Scenario Tree Reduction and Operator Splitting Method for Stochastic Optimization of Energy Systems

A dissertation submitted by

Daniel Mimouni

in partial satisfaction of the requirements for the
PhD. Mid-term Evaluation

École Doctorale 580
Sciences et Technologies de l'Information et de la Communication (STIC)

on
March 7, 2024

based on a work conducted under the supervision of

Paul Malisani	Research scientist, IFPEN	Promoter
Welington de Oliveira	Research scientist, HDR, CMA Mines Paris	Supervisor
Jiamin Zhu	Research scientist, IFPEN	Co-Promoter

Preface

This document compiles the work accomplished during the first year and a half of my thesis. It is structured into two primary sections: one detailing our research on a novel method for computing *Wasserstein Barycenters*, and the other presenting our proposed approach to addressing *scenario tree reduction*. This endeavor has been the result of close collaboration with Paul, Jiamin, and Welington.

Introduction for laypersons

This subsection will present with an example, how scenario tree and decision management under uncertainty are linked.
Intermittent power plants are subject to grid integration strategies. Let us take the example of a solar panel farm supplying electricity to a city. On the one hand, the aim of this system is always to provide enough electricity for the city's needs. On the other hand, electricity production relies on factors beyond our control (sunlight and weather in general). Therefore, this system must be connected to stable electricity sources to mitigate potential power shortages in the event of insufficient sunlight. For example, France would be more likely to rely on nuclear plants, which are more prevalent. Then, the market (out of European laws) works this way: the day prior, the farm manager is tasked with determining the necessary external electricity based on the estimated farm production and projected city consumption, represented as probability densities. Initially, the manager decides to ensure that the electricity supplied aligns with the farm's production. Throughout the day and at each time step (which could be minutes for solar panels), the manager monitors the energy produced by the farm, the energy consumed during each time step, and whether the introduction of external electricity maintains a balanced consumption-to-production equation. Then three cases can happen:

- The consumption exceeds the combined production from the farm and the external electricity supply. The manager must purchase additional external electricity at higher price.
- The combination of production from the farm and external electricity sufficiently meets the city's needs, eliminating the necessity for the manager to purchase additional electricity. Any surplus electricity can be stored in energy storage facilities such as dams or batteries.
- The combination of farm production and external electricity adequately fulfills the city's requirements. However, the manager always has the option to purchase additional external electricity and store it if the combined cost of purchase and storage (based on forecasts modeled as probability distributions) is more favorable at the current time step compared to purchasing it at any future time step.

This description of the problem emphasizes that each decision has repercussions on subsequent decisions, thereby creating a chain of interconnected choices. In any case, the manager's objective is to minimize expenses to the greatest extent possible.

Abstract

Managing uncertainties in multistage stochastic optimization poses a substantial challenge, necessitating a a complex trade-off between, on the one hand, the representation of the uncertainties (i.e. the number of scenarios) and, on the other hand, the computational tractability. Scenario reduction methods, pioneered in 2003 by Dupavcova et al. [26], offer a promising outlooks for achieving a satisfactory trade-off. However, the choice of distance metric for reducing scenario trees significantly influences solution quality. While clustering techniques have been prevalent, recent research has turned to Wasserstein-based methods to minimize transport distance between probabilty measures.

This work presents a comprehensive investigation of the use of Wasserstein distance for scenario tree reduction in the context of multistage stochastic optimization. The Wasserstein barycenter (WB) serves as a tool for summarizing sets of probabilities, it appears in a number of disciplines, including applied probability, clustering and image processing. Numerically efficient methods to computing the WB rely on entropic regularization functions, resulting in approximate solutions due to limitations in solver capabilities. In contrast, this research introduces an exact approach based on the Douglas-Rachford splitting method directly applied to the WB linear optimization problem.The proposed solving algorithm achieves a trade-off between the numerical efficiency of regularization-based methods and the precision of exact LP solvers.

In [41], the authors develop a reduction algorithm based on nested Wasserstein distance. This algorithm consists of computing a significant amount of Wasserstein barycenters. The second contribution of this work is to implement dedicated WB computation algorithms, including the Iterative Bregmann Projection method (IBP), Sinkhorn distance, and the newly introduced Method of Averaged Marginals (MAM) in the algorithm proposed in [41] to accelerate its performances.

By proposing efficient algorithms for computing Wasserstein barycenters and reducing scenario trees, we address critical challenges in managing uncertainties in multistage stochastic optimization. Looking ahead, future research directions include further exploration of the interplay between optimization algorithms and stochastic processes to refine scenario tree reduction methodologies and enhance the applicability of Wasserstein-based methods in complex optimization problems.

Contents

Chapter 1

Introduction

1.1 Context

The global energy landscape is transforming, driven by the imperative to reduce greenhouse gas emissions and mitigate climate change. Central to this transition is the increasing integration of renewable energy sources, such as wind and solar, into the electricity generation mix. While renewable energy offers significant environmental benefits, its inherent intermittency and variability present formidable challenges for grid operators and energy management systems.

In traditional electricity grids dominated by fossil fuels, power generation could be adjusted according to demand fluctuations, providing a stable and predictable supply. However, the rise of renewables introduces a new dynamic, where generation is subject to the vagaries of weather patterns and time-of-day variations. This unpredictability disrupts the conventional energy supply and demand balance paradigm, necessitating innovative solutions for effective grid management.

One key strategy to address this challenge lies in enhancing flexibility on the demand side of the electricity grid. By enabling consumers to adjust their energy consumption patterns in response to real-time conditions, demand-side flexibility offers a powerful tool for optimizing the utilization of renewable energy resources. However, realizing this potential requires sophisticated energy management systems capable of orchestrating complex interactions between diverse stakeholders, energy assets, and market dynamics.

In this context, optimization-based energy management systems emerge as a promising approach to harness the full potential of renewable energy integration. By leveraging advanced mathematical techniques, such as stochastic optimization, these systems can intelligently allocate resources, optimize scheduling, and mitigate risks in the presence of uncertainty. Stochastic optimization techniques are well-suited to model and optimize complex, uncertain systems, making them ideally suited for addressing the inherent variability of renewable energy generation.

1.2 Stochastic optimal control framework for energy management

Mathematically speaking, most energy management systems aim at solving the following multistage stochastic optimization problem

$$\min_{u_1} c_1(x_1, u_1, \xi_1) + \min_{u_2} \mathbb{E}_{\xi_2} \left[c_2(x_2, u_2, \xi_2) + \cdots + \min_{u_T} \mathbb{E}_{\xi_T} \left[c_T(x_T, u_T, \xi_T) \right] \right] \tag{1.1}$$

under the following constraints

$$x_{t+1} = f_t(x_t, u_t, \xi_t) \qquad t = 1, \ldots, T-1 \tag{1.2a}$$

$$(u_t, x_t) \in K_t \subset \mathbb{R}^m \times \mathbb{R}^n \qquad t = 1, \ldots, T \tag{1.2b}$$

$$x_1 = x^0, \tag{1.2c}$$

where $u_t \in \mathbb{R}^m$ denotes the decision vector at stage t, $x_t \in \mathbb{R}^n$ denotes the state variable, $\xi_t \in \mathbb{R}^p$ denotes the random vector at stage t, $c_t : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ denotes the cost function at stage t and f_t is the system dynamics. From the recursive structure of the objective function, one can deduce that an optimal decision at stage t, denoted $\bar{u}_t$, depends on the realizations of the past random variables denoted $(\xi_1, \ldots, \xi_t)$ and on the statistical properties of the future random variables $(\xi_{t+1}, \ldots, \xi_T)$. Thus, an optimal solution of eq. (1.1) is said to be *non-anticipative*: at stage t, we have decided $(u_1, \ldots, u_t)$, but we cannot anticipate decisions for stages $t+1, \ldots, T$ because the future is uncertain. When the uncertainties are represented with finitely many events, this nonanticipative format clearly leads to a scenario tree structure.

Most current energy management methods handle this multistage optimization problem using Model Predictive Control based methods (MPC) or stochastic optimization-based methods.

1.2.1 MPC for multistage stochastic optimal control problems

At each time step t, MPC algorithms [29, 42, 47] consist in predicting a realization of the next K random variables denoted $(\hat{\xi}_{t+1}, \ldots \hat{\xi}_{t+K})$, and solve the following deterministic optimization problem

$$\min_{u_t, \ldots, u_{t+K}} c_t(x_t, u_t, \xi_t) + \sum_{s=t+1}^{t+K} c_s(x_s, u_s, \hat{\xi}_s) \tag{1.3}$$

under constraints described in eq. (1.2), and in using the computed optimal variable $\bar{u}_t$ as a decision variable at stage t. These methods are easy to implement and rely on deterministic optimization algorithms. However, the optimization problem from eq. (1.3) almost does not use the statistical properties of the future random variables, potentially yielding far from sub-optimal decisions with respect to the original optimization problem from eqs. (1.1) and (1.2).

1.2.2 Dynamic programming-based optimization methods

Stochastic optimization methods are often based on sequential decomposition methods based on the principle of dynamic programming [5, 7, 8, 10]. Recently, significant work has been done on Stochastic Dual Dynamic Programming methods [12, 15, 48, 60, 61]. These methods have the advantage of naturally computing non-anticipative decisions and of being able to take into account a large number of uncertainties scenarios. On the other hand, these methods require strong assumptions on stochastic process, such as stage-wise independence. In the context

of energy management problems, this assumption is not always satisfied. In addition, these methods are appealing since they can overcome the dynamic programming's curse of dimensionality when the system's dynamics eq. (1.2a) is linear. Again, this assumption is often not met in practice.

1.2.3 Scenario decomposition for stochastic optimal control problems

Scenario decomposition techniques use a different paradigm of stochastic optimization problems: instead of stage decomposition, minimizing a recursive sequence of recourse functions, scenario decomposition techniques decompose the problem per scenario while keeping the whole time horizon in individual (scenario-based) subproblems. This different paradigm requires a different manner to deal with nonanticipativity, which strongly depends on the scenario tree filtration $\mathcal{F}$. Therefore, scenario decomposition techniques consist in optimizing over a set of functions defined on a probability space denoted $(\Omega, \mathcal{F}, \pi)$. The function spaces on this probability space are denoted as follows

$$\mathrm{U} := \{u : \Omega \mapsto \ell_2([1, \dots, T]; \mathbb{R}^m)\} \tag{1.4}$$
$$\mathrm{X} := \{x : \Omega \mapsto \ell_\infty([1, \dots, T]; \mathbb{R}^n)\} \tag{1.5}$$
$$\Xi := \{\xi : \Omega \mapsto \ell_2([1, \dots, T]; \mathbb{R}^p)\} \tag{1.6}$$

And we endow the space U with the following scalar product

$$\langle f, g \rangle_{\mathrm{U}} := \mathbb{E}\left[\langle f(.), g(.) \rangle_{\ell_2}\right] \tag{1.7}$$

and we denote $\|.\|_U$ its associated norm. The multistage optimization problem then writes

$$\min_{(x,u) \in \mathrm{X} \times \mathrm{U}} \left\{ \mathbb{E}\left[F(x, u, \xi)\right] := \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x_t, u_t, \xi_t)\right] \right\} \tag{1.8}$$

under the following constraints

$$x_{t+1} = f_t(x_t, u_t, \xi_t) \tag{1.9a}$$
$$(x_t, u_t) \in K_t \tag{1.9b}$$
$$x_1 = x^0 \tag{1.9c}$$
$$u - \mathbb{E}\left[u|\xi\right] = 0. \tag{1.9d}$$

The notation $\mathbb{E}[u|\xi]$ is the projection, with respect to $\|.\|_U$, onto the subspace of function adapted to ξ, i.e., it is the projection on the non-anticipative space. Therefore, constraint eq. (1.9d) ensures that the control u is non-anticipative. Now, let us denote $M : u \mapsto u - \mathbb{E}[u|\xi]$, the projection on the space orthogonal to non-anticipative strategies. This operator is auto-adjoint for the scalar product $\langle ., . \rangle_U$. Using this property one can define the augmented Lagrangian of problem eqs. (1.8) and (1.9a) to (1.9d) as follows

$$\sup_{\lambda} \langle \lambda, M(u) \rangle_{\mathrm{U}} = \sup_{\lambda} \langle M(\lambda), u \rangle_{\mathrm{U}}$$

$$L_r(x, u, \lambda) := \mathbb{E}\left[F(x, u, \xi) + \sum_{t=1}^{T-1} I_0\left(f_t(x_t, u_t, \xi_t) - x_{t+1}\right) + \sum_{t=1}^{T} I_{K_t}(x_t, u_t)\right] + \langle M(\lambda), u \rangle_{\mathrm{U}} + \frac{r}{2} \|M(u)\|_{\mathrm{U}}^2 \tag{1.10}$$

where I_A is the indicatrix of the set A. Assuming that the probability space $(\Omega, \mathcal{F}, \pi)$ is discrete and denoting $w := M(\lambda)$, one can use the progressive hedging algorithm (PHA) [56, 57], which is a operator splitting method, as described in algorithm 1.

Algorithm 1 Progressive Hedging Algorithm - PHA

▷ Step 0: Input

$k := 0$; $r > 0$; $w^0 := 0$; success $:= \perp$; tol > 0

while ¬success **do**

▷ Compute optimal solution by scenario

for $\omega_i \in \Omega$ **do**

$$(\hat{x}^{k+1}(\omega_i), \hat{u}^{k+1}(\omega_i)) \in \underset{(x,u)}{\arg\min}\, F(x, u, \xi(\omega_i)) + \langle w^k(\omega_i), u\rangle_{\ell_2} + \sum_t I_0\left(f_t(x_t, u_t, \xi_t(\omega_i)) - x_{t+1}\right) + \sum_t I_{K_t}(x_t, u_t) + \frac{r}{2}\|u - u^k(\omega_i)\|_{\ell_2} \quad (1.11)$$

end for

▷ Update non-anticipative control and dual variables

$u^{k+1} := \mathbb{E}[\hat{u}^{k+1}|\xi]$

$w^{k+1} := w^k + r\left(\hat{u}^{k+1} - u^{k+1}\right)$

success $:= \max\left\{\|\hat{u}^{k+1}(\omega) - u^{k+1}(\omega)\|_{\ell_2} : \omega \in \Omega\right\} \leq$ tol

$k := k + 1$

end while

return $(\hat{x}^k, \hat{u}^k)$

1.2.4 From scenario decomposition to scenario trees

The PHA requires computing the conditional expectation $\mathbb{E}[u|\xi]$ at each iteration. Now, at each time step t, the random variables $\xi_{[t]} := (\xi_1, \ldots, \xi_t)$ generate a partition of Ω denoted $\tilde{\mathcal{A}}_t$ defined as follows

$$\tilde{\mathcal{A}}_t := \{A \subseteq \Omega : \forall \omega_1, \omega_2 \in A, \forall s \leq t, \xi_s(\omega_1) = \xi_s(\omega_2)\} \quad (1.12)$$

The elements of the partition $\tilde{\mathcal{A}}_t$ are called t-atoms and we denote $\mathcal{F}_t$ the smallest σ-algebra generated by the t-atoms, and the sequence $(\mathcal{F}_t)_t$ is the filtration generated by the random variables $\xi_{[t]}$. Finally the sequence of t-partition can be represented as a scenario tree, where each node of depth t corresponds to a t-atom as illustrated on Figure 1.1. The non-anticipative constraint (1.9d) ensures that the optimal solution u yields the same t-partition of Ω as ξ and thus yields the same filtration $(\mathcal{F}_t)_t$. Now, using this t-partition computing the conditional expectation $u := \mathbb{E}[\hat{u}|\xi]$ is straightforward and we have

$$\forall A \in \tilde{\mathcal{A}}_t, \forall \omega \in A, u_{[t]}(\omega) := \frac{1}{\pi(A)} \sum_{\omega' \in A} \pi(\{\omega'\})\, \hat{u}_{[t]}(\omega'). \quad (1.13)$$

1.3 Scenario reduction for multistage stochastic optimization problem

As illustrated in Section 1.2.4, the random variables and the available information at stages $t = 1, \ldots, T$ in a stochastic optimization problem with finitely many scenarios can be represented as a scenario tree. As a result the complexity of the multistage stochastic optimization problem increases with the number of scenarios. Hence,

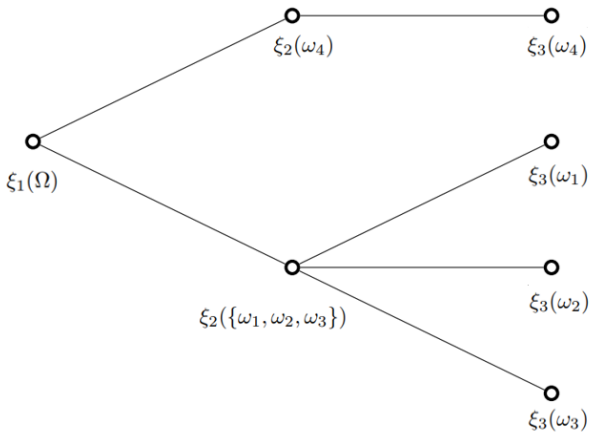

Figure 1.1: Let $\Omega = \{\omega_1, \ldots, \omega_4\}$. At stage $t = 1$, all scenarios have the same value, i.e., $\xi_1(\omega_i) = \xi_1(\omega_j)$, for all i, j. At stage $t = 2$, scenarios $\{\omega_1, \omega_2, \omega_3\}$ are still identical whereas $\xi_2(\omega_4) \neq \xi_2(\{\omega_1, \omega_2, \omega_3\})$. Finally, at stage $t = 3$, all scenarios are different, $\xi_3(\omega_i) \neq \xi_3(\omega_j)$, for all $i \neq j$.

the necessity of finding an equilibrium between representation of uncertainties and numerical tractability gained momentum and the concept of scenario reduction was introduced by [26] in 2003, pioneering forward reduction and backward selection methodologies.

When reducing scenario trees, the choice of the distance to minimize between trees is of primary importance. There exist distances, and thus methods, that ignore the accumulation of information over time on which rely the non-anticipativity of stochastic problems.

Notably, one of the most commonly used reduction algorithms is the approach proposed by Heitsch and Römisch [33], which measures distances between nodes with the same parent, ultimately reducing pairs of nodes until a stopping criterion is met. The work [69] developed algorithms that employ K-means clustering and LP moment-matching methods to approximate multistage scenario trees. In [37] comparative analysis of several clustering algorithms are conducted, it highlights trade-offs between computation time and nested distance but also that the general behaviour of these algorithms concerning the closeness to the exact solution of the stochastic problem is still volatile.

Wasserstein-based methods aim to minimize probabilistic distances, akin to the Wasserstein distance [58, 66], between the original complex tree and a simpler smaller tree whose solution computation will be more tractable; see for instance [24, 26]. Li and Floudas [44] treat scenario reduction as a mixed-integer linear programming (MILP) problem, and aim at reducing the Wasserstein distance between scenarios. Subsequent works [39, 43] improved the LP-based scenario reduction strategy, the latter solved the entropy-regularized optimal transport using the Sinkhorn–Knopp algorithm [64].

The Wasserstein distance approach is a natural choice when tackling probability distributions but does not capture the structural information of multistage scenario trees. Neglecting the tree filtration potentially leads to deviations in solutions, because the solution of a stochastic optimisation problem is non-anticipative. This is why it is of great importance to use a reduction method that takes into account the tree structure, i.e. the *filtration*. Indeed, [36] studied the stability of multistage stochastic program and showed that multistage scenario

tree reductions should be based on a Lr distances and a filtration distances. They introduced the *filtration distance* and derived a method based on this measure [34, 35]. However, this method encounters computational challenges, as they rely on solving NP-hard facility location problems. The work [16] proposed a scenario tree reduction algorithm that circumvents this difficulty by clustering tree nodes based on a new filtration distance and computes an approximation of the reduction problem.

