Accelerating the Unbalanced Optimal Transport Problem Using Dynamic Penalty Updating

1st Xun Su
Graduate School of Fundamental Science and Engineering
WASEDA University
Tokyo, Japan
suxun_opt@asagi.waseda.jp

2nd Hiroyuki Kasai
School of Fundamental Science and Engineering
WASEDA University
Tokyo, Japan
hiroyuki.kasai@waseda.jp

***Abstract*—With the increasing application of Optimal Transport (OT) in machine learning, the unbalanced optimal transport (UOT) problem, as a variant of optimal transport, has gained attention for its improved generality. There is an urgent need for fast algorithms that can efficiently handle large penalty parameters. In this paper, we prove that the recently proposed Majorize-Minimization algorithm for the UOT problem can be viewed as a form of Bregman proximal descent, and we propose to use dynamic penalty updating to make the algorithm converge quickly even for large penalties. By using a dynamic scheme, we can successfully compute better and sparser solutions for the large penalty parameter and approach the computational speed of the well-known Sinkhorn algorithm, which sacrifices accuracy by adding an entropy item.**

***Index Terms*—Optimization, Optimal Transport**

I. Introduction

Optimal transport (OT) has gained significant attention in the fields of machine learning and statistical learning due to its capacity to measure the distance between two probability measures. Combined OT methods have demonstrated superiority over traditional methods in areas such as domain adaptation [1] and generative models [2]. Recently, OT theory has been applied to diverse technical fields, including graph analysis [3]–[5] and sequential data analysis [6]. The popularity of OT can be attributed to the introduction of Sinkhorn's algorithm [7] for the entropy-regularized Kantorovich formulation problem, which has reduced the computational complexity associated with large-scale problems. However, the standard OT problem is limited to handling only *balanced* samples. To accommodate a wider range of applications with *unbalanced* samples, relaxed OT has been proposed, including partial OT (POT) [8], semi-relaxed OT (SROT) [9], [10], and unbalanced optimal transport (UOT) [11], [12]. The UOT has been proposed as a method to replace equality constraints with KL divergence as a penalty function. It is solvable by adding an entropic regularization term and utilizing Sinkhorn's algorithm. Although it is fast, scalable, and differentiable, it is prone to instability and results in larger errors in solutions compared to other regularizers.

Chapel et al. recently proposed a Majorization-Maximization (MM) algorithm to solve the Unbalanced Optimal Transport (UOT) problem without adding an entropy term by exploiting the connection between UOT and non-negative matrix factorization [13]. Although their algorithm is GPU compatible and computationally efficient, it produces a solution that is blurrier than that of Sinkhorn's algorithm and is slower, especially for large penalty terms. In this paper, we propose a novel approach to speed up the optimization process by combining the MM algorithm with a dynamic penalty method that has been successfully used in Penalty and Augmented Lagrangian methods. Our approach is simple and effective, and significantly improves the computational speed of the MM algorithm for larger penalty terms.

Our contributions are twofold:

- We show that the MM algorithm for UOT can be derived from the Bregman Proximal Descent (BPD) algorithm with the theoratical best step size.
- We propose a Dynamic Penalty MM algorithm (DPMM) that combines the MM algorithm with the dynamic penalty method to handle large penalization parameters. Our method achieves faster convergence and better performance, comparable to the Sinkhorn algorithm for balanced samples, and surpasses it for unbalanced samples.

