# Optimal Transport
# Numerical Methods with Entropic Regularization
# Project Report

Ekaterina Filimoshina

## Contents

# 1 Introduction

# 2 Optimal Transport Problem

## 2.1 Notation and definitions

**Definition 1 (Probability simplex)** *A standard probability simplex is a set of the form*

$$S_n(1) := \{s \in \mathbb{R}_n^+ : \langle s, 1_n \rangle := \sum_{i=1}^{n} s_i = 1\}. \tag{1}$$

**Definition 2 (Coupling matrix (Transportation plan))** *A coupling matrix is a matrix $\pi \in \mathbb{R}_+^{n \times n}$ with elements $\pi_{ij}$ prescribing the amount of mass moved from the source point $i$ to the target point $j$.*

Suppose $p \in S_n(1)$ and $q \in S_n(1)$ are two discrete probability measures.

**Definition 3 (Transportation polytope)** *Transportation polytope is a set (a convex hull of finite set of matrices) of all coupling matrices of size $n \times n$:*

$$U(p,q) := \{\pi \in \mathbb{R}_+^{n \times n} : \quad \pi 1_n = p, \quad \pi^T 1_n = q\} \tag{2}$$

$$= \{\pi \in \mathbb{R}_+^{n \times n} : \quad \sum_{j=1}^{n} \pi_{ij} = p_i, \quad \forall i = 1, \ldots, n; \quad \sum_{i=1}^{n} \pi_{ij} = q_j, \quad \forall j = 1, \ldots, n\}. \tag{3}$$

**Definition 4 (Transportation cost matrix)** *A transportation cost matrix $C \in \mathbb{R}_+^{n \times n}$ is a symmetric matrix with elements $C_{ij}$ describing cost of transportation of a unit of mass from the source point $i$ to the target point $j$.*

## 2.2 Monge–Kantorovich problem and dual problem

**Monge–Kantorovich problem** is a problem of finding a transportation plan $\pi$ that minimizes the total cost of transportation of the distribution $p$ to the distribution $q$:

$$\mathrm{L}_C(p,q) := \min_{\pi \in U(p,q)} \langle C, \pi \rangle := \min_{\pi \in U(p,q)} \sum_{i,j} C_{ij} \pi_{ij}, \tag{4}$$

where $\langle C, \pi \rangle$ is a Frobenius inner product of $C$ and $\pi$. The problem (4) is a linear programming problem and can be solved by the corresponding methods.

The problem (4) can be naturally paired with the following dual problem, which is a constrained concave maximization problem.

**Proposition 1** *The Monge–Kantorovich problem (4) admits the dual*

$$\mathrm{L}_C(p,q) = \max_{(f,g) \in \mathrm{R}(C)} \langle f, p \rangle + \langle g, q \rangle, \tag{5}$$

*where $\mathrm{R}(C)$ is the set of admissible dual variables:*

$$\mathrm{R}(C) := \{(f,g) \in \mathbb{R}^n \times \mathbb{R}^n : \quad f \oplus g := f 1_n + 1_n g^T \leq C\} \tag{6}$$

$$= \{(f,g) \in \mathbb{R}^n \times \mathbb{R}^n : \quad f_i + g_j \leq C_{ij}, \quad \forall i,j = 1, \ldots, n\}. \tag{7}$$

The dual variables $(f, g)$ are usually called *Kantorovich potentials*.

**Remark 1.** Before we prove Proposition 1, let us note that the right-hand side of the equation (5) is the lower-bound of $\mathrm{L}_C(p, q)$ (4):

$$\min_{\pi \in U(p,q)} \sum_{i,j} C_{ij}\pi_{ij} \geq \max_{(f,g)\in \mathrm{R}(C)} \langle f, p\rangle + \langle g, q\rangle. \tag{8}$$

This is because for any transport plan $\pi$ (including the optimal one) and for any $(f, g) \in \mathrm{R}(C)$,

$$\sum_{i,j} \pi_{ij}C_{ij} \geq \sum_{i,j} \pi_{ij}(f_i + g_j) = (\sum_i f_i \sum_j \pi_{ij}) + (\sum_j g_j \sum_i \pi_{ij}) = \langle f, p\rangle + \langle g, q\rangle. \tag{9}$$

