A Quasi-Newton Algorithm for Optimal Discretization of Markov Processes

Nils Löhndorf
University of Luxembourg, Luxembourg, nils.loehndorf@uni.lu
David Wozabal
Vrije Universeit Amsterdam, Netherlands, d.wozabal@vu.nl

In stochastic programming and stochastic-dynamic programming discretization of random model parameters is often unavoidable. We propose a quasi-Newton learning algorithm to discretize multi-dimensional, continuous discrete-time Markov processes to scenario lattices by minimizing the Wasserstein distance between the unconditional distributions of process and lattice. Scenario lattices enable accurate discretization of the conditional distributions of Markov processes while avoiding the exponential growth typically observed with scenario trees. Unlike gradient descent methods, the proposed quasi-Newton method does not require manual calibration of stepsize rules but automatically adjusts stepsizes by scaling updates with an approximation of the Hessian of the distance function. Numerical results based on various stochastic decision problems demonstrate that the proposed method achieves superior discretization of stochastic processes and results in lower optimality gaps compared to existing first-order methods. Furthermore, the method can numerically handle higher-order Wasserstein distances, proving advantageous for discretizing high-dimensional processes.

Key words: stochastic programming, dynamic programming, stochastic approximation, machine learning, clustering
History:

1. Introduction

Optimization models and solution approaches for stochastic programs often rely on approximating a continuous probability distribution by a discrete distribution with finite support to make the problem numerically tractable. Consequently, the question of how to find good discretizations has received a lot of attention in the stochastic programming literature. In two-stage stochastic programming, approximations are typically created by drawing a Monte Carlo sample from the original distribution which leads to the well-known sample average approximation (Shapiro et al. 2009, Homem-de Mello and Bayraksan 2014). Another popular idea is to explicitly minimize a *distance* between the original distribution and the discrete approximation – for example in optimal vector quantization as in (Graf and Luschgy 2000, Bally and Pagès 2003) using the Wasserstein distance. Different probability distances lead to approximations based on quasi-random numbers (Pennanen and Koivu 2005) or moment matching (Høyland et al. 2003). See Löhndorf (2016) for a more detailed overview and an empirical comparison of different methods.

In multi-stage stochastic programming, the situation is further complicated by the need to accurately represent the distributions of random model parameters conditional on past realizations. Here, one possibility is to assume conditional independence of the stochastic process and employ a combination of sample average approximation and stochastic dual dynamic programming (Pereira and Pinto 1991, Shapiro 2011). Another approach that does not require this assumption is to approximate the stochastic processes by a scenario tree (e.g., Dupačová et al. 2003, Pennanen 2005, Kaut and Wallace 2007, Heitsch and Römisch 2009, Pennanen 2009, Pflug and Pichler 2014). While scenario trees are the most general representation of discrete stochastic processes, trees that have no deterministic transitions between nodes necessarily grow exponentially in the number of stages. Although attempts have been made to avoid exponential growth (Heitsch and Römisch 2009), this leads to trees where many nodes necessarily only have a single successor, which means that decisions in these nodes are taken under perfect information about (at least part of) future randomness.

An alternative approach to using scenario trees is to approximate the stochastic process by a scenario lattice. This requires that the process has the Markov property – an assumption that is hardly limiting, as most processes commonly used in applications are Markovian or can be transformed into Markov processes by an appropriate state space expansion (Powell 2011). Moreover, unlike scenario trees, lattices do not exhibit exponential growth as they allow recombining scenarios with identical sub-trees. For example, a lattice with a constant number of nodes per stage grows only linearly in the number of stages whereas the equivalent scenario tree grows exponentially.

Lattices have been extensively studied in option pricing starting with the seminal contribution of Cox et al. (1979), who approximate one-dimensional continuous-time diffusion processes by discrete-time binomial lattices that weakly converge to the true process as the temporal resolution gets finer. Consequently, evaluating the expectation of option values on the lattice process asymptotically yields the true option value and avoids the need for closed form solutions. In the literature on real options pricing, these initial ideas have been substantially refined to incorporate information from observed market prices (Rubinstein 1994) and cover mean reverting processes (e.g. Hahn and Dyer 2008) as well as processes with complex volatilities and jumps (e.g. Harikae et al. 2021, Wang and Dyer 2010). In Wang and Dyer (2010) and Chourdakis (2004), multinomial lattice approximations are explored. The general idea of these papers is to construct lattices in such a way that the resulting process matches certain characteristics of the original process, mostly volatility and drift.

Broadie and Glasserman (2004) consider discrete-time problems based on fairly general stochastic processes that are approximated by multinomial lattices called *stochastic meshes*. Similar to our approach, convergence to the true distribution is facilitated by increasing the

number of nodes per time step as opposed to increasing the *number of time steps*. However, the considered sampling scheme is closely motivated by the intended options pricing application.

Felix and Weber (2012) and Bardou et al. (2009) are closest to our approach. The authors consider real options pricing problems with complex decisions and use the Wasserstein metric to measure the difference between the continuous stochastic process and the lattice discretization. Felix and Weber (2012) employ k-means clustering on simulated realizations of general stochastic processes to minimize the distance between lattice and process, while Bardou et al. (2009) restrict their attention to Gaussian randomness and use existing optimal quantizers from the literature to construct lattices.

The effectiveness of using scenario lattices for solving multi-stage stochastic programs with many time stages and multi-dimensional randomness have been demonstrated in Bonnans et al. (2012), Löhndorf et al. (2013), Löhndorf and Shapiro (2019), Löhndorf and Wozabal (2021). Nonetheless, the methods used to construct lattices are mostly ad-hoc and their application in stochastic programming comes without performance guarantees.

The theoretical literature on lattices is sparse. Kiszka and Wozabal (2020) propose a *lattice distance* and show that the objective values of certain stochastic programming problems are Lipschitz continuous with respect to the proposed distance, facilitating a quantitative stability analysis. However, the approach does not directly lend itself to the computationally efficient generation of scenario lattices.

Bally and Pagès (2003) refer to lattices as *vector quantization trees* and propose a first-order learning algorithm that minimizes the distance between the unconditional distribution of a Markov chain and the nodes of the lattice. Their learning algorithm requires user-defined stepsize rules and therefore manual parameter tuning which can lead to sub-optimal results, as we will see in Section 3.

Apart from multi-stage stochastic programming, our research also relates to classic theory on approximations of Markov decision processes (MDPs). However, our setting is different in several notable ways. First, typical formulations of MDPs feature finite state and action spaces along with discrete transition probabilities to model randomness (Bertsekas 1981, Langen 1981, White 1982). By contrast, we consider problems with continuous randomness and uncountable action spaces. Second, problems based on scenario lattices necessarily have a finite horizon, whereas most work on approximations of MDPs deals with infinite horizon problems, exploiting the fact that optimal value functions are fixed points of the Bellman operator (White 1982, Chow and Tsitsiklis 1991, Saldi et al. 2017). Third, research that deals with continuous state spaces usually imposes (Lipschitz) continuity conditions on the probability transition kernels (Bertsekas 1981, Dufour and Prieto-Rumeau 2013, Saldi et al. 2017), which we do not require. That being said, our results contribute to the literature on approximation of finite-horizon MDPs with a continuous state space where exogenous states are independent of endogenous

states. In those cases a lattice can be used as approximation of the exogenous (stochastic) state transition process.