The introduction of the nested distribution and nested distance [51, 52], also called *process distance*, offered a valuable framework for stochastic programs, allowing for the avoidance of the *filtration distance*. This new way of computing a distance between trees that takes into account the filtration, is now broadly used to compare trees and assess the accuracy of reduction methods. Leveraging the nested distance to guide the process approximation enables control over both the statistical quality of the approximation and its impact on the objective in multistage stochastic optimization problems pertaining to the respective stochastic process [36, 41]. Kovacevic and Pichler [41] introduced an algorithm that directly targets the minimization of the nested distance, although this approach is proved complex and computationally expensive. Indeed while it is easy to calculate the process distance between given trees by solving one LP, finding a tree to minimize the process distance is much more difficult. Both probabilities and values (i.e., the states or outcomes) of the approximating tree have to be chosen, such that the process distance is minimized. This leads to a large, non-convex optimization problem, which can be solved in reasonable time only for small instances due to the computation of multiple potentially large-scale LPs. The work [6] provides a variant of the method in [41] capable to efficiently handle very large scenario trees provided *the stochastic process is stage-wise independent*, a strong assumption we do not assume in this work. Introducing a more effective scenario tree reduction method based on the nested distance is of paramount significance.

Based on the Kovacevic and Pichler's work [41], this article introduces a novel approach for reducing general scenario trees. We noticed that the minimization of the Nested Distance between two trees, as employed in the Kovacevic and Pichler's method, makes naturally appear the Wasserstein Barycenter problem. We propose to leverage the optimal transport problem and employ efficient algorithms, such as the Iterative Bregmann Projection method (IBP) [9], the Sinkhorn distance [40, 64] or the newly introduced Method of Averaged Marginals [46], to attractively enhance the Kovacevic and Pichler's reduction method. This work offers a promising solution to the computational challenges of scenario tree reduction.

The remainder of this work is organized as follows. Chapter 2 tackles the central optimization problem of the tree reduction process: we provide background information on optimal transport, delve into the concept of the Wasserstein distance, and discuss its implications in computing barycenters. Subsequently, we introduce a novel algorithm designed to address the problem of computing Wasserstein Barycenter (WB): *the Method of Averaged Marginals*. Chapter 3 presents the Nested Distance and the optimal transport methods applied to stochastic processes to reduce trees. The barycentric approach for reduction is introduced, and the method's algorithm is presented. This work concludes with numerical applications comparing the initial Kovacevic and Pichler's method (KP) with the two other versions introduced in this paper. Finally, Chapter 4 concludes with a roadmap outlining the remaining tasks to be completed before the end of the PhD project.

Chapter 2

The Wasserstein Barycenter problem

2.1 Literature review

In applied probability, stochastic optimization, and data science, a crucial aspect is the ability to compare, summarize, and reduce the dimensionality of empirical 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 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 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 [23, 50, 53]. The Wasserstein distance has several valuable theoretical and practical properties [58, 66] that are transferred to (Wasserstein) barycenters [1, 19, 50, 55]. 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, where datasets often contain many pixels and complex features that must be accurately represented and analyzed [31, 62].

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) that has no analytical solution and cubic worst-case complexity[1] [72]. The situation becomes even worse for computing a WB because its definition involves several optimal transport plans. In the simpler case of fixed support, which is the focus of this work (see Section 2.2 below), computing a WB amounts to solving an LP whose dimensions generally exceed standard solvers' capabilities [1]. Several numerical methods have been proposed in the literature to address the challenge. They invariably fall into one of the following categories:

i) Inexact methods, based on reformulations via an entropic regularization [9, 19–21, 30, 50, 72];

ii) Exact decomposition methods, consisting in solving a sequence of smaller and simpler subproblems [20, 71].

While our proposal falls into the second category, the vast majority of methods found in the literature are inexact ones, employing or not decomposition techniques. Indeed, the work [20] proposes to inexactly compute a WB by solving an approximating model obtained by regularizing the WB problem with an entropy-like function. The technique allows one to employ the celebrate Sinkhorn algorithm [18, 63], which has a simple closed-form

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

expression and can be implemented efficiently using only matrix operations. When combined with a projected subgradient method, this regularizing approach fits into category i) above. However, if instead the underlying transport subproblems are solved exactly without the regularization technique, then Algorithm 1 from [20] falls into category ii).

The regularization technique of [18] opened the way to the *Iterative Bregman Projection* (IBP) method proposed in [9]. IBP is highly memory efficient for distributions with shared support set and is considered to be one of the most effective methods to tackle WB problems. However, as IBP works with an approximating model, the computed point is not a solution to the WB problem, and thus IBP is an inexact method.

Another approach fitting into the category of inexact methods has been recently proposed in [72], 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 [72] is an modification (tailored to the WB problem) of the *Bregman Alternating Direction Method of Multipliers* (B-ADMM) of [67]. 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 is not well understood and the approach should be considered as an heuristic.

An inexact method that disregards entropic regularizations is presented [55], and denoted by *Iterative Swapping Algorithm* (ISA). The approach is a non-parametric algorithm that provides a sharp image of the support of the barycenter and has a quadratic time complexity. Essentially, ISA is designed upon approximating the linear program by a multi-index assignment problem which is solved in an iterative manner. Another approach based on successive approximations of the WB (linear programming) problem is proposed in [13].

Concerning exact methods, the work [14] proposes a simpler linear programming reformulation of the WB problem that leads to an LP that scales linearly with the number of measures. Although the resulting LP is smaller than the WB problem, it still suffers from heavy computation time and memory consumption [55]. In contrast, [71] proposes to address the WB problem via the standard ADMM algorithm, which decomposes the problem in smaller and simpler subproblems. As mentioned by the authors in their subsequent paper [72], the numerical efficiency of the standard ADMM is still inadequate for large datasets.

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 WB. However, such a naive strategy is generally unsatisfactory and limits the use of WBs in many real-life applications such as logistic, medical imaging and others coming from the field of biology [32,59]. 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 [32]. Essentially, such generalizations, known as unbalanced Wasserstein barycenters (UWBs), depend on how one chooses to relax the marginal constraints. In the review paper [59] and references therein, marginal constraints are moved to the objective function with the help of divergence functions. Differently, in [32] 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 [59].

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. While our approach can be seen as an abridged alternative to the thorough methodologies of [32] and [59], its favorable structure for efficient splitting techniques combined with the good quality of the issued UWBs confirm the formulation's practical interest.

To cope with the challenge of computing (balanced and unbalanced) WBs, we propose a new algorithm based on the celebrated Douglas-Rachford splitting operator method (DR) [25,27,28]. Our proposal, which falls into the category of exact decomposition methods, is denoted by *Method of Averaged Marginals* (MAM). The motivation

for its name is due to the fact that, 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. 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 a parameter. 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 to an exact barycenter. Although MAM's convergence speed is not as exceptional as IBP's, it is observed in practice that after a few tens of iterations, the average of marginals computed by MAM is a better approximation of a WB than the solution provided by IBP, no matter how many iterations the latter performs[2]. 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 2.2 introduces the notation and recalls the balanced WB problems' formulation. The proposed formulation for unbalanced WBs is presented in Section 2.3. In Section 2.4 the WB problems are reformulated in a suitable way so that the Douglas-Rachford splitting (DR) method can be applied. The same section briefly recalls the DR algorithm 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 2.5. Convergence analysis is given in the same section by relying on the DR algorithm's properties. Section 2.6 illustrates the numerical performance of the deterministic and randomized variants of MAM on several data sets from the literature. Numerical comparisons with IBP and B-ADMM are presented for the balanced case. Then some applications of the UWB are considered.

2.2 Background on optimal transport and Wasserstein barycenter

2.2.1 Continuous Wasserstein Barycenter

Let $(\Omega, \mathtt{d})$ be a metric space and $P(\Omega)$ the set of Borel probability measures on Ω. Furthermore, let ξ and ζ be two random vectors having probability measures μ and ν in $P(\Omega)$, that is, $\xi \sim \mu$ and $\zeta \sim \nu$.

Definition 1 (Wasserstein Distance)**.** *For $\iota \in [1, \infty)$, and probability measures μ and ν in $P(\Omega)$. Their ι-Wasserstein distance W_ι is :*

$$W_\iota(\mu, \nu) := \left(\inf_{\pi \in U(\mu,\nu)} \int_{\Omega \times \Omega} \mathtt{d}(\xi, \zeta)^\iota d\pi(\xi, \zeta) \right)^{1/\iota}, \tag{WD}$$

where $U(\mu, \nu)$ is the set of all probability measures on $\Omega \times \Omega$ having marginals μ and ν. We denote by $W_\iota^\iota(\mu, \nu)$, W_ι to the power ι , i.e. $W_\iota^\iota(\mu, \nu) := (W_\iota(\mu, \nu))^\iota$.

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

$$\Delta_n(\tau) := \left\{ u \in \mathbb{R}^n_+ : \sum_{i=1}^n u_i = \tau \right\}. \tag{2.1}$$

If $\tau = 1$, then $\Delta_n(\tau)$, denoted simply by Δ_n, is the $n+1$ simplex.

[2]The reason is that IBP converges fast but to the solution of an approximate model, not to an exact WB.

Definition 2 (Wasserstein Barycenter)**.** *Given M measures $\{\nu^{(1)}, \dots, \nu^{(M)}\}$ in $P(\Omega)$ and $\alpha \in \Delta_M$, an ι-*Wasserstein barycenter *with weights α is a solution to the following optimization problem*

$$\min_{\mu \in P(\Omega)} \sum_{m=1}^{M} \alpha_m W_\iota^\iota(\mu, \nu^{(m)}) \,. \tag{2.2}$$

A WB μ exists in generality and, if one of the $\nu^{(m)}$ vanishes on all Borel subsets of Hausdorff dimension $dim(\Omega) - 1$, then it is also unique [1]. If the measures are discrete, then uniqueness is no longer ensured in general.

2.2.2 Discrete Wasserstein Barycenter

This work focus on empirical measures based on finitely many R scenarios $\Xi := \{\xi_1, \dots, \xi_R\}$ for ξ and $S^{(m)}$ scenarios $Z^{(m)} := \{\zeta_1^{(m)}, \dots, \zeta_{S^{(m)}}^{(m)}\}$ for $\zeta^{(m)}$, $m = 1, \dots, M$, i.e., measures of the form

$$\mu = \textstyle\sum_{r=1}^{R} p_r \delta_{\xi_r} \quad \text{and} \quad \nu^{(m)} = \sum_{s=1}^{S^{(m)}} q_s^{(m)} \delta_{\zeta_s^{(m)}}, \quad m = 1, \dots, M, \tag{2.3}$$

with δ_u the Dirac unit mass on $u \in \Omega$, $p \in \Delta_R$, and $q^{(m)} \in \Delta_{S^{(m)}}$, $m = 1, \dots, M$.

In this setting, when the support Ξ is fixed, the ι-Wasserstein distance $W_\iota(\mu, \nu)$ of two empirical measures μ and $\nu^{(m)}$ is the ι^{th} root of the optimal value of the following LP, known as *optimal transportation* (OT) problem

$$\mathtt{OT}_\Xi(p, q) := \begin{cases} \min\limits_{\pi \geq 0} & \sum\limits_{r=1}^{R} \sum\limits_{s=1}^{S} \mathtt{d}(\xi_r, \zeta_s)^\iota \pi_{rs} & \\ \text{s.t.} & \sum_{r=1}^{R} \pi_{rs} = q_s, & s = 1, \dots, S \\ & \sum_{s=1}^{S} \pi_{rs} = p_r, & r = 1, \dots, R. \end{cases} \tag{2.4}$$

The feasible set above is refereed to as the *transportation polytope*, issued by the so-called *marginal constraints*. An optimal solution of this problem is known as an optimal transportation plan. Observe that a transportation plan can be represented as a matrix whose entries are non-negative, the row sum equals the marginal q, and the column sum equals p.

Definition 3 (Discrete Wassertein Barycenter - WB)**.** *A Wassertein barycenter of a set of M empirical probability measures $\nu^{(m)}$ having support $Z^{(m)}$, $m = 1, \dots, M$, is a solution to the following optimization problem*

$$\min_{\Xi, p \in \Delta_R} \sum_{m=1}^{M} \alpha_m \mathtt{OT}_\Xi(p, q^{(m)}). \tag{2.5}$$

The above is a challenging nonconvex optimization problem that is in general dealt with via block-coordinate optimization: at iteration k, the support is fixed Ξ^k, and the convex optimization problem, $\min_{p \in \Delta_R} \sum_{m=1}^{M} \alpha_m \mathtt{OT}_{\Xi^k}(p, q^{(m)})$, is solved to define a vector p^k, which is in turn fixed to solve $\min_\Xi \sum_{m=1}^{M} \alpha_m \mathtt{OT}_\Xi(p^k, q^{(m)})$ that updates the support Ξ^k to Ξ^{k+1}. When the metric $\mathtt{d}(\cdot, \cdot)$ is the Euclidean distance and $\iota = 2$, the last problem has a straightforward solution (see for instance [19, Alg. 2] and [72, § II]). For this reason, in the remainder of this work we focus on the more challenging problem of minimizing w.r.t. the vector p.

Definition 4 (Discrete Wasserstein Barycenter with Fixed Support)**.** *A fixed-support Wasserstein barycenter of a set of M empirical probabilities measures $\nu^{(m)}$ having support $Z^{(m)}$, $m = 1, \dots, M$, is a solution to the following optimization problem*

$$\min_{p \in \Delta_R} \sum_{m=1}^{M} \alpha_m \mathtt{OT}_\Xi(p, q^{(m)}). \tag{2.6}$$

In the above definition, the support Ξ is fixed and the optimization is performed with respect to the vector p. Our approach to solve eq. (2.6) can be better motivated after writing down the extensive formulation of the problem. Indeed, looking for a barycenter μ with fixed atoms ξ and probability p is equivalent to solving the following huge-scale LP, where we denote $d_{rs}^{(m)} := \alpha_m \mathtt{d}(\xi_r, \zeta_s^{(m)})$ $(r = 1, \dots, R,\ s = 1, \dots, S^{(m)},\ m = 1, \dots, M)$:

$$\left\{\begin{array}{lllll} \min\limits_{p,\pi} & \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^{(1)}} d_{rs}^{(1)}\pi_{rs}^{(1)} & +\cdots+ & \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^{(M)}} d_{rs}^{(M)}\pi_{rs}^{(M)} & \\ \text{s.t.} & \sum_{r=1}^{R}\pi_{rs}^{(1)} & & = q_s^{(1)}, & s = 1, \dots, S^{(1)} \\ & & \ddots & \vdots & \\ & & & \sum_{r=1}^{R}\pi_{rs}^{(M)} = q_s^{(M)}, & s = 1, \dots, S^{(M)} \\ & \sum_{s=1}^{S^{(1)}}\pi_{rs}^{(1)} & & = p_r, & r = 1, \dots, R \\ & & \ddots & \vdots & \\ & & & \sum_{s=1}^{S^{(M)}}\pi_{rs}^{(M)} = p_r, & r = 1, \dots, R \\ & p \in \Delta_R, \pi^{(1)} \geq 0 & \cdots & \pi^{(M)} \geq 0, & \end{array}\right. \tag{2.7}$$

The constraint $p \in \Delta_R$ above is redundant and can be removed: we always have, for $m = 1, \dots, M$, $\sum_{r=1}^{R} p_r = \sum_{r=1}^{R}\left(\sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)}\right) = \sum_{s=1}^{S^{(m)}}\left(\sum_{r=1}^{R} \pi_{rs}^{(m)}\right) = \sum_{s=1}^{S^{(m)}} q_s^{(m)} = 1$.

The dimension of the above LP is $R(1 + \sum_{m=1}^{M} S^{(m)})$ and grows rapidly with the number of measures M. For instance, for moderate values such as $R = S^{(m)} = 1600$, $m = 1, \dots, M$ (corresponding to figures with 40×40 pixels) and $M = 100$, the above LP has more than 256 millions of variables[3].

Although variable p is the one of interest, we can remove p from the above formulation and recover it easily by working with the following linear subspace.

Definition 5 (Balanced subspace)**.** *We denote by* balanced subspace *the following linear subspace of balanced transportation plans:*

$$\mathcal{B} := \left\{ \pi = (\pi^{(1)}, \dots, \pi^{(M)}) \;\middle|\; \begin{array}{lcll} \sum_{s=1}^{S^{(1)}} \pi_{rs}^{(1)} & = & \sum_{s=1}^{S^{(2)}} \pi_{rs}^{(2)}, & r = 1, \dots, R \\ \sum_{s=1}^{S^{(2)}} \pi_{rs}^{(2)} & = & \sum_{s=1}^{S^{(3)}} \pi_{rs}^{(3)}, & r = 1, \dots, R \\ & \vdots & & \\ \sum_{s=1}^{S^{(M-1)}} \pi_{rs}^{(M-1)} & = & \sum_{s=1}^{S^{(M)}} \pi_{rs}^{(M)}, & r = 1, \dots, R \end{array} \right\}. \tag{2.8}$$

The term *balanced* is due to the fact that given $\nu^{(m)}$ in eq. (2.3), $\sum_{j=1}^{S(m)} q_j^{(m)} = \sum_{j=1}^{S(m')} q_j^{(m')}$, $m, m' = 1, \dots, M$. Observe that problem eq. (2.7) is equivalent, in terms of optimal value and optimal transportation plans, to the

[3] More precisely, 256 001 600 variables.

following LP

$$\begin{cases} \min\limits_{\pi\in\mathcal{B}} & \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^{(1)}} d_{rs}^{(1)}\pi_{rs}^{(1)} & +\cdots+ & \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^{(M)}} d_{rs}^{(M)}\pi_{rs}^{(M)} & \\ \text{s.t.} & \sum_{r=1}^{R}\pi_{rs}^{(1)} & & = q_s^{(1)}, & s=1,\dots,S^{(1)} \\ & & \ddots & \vdots & \\ & & & \sum_{r=1}^{R}\pi_{rs}^{(M)} = q_s^{(M)}, & s=1,\dots,S^{(M)} \\ & \pi^{(1)}\geq 0 & \cdots & \pi^{(M)}\geq 0, & \end{cases} \tag{2.9}$$

having $R\sum_{m=1}^{M} S^{(m)}$ variables. Note furthermore that an optimal vector p can be easily recovered from any given optimal transportation plan $\pi^{(m)}$ by simply setting $p_r = \sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)}$, $r = 1,\dots,R$.

2.3 Discrete Unbalanced Wasserstein Barycenter

A well-known drawback of the above OT-based concepts is their limitation to measures with equal total mass. To overcome this limitation, a simple idea is to relax the marginal constraints in eq. (2.4), giving rise to an extension of the OT problem often referred to as *unbalanced optimal transportation* (UOT) problem [59] because of its ability to cope with "unbalanced" measures, i.e., with different masses. Different manners to relax the marginal constraints yield various UOT problems that can replace OT problems in the barycenter definition eq. (2.6) to yield different generalizations of the Wasserstein barycenters. As a result, unbalanced Wasserstein barycenter (UWB) deals with the fact that in some applications the non-negative vectors $q^{(m)}$, $m = 1,\dots,M$, are not necessarily probability related: they do not live in any simplex, i.e., $\sum_{j=1}^{S(m)} q_j^{(m)} \neq \sum_{j=1}^{S(m')} q_j^{(m')}$ for at least a pair (m, m') s.t. $m \neq m'$.