**Proof.** [of Proposition 1] Let us write down the Lagrangian to the problem (4):

$$\min_{\pi \in \mathbb{R}_+^{n\times n}} \max_{(f,g)\in \mathbb{R}^n \times \mathbb{R}^n} \langle C, \pi\rangle + \langle p - \pi 1_n, f\rangle + \langle q - \pi^T 1_n, g\rangle, \tag{10}$$

which is equivalent to

$$\min_{\pi \in \mathbb{R}_+^{n\times n}} \max_{(f,g)\in \mathbb{R}^n \times \mathbb{R}^n} \langle C, \pi\rangle + \langle p, f\rangle - \langle f, \pi 1_n\rangle + \langle q, g\rangle - \langle g, \pi^T 1_n\rangle \quad \Leftrightarrow \tag{11}$$

$$\min_{\pi \in \mathbb{R}_+^{n\times n}} \max_{(f,g)\in \mathbb{R}^n \times \mathbb{R}^n} \langle C, \pi\rangle + \langle p, f\rangle - \langle f 1_n^T, \pi\rangle + \langle q, g\rangle - \langle 1_n g^T, \pi\rangle \quad \Leftrightarrow \tag{12}$$

$$\min_{\pi \in \mathbb{R}_+^{n\times n}} \max_{(f,g)\in \mathbb{R}^n \times \mathbb{R}^n} \langle p, f\rangle + \langle q, g\rangle + \langle C - f 1_n^T - 1_n g^T, \pi\rangle. \tag{13}$$

Now we can exchange min and max, which is always possible in linear programs in finite dimension, and get:

$$\max_{(f,g)\in \mathbb{R}^n \times \mathbb{R}^n} \langle p, f\rangle + \langle q, g\rangle + \min_{\pi \in \mathbb{R}_+^{n\times n}} \langle C - f 1_n^T - 1_n g^T, \pi\rangle. \tag{14}$$

Now let us note that

$$\min_{\pi \in \mathbb{R}_+^{n\times n}} \langle Q, \pi\rangle = \begin{cases} 0, & \text{if } Q \in \mathbb{R}_+^{n\times n}, \\ -\infty, & \text{otherwise.} \end{cases} \tag{15}$$

Therefore, from the constraint, we get $C - f 1_n^T - 1_n g^T = C - f \oplus g \geq 0$, and the statement is proved. ■

# 3 Entropic Regularization

Entropic regularization is an important technique used in the context of optimal transport to introduce a smoothing term that enhances computational efficiency and stability. Below, we consider this concept.

**Definition 5 (Discrete entropy of a coupling matrix)** *Discrete entropy of a coupling matrix is defined as*

$$\mathrm{H}(\pi) = -\sum_{i,j} \pi_{ij}(log(\pi_{ij}) - 1). \tag{16}$$

*The definition for vectors $a \in \mathbb{R}^n$ is similar, with the convention that $H(a) = +\infty$ if one of the entries $a_j$ is $\leq 0$.*

Note that the function $-H$ is 1-strongly convex, because its Hessian is

$$\partial^2(-\mathrm{H}(\pi)) = -\partial^2(\mathrm{H}(\pi)) = -(-diag(\frac{1}{\pi_{ij}})) = diag(\frac{1}{\pi_{ij}})$$

and $\pi_{ij} \leq 1$, so $\partial^2(-\mathrm{H}(\pi)) \geq I_{n^2}$.

The idea of **entropic regularization** of the optimal transport problem (4) is to use $-\mathrm{H}$ as a regularizing function to obtain approximate solutions of (4). This results in an approximation of the original problem that is easier to solve:

$$\mathrm{L}_C^{\varepsilon}(p,q) := \min_{\pi \in U(p,q)} \langle \pi, C \rangle - \varepsilon \mathrm{H}(\pi). \tag{17}$$

The objective is $\varepsilon$-strongly convex function, and problem (17) has a unique optimal solution.