II. Preliminaries

A. Notation

We use $\| \cdot \|_2$ to represent the Euclidean norm. $\mathbb{R}^n$ denotes n-dimensional Euclidean space, and $\mathbb{R}^n_+$ denotes the set of vectors in which all elements are non-negative. $\mathbb{R}^{n\times m}_+$ stands for the set of $n \times m$ matrices in which all elements are non-negative. We present vectors as bold lower-case letters $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots$ and matrices as bold-face upper-case letters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$. The i-th element of $\boldsymbol{a}$ and the element at the (i,j) position of $\mathbf{A}$ are stated respectively as a_i and $A_{i,j}$, the i-th column of $\mathbf{A}$ is represented as $\boldsymbol{a}_i$. In addition, $\mathbb{1}_n \in \mathbb{R}^n$ is the n-dimensional vector in which all elements are one. Additionally, we suggest vectorization for $\mathbf{A} \in \mathbb{R}^{n\times m}$ as lowercase letters $\boldsymbol{a} \in \mathbb{R}^{nm}$ and $\boldsymbol{a} = \mathrm{vec}(\mathbf{A}) = [A_{1,1}, A_{1,2}, \cdots, A_{m,n-1}, A_{m,n}]^T$, i.e., the concatenated vector of the transposed row vectors of $\mathbf{A}$. For two matrices of the same size $\mathbf{A}$ and $\mathbf{B}$, $\langle \mathbf{A}, \mathbf{B}\rangle = \mathrm{tr}(\mathbf{A}^T\mathbf{B})$ is the Frobenius dot-product.

B. Backgrounds on Optimal Transport

The balanced OT problem is defined as

$$\begin{aligned} \mathrm{OT}(\boldsymbol{a}, \boldsymbol{b}) \quad &:= \min_{\mathbf{T} \in \mathbb{R}_{+}^{n \times m}} \langle \mathbf{C}, \mathbf{T} \rangle \qquad (1) \\ \text{subject to} \quad & \mathbf{T} \mathbb{1}_n = \boldsymbol{a}, \mathbf{T}^T \mathbb{1}_m = \boldsymbol{b}. \end{aligned}$$

By relaxing the constraints using Kullback-Leibler (KL) divergence, we can obtain the UOT problem:

$$\begin{aligned} &\mathrm{UOT}(\boldsymbol{a}, \boldsymbol{b}) := \\ &\min_{\mathbf{T} \in \mathbb{R}_{+}^{n \times m}} \langle \mathbf{C}, \mathbf{T} \rangle + \tau \mathrm{KL}(\mathbf{T} \mathbb{1}_n, \boldsymbol{a}) + \tau \mathrm{KL}(\mathbf{T}^T \mathbb{1}_n, \boldsymbol{b}), \qquad (2) \end{aligned}$$

where $\mathrm{KL}(\boldsymbol{x}, \boldsymbol{y})$ stands for the KL divergence between $\boldsymbol{x} \in \mathbb{R}_{+}^{n}$ and $\boldsymbol{y} \in \mathbb{R}_{+}^{n}$, which is defined as $\sum_i \boldsymbol{x}_i \log(\boldsymbol{x}_i / \boldsymbol{y}_i) - \boldsymbol{x}_i + \boldsymbol{y}_i$.

C. MM Algorithm for UOT problem

For this UOT problem, Chapel et al. consider it as a composite optimization problem [13], which can be written as:

$$\min_{\boldsymbol{t}} f(\boldsymbol{t}) = \min_{\boldsymbol{t}} g(\boldsymbol{t}) + h(\boldsymbol{t}), \qquad (3)$$

where $g(\boldsymbol{t}) = \boldsymbol{c}^T \boldsymbol{t}$ and $h(\boldsymbol{t}) = \tau D_\phi(\mathbf{H}\boldsymbol{t}, \boldsymbol{y})$. $\boldsymbol{y} = [\boldsymbol{a}^T, \boldsymbol{b}^T]^T$, $\mathbf{H} = [\mathbf{M}^T, \mathbf{N}^T]^T$, $\mathbf{M}$ and $\mathbf{N}$ are the indicate matrix to computing the sum of $\boldsymbol{t}$ according to rows and columns in $\mathbf{T}$ form. where they propose the MM algorithm to solve the UOT problem, by building an auxiliary function for divergence D_ϕ, assuming $\tilde{Z}_{i,j} = \frac{H_{i,j} \tilde{t}_j}{\sum_l H_{i,l} \tilde{t}_l}$

$$\begin{aligned} G_\tau(\boldsymbol{t}, \tilde{\boldsymbol{t}}) = &\sum_{i,j} \tilde{Z}_{i,j} \phi\left(\frac{H_{i,j} t_j}{\tilde{Z}_{i,j}}\right) + \\ &\sum_j \left[\frac{c_j}{\tau} - \sum_i H_{i,j} \phi'(y_i)\right] t_j + \qquad (4) \\ &\sum_i \left[\phi'(y_i) y_i - \phi(y_i)\right] \end{aligned}$$