In this case, the WB problem eq. (2.7) is infeasible and a (balanced) WB is not defined. As an example, think of p as the quantity of goods to be produced, and $q^{(m)}$ as an event of the random demand for these goods. Since the demand events are different, we can not decide on a production p that satisfies all the M future demands exactly: the production of some goods might be overestimated, while for others, underestimated. Hence, it makes sense to produce p that minimizes not only transportation costs but also the (expectation w.r.t. α of the) mismatches between production p and demand $q^{(m)}$. Such an intention can be mathematically formulated in several ways. In this work, we propose a simple one by using a metric that measures the distance of a multi-plan $\pi = (\pi^{(1)},\dots,\pi^{(M)})$ to the balanced subspace $\mathcal{B}$ defined in eq. (2.8). We take such a metric as being the Euclidean distance $\texttt{dist}_{\mathcal{B}}(\pi) := \min_{\theta\in\mathcal{B}} \|\theta - \pi\| = \|\texttt{Proj}_{\mathcal{B}}(\pi) - \pi\|$, and define the following non-linear optimization problem, with $\gamma > 0$ a penalty parameter:

$$\begin{cases} \min\limits_{\pi} & \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^{(1)}} d_{rs}^{(1)}\pi_{rs}^{(1)} & +\cdots+ & \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^{(M)}} d_{rs}^{(M)}\pi_{rs}^{(M)} & + \gamma\,\texttt{dist}_{\mathcal{B}}(\pi) \\ \text{s.t.} & \sum_{r=1}^{R}\pi_{rs}^{(1)} & & = q_s^{(1)}, & s=1,\dots,S^{(1)} \\ & & \ddots & \vdots & \\ & & & \sum_{r=1}^{R}\pi_{rs}^{(M)} = q_s^{(M)}, & s=1,\dots,S^{(M)} \\ & \pi^{(1)}\geq 0 & \cdots & \pi^{(M)}\geq 0. & \end{cases} \tag{2.10}$$

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. (2.10) is a relaxation of eq. (2.9). In the unbalanced setting, any feasible point to eq. (2.10) 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 6 (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. (2.10), and $\tilde{\pi}$ the projection of $\bar{\pi}$ onto the balanced subspace $\mathcal{B}$, that is, $\tilde{\pi} := \texttt{Proj}_{\mathcal{B}}(\bar{\pi})$ (≥ 0). The vector $\bar{p}_r := \sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m)}, r = 1, \dots, R$ (no matter $m = 1, \dots, M$) 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, that also relaxes the constraints $\sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)}$, see for instance [32, 59]. Although the above definition is not as general as the ones of the latter references, it is important to highlight that our UBW definition provides meaningful results (see Section 2.6.4 below), uniqueness of the barycenter (if unbalanced), and is indeed an extension of definition 4.

Proposition 1. *Suppose that $\{\nu^{(1)}, \dots, \nu^{(M)}\}$ are probability measures and let $\gamma > \|\texttt{vec}(d)\|$, in problem eq. (2.10), with $\texttt{vec}(d)$ the vectorization of the matrix $d \in \mathbb{R}^{R \times \sum_{m=1}^{M} S^{(m)}}$. Then any UWB according to definition 6 is also a (balanced) WB.*

Proof. Observe that the linear function $\sum_{m=1}^{M} \sum_{r=1}^{R} \sum_{s=1}^{S^{(m)}} d_{rs}^{(m)} \pi_{rs}^{(m)}$ is obviously Lipschitz continuous with constant $\|\texttt{vec}(d)\|$. Thus, the standard theory of exact penalty methods in optimization (see for instance [11, Prop. 1.5.2]) ensures that, when $\gamma > \|\texttt{vec}(d)\|$, then $\bar{\pi}$ solves[4] problem eq. (2.10) if and only if $\bar{\pi}$ solves eq. (2.9). As a result, $\bar{\pi} = \texttt{Proj}_{\mathcal{B}}(\bar{\pi})$ and definition 6 boils down to definition 4. □

Another advantage of our definition is that the problem yielding the proposed UBW enjoys a favorable structure that can be efficiently exploited by splitting methods.

At the first glance, computing a UWB seems much more challenging than computing a WB: the former is obtained by solving a nonlinear optimization problem followed by the projection onto the balanced subspace, while the latter is a solution of an LP. However, in practice, the LP problem eq. (2.9) is already too large to be solved directly by off-the-shelf solvers and thus decomposition techniques need to come into play. In the next section we show that the computational burden to solve either the LP eq. (2.9) or the nonlinear problem eq. (2.10) by the Douglas-Rachford splitting method is the same. Indeed, it turns out that both problems can be efficiently solved by the algorithm presented in Section 2.5.3.

2.4 Problem reformulation and the DR algorithm

In this section, we reformulate problems eq. (2.9) and eq. (2.10) in a suitable way so that the Douglas Rachford splitting operator method can be easily deployed to compute a barycenter in the balanced and unbalanced settings under the following assumptions: (i) each of the M measures $\nu^{(m)}$ are empirical ones, described by a list of atoms whose weights are $q^{(m)} \in \mathbb{R}_+^{S^{(m)}}$; (ii) the search for a barycenter is considered on a finitely fixed support of R atoms with weights $p \in \mathbb{R}_+^R$.

To this end, we start by first defining the following convex and compact sets

$$\Pi^{(m)} := \left\{ \pi^{(m)} \geq 0 : \sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)},\ s = 1, \dots, S^{(m)} \right\},\ m = 1, \dots, M. \tag{2.11}$$

[4] Note that in the balance case, the objective function of problem eq. (2.10) is no longer strictly convex on the feasible set, and thus multiple solutions may exist.

These are the sets of transportation plans $\pi^{(m)}$ with right marginals $q^{(m)}$. The set with left marginals has already been characterized by the linear subspace $\mathcal{B}$ of balanced plans eq. (2.8).

With the help of 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, we can define the convex functions

$$f^{(m)}(\pi^{(m)}) := \sum_{r=1}^{R} \sum_{s=1}^{S^{(m)}} d_{rs}^{(m)} \pi_{rs}^{(m)} + \mathbf{i}_{\Pi^{(m)}}(\pi^{(m)}), \quad m = 1, \dots, M, \tag{2.12}$$

and recast problems eq. (2.9) and eq. (2.10) in the following more general setting

$$\min_{\pi} \; f(\pi) + g(\pi)\,, \text{ with :} \tag{2.13a}$$

$$f(\pi) := \sum_{m=1}^{M} f^{(m)}(\pi^{(m)}) \quad \text{and} \quad g(x) := \begin{cases} \mathbf{i}_{\mathcal{B}}(\pi) & \text{if balanced} \\ \gamma\, \mathtt{dist}_{\mathcal{B}}(\pi) & \text{if unbalanced.} \end{cases} \tag{2.13b}$$

Since f is polyhedral and eq. (2.13) is solvable, computing one of its solution is equivalent to

$$\text{find} \quad \pi \quad \text{such that} \quad 0 \in \partial f(\pi) + \partial g(\pi). \tag{2.14}$$

Recall that the subdifferential of a lower semicontinuous convex functions is a maximal monotone operator. 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 [3]). Among the existing algorithms, the Douglas-Rachford operator splitting method [25] (see also [3, § 25.2 and § 27.2]) is the most popular one. When applied to problem eq. (2.14), the DR algorithm asymptotically computes a solution by repeating the following steps, with $k = 0, 1, \dots$ and given initial point $\theta^0 = (\theta^{(1),0}, \dots, \theta^{(M),0})$ and prox-parameter $\rho > 0$:

$$\begin{cases} \pi^{k+1} &= \arg\min\limits_{\pi} \; g(\pi) + \dfrac{\rho}{2}\|\pi - \theta^k\|^2 \\ \hat{\pi}^{k+1} &= \arg\min\limits_{\pi} \; f(\pi) + \dfrac{\rho}{2}\|\pi - (2\pi^{k+1} - \theta^k)\|^2 \\ \theta^{k+1} &= \theta^k + \hat{\pi}^{k+1} - \pi^{k+1}. \end{cases} \tag{2.15}$$

By noting that f and g in eq. (2.13b) are lower semicontinuous convex functions and problem eq. (2.13) is solvable, the following is a direct consequence of Theorem 25.6 and Corollary 27.4 of [3].

Theorem 1. *The sequence $\{\theta^k\}$ produced by the DR algorithm eq. (2.15) 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. (2.13);*
- $\{\pi^k\}$ *and* $\{\hat{\pi}^k\}$ *converges to* $\bar{\pi}$.

The DR algorithm is attractive when the two first steps in eq. (2.15) 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. (2.13b) has an additive structure, the computation of $\hat{\pi}^{k+1}$ in eq. (2.15) 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 literature to produce the following randomized version of the DR algorithm eq. (2.15), with α the vector of weights in eq. (2.2):

$$\left\{\begin{array}{lcl} \pi^{k+1} & = & \arg\min\limits_{\pi}\ g(\pi)+\dfrac{\rho}{2}\|\pi-\theta^k\|^2 \\ & & \text{Draw randomly } m\in\{1,2,\ldots,M\} \text{ with probability } \alpha_m>0 \\ \hat{\pi}^{(m),k+1} & = & \arg\min\limits_{\pi^{(m)}}\ f^{(m)}(\pi^{(m)})+\dfrac{\rho}{2}\|\pi^{(m)}-(2\pi^{(m),k+1}-\theta^{(m),k})\|^2 \\ \theta^{(m'),k+1} & = & \left\{\begin{array}{ll} \theta^{(m),k}+\hat{\pi}^{(m),k+1}-\pi^{(m),k+1} & \text{if } m'=m \\ \theta^{(m'),k} & \text{if } m'\neq m. \end{array}\right. \end{array}\right. \tag{2.16}$$

The randomized DR algorithm eq. (2.16) aims at reducing 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 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 the objective function of eq. (2.6) (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. (2.16) 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 2.6.2.2 in the numerical section). The almost surely (i.e., with probability one) convergence of the randomized DR algorithm depicted in eq. (2.16) can be summarized as follows. We refer the interest reader to Theorem 2 in [38] for the proof (see also additional comments in the Appendix of [2]).

Theorem 2. *The sequence $\{\pi^k\}$ produced by the randomized DR algorithm eq. (2.16) converges almost surely to a solution $\bar{\pi}$ of problem eq. (2.13).*

In the next section we further exploit the structure of functions f and g in eq. (2.13) and re-arrange terms in the schemes eq. (2.15) and eq. (2.16) to provide an easy-to-implement and memory-efficient algorithm for computing balanced and unbalanced WBs.

2.5 The Method of Averaged Marginals

Both deterministic and randomized DR algorithms above require evaluating the proximal mapping of function g given in eq. (2.13b).

In the balanced WB setting, g is the indicator function of the balanced subspace $\mathcal{B}$ given in eq. (2.8). Therefore, the solution π^{k+1} above is nothing but 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: $\min\limits_{\pi}\ \texttt{dist}_{\mathcal{B}}(\pi)+\dfrac{\rho}{2\gamma}\|\pi-\theta^k\|^2$. The unique solution to this problem is well-known to be given by

$$\pi^{k+1}=\left\{\begin{array}{ll} \texttt{Proj}_{\mathcal{B}}(\theta^k) & \text{if } \rho\,\texttt{dist}_{\mathcal{B}}(\theta^k)\leq\gamma \\ \theta^k+\frac{\gamma}{\rho\,\texttt{dist}_{\mathcal{B}}(\theta^k)}(\texttt{Proj}_{\mathcal{B}}(\theta^k)-\theta^k) & \text{otherwise.} \end{array}\right. \tag{2.17}$$

Hence, computing π^{k+1} in both balanced and unbalanced settings boils down to projecting onto the balanced subspace (recall that $\texttt{dist}_{\mathcal{B}}(\theta) = \|\texttt{Proj}_{\mathcal{B}}(\theta) - \theta\|$). This fact allows us to provide a unified algorithm for WB and UWB problems.

2.5.1 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 set.

Proposition 2. *With the notation of Section 2.2, let* $\theta \in \mathbb{R}^{R\times\sum_{m=1}^{M} S^{(m)}}$,

$$a_m := \frac{\frac{1}{S^{(m)}}}{\sum_{j=1}^{M}\frac{1}{S^{(j)}}}, \quad p^{(m)} := \sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m)}, \quad \textit{and} \quad p := \sum_{m=1}^{M} a_m p^{(m)}. \tag{2.18a}$$

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

$$\pi_{rs}^{(m)} := \theta_{rs}^{(m)} + \frac{(p_r - p_r^{(m)})}{S^{(m)}}, \quad s = 1,\dots,S^{(m)},\ r = 1,\dots,R,\ m = 1,\dots,M. \tag{2.18b}$$

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

$$\begin{cases} \min\limits_{y^{(1)},\dots,y^{(M)}} & \frac{1}{2}\sum_{m=1}^{M}\|y^{(m)} - \theta^{(m),k}\|^2 & & \\ \text{s.t} & \sum_{s=1}^{S^{(1)}} y_{rs}^{(1)} & = \sum_{s=1}^{S^{(2)}} y_{rs}^{(2)}, & r = 1,\dots,R \\ & \sum_{s=1}^{S^{(2)}} y_{rs}^{(2)} & = \sum_{s=1}^{S^{(3)}} y_{rs}^{(3)}, & r = 1,\dots,R \\ & & \vdots & \\ & \sum_{s=1}^{S^{(M-1)}} y_{rs}^{(M-1)} & = \sum_{s=1}^{S^{(M)}} y_{rs}^{(M)}, & r = 1,\dots,R, \end{cases} \tag{2.19}$$

which is only coupled by the "columns" of π: there is no constraint liking $\pi_{rs}^{(m)}$ 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

$$\begin{cases} \min\limits_{w} & \frac{1}{2}\sum_{m=1}^{M}\sum_{s=1}^{S^{(m)}}\left(w_s^{(m)} - \theta_{rs}^{(m)}\right)^2 \\ \text{s.t} & \sum_{s=1}^{S^{(1)}} w_s^{(1)} = \sum_{s=1}^{S^{(2)}} w_s^{(2)} \\ & \sum_{s=1}^{S^{(2)}} w_s^{(2)} = \sum_{s=1}^{S^{(3)}} w_s^{(3)} \\ & \vdots \\ & \sum_{s=1}^{S^{(M-1)}} w_s^{(M-1)} = \sum_{s=1}^{S^{(M)}} w_s^{(M)}. \end{cases} \tag{2.20}$$

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

$$L_r(w,u) = \frac{1}{2}\sum_{m=1}^{M}\sum_{s=1}^{S^{(m)}}\left(w_s^{(m)} - \theta_{rs}^{(m)}\right)^2 + \sum_{m=1}^{M-1} u^{(m)}\Big(\sum_{s=1}^{S^{(m)}} w_s^{(m)} - \sum_{s=1}^{S^{(m+1)}} w_s^{(m+1)}\Big). \tag{2.21}$$

A primal-dual (w,u) solution to problem eq. (2.20) 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,

$$\begin{cases} \pi_{rs}^{(1)}-\theta_{rs}^{(1)} & + & u^{(1)} & = 0 & s=1,\dots,S^{(1)} \\ \pi_{rs}^{(2)}-\theta_{rs}^{(2)} & + & u^{(2)}-u^{(1)} & = 0 & s=1,\dots,S^{(2)} \\ & \vdots & & & \\ \pi_{rs}^{(M-1)}-\theta_{rs}^{(M-1)} & + & u^{(M-1)}-u^{(M-2)} & = 0 & s=1,\dots,S^{(M-1)} \\ \pi_{rs}^{(M)}-\theta_{rs}^{(M)} & - & u^{(M-1)} & = 0 & s=1,\dots,S^{(M)}. \end{cases} \tag{2.22}$$

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. (2.18a)), and sum, above over s, the first row of system eq. (2.22) to get

$$p_r-p_r^{(1)}+u^{(1)}S^{(1)}=0 \quad\Rightarrow\quad u^{(1)}=\frac{p_r^{(1)}-p_r}{S^{(1)}}, \tag{2.23}$$

Now, by summing the second row in eq. (2.22) over s we get

$$p_r-p_r^{(2)}+u^{(2)}S^{(2)}-u^{(1)}S^{(2)}=0 \quad\Rightarrow\quad u^{(2)}=u^{(1)}+\frac{p_r^{(2)}-p_r}{S^{(2)}}. \tag{2.24}$$

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

$$u^{(m)}=u^{(m-1)}+\frac{p_r^{(m)}-p_r}{S^{(m)}}, \quad m=1,\dots,M-1. \tag{2.25a}$$

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. (2.22) to conclude that the r^{th} row of $\pi=\texttt{Proj}_{\mathcal{B}}(\theta)$ is given as in eq. (2.18b). It is remaining to show that $p_r=\sum_{s=1}^{S^{(m)}}\pi_{rs}^{(m)}$, as defined above, is alternatively given by eq. (2.18a). To this end, observe that $u^{(M-1)}=u^{(M-1)}-u^{(0)}=\sum_{m=1}^{M-1}(u^{(m)}-u^{(m-1)})$, so:

$$u^{(M-1)}=\sum_{m=1}^{M-1}\left(\frac{p_r^{(m)}-p_r}{S^{(m)}}\right)=\sum_{m=1}^{M-1}\frac{p_r^{(m)}}{S^{(m)}}-p_r\sum_{m=1}^{M-1}\frac{1}{S^{(m)}}. \tag{2.26}$$

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. (2.26) yields

$$p_r=S^{(M)}\Big[\frac{p_r^{(M)}}{S^{(M)}}+u^{(M-1)}\Big]=S^{(M)}\Big[\frac{p_r^{(M)}}{S^{(M)}}+\sum_{m=1}^{M-1}\frac{p_r^{(m)}}{S^{(m)}}-p_r\sum_{m=1}^{M-1}\frac{1}{S^{(m)}}\Big], \tag{2.27}$$

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. (2.18a), and the proof is thus 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.

2.5.2 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. (2.15)). Given the addictive structure of f in eq. (2.13b), the above problem can be decomposed into M smaller ones

$$\min_{\pi^{(m)}} \; f^{(m)}(\pi^{(m)}) + \frac{\rho}{2}\|\pi^{(m)} - y^{(m)}\|^2, \quad m = 1, \dots, M. \tag{2.28}$$

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. (2.1). The proximal mapping $\hat{\pi} := \min_\pi \; f(\pi) + \frac{\rho}{2}\|\pi - y\|^2$ can be computed exactly, in parallel along the columns of each transport plan $y^{(m)}$, as follows: for all $m \in \{1, \dots, M\}$,*

$$\begin{pmatrix} \hat{\pi}_{1s}^{(m)} \\ \vdots \\ \hat{\pi}_{Rs}^{(m)}) \end{pmatrix} = \texttt{Proj}_{\Delta_R(q_s^{(m)})} \begin{pmatrix} y_{1s} - \frac{1}{\rho} d_{1s}^{(m)} \\ \vdots \\ y_{Rs} - \frac{1}{\rho} d_{Rs}^{(m)} \end{pmatrix}, \quad s = 1, \dots, S^{(m)}. \tag{2.29}$$

Proof. It has already argued that evaluating this proximal mapping into M smaller subproblems eq. (2.28), which is a quadratic program problem due to the definition of $f^{(m)}$ in eq. (2.12):

$$\begin{cases} \min\limits_{\pi^{(m)} \geq 0} & \sum\limits_{r=1}^{R} \sum\limits_{s=1}^{S^{(m)}} \left[d_{rs}^{(m)} \pi_{rs}^{(m)} + \frac{\rho}{2} \|\pi_{rs}^{(m)} - y_{rs}^{(m)})\|^2 \right] \\ \text{s.t.} & \sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)}, \; s = 1, \dots, S^{(m)}. \end{cases} . \tag{2.30}$$

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

$$\begin{cases} \min\limits_{w \geq 0} & \sum\limits_{r=1}^{R} \left[d_{rs}^{(m)} w_r + \frac{\rho}{2} (w_r - y_{rs}^{(m)})^2 \right] \\ \text{s.t.} & \sum_{r=1}^{R} w_r = q_s^{(m)}, \end{cases} \tag{2.31}$$

which is nothing but eq. (2.29). Such projection can be performed exactly [17]. □

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 [17]. Hence, evaluating the proximal mapping in proposition 3 decomposes into $\sum_{m=1}^{M} S^{(m)}$ independent projections onto Δ_R.*

2.5.3 The Method of Averaged Marginals (MAM)

Putting propositions 2 and 3 together with the general lines of DR algorithm eq. (2.15) and rearranging terms we provide below an easy-to-implement and memory efficient algorithm for computing barycenters. The pseudo code for this algorithm is presented in algorithm 2. The algorithm gathers the DR's three main steps and integrates

an option in case the problem is unbalanced, since treating the Wasserstein barycenter problem the way we did, enables to easily switch from the balanced to the unbalanced case. Note that part of the first DR step has been placed at the end of the *while-loop* iteration in a storage optimization purpose that will be discussed in the following paragraphs. In the following algorithm, the vector $\alpha \in \Delta_M$ of weights is included in the distance matrix definition, as done in eq. (2.12). Some comments on algorithm 2 are in order.