**Proposition 2 (Convergence with $\varepsilon$ [8])** *The unique solution $\pi_\varepsilon$ of (17) converges to the optimal solution with maximal entropy within the set of all optimal solutions of the initial problem (4):*

$$\pi_\varepsilon \stackrel{\varepsilon \to 0}{\to} \arg\min_{\pi} \{-\mathrm{H}(\pi) : \quad \pi \in (p,q), \quad \langle \pi, C \rangle = \mathrm{L}_C(p,q)\} \tag{18}$$

*so that in particular*

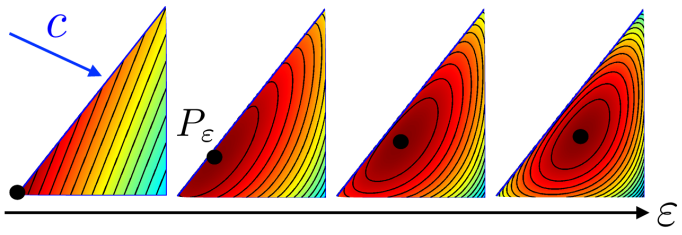
$$\mathrm{L}_C^{\varepsilon}(p,q) \stackrel{\varepsilon \to 0}{\to} \mathrm{L}_C(p,q). \tag{19}$$

*One also has*

$$\pi_\varepsilon \stackrel{\varepsilon \to \infty}{\to} ab^T. \tag{20}$$

**Example 1. Impact of $\varepsilon$ on solution** [8] Figure 1 illustrates the effect of the entropy to regularize a linear program over the simplex $S_1(3)$ (which can thus be visualized as a triangle in two dimensions). The entropy pushes the original LP solution away from the boundary of the triangle. The optimal $P_\varepsilon := \pi_\varepsilon$ progressively moves toward an "entropic center" of the triangle.

Figure 1: Example: impact of $\varepsilon$ on the optimization of a linear function on the simplex, solving $P_\varepsilon := \pi_\varepsilon = \arg\min_\pi \langle \pi, C \rangle - \varepsilon \mathrm{H}(\pi)$ for a varying $\varepsilon$.



# 4 Numerical Methods for Solving the Linear Programming Transport Problem with Entropic Regularization

In this section, we focus on the numerical methods for solving OT problem with entropic regularization. The introduction of entropic regularization allows for more efficient numerical algorithms for OT problem and is useful in applications in ML, computer vision, etc.

Below we discuss, implement, and compare in experiments several methods for OT with entropic regularization: Sinkhorn's algorithm, Sinkhorn–Knopp algorithm, Greenkhorn algorithm, Adaptive Primal–Dual Accelerated Gradient Descent (APDAGD), Primal-Dual Accelerated Alternating Minimization Algorithm (AAM). Their complexities are presented in Table 1.

## 4.1 Sinkhorn's Algorithm [3, 8, 9]

**Definition 6 (Gibbs kernel associated to cost matrix)** *A Gibbs kernel associated to a cost matrix $C \in \mathbb{R}_+^{n\times n}$ is a matrix $K \in \mathbb{R}_+^{n\times n}$ defined as*

$$K_{ij} = e^{\frac{-C_{ij}}{\varepsilon}}$$

**Proposition 3** *The solution to the OT problem with entropic regularization (17) is unique and has the form*

$$\pi_{ij} = u_i K_{ij} v_j, \tag{21}$$

*for two (unknown) scaling variables $(u, v) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$.*

**Proof.** Let us write down the Lagrangian of (17) with two dual variables $f, g \in \mathbb{R}^n$. Similarly to the proof of Proposition 1, we have

$$\min_{\pi\in\mathbb{R}_+^{n\times n}} \max_{(f,g)\in\mathbb{R}^n\times\mathbb{R}^n} \langle C, \pi\rangle - \varepsilon \mathrm{H}(\pi) - \langle f, \pi 1_n - p\rangle - \langle g, \pi^T 1_n - q\rangle. \tag{22}$$

Then the first order condition, by taking the derivative by $\pi_{ij}$ of the expression, gives us

$$C_{ij} + \varepsilon \log(\pi_{ij}) - f_i - g_j = 0. \tag{23}$$

Therefore,

$$\pi_{ij} = e^{\frac{f_i}{\varepsilon}} \frac{-C_{ij}}{\varepsilon} \frac{g_j}{\varepsilon},$$

which can be rewritten in the form provided above with non-negative vectors $u$ and $v$. ■

**Remark 2. Matrix and vector forms of optimal solution** Note that the factorization (21) of an optimal solution can be rewritten in matrix form:

$$\pi = \mathrm{diag}(u) K \mathrm{diag}(v). \tag{24}$$

By the definition of $\pi$, the following equalities should be satisfied:

$$\mathrm{diag}(u) K \mathrm{diag}(v) 1_n = p, \qquad \mathrm{diag}(v) K^T \mathrm{diag}(u) 1_n = q. \tag{25}$$

Note that $\text{diag}(v)1_n = v$ and $\text{diag}(u)1_n = u$, and we get

$$u \odot (Kv) = p, \qquad v \odot (K^T u) = q, \tag{26}$$

where $\odot$ stands for Hadamard (element-wise) product. That problem is known in the numerical analysis community as the matrix scaling problem (see [7]).