By minimizing $G_\tau(\boldsymbol{t}, \boldsymbol{t}^k)$, they can get the updating formula in the matrix form.

$$\begin{aligned} &\mathbf{T}^{(k+1)} = \\ &\mathrm{diag}\left(\frac{\boldsymbol{a}}{\mathbf{T}^{(k)} \mathbb{1}_m}\right)^{\frac{1}{2}} \left(\mathbf{T}^{(k)} \odot \exp\left(-\frac{\mathbf{C}}{2\tau}\right)\right) \mathrm{diag}\left(\frac{\boldsymbol{b}}{\mathbf{T}^{(k)\top} \mathbb{1}_n}\right)^{\frac{1}{2}}. \\ &\qquad (5) \end{aligned}$$

It is worth noting that the updating formula presented in (5) bears remarkable similarities with the widely popular Sinkhorn's algorithm, as it relies solely on matrix multiplication. While Sinkhorn's algorithm solves (2) with an additional regularization term $\epsilon \mathbf{H}(\mathbf{T}) = \epsilon \langle \mathbf{T}, \ln(\mathbf{T} - 1) \rangle$ using an alternative matrix multiplication method, it shares a similar computational structure with MM algorithm. This feature also allows for the use of GPU acceleration to speed up the computation process.

III. Proposed Algorithm

A. The MM algorithm and Its Bregman Proximal Descent Explanation

Traditional Gradient descent can not be applied on some Banach Space which the dual space is not consistent with the primal one, Mirror descent [?], [15] is a generalized method for handling related conditions. When it comes to the composite optimization problem, a proximal descent method can be combined with the mirror descent. Here, we would like to show that the Chapel's algorithm is one specific Bregman Proximal Descent [16].

Theorem 1. *Considering Applying BPD algorithm on 3, the updating formula can be written as*

$$\begin{aligned} &\mathbf{T}^{(k+1)} = \\ &\mathrm{diag}\left(\frac{\boldsymbol{a}}{\mathbf{T}^{(k)} \mathbb{1}_m}\right)^{\frac{\gamma}{\tau}} \left(\mathbf{T}^{(k)} \odot \exp\left(-\frac{\mathbf{C}}{2\tau}\right)\right) \mathrm{diag}\left(\frac{\boldsymbol{b}}{\mathbf{T}^{(k)\top} \mathbb{1}_n}\right)^{\frac{\gamma}{\tau}}, \\ &\qquad (6) \end{aligned}$$

where γ is the step size in the BPD algorithm

Proof. The proximal operator for the function g is defined as:

$$\mathrm{prox}_{\phi,\gamma}(x) = \mathrm{argmin}_z \left(\frac{1}{\gamma} D_\phi(z, x) + g(z)\right). \qquad (7)$$

For the UOT problem, we can get:

$$\mathrm{prox}_{\phi,\gamma}(\boldsymbol{t}) = \frac{\boldsymbol{t}}{e^{\gamma \boldsymbol{c}}}, \qquad (8)$$

then the BPD updating process is

$$\boldsymbol{t}^{k+1} = \mathrm{prox}_{\phi,\gamma}(\mathbf{t}^k - \gamma \nabla f(\boldsymbol{t}^k)) = \frac{\boldsymbol{t}^k}{e^{\gamma(\boldsymbol{c} + \nabla f(\boldsymbol{t}^k))}}. \qquad (9)$$

We can obtain (6) by rearranging (9). □

It is oblivious that (4) is a special condition for the BPD algorithm with step size $\gamma = \frac{\tau}{2}$, As proved in [17], the theoretical step size should be $\frac{1}{L}$, and L is the relative smoothness constant for function f to function ϕ.