Algorithm 2 Method of Averaged Marginals - MAM

▷ Step 0: input

1: Given $\rho > 0$, the distance matrix and initial point $d, \theta^0 \in \mathbb{R}^{R \times \sum_{m=1}^M S^{(m)}}$, and $a \in \Delta_M$ as in eq. (2.18a), 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 $w_r \leftarrow \theta_{rs}^{(m),k} + 2\, t^k \frac{p_r^k - p_r^{(m)}}{S^{(m)}} - \frac{1}{\rho} d_{rs}^{(m)}$, $r = 1, \dots, R$

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 $\bar{p} \leftarrow p^k$

MAM's interpretation A simple interpretation of the *Method of Averaged Marginals* is as follows: 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 an exact barycenter upon specific assumptions on the choice of the index set at line 6 of algorithm 2.

Initialization The algorithm initialization, the choices for $\theta^0 \in \mathbb{R}^{R \times \sum_{m=1}^M S^{(m)}}$ and $\rho > 0$ are arbitrary ones. The prox-parameter $\rho > 0$ is borrowed from the DR algorithm, which is known to has 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 [68, 70] and can be adapted to the setting

of algorithm 2.

Stopping criteria A possible stopping test for the algorithm, with mathematical justification, is to terminate the iterative process as soon as $\|\theta^{k+1} - \theta^k\|_\infty \leq \texttt{Tol}$, where $\texttt{Tol} > 0$ is a given tolerance. In practical terms, this test boils down to checking whether $|\theta_{rs}^{(m),k} + t^k(p_r^k - p_r^{(m)})/S^{(m)} - \hat{\pi}_{rs}| \leq \texttt{Tol}$, for all $r = 1, \dots, R$, $s = 1, \dots, S^{(m)}$, and all $m = 1, \dots, M$. Alternatively, we may stop the algorithm when $\|p^{k+1} - p^k\|$ is small enough. The latter should be understood as a heuristic criterion.

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 algorithm. This is the reason for having the possibility of choosing an index set $\mathcal{M}^k \subseteq \{1, \dots, M\}$ at line 6 of algorithm 2. 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 $\mathcal{M}^k = A_i$.*

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 2.6.2.2, 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 $\mathcal{M}^k \subseteq \{1, \dots, M\}$ 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 distance matrix $d^{(m)}$ 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 2.6.1 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 $d \in \mathbb{R}^{R \times S^{(m)}}$, 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 2 can be easily removed from the algorithm's implementation by merging lines 9-11 into a single one. Hence, for $T := \sum_{m=1}^M S^{(m)}$, the method's memory allocation is $2RT + T + M(R+1)$ floating-points. This number can be reduced if the measures share the same distance matrix, i.e., $d^{(m)} = d^{(m')}$ 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 $RT + RS + T + M(R+1)$ floating-points. Within the light of the previous remark this memory complexity should be treated as an upper bound: the sparser are the data the less memory will be needed.

Balanced and unbalanced settings As already mentioned, our approach can handle both balanced and unbalanced WB problems. All that is necessary it 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 2 defines $t^k = 1$ for all iterations if the WB problem is balanced (because $\gamma = \infty$ in this case)[5], 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. (2.17). 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 2 can be summarized as follows.

Theorem 3 (MAM's convergence analysis)**.** *a) (Deterministic MAM.) Consider algorithm 2 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 2 with the choice $\mathcal{M}^k \subseteq \{1, \dots, m\}$ 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 2 is an implementation of the (randomized) DR algorithm and invoke theorem 1 for item a) and theorem 2 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 $\theta_{rs}^{(m),k} + \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}$, $s = 1, \dots, S^{(m)}$, $r = 1, \dots, R$, $m = 1, \dots, M$, 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 $k = 0$ 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 2 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,

$$\pi_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + t^k \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}, \quad s = 1, \dots, S^{(m)},\ r = 1, \dots, R,\ m = 1, \dots, M. \tag{2.32}$$

In the balanced case, $t^k = 1$ for all k (because $\gamma = \infty$) and thus π^{k+1} is as in eq. (2.18b). Otherwise, π^{k+1} is as in eq. (2.17) (see the comments after algorithm 2). In both cases, π^{k+1} coincides with the auxiliary variable at the first step of the DR scheme eq. (2.15) (see the developments at the beginning of this section). Next, observe that to perform the second step of eq. (2.15) 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 2 yields, thanks to proposition 3, $\hat{\pi}^{k+1}$ as at the second step of eq. (2.15). Furthermore, the updating of θ^{k+1} in the latter coincides with the rule in algorithm 2: 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 2 is the DR Algorithm eq. (2.15) applied to the WB eq. (2.13). theorem 1 thus ensures that the sequence $\{\pi^k\}$ as defined above converges to some $\bar{\pi}$ solving

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

eq. (2.13). 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 $\tilde{\pi}_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + \frac{p_r^k - p_r^{(m)}}{S^{(m)}}$ $s = 1, \dots, S^{(m)}$, $r = 1, \dots, R$, $m = 1, \dots, M$, 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:

$$\bar{p}_r := \sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m)} = \lim_{k\to\infty} \sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m),k+1} = \lim_{k\to\infty} p_r^k. \tag{2.33}$$

We have shown that the whole sequence $\{p^k\}$ converges to $\bar{p}$. By recalling that $\bar{\pi}$ solves eq. (2.13), we conclude that in the balanced setting $\tilde{\pi} = \bar{\pi}$ and thus $\bar{p}$ is a WB according to definition 4. On the other hand, in the unbalanced setting, $\bar{p}$ above is a γ-unbalanced WB according to definition 6.

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 2 (and not on theorem 1 as previously done) to conclude that $\{\pi^k\}$ converges almost surely to some $\bar{\pi}$ solving eq. (2.13). 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 $\bar{p}$. □

2.6 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 from download in the link https://ifpen-gitlab.appcollaboratif.fr/detocs/mam_wb.

2.6.1 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. Indeed, on the one hand, the denser are the distributions, the greater RAM would be needed to store the data per transport plan (see the management of *storage complexity* in Section 2.5.3). On the other hand, the more distributions are treated, the more transport plans would be stored. In both of these configurations, the time per iteration is meant to grow, either because a processor would need to project more columns onto the respected simplex within *Step 2*, or because *Step 2* is repeated as many time as the number of distribution M (see algorithm 2). The dataset at hand, inspired from [9, 20], has been naturally built to control the sparsity (or respectively, density) of the distributions (see section 2.6.1 and section 2.6.1). Note that each image is normalized making it a representation of a probability distribution. The density of a dataset is controlled by the number of nested ellipses: as exemplified in section 2.6.1 and section 2.6.1, measures with only a single ellipse are very sparse, while a dataset with 5 nested ellipses is denser.

In this first experiments we analyze the impact over MAM caused by the sparsity and number of measures. We have set $\rho = 100$ without proper tuning for every dataset. The study has been carried out with one processor to avoid CPU communication management.

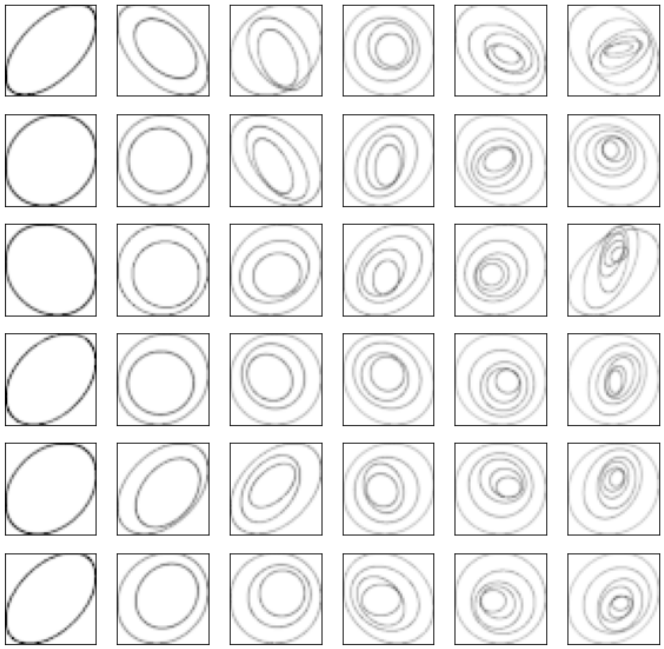

Figure 2.1: Sample of the artificial nested ellipses datasets. First column is taken from the first dataset with 1 ellipse, second columns from the second dataset with 2 nested ellipses, until the sixth column with 6 nested ellipses.

Table 2.1: Mean density with the number of nested ellipses. The density has been calculated by averaging the ratio of non-null pixel per images over 100 generated pictures for each dataset sharing the same number $n_{ellipses}$ of nested ellipses.

$n_{ellipses}$	1	2	3	4	5	6
Density (%)	29.0	51.4	64.3	70.9	73.5	75.0

section 2.6.1 shows that, as expected, the execution time of an iteration increases with increasing density and number of measures. When compared to density it can be seen that the number of measures has greater influence on the method's speed (such a 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 term of speed. Such a result is to be put in regard with algorithms such as B-ADMM [72] that are particularly shaped for sparse dataset but less efficient for denser ones. This is a significant point that will be further developed in Section 2.6.2.3.

2.6.2 Comparison with IBP

The Iterative Bregman Projection (IBP) [9] is a state-of-the-art algorithm for computing Wasserstein barycenters. As mentionned in the Introduction, IBP employes a regularizing function parametrized by $\lambda > 0$. The greater the λ, the worst the approximation. But in practice, λ has to be kept in a moderate magnitude to avoid numerical

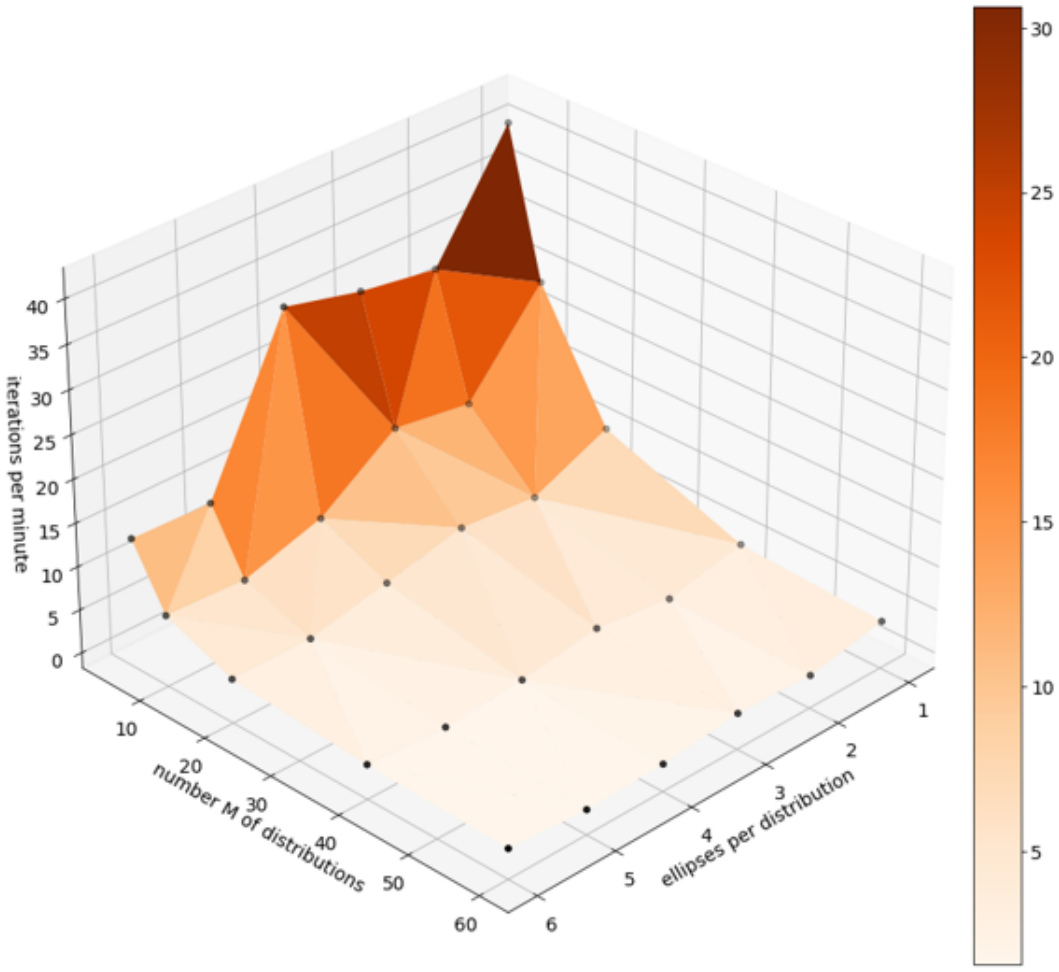

Figure 2.2: Evolution of the number of iterations per minute depending on the density or the number of distributions.

errors (double-precision overflow). IBP is very sensitive to λ, that strongly relies on the dataset at stake. Thus IBP is an inexact method, whereas MAM is exact. Although the study below shows certain advantages of MAM, we make it clear that the aim is not to demonstrate which algorithm is better in general but instead to highlight the differences between the two methods and their advantages depending on the use. Note that the code for IBP is inspired from the original [49].

2.6.2.1 Qualitative comparison

Here we use 100 images per digit of the MNIST database [65] where each digit has been randomly translated and rotated. Each image has 40×40 pixels and can be treated as probability distributions thanks to a normalization, where the pixel location is the *support* and the pixel intensity the *probability*. In section 2.6.2.1, we display intermediate barycenter 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-precison 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).

As illustrated in section 2.6.2.1, for each dataset, IBP gets quickly to a stable approximation of a barycenter. Such a point is obtained shortly after with MAM (less than 5 to 10 seconds after) but MAM continues to converge toward a sharper solution (closer to the exact solution as exemplified quantitatively in section 2.6.2.2). It is clear that the more CPUs used for MAM the better. We have limited the study to a dozen of CPU to allow the reader to reproduce the experimentations. While IBP is not well shaped for CPU parallelization [9, 49, 72], MAM offers a clear advantage depending on the hardware at stake.

2.6.2.2 Quantitative comparison

Next we benchmark MAM, randomized MAM and IBP on a dataset with 60 images per digit of the MNIST database [65] 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 2.6.2.1. Then, at every time step an approximation of the computed barycenter is stored, to compute the error $\bar{W}_2^2(p^k) - \bar{W}_2^2(p_{exact}) := \sum_{m=1}^M \frac{1}{M} \mathtt{OT}(p^k, q^{(m)}) - \sum_{m=1}^M \frac{1}{M} \mathtt{OT}(p_{exact}, q^{(m)})$. All methods were implemented in *python* using a *MPI* based parallelization. Note that IBP is inspired from the code of G. Peyré [49], MAM from algorithm 2 and MAM-randomized (remark 2) has only one distribution treated by processor. section 2.6.2.2 displays the evolution w.r.t time, of the error measure $\bar{W}_2^2(p^k) - \bar{W}_2^2(p_{exact})$, with p_{exact} an exact barycenter obtained by solving LP eq. (2.7) directly.

It is clear that IBP is almost 10 time faster per iteration. However IBP computes an exact solution of an approximated problem that is tuned through the hyperparameter λ (see [9]). Therefore it is natural to witness IBP converging to a solution to the approximated problem, but not to an exact WB. While MAM does converge to an exact solution. So there is a threshold where the accuracy of MAM exceeds IBP: in our case, around 200s - for the computation with the greatest number of processors (see section 2.6.2.2). Such a treshold always exists depending on the computational means (hardware).

This quantitative study explains what have been exemplified with the images of Section 2.6.2.1: the accuracy of IBP is bounded by the choice of λ, itself bounded by an overflow error, while MAM hyperparameters only impact the convergence speed and the algorithm is always improving towards an exact solution. For this dataset, the WB computed by IBP is within 2% of accuracy and thus reasonably good. However, as shown in Table 1 in [72], one can choose other datasets where IBP's accuracy might be unsatisfactory.

Furthermore, section 2.6.2.2 exemplifies an interesting asset of randomized variants of MAM: for some configurations randomized-MAM is more efficient than (deterministic) MAM but for others, the latter seems to be more

Figure 2.3: (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 IBP evolution of its barycenter approximation, the middle panel is MAM evolutions using 10 CPU and the right-hand-side is the exact solution computed by applying *Gurobi* to the LP eq. (2.7).

effective. Note that the curve *MAM 1-random, 1 processor* does not appear on the figure: this is because it is above the y-axis value range due to its bad performance. Indeed, there is a trade-off between time spent per iteration and precision gained after an iteration. For example, with 10 processors, each processor treats 6 measures in the deterministic MAM but only one is treated in the randomized MAM. Therefore, the time spent per iteration is roughly six time shorter in the latter and this counterbalances the loss of accuracy per iterations. On the other hand, when using 20 processors, only 3 measures are treated by each processor and the trade-off is not worth it

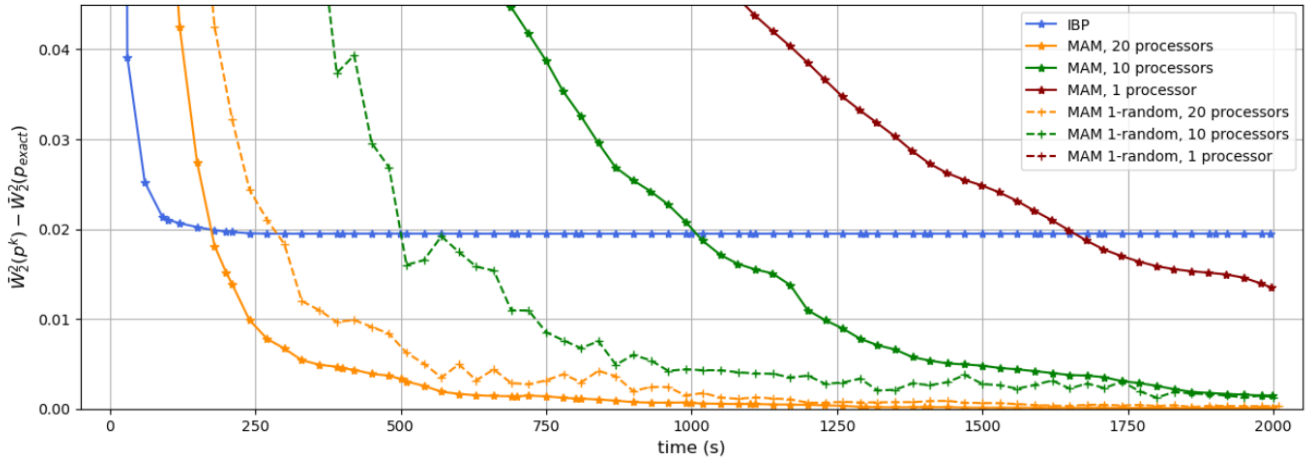

Figure 2.4: Evolution with respect to time of the difference between the Wasserstein barycenter distance of an approximation, $\bar{W}_2^2(p^k)$, and the Wasserstein barycentric distance of the exact solution $\bar{W}_2^2(p_{exact})$ given by the LP. The time step between two points is 30 seconds.

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 dataset and hardware at use. 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.

