Tree Approximation for Discrete Time Stochastic Processes — A Process Distance Approach

Raimund M. Kovacevic* and Alois Pichler†

April 6, 2016

Abstract

Approximating stochastic processes by scenario trees is important in decision analysis. In this paper we focus on improving the approximation quality of trees by smaller, tractable trees. In particular we propose and analyze an iterative algorithm to construct improved approximations: given a stochastic process in discrete time and starting with an arbitrary, approximating tree, the algorithm improves both, the probabilities on the tree and the related path-values of the smaller tree, leading to significantly improved approximations of the initial stochastic process.

The quality of the approximation is measured by the process distance (nested distance), which was introduced recently. For the important case of quadratic process distances the algorithm finds locally best approximating trees in finitely many iterations by generalizing multistage k-means clustering.

Keywords: Stochastic processes and trees, Wasserstein and Kantorovich distance, tree approximation, optimal transport, facility location

Classification: 90C15, 60B05, 90-08

1 Introduction

Decision problems are often stated by employing the notion of stochastic processes and filtered probability spaces to describe the objects being studied. For continuous time and state space processes this setting, however, is often not practicable when implementing realizations for concrete computation. Approximating discrete time and finite state space models are therefore of critical importance. A basic data structure for this purpose is given by scenario trees, which model values, probabilities, and the basic evolution of the process. In fact, scenario trees (shortly called *trees* in what follows) are an important tool for all fields of decision analysis, in particular for multistage stochastic optimization, i.e., stochastic programming.

Moment matching, a widespread approach, is designed to fit values and/or probabilities of the approximating tree such that the difference between suitable moments of the two processes vanishes or at least is minimized. This approach was extended in Høyland and Wallace [HW01] by minimizing the Euclidean distance between arbitrary collections of user-defined summary statistics (cf. also the book [KW13] by King and Wallace). This ad hoc methodology is highly accepted among practitioners, but essentially is an heuristic lacking theoretical foundations. Moreover, it is well known that similar (conditional and unconditional) moments do not guarantee similarity of two (joint) distributions in general. It is also unknown how matching moments relates to the estimation quality of the objective value.

*Department of Statistics and Operations Research,University of Vienna, Austria and Institute of Statistics and Mathematical Methods in Economy, Vienna University of Technology, Austria. This research was partially funded by the Austrian science fund FWF, project P 24125-N13

†Norwegian University of Science and Technology. The author gratefully acknowledges support of the Research Council of Norway (grant 207690/ E20)

Another common technique for constructing trees is *sample average approximation* (SAA), which basically consists in randomly simulating values from the previously estimated conditional distribution at any node of the tree. It was observed by Nemirovski and Shapiro in [SN05] and by Shapiro in [Sha10] that solving sampled multistage optimization problems is often practically intractable. Indeed, their results indicate that $\mathcal{O}\left(\varepsilon^{-2T}\right)$ scenarios have to be sampled to obtain a precision of ε in the objective for a tree of height T. Employing more advanced techniques as described by Graf and Luschgy in [GL00] the number of scenarios can be reduced to $\mathcal{O}\left(\varepsilon^{-T}\right)$, but the growth remains exponential in T.

Further important approaches directly aim at minimizing a distance between the genuine and the approximating process. As an example, Dupačová et al. [DGKR03] consider Wasserstein or Kantorovich distances to measure the difference of probability distributions. Compared to SAA, Wasserstein-based approaches have the advantage that they do not rely on asymptotic arguments to ensure good approximation quality in terms of the value function (see the discussion on uniform bounds for expectations of Lipschitz functions in Section 2.1 below). Given that tractable trees usually have to be small in practice, this is a key property.

Finally it is important to account for the fact that filtrations modeling the evolution of the process over time (i.e., the information available) are essential in stochastic optimization. Any approximating tree will thus not only approximate values and probabilities of a given process, but also the related filtration by imposing a (preferably sparse) tree structure. Heitsch and Römisch study such a functional in [HR11] which they call filtration distance, although it is not a distance in the strict mathematical sense. However, stochastic programs are continuous with respect to this distance function, such that their functional provides a useful upper bound to compare trees. Heitsch and Römisch elaborate fast, theory-based heuristics to compute scenario trees in [HR09a] and to reduce trees in [HR09b].

In the present paper we follow the general distance-based approach, but we use a distance concept introduced in Pflug [Pfl09], called *process distance* or *nested distance* in what follows. It is a distance for stochastic processes which builds on the Wasserstein distance and incorporates the filtrations in a natural way by its nested structure without relying on a separate distance concept for filtrations. Pflug and Pichler [PP12] give a detailed analysis of the process distance in the context of stochastic optimization. In particular, under usual regularity conditions, multistage optimization problems are continuous with respect to the nested distance. The distance moreover provides a sharp upper bound. Hence, by employing the nested distance to control the approximation of the process it is possible to control both, the statistical quality of the approximation and the effect on the objective for every multistage stochastic optimization problem formulated on the corresponding stochastic process.

We present and analyze an algorithm to improve the process distance between a process, modeled by a given large scenario tree, and an approximating smaller tree with given tree structure. While the nested distance of two trees can be formulated as linear optimization problem, finding the best approximating tree leads to a high dimensional, highly nonlinear (in fact nonconvex) and combinatorial optimization problem. The problem cannot be solved directly in a satisfying way. The main part of the article therefore proposes and analyzes an iterative algorithm which exploits the nested structure of the process distance and guarantees successive improvements in terms of the process distance.

The algorithm iteratively reduces the nested distance relative to the initial process in two steps. The first step to find improved probabilities is computationally expensive, while the second step to improve the values on the tree can be executed sufficiently fast (at least for suitable choices of the underlying metric).

Outline of the paper. The following Section 2 reviews main facts about the process distance, which are relevant for the following discussion. This section as well introduces the notation, which is necessary for applications involving trees. Based on this, Section 3 analyzes how to improve the values and the probabilities within a given tree structure in order to improve the approximation quality. This section introduces the overall algorithm and provides instructive numerical examples. The summary (Section 4) concludes with a discussion.

Appendix A gives an overview of approximations with Wasserstein distances. Appendix B motivates and explains the background for the tree notation used, and in particular addresses the relations between trees and filtered probability spaces.

2 The process distance for stochastic processes

The nested distance is a distance for stochastic processes, while the Wasserstein distance is a distance for probability measures. The nested distance is built on the Wasserstein distance. In order to apply the concepts to discrete time processes (i.e., trees) the respective discrete setting is elaborated first. The corresponding linear optimization problems are particularly important in analyzing the approximation algorithm below.

2.1 Wasserstein distance

A comprehensive summary on the Wasserstein distance can be found, e.g., in Rachev and Rüschendorf [RR98] and in Villani [Vil03]. The Wasserstein distance is well adapted in the context of approximating probability measures, because it metrizes weak convergence and discrete measures are dense in the corresponding space of probability measures.

Definition 1 (Wasserstein distance)**.** Given two probability spaces (Ξ, Σ, P) and (Ξ, Σ', P') and a distance function $d\colon \Xi \times \Xi \to \mathbb{R}$, the *Wasserstein distance of order* $r \geq 1$, denoted $\mathsf{d}_r(P, P')$, is the optimal value of the optimization problem

$$\underset{(\text{in } \pi)}{\text{minimize}} \quad \left(\iint d(\xi, \xi')^r \, \pi(\mathrm{d}\xi, \mathrm{d}\xi') \right)^{\frac{1}{r}} \tag{1}$$

$$\text{subject to } \pi(M \times \Xi) = P(M) \qquad (M \in \Sigma), \tag{2}$$

$$\pi(\Xi \times N) = P'(N) \qquad (N \in \Sigma'), \tag{3}$$

where the minimum in (1) is among all bivariate probability measures $\pi \in \mathcal{P}(\Xi \times \Xi)$ which are measures on the product sigma algebra $\Sigma \otimes \Sigma'$. The measure π satisfying the constraints (2) ((3), resp.), is said to have marginals P (P', resp.).

Remark 1 (On the term *Wasserstein distance*). The term Wasserstein distance is not used consistently in the literature. The Wasserstein distance of order $r = 1$ is often called Kantorovich distance. Vershik [Ver06] calls the distance d_r *Kantorovich metric* for all $r \geq 1$, while Rachev and Rüschendorf [RR98, p. 40] (cf. also [Rac91]) propose the name L_r-Wasserstein metric. For a detailed, further discussion and how the term became accepted we refer to Villani [Vil09, bibliographical notes].

As a matter of fact the Wasserstein distance depends on the sigma algebras Σ and Σ', although this fact is neglected by writing $\mathsf{d}_r(P, P')$.[1] Of particular interest is the Wasserstein distance of order $r = 2$ with an Euclidean norm $d(\xi, \xi') = \|\xi - \xi'\|_2$ on a vector space Ξ. We shall refer to this combination as the *quadratic Wasserstein distance*.

The problem (1)–(3) allows a useful interpretation as a transportation problem. The resulting functional $\mathsf{d}_r(\cdot, \cdot)$ is known to be a full distance on probability spaces. Furthermore, convergence in $\mathsf{d}_r(\cdot, \cdot)$ is equivalent to weak* convergence plus convergence of the r-th moment (cf. Rachev's monograph [Rac91, Chapter 5]).

More generally, the transportation problem (1) is often considered for lower semi-continuous cost functions c replacing the distance function d (cf. Schachermayer and Teichmann [ST09]) and for general, measurable cost functions by Schachermayer et al. in [BGMS09, BLS12].

It has moreover been shown (see, e.g., Dupačová et al. [DGKR03]) that single stage expected loss minimization problems with objective function $\mathbb{E}_\xi H(\xi, x)$ are (under some regularity conditions on the loss function H) Lipschitz continuous with respect to the Wasserstein distance.

[1] Notice the notational difference: d is the distance function on the original space Ξ, while d_r denotes the Wasserstein distance.

Remark 2. If $P = \sum_i p_i \delta_{\xi_i}$ and $P' = \sum_j p'_j \delta_{\xi'_j}$ are discrete measures on Ξ, then the Wasserstein distance can be computed by solving the linear program (LP)

$$\begin{array}{ll} \text{minimize} & \sum_{i,j} d_{i,j}^r \pi_{i,j} \\ \text{(in } \pi\text{)} & \\ \text{subject to} & \sum_j \pi_{i,j} = p_i, \\ & \sum_i \pi_{i,j} = p'_j, \\ & \pi_{i,j} \geq 0, \end{array} \tag{4}$$

where $d_{i,j}$ is the matrix with entries $d_{i,j} = d\left(\xi_i, \xi'_j\right)$.

It follows from complementary slackness conditions for linear programs that the optimizing transport plan $\pi_{i,j}$ in (4) is sparse. The matrix π has at most $|\Xi|+|\Xi'|-1$ non-zero entries, which correspond to the number of entries in one row, plus the number of columns of the matrices π or d.

Based on this setup it is straightforward to provide a discrete probability measure, which approximates a given measure in best possible way with respect to the Wasserstein distance. Algorithm 2 in the Appendix provides the corresponding procedure, which is comparably easy from a computational point of view.

Uniform bounds for expectations of Lipschitz functions and curse of dimensionality. The dual of the optimization problem (1) is provided by the Kantorovich–Rubinstein theorem, which implies the inequality

$$|\mathbb{E}_P f - \mathbb{E}_{P'} f| \leq Lip(f) \cdot \mathsf{d}_1(P, P') \leq Lip(f) \cdot \mathsf{d}_r(P, P'), \tag{5}$$

where $Lip(f)$ is the Lipschitz constant of f. The first inequality in (5) is sharp and cannot be improved. It follows thus that the Wasserstein distance is the best possible distance to compare expectations of Lipschitz functions. Useful approximations minimize the right hand side of (5) in order to obtain optimal approximations, which are uniformly best for all Lipschitz functions.

