Computing Wasserstein Barycenter via operator splitting: the method of averaged marginals

D. Mimouni, P Malisani, J. Zhu, W. de Oliveira

Computing Wasserstein Barycenter via operator splitting: the method of averaged marginals

D. Mimouni*[1,3], P. Malisani†[1], J. Zhu‡[2], and W. de Oliveira§[3]

[1]IFP Energies nouvelles, Dpt. Applied Mathematics, 1-4 Av Bois-Préau, 92852 Rueil-Malmaison
[2]IFP Energies nouvelles, Dpt. Control and Signal Processing, 1-4 Av Bois-Préau, 92852 Rueil-Malmaison
[3]Mines Paris, Center for Applied Mathematics, 1, rue Claude Daunesse, F-06904 Sophia Antipolis

Abstract

The Wasserstein barycenter (WB) is an important tool for summarizing sets of probability measures. It finds applications in applied probability, clustering, image processing, etc. When the measures' supports are finite, computing a (balanced) WB can be done by solving a linear optimization problem whose dimensions generally exceed standard solvers' capabilities. In the more general setting where measures have different total masses, we propose a convex nonsmooth optimization formulation for the so-called unbalanced WB problem. Due to their colossal dimensions, we introduce a decomposition scheme based on the Douglas-Rachford splitting method that can be applied to both balanced and unbalanced WB problem variants. Our algorithm, which has the interesting interpretation of being built upon averaging marginals, operates a series of simple (and exact) projections that can be parallelized and even randomized, making it suitable for large-scale datasets. Numerical comparisons against state-of-the-art methods on several data sets from the literature illustrate the method's performance.

1 Introduction

In applied probability, stochastic optimization, and data science, a crucial aspect is the ability to compare, summarize, and reduce the dimensionality of empirical/discrete measures. Since these tasks rely heavily on pairwise comparisons of measures, it is essential to use an appropriate metric for accurate data analysis. Different metrics define different barycenters of a set of measures: a barycenter is a mean element that minimizes the (weighted) sum of all its square distances to the set of target measures. When the chosen metric is the optimal transport one, and there is mass equality between the measures, the underlying barycenter is denoted by (balanced) Wasserstein Barycenter (WB).

The optimal transport metric defines the so-called Wasserstein distance (also known as Mallows or Earth Mover's distance), a popular choice in statistics, machine learning, and stochastic optimization [18, 28, 29]. The Wasserstein distance has several valuable theoretical and practical properties [31, 36] that are transferred to WBs [1, 16, 28, 30]. Indeed, thanks to the Wasserstein distance, one key advantage of WBs is their ability to preserve the underlying geometry of the data, even in high-dimensional spaces. This fact makes WBs particularly useful in image processing,

*daniel.mimouni@ifpen.fr
†paul.malisani@ifpen.fr
‡jiamin.zhu@ifpen.fr
§welington.oliveira@minesparis.psl.eu

where datasets often contain many pixels and complex features that must be accurately represented and analyzed [23, 33].

Being defined by the Wasserstein distance, WBs are challenging to compute. The Wasserstein distance is computationally expensive because, to compute an optimal transport plan, one needs to cope with a large linear program (LP) problem that has no analytical solution and cubic worst-case complexity[1] [41]. The situation becomes even worse for computing a WB of a set of finitely many discrete measures as the problem involves several transport plans [1]. This problem can be written as LP [4, 11], whose size becomes astronomical as it scales exponentially in the number of measures, exceeding thus the capabilities of standard LP solvers even for a small number of measures [4, 10, 11]. For this reason, significant effort has been made to reduce the LP's size and design specialized solvers [2, 4, 9, 11]. In particular, the work [9] proposes reduced LP models that exploit data structure. Although significantly smaller than the original LP problem defining WBs, those models are in general large scale and still hard to solve. The work [2] leverages techniques from computational geometry and combinatorial optimization to propose a specialized LP solver for computing WBs. The approach, which works on the dual problem and implements a separation oracle, is not efficient beyond moderate-scale inputs [2, § 5]. Indeed, a WB cannot be computed in time polynomial in the number of measures, (maximum) support size, and dimension [3].

Given the difficulty of computing exact (free-support) WBs, much research has focused on inexact approaches. A vast body of literature focuses on computing inexact WBs, either by employing approximate LP approaches as in [8, 26, 30], or by restricting the support of the WB to a fixed set, the so-called fixed-support approaches [16, 28, 41]. These techniques often employ a block-coordinate scheme consisting of two steps, first fixing the support and optimizing over the masses, then fixing the masses and optimizing over the support (of a given size). The first of these steps is an LP problem with the same structure as the exact (free-support) WB's LP formulation discussed above. The only difference is the LP's size, as fixing the support reduces the problem significantly. The second step in the block-coordinate scheme has a straightforward solution, provided the quadratic Wasserstein distance is employed.

Hence, whether an exact or inexact approach is employed to compute (approximate) a WB, one invariably has to face a large-scale LP of the form (see equations (9) and (10) for details)

$$\min_{\pi \in \mathcal{B}} \sum_{m=1}^{M} \langle c^{(m)}, \pi^{(m)} \rangle \quad \text{s.t.} \quad \pi^{(m)} \in \Pi^{(m)}, \quad m = 1, \ldots, M, \tag{1}$$

where M stands for the number of discrete measures, c for the transportation costs, $\Pi^{(m)}$ represents a polytope containing measures with given marginal, and $\mathcal{B}$ symbolizes a linear subspace. While exact techniques usually build upon linear programming techniques, inexact approaches tackle (1) via reformulations based on an entropic regularization [6, 16, 17, 22, 28, 41]. Indeed, the work [16] proposes to compute a WB inexactly by decomposing (1) along the measures and then regularizing the resulting optimal transportation problems with an entropy-like function. A projected subgradient method gives rise to a minimization scheme with decomposition to deal with the high dimensions of the LP. The regularization technique allows one to employ the celebrated Sinkhorn algorithm [15, 34], which has a simple closed-form expression and can be implemented efficiently using only matrix operations. Furthermore, this technique opened the way to the *Iterative Bregman Projection* (IBP) method proposed in [6]. IBP is highly memory efficient for distributions with a shared support set and is considered to be one of the most effective methods to tackle fixed-support WB problems. However, as IBP works with an approximating model and fixed support, the method falls in the class of inexact approaches.

Another approach fitting into the category of inexact methods has been recently proposed in [41], which uses the same type of regularization as IBP but decomposes the problem into a sequence of smaller subproblems with straightforward solutions. More specifically, the approach in [41] is a modification (tailored to the WB problem)

[1] More precisely, $O(S^3 log(S))$, with S the size of the input data.

of the *Bregman Alternating Direction Method of Multipliers* (B-ADMM) of [37]. The modified B-ADMM has been shown to compute promising results for sparse support measures and therefore is well-suited in some clustering applications. However, the theoretical convergence properties of the modified B-ADMM algorithm are not well understood and the approach should be considered as a heuristic. In the same vein, the work [40] proposes to address the WB problem via the standard ADMM algorithm, which decomposes the problem into smaller and simpler subproblems. As mentioned by the authors in their subsequent paper [41], the numerical efficiency of the standard ADMM is still inadequate for large datasets.

To cope with the challenge of solving LPs of the form (1) resulting from computing exact (free-support) or inexact (fixed-support) WBs, we propose a new algorithm based on the celebrated Douglas-Rachford splitting operator method (DR) [19–21]. Our proposal, which exploits the problem structure for decomposition, is denoted by *Method of Averaged Marginals* (MAM) as at every iteration, the algorithm computes a barycenter approximation by averaging marginals issued by transportation plans that are updated independently, in parallel, and even randomly if necessary. Accordingly, the algorithm operates a series of simple and exact projections that can be carried out in parallel and even randomly. These compelling features allow for considering data sets beyond moderate sizes in the free-support setting and attaining more accurate results than entropy-based methods usually get in the fixed-support case. Furthermore, MAM can be applied to a more general setting where measures have different total masses.

All the methods mentioned in the above references deal exclusively with sets of probability measures because WBs are limited to measures with equal total masses. A tentative way to circumvent this limitation is to normalize general positive measures to compute a standard (balanced) WB. However, such a naive strategy is generally unsatisfactory and limits the use of WBs in many real-life applications such as logistics, medical imaging, and others coming from the field of biology [24, 32]. Consequently, the concept of WB has been generalized to summarize such more general measures. Different generalizations of the WB exist in the literature, and they are based on variants of *unbalanced optimal transport problems* that define a distance between general non-negative, finitely supported measures by allowing for mass creation and destruction [24]. Essentially, such generalizations, known as unbalanced Wasserstein barycenters (UWBs), depend on how one chooses to relax the marginal constraints. In the review paper [32] and references therein, marginal constraints are moved to the objective function with the help of divergence functions. Differently, in [24] the authors replace the marginal constraints with sub-couplings and penalize their discrepancies. It is worth mentioning that UWB is more than simply copying with global variation in the measures' total masses. Generalized barycenters tend to be more robust to local mass variations, which include outliers and missing parts [32].

For the sake of a unified algorithmic proposal for both balanced and unbalanced WBs, in this work, we consider a different formulation for dealing with sets of measures with different total masses. Instead of relaxing both marginal constraints in each one of the M transportation plans as done in [24, 32] and references therein, our formulation generalizes the balanced WB by relaxing the constraint that the barycenter is a marginal measure of all underlying transportation plans. More specifically, by using the distance function to the subspace $\mathcal{B}$, that is $\texttt{dist}_{\mathcal{B}}(\pi) := \min_{\theta\in\mathcal{B}} \|\theta - \pi\|$, and a penalty parameter $\gamma > 0$, we propose the following nonlinear optimization problem yielding a UWB:

$$\min_{\pi} \sum_{m=1}^{M} \langle c^{(m)}, \pi^{(m)} \rangle + \gamma \texttt{dist}_{\mathcal{B}}(\pi) \quad \text{s.t.} \quad \pi^{(m)} \in \Pi^{(m)}, \quad m = 1, \ldots, M. \tag{2}$$

While our approach can be seen as an abridged alternative to the thorough methodologies of [24] and [32], its favorable structure for efficient splitting techniques combined with the good quality of the issued UWBs confirms the formulation's practical interest.

Thanks to our unified analysis, MAM can be applied to both balanced and unbalanced WB problems without any change: all that is needed is to set up the parameter $\gamma > 0$ in (2). To the best of our knowledge, MAM

is the first approach capable of handling balanced and unbalanced WB problems in a single algorithm, which can be further run in a deterministic or randomized fashion. In addition to its versatility, MAM copes with scalability issues arising from barycenter problems, is memory efficient, and has convergence guarantees. As further contributions, we conduct experiments on several data sets from the literature to demonstrate the computational efficiency and accuracy of the new algorithm and make our Python codes publicly available at the link (`https://ifpen-gitlab.appcollaboratif.fr/detocs/mam_wb`).

The remainder of this work is organized as follows. Section 2 introduces the notation and recalls the formulation of balanced WB problems. The proposed formulation for unbalanced WBs is presented in Section 3. Section 4 briefly recalls the Douglas-Rachford splitting (DR) method and its convergence properties both in the deterministic and randomized settings. The main contribution of this work, the Method of Averaged Marginals, is presented in Section 5. Convergence analysis is given in the same section by relying on the DR algorithm's properties. Section 6 illustrates the numerical performance of the deterministic and randomized variants of MAM on several data sets from the literature. Numerical comparisons with the free-support method [2] and fixed-support approaches in [6] and [41] are presented for the balanced case. Then, some applications of the UWB are considered.

2 Background on optimal transport and Wasserstein barycenter

Throughout this work, for $\tau \geq 0$ a given scalar, the notation $\Delta_R(\tau)$ denotes the set of vectors in $\mathbb{R}^R_+$ adding up to τ, that is,

$$\Delta_R(\tau) := \left\{ u \in \mathbb{R}^R_+ : \sum_{i=1}^{R} u_i = \tau \right\}. \tag{3}$$

If $\tau = 1$, then $\Delta_R(\tau)$, denoted simply by Δ_R, is the $R+1$ simplex. Let $\mathcal{P}(\mathbb{R}^d)$ be the set of Borel probability measures on $\mathbb{R}^d$. Furthermore, let ξ and ζ be two random vectors having probability measures μ and ν in $\mathcal{P}(\mathbb{R}^d)$, that is, $\xi \sim \mu$ and $\zeta \sim \nu$. Their (quadratic) 2-Wasserstein distance is given by:

$$W_2(\mu,\nu) := \left(\inf_{\pi \in U(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\xi - \zeta\|^2 d\pi(\xi,\zeta) \right)^{1/2}, \tag{WD}$$

where $U(\mu,\nu)$ is the set of all probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ having marginals μ and ν. We denote by $W_2^2(\mu,\nu)$ the squared Wassserstein distance, i.e., $W_2^2(\mu,\nu) := (W_2(\mu,\nu))^2$.