2.6.2.3 Influence of the support

This section is echoing Section 2.6.1 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 2.6.2.2: 60 pictures of 3's taken from the classic MNIST database [65]. The second dataset is also composed by these 60 images but each digit has been randomly translated and rotated in the same way as in section 2.6.2.1. Therefore, the union of the support of the second dataset is greater than the first one, as illustrated in section 2.6.2.3.

section 2.6.2.3 presents two graphs that have been obtained just as in section 2.6.2.2, but displaying the evolution w.r.t time in percentage: $\Delta W_{\%} := \frac{\bar{W}_2^2(p^k) - \bar{W}_2^2(p_{exact})}{\bar{W}_2^2(p_{exact})} \times 100$. Once more, the hyperparameters have been fully tuned. The hyperparameter of the IBP method is smaller for the second datacase. Indeed, as stated in [72], the greater is the support, the stronger are the restrictions on λ. And since the smaller is λ the further is the approximated problem to the exact one this is expected to witness rising differences between on the following graphs.

Being an exact method, MAM is insensitive to support size. The density of the dataset has little impact on the convergence time as explained in Section 2.6.1 and exemplified in section 2.6.2.3. Such visual results concerning IBP initialization and parametrization have already been discussed in section 2.6.2.1, some other qualitative results can be found in [55] where the author shows that properties of the distributions can be lost due to the entropy penalization in IBP.

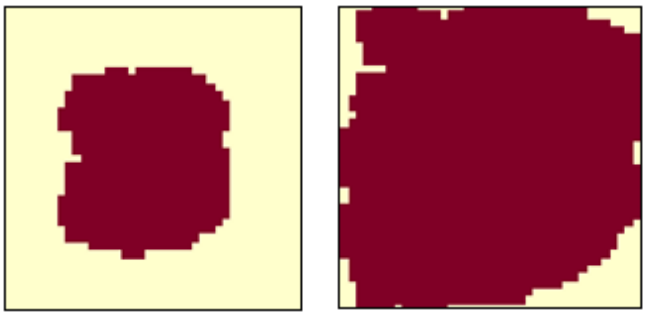

Figure 2.5: Images with 40×40 pixel grid, where the red represents the pixels which are in the union of the dataset support composed with 60 images. (left) for the standard MNIST, (right) for the randomly translated and rotated MNIST.

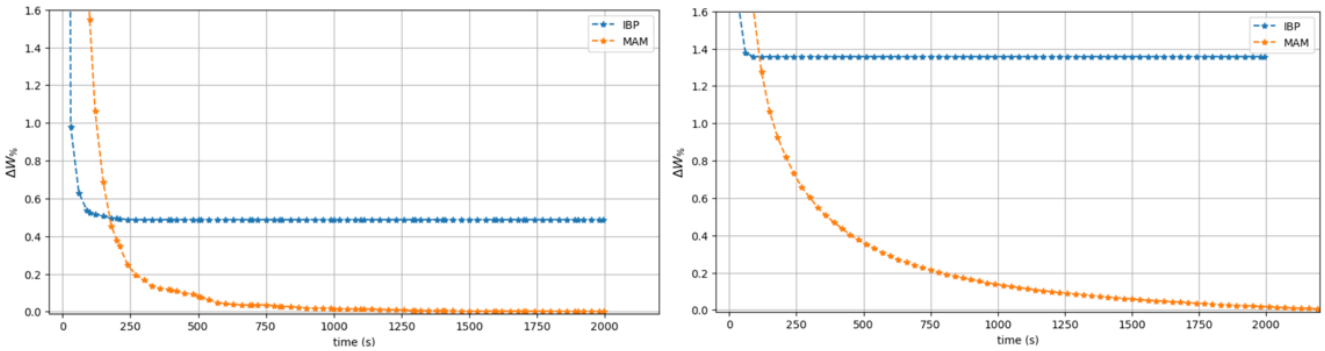

Figure 2.6: Evolution of the percentage of the distance between the exact solution of the barycenter problem and the computed solution using IBP and MAM method with 20 processors: (left) for the standard MNIST, (right) for the randomly translated and rotated MNIST.

2.6.3 Comparison with B-ADMM

This subsection compares MAM with the algorithm B-ADMM of [67] 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 [67, Table I] that IBP is outperformed by B-ADMM in this dataset. As in [67, 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 $R = 60$. An exact WB can be computed by applying an LP solver to the extensive formulation eq. (2.7). Its optimal value 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 let the solvers stop with a maximum number of iterations. table 2.2 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. In contrast with MAM, B-ADMM does not have (at this time) a

Table 2.2: MAM vs B-ADMM. The considered implementation of B-ADMM 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 dataset is the one considered in [67, Section IV]. 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

convergence analysis.

2.6.4 Unbalanced Wasserstein Barycenter

This section treats a particular example to illustrate the interest of using UWB. The artificial dataset is composed by 50 images with resolution 80×80. Each image is divided in four squared. The top left, bottom left and bottom right squared are randomly filled with double nested ellipses and the top right squared is always empty as exemplified in section 2.6.4. In this example, every image is normalized to depict a probability measure so that we can compare WB and UWB.

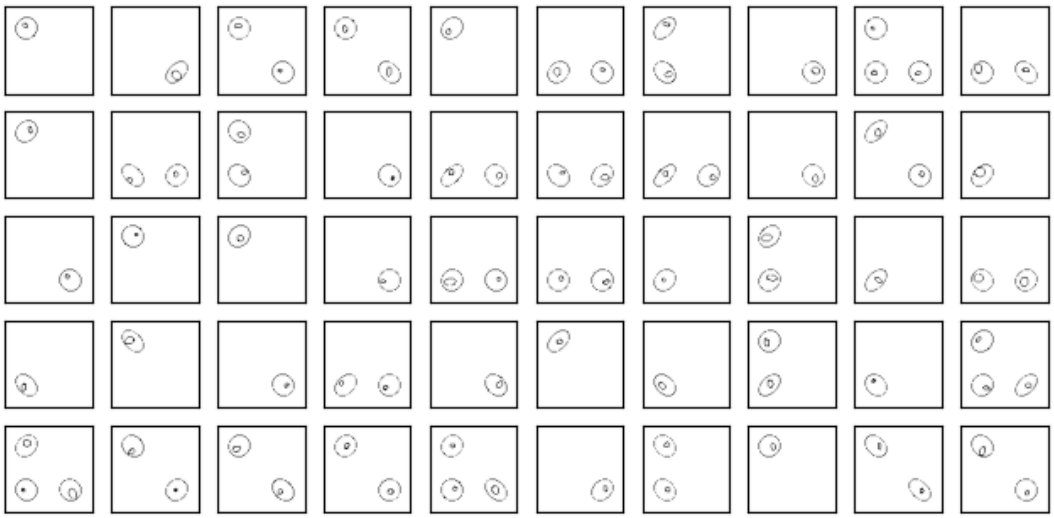

Figure 2.7: Dataset composed by 50 pictures with nested ellipses randomly positionned in the top left, bottom right and left corners.

With respect to eq. (2.10), one set of constraints is relaxed and the influence of the hyperparameter γ is studied. If γ is large enough (i.e. greater than $\|\texttt{vec}(d)\| \approx 1000$, 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 $\mathcal{B}$ that is more and more relaxed. Therefore, as illustrated by section 2.6.4, the resulting UWB splits the image in four parts, giving visual meaning to the barycenter.

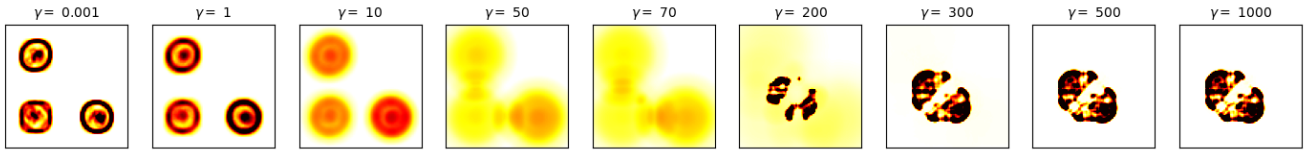

Figure 2.8: UWB computed with MAM for different values of γ.

In the same vein, section 2.6.4 provides an illustrative application of MAM for computing UWB in another dataset.

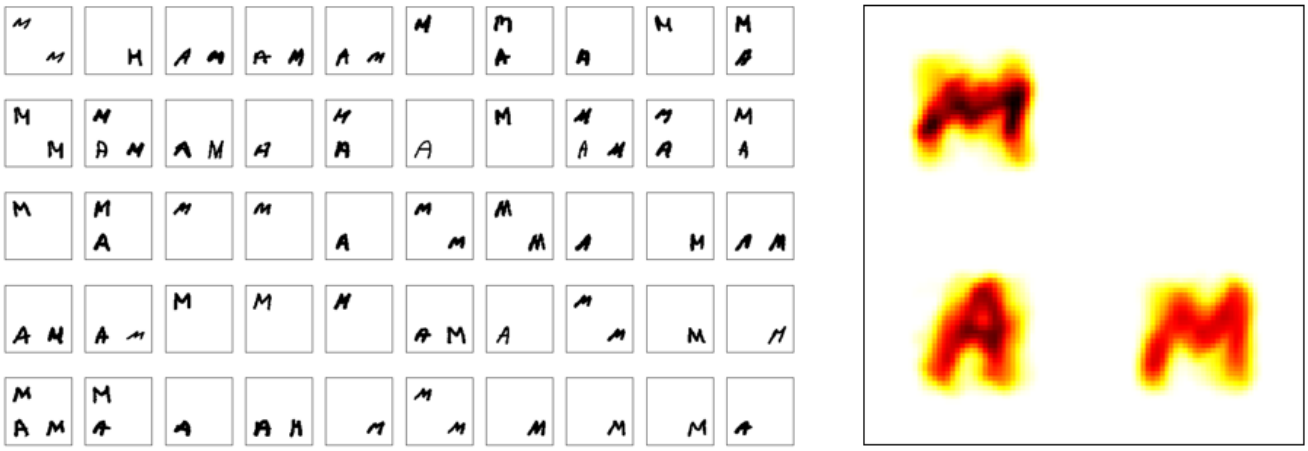

Figure 2.9: *(left)* UWB for a dataset of letters M-A-M built in the same logic than section 2.6.4 with 50 figures: *(right)* resulting UWB with $\gamma = 0.01$, computed in 200 seconds using 10 processors.

Chapter 3

The scenario Tree Reduction Problem

3.1 The Kovacevic and Pichler's approach (KP)

In [51], the author introduced the Nested Distance (ND), which is built on the Wasserstein distance, exploiting its structure and extending it to accommodate the specific characteristics of stochastic processes (see also [52]). It is a tailor-made measure for stochastic processes: it inherits the foundational properties of the Wasserstein distance, such as its sensitivity to local structure and robustness to outliers, while providing a more nuanced and specialized measure for comparing the similarities between different realizations of stochastic processes.

3.1.1 Nested distance

Let $\mathbf{P} := (\Xi, (\mathcal{F}), P)$ and $\mathbf{P}' := (Z, (\mathcal{F}'), P')$ be two filtered probability spaces with finite moment of order ι with respect to a distance $d : \Xi \times \Xi' \to \mathbb{R}$, such that $\Xi := \{\xi^{(1)}, \dots, \xi^{(R)}\}$ and $Z := \{\zeta^{(1)}, \dots, \zeta^{(S)}\}$ are the quantizer values.

Definition 7 (Discrete Nested Distance)**.** *For $\iota \in [1, \infty)$, the process distance of order ι, between $\mathbf{P}$ and $\mathbf{P}'$ is the ι^{th} root of the optimal value of the following LP, known as the* discrete Nested Distance *(ND):*

$$\mathtt{dl}_\iota(\mathbf{P}, \mathbf{P}')^\iota := \begin{cases} \min\limits_{\pi} & \sum\limits_{\xi \in \Xi} \sum\limits_{\zeta \in Z} \mathtt{d}(\xi, \zeta)^\iota \pi(\xi, \zeta) \\ s.t. & \pi(M \times Z | \Xi_t \otimes Z_t) = P(M|\mathcal{F}_t), \quad (M \in \mathcal{F}_T, t = 0, \dots, T) \\ & \pi(\Xi \times N | \Xi_t \otimes Z_t) = P'(N|\mathcal{F}'_t), \quad (N \in \mathcal{F}'_T, t = 0, \dots, T) \\ & \pi \geq 0. \end{cases} \tag{ND}$$

In this problem the minimum is among all bivariate probability measures $\pi \in \mathcal{P}(\Xi, Z)$.
Note that eq. (ND) is a generalization of the Wasserstein distance (see Definition 9). Indeed, the transport plan π does not only respect the marginals imposed by P and P', but also respect the conditional marginals. This constraint is responsible for the filtration of the trees being taken into account in this distance.

3.1.2 Scenario trees and notations

A T-period scenario tree ($\mathcal{N}$, A) is a discrete form of a random process $\xi_0(\omega), \dots, \xi_T(\omega)$ where $\xi_t(\omega) \in \Xi \subset \mathbb{R}^d, t = 0, \dots, T$. For the scenario tree there is only one root node associated with the value $\xi_0(\omega)$, for all $\omega \in \Omega$. It is

accepted to call the vertices $\mathcal{N}$, nodes [53]. In fact, the notion of atoms, seen in eq. (1.12), corresponds to the *nodes* in what follows. To one node n, one can biunivoquely associate a stage t and an element of the t-partition $\tilde{\mathcal{A}}_t$: $\forall n \in \mathcal{N}, \exists!(t, A) : A \in \tilde{\mathcal{A}}_t$. For clearer understanding, we introduce the following notations:

$$S(n) := t \tag{3.1}$$

$$\tilde{A}(n) := A \in \tilde{\mathcal{A}}_{S(n)} \tag{3.2}$$

Thus, we denote the value taken by the process $\xi \in \Xi$ at node n by $\xi_t(\omega) = \xi(n)$, reciproquely $\xi(n) = \xi_{S(n)}(\tilde{A}(n))$. There is no ambiguity in defining the value of the processus at a stage for the *set* $\tilde{A}(n)$ because all of its elements have the same value for $\xi_{S(n)}$ per definition. A node $m \in \mathcal{N}$ is a direct predecessor or parent of the node $n \in \mathcal{N}$, if $(m, n) \in A$. This relation is embodied by the notation $m = n-$. Reciprocally, n is a direct successor (or children) of m and this set is denoted $m+$, such that $n \in m+$ if and only if $m = n-$. In the same vein, we denote the predecessors (or ancestors) of $n \in \mathcal{N}$ as $\mathcal{A}(n)$: if $m_1 = m_2-$ and $m_2 = n-$ (equivalently $n \in m_2+$ and $m_2 \in m_1+$), then $m_1, m_2 \in \mathcal{A}(n)$. We consider only trees with a single root, denoted by 0, i.e., $0- = \emptyset$, but the methods developped here can be generalized for multi root trees (forests). Nodes $n \in \mathcal{N}$ without successor nodes (i.e., $n+ = \emptyset$) are called leaf nodes. For every leaf node n there is a sequence $(n_0, \dots, n_t, \dots, n)$ where $n_0 \in \mathcal{A}(n_1)$, $n_1 \in \mathcal{A}(n_2)$ etc, from the root to the leaf node n (note that $n := n_T$) composed by T nodes. Each of these sequence corresponds a unique event $\omega \in \Omega$.
We denote the probabilities, assigned to node n, by $P(n)$ and we denote conditional probabilities between successors by $P(n|m) = P(n)/P(m)$ for $m = n-$. Furthermore, using a distance $\mathtt{d}$, we denote the distance between two processes as:

$$\mathtt{d}_{n_1 n_2} := \mathtt{d}(\xi_{[S(n_1)]}(\tilde{A}(n_1)), \xi_{[S(n_2)]}(\tilde{A}(n_2)) \tag{3.3}$$

If the problem is considered from another point of view: Let's be a scenario tree, composed by R scenarios $\omega_r := (n_{r_0}, ..., n_{r_t}, ...n_{r_T})$, for $r = 1, \dots, R$. Each scenario is a stochastic process in finite time, with a probability $P(\omega_r) = P(n_{r_T})$. Note that we denoted as *sequence of T nodes* this scenario, where the path is $(n_{r_0}, n_{r_1}, \dots, n_{r_T})$ with $n_{r_0} \in \mathcal{A}(n_{r_1})$, $n_{r_1} \in \mathcal{A}(n_{r_2})$ ... and $n_{r_{T-1}} \in \mathcal{A}(n_{r_T})$. We coupled each node with a value and a probability. These *values* are $\xi(n_{r_t})$, the values of the stochastic process. Then, the whole scenario tree can be described with ξ: a random vector, with the realization $(\xi(n_{r_t}))_{t \in \{0,\dots,T\}}$ having a probability measure μ, such that $\mu : \omega_r \longmapsto P(\xi(n_{r_T}))$.
Knowing that a scenario tree can be described as a random vector ξ having a probability measure μ, it can be compared to another scenario tree with a distance function defined between probability distributions such as the Wasserstein distance. In fact, the Wasserstein distance has been widely employed to compare trees and is the base of a lot of reduction tree methods. But it can be intuitively shown that it is not suitable to distinguish stochastic processes with different flows of information (see 2.1 in [41]). When comparing trees, we do not want to only compare the scenarios but the scenarios and the filtration affiliated to them. While the Wasserstein distance is efficient to compare probability distributions, it is not fitted to take into account the evolution of the available information along the stages of stochastic processes [36, 41]. On the other hand, the Nested Distance is able to take explicitly into account this flow of increasing information being a generalization of the Wasserstein Distance.

3.1.3 Nested Distance for Trees

Using the tree notations introduced in Section 3.1.2, we can derive a discrete model for the ND between two trees $\mathbf{H} := (\Xi, (\mathcal{N}, A), P)$ and $\mathbf{G} := (Z, (\mathcal{N}', A'), P')$. The transport mass between node $i \in \mathcal{N}_t$ and node $j \in \mathcal{N}'_t$ at stage $t \in \{1, \dots, T\}$, is noted $\pi_{i,j}$ or $\pi(i, j)$. The probability measures for eq. (ND) can be given directly by $\pi_{i,j}$ at the leaves $i \in \mathcal{N}_T$ and $j \in \mathcal{N}'_T$. For earlier nodes $m \in \mathcal{N}_t$ and $n \in \mathcal{N}'_t$, it can be derived from the conditional

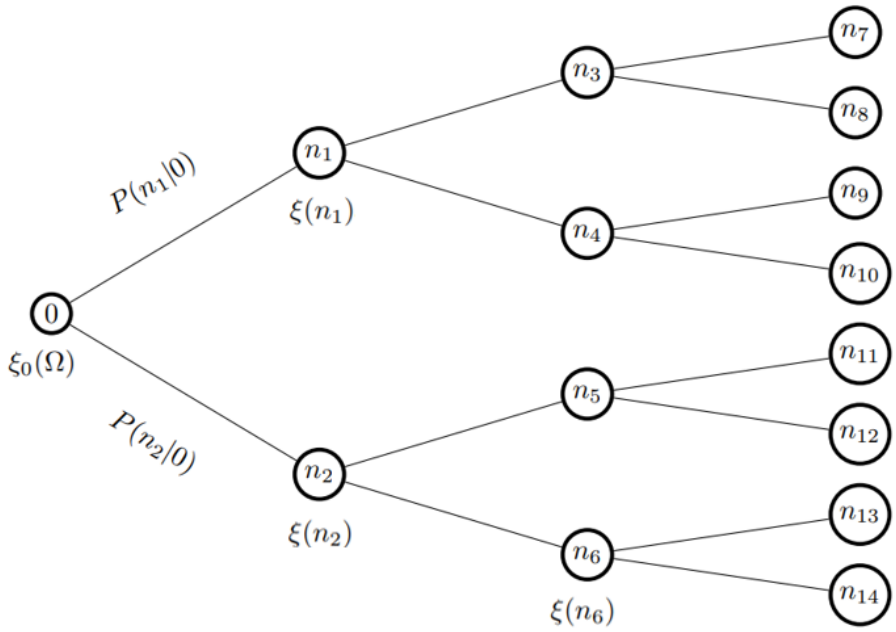

Figure 3.1: Scenario tree notations.

probabilities using $\pi(i,j|m,n) = \frac{\pi_{i,j}}{\pi_{m,n}}$. The problem from eq. (ND) to compute the Nested Distance reformulated for trees, thus reads as:

Definition 8 (Nested Distance for Trees)**.** *For $\iota \in [1,\infty)$, the process distance of order ι, between* $\mathbf{H}$ *and* $\mathbf{G}$ *is the ι^{th} root of the optimal value of the following LP,*

$$\mathtt{dl}_\iota(\mathbf{H},\mathbf{G})^\iota := \begin{cases} \min\limits_{\pi} & \sum\limits_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \pi_{i,j}\mathtt{d}^\iota_{i,j} \\ s.t. & \sum_{\{j:n\in\mathcal{A}(j)\}} \pi(i,j|m,n) = P(i|m), \quad (m\in\mathcal{A}(i),n) \\ & \sum_{\{i:m\in\mathcal{A}(i)\}} \pi(i,j|m,n) = P'(j|n), \quad (n\in\mathcal{A}(j),m) \\ & \pi_{i,j} \geq 0 \text{ and } \sum_{i,j}\pi_{i,j} = 1. \end{cases} \tag{NDT}$$

This linear programming problem (LP) can be rather large and challenging to compute, this is why Kovacevic and Pichler [41] introduced an efficient recursive method. Then [54] developped an even faster algorithm based on the Sinkhorn distance. In the litterature, this distance is used and conveniently computed with the recursive exact LP resolution or the Sinkhorn alternative algorithm.