Dudley characterizes the quality of discrete approximations by providing a lower, asymptotic bound (see [Dud69, Proposition 2.1] for the exact formulation, cf. also [GL00, Theorem 6.2]): a continuous measure P cannot be approximated better by $P_n = \sum_{i=}^n p_i \delta_{\xi_i}$ (a discrete measure concentrated on not more than n points $\left\{\xi_i \in \mathbb{R}^k : i = 1, \dots n\right\}$), than

$$\mathsf{d}_r\left(P,\, P_n\right) \geq \gamma \cdot n^{-1/k} \tag{6}$$

($\gamma > 0$ depends on the measure P, but not on the finite set $\{\xi_i : i = 1, \dots n\}$). The approximation quality of discrete approximations thus depends strongly on the dimension k of the underlying space. This shows that the quantity on the left hand side of (5) can be large, particularly for high dimensional problems. This fact is often referred to as *curse of dimensionality*.

Faster convergence rates than stated in (6) can only be obtained by restricting the test functions to smaller classes. As an (extreme) example consider the best approximation, which is located on a single point for $r = 2$, given explicitly by

$$P' = \delta_{x_\mu}, \text{ where } x_\mu := \int x\, P(\mathrm{d}x) \tag{7}$$

is the barycentre of the measure P (cf. Graf and Luschgy [GL00, Remark 4.6]). For linear functions f expectations then are even exact, as

$$\mathbb{E}_{P'} f = \int f \mathrm{d}P' = f(x_\mu) = \int f \mathrm{d}P = \mathbb{E}_P f. \tag{8}$$

by linearity of f and the expectation.

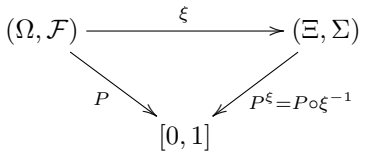


Figure 1: Diagram for the pushforward measure P^ξ

An immediate extension to processes. The Wasserstein distance can also be used as a distance for random processes. Consider a stochastic process $\xi = (\xi_t)_{t \in \{0,\dots T\}}$ in finite time $t \in \{0, \dots T\}$, where $\xi_t \colon (\Omega, \mathcal{F}) \to (\Xi_t, d_t)$ are random variables with possibly different state spaces (Ξ_t, Σ_t). Here, Ξ_t is equipped with the Borel sigma algebra Σ_t induced by a metric d_t.

The product space $\Xi := \Xi_0 \times \cdots \times \Xi_T$ itself can be equipped with the product sigma algebra $\Sigma := \sigma(\Sigma_0 \otimes \cdots \otimes \Sigma_T)$. In fact

$$\begin{aligned} \xi : (\Omega, \mathcal{F}) \to & \quad (\Xi, \Sigma) \\ \omega \mapsto & \quad (\xi_t(\omega))_{t \in \{0,\dots T\}} \end{aligned}$$

is a random variable, mapping any outcome $\omega \in \Omega$ to its entire path $(\xi_t(\omega))_{t=0}^T$.

While the process is originally defined on an abstract probability space $(\Omega, \mathcal{F}, P)$, we are first of all interested in distances related to the state space Ξ of paths. Therefore recall that any random variable

$$\xi \colon (\Omega, \mathcal{F}) \to (\Xi, d)$$

on a probability space $(\Omega, \mathcal{F}, P)$ naturally induces the *pushforward measure* (also *induced* or *image measure*)

$$P^\xi := P \circ \xi^{-1} \colon \Sigma \to [0, 1],$$

where Σ is the sigma algebra induced by the Borel sets generated by the distance d (cf. Figure 1). Hence the distance $\mathsf{d}_r\left(P^\xi, P'^{\xi'}\right)$ is available on the image space, where

$$\xi' \colon (\Omega', \mathcal{F}') \to (\Xi, d),$$

is another random variable with same state space as ξ and

$$d \colon \Xi \times \Xi \to \mathbb{R}$$

is the distance function employed by the Wasserstein distance.

This idea can be used also for processes. The law of a process $\xi(\omega) = (\xi_t(\omega))_{t=0}^T$,

$$P^\xi := P \circ \xi^{-1} \colon \Sigma \to [0, 1],$$

is the pushforward measure on $\Xi = \Xi_0 \times \cdots \times \Xi_T$. In this way the Wasserstein distance $\mathsf{d}\left(P^\xi, P'^{\xi'}\right)$ can be applied to processes ξ and ξ'.

However, Wasserstein distances do not correctly separate processes having different filtrations. The following example illustrates this shortfall.

Example 1. Figure 2 displays an example, where similar paths (small values of $\varepsilon > 0$) lead to a small Wasserstein distance between the first and the second tree:

$$\mathsf{d}(1^{st}, 2^{nd}) \sim \varepsilon, \ \mathsf{d}(1^{st}, 3^{rd}) \sim 1 \text{ and } \mathsf{d}(2^{nd}, 3^{rd}) \sim 1.$$

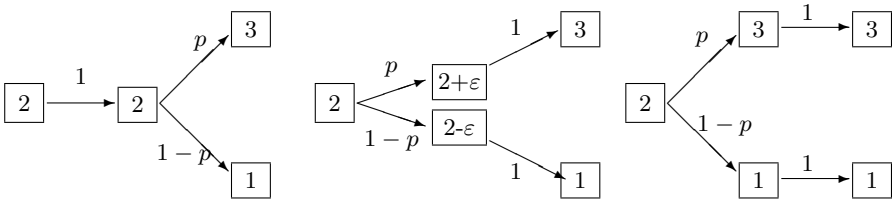


Figure 2: Three tree processes illustrating three different flows of information (cf. Heitsch et al. [HRS06]). The Wasserstein distance of the first two trees vanishes (cf. Example 1), while the nested distance does not (cf. Example 2).

The vanishing Wasserstein distance of the first two trees reflects the fact that the distance on the probability space ignores the filtrations, that is the information, which is available at an earlier stage of the tree: while nothing is known about the final outcome in the first tree, perfect information about the final outcome is available already at the intermediary step for the second tree. However, their Wasserstein distance vanishes. It has to be concluded that the Wasserstein distance is not suitable to distinguish stochastic processes. Example 2 below resumes the trees from Figure 2 and resolves the problem.

Remark 3. Several distance functions $d\colon \Xi\times\Xi' \to \mathbb{R}$ are available on the product spaces: the ℓ^1-distance

$$d(\xi,\xi') = \sum_{t=0}^{T} d_t(\xi_t,\xi'_t),$$

or the ℓ^∞-distance

$$d(\xi,\xi') = \max_{t\in\{0,\dots T\}} d_t(\xi_t,\xi'_t)$$

are immediate candidates. In what follows we shall often employ the *(weighted) Euclidean* distance

$$d(\xi,\xi') = \left(\sum_{t=0}^{T} w_t \cdot \|\xi_t - \xi'_t\|_2^2\right)^{1/2},$$

where $w_t > 0$ are positive weights and each norm $\|\cdot\|_2$ satisfies the parallelogram law.

2.2 Process distance

Process distances are multistage distances, extending and generalizing the Wasserstein distance to stochastic processes. They were recently introduced by Pflug [Pfl09] and analyzed in [PP12]. Such distances account for the values and probability laws of stochastic processes (as is the case for the Wasserstein distance), but take the filtration into account in addition.

Definition 2 (Nested distance)**.** For two filtered probability spaces $\mathbb{P} := \left(\Xi, (\Sigma_t)_{t=0}^T, P\right)$, $\mathbb{P}' := \left(\Xi, (\Sigma'_t)_{t=0}^T, P'\right)$ and a real-valued distance function $d\colon \Xi\times\Xi \to \mathbb{R}$ the *process distance of order* $r \geq 1$, denoted $\mathsf{dl}_r(\mathbb{P},\mathbb{Q})$, is the optimal value of the optimization problem

$$\begin{aligned}
&\underset{(\text{in } \pi)}{\text{minimize}} && \left(\iint d(\xi,\xi')^r \,\pi(\mathrm{d}\xi,\mathrm{d}\xi')\right)^{\frac{1}{r}} && (9)\\
&\text{subject to } && \pi(M\times\Xi' \mid \Sigma_t\otimes\Sigma'_t) = P(M\mid\Sigma_t) \qquad (M\in\Sigma_T,\, t=0,\dots T), && (10)\\
&&& \pi(\Xi\times N \mid \Sigma_t\otimes\Sigma'_t) = P'(N\mid\Sigma'_t) \qquad (N\in\Sigma'_T,\, t=0,\dots T), && (11)
\end{aligned}$$

where the infimum is among all bivariate probability measures $\pi \in \mathcal{P}(\Xi\times\Xi)$, which are probability measures on the product sigma algebra $\Sigma\otimes\Sigma'$. The process distance dl_2 (order $r=2$) with d a weighted Euclidean distance, is referred to as *quadratic process distance*.

The minimization (1) to compute the Wasserstein distance $\mathsf{d}_r(P, P')$ is a relaxation of (9), because the measures π in (9) do not only respect the marginals imposed by P and P', they respect the conditional marginals (10) and (11) as well. The Wasserstein distance thus is always less or equal than the related process distance,

$$\mathsf{d}_r(P, P') \leq \mathsf{dl}_r(\mathbb{P}, \mathbb{P}').$$

It is possible therefore to interpret any process distance as consisting of two parts: a first part accounts for the distance of the measures, while the gap $\mathsf{dl}_r(\mathbb{P}, \mathbb{P}') - \mathsf{d}_r(P, P')$ is caused by the filtration, which results from employing the additional, marginal constraints.

Analogical to (5) the process distance $\mathsf{dl}_r(\cdot, \cdot)$ also preserves important regularity properties such as Lipschitz or Hölder continuity of the utility function of multistage stochastic programs with expected utility (or similar) objectives. In particular this leads to bounds on the distance between the optimal values of the objective functions of the original and the approximating optimization problem (cf. Pflug and Pichler [PP12, Section 6]).

Example 2 (Continuation of Example 1)**.** The process distance (nested distance) is designed to detect the impact of the filtrations. The nested distances of the trees in Figure 2 are ancestor

$$\mathsf{dl}(1^{st}, 2^{nd}) \sim 1,\ \mathsf{dl}(1^{st}, 3^{rd}) \sim 1 \text{ and } \mathsf{dl}(2^{nd}, 3^{rd}) \sim 2,$$

which is in essential contrast to the Wasserstein distance.

2.3 Scenario trees and notation

In what follows we focus on modeling approximations of stochastic processes by trees. A tree is basically a directed graph $(\mathcal{N}, A)$ without cycles (cf. Figure 3). It is accepted to call the vertices $\mathcal{N}$ *nodes* (cf. Pflug and Römisch [PR07, p. 216]). 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 predecessor relation between m and n is denoted by $m = n_-$. The set of *direct successor*s (or *children*) of a vertex m is denoted by m_+, such that $n \in m_+$ if and only if $m = n_-$. Moreover, a node $m \in \mathcal{N}$ is said to be a *predecessor* (or ancestor) of $n \in \mathcal{N}$ — in symbols: $m \in \mathcal{A}(n)$ — if $n_1 \in m_+$, $n_2 \in n_{1+}$, and finally $n \in n_{k+}$ for some sequence $n_k \in \mathcal{N}$. It holds in particular that $n_- \in \mathcal{A}(n)$.

We consider only trees with a single *root*, denoted by 0, i.e. $0_- = \emptyset$. 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 (a path)

$$\omega = (0, \ldots, n_-, n)$$

from the root to the leaf node. Every such sequence is called a *scenario*. We shall write $(0, \ldots, n) = (n_0, \ldots, n_t, \ldots n_T)$ for every scenario induced by a leaf n, if $n_0 \in \mathcal{A}(n_1)$, $n_1 \in \mathcal{A}(n_2)$ etc. and $n_T = n$.

In an obvious manner we denote the probabilities, assigned to node n, by $P(n)$ and the values taken by the process ξ at node n by $\xi(n)$. We denote conditional probabilities between successors by $P(m|n) = P(m)/P(n)$ for $n = m_-$. Furthermore, using a distance d, we write $d_{mn} := d\big(\xi(m), \xi(n)\big)$. Appendix B collects further details on the relation between tree structure and filtrations.