Definition 1 (Wasserstein Barycenter - WB)**.** *Given M measures $\{\nu^{(1)},\dots,\nu^{(M)}\}$ in $\mathcal{P}(\mathbb{R}^d)$ and $\alpha \in \Delta_M$, a* Wasserstein barycenter *with weights α is a solution to the following optimization problem*

$$\min_{\mu \in \mathcal{P}(\mathbb{R}^d)} \sum_{m=1}^{M} \alpha_m W_2^2(\mu,\nu^{(m)}). \tag{4}$$

Informally, a WB μ is a measure such that the total cost for transporting from μ to all $\nu^{(m)}$ is minimal concerning the quadratic Wasserstein distance. A WB μ exists in generality and, if one of the $\nu^{(m)}$ vanishes on all Borel subsets of Hausdorff dimension $d-1$, then it is also unique [1]. In this work, we are given M empirical (discrete) measures $\nu^{(m)}$ having finite support sets:

$$\mathtt{supp}(\nu^{(m)}) := \left\{ \zeta_1^{(m)},\dots,\zeta_{S^{(m)}}^{(m)} \right\} \quad \text{and} \quad \nu^{(m)} = \sum_{s=1}^{S^{(m)}} q_s^{(m)} \delta_{\zeta^{(m)}}, \tag{5}$$

with δ_u the Dirac unit mass on $u \in \mathbb{R}^d$ and $q^{(m)} \in \Delta_{S^{(m)}}$, $m = 1, \dots, M$. In this case, the uniqueness of WB is no longer ensured in general but the following results hold [4].

Theorem 1 (From [4]). *Consider M empirical measures $\nu^{(m)}$ as in (5). Then, problem (4) has at least one solution.*

a) Every solution μ satisfies

$$\mathtt{supp}(\mu) \subset \Xi := \left\{ \sum_{m=1}^{M} \alpha_m \zeta_s^{(m)} \,:\, \zeta_s^{(m)} \in \mathtt{supp}(\nu^{(m)}),\ m = 1, \dots, M \right\}. \tag{6}$$

b) There exists a sparse solution $\bar{\mu}$ such that

$$|\mathtt{supp}(\bar{\mu})| \leq T - M + 1, \text{ where } T := \sum_{m=1}^{M} S^{(m)} \tag{7}$$

c) If $\nu^{(m)}$, $m = 1, \dots, M$, are supported on the same grid $K_1 \times \cdots \times K_d$-grid in $\mathbb{R}^d$, and $\alpha_m = \frac{1}{M}$ for all m, then there exits a solution μ to (4) supported on $(M(K_1 - 1) + 1) \times \cdots \times (M(K_d - 1) + 1)$-grid, uniform in all directions.

Proof. Item a) is Proposition 1 (iii) in [4], Items b) and c) are Theorem 2 and Corollary 1 in the same paper. □

Let $R := |\Xi|$ be the number of points ξ in the finite set Ξ. It follows from item a) that any solution μ to problem (4) defined with discrete measures has the form

$$\mu = \sum_{r=1}^{R} p_r \delta_{\xi_r}, \quad \text{with} \quad p \in \Delta_R.$$

By letting $\mathcal{P}_\Xi(\mathbb{R}^d) := \left\{\mu \in \mathcal{P}(\mathbb{R}^d) : \mathtt{supp}(\mu) \subset \Xi\right\}$, problem (4) can be reformulated as a finite-dimensional LP by replacing the constraint $\mu \in \mathcal{P}(\mathbb{R}^d)$ with $\mu \in \mathcal{P}_\Xi(\mathbb{R}^d)$. Indeed, by considering all the R points in Ξ, problem (4) boils down to

$$\begin{cases} \min\limits_{p \in \Delta_R,\, \pi \geq 0} & \sum\limits_{m=1}^{M} \alpha_m \sum\limits_{r=1}^{R} \sum\limits_{s=1}^{S^{(m)}} \|\xi_r - \zeta_s^{(m)}\|^2 \pi_{rs}^{(m)} \\ \text{s.t.} & \sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)}, \quad s = 1, \dots, S^{(m)},\ m = 1, \dots, M \\ & \sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)} = p_r, \quad r = 1, \dots, R,\ m = 1, \dots, M. \end{cases} \tag{8}$$

Such LP scales exponentially in the number of measures. To see that, assume that all measures have support of same cardinality S, i.e., $S^{(m)} = S$ for all $m = 1, \dots, M$: then $R = S^M$ and the LP has $MRS + R = M(S)^{M+1} + (S)^M$ variables and $M(R + S) = M(S)^M + MS$ equality constraints. When the measures are supported on the same discrete grid in $\mathbb{R}^d$ and $\alpha_m = \frac{1}{M}$ for all m, the number of different points in Ξ reduces drastically: in this case, $S = K^d$ and $R = ((K - 1)M + 1)^d$ from item c) above, which is significantly smaller than $S^M = K^{dM}$ in the previous general setting (however still colossal in real-life applications) [9, 10]. These observations shed light on how the number of measures, the sizes of their support sets, and dimension d impact the size of problem (8). The paper [10] investigates the complexity of computing a sparse Wasserstein barycenter, and [3] shows that a WB cannot be computed in time polynomial in the number of measures, (maximum) support size, and dimension d.

Item b) ensures that a sparse solution exists with a support size of at most $M(S-1)+1$, motivating thus the so-called *fixed-support* approaches that generally employ a block-coordinate optimization heuristic: at iteration k, a support Ξ^k of size R (say $R \leq M(S-1)+1$) is fixed and the LP (8) (with $\xi_r \in \Xi^k$) is solved to get an optimal plan π^k, which is in turn fixed in the optimization problem $\min_\xi \sum_{m=1}^M \alpha_m \sum_{r=1}^R \sum_{s=1}^{S^{(m)}} \|\xi_r - \zeta_s^{(m)}\|^2 \pi_{rs}^{k,(m)}$ yielding a new fixed support Ξ^{k+1}. Observe that this last problem has a straightforward solution (see for instance [16, Alg. 2] and [41, § II]). Otherwise, when the *free-support* approach is taken, computing a WB amounts to solve (8) by considering all points $\xi \in \Xi$, thus yielding an LP of astronomical size. Hence, whether an exact (free-support) or inexact (fixed-support) approach is employed to compute (approximate) a WB, one invariably has to face a large-scale LP of the form (8), which fits into the structure of (1) by dropping the decision variable[2] p, setting

$$\Pi^{(m)} := \left\{ \pi^{(m)} \geq 0 : \sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)},\ s = 1, \ldots, S^{(m)} \right\},\ m = 1, \ldots, M, \tag{9}$$

and the linear subspace

$$\mathcal{B} := \left\{ \pi = (\pi^{(1)}, \ldots, \pi^{(M)}) \;\middle|\; \begin{array}{lcll} \sum_{s=1}^{S^{(1)}} \pi_{rs}^{(1)} & = & \sum_{s=1}^{S^{(2)}} \pi_{rs}^{(2)}, & r = 1, \ldots, R \\ \sum_{s=1}^{S^{(2)}} \pi_{rs}^{(2)} & = & \sum_{s=1}^{S^{(3)}} \pi_{rs}^{(3)}, & r = 1, \ldots, R \\ & \vdots & & \\ \sum_{s=1}^{S^{(M-1)}} \pi_{rs}^{(M-1)} & = & \sum_{s=1}^{S^{(M)}} \pi_{rs}^{(M)}, & r = 1, \ldots, R \end{array} \right\}. \tag{10}$$

The polytope $\Pi^{(m)}$ is composed of transportation plans $\pi^{(m)}$ with right marginals $q^{(m)}$. The set with all left marginals is characterized by the linear subspace $\mathcal{B}$ of "balanced plans".

In light of the above observations, we focus on a decomposition technique for solving LPs of the form (1) to render computing a (free or fixed support) WB possible beyond moderate-scale data inputs. We mention in passing that no assumption on the costs of (1) is required. This fact opens the way to consider, for instance, W_ι^ι Wasserstein distances with $\iota \in [1, \infty)$.

3 Discrete unbalanced Wasserstein Barycenter

A well-known drawback of formulation (4) is its limitation to measures with equal total masses, so the feasible set defining the Wasserstein distance (WD) is nonempty. To overcome this limitation, an idea is to relax the marginal constraints in (WD) to cope with "unbalanced" measures, i.e., with different masses [32]. Different manners to relax these marginal constraints yield different generalizations of the concept of Wasserstein barycenter, known in the literature by the name of *unbalanced Wasserstein barycenters* (UWBs). In this work, we propose a new formulation that uses a metric to measure the distance of a multi-plan $\pi = (\pi^{(1)}, \ldots, \pi^{(M)})$ to the balanced subspace $\mathcal{B}$ defined in eq. (10). We take such a metric as being the Euclidean distance $\texttt{dist}_{\mathcal{B}}(\pi) = \min_{\theta \in \mathcal{B}} \|\theta - \pi\|$ and define the following nonlinear optimization problem, with $\gamma > 0$ a penalty parameter, $\Pi^{(m)}$ given in (9), and $\mathcal{B}$ in (10):

$$\left\{ \begin{array}{ll} \min\limits_{\pi} & \displaystyle\sum_{m=1}^{M} \alpha_m \sum_{r=1}^{R} \sum_{s=1}^{S^{(m)}} \|\xi_r - \zeta_s^{(m)}\|^2 \pi_{rs}^{(m)} + \gamma \texttt{dist}_{\mathcal{B}}(\pi) \\ \text{s.t.} & \pi^{(m)} \in \Pi^{(m)},\ m = 1, \ldots, M. \end{array} \right. \tag{11}$$

[2] Although variable p is the one of interest, it can be removed from (8) and easily recovered thanks to the balanced subspace (10).

This problem has always a solution because the objective function is continuous and the non-empty feasible set is compact. Note that in the balanced case, problem eq. (11) is a relaxation of eq. (1). In the unbalanced setting, any feasible point to eq. (11) yields $\texttt{dist}_{\mathcal{B}}(\pi) > 0$. As this distance function is strictly convex outside $\mathcal{B}$, the above problem has a unique solution.

Definition 2 (Discrete Unbalanced Wassertein Barycenter - UWB)**.** *Given a set $\{\nu^{(1)}, \dots, \nu^{(M)}\}$ of unbalanced non-negative vectors, let $\bar{\pi} \geq 0$ be the unique solution to problem eq. (11), and $\tilde{\pi}$ the projection of $\bar{\pi}$ onto the balanced subspace $\mathcal{B}$, that is, $\tilde{\pi} := \texttt{Proj}_{\mathcal{B}}(\bar{\pi})$. The measure $\mu = \sum_{r=1}^{R} p_r \delta_{\xi_r}$ with $p_r := \sum_{s=1}^{S(m)} \tilde{\pi}_{rs}^{(m)}$, $r = 1, \dots, R$ (no matter $m \in \{1, \dots, M\}$) is defined as the γ-unbalanced Wasserstein barycenter of $\{\nu^{(1)}, \dots, \nu^{(M)}\}$.*

The above definition differs from the ones found in the literature, which relaxes the constraints $\sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)}$, see for instance [24,32]. Although the above definition is not as general as the ones of the latter references, it provides meaningful results (see Section 6.4 below), uniqueness of the barycenter (if unbalanced), and is indeed an extension of (balanced) WB as the LP (8) is for (1) what the nonlinear problem (11) is for (2).

Proposition 1. *Suppose that $\{\nu^{(1)}, \dots, \nu^{(M)}\}$ are probability measures and let $\gamma > \|c\|$ in problem eq. (2). Then $\bar{\pi}$ solves (2) if and only if $\bar{\pi}$ solves (1). In particular, any UWB according to definition 2 is also a (balanced) WB.*

Proof. Being a linear function, the objective of (1) is Lipschitz continuous with constant $\|c\|$. Thus, the standard theory of exact penalty methods in optimization (see for instance [7, Prop. 1.5.2]) ensures that, when $\gamma > \|c\|$, $\bar{\pi}$ solves problem eq. (2) if and only if $\bar{\pi}$ solves eq. (1). In particular, for $\Pi^{(m)}$ and $\mathcal{B}$ given in (9) and (10), respectively, $\bar{\pi} = \texttt{Proj}_{\mathcal{B}}(\bar{\pi})$ and the measure $\mu = \sum_{r=1}^{R} p_r \delta_{\xi_r}$ with $p_r = \sum_{s=1}^{S(m)} \bar{\pi}_{rs}^{(m)}$ (as in definition 2) solves (8). □

Another advantage of definition 2 is that the problem yielding the proposed UBW enjoys a favorable structure that can be efficiently exploited by splitting methods. Indeed, it turns out that computing a balanced or unbalanced WB can be done by the algorithm presented in Section 5.3. In the next section, we show that the computational burden to solve either the LP eq. (1) or the nonlinear problem eq. (2) by the Douglas-Rachford splitting method is the same.