3.1.4 The KP algorithm for scenario tree reduction

The scenario tree reduction mechanism resides in solving the problem eq. (NDT), between the original and smaller trees. The latter has known filtration (same number of stages but considerably fewer number of nodes than the original tree), initialized probabilities, and quantizer values.

Even though, the distance eq. (NDT) to measure the closeness between scenario trees is broadly adopted, no effective method to reduce general scenario trees based directly on this distance has been proposed since Kovacevic and Pichler [41] (2015), because this approach has been judged too computationally demanding. An exception is [6], which focuses on the particular class of stage-wise independent scenario trees. To deal with the non-convex nature of the problem, due to the optimization of both quantizers and probabilities, the method in [41] operates a classical block coordinate optimization scheme. As illustrated in Figure 3.2, after a filtration is chosen, the first step is the optimization of the probabilities P' for fixed quantizers and the second step the optimization of the quantizer values $\xi \in \Xi$ for fixed probability. The latter step is developed in [41], it is straightforward and analytically exact in the euclidean case $\iota = 2$. The probability optimization is demanding though, because it requires solving multiple (potentially large-scale) LPs. In the remainder of this work we will only develop the probability optimization and will only recall briefly the quantizer optimization in Algorithm 5.

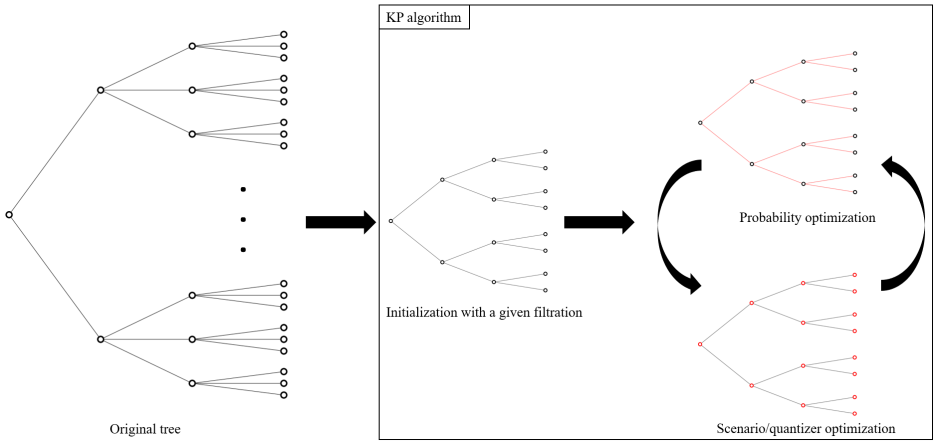

Figure 3.2: KP algorithm: to approximate a tree, a smaller tree with a given filtration is improved in order to minimize the nested distance with the original tree. The probabilities and the quantizers are alternatively optimized until convergence.

The multistage problem of optimal probabilities can be stated as follows: given the stochastic process quantizers ($\zeta \in Z$) and filtration $\mathcal{F}' := (\mathcal{N}', A')$, which probability measure P' is optimal to approximate $\mathbf{H} :=(\Xi, (\mathcal{N}, A), P)$ regarding the nested distance? Inspecting formulation eq. (NDT), it turns out that P' can be computed by jointly

optimizing with respect to $\pi_{i,j}$ and $P'(j|j-)$. This leads to the following large non-convex optimization problem:

$$\begin{cases} \min\limits_{\pi,P'} & \sum\limits_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \pi_{i,j}\mathtt{d}^\iota_{i,j} \\ \text{s.t.} & \sum_{j\in n+}\pi(i,j|m,n) = P(i|m), \quad (m\in\mathcal{A}(i),n) \\ & \sum_{i\in m+}\pi(i,j|m,n) = P'(j|n), \quad (n\in\mathcal{A}(j),m) \\ & \pi_{i,j}\geq 0 \text{ and } \sum_{i,j}\pi_{i,j} = 1 \\ & P'(j|j-)\geq 0. \end{cases} \tag{3.4}$$

This is a bilinear problem, hence difficult to handle: there is a large number of decision variables and bilinear constraints (issued by the conditional probabilities in the second group of constraints, which involve the decision variables composing π and P'). To get around the mathematical difficulties of eq. (3.4), the authors of [41] propose first to write the problem in a recursive way so that bilinear terms are moved to the objective function, and then approximate the objective. Indeed, using the conditional probabilities $\pi_{i,j} = \pi(i,j|m,n)\times\pi_{m,n}$, we can derive the recursive formula:

$$\sum_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T}\pi(i,j)\mathtt{d}^\iota_{i,j} = \sum_{n\in\mathcal{N}'_{T-1}}\sum_{m\in\mathcal{N}_{T-1}}\pi(m,n)\sum_{i\in m+,j\in n+}\pi(i,j|m,n)d^\iota_{i,j} \tag{3.5a}$$

$$= \sum_{\tilde{n}\in\mathcal{N}'_{T-1}}\sum_{\tilde{m}\in\mathcal{N}_{T-1}}[\pi(\tilde{m},\tilde{n})\times\pi(m,n|\tilde{m},\tilde{n})]\times dl_\iota(m,n)^\iota \tag{3.5b}$$

$$= \sum_{\tilde{n}\in\mathcal{N}'_{T-2}}\sum_{\tilde{m}\in\mathcal{N}_{T-2}}\pi(\tilde{m},\tilde{n})\sum_{i\in\tilde{m}+,j\in\tilde{n}+}\pi(m,n|\tilde{m},\tilde{n})dl_\iota(m,n)^\iota \tag{3.5c}$$

Note that for the leaves i and j of the trees, $dl_\iota(i,j)^\iota = d_\iota(\xi_i,\zeta_j)^\iota = \mathtt{d}^\iota_{i,j}$ because the filtration in eq. (ND) are transparent at the leaves so the simple d^ι_i can be used. Thanks to this recursive formula, the problem can be split into recursive smaller problems for $m\in\mathcal{N}_t$ and $n\in\mathcal{N}'_t$: the conditional probability $\pi(.,.|m,n)$ is a solution to

$$\begin{cases} \min\limits_{\pi(.,.|m,n)} & \sum_{m\in\mathcal{N}_t}\pi(m,n)\sum_{i\in m+,j\in n+}\pi(i,j|m,n)dl_\iota(i,j)^\iota \\ \text{s.t.} & \sum_{j\in n+}\pi(i,j|m,n) = P(i|m), & (i\in m+) \\ & \sum_{i\in m+}\pi(i,j|m,n) = \sum_{i\in\tilde{m}+}\pi(i,j|\tilde{m},n), & (j\in n+ \text{ and } m,\tilde{m}\in\mathcal{N}_t) \\ & \pi(i,j|m,n)\geq 0. \end{cases} \tag{3.6}$$

The complication in this subproblem is the bilinear term in the objective. To get this difficulty, [41] proposes to fix $\pi(m,n)$ with the values computed from the previous iteration (or initialized at first iteration) giving rise to a LP approximation. The authors have empirically shown that after few iterations of their algorithm the values assigned to $\pi(m,n)$ converge (not necessarily to the optimal solution).

Despite this approximation, the problem is still challenging due to its huge dimensions. Thus, the method's main limitation is its computational burden that becomes prohibitive for large-scale scenario trees, as exemplified in Section 3.3. The reason is that the method requires solving potentially large-scale LPs as in (3.6) repeatedly. We address the challenge by noticing that (3.6) is nothing but a *Wassertein barycenter problem*, for which specialized and efficient algorithms exist.

3.2 Scenario tree reduction via Wasserstein Barycenters

In this section we show that subproblem (3.6) is a Wassertein barycenter problem.

Definition 9 (Discrete Wasserstein Distance)**.** *Let* $(\Omega, \mathtt{d})$ *be a metric space and* $P(\Omega)$ *the set of Borel probability measures on* Ω*. Furthermore, let* ξ *and* ζ *be two random vectors having probability measures* μ *and* ν *in* $P(\Omega)$*, that is,* $\xi \sim \mu$ *and* $\zeta \sim \nu$*. For* $\iota \in [1, \infty)$*, and probability measures* μ *and* ν *in* $P(\Omega)$*. Their* ι*-Wasserstein distance* W_ι *is :*

$$W_\iota(\mu, \nu) := \left(\min_{\pi \in U(\mu,\nu)} \sum_{\Omega \times \Omega} \mathtt{d}(\xi, \zeta)^\iota d\pi(\xi, \zeta) \right)^{1/\iota}, \tag{WD}$$

where $U(\mu, \nu)$ *is the set of all probability measures on* $\Omega \times \Omega$ *having marginals* μ *and* ν*. We denote by* $W_\iota^\iota(\mu, \nu)$*,* W_ι *to the power* ι *, i.e.* $W_\iota^\iota(\mu, \nu) := (W_\iota(\mu, \nu))^\iota$*.*

With, for $\tau \geq 0$ a given scalar, $\Delta_n(\tau) := \left\{ u \in \mathbb{R}^n_+ : \sum_{i=1}^n u_i = \tau \right\}$ is the set of non negative vectors in $\mathbb{R}^n$ adding up to τ. If $\tau = 1$, then $\Delta_n(\tau)$, denoted simply by Δ_n, is the $n+1$ simplex.

Definition 10 (Wasserstein Barycenter)**.** *Given* M *measures* $\{\nu^{(1)}, \ldots, \nu^{(M)}\}$ *in* $P(\Omega)$ *(see eq.* (WD)*) and* $\alpha \in \Delta_M$*, an* ι-Wasserstein barycenter *with weights* α *is a solution to the following optimization problem*

$$\min_{\mu \in P(\Omega)} \sum_{m=1}^{M} \alpha_m W_\iota^\iota(\mu, \nu^{(m)}) \,. \tag{3.7}$$

Now, to show that (3.6) fits into the structure of (3.7), let $\alpha_{m_i}^{(n)} := \pi(n, m_i)$ for $i = 1, \ldots, M$, $M = |\mathcal{N}_t|$, and $P(|n)$ an auxiliary vector of probabilities allowing to reformulate and couple the last group of constraints in (3.6). Given $t \in \{1, \ldots, T\}$ and $n \in \mathcal{N}'_t$, problem (3.6) reads as:

$$\begin{cases} \min\limits_{\pi(.,.|m,n), P'(|n)} & \alpha_1^{(n)} \sum\limits_{i \in m_1+, j \in n+} \pi(i,j|m_1,n) dl_\iota(i,j)^\iota \quad + \cdots + \quad \alpha_M^{(n)} \sum\limits_{i \in m_M+, j \in n+} \pi(i,j|m_M,n) dl_\iota(i,j)^\iota \\ \text{s.t.} & \sum_{j \in n+} \pi(i,j|m_1,n) & = P(i|m_1), & (i \in m_1+) \\ & \qquad \ddots & \vdots & \\ & \qquad \sum_{j \in n+} \pi(i,j|m_M,n) & = P(i|m_M), & (i \in m_M+) \\ & \sum_{i \in m_1+} \pi(i,j|m_1,n) & = P'(j|n), & (j \in n+) \\ & \qquad \ddots & \vdots & \\ & \qquad \sum_{i \in m_M+} \pi(i,j|m_M,n) & = P'(j|n), & (j \in n+) \\ & \sum_{j \in n+} P'(j|n) = 1, \pi(i,j|m_1,n) \geq 0 \quad \cdots & \pi(i,j|m_M,n) \geq 0, & \end{cases} \tag{3.8}$$

which is equivalent to

$$\min_{P'(|n)\geq 0} \sum_{m\in\mathcal{N}_t} \alpha_m^{(n)} W_\iota^t(P'(|n), P(|m)) \quad \text{s.t.} \quad \sum_{j\in n+} P'(j|n) = 1,$$

i.e., a Wasserstein Barycenter problem. This interpretation of (3.6) is crucial for rendering the scenario reduction algorithm of [41] practical for huge-scale settings. The reason is that the above LP (equivalent to (3.6)) can be efficiently handled by state-of-the-art methods such as the Iterative Bregman Projection method [9] or the Method of Averaged Marginals [46]. While the former is an inexact method, the latter can indeed asymptotically compute an exact barycenter, i.e., it can solve the large-scale LP (3.6) issued by the Nested Distance.

To further illustrate our interpretation of problem (3.6) as a Wassertein Barycenter problem, consider Figure 3.3. The boxed subtree of the approximated (smaller) tree forms a probability measure with support (n_7, n_8) and probabilities $(P(n_7|n_3), P(n_8|n_3))$ that form a barycenter of the (original tree's) subtrees with initial nodes m_3, m_4, m_5, m_6. Once the Wasserstein Barycenter $(P(n_7|n_3), P(n_8|n_3))$ is computed by solving (3.8), we pass to the next (approximated tree's) subtree, that is the one issued by node n_4. The probabilities $(P(n_9|n_4), P(n_{10}|n_4))$ are then computed by solving another barycenter problem, but with a different weights $\alpha_m^{(n_4)}$ (and costs) for $m \in \{m_3, m_4, m_5, m_6\}$ (see eq. (3.8)).

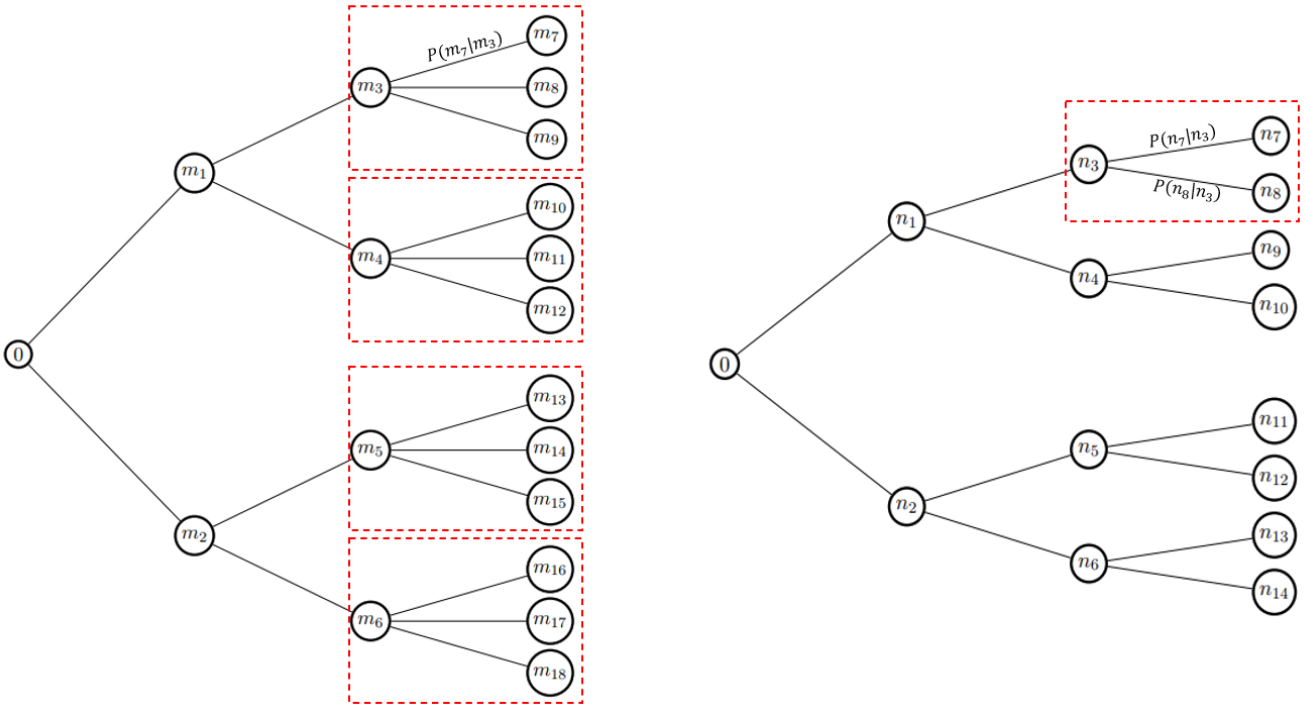

Figure 3.3: *(left)* Original Tree, *(right)* Approximated tree. The probabilities $(P(n_7|n_3), P(n_8|n_3))$ are computed as the Wasserstein Barycenter of the set of (known) probabilities associated to the boxed subtrees on the left.

It is thus clear that several Wasserstein Barycenter problems must be solved at every iteration of the scenario tree reduction algorithm from [41], The faster the computation of such barycenters, the faster the scenario tree reduction procedure.

3.2.1 Wasserstein barycenters techniques for scenario reduction

Problem from eq. (3.8) is an LP and could, in principle, be solved by LP solvers such as Gurobi, Cplex, scipy, HiGHs, and others. However, in real life applications of scenario tree reduction, problem from eq. (3.8) (equivalently problem from eq. (3.6)) has huge dimensions and intractable by LP solvers, hindering thus the whole scenario tree reduction process. Many methods can be employed to tackle eq. (3.8) [9,19–21,30,50,71,72]. In the remaining of the paper, we will use two different WB algorithms, namely MAM presented in details in Chapter 2 and the Iterative Bregmann Projection algorithm (IBP) [9] associated with the Sinkhorn algorithm [40, 64]. Let us now describe the latter method.

3.2.1.1 Computation of Transport Plans via regularized techniques

The following paragraph explains how leveraging efficient regularized method enables to compute the transport plans that the scenario tree reduction approach needs. First IBP is utilized to compute the barycenter of the subtrees (see fig. 3.3), then the Sinkhorn's algorithm is used to compute the transport plans between the computed barycenter and each subtree. Note that, the *Sinkhorn algorithm* and the *Iterative Bregmann Projection* (IBP) both rely on a regularization of the optimization problems presented in eqs. (3.8) and (WD). Next, we use matrix notation to make the presentation lighter. Thus, (WD) reads as:

$$\mathtt{OT}(p,q) := \min_{\pi \in U(p,q)} \langle D, \pi \rangle. \tag{3.9}$$

Recall that the ι^{th} root of $\mathtt{OT}$(p,q) is $W_\iota(\mu,\nu)$ and $p \in \Delta_R$, and $q \in \Delta_S$ are the masses of μ and ν. Cost D is composed by the distance values (d_{rs}), and π is the transport matrix between the two probability densities. The regularization leveraged by the following algorithms is the Kullback-Leibler divergence between $\pi, Q \in \mathbb{R}^{R\times S}$ and $Q_{r,s} > 0$ for all (r,s), defined as:

$$KL(\pi|Q) := \sum_{r,s} \pi_{r,s}(\log(\frac{\pi_{r,s}}{Q_{r,s}}) - 1) \tag{3.10}$$

with the convention $0\log(0) = 0$. Function KL is a Bregmann divergence, it is strongly convex, and assures that $KL(\pi|Q) \geq 0$ and $KL(\pi|Q) = 0$ if and only if $\pi = Q$. This function is employed to regularize the LP (3.9), leading to the following nonlinear optimization problem:

$$\mathtt{OT}^\epsilon(p,q) := \min_{\pi \in U(p,q)} \langle D, \pi \rangle + \epsilon KL(\pi|p \otimes q), \tag{3.11}$$

with $p \otimes q = (p_r q_s)_{r,s}$ and $\epsilon > 0$ a given parameter. Since problem (3.11) is ϵ-strongly convex, it has a unique optimal solution.