Definition 2. [*Proposition 1.1 [18]*] *if function h is L-relatively smooth to function ϕ, $L \in \mathbb{R}^{+}$, then function $h - L\phi$ is convex, or $D_h(\boldsymbol{x}, \boldsymbol{y}) < L D_\phi(\boldsymbol{x}, \boldsymbol{y})$.*

Theorem 3. *For function $h(\boldsymbol{t}) = \tau D_\phi(\mathbf{H}\boldsymbol{t}, \boldsymbol{y})$, its relatively smoothness $L = 2/\tau$*

Proof. If we want to proof $h - L\phi$ is convex, it is equal to proof for $\forall \boldsymbol{d} \in \mathbb{R}^n$, we have $\boldsymbol{d}^T \Delta(h - L\phi)\boldsymbol{d} \succeq 0$

$$\begin{aligned}
\boldsymbol{d}^T \Delta(\frac{h(\boldsymbol{t})}{\tau})\boldsymbol{d} &= \sum_{i=1}^{n} \frac{(\boldsymbol{d}^T \boldsymbol{m}_i)^2}{\boldsymbol{t}^T \boldsymbol{M}_i} + \sum_{i=1}^{n} \frac{(\boldsymbol{d}^T \boldsymbol{n}_i)^2}{\boldsymbol{t}^T \boldsymbol{n}_i} \\
&\leq \sum_{i=1}^{n} \sum_{j=1}^{n^2} \frac{(d_j M_{ij})^2}{M_{ij} t_j} + \frac{(d_j N_{ij})^2}{N_{ij} t_j} \text{ (Cauchy Inequality)} \\
&= \sum_{j=1}^{n^2} (\sum_{i=1}^{n} (\frac{(M_{ij})^2}{M_{ij}} + \frac{(N_{ij})^2}{N_{ij}}) \frac{d_j^2}{t_j}) \\
&= \sum_{j=1}^{n^2} (\sum_{i=1}^{n} (M_{ij} + N_{ij}) \frac{d_j^2}{t_j}) \\
&= \sum_{j=1}^{n^2} (\mathbf{M}^T \mathbb{1} + \mathbf{N}^T \mathbb{1})_j \frac{d_j^2}{t_j} \\
&\leq \max_j (\mathbf{M}^T \mathbb{1} + \mathbf{N}^T \mathbb{1})_j \sum_{j}^{n^2} \frac{d_j^2}{t_j} \\
&\leq \frac{L}{\tau} \sum_{j}^{n^2} \frac{d_j^2}{t_j} \\
&= \frac{L}{\tau} \boldsymbol{d}^T \Delta h(\boldsymbol{t}) \boldsymbol{d}.
\end{aligned}$$

Then we have $L/\tau \geq \max_j (\mathbf{M}^T \mathbb{1} + \mathbf{N}^T \mathbb{1})_j = 2$, and the best theoretical learning rate is $2/\tau$. we can get the same updating formula as (5) after putting it into (6) □

B. Dynamic Penalized MM Algorithm

The idea of relaxing a Constrained Optimization problem by penalty function is firstly proposed as the penalty function method or Barrier method. Similar ideas appeared in the Augmented lagrangian method to speed up the convergence of the lagrangian method. When the parameter of the penalty function is too small, it is difficult to obtain an accurate solution, while when it is too large, the function becomes ill-conditioned, leading to slow convergence. To avoid this, the penalty method [19], [20] and the augmented Lagrangian method [21] often gradually increase the penalty parameter to prevent this situation.

Since the UOT problem introduces a penalty function to relax the constraints, it also faces similar difficulties, especially when the parameter of the penalty function is very large, the ill-conditioned nature of the problem makes convergence difficult, as illustrated in Figure 2. Therefore, we attempt to introduce the dynamic parameter update method of the penalty function and augmented Lagrangian method into the MM algorithm of the UOT problem and propose the DPMM algorithm.

By using a *dynamic penalization term*, we gradually increase its influence throughout the optimization process. we demonstrate its effectiveness in our proposed algorithm.