2.4 The process distance for trees

The Wasserstein distance between discrete probability measures can be calculated by solving the linear program (4) above. To establish a similar linear program for the process distance we use trees that model a process and the related filtration. For this observe that the probability measure for the nested distance in (9)–(11) can be given by masses $\pi_{i,j}$ at the leaves $i \in \mathcal{N}_T$ and $j \in \mathcal{N}'_T$. The probability at earlier nodes $m \in \mathcal{N}_t$ and $n \in \mathcal{N}'_t$ can be given as well by $\pi_{m,n} = \sum_{\{i \in \mathcal{N}_T:\, m \in \mathcal{A}(i)\}} \sum_{\{j \in \mathcal{N}'_T:\, n \in \mathcal{A}(j)\}} \pi_{i,j}$, and particularly the conditional probabilities

$$\pi(i, j \,|\, m, n) = \frac{\pi_{i,j}}{\pi_{m,n}} \tag{12}$$

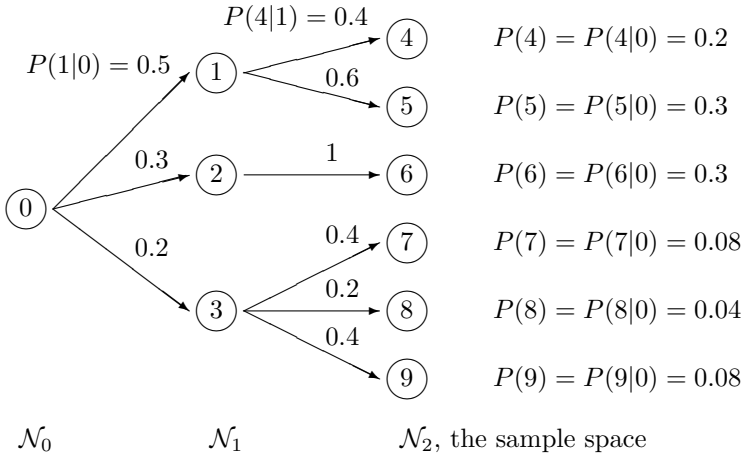


Figure 3: An exemplary finite tree process $\nu = (\nu_0, \nu_1, \nu_2)$ with nodes $\mathcal{N} = \{0, \dots 9\}$ and leaves $\mathcal{N}_2 = \{4, \dots 9\}$ at $T = 2$ stages. The filtrations, generated by the respective atoms, are $\mathcal{F}_2 = \sigma(\{4\}, \{5\}, \dots \{9\})$, $\mathcal{F}_1 = \sigma(\{4, 5\}, \{6\}, \{7, 8, 9\})$ and $\mathcal{F}_0 = \sigma(\{4, 5, \dots 9\})$ (cf. [PR07, Section 3.1.1])

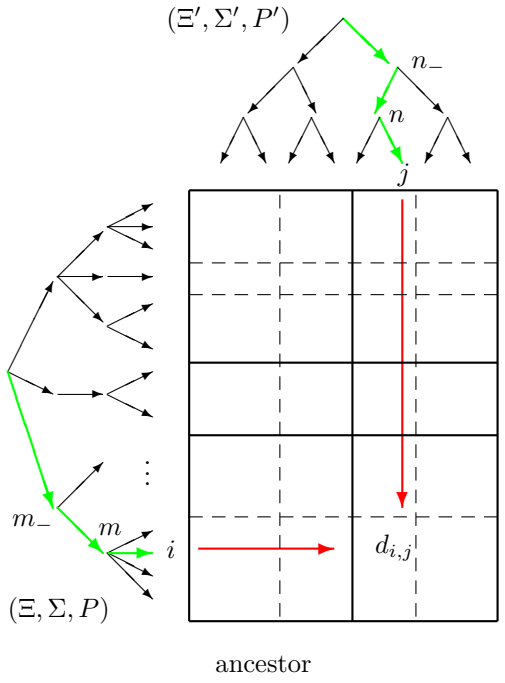


Figure 4: Structure of the transport matrix π for two trees, each of height $T = 3$. m and n are intermediary nodes, i and j are leaves

thus are available.

The problem (9) to compute the nested distance, reformulated for trees, thus reads

$$\begin{array}{ll} \text{minimize} \\ \text{in } \pi \\ \text{subject to} \end{array} \quad \begin{array}{l} \sum_{i\in\mathcal{N}_T, j\in\mathcal{N}_{T'}} \pi_{i,j}\cdot d_{i,j}^r \\ \sum_{\{j:\, n\in\mathcal{A}(j)\}} \pi\left(i,j\,|\,m,n\right) = P\left(i\,|\,m\right) \quad (m\in\mathcal{A}(i),\ n), \\ \sum_{\{i:\, m\in\mathcal{A}(i)\}} \pi\left(i,j\,|\,m,n\right) = P'\left(j\,|\,n\right) \quad (n\in\mathcal{A}(j),\ m), \\ \pi_{i,j}\geq 0 \text{ and } \sum_{i,j}\pi_{i,j}=1, \end{array} \tag{13}$$

where $i\in\mathcal{N}_t$, $j\in\mathcal{N}_t'$ are arbitrary nodes at stages $0,1,\ldots,T-1$ and $\pi\left(i,j\,|\,m,n\right)$ is as defined in (12). The nested structure of the transportation plan π, which is induced by the trees, is depicted schematically in Figure 4. In contrast to the Wasserstein case, (4), it is necessary to include the constraint $\sum_{i,j}\pi_{i,j}=1$ in (13), because otherwise every multiple $\lambda\cdot\pi$ ($\lambda\in\mathbb{R}$) would be feasible as well.

By replacing the conditional probabilities $\pi(\cdot|\cdot)$ by (12) and observing that the conditional probabilities $P(\cdot|\cdot)$, $P'(\cdot|\cdot)$ are given, the equations (13) can naturally be rewritten as a linear optimization problem in the joint probabilities $\pi_{i,j}$.

Note that many constraints in (13) and its reformulation are linearly dependent. For computational reasons (loss of significance during numerical evaluations, which can impact linear dependencies and the feasibility) it is advisable to remove linear dependencies. In particular it is possible to narrow down (13) by using only one step conditional probabilities leading to the equivalent (justified in [PP12, Lemma 10]) problem

$$\begin{array}{ll} \text{minimize} \\ \text{in } \pi \\ \text{subject to} \end{array} \quad \begin{array}{ll} \sum_{i,j}\pi_{i,j}\cdot d_{i,j}^r & \\ \sum_{j\in n_+}\pi\left(i,j\,|\,i_-,n\right)=P\left(i\,|\,i_-\right) & \left(i_-\in\mathcal{N}_t,\ n\in\mathcal{N}_t'\right), \\ \sum_{i\in m_+}\pi\left(i,j\,|\,m,j_-\right)=P'\left(j\,|\,j_-\right) & \left(j_-\in\mathcal{N}_t,\ m\in\mathcal{N}_t'\right), \\ \pi_{i,j}\geq 0 \text{ and } \sum_{i,j}\pi_{i,j}=1, & \end{array} \tag{14}$$

which represents an LP by substituting the conditional probabilities (12). Further constraints can be removed from (14) by taking into account that $\sum_{i_-=m_-}\frac{P(i)}{P(m_-)}=1$: for each node m it is possible to drop one constraint out of all $\left|(m_-)_+\right|$ related equations.

Recursive computation. It will be important in the following that the process distance can also be calculated in a recursive way instead of solving (14). Indeed, define first

$$\mathsf{dl}_r\left(i,j\right) := d\left(\xi_i,\xi_j'\right) \tag{15}$$

for leaf nodes $i\in\mathcal{N}_T$, $j\in\mathcal{N}_T'$. Given $\mathsf{dl}_r\left(i,j\right)$ for $i\in\mathcal{N}_{t+1}$ and $j\in\mathcal{N}_{t+1}'$ set

$$\mathsf{dl}_r\left(m,n\right)^r := \sum_{i\in m_+, j\in n_+}\pi\left(i,j\,|\,m,n\right)\cdot\mathsf{dl}_r\left(i,j\right)^r \qquad \left(m\in\mathcal{N}_t,\ n\in\mathcal{N}_t'\right) \tag{16}$$

for $m\in\mathcal{N}_t$, $n\in\mathcal{N}_t'$, where the conditional probabilities $\pi(\cdot,\cdot\,|\,m,n)$ solve

$$\begin{array}{ll} \text{minimize} \\ \text{in } \pi\left(.,.\,|\,m,n\right) \\ \text{subject to} \end{array} \quad \begin{array}{ll} \sum_{i\in m+, j\in n+}\pi\left(i,j\,|\,m,n\right)\cdot\mathsf{dl}_r\left(i,j\right)^r & \\ \sum_{j\in n_+}\pi\left(i,j\,|\,m,n\right)=P\left(i\,|\,m\right) & \left(i\in m_+\right), \\ \sum_{i\in m_+}\pi\left(i,j\,|\,m,n\right)=P'\left(j\,|\,n\right) & \left(j\in n_+\right), \\ \pi\left(i,j\,|\,m,n\right)\geq 0. & \end{array} \tag{17}$$

Each of the problems (17) is linear in the conditional probabilities and only conditional probabilities between one node and its descendants are used within each instance of (17). The values $\mathsf{dl}_r\left(i,j\right)$ can

be interpreted as conditional process distances for the subtrees starting in nodes i (j, resp.), such that the process distance of the full trees is given by $\mathsf{dl}\left(\mathbb{P},\mathbb{P}'\right)^r = \mathsf{dl}_r\left(0,0\right)$.

Finally the transport plan π on the leaves can be recomposed as

$$\pi\left(i,j\right) = \pi\left(i,j\middle|\, i_{T-1},j_{T-1}\right)\cdot\pi\left(i_{T-1},j_{T-1}\middle|\, i_{T-2},j_{T-2}\right)\cdot\ldots\pi\left(i_1,j_1\middle|\, 0,0\right)$$

by combining all results of (17).

3 Improving an approximating tree

This section addresses the question of finding trees, which are close to a given tree in terms of a process distance. The approximating tree has the same number of stages, but typically a considerably smaller number of nodes than the original tree to allow fast further computations. While it is easily possible to calculate the process distance between given trees by solving one large LP (or a sequence of smaller LPs as outlined in the previous section), finding an optimal approximating tree is much more difficult. Both, probabilities and the 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, nonconvex optimization problem, which can be solved in reasonable time only for small instances. In what follows we present an iterative algorithm for improving the process distance between a given tree and an approximating tree which allows for larger problem sizes.

The algorithm performs the following improvement steps in an iterative way:

(i) Given values (i.e., outcomes, or locations of the process) related to each node, find *probabilities* on a given tree structure with attached values, which decrease the process distance to the given tree.[2] This step involves solving several LPs.

(ii) Facility location: given probabilities on a tree structure, find values to improve the approximation. We propose a descent method within each iteration here.

While the algorithm iterates over steps (i) and (ii), in its first step there is an additional iteration over the stages of the tree. We discuss both steps separately in Sections 3.1 and 3.2 and summarize the overall algorithm in Section 3.3. A similar approach for improving the pure Wasserstein distance is added for completeness in Appendix A.

3.1 Optimal probabilities

Proposition 1 in Appendix A provides a closed form solution for the best approximation of a probability measure P by a measure P_Q^*, which is located (supported) just on the points $Q = \{q_1 \ldots q_n\}$ in terms of the Wasserstein distance (an optimal supporting points $q \in Q$ is occasionally called *quantizer* or *representative point* in the literature; note, that $P_Q^*\left(Q\right) = 1$). In the multistage environment, no closed form solution is available for the problem of optimal probabilities.