4 Problem reformulation and the DR algorithm

We have recalled that computing a free or fixed-support (balanced) WB requires solving one or more LPs of the form (1), with $\Pi^{(m)}$ and $\mathcal{B}$ given in (9) and (10), respectively. Furthermore, according to our new Definition 2, computing a UWB requires solving one (or more) nonlinear problems of the form (2). In this section, we focus on problems eq. (1) and eq. (2) and reformulate them in a suitable way so that the Douglas-Rachford splitting operator method can be easily deployed to compute a discrete barycenter in the balanced and unbalanced settings. To this end, let us consider the indicator function $\mathbf{i}_C$ of a convex set C (that is $\mathbf{i}_C(x) = 0$ if $x \in C$ and $\mathbf{i}_C(x) = \infty$ otherwise) to define the convex functions

$$f^{(m)}(\pi^{(m)}) := \sum_{r=1}^{R} \sum_{s=1}^{S^{(m)}} c_{rs}^{(m)} \pi_{rs}^{(m)} + \mathbf{i}_{\Pi^{(m)}}(\pi^{(m)}), \quad m = 1, \dots, M, \tag{12}$$

and recast problems eq. (1) and eq. (2) in the following more general setting

$$\min_{\pi} \; f(\pi) + g(\pi)\,, \;\text{with}: \tag{13a}$$

$$f(\pi) := \sum_{m=1}^{M} f^{(m)}(\pi^{(m)}) \quad \text{and} \quad g(x) := \begin{cases} \mathbf{i}_{\mathcal{B}}(\pi) & \text{if balanced} \\ \gamma\, \texttt{dist}_{\mathcal{B}}(\pi) & \text{if unbalanced.} \end{cases} \tag{13b}$$

Since f is polyhedral, $\mathcal{D}\mathrm{om}(f) \cap \mathtt{ri}(\mathcal{D}\mathrm{om}(g)) \neq \emptyset$[3], and eq. (13) is solvable, it follows from [5, Thm 27.2] that computing one of its solutions is equivalent to

$$\text{find} \quad \pi \quad \text{such that} \quad 0 \in \partial f(\pi) + \partial g(\pi). \tag{14}$$

Recall that the subdifferential of a proper convex lower semicontinuous functions is a maximal monotone operator [5, Thm 20.40]. Thus, the above generalized equation is nothing but the problem of finding a zero of the sum of two maximal monotone operators, a well-understood problem for which several methods exist (see, for instance, Chapters 25 and 27 of the textbook [5]). Among the existing algorithms, the Douglas-Rachford operator splitting method [19] (see also [5, § 25.2 and § 27.2]) is the most popular one. When applied to problem eq. (14), the DR algorithm asymptotically computes a solution by repeating the following steps, with $k = 0, 1, \ldots$, given initial point $\theta^0 = (\theta^{(1),0}, \ldots, \theta^{(M),0})$ and prox-parameter $\rho > 0$:

$$\left\{ \begin{array}{rcl} \pi^{k+1} & = & \arg\min\limits_{\pi} \; g(\pi) + \dfrac{\rho}{2}\|\pi - \theta^k\|^2 \\ \hat{\pi}^{k+1} & = & \arg\min\limits_{\pi} \; f(\pi) + \dfrac{\rho}{2}\|\pi - (2\pi^{k+1} - \theta^k)\|^2 \\ \theta^{k+1} & = & \theta^k + \hat{\pi}^{k+1} - \pi^{k+1}. \end{array} \right. \tag{15}$$

By noting that f and g in eq. (13b) are proper convex lower semicontinuous functions and problem eq. (13) is solvable (so is (14) [5, Thm 27.2(ii)]), the following is a direct consequence of Theorem 25.6 and Corollary 27.4 of [5].

Theorem 2. *The sequence $\{\theta^k\}$ produced by the DR algorithm eq. (15) converges to a point $\bar{\theta}$, and the following holds: $\bar{\pi} := \arg\min_\pi \; g(\pi) + \frac{\rho}{2}\|\pi - \bar{\theta}\|^2$ solves eq. (13), and $\{\pi^k\}$ and $\{\hat{\pi}^k\}$ converge to $\bar{\pi}$.*

The DR algorithm is attractive when the two first steps in eq. (15) are convenient to execute, which is the case in our settings. As we will shortly see, the iterate π^{k+1} above has an explicit formula in both balanced and unbalanced cases, and computing $\hat{\pi}^{k+1}$ amounts to executing a series of independent projections onto the simplex. This task can be accomplished exactly and efficiently by specialized algorithms.

Since f in eq. (13b) has a separable structure, the computation of $\hat{\pi}^{k+1}$ in eq. (15) breaks down to a series of smaller and simpler subproblems as just mentioned. Hence, we may exploit such a structure by combining recent developments in DR's literature to produce the following randomized version of the DR algorithm eq. (15), with α the vector of weights in eq. (4):

$$\left\{ \begin{array}{rcl} \pi^{k+1} & = & \arg\min\limits_{\pi} \; g(\pi) + \dfrac{\rho}{2}\|\pi - \theta^k\|^2 \\ & & \text{Draw randomly } m \in \{1, 2, \ldots, M\} \text{ with probability } \alpha_m > 0 \\ \hat{\pi}^{(m),k+1} & = & \arg\min\limits_{\pi^{(m)}} \; f^{(m)}(\pi^{(m)}) + \dfrac{\rho}{2}\|\pi^{(m)} - (2\pi^{(m),k+1} - \theta^{(m),k})\|^2 \\ \theta^{(m'),k+1} & = & \left\{ \begin{array}{ll} \theta^{(m),k} + \hat{\pi}^{(m),k+1} - \pi^{(m),k+1} & \text{if } m' = m \\ \theta^{(m'),k} & \text{if } m' \neq m. \end{array} \right. \end{array} \right. \tag{16}$$

The randomized DR algorithm eq. (16) aims at reducing the computational burden and accelerating the optimization process. Such goals can be attained in some situations, depending on the underlying problem and available computational resources. The particular choice of $\alpha_m > 0$ as the probability of picking up the m^{th} subproblem is

[3] $\mathtt{ri}$ denotes the relative interior of a set.

not necessary for convergence: the only requirement is that every subproblem is picked up with a fixed and positive probability. The intuition behind our choice is that measures that play a more significant role in eq. (4) (i.e., higher α_m) should have more chance to be picked by the randomized DR algorithm. Furthermore, the presentation above where only one measure (subproblem) in eq. (16) is drawn is made for the sake of simplicity. One can perfectly split the set of measures into $nb < M$ bundles, each containing a subset of measures, and select randomly bundles instead of individual measures. Such an approach proves advantageous in a parallel computing environment with nb available machines/processors (see section 6.2.3 in the numerical section). The almost surely (i.e., with probability one) convergence of the randomized DR algorithm depicted in eq. (16) can be summarized as follows [25, Thm 2].

Theorem 3. *The sequence $\{\pi^k\}$ produced by the randomized DR algorithm eq. (16) converges almost surely to a random variable $\bar{\pi}$ taking values in the solution set of problem eq. (13).*

This result is a special case of a thorough analysis given in [13] (see, in particular, Remark 3.5 and Section 5 in that paper.) We note that the practical performance of the randomized scheme (16) depends on computational resources and is thus not always effective (see Figure 6.2.3 below). The deterministic and asynchronous decomposition methods in [12] provide significantly more flexibility in selecting the measure $\nu^{(m)}$ (or even part of it) activated at every iteration and thus may perform better than the randomized scheme above. As these methods do not follow the general lines of the DR algorithm, we leave the specialization of such approaches to the WB problem for future research.

In the next section, we further exploit the structure of functions f and g in eq. (13) and rearrange terms in the schemes eq. (15) and eq. (16) to provide an easy-to-implement and memory-efficient algorithm for computing balanced and unbalanced WBs.

5 The Method of Averaged Marginals

Both deterministic and randomized DR algorithms above require evaluating the proximal mapping of the function g given in eq. (13b). In the balanced WB setting, g is the indicator function of $\mathcal{B}$ given in eq. (10), and thus π^{k+1} in (15) is the projection of θ^k onto $\mathcal{B}$: $\pi^{k+1} = \texttt{Proj}_{\mathcal{B}}(\theta^k)$. On the other hand, in the unbalanced WB case, $g(\cdot)$ is the penalized distance function $\gamma\,\texttt{dist}_{\mathcal{B}}(\cdot)$. Computing π^{k+1} then amounts to evaluating the proximal mapping of the distance function: $\min_\pi\ \texttt{dist}_{\mathcal{B}}(\pi) + \frac{\rho}{2\gamma}\|\pi - \theta^k\|^2$. The unique solution to this problem is given by [5, Example 24.28]

$$\pi^{k+1} = \begin{cases} \texttt{Proj}_{\mathcal{B}}(\theta^k) & \text{if } \ \rho\,\texttt{dist}_{\mathcal{B}}(\theta^k) \le \gamma \\ \theta^k + \frac{\gamma}{\rho\,\texttt{dist}_{\mathcal{B}}(\theta^k)}(\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k) & \text{otherwise.} \end{cases} \tag{17}$$

Hence, computing π^{k+1} in both balanced and unbalanced settings boils down to projecting onto the balanced subspace. This fact allows us to provide a unified algorithm for WB and UWB problems.

5.1 Projecting onto the subspace of balanced plans

In what follows we exploit the particular geometry of $\mathcal{B}$ to provide an explicit formula for projecting onto this linear subspace.

Proposition 2. *With the notation of Section 2, let $\theta \in \mathbb{R}^{R\times T}$,*

$$a_m := \frac{\frac{1}{S^{(m)}}}{\sum_{j=1}^{M}\frac{1}{S^{(j)}}}, \quad p^{(m)} := \left(\sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m)}\right)_{1\le r\le R}, \quad \textit{and} \quad p := \sum_{m=1}^{M} a_m p^{(m)}. \tag{18a}$$

The projection $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$ has the explicit form:

$$\pi_{rs}^{(m)} := \theta_{rs}^{(m)} + \frac{(p_r - p_r^{(m)})}{S^{(m)}}, \quad s = 1, \dots, S^{(m)},\ r = 1, \dots, R,\ m = 1, \dots, M. \tag{18b}$$

Proof. First, observe that $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$ solves the QP problem

$$\begin{cases} \min\limits_{y^{(1)},\dots,y^{(M)}} & \frac{1}{2}\sum\limits_{m=1}^{M} \|y^{(m)} - \theta^{(m),k}\|^2 \\ \text{s.t} & \sum_{s=1}^{S^{(m)}} y_{rs}^{(m)} = \sum_{s=1}^{S^{(m+1)}} y_{rs}^{(m+1)}, \quad r = 1, \dots, R,\ m = 1, \dots, M-1, \end{cases} \tag{19}$$

which is only coupled by the "columns" of π: there is no constraint linking $\pi_{rs}^{(m)}$ with $\pi_{r's}^{(m')}$ for $r \neq r'$ and m and m' arbitrary. Therefore, we can decompose it by rows: for $r = 1, \dots, R$, the r^{th}-row $(\pi_{r1}^{(1)}, \dots, \pi_{rS^{(1)}}^{(1)}, \dots, \pi_{r1}^{(M)}, \dots, \pi_{rS^{(M)}}^{(M)})$ of π is the unique solution to the problem

$$\begin{cases} \min\limits_{w} & \frac{1}{2}\sum\limits_{m=1}^{M}\sum\limits_{s=1}^{S^{(m)}} \left(w_s^{(m)} - \theta_{rs}^{(m)}\right)^2 \\ \text{s.t} & \sum_{s=1}^{S^{(m)}} w_s^{(m)} = \sum_{s=1}^{S^{(m+1)}} w_s^{(m+1)},\ m = 1, \dots, M-1. \end{cases} \tag{20}$$

The Lagrangian function to this problem is, for a dual variable u, given by

$$L_r(w, u) = \frac{1}{2}\sum_{m=1}^{M}\sum_{s=1}^{S^{(m)}} \left(w_s^{(m)} - \theta_{rs}^{(m)}\right)^2 + \sum_{m=1}^{M-1} u^{(m)} \Big(\sum_{s=1}^{S^{(m)}} w_s^{(m)} - \sum_{s=1}^{S^{(m+1)}} w_s^{(m+1)} \Big). \tag{21}$$