Our proposed algorithm is summarized in **Algorithm. 1**, where we set a small constant $q \in \mathbb{R}_+$ and gradually increase the value of $\tilde{\tau} \in \mathbb{R}_+$ as the optimization error reduces. This warm start initialization allows the solver to avoid the ill-conditioned Hessian matrix issue encountered in the early stages, and the process enables our algorithm to obtain a sparser initialization,

Algorithm 1 Dynamic Penalization MM algorithm (DPMM)

```
Input: $\mathbf{T}^0, \mathbf{C}, \tilde{\tau}, \tau, q$
Output: $\mathbf{T}^K$
  $\mathbf{G} = \exp(-\frac{\mathbf{C}}{2\tilde{\tau}})$
  for $k = 1$ to $K$ do
    $\boldsymbol{u} = (\frac{\boldsymbol{a}}{\mathbf{T}\mathbb{1}_n})^{\frac{1}{2}}, \boldsymbol{v} = (\frac{\boldsymbol{b}}{\mathbf{T}^T\mathbb{1}_m})^{\frac{1}{2}}$
    $\mathbf{T}^k = \mathbf{T}^k \odot (\boldsymbol{u}^T \mathbf{G} \boldsymbol{v})$
    $err = \|\mathbf{T}^{k-1} - \mathbf{T}^k\|_2$
    if $err \leq \frac{q}{\tilde{\tau}}$ and $\tilde{\tau} \leq \tau$ then
      $\tilde{\tau} = \min(\tau, 2\tilde{\tau})$
    end if
  end for
```

Since $\tilde{\tau}$ only doubles during the MM-IP algorithm for $O(log(\tau))$ times, the computation burden for recomputing matrix $\mathbf{K} = \exp\left(-\frac{\mathbf{C}}{2\lambda}\right)$ is ignorable compared with the MM algorithm.

IV. Experiments

We conducted experiments using randomly generated Gaussian distributions. In particular, we generated five pairs of 100-dimensional Gaussian distributions, each with the same mass. To test the performance of our approach in the case of unequal mass, we multiplied the mass of $\boldsymbol{a}$ by 1.2. For the mass-equal case, we obtained the analytical optimal solution $\mathbf{T}^*$ using linear programming. For both cases, we set $\tau = 1000$, and for Sinkhorn's algorithm, we set the regularizer parameter $\epsilon = 10^{-3}$. Additionally, we set the initial value of $\tilde{\tau} = 0.1$ and $q = 10^{-4}$ for our DPMM algorithm. We also incorporated Nesterov acceleration into our algorithm to obtain DPAMM. Figure 1 presents the results of our experiments.

These findings indicate that the MM algorithm struggles to minimize the transport cost when faced with a large penalization parameter. This leads to a significantly higher error compared to other methods in balanced samples. In the case of unbalanced samples, the algorithm's convergence is extremely slow. This is clearly illustrated in Figure 2, where the large value of τ causes the MM algorithm preserve to be dense, which is inferior to both Sinkhorn's algorithm and our proposed methods. Our approach can not only quickly solve with a clear structure similar to Sinkhorn's algorithm, but also maintain a small error for unbalanced samples, where Sinkhorn suffers from errors brought by the regularizer.

V. Conclusion

We focus on the recent progress of the UOT optimization algorithm, using the BPD algorithm to illustrate the MM algorithm in the UOT problem. Our experimental results illustrate the effectiveness of our proposed DPMM algorithm. Compared to the MM algorithm, our proposed method can