**Remark 3. Sinkhorn's algorithm updates** An intuitive way to handle the equations (26) is to solve them iteratively, by modifying first $u$ so that it satisfies the left-hand side of (26) and then $v$ to satisfy its right-hand side. These two updates define Sinkhorn's algorithm:

$$u^{l+1} := \frac{p}{Kv^{(l)}}, \qquad v^{l+1} := \frac{q}{K^T u^{(l+1)}}, \tag{27}$$

where $v^{(0)}$ can be initialized as $1_n$ and the division is performed element-wise. Note that a different initialization will likely lead to a different solution for $u, v$, since $u, v$ are only defined up to a multiplicative constant, but all result in the same optimal coupling $\text{diag}(u)K\text{diag}(v)$.

**Algorithm 1** Sinkhorn's Algorithm

**Input:** Cost matrix $C \in \mathbb{R}^{n \times n}$, marginals $p \in \mathbb{R}^n$, $q \in \mathbb{R}^n$, reg. parameter $\varepsilon > 0$, tolerance $\tau > 0$
Initialize $K \leftarrow \exp\left(-\frac{C}{\varepsilon}\right)$ (element-wise exponent)
Initialize $v \leftarrow \mathbf{1}_n$ (vector of ones)
**while** not converged **do**
  $u \leftarrow p/(Kv)$ element-wise
  $v \leftarrow q/(K^T u)$ element-wise
  **if** $\|u \odot (Kv) - p\|_1 + \|v \odot (K^T \cdot u) - q\|_1 < \tau$ **then**
    break
  **end if**
**end while**
**Output:** Approximate matrix $\pi \in \mathbb{R}^{n \times n}$

## 4.2 Sinkhorn-Knopp Algorithm [6]

Sinkhorn-Knopp Algorithm is based on the observation that the coupling matrix $\pi$ can be efficiently adjusted through scaling operations to enforce the marginal constraints while minimizing the entropic regularized cost. The Sinkhorn-Knopp algorithm minimizes the regularized optimal transport objective (17) by alternating between two steps: row and column scaling.

**Algorithm 2** Sinkhorn-Knopp Algorithm

**Input:** Cost matrix $C$, source distribution $p$, target distribution $q$, regularization parameter $\varepsilon$

Initialize $\pi^{(1)} \leftarrow \exp\left(-\frac{C}{\varepsilon}\right)$

**for** each iteration $k = 1, \ldots, N$ **do**

$\pi^{(k+\frac{1}{2})} \leftarrow \operatorname{diag}\left(\frac{p}{\pi^{(k)} 1_n}\right) \pi^{(k)}$ ▷ Row scaling (element-wise division)

$\pi^{(k+1)} \leftarrow \pi^{(k+\frac{1}{2})} \left(\frac{q}{(\pi^{(k+\frac{1}{2})})^T 1_n}\right)$ ▷ Column scaling (element-wise division)

**end for**

**Output:** Approximate matrix $\pi \in \mathbb{R}^{n \times n}$

## 4.3 Greenkhorn's Algorithm [4, 5]

In each iteration of the Sinkhorn's (and Sinkhorn–Knopp) algorithm, all the elements of $u^{(k)}$ or $v^{(k)}$ are updated simultaneously such that the row sum of $\pi^{(k)}$ equals $p$ or the column sum of $\pi^{(k)}$ equals $q$. In the Greenkhorn algorithm, in contrast, only a single element of $u^{(k)}$ or $v^{(k)}$ is updated at a time, such that only one element of the row sum or column sum of $\pi^{(k)}$ is equal to the target value. To determine which element of $u^{(k)}$ or $v^{(k)}$ is updated, the following scalar version of the KL-divergence is used to quantify the mismatch between the elements of $p$ or $q$ and the corresponding elements of $\pi 1_n$ or $\pi^T 1_n$:

**Definition 7 (Scalar-value KL-divergence)** *For two scalars $x, y \in \mathbb{R}$, KL-divergence is defined as*

$$\rho(x, y) := y - x + x \log \frac{x}{y}. \tag{28}$$

*Note that $\rho(x, y)$ is indeed the Bregman distance between $x$ and $y$ associated with the function $\phi(t) = t \log t$.*