A primal-dual (w, u) solution to problem eq. (20) must satisfy the Lagrange system, in particular $\nabla_w L_r(w, u) = 0$ with w the r^{th} row of $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$, that is,

$$\begin{cases} \pi_{rs}^{(1)} - \theta_{rs}^{(1)} & + & u^{(1)} & = 0 & s = 1, \dots, S^{(1)} \\ \pi_{rs}^{(2)} - \theta_{rs}^{(2)} & + & u^{(2)} - u^{(1)} & = 0 & s = 1, \dots, S^{(2)} \\ & \vdots & & & \\ \pi_{rs}^{(M-1)} - \theta_{rs}^{(M-1)} & + & u^{(M-1)} - u^{(M-2)} & = 0 & s = 1, \dots, S^{(M-1)} \\ \pi_{rs}^{(M)} - \theta_{rs}^{(M)} & - & u^{(M-1)} & = 0 & s = 1, \dots, S^{(M)}. \end{cases} \tag{22}$$

Let us denote $p_r = \sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)}$ (no matter $m \in \{1, \dots, M\}$ because $\pi \in \mathcal{B}$), $p_r^{(m)} = \sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m)}$ (the r^{th} component of $p^{(m)}$ as defined in eq. (18a)), and sum over s the first row of system eq. (22) to get

$$p_r - p_r^{(1)} + u^{(1)} S^{(1)} = 0 \quad \Rightarrow \quad u^{(1)} = \frac{p_r^{(1)} - p_r}{S^{(1)}}, \tag{23}$$

Now, by summing the second row in eq. (22) over s we get

$$p_r - p_r^{(2)} + u^{(2)} S^{(2)} - u^{(1)} S^{(2)} = 0 \quad \Rightarrow \quad u^{(2)} = u^{(1)} + \frac{p_r^{(2)} - p_r}{S^{(2)}}. \tag{24}$$

By proceeding in this way and setting $u^{(0)} := 0$ we obtain

$$u^{(m)} = u^{(m-1)} + \frac{p_r^{(m)} - p_r}{S^{(m)}}, \quad m = 1, \dots, M-1. \tag{25a}$$

Furthermore, for $M-1$ we get the alternative formula $u^{(M-1)} = -\frac{p_r^{(M)} - p_r}{S^{(M)}}$. Given these dual values, we can use eq. (22) to conclude that the r^{th} row of $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$ is given as in eq. (18b). It is remaining to show that $p_r = \sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)}$, as defined above, is alternatively given by eq. (18a). To this end, observe that $u^{(M-1)} = u^{(M-1)} - u^{(0)} = \sum_{m=1}^{M-1}(u^{(m)} - u^{(m-1)})$, so:

$$u^{(M-1)} = \sum_{m=1}^{M-1} \left(\frac{p_r^{(m)} - p_r}{S^{(m)}} \right) = \sum_{m=1}^{M-1} \frac{p_r^{(m)}}{S^{(m)}} - p_r \sum_{m=1}^{M-1} \frac{1}{S^{(m)}}. \tag{26}$$

Recall that $u^{(M-1)} = \frac{p_r - p_r^{(M)}}{S^{(M)}}$, i.e., $p_r = p_r^{(M)} + u^{(M-1)} S^{(M)}$. Replacing $u^{(M-1)}$ with the expression eq. (26) yields

$$p_r = S^{(M)} \left[\frac{p_r^{(M)}}{S^{(M)}} + u^{(M-1)} \right] = S^{(M)} \left[\frac{p_r^{(M)}}{S^{(M)}} + \sum_{m=1}^{M-1} \frac{p_r^{(m)}}{S^{(m)}} - p_r \sum_{m=1}^{M-1} \frac{1}{S^{(m)}} \right], \tag{27}$$

which implies $p_r \sum_{m=1}^{M} \frac{1}{S^{(m)}} = \sum_{m=1}^{M} \left(\frac{p_r^{(m)}}{S^{(m)}} \right)$. Hence, p is as given in eq. (18a), and the proof is complete. □

Note that projection can be computed in parallel over the rows, and the average p of the marginals $p^{(m)}$ is the gathering step between parallel processors.

5.2 Evaluating the Proximal Mapping of Transportation Costs

In this subsection we turn our attention to the DR algorithm's second step, which requires solving a convex optimization problem of the form: $\min_\pi \; f(\pi) + \frac{\rho}{2}\|\pi - y\|^2$ (see eq. (15)). Given the additive structure of f in eq. (13b), the above problem can be decomposed into M smaller ones

$$\min_{\pi^{(m)}} \; f^{(m)}(\pi^{(m)}) + \frac{\rho}{2}\|\pi^{(m)} - y^{(m)}\|^2, \quad m = 1, \dots, M. \tag{28}$$

Then looking closely at every subproblem above, we can see that we can decompose it even more: the columns of the the transportation plan $\pi^{(m)}$ are independent in the minimization. Besides, as the following result shows, every column optimization is simply the projection of an R-dimensional vector onto the simplex Δ_R.

Proposition 3. *Let $\Delta_R(\tau)$ be as in eq. (3). The minimization $\hat{\pi} := \min_\pi \; f(\pi) + \frac{\rho}{2}\|\pi - y\|^2$ can be performed exactly, in parallel along the columns of each transport plan $y^{(m)}$, as follows: for all $m \in \{1, \dots, M\}$,*

$$\begin{pmatrix} \hat{\pi}_{1s}^{(m)} \\ \vdots \\ \hat{\pi}_{Rs}^{(m)} \end{pmatrix} = \texttt{Proj}_{\Delta_R(q_s^{(m)})} \begin{pmatrix} y_{1s} - \frac{1}{\rho} c_{1s}^{(m)} \\ \vdots \\ y_{Rs} - \frac{1}{\rho} c_{Rs}^{(m)} \end{pmatrix}, \quad s = 1, \dots, S^{(m)}. \tag{29}$$

Proof. It has already been argued that evaluating this proximal mapping into M smaller subproblems eq. (28), which is a quadratic program problem due to the definition of $f^{(m)}$ in eq. (12):

$$\min_{\pi^{(m)} \geq 0} \sum_{r=1}^{R} \sum_{s=1}^{S^{(m)}} \left[c_{rs}^{(m)} \pi_{rs}^{(m)} + \frac{\rho}{2} \left| \pi_{rs}^{(m)} - y_{rs}^{(m)} \right|^2 \right] \quad \text{s.t.} \quad \sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)}, \; s = 1, \dots, S^{(m)}. \tag{30}$$

By taking a close look at the above problem, we can see that the objective function is decomposable, and the constraints couple only the "rows" of $\pi^{(m)}$. Therefore, we can go further and decompose the above problem per columns: for $s = 1, \dots, S^{(m)}$, the s^{th}-column of $\hat{\pi}^{(m)}$ is the unique solution to the R-dimensional problem

$$\min_{w \geq 0} \sum_{r=1}^{R} \left[c_{rs}^{(m)} w_r + \frac{\rho}{2} (w_r - y_{rs}^{(m)})^2 \right] \quad \text{s.t.} \quad \sum_{r=1}^{R} w_r = q_s^{(m)}, \tag{31}$$

which is nothing but eq. (29). Such projection can be performed exactly [14]. □

Remark 1. *If $\tau = 0$, then $\Delta_R(\tau) = \{0\}$ and the projection onto this set is trivial. Otherwise, $\tau > 0$ and computing $\texttt{Proj}_{\Delta_R(\tau)}(w)$ amounts to projecting onto the $R+1$ simplex Δ_R: $\texttt{Proj}_{\Delta_R(\tau)}(w) = \tau \texttt{Proj}_{\Delta_R}(w/\tau)$. The latter task can be performed* exactly *by using efficient methods [14]. Hence, evaluating the proximal mapping in proposition 3 decomposes into T independent projections onto Δ_R.*

5.3 The Method of Averaged Marginals (MAM)

Our approach is presented in Algorithm 1, which gathers the three main steps from the DR algorithm and integrates a choice of γ for a simple switch between the balanced and unbalanced cases.

MAM's interpretation At every iteration, the barycenter approximation p^k is a weighted average of the M marginals $p^{(m)}$ of the plans $\theta^{(m),k}$, $m = 1, \dots, M$. As we will shortly see, the whole sequence $\{p^k\}$ converges (almost surely or deterministically) to a barycenter upon specific assumptions on the choice of the index set at line 6 of algorithm 1.

Initialization The choices for $\theta^0 \in \mathbb{R}^{R \times T}$ and $\rho > 0$ are arbitrary ones. The prox-parameter $\rho > 0$ is borrowed from the DR algorithm, which is known to have an impact on the practical convergence speed. Therefore, ρ should be tuned for the set of distributions at stakes. Some heuristics for tuning this parameter exist for other methods derived from the DR algorithms [38, 39] and can be adapted to the setting of algorithm 1.

Stopping criteria A possible stopping test is $\|\theta^{k+1} - \theta^k\|_\infty \leq \texttt{Tol}$, where $\texttt{Tol} > 0$ is a given tolerance. Alternatively, we may stop the algorithm when $\|p^{k+1} - p^k\| \leq \texttt{Tol}$. or $\text{dist}_B(\hat{\pi}^k) \leq \texttt{Tol}$. These latter tests should be understood as heuristic criteria.

Deterministic and random variants of MAM The most computationally expensive step of MAM is Step 2, which requires a series of independent projections onto the $R+1$ simplex (see remark 1). Our approach underlines that this step can be conducted in parallel over s or, if preferable, over the measures m. As a result, it is a natural idea to derive a randomized variant of the algorithm. This is the reason for having the possibility of choosing an index set $\mathcal{M}^k \subsetneq \{1, \dots, M\}$ at line 6 of algorithm 1. For example, we may employ an economical rule and choose $\mathcal{M}^k = \{m\}$ randomly (with a fixed and positive probability, e.g. α_m) at every iteration, or the costly one $\mathcal{M}^k = \{1, \dots, M\}$ for all k. The latter yields the deterministic method of averaged marginals, while the former gives rise to a randomized variant of MAM. Depending on the computational resources, intermediate choices between these two extremes can perform better in practice.

Remark 2. *Suppose that $1 < \texttt{nb} < M$ processors are available. We may then create a partition $A_1, \dots, A_{\texttt{nb}}$ of the set $\{1, \dots, M\}$ $(= \cup_{i=1}^{\texttt{nb}} A_i)$ and define weights $\beta_i := \sum_{m \in A_i} \alpha_m > 0$. Then, at every iteration k, we may draw with probability β_i the subset A_i of measures and set $\mathcal{M}^k = A_i$.*

Algorithm 1 Method of Averaged Marginals - MAM

▷ Step 0: input

1: Given $\rho > 0$, the cost matrix and initial point $c, \theta^0 \in \mathbb{R}^{R\times T}$, and $a \in \Delta_M$ as in eq. (18a), set $k \leftarrow 0$ and $p_r^{(m)} \leftarrow \sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m),0}$, $r = 1, \dots, R$, $m = 1, \dots, M$

2: Set $\gamma \leftarrow \infty$ if $q^{(m)} \in \mathbb{R}_+^{S^{(m)}}$, $m = 1, \dots, M$, are balanced; otherwise, choose $\gamma \in [0, \infty)$

3: **while** not converged **do**

▷ Step 1: average the marginals

4: Compute $p^k \leftarrow \sum_{m=1}^M a_m p^{(m)}$

5: Set $t^k = 1$ if $\rho \sqrt{\sum_{m=1}^M \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}} \leq \gamma$; otherwise, $t^k \leftarrow \gamma / \left(\rho \sqrt{\sum_{m=1}^M \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}} \right)$

6: Choose an index set $\emptyset \neq \mathcal{M}^k \subseteq \{1, \dots, M\}$

7: **for** $m \in \mathcal{M}^k$ **do**

▷ Step 2: update the m^{th} plan

8: **for** $s = 1, \dots, S^{(m)}$ **do**

9: Define $w_r \leftarrow \theta_{rs}^{(m),k} + 2\, t^k \frac{p_r^k - p_r^{(m)}}{S^{(m)}} - \frac{1}{\rho} c_{rs}^{(m)}$, $r = 1, \dots, R$

10: Compute $(\hat{\pi}_{1s}^{(m)}, \dots, \hat{\pi}_{Rs}^{(m)}) \leftarrow \texttt{Proj}_{\Delta_R(q_s^{(m)})}(w)$

11: Update $\theta_{rs}^{(m),k+1} \leftarrow \hat{\pi}_{rs}^{(m)} - t^k \frac{p_r^k - p_r^{(m)}}{S^{(m)}}$, $r = 1, \dots, R$

12: **end for**

▷ Step 3: update the m^{th} marginal

13: Update $p_r^{(m)} \leftarrow \sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m),k+1}$, $r = 1, \dots, R$

14: **end for**

15: **end while**

16: Return $\bar{p} \leftarrow p^k$

This randomized variant would enable the algorithm to compute more iterations per time unit but with less precision per iteration (since not all the marginals $p^{(m)}$ are updated). Such a randomized variant of MAM is benchmarked against its deterministic counterpart in Section 6.2.3, where we demonstrate empirically that with certain configurations (depending on the number M of probability distributions and the number of processors) this randomized algorithm can be effective. We highlight that other choices for $\mathcal{M}^k$ rather than randomized ones or the deterministic rule $\mathcal{M}^k = \{1, \dots, M\}$ should be understood as heuristics. Within such a framework, one may choose $\mathcal{M}^k \subsetneq \{1, \dots, M\}$ deterministically, for instance cyclically or yet by the discrepancy of the marginal $p^{(m)}$ with respect to the average p^k.