More concretely, the multistage problem of optimal probabilities can be stated as follows: which probability measure P_Q^* is *best* to approximate $\mathbb{P} = (\Xi, \Sigma, P)$ in nested distance, provided that the *states* $Q \subset \Xi'$ *and the filtration* Σ' of the stochastic processes are given? Knowing the branching structure of the tree, we seek for the best probabilities such that the process distance to $\mathbb{P}$ is as small as possible. In explicit terms this best approximation P_Q^* satisfies

$$\mathsf{dl}_r\big(\mathbb{P},\,(\Xi',\Sigma',P_Q^*)\big) \le \mathsf{dl}_r\big(\mathbb{P},\,(\Xi',\Sigma',P')\big) \quad \left(P'\left(Q\right)=1\right),$$

where $Q = \{q_1, \ldots q_n\} \subset \Xi'$.

By inspecting formulation (14) one sees that finding optimal probabilities for the approximating tree means that the related conditional probabilities $P'(j|j')$ are not known and have to be considered as decision variables. Because we want to minimize the distance, this leads to the optimization problem

[2] In the context of transportation and transportation plans, the paths of the stochastic process are called locations.

$$\begin{array}{ll} \text{minimize} & \sum_{i,j} \pi_{i,j} \cdot d_{i,j}^r \\ \text{(in } P' \text{ and } \pi) & \\ \text{subject to} & \sum_{j \in n_+} \pi\left(i,j \middle| i_-, n\right) = P\left(i \middle| i_-\right) \qquad (i_- \in \mathcal{N}_t,\, n \in \mathcal{N}'_t)\,, \\ & \sum_{i \in m_+} \pi\left(i,j \middle| m, j_-\right) = P'\left(j \middle| j_-\right) \qquad (j_- \in \mathcal{N}_t,\, m \in \mathcal{N}'_t)\,, \\ & \pi_{i,j} \geq 0,\ \sum_{i,j} \pi_{i,j} = 1 \text{ and} \\ & P'\left(j \middle| j_-\right) \geq 0. \end{array} \tag{18}$$

Unfortunately, substituting the conditional probabilities (12) now leads to constraints of the form

$$\sum_{i \in m_+} \pi_{i,j} = \pi_{m,j_-} \cdot P'\left(j \middle| j_-\right),$$

which are bilinear as both π and P' are decision variables. Problem (18) and its reformulation is therefore much more difficult to handle than (12). In fact, given the high number of decision variables and bilinear constraints, there is no hope for finding solutions for typical instances within reasonable time.

Recursive computation of the approximating probabilities

The computational difficulties of formulation (18) and the fact that the process distance can be calculated in a recursive way (see (15) and (16)) leads to the idea of calculating improved probabilities in a recursive way. In this way all constraints remain linear for each individual optimization problem, because they are formulated in terms of conditional probabilities.

Assume that π is feasible for given quantizers Q. Define

$$\mathsf{dl}_r(i,j) := d\left(\xi_i, q_j\right) \tag{19}$$

for $i \in \mathcal{N}_T$, $j \in \mathcal{N}'_T$. For $\mathsf{dl}_r(i,j)$ ($i \in \mathcal{N}_{t+1}$ and $j \in \mathcal{N}'_{t+1}$) given compute

$$\mathsf{dl}_r(m,n)^r := \sum_{i \in m_+, j \in n_+} \pi^*\left(i,j \middle| m,n\right) \cdot \mathsf{dl}_r\left(i,j\right)^r \qquad (m \in \mathcal{N}_t) \tag{20}$$

recursively for $m \in \mathcal{N}_t$, $n \in \mathcal{N}'_t$, where the conditional probabilities $\pi^*\left(\cdot,\cdot \middle| m,n\right)$ solve

$$\begin{array}{ll} \text{minimize} & \sum_{m \in \mathcal{N}_t} \tilde{\pi}\left(m,n\right) \cdot \sum_{i \in m_+, j \in n_+} \pi\left(i,j \middle| m,n\right) \cdot \mathsf{dl}_r\left(i,j\right)^r \\ \text{in } \pi\left(.,. \middle| m,n\right) & \\ \text{subject to} & \sum_{j \in n_+} \pi\left(i,j \middle| m,n\right) = P\left(i \middle| m\right) \qquad (i \in m_+), \\ & \sum_{i \in m_+} \pi\left(i,j \middle| m,n\right) = \sum_{i \in \tilde{m}_+} \pi\left(i,j \middle| \tilde{m},n\right) \qquad (j \in n_+ \text{ and } m,\, \tilde{m} \in \mathcal{N}_t), \\ & \tilde{\pi}\left(i,j \middle| m,n\right) \geq 0. \end{array} \tag{21}$$

The constraints

$$\sum_i \pi^*\left(i,n \middle| m_-,n_-\right) = \sum_i \pi^*\left(i,n \middle| \tilde{m}_-,n_-\right) \qquad (m,\, \tilde{m} \in \mathcal{N}_t) \tag{22}$$

for nodes m and $\tilde{m}$ at the same stage t in (21) ensure that

$$P'\left(j \middle| j_-\right) := \sum_i \pi^*\left(i,n \middle| m_-,n_-\right)$$

is well defined (as it is independent of m), allowing thus to reconstruct a measure P' by $P' = \sum_j \delta_{q_j} \cdot \sum_i \pi^*_{i,j}$.

Recomposing the transport plan π^* on the leaves $i \in \mathcal{N}_T$ and $j \in \mathcal{N}'_T$ by

$$\pi^*\left(i,j\right) = \pi^*\left(i_T, j_T \middle| i_{T-1}, j_{T-1}\right) \cdot \pi^*\left(i_{T-1}, j_{T-1} \middle| i_{T-2}, j_{T-2}\right) \cdot \ldots \pi^*\left(i_1, j_1 \middle| 0,0\right) \tag{23}$$

finally leads to improved probabilities, as the following theorem outlines.

Theorem 1. *Let P' be the measure related to the feasible transport probabilities π and P'^* be related to the probabilities π^* by*

$$P'^* := \sum_j \delta_{q_j} \cdot \sum_i \pi^*(i,j).$$

Then $\mathsf{dl}_r(\mathbb{P}, \mathbb{P}'^*) \leq \mathsf{dl}_r(\mathbb{P}, \mathbb{P}')$ and the improved distance is given by

$$\mathsf{dl}_r(\mathbb{P}, \mathbb{P}'^*) = \mathsf{dl}_r(0,0).$$

Proof. Observe that the measures π and π^* have the iterative decomposition

$$\begin{aligned}\pi(i,j) &= \pi(i_T, j_T) \\ &= \pi(i_T, j_T | i_{T-1}, j_{T-1}) \cdot \pi(i_{T-1}, j_{T-1} | i_{T-2}, j_{T-2}) \cdot \ldots \pi(i_1, j_1 | 0, 0)\end{aligned}$$

for all leafs $i \in \mathcal{N}_T$ and $i \in \mathcal{N}'_T$ (cf. [Dur04, Chapter 4, Theorem 1.6]). The terminal distance ($t = T$), given the entire history up to (i,j), is $\mathsf{dl}_{T;r}(i,j) := d(i,j)$, which serves as a starting value for the iterative procedure. To improve a given transport plan π the algorithm in (21) fixes the conditional probabilities $\pi(m,n)$ in an iterative step at stage t.

For

$$\sum_{i \in m_+} \sum_{j \in n_+} \pi^*(i,j|m,n) = \sum_{i \in m_+} P(i|m) = 1,$$

the constraints in (21) ensure that π^* again is a probability measure for each $m \in \mathcal{N}'_t$, and hence, by (23), π^* is a probability measure on $\mathcal{N}_T \times \mathcal{N}'_T$. Furthermore the constraints ensure that π^* respects the tree structures of both trees, that is, π^* is feasible for (14). Finally, due to the recursive construction, it holds that

$$\sum_{i,j} \pi^*_{i,j} d(i,j)^r = \mathsf{dl}_r(0,0)^r.$$

As the initial π is feasible as well for all equations in (21) it follows from the construction that

$$\mathsf{dl}_r(\mathbb{P}, \mathbb{P}'^*)^r = \mathsf{dl}_r(0,0)^r = \mathbb{E}_{\pi^*}(d^r) \leq \mathbb{E}_\pi(d^r).$$

As π was chosen arbitrarily with respective marginals it follows that

$$\mathsf{dl}_r(\mathbb{P}, \mathbb{P}'^*) \leq \mathsf{dl}_r(\mathbb{P}, \mathbb{P}'),$$

which shows that P'^* is an improved approximation of $\mathbb{P}$. □

3.2 Optimal scenarios: the problem of facility location

Consider the quantizers (or representative points)

$$Q = \{q_1, \ldots q_n\},$$

where each $q_j = (q_{j,0}, \ldots q_{j,T})$ is a path in the tree. Given a fixed, feasible measure π define the function

$$D_\pi(\{q_1, \ldots q_n\})^r := \mathbb{E}_\pi(d^r) = \sum_{i,j} \pi_{i,j} d(\xi_i, q_j)^r. \tag{24}$$

The problem of finding optimal quantizers then consists in solving the minimization problem

$$\min_{q_1, \ldots q_n} D_\pi(\{q_1, \ldots q_n\}). \tag{25}$$

In general it is difficult to solve the facility location problem (25), which is nonlinear and nonconvex, with many local minima. However, in an iterative procedure as proposed in Section 3.3 below, a few steps of significant descent in each iteration will be sufficient to considerably improve the approximation.

In many applications the gradient of function (24) is available as an analytic expression, for example if $d\left(\xi_i, \xi'_j\right) = \left(\sum_t d_t\left(\xi_i, \xi'_j\right)^p\right)^{1/p}$. In this situation the derivative of $D_\pi\left(\{q_1, \dots q_n\}\right)^r$ can be evaluated by using the chain rule, which leads to

$$\nabla_{\xi'_{j,t}} D\left(\xi'\right) = D_\pi\left(\xi'\right)^{1-r} \cdot \sum_i \pi_{i,j} d\left(\xi_i, \xi'_j\right)^{r-p} \cdot d_t\left(\xi_{i,t}, \xi'_{j,t}\right)^{p-1} \cdot \nabla_{\xi'_{j,t}} d_t\left(\xi_{i,t}, \xi'_{j,t}\right) \quad \left(j \in \mathcal{N}_t\right).$$

If in addition the metric at stage t is an ℓ^s-norm, $d_t\left(\xi_{i,t}, \xi'_{j,t}\right) = \left\|\xi_{i,t} - \xi'_{j,t}\right\|_s$, then it holds that

$$\nabla_{\xi'_{j,t}} d_t\left(\xi_{i,t}, \xi'_{j,t}\right) = d_t\left(\xi_{i,t}, \xi'_{j,t}\right)^{1-s} \cdot \left|\xi_{i,t} - \xi'_{j,t}\right|^{s-2} \cdot \left(\xi_{i,t} - \xi'_{j,t}\right),$$

which is obtained by direct computation.

To compute the minimum in (25) a few steps by the steepest descent method will ensure some successive improvements. Other possible methods are the nonlinear conjugate gradient method (cf. Ruszczyński [Rus06]) or the limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, cf. Nocedal [Noc80].

In the special case of the quadratic process distance the facility location problem can be accomplished by explicit evaluations.

Theorem 2. *For a quadratic process distance (Euclidean norm and $r = 2$) the scenarios*

$$q_t\left(n_t\right) := \sum_{m \in \mathcal{N}_t} \frac{\pi\left(m, n_t\right)}{\sum_{i \in \mathcal{N}_t} \pi\left(i, n_t\right)} \cdot \xi_t\left(m\right)$$

are the best possible choice to solve the facility location problem (25) (cf. (26) in Algorithm 1).