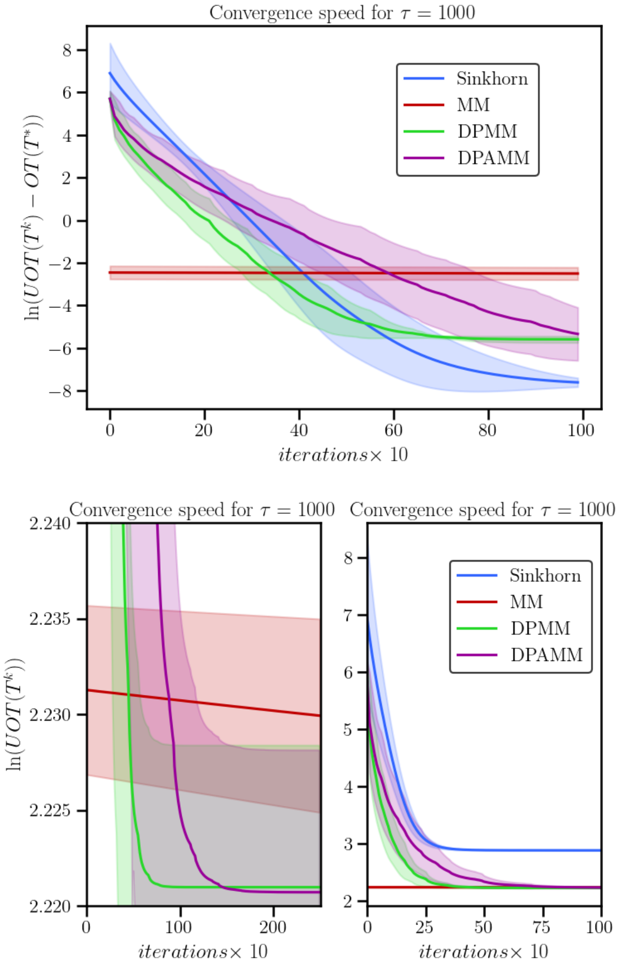

Fig. 1. Comparison of the convergence speed for different algorithms. The upper plot represents the results for balanced samples, while the lower plot displays the results for unbalanced samples. Using $\mathrm{OT}(\mathbf{T}^*)$ to represents the value of (1), and $\mathrm{UOT}(\mathbf{T}^k)$ to represents the function value calculated by replacing the optimal $\mathbf{T}$ in (2) with $\mathbf{T}^k$

effectively handle the challenge of larger τ values by utilizing an dynamic penalization process that avoids poor initialization. This results in a superior solution quality that is competitive with the widely-known Sinkhorn algorithm. In the future, we plan to incorporate our expertise in the field of ALM to further accelerate the MM algorithm.

References

[1] N. Courty, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 214–223. [Online]. Available: https://proceedings.mlr.press/v70/arjovsky17a.html

[3] J. Huang, Z. Fang, and H. Kasai, "LCS graph kernel based on Wasserstein distance in longest common subsequence metric space," *Digital Signal Processing*, vol. 189, p. 108281, 2021.