The contributions of this paper can be summarized as follows: In Section 2, we propose a stochastic quasi-Newton algorithm that minimizes the Wasserstein distance between the unconditional distributions of a continuous Markov process and a scenario lattice. We use second-order information to choose the algorithm's learning rates. The use of second-order information as in quasi-Newton methods has been found to speed up convergence in statistical learning (Byrd et al. 2016, Wang et al. 2017), as it provides a way for selecting good learning rates, which is a problem of practical importance in many machine learning algorithms (Martens 2010).

Similar to quasi-Newton methods, we employ an approximation of the Hessian, specifically using a diagonal approximation that can be efficiently estimated and for which inversion is computationally inexpensive. Our approach is related to LeCun et al. (2002), who describe how estimates of the Hessian's diagonal can accelerate the training of multi-layer neural networks. Likewise, Bottou and Bengio (1994) discuss the utilization of second-order information for k-means clustering. However, neither of these studies provides convergence guarantees. In this work, we close this gap by showing that, under mild moment conditions, every limit point of our second-order learning algorithm is a critical point of the discretization problem.

In a computational experiment, which we summarize in Section 3, we compare the proposed quasi-Newton method with the gradient descent method proposed in Bally and Pagès (2003) based on four different stochastic programs for which optimal solutions are either known or can be approximated by tight bounds: (1) a two-stage newsvendor model, (2) a multi-period/multi-product newsvendor problem, (3) a multi-period inventory control problem with Markovian demand, and (4) a realistic option pricing problem for natural gas storages. The findings demonstrate that the method produces discretizations that are closer to the true distributions than the corresponding discretizations produced by a benchmark method based on gradient descent. Furthermore, we demonstrate that the method is numerically stable allowing for the usage of higher-order Wasserstein distances, which proves advantageous for discretization of high-dimensional stochastic processes and leads to better approximate policies out-of-sample.

Notation : Throughout the paper, we work on a general probability space (Ω, σ) and denote by $\mathcal{L}^1(\mathbb{R}^d)$ the space of measurable functions to $\mathbb{R}^d$ that are r-integrable ($r \geq 1$) with respect to the Lebesgue measure λ. We denote by T the number of decision stages and by $\xi = (\xi_1, \ldots, \xi_T)$ with $\xi_t : \Omega \to \mathbb{R}^{d_t}$ a discrete-time stochastic process with values in $1, \ldots, T$, by $\tilde{\xi}_t$ samples from ξ_t, and by $\bar{\xi}_t$ an approximated discrete version of ξ_t. Furthermore, for any $N \in \mathbb{N}$, we denote the set $\{1, \ldots, N\}$ by $[N]$, the indicator function of set A by $\mathbb{1}_A$, the inner product in $\mathbb{R}^d$ by $\langle \cdot, \cdot \rangle$, the p-norm in $\mathbb{R}^d$ by $||\cdot||_p$, and by $\mathrm{diag}_d(x) \in \mathbb{R}^{d \times d}$ the diagonal matrix with the diagonal equal to the vector $x \in \mathbb{R}^d$. If $c \in \mathbb{R}$ is a scalar, A is a matrix, and x is a vector, then cx and cA denote the point-wise (Hadamard) multiplication.

2. An Algorithm to Generate Scenario Lattices

We start this section by introducing scenario lattices as canonical discrete approximations of Markov processes. Our objective is to construct lattices such that optimal values and optimal decisions for a stochastic programming problem based on the lattice are close to the optimal value of the problems using the *true* processes.

In Section 2.1, we contrast scenario lattices with scenario trees which are commonly used to approximate general stochastic processes and argue that lattices give rise to computationally simpler problems. In Section 2.2, we propose a quasi-Newton algorithm that generates discrete approximations for the unconditional distributions of a Markov process by minimizing the Wasserstein distance between the discrete and the original continuous distribution. In Section 2.3, we describe an algorithm to approximate inhomogeneous Markov processes based on the approximations of the unconditional distributions for all stages.

2.1. Scenario Lattices

Stochastic programming problems with parameters that are driven by high-dimensional random processes are generally computationally intractable – a phenomenon referred to as the *curse of dimensionality* (Hanasusanto et al. 2016). The problem is further aggravated in multi-stage stochastic programming where the number of possible outcomes increases exponentially in the number of stages. In this section, we introduce scenario lattices as discretizations of Markov processes that can be used to mitigate this problem.

A popular approach to discretize stochastic processes are scenario trees. To deal with exponential growth of the number of nodes in the number of stages, scenario trees typically have only a few stages or contain nodes where the tree does not branch, which effectively leads to deterministic problems at these nodes.

The complexity of a scenario tree can be reduced substantially if the data process is Markovian. In this case, the conditional distributions of a random process $\xi = (\xi_1, \dots, \xi_T)$ in $t+1$ no longer depend on the entire history of the process but only on outcomes in t. This means that all histories $(\xi_1, \dots, \xi_{t-1})$ of the process that lead to the same state ξ_t continue in identical sub trees that can be merged without loss of information. We refer to such *recombining* scenario trees as *scenario lattices*, in line with the terminology often used in mathematical finance.

Formally, a scenario lattice is a graph organized in a finite number of layers. Each layer is associated with a discrete point in time and contains a finite number of nodes. Successive layers are connected by arcs. A node represents a possible state of the stochastic process, and an arc indicates the possibility of a state transition between the two connected nodes. Each arc is associated with a probability weight, and the weights of outgoing arcs of a node add up to one. Note that this definition in particular covers inhomogeneous Markov processes where

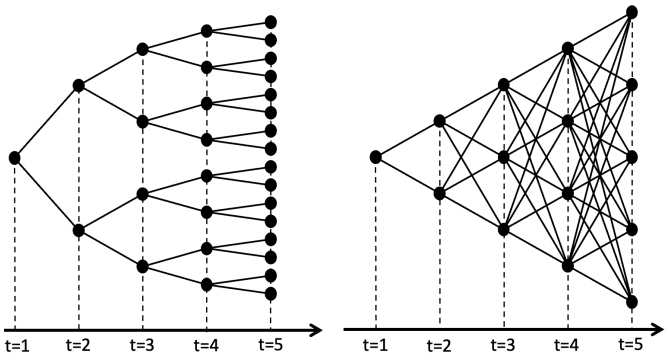


Figure 1 A tree with 31 nodes representing 16 scenarios on the left and a lattice with 15 nodes representing 120 scenarios on the right.

conditional distributions change over time. A scenario tree differs from a scenario lattice by the additional requirement that every node in t has only one predecessor in $(t-1)$.

We denote the lattice process by $\bar{\xi} = (\bar{\xi}_1, \ldots, \bar{\xi}_T)$, the number of nodes of the lattice in stage t by N_t, and $\bar{\xi}_{tn}$, $n \in [N_t]$ as the state of the lattice process at node n in stage t. Assuming that all state transitions between nodes in consecutive stages have positive probabilities, the number of scenarios modeled by the lattice equals $\prod_{t=1}^{T} N_t$. As the number of stages grows, additional nodes needed to construct a lattice are those of the newly added stages. By contrast, the number of nodes in a scenario tree – assuming that each node has more than one successor – grows exponentially in the number of stages. See Figure 1 for an illustrative comparison between a typical scenario tree and a scenario lattice.

Scenario lattices are therefore a natural choice for Markovian decision problems with many time stages, as they allow solving problems with hundreds or even thousands of time stages, which is far beyond the scope of conventional scenario trees.