The Iterative Bregman Projections - IBP. Given input histogram $q^{(m)} \in \Delta_S^{(m)}$ for $m = 1, \ldots, M$, a Wasserstein Barycenter with weights $\alpha \in \Delta_M$ is a solution of

$$\min_{p \in \Delta_R} \sum_{m=1}^{M} \alpha_m \mathtt{OT}(p, q^{(m)}) \tag{3.12}$$

Then, applying the entropic regularization to (3.11) with $\epsilon > 0$, one can approximate a barycenter by the solution to the following problem

$$\min_{p \in \Delta_R} \sum_{m=1}^{M} \alpha_m \mathtt{OT}^\epsilon(p, q^{(m)}). \tag{3.13}$$

This is a smooth strongly convex minimization problem, which can be tackled using gradient descent [19]. Another more interesting way is to use the work of [9] and treat eq. (3.13) with an iterative method just like the Sinkhorn algorithm. [1] The algorithm is detailed in Algorithm 3.

Algorithm 3 does not return the transport plans between the computed barycenter p and the histograms $q^{(m)}, m = 1, \dots, M$. But the KP algorithm exploits the transport plans only (see eq. (3.6)). Therefore it is necessary to first compute the barycenter p using the *IBP algorithm* and then find one by one the transport plans between p and each subtrees using the *Sinkhorn's algorithm*.

Entropic Regularization for discrete measures Introducing, to problem eq. (3.11), dual variables $f \in \mathbb{R}^R$ and $g \in \mathbb{R}^S$ for each marginal constraints, the Lagrangian of (3.11) is given by

$$L(\pi, f, g) = \langle D, \pi \rangle + \epsilon KL(\pi | p \otimes q) + \langle f, p - \pi \mathbb{1}_R \rangle + \langle g, q - \pi^T \mathbb{1}_S \rangle \tag{3.17}$$

The derivative of L with respect to $\pi_{r,s}$ gives

$$\frac{\partial L}{\partial \pi_{r,s}} = D_{r,s} + \epsilon \log(\frac{\pi_{r,s}}{p_r q_s}) - f_r - g_s = 0, \tag{3.18}$$

Thus

$$\pi_{r,s} = p_r q_s e^{\frac{f_r + g_s - D_{r,s}}{\epsilon}} := u_r K_{r,s} v_s, \tag{3.19}$$

with $u := (p_r e^{f_r/\epsilon})_r$, $K := (e^{-D_{r,s}/\epsilon})_{r,s}$ and $v := (q_s e^{g_s/\epsilon})_s$.
The latter equation can be rewritten in matrix form as $\pi = \mathrm{diag}(u) K \mathrm{diag}(v)$. Note that additionally the marginal

[1] Another more interesting way is to remark that if we define the penalization $E(\pi) := -\sum_{r,s} \pi_{r,s}(\log(\pi_{r,s}) - 1)$ instead of the Kullback-Leibler divergence, then:

$$\mathtt{OT}^\epsilon(p, q) := \langle D, \pi \rangle + \epsilon E(\pi) \tag{3.14a}$$

$$= \epsilon \sum_{r,s} (\frac{D_{r,s}}{\epsilon} \pi_{r,s} + \pi_{r,s} \log(\pi_{r,s}) - \pi_{r,s}) \tag{3.14b}$$

$$= KL(\pi | K) \tag{3.14c}$$

Then by noticing that $p = \pi \mathbb{1}_R$, eq. (3.12) can be rewritten:

$$\min_{\substack{\pi^{(m)} \in U(p, q^{(m)}), \\ m = 1, \dots, M}} \sum_{m=1}^{M} \alpha_m KL(\pi^{(m)} | K^{(m)}) \tag{3.15}$$

Then, following [9], the optimal coupling $\pi^{(m)}, m = 1, \dots, M$ can be derived after iterative KL projections onto the right and left constraints, embodied by the sets $U(p, q^{(m)}), m = 1, \dots, M$, following the same pattern as the entropic regularization of the Sinkhorn's method.

$$\pi = \mathrm{diag}(u) K \mathrm{diag}(v) \tag{3.16a}$$

$$v^{(m),k+1} = \frac{q^{(m)}}{(K^{(m)})^T u^{(m),k}} \text{ and } u^{(m),k+1} = \frac{p^{k+1}}{K^{(m)} v^{(m),k+1}} \tag{3.16b}$$

Where p^{k+1} is the current estimate of the barycenter ($p^{k+1} = \prod_{m=1}^{M} (K^{(m)} v^{(m),k+1})^{\alpha_m}$). This result comes from the derivation of the Lagrangian and the proof can be found in [9]. This algorithm can be computed in parallel and the transport plan can be directly extracted and used in the scenario tree reduction algorithm.
In the following of this work, only Algorithm 3 and the following remarks are used and presented, but in futur research this method will be used instead.

Algorithm 3 IBP ALGORITHM

▷ Step 0: input

Given $\epsilon > 0$, for all $m \in \{1, \dots, M\}$, compute the distance matrices $D^{(m)}$ and initialize $u^{(m),0}$ with an arbitrary positive vector, for example $\mathbb{1}_S$
Define $K = e^{-D/\epsilon}$

▷ Step 1: Iterate

while not converged **do**
  for m=1,...,M **do**
    $v^{(m),k+1} = \frac{q^{(m)}}{(K^{(m)})^T u^{(m),k}}$
  end for
  $p^{k+1} = \prod_{m=1}^{M} (K^{(m)} v^{(m),k+1})^{\alpha_m}$
  for m=1,...,M **do**
    $u^{(m),k+1} = \frac{p^{k+1}}{K^{(m)} v^{(m),k+1}}$
  end for
end while

return $p := p^{k+1}$

constraints give: $\text{diag}(u)K\text{diag}(v)\mathbb{1}_R = p$ and $\text{diag}(v)K^T\text{diag}(u)\mathbb{1}_S = q$. A standard way to solve these equations is to use an iterative method: first modify u to satisfy the left constraints then modify v. These two updates define the Sinkhorn algorithm:

Algorithm 4 SINKHORN'S ALGORITHM

▷ Step 0: input

Given $\epsilon > 0$, initialize $v^{(0)}$ with an arbitrary positive vector, for example $\mathbb{1}_S$
Define $K = e^{-D/\epsilon}$

▷ Step 1: Iterate

while not converged **do**
  $u^{(k+1)} = \frac{p}{Kv^{(k)}}$
  $v^{(k+1)} = \frac{q}{K^T u^{(k+1)}}$
end while

return $\pi = \text{diag}(u)K\text{diag}(v)$

Observe that all the algorithm's steps consist of matrix-vector multiplication, and is thus simple to execute. The algorithm's drawback is its accuracy, which strongly depends on $\epsilon > 0$. The smaller ϵ the closer the solution of (3.11) to an exact solution of (3.9). However, ϵ is bounded because for too small ϵ the values of K diverge and the computation hurdle with double-precision overflow error.

3.2.2 Algorithm

We now rely on the previous subsections about Wasserstein barycenters to provide the following improved variant of the scenario tree reduction algorithm of [41]. In Algorithm 5, given a multistage scenario tree represented by $\mathbf{P} = (\Xi, \mathcal{F}, P)$, with Ξ the support, $\mathcal{F}$ a filtration, and P a probability measure, the algorithm constructs a smaller

scenario tree $\mathbf{P}' = (\Xi', \mathcal{F}', P')$ by updating the support Ξ' (set of reduced scenarios) and probability P' iteratively. The filtration $\mathcal{F}'$ is not a variable but a data given to the algorithm. In other words, the number of scenarios and the structure of the reduced tree is an input data, and the algorithm seeks for Ξ' and P' that minimizes the nested distance between $\mathbf{P}$ and $\mathbf{P}'$.

Algorithm 5 Scenario tree reduction via nested distance and Wasserstein barycenters

▷ Step 0: input

Let the original T-stage scenario tree $\mathbf{P} = (\Xi, \mathcal{F}, P)$ be given together with an (initial) smaller T-stage tree $\mathbf{P}^{'0} = (\Xi^{'0}, \mathcal{F}', P^{'0})$. Compute the transport probabilities $\pi^0(i,j)$ between scenarios $\xi^i \in \Xi$ and $\xi^{'j} \in \Xi^{'0}$

Set $k \leftarrow 0$ and choose a tolerance $\delta > 0$

for $k = 1, 2, \ldots$ **do**

▷ **Step 1:** improve the scenario values (quantizers)

  Set $\xi^{'k+1}(n_t) = \sum_{m \in \mathcal{N}_t} \frac{\pi^k(m,n_t)}{\sum_{i \in \mathcal{N}_t} \pi^k(i,n_t)} \xi_t(m)$ for all $t = 1, \ldots, T$ and nodes n_t of the smaller tree

  Let $\Xi^{'k+1}$ be the set of such new scenarios

▷ **Step 2:** improve the probabilities

  Set $dl^{k+1}(i,j) \leftarrow \|\xi^i - \xi^{'j}\|^2$ for all $\xi^i \in \Xi$ and $\xi^{'j} \in \Xi^{'k+1}$

  for $t = T-1, \ldots, 0$ **do** ▷ Recursivity

    Set $\alpha_m^{(n)} \leftarrow \pi^k(m,n)$, $m \in \mathcal{N}_t, n \in \mathcal{N}_t'$

    for all $n \in \mathcal{N}_t'$ **do** ▷ Wasserstein barycenters

      Use IBP followed by Sinkhorn, or MAM to compute $\pi^{k+1}(\cdot, \cdot|\cdot, n)$ solving (3.8)

    end for

    Set $dl^{k+1}(m,n) \leftarrow \sum_{i \in m+, j \in n+} \pi^{k+1}(i,j|m,n) dl^{k+1}(i,j)$, $m \in \mathcal{N}_t, n \in \mathcal{N}_t'$

  end for

▷ Unconditional transport plan matrix

  Set $\pi^{k+1}(0,0) \leftarrow 1$

  for t=1,...,T **do**

    Compute $\pi^{k+1}(m,n) = \frac{\pi^{k+1}(i,j|m,n)}{\pi^{k+1}(i,j)}$, $m \in \mathcal{N}_t, n \in \mathcal{N}_t'$ for any $i \in m-, j \in n-$

  end for

▷ **Step 3:** Stopping test

  if $dl^k(0,0) - dl^{k+1}(0,0) \leq \delta$ **then**

    Define $P_j' = \sum_{i \in \mathcal{N}_T} \pi^{k+1}(i,j)$ for all $j \in \mathcal{N}_T'$ and set $d(\mathbf{P}, \mathbf{P}') \leftarrow dl^{k+1}(0,0)$

    Stop and return with the reduced tree $\mathbf{P}^{'} = (\Xi^{'}, \mathcal{F}', P^{'})$ and nested distance $d(\mathbf{P}, \mathbf{P}')$

  end if

end for

3.2.2.1 Initialization

Observe that $P^{'0}$ is useful to determine the initial transport plan π^0: the algorithm can use the Sinkhorn algorithm or MAM for this task.

3.2.2.2 Hyperparameters

IBP and Sinkhorn's algorithms relies on the use of a parameter $\epsilon > 0$ chosen by the user [9, 20]. Such parameter has an impact on the result's accuracy. The MAM algorithm also requires setting a parameter, but it only impacts the convergence speed and can be determined with a sensitivity analysis [46].

3.2.2.3 Stopping criteria

The given stopping criteria test for is a heuristic: the algorithm terminates when the progress concerning the nested distance between the two trees is below a certain level of tolerance δ.

3.2.2.4 Storage complexity

The distance matrix dl and the transport matrix π whose size are equal to the number of nodes of the initial tree times the number of nodes in the approximate tree can be quite large to store but using adequate libraries, such as *numpy* in *python*, allow for fast computation of the linear operations that are used in the method.

3.2.2.5 Convergence

The algorithm leads to an improvement in each iteration step and converges in finitely many steps [41] provided $\delta > 0$. It should be kept in mind that the above algorithm is nothing but a heuristic, as it is a block-coordinate algorithm seeking to minimize the nested distance by alternating minimization over scenarios and then with respect to probabilities. As already discussed, the latter is bilinear optimization problem that is approximated by a recursion of LPs.

3.3 Applications

This section illustrates the performance of algorithm 5 and its behaviour by employing different solvers to compute Wasserstein barycenters. If a standard LP solver is employed, then algorithm 5 boils down to the setting of [41]. We ensure that the considered solvers attain the same level of precision when tackling the Wasserstein barycenter problems at Step 2 of Algorithm 5.

In what follows, we consider scenarios trees composed by T =4 to 8 stages and cpn =5 to 6 children per nodes, offering numbers of scenarios and nodes spanning from hundreds to ten of thousands. In our preliminary tests, the reduced tree is always a binary tree having the same number of stages than the original one.

Numerical experiments were conducted using 4 cores (*Intel(R) Xeon(R) Gold 5120 CPU*) and *Python 3.9*.

Algorithm 5 is implemented in *python* using a *MPI* based parallelization when possible (only MAM can be run in parallel). We consider the following variants of the algorithm, by changing the solver at Step 2:

- *LP*: Algorithm 5 employing the LP solver HiGHS for solving the Wasserstein Barycenter problem from eq. (3.8);
- *MAM*: Algorithm 5 employing the *Method of Averaged Marginals* [46] to solve eq. (3.8) exactly using the algorithm presented in **??**;
- *IBP*: Algorithm 5 employing IBP *Sinkhorn's algorithm*, inspired from the code of G. Peyré [49]. (These algorithms rely on the tune of hyperparameters that have been preset to guarantee their best efficiency and convergence.)

All variants use $\delta = 0.1$ as a tolerance for the stopping test. It has been empirically verified that results do not improve significantly if we decrease further this tolerance.

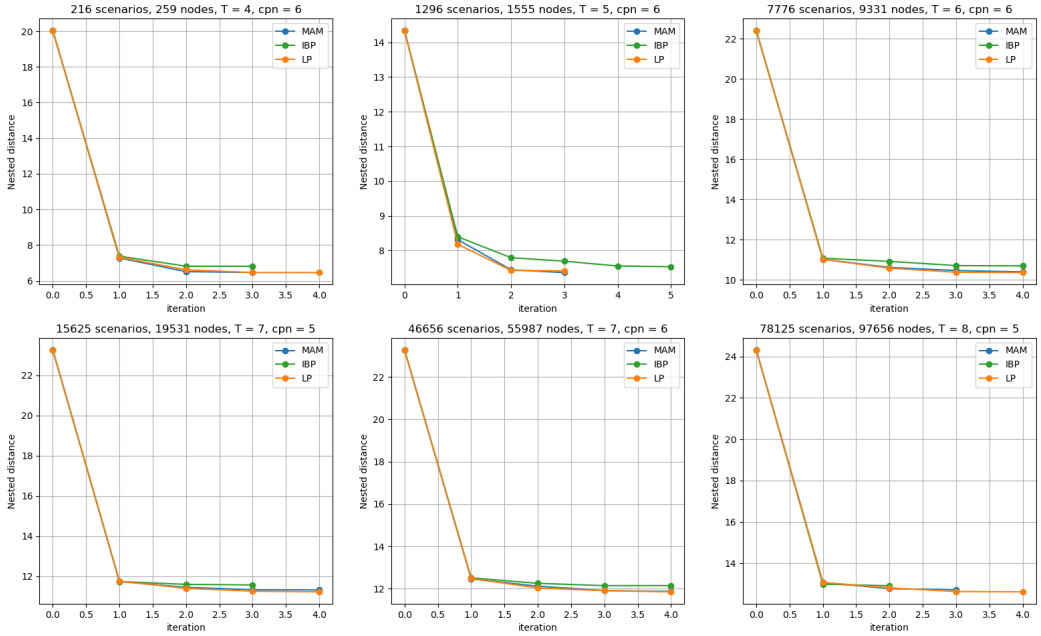

Figure 3.4: Evolution of the Nested Distance along the reduction iterations for different initial trees.

Figure 3.4 shows that the exact nested distance between the original tree and the approximate one iteratively decreases, not matter the variant of algorithm 5. All variants are internalized with the same initial tree, obtained randomly (probabilities and scenarios). Note that the initial ND is always at least halved after the reduction. This emphasizes how important is the use of a reduction method. Within this scale, one can see that every variant converges to approximately the same precision, although not necessarily to the same reduced scenario tree (because different solvers compute different optimal transportation plans, impacting the construction of scenarios composing the reduced tree).

Even though the ND decreases with all variants, such decreases seam faster when using LP or MAM. Also, the IBP algorithm tends to reach a plateau where it cannot improve the approximate tree after few iterations. This is due to the core of the IBP and Sinkhorn methods, which are inexact algorithms. MAM being an exact algorithm for solving eq. (3.8), it naturally follows the lead of LP. The slight differences between the variants LP and MAM can be explained by the fact that the general problem is non-convex, and different optimal transportation plans computed by different solvers can lead to different optimization paths that result in different reduced scenario trees. Therefore, depending on the employed solvers and the initialization, the approximated trees could be different while still having close ND with the original tree. Note that for $T = 5$, cpn$= 6$ the variant MAM yields a better approximation than the LP.