Proof. The explicit decomposition of the process distance allows the rearrangement

$$\begin{aligned}
\mathsf{dl}_2\left(\mathbb{P}, \mathbb{P}'\right)^2 &= \sum_{i,j} \pi_{i,j} \cdot d\left(\xi_i, q_j\right)^2 \\
&= \sum_{i,j} \pi_{i,j} \sum_{t=0}^{T} w_t \cdot \left\|\xi_{i_t} - q_{j_t}\right\|_2^2 \\
&= \sum_{t=0}^{T} w_t \cdot \sum_{n_t \in \mathcal{N}'_t} \left(\sum_{m_t \in \mathcal{N}_t} \pi\left(m_t, n_t\right) \cdot \left\|\xi\left(m_t\right) - q_t\left(n_t\right)\right\|_2^2 \right).
\end{aligned}$$

As the conditional expectation minimizes this inner expression (cf. also the proof of Theorem 4 in Appendix A for the corresponding situation for the Wasserstein distance) the assertion follows for every $n_t \in \mathcal{N}_t$ by considering and minimizing every map

$$q \mapsto \sum_{m_t \in \mathcal{N}_t} \pi\left(m_t, n_t\right) \cdot \left\|\xi\left(m_t\right) - q\right\|_2^2$$

separately. □

We summarize the individual steps in Algorithm 1, Step 2. For the quadratic nested distance this resulting procedure clearly is fast in implementations.

3.3 The overall algorithm

The following Algorithm 1 describes the course of action for iterative improvements of the approximation: starting with an initial guess for the quantizers (the scenario paths, resp.) and using the related transport probabilities π^0 the algorithm iterates between improving the quantizers (step 2) and improving the transport probabilities (step 3). Step 2 goes backward in time and uses conditional

Algorithm 1
Sequential improvement of the measure P^k to approximate $P = \sum_i p_i \delta_{\xi_i}$ in the process distance on the trees $(\mathcal{F}_t)_{t\in\{0,\dots T\}}$ ($(\mathcal{F}'_t)_{t\in\{0,\dots T\}}$, resp.).

Step 1—Initialization
Set $k \leftarrow 0$, and let q^0 be process quantizers with related transport probabilities $\pi^0(i,j)$ between scenario i of the original $\mathbb{P}$-tree and scenario q_j^0 of the approximating $\mathbb{P}'$-tree; $\mathbb{P}^0 := \mathbb{P}'$.

Step 2—Improve the quantizers
Find improved quantizers q_j^{k+1}:

- In case of the quadratic Wasserstein distance (Euclidean distance and Wasserstein of order $r = 2$) set
$$q^{k+1}(n_t) := \sum_{m\in\mathcal{N}_t} \frac{\pi(m,n_t)}{\sum_{i\in\mathcal{N}_t}\pi(i,n_t)} \cdot \xi_t(m), \tag{26}$$
- or solve (25), for example by applying the steepest descent method, conjugate gradient methods, or the limited memory BFGS method.

Step 3—Improve the probabilities
Setting $\tilde{\pi} \leftarrow \pi^k$ and $q \leftarrow q^{k+1}$ use (19), (20), (21) and (23) to calculate all conditional probabilities $\pi^{k+1}(\cdot,\cdot|m,n) = \pi^*(\cdot,\cdot|m,n)$, the unconditional transport probabilities $\pi^{k+1}(\cdot,\cdot)$ and the distance $\mathsf{dl}_r^{k+1}(0,0) = \mathsf{dl}_r(0,0)$.

Step 4
Set $k \leftarrow k+1$ and continue with **Step 2** if
$$\mathsf{dl}_r^{k+1}(0,0) < \mathsf{dl}_r^{k}(0,0) - \varepsilon,$$
where $\varepsilon > 0$ is the desired improvement in each cycle k.
Otherwise, set $q^* \leftarrow q^k$, define the measure
$$P^{k+1} := \sum_j \delta_{q_j^{k+1}} \cdot \sum_i \pi^{k+1}(i,j),$$
for which $\mathsf{dl}_r\left(\mathbb{P},\mathbb{P}^{k+1}\right) = \mathsf{dl}_r^{k+1}(0,0)$ and stop.
In case of the quadratic process distance ($r = 2$) and the Euclidean distance the choice $\varepsilon = 0$ is possible.

versions $\mathsf{dl}_r^{k+1}(m,n)$ of the process distance, which are related to nodes m and n, in order to resemble an approximation of the full process distance. To improve the locations q, step 3 either uses classical optimization algorithms for the general case or a version of the k-means algorithm in the important case of the quadratic process distance.

The algorithm leads to an improvement in each iteration step (Theorem 1 and Theorem 2) and converges in finitely many steps.

Theorem 3. *Provided that the minimization (25) can be done exactly—as is the case for the quadratic process distance—Algorithm 1 terminates at the optimal distance $\mathsf{dl}_r\left(P,P^{k^*}\right)$ after finitely many iterations (k^*, say).*

Proof. It is possible—although very inadvisable for computational purposes—to rewrite the computa-

Stages	4	5	5	6	* 7	7
Nodes of the initial tree	53	309	188	1.365	1.093	2.426
Nodes of the approximating tree	15	15	31	63	127	127
Time/ sec.	1	10	4	160	157	1.044

Table 1: Time to perform an iteration in Algorithm 1.
The example indicated by the asterisk (*) corresponds to Figure 6.

tion of $\mathsf{dl}_r^{k+1}(0,0)$ in Algorithm 1 as a single linear program of the form

$$\begin{array}{rl} \text{minimize} \\ \text{in } \pi^{k+1} & c\left(\pi^{k+1}\middle|\,\pi^k\right) \\ \text{subject to} & A\pi^{k+1} = b, \\ & \pi^{k+1} \geq 0, \end{array}$$

where the matrix A and the vector b collect all linear conditions from (21), and $\pi \mapsto c\left(\pi\middle|\,\tilde{\pi}\right)$ is multilinear. Note that the constraints $\Pi := \{\pi\colon A\pi = b,\, \pi \geq 0\}$ form a convex polytope, which is independent of the iterate π^k . Without loss of generality one may assume that π^k is an edge of the polytope Π. Because Π has finitely many edges and each edge $\pi \in \Pi$ can be associated with a unique quantization scenario $q(\pi)$, by assumption it is clear that the decreasing sequence

$$\mathsf{dl}_r^{k+2}\left(\mathbb{P}, \mathbb{P}^{k+2}\right) = c\left(\pi^{k+2}\middle|\,\pi^{k+1}\right) \leq c\left(\pi^{k+1}\middle|\,\pi^k\right) = \mathsf{dl}_r^{k+1}\left(\mathbb{P}, \mathbb{P}^{k+1}\right)$$

cannot improve further whenever the optimal distance is met. □

The same statement as for the Wasserstein distance holds true here for the process distance: for other distances than the quadratic ones, P^k can be used as a starting point but in general *is not* even a local minimum.

An initial guess for the approximating tree can be obtained from any other tree reduction or generation method, or even from a random generation within the range of observed values. On the other hand, it should be kept in mind that the recursive calculations lead only to local optima. Therefore some sensitivity of the results with respect to the reduced tree chosen at the beginning is natural. Experience from calculations show that in most cases one can trust on stability. In rare situations the pure method described above may truncate branches, resulting in a probability of zero for whole subtrees, which definitely is an unfavorable local minimum. This effect is easily resolved by ensuring that all probabilities $\pi^k(m,n)$ are larger than a small number $\varepsilon > 0$ by setting $\pi'(m,n) = \max\left\{\pi^{k+1}(m,n), \varepsilon_k\right\}$ and redefining $\pi^{k+1} = \frac{\pi^{k+1}(m,n)}{\sum \pi^{k+1}(m,n)}$ in step 3 of the overall algorithm.

3.4 Selected numerical examples and derived applications

To illustrate the results of Algorithm 1 we have implemented all steps of the discussed algorithms in MATLAB®. All linear programs were solved using the function linprog. It is a central observation that optimization for Euclidean norms and the quadratic Wasserstein distance is fastest. This is because the facility location problem can be avoided and replaced by computing the conditional expectation in a direct way. Moreover, when applying the methods, it was a repeated pattern that the first few iteration steps improve the distance significantly, whereas following steps just give minor improvements of the objective. The following results were calculated with the process distance based on the Euclidean distance and $r = 2$.

Computation times for an iteration step in Algorithm 1 for varying tree structures are collected in Table 1 (on a customary, standard laptop).

Consistency. It is desirable that Algorithm 1 will reproduce the initial tree, if started with a shifted version of the initial tree, where the probabilities and states are changed, but the tree topology, i.e.,

the branching structure, is unchanged. Algorithm 1 indeed reproduces the initial tree for many of our test cases.

Example 3. As a variant of this type of consistency consider Figure 5. The first tree is an approximation of a Gaussian walk. It is constructed by replacing the (conditional) normal distribution at every node by the best d_2 approximation with 4 supporting points and adapted weights (cf. Graf and Luschgy [GL00, Table 5.1]).

To demonstrate the strength of the algorithm we start with a randomly generated, distorted binary tree (Figure 5b, left), which has very different states and probabilities. This initial tree has a distance of 6.7 to the original tree. The first tree the algorithm produces is much closer, it has a distance of 2.32. This result is already very close to the further tree depicted in Figure 5b (right), which is the result after 5 iterations. The resulting binary tree has a nested distance of 2.24 to the initial tree and is evidently a useful approximation to the full tree with four branches.

Example 4. Figure 6 exemplary depicts the situation of a more complex instance (no i.i.d. increments, more branches and irregular branching) with 5 stages. The initial approximating tree again was chosen as a binary tree (bottom, left) constructed simply by removing all branches except those two, which have highest probability (alternatively, starting trees can be constructed using the procedures presented in the papers [HR03], [HR07] and [HR09b] by Heitsch and Römisch). The starting tree is at a process distance of about 2.1. Algorithm 1 then produces the new tree (bottom, right) with process distance 0.42 to the original tree.

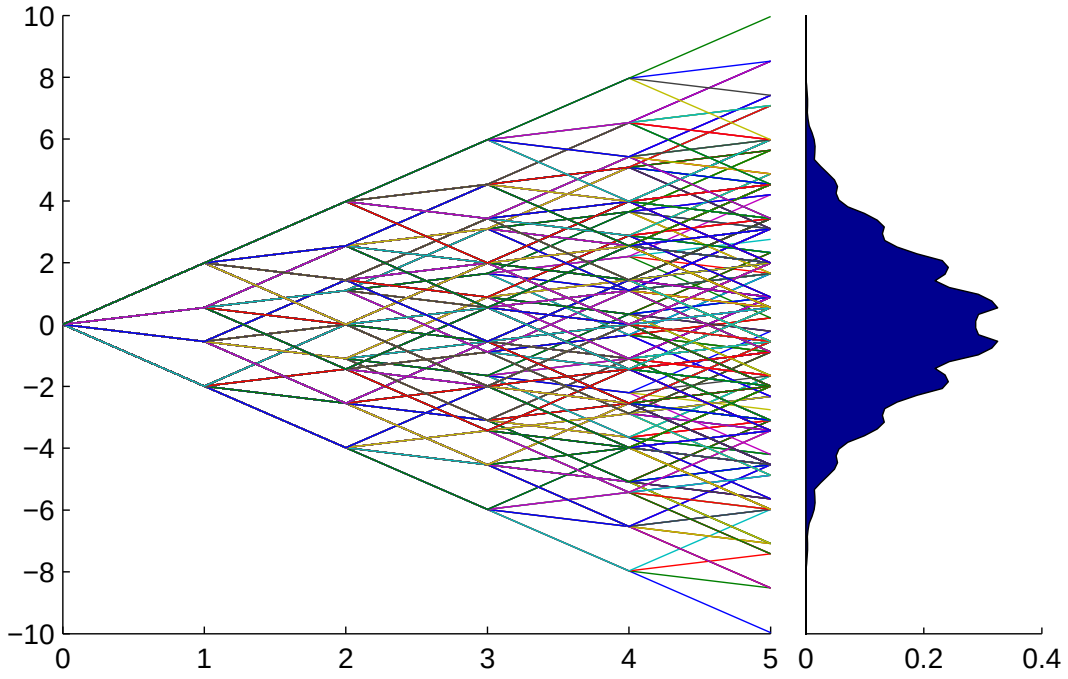
Reduction of variance and tail behavior. The final distribution of the resulting tree in Figure 5b and Figure 6 display a tighter support than the end distribution of the original tree. Graf and Luschgy [GL00, Remark 4.6] elaborate on this effect, where it is explained that the best approximation of a distribution in the Wasserstein distance reduces the variance (particularly for $r = 2$ and the Euclidean distance. The best approximation (7) has even variance 0). A reduced variance is an essential characteristic for approximations in the Wasserstein distance.