**Algorithm 3** Greenkhorn Algorithm

**Input:** Cost matrix $C \in \mathbb{R}^{n \times n}$, marginals $p \in \mathbb{R}^n$, $q \in \mathbb{R}^n$, reg. parameter $\varepsilon > 0$, tolerance $\tau > 0$
Initialize $K \leftarrow \exp\left(-\frac{C}{\varepsilon}\right)$ ▷ (element-wise exponent)
Initialize $u^{(0)} \leftarrow p$, $v^{(0)} \leftarrow q$, $\pi^{(0)} \leftarrow \mathrm{diag}(u_0) K \mathrm{diag}(v_0)$
**while** not converged **do**
  $I \leftarrow \arg\max_i \rho(p_i, (\pi^{(k)} 1_n)_i)$ ▷ Choose row of $\pi^{(k)}$ "farest" from $p$
  $J \leftarrow \arg\max_j \rho(q_j, ((\pi^{(k)})^T 1_n)_j)$ ▷ Choose column of $\pi^{(k)}$ "farest" from $q$
  $u^{(k+1)} \leftarrow u^{(k)}$, $v^{(k+1)} \leftarrow v^{(k)}$,
  **if** $\rho(p_I, (\pi^{(k)} 1_n)_I) > \rho(q_J, ((\pi^{(k)})^T 1_n)_J)$ **then** ▷ If current rows are "worse" than columns
    $(u^{(k+1)})_I \leftarrow \frac{p_I}{(K v^{(k)})_I}$
  **else** $(v^{(k+1)})_J \leftarrow \frac{q_J}{(K^T u^{(k)})_J}$
  **end if**
  $\pi^{(k+1)} \leftarrow \mathrm{diag}(u^{(k+1)}) K \mathrm{diag}(v^{(k+1)})$
  **if** $\|u^{(k+1)} \odot (K v^{(k+1)}) - p\|_1 + \|v^{(k+1)} \odot (K^T \cdot u^{(k+1)}) - q\|_1 < \tau$ **then**
    break
  **end if**
  $k \leftarrow k + 1$
**end while**
**Output:** Matrix $\pi^{(k)} \in \mathbb{R}^{n \times n}$

In practice, Greenkhorn algorithm usually gives better results than Sinkhorn's algorithm [5].

Table 1: Complexities of various algorithms.

| **Algorithm** | **Complexity** |
|---|---|
| Sinkhorn | $O(n^2 \|C^3\|_\infty \log n/\delta^3)$ [5] |
| Sinkhorn–Knopp | $O(n^2 \|C^3\|_\infty \log n/\delta^3)$ [5] |
| Greenkhorn | $O(n^2 \|C^3\|_\infty \log n/\delta^3)$ [5] |
| Primal-Dual AGD | $O(\frac{n^{\frac{5}{2}} \sqrt{\log n} \|C\|_\infty}{\delta})$[1] |
| Primal-Dual AAM | $O(\frac{n^{\frac{5}{2}} \sqrt{\log n} \|C\|_\infty}{\delta})$ [2] |

# 5 Experiments

I have implemented four algorithms: Sinkhorn, Sinkhorn–Knopp, Greenkhorn, and AGD. The code and the provided experiments are presented here: https://github.com/katyafilimoshina/otproject/tree/main.

### 3-dimensional task, $\varepsilon = 0.1,\ 0.0001,\ 0.5$

In this toy example, we consider

$$p = (0.4, 0.3, 0.3), \qquad q = (0.5, 0.2, 0.3) \tag{29}$$

$$C = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \tag{30}$$

So the costs of transportation from source $i$ to $j$, $i \neq j$, are equal. We solve the OT problem with entropic regularization (17) with different regularization parameters $\varepsilon = 0.1,\ 0.0001,\ 0.5$. The results by Sinkhorn, Sinkhorn–Knopp, and Greenkhorn algorithms are similar, the result by AGD is quite different. All of them are presented in Colab notebook in GitHub repository.

As we see from Tables 2–4, the AGD method has the best performance in all the experiments, because the final transportation cost is the smallest. Also note that when $\varepsilon$ very small, then the first 3 algorithms fail and give incorrect solution that $\pi \notin U(p, q)$. Also it connects with the theory, that when $\varepsilon$ is small (experiment 1), than the solution is sparse, and when $\varepsilon$ is large (experiment 3), the output is dense.