2.2. Optimal Discretization of Unconditional Probabilities

In this section, we discuss the stage-wise discretization of the process ξ_t based on the Wasserstein distance. For the purpose of this discussion and to simplify notation, ξ will denote a random vector, $\xi : \Omega \to \mathbb{R}^d$, instead of a stochastic process. From stability theory of stochastic programs, it is known that minimizing the Wasserstein distance guarantees favorable properties of the discretization (Bally and Pagès 2003, Heitsch and Römisch 2009, Pflug and Pichler 2012).

2.2.1. The Wasserstein Distance We start by reviewing basic properties of the Wasserstein distance and its connections to stochastic programming. Loosely speaking, the Wasserstein distance between two distributions can be defined as the minimal cost of transporting mass from one probability distribution to the other, whereby cost are measured by the distance between the respective outcomes (see Villani 2003, for a comprehensive introduction).

DEFINITION 1. Let $\xi : \Omega \to \mathbb{R}^d$ and $\bar{\xi} : \bar{\Omega} \to \mathbb{R}^d$ be two random vectors with image measure $\mathbb{P}$ and $\bar{\mathbb{P}}$, respectively. The Wasserstein distance of order ($r \geq 1$) between ξ and $\bar{\xi}$ is defined as

$$W_r(\xi, \bar{\xi}) = \begin{cases} \inf\limits_{\pi} \left(\int\limits_{\mathbb{R}^d \times \mathbb{R}^d} \|\xi - \bar{\xi}\|_r^r \, \pi\left(d\xi, d\bar{\xi}\right) \right)^{\frac{1}{r}} \\ \text{s.t. } \pi[A \times \mathbb{R}^d] = \mathbb{P}[A] \quad \forall A \in \mathcal{A}, \\ \qquad \pi[\mathbb{R}^d \times B] = \bar{\mathbb{P}}[B] \quad \forall B \in \bar{\mathcal{A}}, \end{cases} \tag{1}$$

where $||\cdot||_r$ is the r-norm in $\mathbb{R}^d$, the infimum is taken over all probability measures π on $(\mathbb{R}^d \times \mathbb{R}^d, \mathcal{A} \otimes \bar{\mathcal{A}})$, and $\mathcal{A}$ and $\bar{\mathcal{A}}$ are sigma-algebras generated by ξ and $\bar{\xi}$ on $\mathbb{R}^d$.

Note that the Wasserstein distance in general only requires a distance in the objective of (1), which need not necessarily be generated by a norm. For the purpose of this paper, we restrict our attention to the distances generated by r-norms in $\mathbb{R}^d$.

For $r = 1$, the Wasserstein distance has the following dual representation

$$W_1(\xi, \bar{\xi}) = \sup_{||f||_{Lip} \leq 1} \left\{ \int f(\xi) \, P(d\xi) - \int f(\bar{\xi}) \, \bar{P}(d\bar{\xi}) \right\},$$

where $||f||_{Lip}$ is the Lipschitz constant of f, i.e., the supremum is taken over all Lipschitz functions with a Lipschitz constant smaller than 1. Based on this result, W_1 can be directly used to bound the error in two-stage stochastic programming. In particular, for a well behaved two-stage problem,

$$\Pi_P^* = \begin{cases} \max\limits_{x_1} C_1(x_1) + \mathbb{E}_P(V(x_1, \xi)) \\ \text{s.t.} \quad x_1 \in \mathcal{X}_1, \end{cases} \qquad V(x_1, \xi) = \begin{cases} \max\limits_{x_2} C_2(x_2, \xi) \\ \text{s.t.} \quad x_2 \in \mathcal{X}_2(x_1), \end{cases} \tag{2}$$

where the second stage's cost C_2 is dependent on a random variable ξ with distribution $\mathbb{P}$, the recourse function V is Lipschitz, i.e., there is a $L > 0$ such that

$$|V(x, \xi) - V(x, \bar{\xi})| \leq L \, \|\xi - \bar{\xi}\|_1, \quad \forall x \in \mathcal{X}_\infty \tag{3}$$

(see Kall and Mayer 2006, Theorem 2.3 for the case of linear problems).

If in problem (2) the random variable ξ is such that a numerical solution becomes computationally intractable, one might want to replace ξ with a *simpler* approximation $\bar{\xi}$ with distribution $\bar{\mathbb{P}}$ in order to solve the problem at least approximately. The error induced by this step can be bounded using the Wasserstein distance assuming without loss of generality that $\Pi_{\bar{P}}^* \geq \Pi_P^*$

$$\Pi_{\bar{P}}^* - \Pi_P^* \leq \mathbb{E}_{\bar{P}}(V(x_{\bar{P}}^*, \bar{\xi})) - \mathbb{E}_P(V(x_{\bar{P}}^*, \xi)) \leq L \, W_1(\xi, \bar{\xi}), \tag{4}$$

with $x_{\bar{P}}^*$ as the first-stage optimal solution of (2) for the approximated problem using $\bar{\xi}$. In case that $\Pi_{\bar{P}}^* \leq \Pi_P^*$ an analogous argument establishes that the objective value is Lipschitz continuous with respect to the Wasserstein distance. This basic stability result suggests that it is desirable to select a discrete random variable $\bar{\xi}$ such that $W_1(\xi, \bar{\xi})$ and therefore the approximation error remains small.

We will restrict our attention to the case $r \geq 2$ and make the following assumption:

ASSUMPTION 1. *The image measure* $\mathbb{P}$ *of* ξ *is absolutely continuous with respect to the Lebesgue measure* λ *on* $\mathbb{R}^d$*,* $r \geq 2$*, and* $\xi \in \mathcal{L}^{2r-2}(\mathbb{R}^d)$*.*

To better understand the problem of finding an approximating $\bar{\xi}$, we next discuss the relation of the Wasserstein distance to the N-center problem. To that end, note that every set of N points $\bar{\xi}_1, \ldots, \bar{\xi}_N$ with $\bar{\xi}_i \in \mathbb{R}^d$ uniquely defines a partition $\Gamma = (\Gamma_1, \ldots, \Gamma_N)$ of $\mathbb{R}^d$ by

$$\Gamma_i = \left\{ y \in \mathbb{R}^d : i = \operatorname*{arg\,min}_{1 \leq j \leq N} ||y - \bar{\xi}_j||_p \right\},$$

which is called a Voronoi diagram. For each Voronoi diagram, we can define an associated discrete random variable $\bar{\xi} : \Omega \to \mathbb{R}^N$ with N atoms via

$$\bar{\xi}(\omega) = \sum_{n=1}^{N} \mathbb{1}_{\Gamma_i}(\xi(\omega)) \bar{\xi}_n \tag{5}$$

and $\mathbb{P}[\bar{\xi} = \bar{\xi}_n] = \mathbb{P}[\Gamma_n]$. Note that the ambiguity of the definition of $\bar{\xi}$ at the borders of Voronoi cells is inconsequential due to Assumption 1 and can be resolved arbitrarily.