The variance reducing property is notably not in contrast to the statement (5) of the Kantorovich–Rubinstein Theorem, as the function $x \mapsto x^2$ is not Lipschitz continuous. Notice as well that Lipschitz continuity depends on the distance function chosen on the underlying space. Hence the class of Lipschitz functions can be extended by adapting the distance function, as is achieved, e.g., by the Fortet–Mourier distance (cf. [Röm03]).

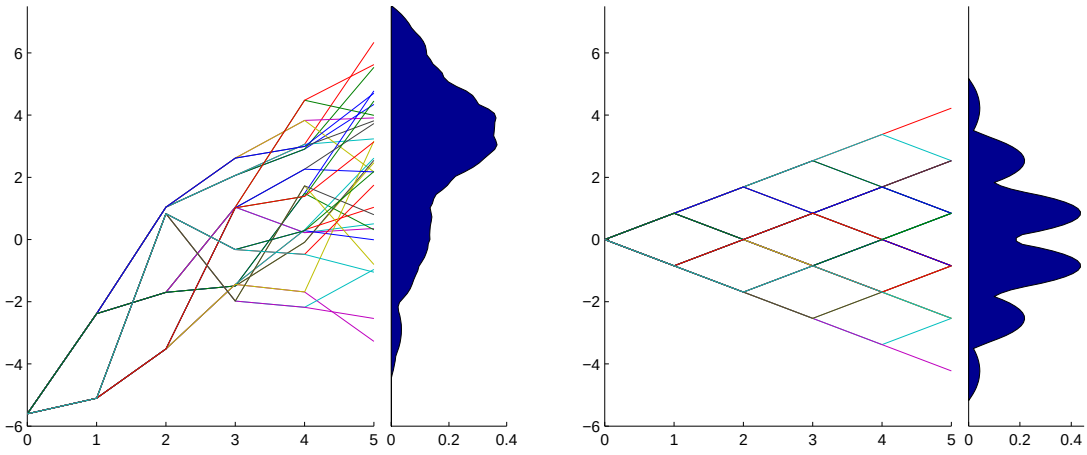
Discrete probability measures cannot replicate the behavior of probabilities with a density in their tails, which is important when considering risk. Important risk measures, however, are continuous in Wasserstein distance, such that the expectation in (5) can be replaced by risk measures in various situations (cf. Pichler [Pic13]). In these situations tree methods remain candidates to mResearch Council of Norway (grant 207690/ E20)odel the problem.

Limitations. The method is designed to improve the distance in any case, in this sense it is not a heuristic (cf. e.g. [HR09b, HR09a]). However, while accounting for the full tree structure enhances the approximation, it also leads to substantially higher complexity. The algorithm basically has the same limitations as the computation of the nested, or process distance itself. Its computational complexity, i.e. the number of variables and constraints, is of the same order for both problems. In addition, while the calculating the distance is linear, the approximation problem is nonlinear, in fact even non-convex. For the quadratic nested distance it is the first step of the algorithm - improving the probabilities - which is computationally expensive, while it is comparably cheap to improve the states in the second step.

The present paper concentrates on the basic theoretical properties of the algorithm, hence the presented examples and the underlying implementation in MATLAB® should be understood as purely illustrative. Developing the implementation further, clearly would involve the development of more efficient code in a lower level programming language and usage of faster LP solvers. Parallelization of the problem is possible and also will increase the calculation speed. Furthermore, we assume that solving the dual problem instead of the primal to find the probabilities is faster as well, although this is



(a) The initial tree process is an approximation of a Gaussian walk in 5 stages. Annotated is a density plot of its final probabilities



(b) Starting tree (left), and the resulting tree after 5 iterations (right)

Figure 5: Approximating a Gaussian walk by a binary tree (cf. Example 3)

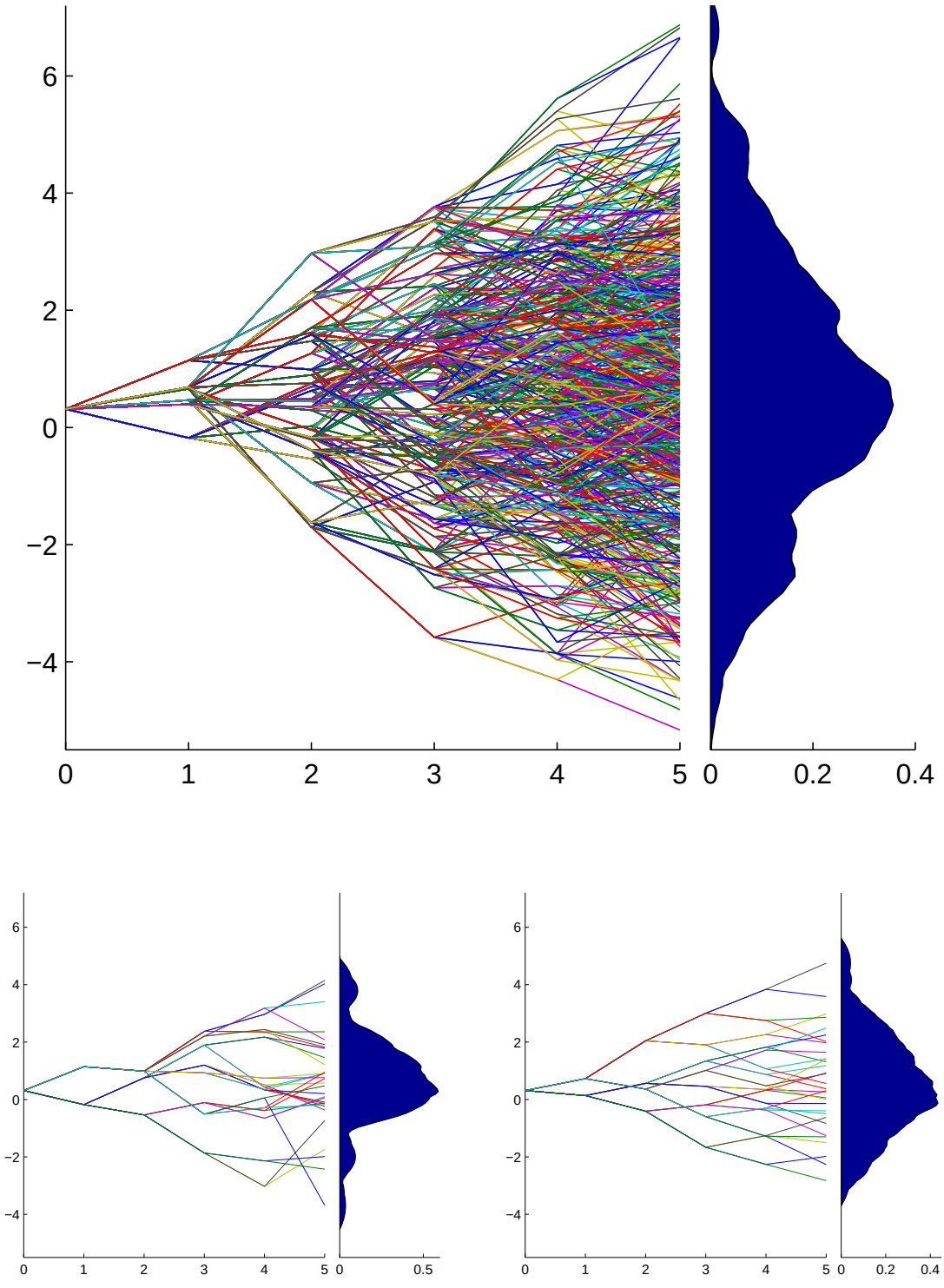


Figure 6: The initial tree has 1093 nodes at 5 stages (top).
The approximating binary tree (left, 127 nodes) is obtained by cutting branches with smallest probabilities. The tree at the right is obtained after 5 iterations, its process distance to the larger tree is 0.42.

a conjecture at this time. Solving the dual approximately, however, would just provide a lower bound (evidently, more useful are upper bound).

4 Summary

This paper addresses the problem of approximating stochastic processes in discrete time by trees, which are discrete stochastic processes. For this purpose we build on the recently introduced process or nested distance, generalizing the well known Wasserstein or Kantorovich distance to stochastic processes. This distance takes notice of the effects caused by filtrations related to stochastic processes.

We adapt this process distance to compare trees, which are important tools for discretizing stochastic optimization problems. The aim is to reduce the distance between a given tree and a smaller tree where both, the probabilities and the states are subject to changes. This problem is of fundamental interest in stochastic programming, as the number of variables of the initial process can be reduced significantly by the techniques and algorithms proposed.

The paper analyzes the relations between processes and trees, highlights the essential properties of Wasserstein distances and process distances and finally proposes and analyzes an iterative algorithm for improving the process distance between trees. For the important special case of process distances of order 2 (based on Euclidean distances) the algorithm can be enhanced by using k-means clustering in order to improve calculation speed.

5 Acknowledgment

We thank the referees for their constructive criticism. We wish to thank two anonymous referees for their dedication to review the paper. Their valuable comments significantly improved the content and the presentation.

Parts of this paper are addressed in the book *Multistage Stochastic Optimization* (Springer) by Pflug and Pichler, which also summarizes many more topics in multistage stochastic optimization and which had to be completed before final acceptance of this paper.

6 Compliance with Ethical Standards

6.1 Disclosure of potential conflicts of interest

Funding: This research was partially funded by the Austrian science fund FWF, project P 24125-N13 and by the Research Council of Norway, grant 207690/ E20.

References

[BGMS09] Mathias Beiglböck, Martin Goldstern, Gabriel Maresch, and Walter Schachermayer. Optimal and better transport plans. *Journal of Functional Analysis*, 256(6):1907–1927, 2009. 3

[BLS12] Mathias Beiglböck, Christian Léonard, and Walter Schachermayer. A general duality theorem for the Monge-Kantorovich transport problem. *Studia Mathematica*, 209:151–167, 2012. 3

[BPP05] Vlad Bally, Gilles Pagès, and Jacques Printems. A quantization tree method for pricing and hedging multidimensional american options. *Mathematical Finance*, 15(1):119–168, 2005. 23

[DGKR03] Jitka Dupačová, Nicole Gröwe-Kuska, and Werner Römisch. Scenario reduction in stochastic programming. *Mathematical Programming, Ser. A*, 95(3):493–511, 2003. 2, 3, 22

[DH02] Z. Drezner and H. W. Hamacher. *Facility Location: Applications and Theory*. Springer, New York, NY, 2002. 23

[Dud69] R. M. Dudley. The speed of mean Glivenko-Cantelli convergence. *The Annals of Mathematical Statistics*, 40(1):40–50, 1969. 4

[Dur04] Richard A. Durrett. *Probability. Theory and Examples*. Duxbury Press, Belmont, CA, second edition, 2004. 12

[GL00] Siegfried Graf and Harald Luschgy. *Foundations of Quantization for Probability Distributions*, volume 1730 of *Lecture Notes in Mathematics*. Springer-Verlag Berlin Heidelberg, 2000. 2, 4, 16, 23

[HR03] Holger Heitsch and Werner Römisch. Scenario reduction algorithms in stochastic programming. *Comput. Optim. Appl. Stochastic Programming*, 24(2-3):187–206, 2003. 16

[HR07] Holger Heitsch and Werner Römisch. A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, 6:731–738, 2007. 16

[HR09a] Holger Heitsch and Werner Römisch. Scenario tree modeling for multistage stochastic programs. *Math. Program. Ser. A*, 118:371–406, 2009. 2, 16