Storage complexity Note that the operation at line 10 is trivial if $q_s^{(m)} = 0$. This motivates us to remove all the zero components of $q^{(m)}$ from the problem's data, and consequently, all the columns s of the cost matrix $c^{(m)}$ and variables $\theta, \hat{\pi}$ corresponding to $q_s^{(m)} = 0$, $m = 1, \dots, M$. In some applications (e.g. general sparse problems), this strategy significantly reduces the WB problem and thus memory allocation, since the non-taken columns are both not stored and not treated in the *for loops*. This remark raises the question of how sparse data impacts the practical performance of MAM. Section 6.1 conducts an empirical analysis on this matter.

In nominal use, the algorithm needs to store the decision variables $\theta^{(m)} \in \mathbb{R}^{R\times S^{(m)}}$ for all $m = 1,\dots,M$ (transport plans for every measure), along with M distance matrices $c \in \mathbb{R}^{R\times S^{(m)}}$, one barycenter approximation $p^k \in \mathbb{R}^R$, M approximated marginals $p^{(m)} \in \mathbb{R}^R$ and M marginals $q^{(m)} \in \mathbb{R}^{S(m)}$. Note that in practical terms, the auxiliary variables w and $\hat{\pi}$ in algorithm 1 can be easily removed from the algorithm's implementation by merging lines 9-11 into a single one. Hence, by letting $T = \sum_{m=1}^{M} S^{(m)}$, the method's memory allocation is $2RT + T + M(R+1)$ floating-points. This number can be reduced if the measures share the same cost matrix, i.e., $c^{(m)} = c^{(m')}$ for all $m, m' = 1,\dots,M$. In this case, $S^{(m)} = S$ for all m, $T = M\,S$ and the method's memory allocation drops to $RT + R\,S + T + M(R+1)$ floating-points. In the light of the previous remark this memory complexity should be treated as an upper bound: the sparser the data the less memory will be needed.

Computation complexity Step 2 of the algorithm involves two main components: projection onto $\mathcal{B}$, which comprises straightforward operations detailed in Section 5.1, and projection onto Π, which relies on leveraging the simplex projection technique discussed in Section 5.2. For each probability measure, the projection onto $\mathcal{B}$ requires $3RS^{(m)}$ computation operations, where R is the barycenter support size and $S^{(m)}$ denotes the support size of the probability measure m, which undergoes iteration over its columns. On the other hand, the simplex projection (line 10) is computationally more intensive. This is due to the adoption of a state-of-the-art algorithm proposed by Condat [14], which operates in $\mathrm{O}(R\log(R))$. Therefore, the complexity of $S^{(m)}$ times line 10 amounts to $\mathrm{O}(S^{(m)}R\log(R))$. Deriving the precise number of computational operations is challenging due to the use of a sorting algorithm in [14], the complexity of which depends on the characteristics of the input data, hence the asymptotic complexity estimation. Note that Step 2 must be executed for the M probability measures, but these operations can be performed in parallel (multiprocessing).

Balanced and unbalanced settings As already mentioned, our approach can handle both balanced and unbalanced WB problems. All that is necessary is to choose a finite (positive) value for the parameter γ in the unbalanced case. Such a parameter is only used to define $t^k \in (0,1]$ at every iteration. Indeed, algorithm 1 defines $t^k = 1$ for all iterations if the WB problem is balanced (because $\gamma = \infty$ in this case)[4], and $t^k = \gamma/\left(\rho\sqrt{\sum_{m=1}^{M}\frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}}\right)$ otherwise. This rule for setting up t^k is a mere artifice to model eq. (17). Indeed, $\texttt{dist}_{\mathcal{B}}(\theta^k) = \|\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k\|$ reduces to $\sqrt{\sum_{m=1}^{M}\frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}}$ thanks to proposition 2.

Convergence analysis The convergence analysis of algorithm 1 can be summarized as follows.

Theorem 4 (MAM's convergence analysis)**.** *a) (Deterministic MAM.) Consider algorithm 1 with the choice $\mathcal{M}^k = \{1,\dots,m\}$ for all k. Then the sequence of points $\{p^k\}$ generated by the algorithm converges to a point $\bar{p}$. If the measures are balanced, then $\bar{p}$ is a balanced WB; otherwise, $\bar{p}$ is a γ-unbalanced WB.*

b) (Randomized MAM.) Consider algorithm 1 with the choice $\mathcal{M}^k \subset \{1,\dots,m\}$ as in remark 2. Then the sequence of points $\{p^k\}$ generated by the algorithm converges almost surely to a point $\bar{p}$. If the measures are balanced, then $\bar{p}$ is almost surely a balanced WB; otherwise, $\bar{p}$ is almost surely a γ-unbalanced WB.

Proof. It suffices to show that algorithm 1 is an implementation of the (randomized) DR algorithm and invoke theorem 2 for item a) and theorem 3 for item b). To this end, we first rely on proposition 2 to get that the projection of θ^k onto the balanced subspace $\mathcal{B}$ is given by $\theta^{(m),k}_{rs} + \frac{(p^k_r - p^{(m)}_r)}{S^{(m)}}$, $s = 1,\dots,S^{(m)}$, $r = 1,\dots,R$, $m = 1,\dots,M$, where p^k is computed at Step 1 of the algorithm, and the marginals $p^{(m)}$ of θ^k are computed at Step 0 if $k = 0$

[4]Observe that line 5 can be entirely disregarded in this case, by setting $t^k = t = 1$ fixed at initialization.

or at Step 3 otherwise. Therefore, $\texttt{dist}_{\mathcal{B}}(\theta^k) = \|\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k\| = \sqrt{\sum_{m=1}^{M} \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}}$. Now, given the rule for updating t^k in algorithm 1 we can define the auxiliary variable π^{k+1} as $\pi^{k+1} = \theta^k + t^k(\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k)$, or alternatively,

$$\pi_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + t^k \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}, \quad s = 1, \dots, S^{(m)},\ r = 1, \dots, R,\ m = 1, \dots, M. \tag{32}$$

In the balanced case, $t^k = 1$ for all k (because $\gamma = \infty$) and thus π^{k+1} is as in eq. (18b). Otherwise, π^{k+1} is as in eq. (17) (see the comments after algorithm 1). In both cases, π^{k+1} coincides with the auxiliary variable at the first step of the DR scheme eq. (15) (see the developments at the beginning of this section). Next, observe that to perform the second step of eq. (15) we need to assess $y = 2\pi^{k+1} - \theta^k$, which is thanks to the above formula for π^{k+1} given by $y_{rs}^{(m)} = \theta_{rs}^{(m),k} + 2\,t^k \frac{p_r^k - p_r^{(m)}}{S^{(m)}}$, $s = 1, \dots, S^{(m)}$, $r = 1, \dots, R$, $m = 1, \dots, M$.

As a result, for the choice $\mathcal{M}^k = \{1, \dots, M\}$ for all k, Step 2 of algorithm 1 yields, thanks to proposition 3, $\hat{\pi}^{k+1}$ as at the second step of eq. (15). Furthermore, the updating of θ^{k+1} in the latter coincides with the rule in algorithm 1: for $s = 1, \dots, S^{(m)}$, $r = 1, \dots, R$, and $m = 1, \dots, M$,

$$\begin{aligned}\theta_{rs}^{(m),k+1} &= \theta_{rs}^{(m),k} + \hat{\pi}_{rs}^{(m),k+1} - \pi_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + \hat{\pi}_{rs}^{(m),k+1} - \left(\theta_{rs}^{(m),k} + t^k \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}\right) \\ &= \hat{\pi}_{rs}^{(m),k+1} - t^k \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}.\end{aligned}$$

Hence, for the choice $\mathcal{M}^k = \{1, \dots, M\}$ for all k, algorithm 1 is the DR Algorithm eq. (15) applied to the WB eq. (13). Theorem 2 thus ensures that the sequence $\{\pi^k\}$ as defined above converges to some $\bar{\pi}$ solving eq. (13). To show that $\{p^k\}$ converges to a barycenter, let us first use the property that $\mathcal{B}$ is a linear subspace to obtain the decomposition $\theta = \texttt{Proj}_{\mathcal{B}}(\theta) + \texttt{Proj}_{\mathcal{B}^\perp}(\theta)$ that allows us to rewrite the auxiliary variable π^{k+1} differently:

$$\pi^{k+1} = \theta^k + t^k(\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k) = \theta^k - t^k \texttt{Proj}_{\mathcal{B}^\perp}(\theta^k).$$

Let us denote $\tilde{\pi}^{k+1} := \texttt{Proj}_{\mathcal{B}}(\pi^{k+1})$. Then $\tilde{\pi}^{k+1} = \texttt{Proj}_{\mathcal{B}}(\theta^k - t^k \texttt{Proj}_{\mathcal{B}^\perp}(\theta^k)) = \texttt{Proj}_{\mathcal{B}}(\theta^k)$, and thus proposition 2 yields

$$\tilde{\pi}_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + \frac{p_r^k - p_r^{(m)}}{S^{(m)}} \quad s = 1, \dots, S^{(m)},\ r = 1, \dots, R,\ m = 1, \dots, M,$$

which in turn gives (by recalling that $\sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m),k} = p_r^{(m)}$): $\sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m),k+1} = p_r^k$, $r = 1, \dots, R$, $m = 1, \dots, M$. As $\lim_{k\to\infty} \pi^k = \bar{\pi}$, $\lim_{k\to\infty} \tilde{\pi}^k = \lim_{k\to\infty} \texttt{Proj}_{\mathcal{B}}(\pi^k) = \texttt{Proj}_{\mathcal{B}}(\bar{\pi}) =: \tilde{\pi}$. Therefore, for all $r = 1, \dots, R$, $m = 1, \dots, M$, the following limits are well defined:

$$\bar{p}_r := \sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m)} = \lim_{k\to\infty} \sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m),k+1} = \lim_{k\to\infty} p_r^k. \tag{33}$$

We have shown that the whole sequence $\{p^k\}$ converges to $\bar{p}$. By recalling that $\bar{\pi}$ solves eq. (13), we conclude that in the balanced setting $\tilde{\pi} = \bar{\pi}$ and thus $\bar{p}$ is a WB. On the other hand, in the unbalanced setting, $\bar{p}$ above is a γ-unbalanced WB according to definition 2.

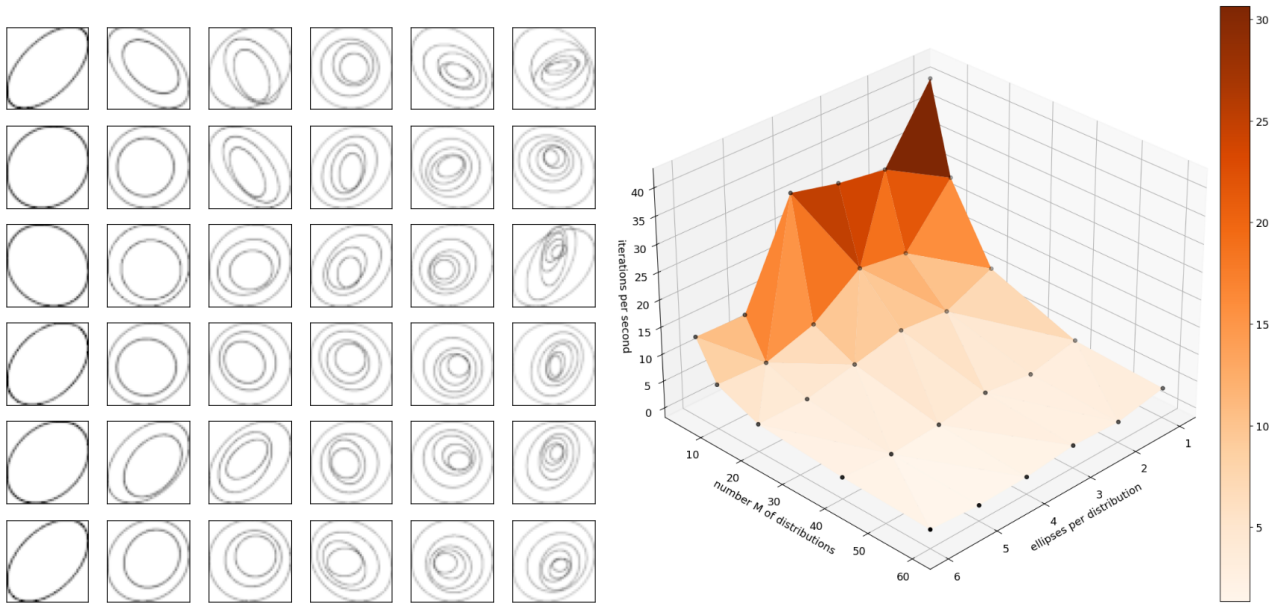
The proof of item b) is a verbatim copy of the above proof: the sole difference, given the assumptions on the choice of $\mathcal{M}^k$, is that we need to rely on theorem 3 (and not on theorem 2 as previously done) to conclude that $\{\pi^k\}$ converges almost surely to some $\bar{\pi}$ solving eq. (13). Thanks to the continuity of the orthogonal projection onto the subspace $\mathcal{B}$, the limits above yield almost surely convergence of $\{p^k\}$ to a barycenter $\bar{p}$. □

6 Numerical Experiments

This section illustrates the MAM's practical performance on some well-known datasets. The impact of different data structures is studied before the algorithm is compared to state-of-the-art methods. This section closes with an illustrative example of MAM to compute UWBs. Numerical experiments were conducted using 20 cores (*Intel(R) Xeon(R) Gold 5120 CPU*) and *Python 3.9*. The test problems and solvers' codes are available for download in the link `https://ifpen-gitlab.appcollaboratif.fr/detocs/mam_wb`.