Table 2: Results of experiments with $\varepsilon = 0.1$

| **Algorithm** | **Num of iter** | **Transport cost** | $\pi \in U(p, q)$? |
|---|---|---|---|
| Sinkhorn | 786 | 0.1012 | yes |
| Sinkhorn–Knopp | 788 | 0.1012 | yes |
| Greenkhorn | 146 | 0.1012 | yes |
| Primal-Dual AGD | 56 | 0.1 | yes |

Table 3: Results of experiments with $\varepsilon = 0.0001$

| **Algorithm** | **Num of iter** | **Transport cost** | $\pi \in U(p, q)$? |
|---|---|---|---|
| Sinkhorn | max iter (1000) | 0 | no |
| Sinkhorn–Knopp | max iter (1000) | 0 | no |
| Greenkhorn | max iter (1000) | 0 | no |
| Primal-Dual AGD | 19998 | 0.1 | yes |

Table 4: Results of experiments with $\varepsilon = 0.5$

| **Algorithm** | **Num of iter** | **Transport cost** | $\pi \in U(p, q)$? |
|---|---|---|---|
| Sinkhorn | 16 | 0.2413 | yes |
| Sinkhorn–Knopp | 17 | 0.2413 | yes |
| Greenkhorn | 58 | 0.2413 | yes |
| Primal-Dual AGD | 11 | 0.1043 | yes |

Figure 2: Transport cost VS iteration for $\varepsilon = 0.5$

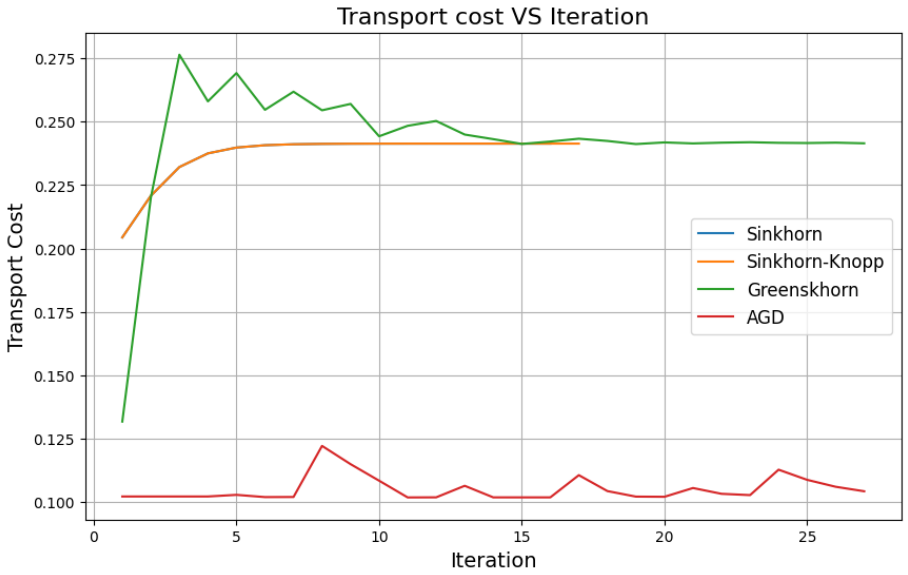
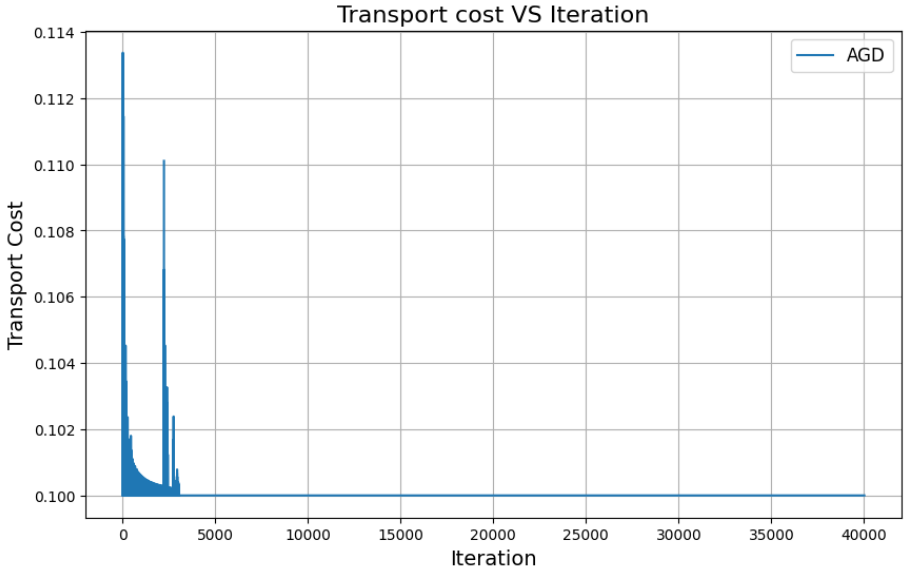