[HR09b] Holger Heitsch and Werner Römisch. Scenario tree reduction for multistage stochastic programs. *Computational Management Science*, 2:117–133, 2009. 2, 16

[HR11] Holger Heitsch and Werner Römisch. Stability and scenario trees for multistage stochastic programs. In Gerd Infanger, editor, *Stochastic Programming*, volume 150 of *International Series in Operations Research & Management Science*, pages 139–164. Springer New York, 2011. 2

[HRS06] Holger Heitsch, Werner Römisch, and Cyrille Strugarek. Stability of multistage stochastic programs. *SIAM J. Optimization*, 17(2):511–525, 2006. 6

[HW01] Kjetil Høyland and Stein William Wallace. Generating scenario trees for multistage decision problems. *Management Science*, 47:295–307, 2001. 1

[KW13] Alan J. King and Stein W. Wallace. *Modeling with Stochastic Programming*, volume XVI of *Springer Series in Operations Research and Financial Engineering*. Springer, 2013. 1

[Llo82] Stuart P. Lloyd. Least square quantization in PCM. *IEEE Transactions of Information Theory*, 28(2):129–137, 1982. 23

[Noc80] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. 13

[Pfl09] Georg Ch. Pflug. Version-independence and nested distribution in multistage stochastic optimization. *SIAM Journal on Optimization*, 20:1406–1420, 2009. 2, 6

[Pic13] Alois Pichler. Evaluations of risk measures for different probability measures. *SIAM Journal on Optimization*, 23(1):530–551, 2013. 16

[PP12] Georg Ch. Pflug and Alois Pichler. A distance for multistage stochastic optimization models. *SIAM Journal on Optimization*, 22(1):1–23, 2012. 2, 6, 7, 9

[PR07] Georg Ch. Pflug and Werner Römisch. *Modeling, Measuring and Managing Risk*. World Scientific, River Edge, NJ, 2007. 7, 8

[Rac91] Svetlozar T. Rachev. *Probability metrics and the stability of stochastic models*. John Wiley and Sons Ltd., West Sussex PO19, 1UD, England, 1991. 3

[Röm03] Werner Römisch. Stability of stochastic programming problems. In Andrzej Ruszczyński and Alexander Shapiro, editors, *Stochastic Programming, Handbooks in Operations Research and Management Science*, volume 10, chapter 8. Elsevier, Amsterdam, 2003. 16

[RR98] Svetlozar T. Rachev and Ludger Rüschendorf. *Mass Transportation Problems Vol. I: Theory, Vol. II: Applications*, volume XXV of *Probability and its applications*. Springer, New York, 1998. 3

[Rus06] Andrzej Ruszczyński. *Nonlinear Optimization*. Princeton University Press, 2006. 13

[Sha10] Alexander Shapiro. Computational complexity of stochastic programming: Monte Carlo sampling approach. In *Proceedings of the International Congress of Mathematicians*, pages 2979–2995, Hyderabad, India, 2010. 2

[Shi96] Albert Nikolayevich Shiryaev. *Probability*. Springer, New York, 1996. 26

[SN05] Alexander Shapiro and Arkadi Nemirovski. On complexity of stochastic programming problems. In V. Jeyakumar and A.M. Rubinov, editors, *Continuous Optimization: Current Trends and Applications*, pages 111–144. Springer, 2005. 2

[ST09] Walter Schachermayer and Josef Teichmann. Characterization of optimal transport plans for the monge-kantorovich problem. *Proceedings of the A.M.S.*, 137(2):519–529, 2009. 3

[Ver06] Anatoly M. Vershik. Kantorovich metric: Initial history and little-known applications. *Journal of Mathematical Sciences*, 133(4):1410–1417, 2006. 3

[Vil03] Cédric Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2003. 3

[Vil09] Cédric Villani. *Optimal transport, old and new*, volume 338 of *Grundlehren der Mathematischen Wissenschaften*. Springer, Berlin, 2009. 3

[Wil91] David Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, 1991. 22

A Scenario approximation with Wasserstein distances

Given a probability measure P we ask for an approximating probability measure, which is located on Ξ', that is to say its support is contained in Ξ'. The following proposition reveals that the push-forward measure $P^{\mathbf{T}}$, where the mapping $\mathbf{T}$ is defined in (ii) of the following proposition, is the best approximation of P located just on Ξ', i.e., $P^{\mathbf{T}}$ satisfies

$$\mathsf{d}_r\left(P,\, P^{\mathbf{T}}\right) \leq \mathsf{d}_r\left(P, P'\right) \qquad \left(P'\left(\Xi'\right) = 1\right). \tag{27}$$

Proposition 1 (Lower bounds and best approximation)**.** *Let P and P' be probability measures.*

(i) The Wasserstein distance has the lower bound

$$\int_{\Xi} \min_{\xi' \in \Xi'} d\left(\xi, \xi'\right)^r P\left(\mathrm{d}\xi\right) \leq \mathsf{d}_r\left(P,\, P'\right)^r. \tag{28}$$

(ii) The lower bound in (28) is attained for the pushforward measure $P^{\mathbf{T}} := P \circ \mathbf{T}^{-1}$ on Ξ', where the transport map $\mathbf{T} : \Xi \to \Xi'$ is defined by[3]

$$\mathbf{T}\left(\xi\right) \in \operatorname*{argmin}_{\xi' \in \Xi'} d\left(\xi, \xi'\right).$$

[3]The selection has to be chosen in a measurable way.

It holds that[4]

$$\mathsf{d}_r\left(P,\, P^{\mathbf{T}}\right)^r = \int \min_{\xi' \in \Xi'} d\left(\xi, \xi'\right)^r P\left(\mathrm{d}\xi\right) = \mathbb{E}\left[d\left(\mathrm{id}_\Xi, \mathbf{T}\left(\mathrm{id}_\Xi\right)\right)^r\right],$$

where the identity $\mathrm{id}_\Xi\left(\xi\right) = \xi$ *on* Ξ *is employed for notational convenience.*

(iii) If $\Xi = \Xi'$ *is a vector space and* $\mathbf{T}$ *as in (ii), then*

$$\mathsf{d}_r\left(P,\, P^{\tilde{\mathbf{T}}}\right) \leq \mathsf{d}_r\left(P,\, P^{\mathbf{T}}\right),$$

where $\tilde{\mathbf{T}}$ *is defined by* $\tilde{\mathbf{T}}\left(\xi\right) := \mathbb{E}_P\left[\tilde{\xi} \,\middle|\, \mathbf{T}\left(\tilde{\xi}\right) = \mathbf{T}\left(\xi\right)\right]$.

Proof. Let π have the marginals of P and P'. Then

$$\begin{aligned}\int_{\Xi\times\Xi'} d\left(\xi,\xi'\right)^r \pi\left(\mathrm{d}\xi, \mathrm{d}\xi'\right) &\geq \int_\Xi \int_{\Xi'} \min_{\xi'\in\Xi'} d\left(\xi,\xi'\right)^r \pi\left(\mathrm{d}\xi, \mathrm{d}\xi'\right) \\ &= \int_\Xi \min_{\xi'\in\Xi'} d\left(\xi,\xi'\right)^r P\left(\mathrm{d}\xi\right).\end{aligned}$$

Taking the infimum over π reveals the lower bound (28).

Define the transport plan $\pi := P \circ \left(\mathrm{id}_\Xi \times \mathbf{T}\right)^{-1}$ by employing the transport map $\mathbf{T}$. Then

$$\pi\left(A\times B\right) = P\left(\left\{\xi\colon\ \left(\xi, \mathbf{T}\left(\xi\right)\right) \in A\times B\right\}\right) = P\left(\left\{\xi\colon\ \xi\in A \text{ and } \mathbf{T}\left(\xi\right)\in B\right\}\right).$$

π is feasible, it has the marginals $\pi\left(A\times\Xi'\right) = P\left(\left\{\xi\colon\, \xi\in A,\, \mathbf{T}\left(\xi\right)\in\Xi'\right\}\right) = P\left(A\right)$ and $\pi\left(\Xi\times B\right) = P\left(\left\{\xi\colon\, \mathbf{T}\left(\xi\right)\in B\right\}\right) = P^{\mathbf{T}}\left(B\right)$. For this measure π thus

$$\iint_{\Xi\times\Xi'} d\left(\xi,\xi'\right)^r \pi\left(\mathrm{d}\xi,\mathrm{d}\xi'\right) = \int_\Xi d\left(\xi, \mathbf{T}\left(\xi\right)\right)^r P\left(\mathrm{d}\xi\right) = \int_\Xi \min_{\xi'\in\Xi'} d\left(\xi,\xi'\right)^r P\left(\mathrm{d}\xi\right),$$

which proves (ii).

For the last assertion apply the conditional Jensen's inequality (cf., e.g., Williams [Wil91]) $\varphi\left(\mathbb{E}\left(X\middle|\mathbf{T}\right)\right) \leq \mathbb{E}\left(\varphi\left(X\right)\middle|\mathbf{T}\right)$ to the convex mapping $\varphi\colon\, y\mapsto d\left(\xi,y\right)^r$ and obtain

$$d\left(\xi,\, \mathbb{E}\left(\mathrm{id}\middle|\mathbf{T}\right)\circ\mathbf{T}\right) \leq \mathbb{E}\left(d\left(\xi,\mathrm{id}\right)\middle|\mathbf{T}\right)\circ\mathbf{T}.$$

The measure $\tilde{\pi}\left(A\times B\right) := P\left(A\cap\tilde{\mathbf{T}}^{-1}\left(B\right)\right)$ has marginals P and $P^{\tilde{\mathbf{T}}}$, from which follows that

$$\begin{aligned}\mathsf{d}_r\left(P, P^{\tilde{\mathbf{T}}}\right)^r &\leq \int d\left(\xi, \tilde{\mathbf{T}}\left(\xi\right)\right)^r P\left(\mathrm{d}\xi\right) = \int d\left(\xi,\, \mathbb{E}\left(\mathrm{id}\middle|\mathbf{T}\right)\circ\mathbf{T}\left(\xi\right)\right)^r P\left(\mathrm{d}\xi\right) \\ &\leq \int \mathbb{E}\left(d\left(\xi,\mathrm{id}\right)^r\middle|\mathbf{T}\right)\left(\mathbf{T}\left(\xi\right)\right) P\left(\mathrm{d}\xi\right) = \int d\left(\xi,\mathbf{T}\left(\xi\right)\right)^r P\left(\mathrm{d}\xi\right) = \mathsf{d}_r\left(P,P^{\mathbf{T}}\right)^r,\end{aligned}$$

which is the assertion. □

It was addressed in the introduction that the approximation can be improved by relocating the scenarios themselves, and by allocating adapted probabilities to these scenarios. The following two sections address these issues by applying the previous Proposition 1.

[4] see also Dupačová et al. [DGKR03, Theorem 2].

Optimal probabilities

The optimal measure $P^{\mathbf{T}}$ in Proposition 1 notably does *not* depend on the order r. Moreover, given a probability measure P, Proposition 1 (ii) allows to find the best approximation, which is located just on finitely many points $Q = \{q_1 \dots q_n\}$. The points $q_j \in Q$ are often called *quantizers,* and we adopt this notion in what follows (see the œuvre of Gilles Pagès, e.g., [BPP05] for a comprehensive treatment).

Consider now $\Xi' := Q$, define $p_j^* := P(\mathbf{T} = q_j)$, then the collection of distinct sets $\{\mathbf{T} = q_j\}$ is a tessellation of Ξ (a Voronoi tessellation, see Graf and Luschgy [GL00]) and set $P^Q := P^{\mathbf{T}} = \sum_j p_j^* \cdot \delta_{q_j}$, as above. Then $\mathsf{d}_r\left(P, P^Q\right)^r = \int \min_{q \in Q} d(\xi, q)^r \, P(\mathrm{d}\xi)$, and no better approximation is possible by Proposition 1.

