv:1709.01953* (2017) (cit. on p. 3).

[26] Gyutaek Oh et al. "Unpaired deep learning for accelerated MRI using optimal transport driven CycleGAN". In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 1285–1296 (cit. on p. 1).

[27] Gabriel Peyré, Marco Cuturi, et al. "Computational optimal transport: With applications to data science". In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607 (cit. on pp. 1, 3).

[28] Bernhard Schmitzer. "Stabilized sparse scaling algorithms for entropy regularized transport problems". In: *SIAM Journal on Scientific Computing* 41.3 (2019), A1443–A1481 (cit. on p. 1).

[29] Richard Sinkhorn. "A relationship between arbitrary positive matrices and doubly stochastic matrices". In: *The annals of mathematical statistics* 35.2 (1964), pp. 876–879 (cit. on p. 1).

[30] Richard Sinkhorn and Paul Knopp. "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348 (cit. on p. 1).

[31] Justin Solomon et al. "Wasserstein propagation for semi-supervised learning". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 306–314 (cit. on p. 1).

[32] Yingjie Tian and Yuqi Zhang. "A comprehensive survey on regularization strategies in machine learning". In: *Information Fusion* 80 (2022), pp. 146–166 (cit. on p. 3).

[33] Jingwei Zhang, Tongliang Liu, and Dacheng Tao. "An Optimal Transport Analysis on Generalization in Deep Learning". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021) (cit. on p. 1).

[34] Han Zhao et al. "Learning Neural Networks with Adaptive Regularization". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/2281f5c898351dbc6dace2ba201e7948-Paper.pdf (cit. on p. 3).

[35] Dixian Zhu et al. "A machine learning approach for air quality prediction: Model regularization and optimization". In: *Big data and cognitive computing* 2.1 (2018), p. 5 (cit. on p. 3).