Figure 3: Transport cost VS iteration in AGD for $\varepsilon = 0.0001$



### 100-dimensional task, $\varepsilon = 0.1,\ 0.0001,\ 0.5$

In this task, we randomly generate a symmetric cost matrix $C \in \mathbb{R}^{100\times 100}$ with 0-s on the main diagonal and two random vectors $p, q \in \mathbb{R}^{100}_{+}$ from probability simplex. The results are more or less similar to the results in the case of the $\mathbb{R}^3$. The results are in the notebook and in Tables 5–7.

Table 5: Results of experiments with $\varepsilon = 0.1$

| **Algorithm** | **Num of iter** | **Transport cost** | $\pi \in U(p,q)$? |
|---|---|---|---|
| Sinkhorn | 9 | 0.1719 | yes |
| Sinkhorn–Knopp | 10 | 0.1719 | yes |
| Greenkhorn | max iter (1000) | 0.1719 | yes |
| Primal-Dual AGD | 43 | 0.0586 | yes |

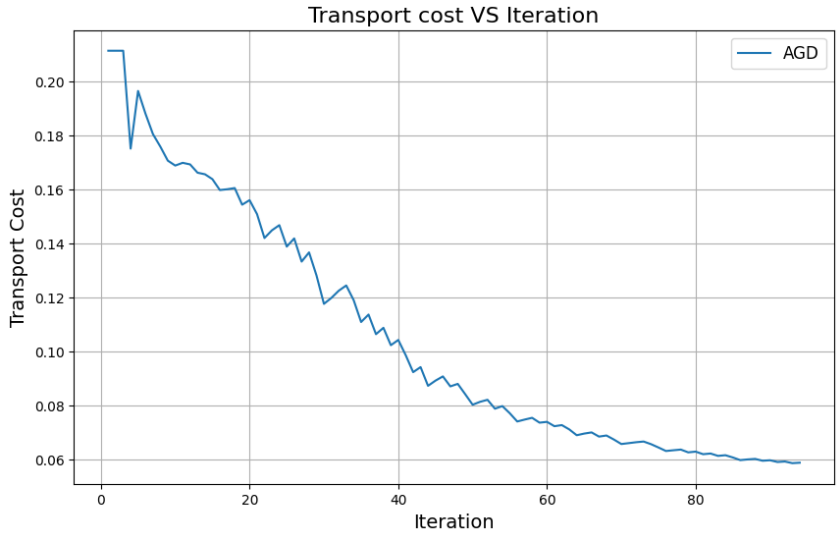
Table 6: Results of experiments with $\varepsilon = 0.0001$

| **Algorithm** | **Num of iter** | **Transport cost** | $\pi \in U(p,q)$? |
|---|---|---|---|
| Sinkhorn | - | - | no |
| Sinkhorn–Knopp | - | - | no |
| Greenkhorn | - | - | no |
| Primal-Dual AGD | 19998 | 0.1057 | yes |

Table 7: Results of experiments with $\varepsilon = 0.5$

| **Algorithm** | **Num of iter** | **Transport cost** | $\pi \in U(p,q)$? |
|---|---|---|---|
| Sinkhorn | 3 | 0.4035 | yes |
| Sinkhorn–Knopp | 4 | 0.4035 | yes |
| Greenkhorn | 610 | 0.4035 | yes |
| Primal-Dual AGD | 6 | 0.1187 | yes |

Figure 4: Trasnport cost vs iteration for AGD with $\varepsilon = 0.1$




## 10000-dimensional task, $\varepsilon = 0.1$

In this task, we randomly generate a symmetric cost matrix $C \in \mathbb{R}^{10000\times 10000}$ with 0-s on the main diagonal and two random vectors $p, q \in \mathbb{R}_+^{10000}$ from probability simplex.