We define the *quantization error* of the centers $\bar{\xi}$ as (see Graf and Luschgy 2000, Lemma 3.1)

$$D(\bar{\xi}) = r^{-1} W_r^r(\bar{\xi}, \xi) = r^{-1} \sum_{n=1}^{N} \int_{\Gamma_n(\bar{\xi})} ||\bar{\xi}_n - \xi||_r^r \, \mathbb{P}(d\xi) = r^{-1} \int_{\mathbb{R}^d} \min_{1 \leq n \leq N} ||\bar{\xi}_n - \xi||_r^r \, \mathbb{P}(d\xi).$$

Consequently, finding an optimal discretization of ξ with a fixed number of N atoms $\bar{\xi}_1, \ldots, \bar{\xi}_N \in \mathbb{R}^d$ in terms of the Wasserstein distance is equivalent to solving

$$\inf_{\bar{\xi}} D(\bar{\xi}) \tag{6}$$

and then defining a random variable $\bar{\xi}$ as in (5).

For an arbitrary Voronoi diagram Γ, we furthermore define

$$D(\bar{\xi}, \Gamma) = r^{-1} \sum_{n=1}^{N} \int_{\Gamma_n} ||\bar{\xi}_n - \xi||_r^r \, \mathbb{P}(d\xi),$$

which defines the effort of moving the mass from the Voronoi cells Γ_n to the points $\bar{\xi}_n$, even if the n-th centers is not the closest. Clearly, $D(\bar{\xi}) = D(\bar{\xi}, \Gamma(\bar{\xi}))$.

Unfortunately, (6) is an NP-hard, non-convex optimization problem for which even finding a local optimum can be computationally expensive for large N. In line with Bally and Pagès (2003), we will use stochastic approximation to learn the atoms in an online fashion from samples of ξ. In contrast to their approach, we propose a quasi-Newton algorithm that, much like the Newton-Raphson algorithm, updates gradients based on an approximation of the Hessian of D_r which makes the method parameter-free.

We start our analysis of the discretization problem by collecting some well known facts about D_r in the following Lemma. To that end, we denote by $\nabla_{\bar{\xi}_i} D_r(\bar{\xi}) \in \mathbb{R}^d$ the vector of partial derivatives of the Wasserstein distance with respect to the elements of the center $\bar{\xi}_i \in \mathbb{R}^d$, where a change in a center changes $\bar{\xi}$ according to (5). Similarly, we write $\nabla D_r(\bar{\xi})$ for the whole gradient in $\mathbb{R}^{N \times d}$.

LEMMA 1. *Under Assumption 1 the following holds:*

1. $\bar{\xi} \mapsto D_r(\bar{\xi})$ *is differentiable and the gradients are given by*

$$\nabla_{\bar{\xi}_i} D_r(\bar{\xi}) = \int_{\Gamma_i(\bar{\xi})} \operatorname{sign}(\bar{\xi}_i - \xi) \cdot |\bar{\xi}_i - \xi|^{r-1} \, \mathbb{P}(d\xi), \quad \forall i \in [N]. \tag{7}$$

where $\cdot$ *is the pointwise multiplication of vectors.*

2. *The mapping* $\bar{\xi}_i \to \nabla_{\bar{\xi}_i} D_r(\bar{\xi})$ *is Lipschitz with respect to* $||\cdot||_2$.

3. *For given Voronoi diagram* Γ*, the the Hessian* $\nabla^2_{\bar{\xi}} D_r$ *of the mapping* $\bar{\xi} \mapsto D_r(\bar{\xi}, \Gamma)$ *has the following elements*

$$(\nabla^2_{\bar{\xi}} D_r(\bar{\xi}, \Gamma))_{jj} = \begin{cases} (r-1) \int_{\Gamma_i} |\bar{\xi}_{ij} - \xi_j|^{r-2} \, \mathbb{P}(d\xi), & \text{for } d(i-1) < j \leq di \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Proof. The first point follows from Graf and Luschgy (2000), Lemma 4.10, since the distribution of ξ is continuous and therefore the Voronoi diagram associated with $\bar{\xi}$ is a tesselation. The second point follows by Proposition 1 in Pflug (2001). To see the third point notice that $\nabla_{\xi} D_r(\bar{\xi}, \Gamma)$ equals (7) with $\Gamma_i(\bar{\xi})$ replaced with Γ. The result then follows by differentiation under the integral sign. □

Note that the Hessian of $D_r(\bar{\xi})$ is analytically intractable, since changes in $\bar{\xi}$ change the boundaries of the Voronoi cells and therefore the results for the second derivative. Furthermore, $\nabla^2_{\bar{\xi}} D_r(\bar{\xi})$ is a dense matrix. We therefore use $\nabla^2_{\bar{\xi}} D_r(\bar{\xi}, \Gamma)$ as a diagonal approximation for $\nabla^2_{\bar{\xi}} D_r(\bar{\xi})$, which is much easier to estimate and allows for a numerically inexpensive calculation of the inverse.

REMARK 1. For the case of $r = 2$, the $(\nabla^2_{\bar{\xi}} D_2(\bar{\xi}, \Gamma)$ is a matrix with elements

$$(\nabla^2_{\bar{\xi}} D_2(\bar{\xi}, \Gamma))_{kl} = \begin{cases} \mathbb{P}[\Gamma^i], & \text{for } d(i-1) < k = l \leq di \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

2.2.2. A stochastic quasi-Newton algorithm. Based on the preliminary results of the previous section, we propose a quasi-Newton algorithm tailored to solving the N-center problem. Solving (6) using the Newton-Raphson method entails choosing a starting point $\bar{\xi}^1$ that is updated recursively as follows

$$\bar{\xi}^{k+1} = \bar{\xi}^k - (\nabla^2 D(\bar{\xi}^k))^{-1} \nabla D(\bar{\xi}^k). \tag{10}$$

The use of second-order information typically leads to faster convergence than first-order methods and mitigates the need to manually choose stepsizes, making the method parameter free (e.g. Byrd et al. 2016). Also note that the stepsizes implied by using information in $\nabla^2_{\bar{\xi}} D_r(\bar{\xi}, \Gamma)$ are different for different components of $\bar{\xi}^k$, resulting in tailored learning rates for every center.

Since the calculation of the exact expectations in the gradient $\nabla D(\bar{\xi}^k)$ is expensive, we approximate (10) by sampling from ξ as it is common in stochastic gradient descent algorithms.

Furthermore, we approximate the Hessian $\nabla^2 D(\bar{\xi}^k)$ by a sampled version of $\nabla^2_{\bar{\xi}} D_r(\bar{\xi}, \Gamma)$, where Γ is held fixed to the Voronoi diagram of the last point. To this end, we define the starting point $\bar{\xi}^1 \in \mathbb{R}^{N\times d}$ by randomly selecting N samples from ξ.