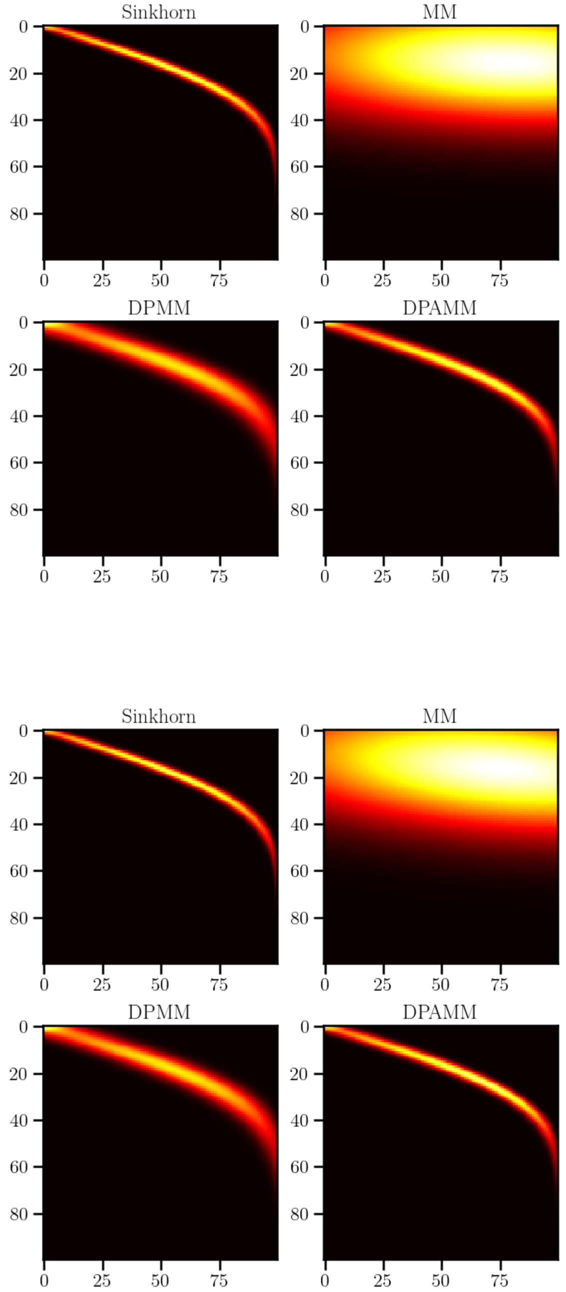

Fig. 2. Comparison of the solutions obtained using different optimization methods over 1000 iterations, The upper plot represents the results for balanced samples, while the lower plot displays the results for unbalanced samples. The MM algorithm fails to converge quickly to a near-sparse solution in any condition. MM-IP and AMM-IP methods perform significantly better, producing solutions not only that have a similar structure to Sinkhorn's but also better accuracy for unbalanced samples.

[4] J. Huang and H. Kasai, “Graph embedding using multi-layer adjacent point merging model,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[5] Z. Fang, S. X. Huang, Jianming, and H. Kasai, “Wasserstein graph distance based on L1-approximated tree edit distance between Weisfeiler-Lehman subtrees,” in *AAAI-23*, 2023.

[6] M. Horie and H. Kasai, “Auto-weighted sequential Wasserstein distance and application to sequence matching,” in *European Signal Processing Conference (EUSIPCO)*, 2022.

[7] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in *NeurIPS*, 2013.

[8] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol, “Regularized discrete optimal transport,” *SIAM*, vol. 7, no. 3, pp. 1853–1882, 2013.

[9] T. Fukunaga and H. Kasai, “Block-coordinate frank-wolfe algorithm and convergence analysis for semi-relaxed optimal transport problem,” in *ICASSP*, 2022.

[10] ——, “On the convergence of semi-relaxed sinkhorn with marginal constraint and ot distance gaps,” *arXiv preprint: arXiv:2205.13846*, 2022.

[11] L. A. Caffarelli and R. J. McCann, “Free boundaries in optimal transport and Monge-Ampère obstacle problems,” *Annals of Mathematics*, vol. 171, no. 2, pp. 673–730, 2010.

[12] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, “Scaling algorithms for unbalanced transport problems,” *arXiv preprint: arXiv:1607.05816*, 2017.

[13] L. Chapel, R. Flamary, H. Wu, C. Févotte, and G. Gasso, “Unbalanced optimal transport through non-negative penalized linear regression,” in *NeurIPS*, 2021.

[14] Y. Xie, X. Wang, R. Wang, and H. Zha, “A fast proximal point method for computing exact wasserstein distance,” in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, ser. Proceedings of Machine Learning Research, R. P. Adams and V. Gogate, Eds., vol. 115. PMLR, 22–25 Jul 2020, pp. 433–453. [Online]. Available: https://proceedings.mlr.press/v115/xie20b.html

[15] A. Beck and M. Teboulle, “Mirror descent and nonlinear projected subgradient methods for convex optimization,” *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637702002316

[16] F. Hanzely, P. Richtárik, and L. Xiao, “Accelerated bregman proximal gradient methods for relatively smooth convex optimization,” *Comput. Optim. Appl.*, vol. 79, no. 2, pp. 405–440, 2021. [Online]. Available: https://doi.org/10.1007/s10589-021-00273-8

[17] H. H. Bauschke, J. Bolte, and M. Teboulle, “A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications,” *Mathematics of Operations Research*, vol. 42, no. 2, pp. 330–348, 2017.

[18] H. Lu, R. M. Freund, and Y. Nesterov, “Relatively smooth convex optimization by first-order methods, and applications,” *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 333–354, 2018. [Online]. Available: https://doi.org/10.1137/16M1099546

[19] J. Joines and C. Houck, “On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga’s,” in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 1994, pp. 579–584 vol.2.

[20] D. Coit, A. Smith, and D. Tate, “Adaptive penalty methods for genetic optimization of constrained combinatorial problems,” *INFORMS Journal on Computing*, vol. 8, 06 1998.

[21] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611973365