For this task, we used Sinkhorn algorithm. The results are presented in the notebook. The algorithm converged in 3 iterations, to the solution $\pi \in U(p, q)$. Final transport cost is 0.1929.

## Application of Sinkhorn to color transfer

In this experiment, we apply optimal transport Sinkhorn algorithm to the color transfer task. The task is to transfer color style from Image 2 to some Image 1. The cost matrix is defined using Euclidean distances between all pairs of pixels of two images. The vectors $p$ and $q$ correspond to the distribution of colors in images 1 and 2 respectively.

Since this problem is computational costly for naively implemented (by me) Sinkhorn algorithm, we use its implementation from POT library for Python. The implementation is presented in our GitHub repository. The results of several experiments are on Figures 5 and 6.

Figure 5: Sinkhorn to color transfer, ex. 1


Figure 6: Sinkhorn to color transfer, ex. 2


# 6 Conclusions

## Results

Within this project, I have:

- studied the concepts of OT and dual tasks with and without entropic regularization.
- considered the algorithms for OT with entropic regularization: Sinkhorn, Sinkhorn–Knopp, Greenkhorn, AGD, AAM.
- implemented in Python: Sinkhorn, Sinkhorn–Knopp, Greenkhorn, AGD algorithms.
- coded experiments for these algorithms in the case of $\mathbb{R}^3$, $\mathbb{R}^{100}$, and $\mathbb{R}^{10000}$ and $\varepsilon = 0.1, 0.0001, 0.5$.
- studied and coded application of OT Sinkhorn algorithm to color transfer task.

## Advantages and drawbacks of entropic regularization

**The advantages** of entropic regularization and ideas behind it:

+ As $\varepsilon$ increases, the optimal coupling becomes less and less sparse (in the sense of having entries larger than a prescribed threshold), which in turn has the effect of both accelerating computational algorithms (as we see, for example, in Sinkhorn's algorithm) and leading to faster convergence.

+ The idea to regularize the optimal transport problem by an entropic term can be traced back to modeling ideas in transportation theory [Wilson, 1969]: Actual traffic patterns in a network do not agree with those predicted by the solution of the optimal transport problem. Indeed, the former are more diffuse than the latter, which tend to rely on a few routes as a result of the sparsity of optimal couplings for (4).

+ Adding entropy prevents numerical instability and ill-conditioning, which can occur in the unregularized OT problem.

+ Entropic regularization allows the use of more efficient algorithms, like Sinkhorn's algorithm, which can be faster and more scalable than direct methods.

**The drawbacks** of entropic regularization in OT:

- The solution with entropy regularization is only an approximation of the true OT solution. As the regularization parameter $\varepsilon$ decreases, the solution becomes closer to the true OT solution, but this requires more computation.

- The performance depends on the choice of the regularization parameter $\varepsilon$, which must be tuned. A poorly chosen parameter can lead to suboptimal performance.

## Why is it convenient to consider dual task in OT?

+ The dual formulation of OT has already made numerical algorithms (for more general tasks in linear programming).

+ As discussed above, the dual solution provides a tight lower bound for the primal problem.

# References


[1] Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. *arXiv preprint arXiv:1802.04367*, 2018.

[2] Sergey Guminov, Pavel Dvurechensky, Nazarii Tupitsa, and Alexander Gasnikov. On a combination of alternating minimization and nesterov's momentum. 2021.

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

[4] Tianyi Lin, Nhat Ho, and Michael Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3982–3991. PMLR, 2019.

[5] Jianzhou Luo, Dingchuan Yang, and Ke Wei. Improved complexity analysis of the sinkhorn and greenkhorn algorithms for optimal transport. *arXiv preprint arXiv:2305.14939*, 2023.

[6] Jose Rafael Espinosa Mena. On the convergence of the sinkhorn-knopp algorithm with sparse cost matrices. *arXiv preprint arXiv:2405.20528v3 [math.OC]*, 2024.

[7] Arkadi Nemirovski and Uriel Rothblum. On complexity of matrix scaling. *Linear Algebra and its Applications*, 302-303:435–460, 1999.

[8] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *arXiv preprint arXiv:1803.00567*, 2020.

[9] Nazarii Tupitsa, Pavel Dvurechensky, Darina Dvinskikh, and Alexander Gasnikov. Numerical methods for large-scale optimal transport. *arXiv preprint arXiv:2210.11368*, 2022.