Table 3.1 shows that for small initial trees, up to 7776 scenarios, the LP variant is very effective: it provides the lowest ND solution in the shortest time. But from 15625 scenarios and more, IBP is faster. As the number of scenarios increases, the relative performances of MAM get better and, in our case with more than 46656 scenarios, MAM is eventually two times faster than LP while reaching the same precision as depicted in Figure 3.4. As

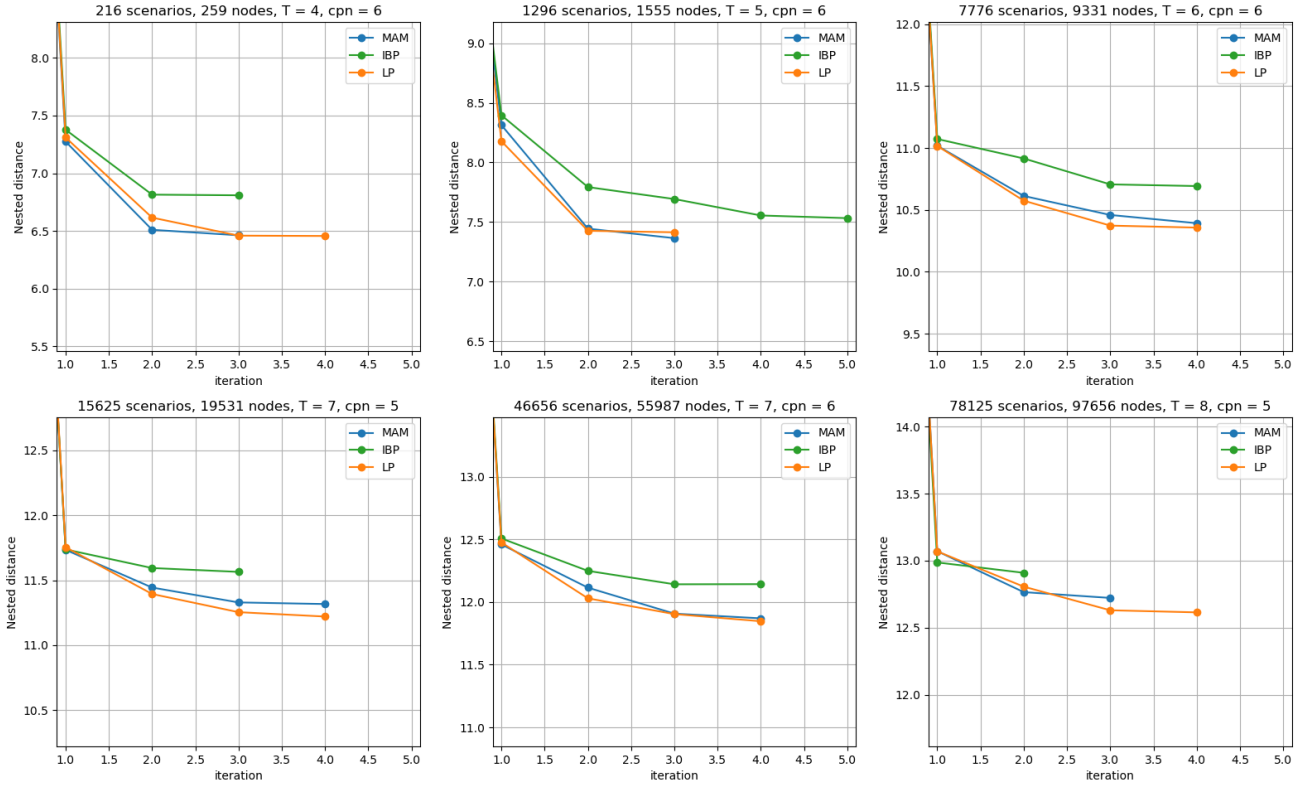

Figure 3.5: Evolution of the Nested Distance along the reduction iterations for different initial trees with a zoom.

shown in the last graph, with even more nodes and scenarios (78125 and 97656, respectively), MAM is the fastest variant. Note that IBP is a very robust method: not only is the precision reached more than reasonable, according to Figure 3.4, but the total time of execution is always in the ballpark of the fastest execution time of all algorithms. Leveraging that MAM is embarrassingly parallelizable, we ran results using 4 processors. We witnessed that in this configuration, the variant MAM is the most advantageous one when the initial tree has more than 10000 scenarios.

	LP	IBP	MAM	MAM 4 processors
T=4, cpn=6	0.17	0.49	2.21	0.56
T=5, cpn=6	1.54	14.83	18.23	6.28
T=6, cpn=6	74.25	161.19	344.83	124.44
T=7, cpn=5	487.58	323.76	816.46	341.62
T=7, cpn=6	4905	2136	2541	1256
T=8, cpn=5	13797	4334	3458	1635

Table 3.1: Total time (in seconds) per method for the studied trees.

Chapter 4

Concluding remarks and future work

This ongoing thesis concerns scenario tree reduction in multistage stochastic programming, a branch of mathematical optimization dedicated to decision-making under uncertainty along several time steps. Multistage stochastic programming has many applications and is particularly valuable in the energy industry. Multistage stochastic problems are recognized as challenging due to their astronomical dimensions, which are issued by the interplay of decision and random vectors along time steps. Finding ways to reduce the problem's dimensions while keeping approximately the same solution set and optimal value is essential in real-life applications of stochastic programming. This is why this thesis focuses on methodologies for reducing multistage scenario trees.

Central to our contribution is the development of a computational framework that integrates cutting-edge methods to enhance barycenter computation, thus facilitating the scenario tree reduction process. Recognizing the significance of accurate barycenter computation, we devised a new method tailored to address the Wasserstein Barycenter Problem. Our objective was to provide an exact solution capable of competing with existing methods in terms of efficiency and effectiveness. This new methodology leverages the Douglas-Rachford Operator splitting algorithm, enabling precise computation of barycenter probability densities through the calculation of transport plan marginals, which we term the Method of Averaged Marginals (MAM). This technique represents a significant advancement in the field, offering a more refined approach to scenario tree reduction. Furthermore, we have extended the applicability of MAM to tackle unbalanced barycenter problems, thereby addressing a broader range of applications. This extended version of MAM holds promise for enhancing the versatility and practicality of our approach, opening avenues for further exploration and refinement in scenario tree reduction methodologies. The manuscript detailing the MAM algorithm has been submitted to the *SIAM Journal on Mathematics and Data Sciences* (SIMODS) [1]. (We have received positive feedback by the journal's Referees and Editor while finalizing this report - they are advising us for further improvements on the manuscript). Additionally, a manuscript concerning our innovative approach to scenario tree reduction will be submitted to a specialized journal in the coming weeks.

In summary, this ongoing thesis has so far introduced a novel methodology for computing Wasserstein barycenters, identified that the large-scale subproblems in the scenario tree reduction algorithm of Kovacevic and Pichler [41] are indeed Wasserstein barycenter problems, and proposed variants of the latter algorithm exploiting the particular structure of such subproblems. Preliminary numerical experiments show that furnishing the scenario tree reduction algorithm with specialized solvers for computing Wasserstein barycenters can significantly boost its performance, making the algorithm effective even in situations considered impractical prior to this work.

[1] https://www.siam.org/publications/journals/siam-journal-on-mathematics-of-data-science-simods

4.1 Participation in scientific events and valorization of our research

The MAM algorithm has been presented in several scientific events:

- Presentation during the PGMO days, the annual conference of the Optimization, OR, and Data Science program of the FMJH (Fondation Mathématiques Jacques Hadamard). (https://smf.emath.fr/evenements-smf/pgmo-days-2023)
- Presentation of a poster at the Consortium in Applied Mathematics (CIROQUO). (https://ciroquo.ec-lyon.fr/evenements.html)
- Presentation of a poster during the DATA IA days of CentraleSupelec. (https://www.dataia.eu/)

Notable conferences and events I attended this year include:

- NeurIPS Paris (https://scai.sorbonne-universite.fr/public/events/view/4ad2190f2c212abfd60f/8)

Key courses I participated in:

- MVA/MASH Course on Computational Optimal Transport by Gabriel Peyré (https://www.master-mva.com/cours/computational-optimal-transport/)
- PSL week on Stochastic Optimization by Welington de Oliveira (https://www.oliveira.mat.br/teaching)

Courses I instructed:

- Mines Paris, *Mathematics for Data Science* (L3), Centre de Morphologie Mathématique (https://www.cmm.minesparis.psl.eu/), with Bruno Figliuzzi and Chloé-Agathe Azencott.

4.2 Future research directions

Several topics will be considered throughout the remainder of this PhD work:

- We have just received the Referee Reports on the manuscript submitted to SIMODS. Consequently, we will carefully consider all their remarks and corrections before resubmitting it within the next month. This includes comparisons with additional methods for computing Wasserstein barycenters.
- Such additional methods might be considered, if competitive with MAM, in the second manuscript we are currently writing about scenarios tree reduction.
- Further improvements on the combination of Wasserstein barycenters and scenario tree reduction are conceivable. We plan to experiment warm-starting of MAM through several node-dependent barycenter problems.
- We will delve into exploring a new approach to accelerate the algorithm for the unbalanced barycenter problem, inspired by the Frank-Wolfe algorithm (FW) [4]. This method has been investigated within a Sinkhorn's approach [45]. However, exploring it within the unbalanced formulation we proposed, eq. (2.10) could yield interesting insights. The idea involves linearizing the square euclidean distance used in eq. (2.10), making minimization with the FW algorithm explicit. This approach could potentially bypass the most costly step of the MAM method, which is the projection onto the simplex. The main focus will be on studying the convergence tradeoff: recognizing that while each FW iteration is fast, the number of iterations needed for convergence can be considerable.

- We anticipate in proposing a more exact and efficient approach for the reduction scenario tree problem: We would like to enhance the MAM method to address non-convex problems. We would consider a Clipped Wasserstein Barycenter Problem, inspired by [22] (see Example 3). The main idea would be to force the WB to be sparser. Alternatively, leveraging a Progressive Decoupling Algorithm could offer a viable solution.
- We want to propose a new concept of barycenters, which we plan to call *Process Barycenter* and will be defined by employing the Nested/Process distance of G. Pflug [52]. While proposing such a new concept is a simple task, investigating whether such an original barycenter offers new insights, mathematical properties, or computational features for scenario tree reduction is still to be determined.

Key engagements I have planned include presenting additional findings on an enhanced iteration of the MAM method at the prestigious international conference ISMP 2024 (International Symposium on Mathematical Programming: https://ismp2024.gerad.ca/). Additionally, in April, I am going to visit the academic thesis advisor at CMA (Sohia Antipolis) for one month, to conduct technical research on the topics introduced earlier.

Moving forward, my objective for the remainder of this PhD thesis will be to share results with the international community by participating in renowned conferences such as ICML, NeurIPS, AISTATS, among others.

4.3 Career plan

This PhD experience has deepened my passion for science and research, which was already ignited during my two-year experience as a data scientist post-graduation. It has become increasingly clear to me that my career aspirations lie in research, ideally within the Research and Development department of a technology company. I am particularly drawn to companies whose primary products are grounded in research, fostering an environment of curiosity and intellectual challenge.

While I am open to exploring various industrial sectors, my interests are particularly piqued by energy and biomedical research. Ideally, I would pursue a role focused on optimal transport research. However, I am also open to opportunities in other sectors, especially at leading tech companies like Google, Facebook, Amazon, etc., known for their high scientific standards and emphasis on state-of-the-art methods.

While I am intrigued by quantitative research in fields like finance, I am less inclined to pursue projects in this sector. However, I remain open to learning more about quantifier research, as it aligns with my mathematical preferences. Despite this interest, I am unlikely to actively seek out such projects.

I aspire to explore professional opportunities abroad. However, I am hesitant to pursue a Postdoc position due to the demanding and precarious nature of the academic system. While I am deeply attracted to pure science, I am not willing to sacrifice my personal life for it.

Bibliography

[1] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *Siam Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[2] Gilles Bareilles, Yassine Laguel, Dmitry Grishchenko, Franck Iutzeler, and Jérôme Malick. Randomized progressive hedging methods for multi-stage stochastic programming. *Annals of Operations Research*, 295(2):535–560, sep 2020.

[3] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2nd edition, 2017.

[4] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[5] R Bellman. Dynamic programming (new jersey: Princeton university press). 1957.

[6] Felipe Beltran, Welington De Oliveira, and Erlon Cristian Finardi. Application of scenario tree reduction via quadratic process to medium-term hydrothermal scheduling problem. *IEEE Transactions on Power Systems*, 32(6):4351–4361, 2017.

[7] Felipe Beltrán, Erlon C Finardi, and Welington de Oliveira. Two-stage and multi-stage decompositions for the medium-term hydrothermal scheduling problem: A computational comparison of solution techniques. *International Journal of Electrical Power & Energy Systems*, 127:106659, 2021.

[8] Felipe Beltrán, Erlon C Finardi, Guilherme M Fredo, and Welington de Oliveira. Improving the performance of the stochastic dual dynamic programming algorithm using chebyshev centers. *Optimization and engineering*, pages 1–22, 2020.

[9] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[10] Dimitri Bertsekas. Dynamic programming and optimal control: Volume i, 2012.

[11] Dimitri P. Bertsekas. *Convex Optimization Algorithms*. Number 1st. Athena Scientific, 2015.

[12] J. Frédéric Bonnans. *Convex and Stochastic Optimization*. Universitext. Springer International Publishing, Cham, 2019.

[13] Steffen Borgwardt. An lp-based, strongly-polynomial 2-approximation algorithm for sparse wasserstein barycenters. *Operational Research*, 22(2):1511–1551, Apr 2022.

[14] Guillaume Carlier, Adam Oberman, and Edouard Oudet. Numerical methods for matching for teams and wasserstein barycenters. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1621–1642, nov 2015.

[15] Pierre Carpentier, Jean-Philippe Chancelier, Guy Cohen, and Michel De Lara. *Stochastic Multi-Stage Optimization: At the Crossroads between Discrete Time Stochastic Control and Stochastic Programming*, volume 75 of *Probability Theory and Stochastic Modelling*. Springer International Publishing, Cham, 2015.

[16] Zhiping Chen and Zhe Yan. Scenario tree reduction methods through clustering nodes. *Computers & Chemical Engineering*, 109:96–111, 2018.

[17] Laurent Condat. Fast projection onto the simplex and the $\boldsymbol{l_1}$ball. *Mathematical Programming*, 158(1):575–585, Jul 2016.

[18] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[19] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 685–693, Bejing, China, 22–24 Jun 2014. PMLR.

[20] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. *International Conference on Machine Learning*, 32(2):685–693, 2014.

[21] Marco Cuturi and Gabriel Peyré. A smoothed dual approach for variational wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016.

[22] Welington de Oliveira. The abc of dc programming. *Set-Valued and Variational Analysis*, 28:679–706, 2020.

[23] Welington de Oliveira, Claudia Sagastizábal, Débora Dias Jardim Penna, Maria Elvira Pineiro Maceira, and Jorge Machado Damázio. Optimal scenario tree reduction for stochastic streamflows in power generation planning problems. *Optimization Methods and Software*, 25(6):917–936, 2010.

[24] Welington Luis de Oliveira, Claudia Sagastizábal, Débora Dias Jardim Penna, Maria Elvira Pineiro Maceira, and Jorge Machado Damázio. Optimal scenario tree reduction for stochastic streamflows in power generation planning problems. *Optimisation Methods & Software*, 25(6):917–936, 2010.

[25] Jim Douglas and Henry H Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2):421–439, 1956.

[26] Jitka Dupačová, Nicole Gröwe-Kuska, and Werner Römisch. Scenario reduction in stochastic programming. *Mathematical programming*, 95:493–511, 2003.

[27] Jonathan Eckstein and Dimitri P. Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, apr 1992.

[28] Anqi Fu, Junzi Zhang, and Stephen Boyd. Anderson accelerated douglas–rachford splitting. *SIAM Journal on Scientific Computing*, 42(6):A3560–A3583, jan 2020.

[29] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[30] A. Gramfort, G. Peyré, and M. Cuturi. Fast optimal transport averaging of neuroimaging data. In Sebastien Ourselin, Daniel C. Alexander, Carl-Fredrik Westin, and M. Jorge Cardoso, editors, *Information Processing in Medical Imaging*, pages 261–272, Cham, 2015. Springer International Publishing.

[31] Tartavel Guillaume, Peyré Gabriel, and Gousseau Yann. Wasserstein loss for image synthesis and restoration. *SIAM Journal on Imaging Sciences*, 9(4):1726–1755, 2016.

[32] Florian Heinemann, Marcel Klatt, and Axel Munk. Kantorovich–rubinstein distance and barycenter for finitely supported measures: Foundations and algorithms. *Applied Mathematics & Optimization*, 87(1):4, Nov 2022.

[33] Holger Heitsch and Werner Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118:371–406, 2009.

[34] Holger Heitsch and Werner Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118:371–406, 2009.

[35] Holger Heitsch and Werner Römisch. Scenario tree reduction for multistage stochastic programs. *Computational Management Science*, 6:117–133, 2009.

[36] Holger Heitsch, Werner Römisch, and Cyrille Strugarek. Stability of multistage stochastic programs. *SIAM Journal on Optimization*, 17(2):511–525, 2006.

[37] Markéta Horejšová, Sebastiano Vitali, Miloš Kopa, and Vittorio Moriggia. Evaluation of scenario reduction algorithms with nested distance. *Computational Management Science*, 17(2):241–275, 2020.

[38] Franck Iutzeler, Pascal Bianchi, Philippe Ciblat, and Walid Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. In *52nd IEEE Conference on Decision and Control*. IEEE, dec 2013.

[39] Sanjula Kammammettu and Zukui Li. Scenario reduction and scenario tree generation for stochastic programming using sinkhorn distance. *Computers & Chemical Engineering*, 170:108122, 2023.

[40] Philip A Knight. The sinkhorn–knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.

[41] Raimund M Kovacevic and Alois Pichler. Tree approximation for discrete time stochastic processes: a process distance approach. *Annals of operations research*, 235(1):395–421, 2015.

[42] Mohamed Yacine Lamoudi. *Distributed model predictive control for energy management in buildings*. PhD thesis, Université de Grenoble, 2012.

[43] Zhuangzhi Li and Zukui Li. Linear programming-based scenario reduction using transportation distance. *Computers & Chemical Engineering*, 88:50–58, 2016.

[44] Zukui Li and Christodoulos A Floudas. Optimal scenario reduction framework based on distance of uncertainty distribution and output performance: I. single reduction via mixed integer linear optimization. *Computers & Chemical Engineering*, 70:50–66, 2014.

[45] Giulia Luise, Saverio Salzo, Massimiliano Pontil, and Carlo Ciliberto. Sinkhorn barycenters with free support via frank-wolfe algorithm. *Advances in neural information processing systems*, 32, 2019.

[46] Daniel Mimouni, Paul Malisani, Jiamin Zhu, and Welington de Oliveira. Computing wasserstein barycenter via operator splitting: the method of averaged marginals. *arXiv preprint arXiv:2309.05315*, 2023.

[47] Frauke Oldewurtel. *Stochastic model predictive control for energy efficient building climate control*. PhD thesis, ETH Zurich, 2011.

[48] Francois Pacaud, Michel De Lara, Jean-Philippe Chancelier, and Pierre Carpentier. Distributed Multistage Optimization of Large-Scale Microgrids Under Stochasticity. *IEEE Transactions on Power Systems*, 37(1):204–211, 2022.

[49] Gabriel Peyré. Bregmanot, 2014.

[50] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[51] G Ch Pflug. Version-independence and nested distributions in multistage stochastic optimization. *SIAM Journal on Optimization*, 20(3):1406–1420, 2010.

[52] Georg Ch Pflug and Alois Pichler. A distance for multistage stochastic optimization models. *SIAM Journal on Optimization*, 22(1):1–23, 2012.

[53] Georg Ch. Pflug and Alois Pichler. *Multistage Stochastic Optimization*. Springer International Publishing, 2014.

[54] Alois Pichler and Michael Weinhardt. The nested sinkhorn divergence to learn the nested distance. *Computational Management Science*, 19(2):269–293, 2022.

[55] Giovanni Puccetti, Ludger Rüschendorf, and Steven Vanduffel. On the computation of wasserstein barycenters. *Journal of Multivariate Analysis*, 176(104581), 2020.

[56] R. Tyrrell Rockafellar. Solving Stochastic Programming Problems with Risk Measures by Progressive Hedging. *Set-Valued and Variational Analysis*, 26(4):759–768, 2018.

[57] R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.

[58] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov 2000.

[59] Thibault Sejourne, Gabriel Peyre, and Francois-Xavier Vialard. Unbalanced optimal transport, from theory to numerics. *Handbook of Numerical Analysis*, 24:407–471, 2023.

[60] Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[61] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics, 2009.

[62] Dror Simon and Aviad Aberdam. Barycenters of natural images constrained wasserstein barycenters for image morphing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7910–7919, 2020.

[63] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. ii. *Proceedings of the American Mathematical Society*, 45(2):195–198, 1974.

[64] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[65] Tijmen. affnist, 2013.

[66] Cedric Villani. *Optimal transport: onld and new*, volume 338. Springer Verlag, 2009.

[67] Huahua Wang and Arindam Banerjee. Bregman alternating direction method of multipliers. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[68] Jean-Paul Watson and David L. Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, jul 2010.

[69] Daobao Xu, Zhiping Chen, and Li Yang. Scenario tree generation approaches using k-means and lp moment matching methods. *Journal of Computational and Applied Mathematics*, 236(17):4561–4579, 2012.

[70] Zheng Xu, Mario Figueiredo, and Tom Goldstein. Adaptive ADMM with Spectral Penalty Parameter Selection. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 718–727. PMLR, 20–22 Apr 2017.

[71] Jianbo Ye and Jia Li. Scaling up discrete distribution clustering using ADMM. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5267–5271, 2014.

[72] Jianbo Ye, Panruo Wu, James Z. Wang, and Jia Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65:2317–2332, May 2017.