To calculate updates, we draw a sequence of i.i.d. samples $\tilde{\xi}^1$, $\tilde{\xi}_2$, ... from the distribution of ξ and denote the sampling space by

$$(\Omega^\infty, \mathcal{F}^\infty) = (\times_{k=1}^\infty \Omega, \times_{k=1}^\infty \mathcal{F}). \tag{11}$$

All claims of almost sure convergence are with respect to $(\Omega^\infty, \mathcal{F}^\infty)$. We denote by $\mathcal{F}_k \subseteq \mathcal{F}^\infty$ the σ-algebra generated by the first k random draws $\tilde{\xi}^1, \ldots, \tilde{\xi}^k$. Furthermore, we initialize matrices H_i^0 for $1 \le i \le N$ as unit matrices in $\mathbb{R}^d$, i.e., $H_i^0 = \mathrm{diag}_d((1,\ldots,1)^\top) \in \mathbb{R}^{d\times d}$. The algorithm is given by the following update rule:

$$\bar{\xi}_i^{k+1} = \begin{cases} \bar{\xi}_i^k - H_i^k \left(\mathrm{sign}(\bar{\xi}_i^k - \tilde{\xi}^k) \cdot |\bar{\xi}_i^k - \tilde{\xi}^k|^{r-1} \right), & \text{for } i = \arg\min_j \, ||\bar{\xi}_i^k - \tilde{\xi}^k||_r, \\ \bar{\xi}_i^k, & \text{otherwise,} \end{cases} \tag{12}$$

$$H_i^k = \begin{cases} \left((H_i^{k-1})^{-1} + \mathrm{diag}_d \left((r-1)|\bar{\xi}_i^k - \tilde{\xi}^k|^{r-2} \right) \right)^{-1}, & \text{for } i = \arg\min_j \, ||\bar{\xi}_i^k - \tilde{\xi}^k||_r, \\ H_i^{k-1}, & \text{otherwise,} \end{cases} \tag{13}$$

for all $i = 1, \ldots, N$ and all $k \in \mathbb{N}$. Note that (12) and (13) approximate the gradient in (10) based on a single sample $\tilde{\xi}^k$ and replacing the Hessian by the approximation H^k. We only update the center $\bar{\xi}_i$ which is closest to the new sample in iteration k so that H_i^k is the part of the H^k that corresponds to the i-th center $\bar{\xi}_i$.

Note that the idea of using $\nabla^2_{\bar{\xi}} D_r(\bar{\xi}, \Gamma)$ instead of $\nabla^2_{\bar{\xi}} D_r(\bar{\xi})$ for the update is similar to the idea employed in the k-means algorithm where partitions are also held constant when updating cluster centers, which is effectively what we are doing, when we ignore the changes of the boundaries of the Voronoi cells in the computations of the second derivatives.

Remark 2. Note that the computational complexity of performing the steps in (12) and (13) is low, since the update of every node requires a handful of basic floating point operations whose number grows linearly in the dimension d of the problem. The overall runtime complexity is therefore jointly linear in d, the number of stages, and the number of updates. Hence, as we will see in the numerical results section, millions of updates can be performed in a matter of seconds on a standard computer.

Also note that for $r = 2$, the update is especially simple, since the diagonal matrix $(H_i^k)^{-1}$ contains the number of samples that were sampled up to iteration k in the Voronoi cell defined by node i as diagonal entries. Consequently, if the centers stabilize, diagonal elements of kH_i^k converge to $\mathbb{P}[\Gamma_i]$.

2.2.3. Convergence analysis. In the following, we will prove that the proposed algorithm converges almost surely. We start with a technical result, which establishes that the expected locations of the centers found by the algorithm stay in a bounded region.

LEMMA 2. *At iteration k, for every $i \in [N]$ there is a $C_i > 0$ with $C_i \triangleleft \mathcal{F}_{k-1}$ such that*

$$\mathbb{E}\left[|\bar{\xi}_i^l|^{r-1}|\mathcal{F}_{k-1}\right] \leq C_i < \infty, \quad \forall l \in \mathbb{N} : l \geq k+1,$$

where the above inequalities holds component-wise.

Proof. We denote by Γ_i^k the Voronoi cell associated with $\bar{\xi}_i^k$ and split up $\Gamma_i^k = A_1 \cup A_2$ with $A_1 = \Gamma_i^k \cap \prod_{j=1}^d [-\bar{\xi}_j^k, \bar{\xi}_j^k]$ and $A_2 = \Gamma_i^k \setminus A_1$. Note that A_1 represents the part of the cell that is not *too far* from the current center while A_2 is the part that is *farther away*. We will subsequently treat these two sets separately.

First, fix $l = k+1$ and write

$$\begin{aligned}\mathbb{E}\left[|\bar{\xi}_i^{k+1}|^{r-1}|\mathcal{F}_{k-1}\right] &= |\bar{\xi}_i^k|^{r-1}\mathbb{P}[\mathbb{R}^d \setminus \Gamma_i^k] \\ &\quad + \int_{\Gamma_i^k} \left|\bar{\xi}_i^k - H_i^k \operatorname{sign}(\bar{\xi}_i^k - \tilde{\xi}^k)|\bar{\xi}_i^k - \tilde{\xi}^k|^{r-1}\right|^{r-1} \mathbb{P}(d\tilde{\xi}^k).\end{aligned}$$

Furthermore, note that

$$\begin{aligned}H_i^k|\bar{\xi}_i^k - \tilde{\xi}^k|^{r-1} &= H_i^k \operatorname{diag}(|\bar{\xi}_i^k - \tilde{\xi}^k|)^{r-2}|\bar{\xi}_i^k - \tilde{\xi}^k| \\ &= \left((\operatorname{diag}(|\bar{\xi}_i^k - \tilde{\xi}^k|)^{r-2}H_i^{k-1})^{-1} + (r-1)\right)^{-1}|\bar{\xi}_i^k - \tilde{\xi}^k| \\ &\leq |\bar{\xi}_i^k - \tilde{\xi}^k| \qquad (14)\end{aligned}$$

where the last inequality follows since $(r-1) \geq 1$ and the elements of H_i^{k-1} are strictly positive. Therefore

$$\int_{A_1} \left|\bar{\xi}_i^k - H_i^k \operatorname{sign}(\bar{\xi}_i^k - \tilde{\xi}^k)|\bar{\xi}_i^k - \tilde{\xi}^k|^{r-1}\right|^{r-1} \mathbb{P}(d\tilde{\xi}^k) \leq \int_{A_1} |\bar{\xi}_i^k|^{r-1}\, \mathbb{P}(d\tilde{\xi}^k) \leq \mathbb{P}[A_1]|\bar{\xi}_i^k|^{r-1}.$$

Furthermore, by (14) and Hölder's inequality

$$\begin{aligned}\int_{A_2} \left|\bar{\xi}_i^{k+1}\right|^{r-1} \mathbb{P}(d\tilde{\xi}^{k+1}) &\leq \int_{A_2} \left|\tilde{\xi}_i^k\right|^{r-1} \mathbb{P}(d\tilde{\xi}^k) \leq \mathbb{P}[A_2] \int_{\mathbb{R}^d} \left|\tilde{\xi}_i^k\right|^{2(r-1)} \mathbb{P}(d\tilde{\xi}^k) \\ &\leq \mathbb{P}[A_2]||\xi||_{2(r-1)}^{2(r-1)}.\end{aligned}$$

Combining these two results, we can approximate

$$\mathbb{E}\left[|\bar{\xi}_i^{k+1}|^{r-1}|\mathcal{F}_{k-1}\right] \leq \mathbb{P}[A_1]|\bar{\xi}_i^k|^{r-1} + \mathbb{P}[A_2]||\xi||_{2(r-1)}^{2(r-1)} < \infty,$$

due to the moment conditions in Assumption 1. Iterating this argument yields the claimed result. □

As we will show in the next proposition all the limit points of the sequence $\bar{\xi}^k$ are stationary points of D_r. The following lemma facilitates this result by establishing that the changes in H_i^k are of exactly the right size in order for the stochastic algorithm to converge almost surely.

For iteration k of the updates defined in (12) and (13), we define $i_k^* = \arg\min_j ||\bar{\xi}_i^k - \tilde{\xi}^k||_r$ and $H_{i_k^*,j}^k$ as the j-th diagonal entry of the matrix $H_{i_k^*}^k$.