According to Proposition 1 the best approximating measure for $P = \sum_i p_i \delta_{\xi_i}$, which is located on Q, is given by $P^Q = \sum_j p_j^* \delta_{q_j}$. For a discrete measure this can be formulated by a linear program as

$$\begin{array}{ll} \text{minimize} \\ \text{(in } \pi\text{)} & \sum_{i,j} d_{i,j}^r \pi_{i,j} \\ \text{subject to} & \sum_j \pi_{i,j} = p_i, \\ & \pi_{i,j} \geq 0, \end{array}$$

which is solved by the optimal transport plan

$$\pi_{i,j}^* := \begin{cases} p_i & \text{if } d(\xi_i, q_j) = \min_{q \in Q} d(\xi_i, q) \\ 0 & \text{else,} \end{cases} \tag{29}$$

such that

$$p_j^* = \sum_i \pi_{i,j}^* \qquad \text{and} \qquad \mathsf{d}_r\left(P, P^Q\right)^r = \mathbb{E}_{\pi^*}\left(d^r\right). \tag{30}$$

Observe as well that the matrix π^* in (29) has just $|\Xi|$ non-zero entries, as in every row i of π^* there is just one non-zero entry $\pi_{i,j}^*$. This is a simplification in comparison with Remark 2, as the solution π of (4) has $|\Xi| + |\Xi'| - 1$ non-zero entries, if the probability measure P' is specified.

Finally, given the support points Q, it is an easy exercise to look up the closest points according to (29), and sum up their probabilities according (30), such that the solution of (27), the closest measure to P located on Q, is immediately obtained by $P^Q = \sum_j p_j^* \delta_{q_j}$.

Optimal supporting points—facility location

Given the previous results on optimal probabilities the problem of finding a sufficiently good approximation of P in the Wasserstein is reduced to the problem of finding good locations Q, that is to minimize the function

$$\begin{aligned} \{q_1, \dots q_n\} \mapsto \mathsf{d}_r\left(P, P_{\{q_1, \dots q_n\}}\right)^r &= \int \min_{q \in \{q_1, \dots q_n\}} d(\xi, q)^r \, P(\mathrm{d}\xi) \\ &= \mathbb{E}_\xi \left[\min_{q \in \{q_1, \dots q_n\}} d(\xi, q)^r \right]. \end{aligned} \tag{31}$$

Minimizing (31) with respect to the quantizers $\{q_1, \dots q_n\}$ is often referred to as *facility location,* as in Drezner and Hamacher [DH02]. This problem is not convex, and no closed form solution exists in general, it hence has to be handled with adequate numerical algorithms. Moreover, it is well known that the facility location problems are is NP-hard.

For the important case of the quadratic Wasserstein distance, Proposition 1 (iii) and its proof give rise for an adaption of the k-means clustering algorithm (also referred to as Lloyd's algorithm, cf. [Llo82]), which is described in Algorithm 2. In this case the conditional average is the best approximation in terms of the Euclidean norm, such that the algorithm terminates after finitely many iterations at a local minimum.

Algorithm 2
Facility location for $P = \sum_i p_i \delta_{\xi_i}$ in the special case of the Euclidean distance and quadratic Wasserstein distance (order $r = 2$).

Initialization ($k = 0$):
Choose n points $Q^0 := \left\{q_i^0 \colon i = 1, \dots n\right\}$, for example by randomly picking n distinct points from $\{\xi_i \colon i\}$.

Assignment Step:
In each step k assign each ξ_i to the cluster with the closest mean,

$$T_j^k := \left\{\xi_i \colon \ \left\|\xi_i - q_j^k\right\| \leq \left\|\xi_i - q_{j'}^k\right\| \ \text{ for all } q_{j'}^k \in Q^k\right\}$$

for all $j = 1, \dots n$, and set

$$P^k(\cdot) := \sum_{j=1}^{n} P\left(T_j^k\right) \delta_{q_j^k}(\cdot).$$

Update Step:
Set $Q^{k+1} := \left\{q_j^{k+1} \colon \ j = 1, \dots n\right\}$, where

$$q_j^{k+1} := \sum_{\xi_i \in T_j^k} \frac{P\left(\xi_i\right)}{P\left(T_j^k\right)} \xi_i \tag{32}$$

for $j = 1, \dots n$.

Iteration:
Set $k \leftarrow k + 1$ and continue with an assignment step until $\left\{q_j^k : j = 1, \dots n\right\}$ is met again.

Theorem 4. *The measures P^k generated by Algorithm 2 are improved approximations for P, they satisfy*

$$\mathsf{d}_r\left(P, P^{k+1}\right) \leq \mathsf{d}_r\left(P, P^k\right),$$

and the algorithm terminates after finitely many iterations.

In the case of the quadratic Wasserstein distance Algorithm 2 terminates at a local minimum $\{q_1, \dots q_n\}$ of (31).

Proof. Algorithm 2 is an iterative refinement technique, which finds the measure

$$P^k = \sum_{j=1}^{n} P\left(T_j^k\right) \delta_{q_j^k}$$

after k iterations. By construction of (32) it is an improvement due to Proposition 1, (ii) and (iii), and hence

$$\mathsf{d}_r\left(P, P^{k+1}\right) \leq \mathsf{d}_r\left(P, P^k\right).$$

The algorithm terminates after finitely many iterations because there are just finitely many Voronoi-combinations T_j.

For the Euclidean distance and $r = 2$ the expectation $\mathbb{E}\left(\xi\right) = \sum_i p_i \xi_i$ minimizes the function

$$q \mapsto \sum_i p_i \cdot \|q - \xi_i\|_2^2 = \mathbb{E}_\xi\left(\|q - \xi\|_2^2\right).$$

In this case P^k thus is a local minimum of (31). □

For other distances than the quadratic Wasserstein distance, P^k is possibly a good starting point to solve (31), but in general *not* already a local (global) minimum.

B Stochastic processes and trees

Any tree induces a filtration

Any tree with height T and finitely many nodes $\mathcal{N}$ naturally induces a filtration $\mathcal{F}$: First use $\mathcal{N}_T$ as sample space. For any $n \in \mathcal{N}$ define the atom[5] $a(n) \subset \mathcal{N}_T$ in a backward recursive way by

$$a(n) := \begin{cases} \{n\} & \text{if } n \in \mathcal{N}_T \\ \bigcup_{j \in n_+} a(j) & \text{else.} \end{cases}$$

Employing these atoms, the related sigma algebra is defined by

$$\mathcal{F}_t := \sigma\left(a(n) : n \in \mathcal{N}_t\right).$$

From the construction of the atoms it is evident that $\mathcal{F}_0 = \{\emptyset, \mathcal{N}_T\}$ for a rooted tree and that $\mathcal{F} = (\mathcal{F}_0, \ldots \mathcal{F}_T)$ is a filtration on the sample space $\mathcal{N}_T$, i.e. it holds that $\mathcal{F}_t \subset \mathcal{F}_{t+1}$. Notice that node m is a predecessor of n, i.e. $m \in \mathcal{A}(n)$, if and only if

$$a(m) \in \mathcal{A}(a(n)).$$

Employing the atoms $a(n)$ a *tree process* can be defined by

$$\begin{aligned} \nu : \{0, \ldots T\} \times \mathcal{N}_T &\to \mathcal{N} \\ (t,\, i) &\mapsto n \text{ if } i \in a(n) \text{ and } n \in \mathcal{N}_t \text{ (i.e. } n \in \mathcal{A}(i)), \end{aligned}$$

such that each

$$\begin{aligned} \nu_t : \mathcal{N}_T &\to \mathcal{N}_t \\ i &\mapsto \nu(t, i) \end{aligned}$$

is $\mathcal{F}_t-$measurable. Moreover, the process ν is *adapted* to its *natural filtration*, i.e.

$$\mathcal{F}_t = \sigma(\nu_0, \ldots \nu_t) = \sigma(\nu_t).$$

It is natural to introduce the notation $i_t := \nu_t(i)$ which denotes the state of the tree process for any final outcome $i \in \mathcal{N}_T$ at stage t. It then holds that $i_T = i$, and moreover that $i_t \in \mathcal{A}(i_\tau)$ whenever $t \leq \tau$, and finally – for a rooted tree – $i_0 = 0$. The *sample path* from the root node 0 to a final node $i \in \mathcal{N}_T$ is

$$(\nu_t(i))_{t=0}^T = (i_t)_{t=0}^T.$$

Any filtration induces a tree

On the other hand, given a filtration $\mathcal{F} = (\mathcal{F}_0, \ldots \mathcal{F}_T)$ on a finite sample space Ω it is possible to define a tree, representing the filtration: Just consider the sets A_t that collect all atoms that generate $\mathcal{F}_t$ ($\mathcal{F}_t = \sigma(A_t)$), and define the nodes

$$\mathcal{N} := \{(a, t) : a \in A_t\}$$

and the arcs

$$A = \{((a, t), (b, t+1)) : a \in A_t,\, a \in \mathcal{A}(b) \in A_{t+1}\}.$$

$(\mathcal{N}, A)$ then is a directed tree respecting the filtration $\mathcal{F}$.

Hence filtrations on a finite sample space and finite trees are equivalent structures up to possibly different labels, and in the following, we will not distinguish between them.

[5] A $\mathcal{F}-$measurable set $a \in \mathcal{F}$ is an atom if $b \subsetneq a$ implies that $P(b) = 0$.

Measures on trees

Let P be a probability measure on $\mathcal{F}_T$, such that $(\mathcal{N}_T, \mathcal{F}_T, P)$ is a probability space. The notions introduced above allow to *extend* the probability measure to the entire tree via the definition (cf. Figure 3)

$$P^{\nu}(A) := P\left(\bigcup_{t\in\{0,\dots T\}} \nu_t^{-1}(A \cap \mathcal{N}_t)\right) \qquad (A \subset \mathcal{N}).$$

In particular this definition includes the unconditional probabilities

$$P(\{n\}) =: P(n)$$

for each node. Furthermore it can be used to define conditional probabilities

$$P(\{n\}|\{m\}) =: P(n|m),$$

representing the probability of transition from n to m, if $m \in \mathcal{A}(n)$.

Value and decision processes

In a multi-period, discrete time setup the *outcomes* or *realizations* of a stochastic process are of interest, not the concrete model (the sample space): in focus is the sample space

$$\Xi := \Xi_0 \times \dots \Xi_T$$

of the stochastic process

$$\xi : \{0, \dots T\} \times \mathcal{N}_T \to \Xi.$$

The process is measurable with respect to each $\mathcal{F}_t = \sigma(\nu_t)$, from which follows (cf. [Shi96, Theorem II.4.3]) that ξ can be decomposed as

$$\xi_t = \xi_t \circ \nu_t,$$

(i.e. $\mathrm{id}_t \circ \xi = \xi_t \circ \nu_t$, where $\mathrm{id}_t : \Xi \to \Xi_t$ is the natural projection). Notice that $\xi_t \in \Xi_t$ is an observation of the stochastic process at stage t and measurable with respect to $\mathcal{F}_t$ (in symbols $\xi_t \lhd \mathcal{F}_t$), and at this stage t all prior observations

$$\xi_{0:t} := (\xi_0, \dots \xi_t)$$

are $\mathcal{F}_t-$measurable as well.

In *multistage* stochastic programming, a decision maker has the possibility to influence the results to be expected at the very end of the process by making a decision x_t at any stage t of time, having available the information which occurred up to the time when the decision is made, that is $\xi_{0:t}$. The decision has to be taken prior to the next observation ξ_{t+1} (e.g., a decision about a new portfolio allocation has to be made *before* knowing next days security prices).

This *nonanticipativity* property of the decisions is modeled by the assumption that any x_t is measurable with respect to $\mathcal{F}_t$ ($x_t \lhd \mathcal{F}_t$), such that again

$$x_t = x_t \circ \nu_t$$

(i.e. $\mathrm{id}_t \circ x = x_t \circ \nu_t$).