6.1 Study on data structure influence

We start by evaluating the impact of conditions that influence the storage complexity and the algorithm performance. The main conditions are the *sparsity* of the data and the *number of distributions* M. Naturally, the denser the distributions or the more distributions are treated, the greater the storage. In these configurations, the time per iteration grows because the number of projects onto the simplex increases. To assess the impact of data sparsity and the number of measures on the algorithm's performance, we consider a fixed-support approach and experiment on datasets inspired by [6, 16]. The number of nested ellipses controls the density of a dataset: as exemplified in section 6.1(a) and section 6.1, measures with only a single ellipse are very sparse. In contrast, a dataset with 5 nested ellipses is denser. In this first experiment, we apply MAM with $\rho = 100$ (without proper tuning) for every



(a) A sample of datasets

(b) MAM's number of iterations per second

Figure 1: (a) Sample of the artificial nested ellipses datasets. The first column is taken from the first dataset with 1 ellipse, the second column from the second dataset with 2 nested ellipses, and the sixth column with 6 nested ellipses. (b) Evolution of the number of iterations per second depending on the density or the number of distributions.

dataset. Only a single processor was considered to avoid CPU communication management. Figure 6.1(b) shows

Table 1: Mean density with the number of nested ellipses. The density has been calculated by averaging the ratio of non-null pixels per image over 100 generated pictures for each dataset sharing the same number of nested ellipses.

Number of ellipses	1	2	3	4	5	6
Density (%)	29.0	51.4	64.3	70.9	73.5	75.0

that, as expected, the execution time of an iteration increases with increasing density and number of measures. The number of measures influences the method's speed more than density (this phenomenon can be due to the *numpy* matrix management). This means the quantity of information in each measure does not seem to make the algorithm less efficient in terms of speed. Such a result is to be put in regard with algorithms such as B-ADMM [41] that are particularly shaped for sparse datasets but less efficient for denser ones. Section 6.2.4 develops this further. Additionally, it is worth noting that the proposed method can harness parallel computation, enabling the distribution of work across the M measures. This approach effectively mitigates the impact of the measure count on computational efficiency.

The growing dimensions of images have an impact on the computation time, as seen in Section 5.3. For example, when treating dense $K \times K$ images for a fixed support problem, the number of operations per probability density for the projection onto $\mathcal{B}$ is $O_1^{\mathcal{B}} = 3 \cdot K^2 \cdot K^2 = 3 \cdot K^4$ and onto the simplex $O_1^{\Delta} = K^2 \cdot K^2 \cdot \log(K^2) = 2K^4 \cdot \log(K)$.

- For a fixed-support problem with dense $(nK) \times (nK)$ images, $O_{nK}^{\mathcal{B}} = 3 \cdot (nK)^4 = n^4 \cdot O_1^{\mathcal{B}}$ and $O_{nK}^{\Delta} = n^4 K^4 \log(n^2 K^2) \approx n^4 \cdot O_1^{\Delta}$.
- For a fixed-support problem with dense $K \times \cdots \times K = K^d$ measures, $O_{K^d}^{\mathcal{B}} = 3 \cdot (K^d)^2 = K^{2d-4} \cdot O_1^{\mathcal{B}}$ and $O_{K^d}^{\Delta} = (K^d)^2 \log(K^d) = \frac{d}{2} K^{2d-4} \cdot O_1^{\Delta}$.
- For a free-support problem, in dimension d, with dense K^d grids, the size of the support R depends on the number M of treated measures, $R = ((K-1)M+1)^d$. Following the details of Section 5.3, $O_{free,K^d}^{\mathcal{B}} = 3((K-1)M+1)^d K^d \approx M^d K^{2d-4} O_1^{\mathcal{B}}$ and $O_{free,K^d}^{\Delta} = ((K-1)M+1)^d K^d \log(((K-1)M+1)^d) \approx \frac{d}{2} M^d K^{2d-4} \cdot O_1^{\Delta}$.

For instance, for a fixed-support problem with 40×40 images (see section 6.1), the algorithm computes the projections for one measure in an average time of 0.01 seconds. However, for the free-support problem formulation with this dataset of 6 images, it takes 6 seconds per measure. Similarly, for a fixed-support problem with $40 \times 40 \times 40$ objects (ellipsoids with similar properties as in section 6.1 in 3D), the projections for one measure take 16 seconds.

6.2 Fixed-support approach

This section focuses on the fixed-support approach: R in (8) is equal to K^2, the number of pixels of a $K \times K$ image.

6.2.1 Comparison with IBP

The Iterative Bregman Projection (IBP) [6] is a well-known algorithm for computing Wasserstein barycenters. As mentioned in the Introduction, IBP employs a regularizing function parameterized by $\lambda > 0$, which impacts precision and must kept at a moderate magnitude to avoid numerical errors (double-precision overflow). The experiment below sheds light on the differences between MAM and IBP and their advantages depending on the use. Our IBP code is inspired by the original `MATLAB` code by G. Peyré[5].

[5] https://github.com/gpeyre/2014-SISC-BregmanOT

6.2.2 Qualitative comparison

Here, we use 100 images per digit of the MNIST database [35], where each digit has been randomly translated and rotated. Each image has 40×40 pixels and can be treated as probability distributions after normalization. Section 6.2.2 displays intermediate solutions for digits $3, 4, 5$ at different time steps both for MAM and IBP. For the two methods, the hyperparameters have been tuned: for instance, $\lambda = 1700$ is the greatest lambda that enables IBP to compute the barycenter of the 3's dataset without double-precision overflow error. Regarding MAM, a range of values for $\rho > 0$ have been tested for 100 seconds of execution, to identify which one provides good performance (for example, $\rho = 50$ for the dataset of 3's). Section 6.2.2 shows that, for each dataset, IBP gets quickly to a

Figure 2: (top) For each digit 36 out of the 100 scaled, translated, and rotated images considered for each barycenter. (bottom) Barycenters after $t = 10, 50, 500, 1000, 2000$ seconds, where the left-hand-side is the IBP evolution of its barycenter approximation, the middle panel is MAM's evolution using 10 CPU and the right-hand-side is a solution computed by applying *Gurobi* to the LP eq. (8).

stable barycenter approximation. Such a point is obtained shortly after with MAM (less than 10 seconds after).

However, MAM continues to move towards a sharper solution. It is clear that the more CPUs used for MAM, the better. Furthermore, while IBP is not well-suitable for CPU parallelization [6,27,41], MAM offers a clear advantage depending on the hardware at stake.

6.2.3 Quantitative comparison

Next, we benchmark MAM, randomized MAM and IBP on a dataset with 60 images per digit of the MNIST database [35], where every digit is a normalized image 40×40 pixels. First, all three methods have their hyperparameters tuned thanks to a sensitivity study as explained in Section 6.2.2. Then, at every time step an approximation of the computed barycenter is stored to compute the error $\bar{W}_2^2(p^k) - \bar{W}_2^2(p_G) := \sum_{m=1}^M \frac{1}{M} W_2^2(\mu^k, \nu^{(m)}) - \sum_{m=1}^M \frac{1}{M} W_2^2(\mu_G, \nu^{(m)}))$, where μ_G is a fixed-support barycenter computed using *Gurobi* to solve the LP eq. (8).

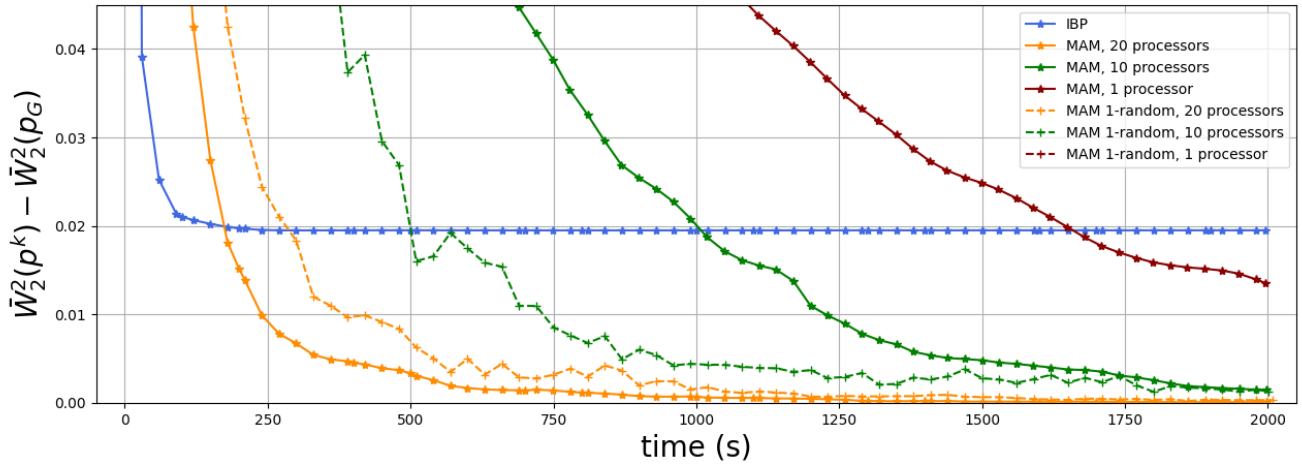


Figure 3: Evolution with respect to time of the difference between the Wasserstein barycenter distance of an approximation, $\bar{W}_2^2(p^k)$, and the Wasserstein barycentric distance of the exact solution $\bar{W}_2^2(p_G)$ given by the LP. The time step between two points is 30 seconds.

For this dataset, section 6.2.3 shows that IBP is almost 10 times faster per iteration. However, IBP computes a solution to the regularized model, not to the (fixed-support) WB linear problem (8). Instead, MAM does converge to a solution of (8). So there is a threshold where the accuracy of MAM exceeds that of IBP: in our case, around 200s - for the computation with the greatest number of processors (see section 6.2.3). Such a threshold always exists depending on the computational means (hardware). This quantitative study explains what has been exemplified with the images of Section 6.2.2: the accuracy of IBP is bounded by the choice of λ, itself bounded by an overflow error. In contrast, the MAM hyperparameter only impacts the convergence speed. For this dataset, the WB computed by IBP is within 2% of accuracy and thus reasonably good. However, as shown in Table 1 in [41], one can choose other datasets where IBP's accuracy might be unsatisfactory.

Furthermore, section 6.2.3 exemplifies an attractive asset of randomized variants of MAM: in some configurations, randomized MAM is more efficient than (deterministic) MAM. (The curve *MAM 1-random, 1 processor* does not appear in the figure because it is above the y-axis value range due to its bad performance.) Indeed, a trade-off exists between time spent per iteration and precision gained after an iteration. For example, with 10 processors, each processor treats six measures in the deterministic MAM, but only one is treated in the randomized MAM. Therefore, the time spent per iteration is roughly six times shorter in the latter, which counterbalances the loss

of accuracy per iteration. On the other hand, when using 20 processors, only three measures are treated by each processor, and the trade-off is not worth it anymore: the gain in time does not compensate for the loss in accuracy per iteration. One should adapt the use of the algorithm with care since this trade-off conclusion is only heuristic and strongly depends on the underlying dataset and hardware. A sensitivity analysis is always a good thought for choosing the most effective amount of measures handled per processor while using the randomized MAM against the deterministic MAM

6.2.4 Influence of the support

This section echoes Section 6.1 and studies the influence of the support size. To do so, two datasets have been tested for MAM and IBP. The first dataset is already used in Section 6.2.3: 60 pictures of 3's taken from the MNIST database [35]. The second dataset is also composed of these 60 images but each digit has been randomly translated and rotated in the same way as in section 6.2.2. Therefore, the union of the support of the second dataset is greater than the first one.

section 6.2.4 presents two graphs that have been obtained just as in Section 6.2.3, but displaying the evolution in percentage: $\Delta W_{\%} := \frac{\bar{W}_2^2(p^k) - \bar{W}_2^2(p_G)}{\bar{W}_2^2(p_G)} \times 100$. Once more, the hyperparameters have been fully tuned. The hyperparameter of the IBP method is smaller for the second dataset. Indeed, as stated in [41], the greater the support, the stronger the restrictions on λ, and thus, the less precise IBP.