LEMMA 3. *Under Assumption 1 the following holds:*

1. *Except on a set of measure* 0*, the sum of elements in* $H_{i_k^*}^k$ *diverges, i.e.,*

$$\sum_{k=1}^{\infty} H_{i_k^*,j}^i = \infty, \quad \forall j \in [d], \text{ almost surely.}$$

2. $H_{i_k^*}^k$ *goes to zero almost surely.*
3. *For all* $K \in \mathbb{N}$

$$\mathbb{E}\left[H_{i_k^*}^{k\top} H_{i_k^*}^k \middle| \mathcal{F}_K\right] \leq O(k^{-2}),\ k \geq K, \text{ almost surely.} \tag{15}$$

Proof. Fix an arbitrary increasing sequence n_k in $\mathbb{N}$ with $n_k = O(k)$. Considering the j-th component $\bar{\xi}_{i_k^*,j}^k$ of $\bar{\xi}_{i_k^*}^k$, it follows by $r \geq 2$ and the (generalized) Markov inequality that

$$\mathbb{P}\left[|\bar{\xi}_{i_k^*,j}^k - \tilde{\xi}^k|^{r-2} \geq n_k\right] \leq \frac{\mathbb{E}[|\bar{\xi}_{i_k^*,j}^k - \tilde{\xi}_j^k|^{r-1}]}{n_k^\alpha}$$

with $\alpha = \frac{r-1}{r-2} > 1$. By Lemma 2, Assumption 1, and the fact that $|a-b|^r \leq 2^{r-1}(|a|^r + |b|^r)$ for $r \geq 1$ we can conclude that

$$\mathbb{E}[|\bar{\xi}_{i_k^*,j}^k - \tilde{\xi}_j^k|^{r-1}] \leq 2^{r-2}\left(\mathbb{E}\left[|\bar{\xi}_{i_k^*,j}^k|^{r-1}\right] + \mathbb{E}\left[|\tilde{\xi}_j^k|^{r-1}\right]\right) \leq C < \infty.$$

Hence, we obtain

$$\sum_{k=1}^{\infty} \mathbb{P}\left[|\bar{\xi}_{i_k^*,j}^k - \tilde{\xi}_j^k|^{r-2} \geq n_i\right] \leq C \sum_{k=1}^{\infty} n_k^{-\alpha} < \infty$$

and use the Borel-Cantelli lemma to infer that almost surely $|\bar{\xi}_{i_k^* j}^k - \tilde{\xi}_j^k|^{r-2} \geq n_k$ happens only finitely often. This implies that for almost every $\omega \in \Omega$, there is an $N \in \mathbb{N}$ such that

$$\sum_{k=N}^{\infty} H_{i_k^*,j}^k \geq \sum_{k=N}^{\infty} \left(1 + (r+1)|\bar{\xi}_{i_k^* j}^k - \tilde{\xi}_j^k|^{r-2}\right)^{-1} \geq \sum_{k=N}^{\infty} \frac{1}{n_k} = \infty,$$

showing the first part of the result.

To prove the second point, note that for a given $I \in \mathbb{N}$ independent of the location of the centers $\bar{\xi}_i^k$ at least $^{I-1}/_I$ of the mass is located in cells with probability larger than $(IN)^{-1}$. Denote the set of indices of these centers as $\mathcal{I} \subseteq [N]$.

For a given $j \in [d]$ consider the marginal distribution $\mathbb{P}_j$ of the j-th component ξ_j of ξ and define the ball $B_u(x) = \{y \in \mathbb{R} : (r-1)|x-y|^{r-2} \leq u\}$. Now choose $u^* > 0$ such that

$$\mathbb{P}_j[B_{u^*}(x)] \leq (I^2 N)^{-1}, \quad \forall x \in \mathbb{R}.$$

Note that such an u^* exists by absolute continuity of $\mathbb{P}_j$ with respect to the Lebesgue measure λ. Hence, it follows that the probability that the update to $H_{i_k^*}$ is at least u^* can be bounded below as follows

$$\begin{aligned}\mathbb{P}\left[|\bar{\xi}^k_{i_k^*,j} - \tilde{\xi}^k_j|^{r-2} > u^*\right] &\geq \sum_{i\in\mathcal{I}} \left(\mathbb{P}[\Gamma_i^k] - (I^2N)^{-1}\right) \\ &\geq \frac{I-1}{I} - \sum_{i\in\mathcal{I}}(I^2N)^{-1} \geq \frac{I-1}{I} - \frac{1}{I} \xrightarrow{I\to\infty} 1. \end{aligned} \tag{16}$$

For a given $\varepsilon > 0$ and $\delta > 0$, choose I such that (16) is larger than $(1-\delta)$ for some $u^* > 0$. Now for a given ω consider those cells $\mathcal{I} \subseteq [N]$ which receive infinitely many updates larger than $(IN)^{-1}$ and choose a K such that for all $k \geq K$ all other cells get updates smaller than $(IN)^{-1}$. Clearly, the values updates of cells in $i \in \mathcal{I}$ happen infinitely often after k and each update has a positive probability of being at least of size u^*. Therefore for almost all ω there is a $K' \geq K$ such that H_i^k is eventually smaller than ε for all $i \in \mathcal{I}$. Becasue of the choice of δ, we therefore have $\mathbb{P}[H^k_{i_k^*,j} < \varepsilon] < \delta$ for $k \geq K'$ and the sequence converges in probability. Since $H_{i_k^*,j}$ are bounded and therefore uniformly integrable, they also converge in L^1 and therefore almost surely establishing the second claim.

To show the last statement, fix $I \geq 4$ making the chance that individual updates are larger than u^* larger than $1/2$. Furthermore, note that the expected $H^k_{i_k^*,j}$ are smallest if all the updates are spread evenly across all cells $i \in [N]$. It follows that

$$\mathbb{E}\left[(H^k_{i^*,j})^2\right] \leq (u^*)^{-2}\mathbb{E}\left[\frac{1}{(1+X)^2}\right] = (u^*)^{-2}2^{-\tilde{k}}\sum_{l=0}^{\tilde{k}}\binom{\tilde{k}}{l}(1+l)^{-2},$$

where X is binomially distributed with $\tilde{k} = \lfloor kN^{-1}\rfloor$ trials and probability of success $1/2$, where a success models an update of size u^* and larger while a failure models a smaller update.

To show that $\mathbb{E}\left[(H^k_{i^*,j})^2\right]$ converges to 0 at least at the speed of k^{-2}, we note that

$$\mathbb{E}\left[(H^k_{i^*,j})^2\tilde{k}^2\right] \leq (u^*)^{-2}2^{-\tilde{k}}\sum_{l=0}^{\tilde{k}}\binom{\tilde{k}}{l}\left(\frac{\tilde{k}}{l+1}\right)^2 \xrightarrow{\tilde{k}\to\infty} (u^*)^{-2}f(1/2) = 4(u^*)^{-2}$$

for the function $f(x) = \min(\varepsilon^{-2}, x^{-2})$ for an $\varepsilon < 2^{-1}$ due to the approximating property of the Bernstein polynomials at $x = \frac{1}{2}$ (see Feller 1966, Chapter VII, Section 2, Theorem 1). Since $\tilde{k} = \lfloor kN^{-1}\rfloor$ and k are of the same asymptotic order, this proves (15) for $K = 0$. The proof for general $K \in \mathbb{N}$ is analogous. □

Proposition 1. *If Assumption 1 holds, then*

$$\nabla D_r(\bar{\xi}^k) \xrightarrow{a.s.} 0,$$

i.e., every limit point of the generated sequence is a stationary point of D_r.