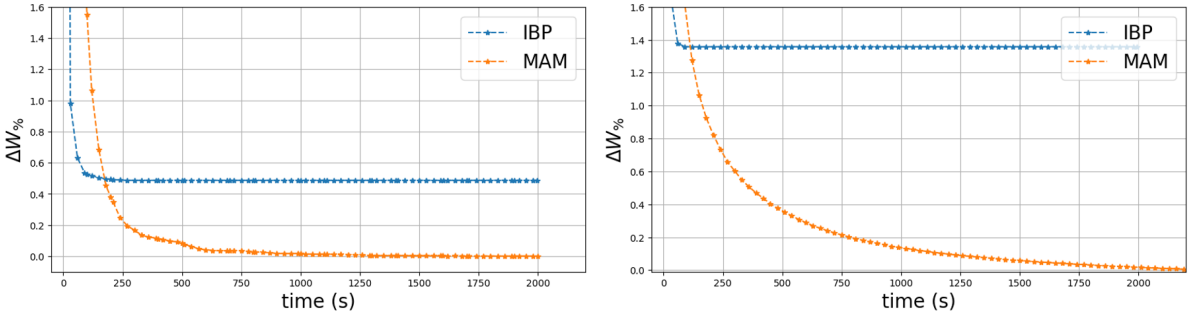


Figure 4: Evolution of the percentage of the distance between the exact solution of the barycenter problem and the computed solution using IBP and MAM method with 20 processors: (left) for the standard MNIST, (right) for the randomly translated and rotated MNIST.

6.2.5 Comparison with B-ADMM

This subsection compares MAM with the algorithm B-ADMM of [37] using the dataset and `MATLAB` implementation provided by the authors at the link `https://github.com/bobye/d2_kmeans`. We omit IBP in our analysis because it has already been shown in [37, Table I] that IBP is outperformed by B-ADMM in this dataset. As in [37, Section IV], we consider $M = 1000$ discrete measures, each with a sparse finite support set obtained by clustering pixel colors of images. The average number of support points is around 6, and the barycenter's number of fixed-support points is $R = 60$. The optimal value of (8) is 712.7, computed in 10.6 seconds by the Gurobi LP solver. We have coded MAM in `MATLAB` to have a fair comparison with the `MATLAB` B-ADMM algorithm provided at the above link. Since MAM and B-ADMM use different stopping tests, we have set their stopping tolerances equal to zero and

Table 2: MAM vs B-ADMM. B-ADMM code is the one provided by its designers without changing parameters (except the stopping set to zero and the maximum number of iterations). Both algorithms use the same initial point. The optimal value of the WB barycenter for this dataset is 712.7, computed by Gurobi in 10.6 seconds.

Iterations	Objective value		Seconds	
	B-ADMM	MAM	B-ADMM	MAM
100	742.8	716.7	1.1	1.1
200	725.9	714.1	2.4	2.2
500	716.5	713.3	5.6	5.4
1000	714.1	712.9	11.8	10.8
1500	713.5	712.8	18.9	16.2
2000	713.3	712.8	25.1	21.6
2500	713.2	712.8	31.0	27.1
3000	713.1	712.7	39.8	32.4

let the solvers stop with a maximum number of iterations. Table 2 below reports CPU time in seconds and the objective values yielded by the (approximated) barycenter $\tilde{p}$ computed by both solvers: $\bar{W}_2^2(\tilde{p})$.

The results show that, for the considered dataset, MAM and B-ADMM are comparable regarding CPU time, with MAM providing more precise results. B-ADMM currently lacks a convergence analysis, unlike MAM.

6.3 Free-support approach

This section considers the *free-support* problem (see Section 2, theorem 1), where the measures are supported on the same discrete grid in $\mathbb{R}^2$ ($d = 2$) and $\alpha_m = \frac{1}{M}$ for all $m = 1, \ldots, M$. The dataset we use is the one from [2], illustrated in section 6.3. In this case, $M = 10$ measures, $S = K^2 = 60^2$ and $R = ((K-1)M+1)^d = 591^2 = 349281$. The resulting LP problem is too large to be solved by standard solvers. Therefore, we employed the dedicated solver of [2], available at the link https://github.com/eboix/high_precision_barycenters.

Figure 5: Five 60×60 images from the nested ellipses dataset in [2].

Section 6.3 presents the evolution of the points computed by MAM along 9 hours of processing. The image on the right-hand side is an exact barycenter computed by the solver of [2] after 3.5 hours. We recall that [2] handles the dual of (8) by employing a geometry-based separation oracle. Once the dual is solved, the method recovers a primal vertex, yielding thus a sparse WB. As a result, the right-hand side image in Section 6.3 is sharp. Such an exact WB is sharper than the point provided by MAM after 9 hours. Despite the visual differences, the point provided by MAM is a Wasserstein barycenter. To see this, we compare the values of the objective function in (8), i.e., the Wassersein barycentric distance. The exact solution of the method in [2] has a barycentric distance of 0.2666. After 1 hour of processing, our method had a barycenter distance of 0.2702, which improved to 0.2667 after 3.5 hours, when the solver [2] halts. The slight visual difference stems from the fact that [2] finds a vertex solution

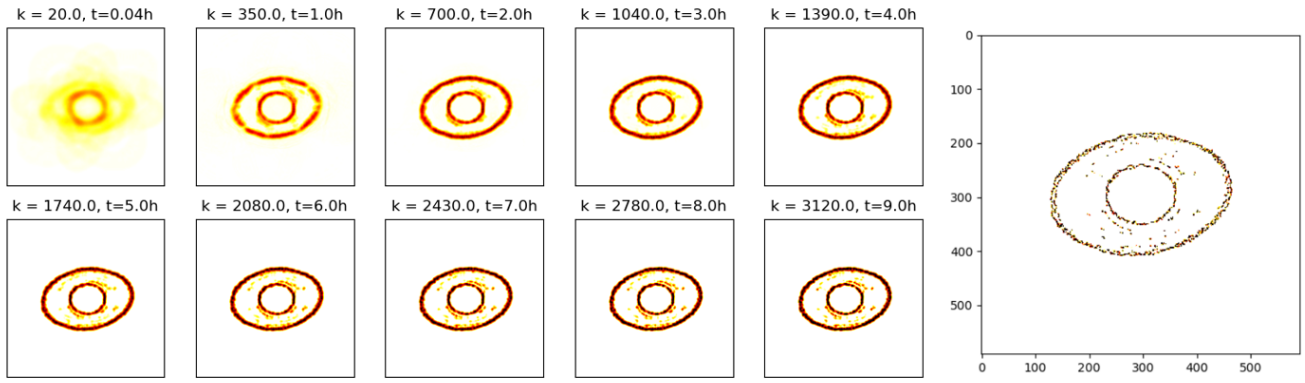


Figure 6: Evolution of the approximated MAM barycenter with time in regards with the exact barycenter of the Altschuler and Bois-Adsera algorithm computed in 4 hours [2].

to WB problem while MAM does not.

Section 6.3 illustrates MAM's iterative process. Let $\hat{\pi}^{(m),k}$ be the m^{th} transportation plan computed by MAM at iteration k (see Line 10 of Algorithm 1). Note that $\hat{W}_2^2(p^k) := \sum_{m=1}^{M} \langle c^{(m)}, \hat{\pi}^{(m),k} \rangle$ is an approximation to $\bar{W}_2^2(p^k) := \sum_{m=1}^{M} W_2^2(\mu^k, \nu^{(m)})$, the exact function value (barycentric distance) at iteration k. Furthermore, as it can be seen from the Douglas-Rachford algorithm in eq. (15), the transport plans $\hat{\pi}^{(m),k}$ do not necessarily respect the constraint embodied by $\mathcal{B}$ and is thus infeasible to the WB problem (8). Thus, the approximate value $\hat{W}_2^2(p^k)$ has to be seen in perspective with the distance of $\hat{\pi}^k$ to $\mathcal{B}$, i.e., $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k)$. Section 6.3 shows the evolution of the approximate barycentric distance $\hat{W}_2^2(p^k)$, infeasibility measure $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k)$, exact barycentric distance $\bar{W}_2^2(p^k)$, and optimal value $\bar{W}_2^2(p_{exact}) = 0.2666$. We emphasize that $\bar{W}_2^2(p^k)$ is computed (by Gurobi) after terminating MAM, while $\hat{W}_2^2(p^k)$ and $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k)$ are computed along the iterative process. After 3.5 hours, MAM provides $\bar{W}_2^2(p^k) = 0.2667$, $\hat{W}_2^2(p^k) = 0.2658$ and $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k) = 1.27 \cdot 10^{-5}$.

Because of its structure, the algorithm of [2] cannot provide intermediary approximations of the barycenters, which is a disadvantage of the method over MAM. As an example, we consider $M = 10$ images 40×40 presented in Section 6.2.3. Although smaller, these images are much denser than the ones in Section 6.3 and thus the WB problem is more complicated. While MAM can provide *free-support* WB approximations all along its iterative process, the solver of [2] could not provide a solution after 50 hours of processing.

6.4 Unbalanced Wasserstein Barycenter

This section treats a particular example to illustrate the interest in using UWB. The artificial dataset is composed of 50 images with resolution 80×80. Each image is divided into four squares. The top left, bottom left, and bottom right squares are randomly filled with double nested ellipses and the top right square is always empty as exemplified in section 6.4. In this example, every image is normalized to depict a probability measure so that we can compare (fixed-support) WB and UWB.

With respect to eq. (11), one set of constraints is relaxed and the influence of the hyperparameter γ is studied. If γ is large enough (i.e. greater than $\|\texttt{vec}(c)\| \approx 1000$, see proposition 1), the problem boils down to the standard WB problem since the example deals with probability measures: the resulting UWB is indeed a WB. When decreasing γ the transportation costs take more importance than the distance to $\mathcal{B}$ which is more and more relaxed. Therefore, as illustrated by section 6.4, the resulting UWB splits the image into four parts, giving visual meaning to the

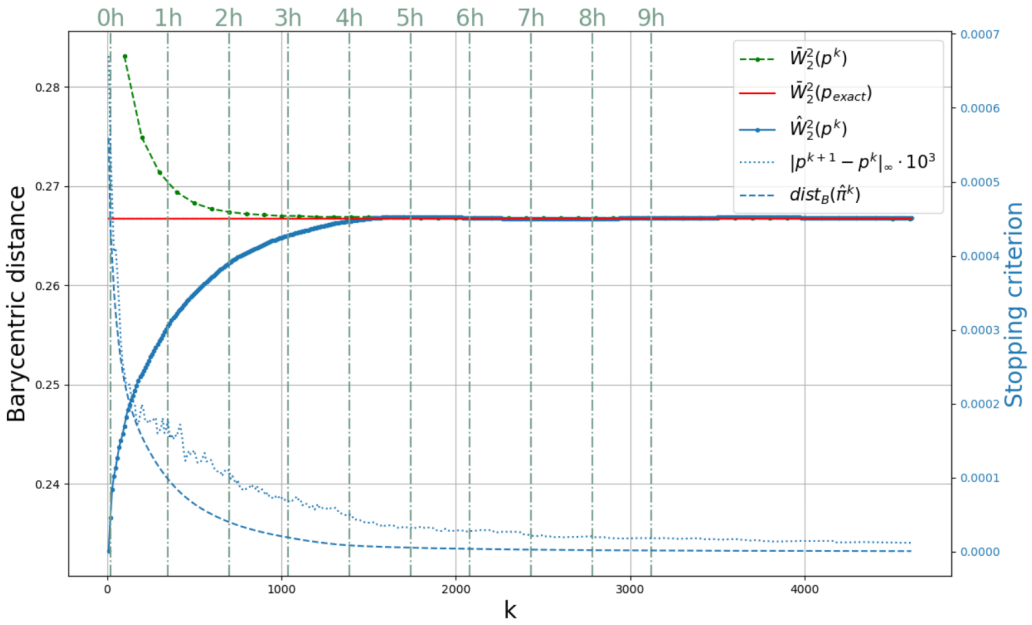


Figure 7: Evolution of the approximated Wasserstein barycenter distance $\hat{W}_2^2(p^k)$ with iterations (k) and time.

fixed-support barycenter.