Proof. For the purpose of this proof, we write D instead of D_r and define the short-hand notation $\nabla_i D(\bar{\xi}) := \nabla_{\bar{\xi}_i} D(\bar{\xi})$, i.e., the derivative of D with respect to the components of $\bar{\xi}_{ti}$ at the point $\bar{\xi}$.

For an $I \subseteq \{1,\dots,N\}$, we define the event

$$\Omega_I = \{\omega \in \Omega : i_k^*(\omega) = i \text{ infinitely often }, \forall i \in I\}.$$

Since $\Omega = \bigcup_{I\subseteq\{1,\dots,N\}} \Omega_I$, showing the result for arbitrary Ω_I proves the result. We will therefore in the following focus on a fixed Ω_I.

Note that for $i \notin I$, $\mathbb{P}[\Gamma_i^k] \xrightarrow{k\to\infty} 0$ and $\bar{\xi}_i^k$ converges almost surely by definition. In the first part of the proof, we will show that $\bar{\xi}_i^k$ also converges for all $i \in I$. For that purpose, we will consider updates for iterations $k \geq k_0$, where k_0 is the last iteration with an update of a cell $i \notin I$.

The recursion in (12) can be re-written as

$$\bar{\xi}_i^{k+1} = \begin{cases} \bar{\xi}_i^k + H_i^k(Z_{ki} - \nabla_i D(\bar{\xi}_t^k)), & i = i_k^* \\ \bar{\xi}_{ti}^k, & \text{otherwise,} \end{cases}$$

where Z_{ki} is defined as

$$Z_{ki} = \mathbb{1}_{\{i_k^*\}}(i)\left(\nabla_i D(\bar{\xi}^k) - \operatorname{sign}(\bar{\xi}_i^k - \tilde{\xi}^k)\cdot|\bar{\xi}_i^k - \tilde{\xi}^k|^{r-1}\right).$$

Hence, $Z_{ki} \in \mathbb{R}^d$ represents the deviation of the update from the *true* gradient $\nabla_i D(\bar{\xi}^k)$, which is 0 in expectation.

Denoting by L the Lipschitz constant of ∇D with respect to $||\cdot||_2$ (see Lemma 1), we use the mean value theorem to show for some $\theta \in [0,1]$

$$\begin{aligned} D(\bar{\xi}^{k+1}) &= D(\bar{\xi}_t^k) + \nabla_{i_k^*} D(\bar{\xi}_t^k + \theta(H_{i_k^*}^k Z_{ki_k^*} - H_{i_k^*}^k \nabla_{i_k^*} D(\bar{\xi}_{ti_k^*}^k)))^\top (H_{i_k^*}^k Z_{ki_k^*} - H_{i_k^*}^k \nabla_{i_k^*} D(\bar{\xi}_{ti_k^*}^k)) \\ &\leq D(\bar{\xi}^k) + \nabla_{i_k^*} D(\bar{\xi}^k)^\top (H_{i_k^*}^k Z_{ki_k^*} - H_{i_k^*}^k \nabla_{i_k^*} D(\bar{\xi}_t^k)) + L||H_{i_k^*}^k Z_{ki_k^*} - H_{i_k^*}^k \nabla_{i_k^*} D(\bar{\xi}^k)||_2^2 \\ &\leq D(\bar{\xi}^k) + H_{i_k^*}^k \nabla_{i_k^*} D(\bar{\xi}^k)^\top Z_{ki_k^*} - \nabla_{i_k^*} D(\bar{\xi}^k)^\top H_{i_k^*}^k \nabla_{i_k^*} D(\bar{\xi}^k) \\ &\quad + 2L(Z_{ki_k^*}^\top (H_{i_k^*}^k)^2 Z_{ki_k^*} + \nabla_{i_k^*} D(\bar{\xi}^k)^\top (H_{i_k^*}^k)^2 \nabla_{i_k^*} D(\bar{\xi}^k)), \end{aligned} \tag{17}$$

where the first inequality follows from Lipschitz continuity and the second one from $(a-b)^2 \leq 2(a^2+b^2)$.

Note that by the second statement in Lemma 3, for an arbitrary $\varepsilon > 0$, we can choose a k large enough such that $H_{i_*^k}^k < \varepsilon$, which implies

$$\mathbb{E}\left[\nabla_{i_k^*} D(\bar{\xi}^k)^\top (H_{i_k^*}^k)^2 \nabla_{i_k^*} D(\bar{\xi}^k) | \mathcal{F}_{k-1}\right] \leq \varepsilon \mathbb{E}\left[\nabla_{i_k^*} D(\bar{\xi}^k)^\top H_{i_k^*}^k \nabla_{i_k^*} D(\bar{\xi}^k) | \mathcal{F}_{k-1}\right].$$

Choosing $\varepsilon < (4L)^{-1}$, taking conditional expectations in (17), and using $\mathbb{E}[Z_{ki_k^*}|\mathcal{F}_{k-1}] = 0$, we obtain

$$\begin{aligned}\mathbb{E}[D(\bar{\xi}^{k+1})|\mathcal{F}_{k-1}] &\leq D(\bar{\xi}^k) - \mathbb{E}[\nabla_{i_k^*} D(\bar{\xi}^k)^\top H^k_{i_k^*} \nabla_{i_k^*} D(\bar{\xi}^k)|\mathcal{F}_{k-1}] \\ &\quad + 2L\,\mathbb{E}\left[\nabla_{i_k^*} D(\bar{\xi}^k)^\top (H^k_{i_k^*})^2 \nabla_{i_k^*} D(\bar{\xi}^k) + Z^\top_{ki_k^*} (H^k_{i_k^*})^2 Z_{ki_k^*}|\mathcal{F}_{k-1}\right] \\ &\leq D(\bar{\xi}^k) - 2^{-1}\mathbb{E}\left[\nabla_{i_k^*} D(\bar{\xi}^k)^\top H^k_{i_k^*} \nabla_{i_k^*} D(\bar{\xi}^k) + 2LZ^\top_{ki_k^*} (H^k_{i_k^*})^2 Z_{ki_k^*})|\mathcal{F}_{k-1}\right]. \quad (18)\end{aligned}$$

Integrating the squared j-th component of Z_{li}, we obtain for $l > k$

$$\begin{aligned}\mathbb{E}[Z^2_{lij}|\mathcal{F}_k] &= \mathbb{E}\left[\mathbb{1}_{\Gamma_i^{l-1}}\left(\int_{\Gamma_i^{l-1}} \operatorname{sign}(\bar{\xi}^l_i - \tilde{\xi}^l)\cdot|\bar{\xi}^l_i - \tilde{\xi}^l|^{r-1}\,\mathbb{P}(d\xi) - \operatorname{sign}(\bar{\xi}^l_i - \tilde{\xi}^l)\cdot|\bar{\xi}^l_i - \tilde{\xi}^l|^{r-1}\right)^2\right] \\ &\leq \left(\int |\bar{\xi}^l_{ij} - \tilde{\xi}^l|^{r-1}\,\mathbb{P}(d\xi)\right)^2 + \int |\bar{\xi}^l_{ij} - \tilde{\xi}^l|^{2r-2}\,\mathbb{P}(d\tilde{\xi}) \leq 2\int |\bar{\xi}^l_{ij} - \tilde{\xi}^l|^{2r-2}\,\mathbb{P}(d\tilde{\xi}) \\ &\leq 2^{2r-2}\left(\int |\bar{\xi}^l_{ij}|^{2r-2}\,\mathbb{P}(d\tilde{\xi}) + \int |\tilde{\xi}^l|^{2r-2}\,\mathbb{P}(d\tilde{\xi})\right) \leq B < \infty \quad (19)\end{aligned}$$

since $\xi \in \mathcal{L}^{2r-2}(\mathbb{R}^d)$ and because of Lemma 2. Essentially the same argument can be used to show that Z^2_{lij} are uniformly integrable.