In the same vein, section 6.4 provides an illustrative application of MAM for computing UWB in another dataset.

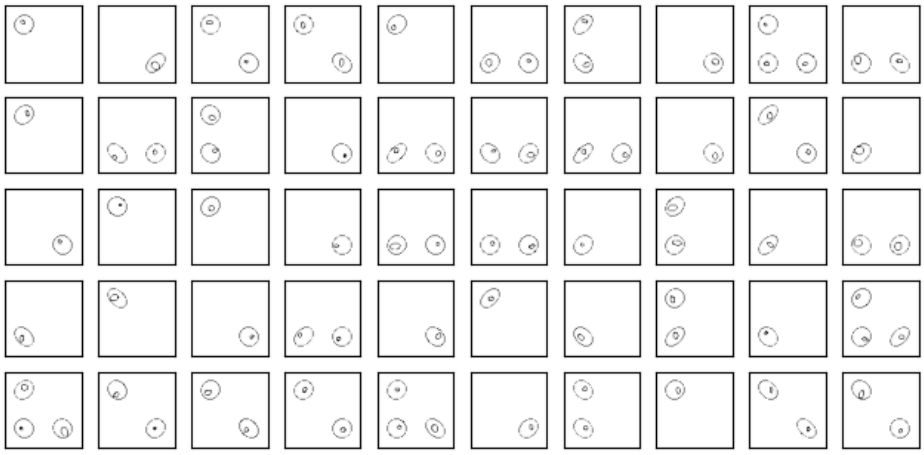


Figure 8: Dataset composed of 50 pictures with nested ellipses randomly positioned in the top left, bottom right, and left corners.

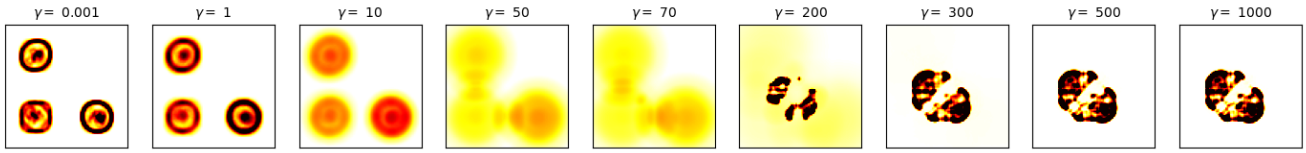


Figure 9: UWB computed with MAM for different values of γ.

References

References

[1] M. Agueh and G. Carlier, *Barycenters in the wasserstein space*, Siam Journal on Mathematical Analysis, 43 (2011), pp. 904–924, https://doi.org/10.1137/100805741.

[2] J. M. Altschuler and E. Boix-Adsera, *Wasserstein barycenters can be computed in polynomial time in fixed dimension*, Journal of Machine Learning Research, 22 (2021), pp. 1–19.

[3] J. M. Altschuler and E. Boix-Adserà, *Wasserstein barycenters are NP-Hard to compute*, SIAM Journal on Mathematics of Data Science, 4 (2022), pp. 179–203, https://doi.org/10.1137/21M1390062.

[4] E. Anderes, S. Borgwardt, and J. Miller, *Discrete wasserstein barycenters: optimal transport for discrete data*, Mathematical Methods of Operations Research, 84 (2016), pp. 389–409, https://doi.org/10.1007/s00186-016-0549-x.

[5] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer International Publishing, 2nd ed., 2017, https://doi.org/10.1007/978-3-319-48311-5.

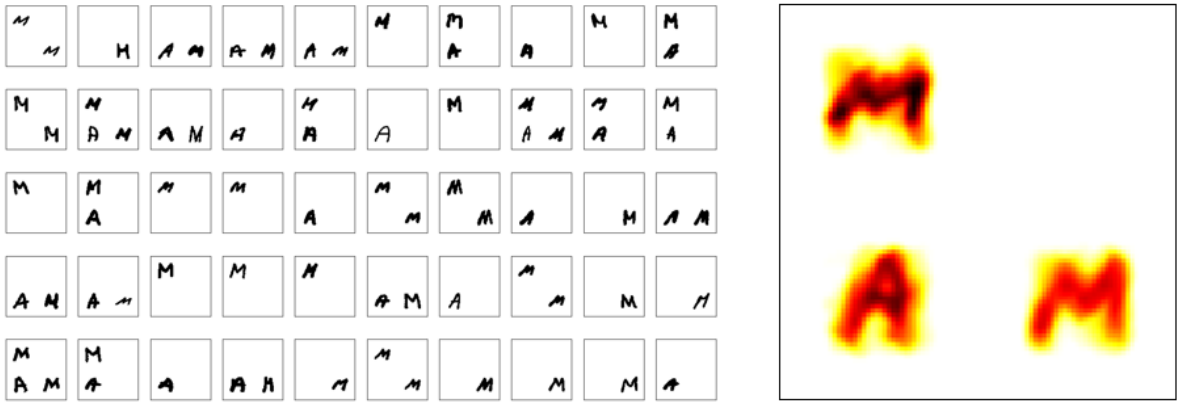


Figure 10: *(left)* UWB for a dataset of letters M-A-M built in the same logic than section 6.4 with 50 figures: *(right)* resulting UWB with $\gamma = 0.01$, computed in 200 seconds using 10 processors.

[6] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, *Iterative bregman projections for regularized transportation problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. 1111–1138, https://doi.org/10.1137/141000439.

[7] D. P. Bertsekas, *Convex Optimization Algorithms*, no. 1st, Athena Scientific, 2015, https://doi.org/ISBN1-886529-28-0.

[8] S. Borgwardt, *An LP-based, strongly-polynomial 2-approximation algorithm for sparse Wasserstein barycenters*, Operational Research, 22 (2022), pp. 1511–1551, https://doi.org/10.1007/s12351-020-00589-z.

[9] S. Borgwardt and S. Patterson, *Improved linear programs for discrete barycenters*, INFORMS Journal on Optimization, 2 (2020), pp. 14–33, https://doi.org/10.1287/ijoo.2019.0020.

[10] S. Borgwardt and S. Patterson, *On the computational complexity of finding a sparse Wasserstein barycenter*, Journal of Combinatorial Optimization, 41 (2021), pp. 736–761.

[11] G. Carlier, A. Oberman, and E. Oudet, *Numerical methods for matching for teams and Wasserstein barycenters*, ESAIM: Mathematical Modelling and Numerical Analysis, 49 (2015), pp. 1621–1642, https://doi.org/10.1051/m2an/2015033.

[12] P. L. Combettes and J. Eckstein, *Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions*, Mathematical Programming, 168 (2018), pp. 645–672, https://doi.org/10.1007/s10107-016-1044-0.

[13] P. L. Combettes and J.-C. Pesquet, *Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping*, SIAM Journal on Optimization, 25 (2015), pp. 1221–1248, https://doi.org/10.1137/140971233.

[14] L. Condat, *Fast projection onto the simplex and the* $\boldsymbol{l_1}$ *ball*, Mathematical Programming, 158 (2016), pp. 575–585.

[15] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, in Advances in Neural Information Processing Systems, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds., vol. 26, Curran Associates, Inc., 2013.

[16] M. Cuturi and A. Doucet, *Fast computation of Wasserstein barycenters*, in Proceedings of the 31st International Conference on Machine Learning, E. P. Xing and T. Jebara, eds., vol. 32 of Proceedings of Machine Learning Research, Bejing, China, 22–24 Jun 2014, PMLR, pp. 685–693.

[17] M. Cuturi and G. Peyré, *A smoothed dual approach for variational Wasserstein problems*, SIAM Journal on Imaging Sciences, 9 (2016), pp. 320–343, https://doi.org/10.1137/15M1032600.

[18] W. de Oliveira, C. Sagastizábal, D. D. J. Penna, M. E. P. Maceira, and J. M. Damázio, *Optimal scenario tree reduction for stochastic streamflows in power generation planning problems*, Optimization Methods and Software, 25 (2010), pp. 917–936.

[19] J. Douglas and H. H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Transactions of the American Mathematical Society, 82 (1956), pp. 421–439.

[20] J. Eckstein and D. P. Bertsekas, *On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming, 55 (1992), pp. 293–318, https://doi.org/10.1007/bf01581204.

[21] A. Fu, J. Zhang, and S. Boyd, *Anderson accelerated Douglas–Rachford splitting*, SIAM Journal on Scientific Computing, 42 (2020), pp. A3560–A3583, https://doi.org/10.1137/19m1290097.

[22] A. Gramfort, G. Peyré, and M. Cuturi, *Fast optimal transport averaging of neuroimaging data*, in Information Processing in Medical Imaging, S. Ourselin, D. C. Alexander, C.-F. Westin, and M. J. Cardoso, eds., Cham, 2015, Springer International Publishing, pp. 261–272.

[23] T. Guillaume, P. Gabriel, and G. Yann, *Wasserstein loss for image synthesis and restoration*, SIAM Journal on Imaging Sciences, 9 (2016), pp. 1726–1755.

[24] F. Heinemann, M. Klatt, and A. Munk, *Kantorovich–Rubinstein distance and barycenter for finitely supported measures: Foundations and algorithms*, Applied Mathematics & Optimization, 87 (2022), p. 4, https://doi.org/10.1007/s00245-022-09911-x.

[25] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, *Asynchronous distributed optimization using a randomized alternating direction method of multipliers*, in 52nd IEEE Conference on Decision and Control, IEEE, dec 2013, https://doi.org/10.1109/cdc.2013.6760448.

[26] J. v. Lindheim, *Simple approximative algorithms for free-support wasserstein barycenters*, Computational Optimization and Applications, 85 (2023), pp. 213–246, https://doi.org/10.1007/s10589-023-00458-3.

[27] G. Peyré, *Bregmanot*, 2014, https://github.com/gpeyre/2014-SISC-BregmanOT.

[28] G. Peyré and M. Cuturi, *Computational optimal transport: With applications to data science*, Foundations and Trends in Machine Learning, 11 (2019), pp. 355–607, https://doi.org/10.1561/2200000073, http://dx.doi.org/10.1561/2200000073.

[29] G. C. Pflug and A. Pichler, *Multistage Stochastic Optimization*, Springer International Publishing, 2014, https://doi.org/10.1007/978-3-319-08843-3.

[30] G. Puccetti, L. Rüschendorf, and S. Vanduffel, *On the computation of wasserstein barycenters*, Journal of Multivariate Analysis, 176 (2020), https://doi.org/10.1016/j.jmva.2019.104581.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas, *The earth mover's distance as a metric for image retrieval*, International Journal of Computer Vision, 40 (2000), pp. 99–121, https://doi.org/10.1023/A:1026543900054.

[32] T. Sejourne, G. Peyre, and F.-X. Vialard, *Unbalanced optimal transport, from theory to numerics*, Handbook of Numerical Analysis, 24 (2023), pp. 407–471, https://doi.org/10.1016/bs.hna.2022.11.003.

[33] D. Simon and A. Aberdam, *Barycenters of natural images constrained Wasserstein barycenters for image morphing*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7910–7919.

[34] R. Sinkhorn, *Diagonal equivalence to matrices with prescribed row and column sums. ii*, Proceedings of the American Mathematical Society, 45 (1974), pp. 195–198.

[35] Tijmen, *affnist*, 2013, https://www.cs.toronto.edu/~tijmen/affNIST/.

[36] C. Villani, *Optimal transport: onld and new*, vol. 338, Springer Verlag, 2009.

[37] H. Wang and A. Banerjee, *Bregman alternating direction method of multipliers*, in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds., vol. 27, Curran Associates, Inc., 2014.

[38] J.-P. Watson and D. L. Woodruff, *Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems*, Computational Management Science, 8 (2010), pp. 355–370, https://doi.org/10.1007/s10287-010-0125-4.

[39] Z. Xu, M. Figueiredo, and T. Goldstein, *Adaptive ADMM with Spectral Penalty Parameter Selection*, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, A. Singh and J. Zhu, eds., vol. 54 of Proceedings of Machine Learning Research, PMLR, 20–22 Apr 2017, pp. 718–727.

[40] J. Ye and J. Li, *Scaling up discrete distribution clustering using ADMM*, in 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 5267–5271, https://doi.org/10.1109/ICIP.2014.7026066.

[41] J. Ye, P. Wu, J. Z. Wang, and J. Li, *Fast discrete distribution clustering using Wasserstein barycenter with sparse support*, IEEE Transactions on Signal Processing, 65 (2017), pp. 2317–2332, https://doi.org/10.1109/TSP.2017.2659647.