It follows from Lemma 3 that $(H^k_{j^*})^2 = O_p(k^{-2})$ component-wise, which together with (19) implies

$$\sum_{k=K}^{\infty} Z^\top_{ki_k^*} (H^k_{i_k^*})^2 Z_{ki_k^*} \xrightarrow{K\to\infty} 0 \text{ in probability.} \quad (20)$$

Uniform integrability of $Z^\top_{ki_k^*} (H^k_{i_k^*})^2 Z_{ki_k^*}$ and (19) implies that

$$S_K = 2L\sum_{k=1}^{K} \mathbb{E}[Z^\top_{ki_k^*} (H^k_{i_k^*})^2 Z_{ki_k^*}|\mathcal{F}_{k-1}]$$

converges in $\mathcal{L}^1(\mathbb{R})$. Using the definition of S_k and (18), we write

$$\begin{aligned}\mathbb{E}[D(\bar{\xi}^{k+1}) - S_k|\mathcal{F}_{k-1}] &= \mathbb{E}[D(\bar{\xi}^{k+1})|\mathcal{F}_{k-1}] - S_k \quad (21) \\ &\leq D(\bar{\xi}^k) - 2^{-1}\mathbb{E}[\nabla_{i_k^*} D(\bar{\xi}^k)^\top H^k_{i_k^*} \nabla_{i_k^*} D(\bar{\xi}^k)|\mathcal{F}_{k-1}] - S_{k-1} \\ &\leq D(\bar{\xi}^k) - S_{k-1}. \quad (22)\end{aligned}$$

Since S_{k-1} is $\mathcal{L}^1(\mathbb{R})$-bounded and $D \geq 0$, it follows that $(D(\bar{\xi}^{k+1}) - S_k)$ is a bounded supermartingale, which converges almost surely by the supermartingale convergence theorem (see Billingsley 1986, Theorem 35.4) and so does $D(\bar{\xi}^k)$, showing the first part of the proposition.

Note that, as above, (22) can be re-written as

$$\mathbb{E}[D(\bar{\xi}^{k+1}) - S_k + T_k|\mathcal{F}_{k-1}] \leq D(\bar{\xi}^k) - S_{k-1} + T_{k-1}$$

with $T_K = 2^{-1}\sum_{k=1}^{K}\mathbb{E}[\nabla_{i_k^*}D(\bar{\xi}^k)^\top H_{i_k^*}^k \nabla_{i_k^*}D(\bar{\xi}^k)|\mathcal{F}_{k-1}]$, which implies that T_K converges almost surely by another application of the supermartingale convergence theorem. Similarly, since

$$\begin{aligned}\mathbb{E}\left[\sum_{k=1}^{K} H_{i_k^*}^k Z_{ki_k^*}|\mathcal{F}_{K-1}\right] &= \sum_{k=1}^{K-1} H_{i_k^*}^k Z_{ki_k^*} + \mathbb{E}[H_{i_K^*}^K Z_{Ki_K^*}|\mathcal{F}_{K-1}] \\ &\leq \sum_{k=1}^{K-1} H_{i_k^*}^k Z_{ki_k^*} + H_{i_K^*}^{K-1}\mathbb{E}[Z_{Ki_K^*}|\mathcal{F}_{K-1}] \\ &= \sum_{k=1}^{K-1} H_{i_k^*}^k Z_{ki_k^*},\end{aligned}$$

$U_K = \sum_{k=1}^{K} H_{i_k^*}^k Z_{ki_k^*}$ is a bounded supermartingale and therefore converges almost surely.

Since T_K converges and $\sum_{k=1}^{\infty} H_{i_k^*}^k = \infty$ almost surely by the first point in Lemma 3, $\liminf_k ||\nabla D(\bar{\xi}^k)||_2 = 0$ almost surely. Suppose there is a set $A \subseteq \Omega$ with $\mathbb{P}[A] > 0$ and an $\varepsilon > 0$, such that $\limsup_k ||\nabla D(\bar{\xi}^k)||_2 > \varepsilon$. For a $\omega \in A$, fix a K_0 such that

$$||U_k - U_{k'}||_2^2 \leq \frac{\varepsilon}{8L^2}, \quad \forall k, k' \geq K_0, \quad \sum_{k \geq K_0} \nabla_{i_k^*}D(\bar{\xi}^k)^\top H_{i_k^*}^k \nabla_{i_k^*}D(\bar{\xi}^k) \leq \frac{\varepsilon}{8L^2}.$$

Fix a $k^* > K_0$ such that $||\nabla D(\bar{\xi}^{k^*})||_2 > \varepsilon$ and $\tilde{k}$ the largest integer smaller than k^* such that $||\nabla D(\bar{\xi}^{\tilde{k}})||_2 \leq \frac{\varepsilon}{2}$, which exists because $\liminf_k ||\nabla D(\bar{\xi}^k)||_2 = 0$, then

$$\begin{aligned}||\nabla D(\bar{\xi}^{k^*}) - \nabla D(\bar{\xi}^{\tilde{k}})||_2^2 \leq L^2||\bar{\xi}^{k^*} - \bar{\xi}^{\tilde{k}}||_2^2 &\leq L^2\left\|\sum_{k=\tilde{k}}^{k^*} H_{i_k^*}^k \nabla_{i_k^*}D(\bar{\xi}^k) - H_{i_k^*}^k Z_{k,i_k^*}\right\|_2^2 \\ &\leq 2L^2\sum_{j=1}^{d}\sum_{k=\tilde{k}}^{k^*}\left(H_{i_k^*,j}^k \nabla_{i_k^*,j}D(\bar{\xi}^k)\right)^2 + 2L^2||U_{k^*} - U_{\tilde{k}}||_2^2 \\ &\leq 2L^2\sum_{k=\tilde{k}}^{k^*}\nabla_{i_k^*}D(\bar{\xi}^k)^\top H_{i_k^*}^k \nabla_{i_k^*}D(\bar{\xi}^k) + \frac{\varepsilon}{4} \leq \frac{\varepsilon}{2}\end{aligned}$$

which is a contradiction and thereby establishes that $\limsup_k ||\nabla D(\bar{\xi}^k)||_2 = 0$ almost surely, proving the second part of the proposition. □

2.3. Building Scenario Lattices

As established in the previous section, the iterative updates in (12) and (13) yield a discretization of the unconditional distribution of ξ_t with N_t-points. In this section, we outline how this method can be used to build a scenario lattice.

Algorithm 1 provides a listing of the individual steps. The algorithm first uses the second-order learning method to learn locations of nodes that minimize the Wasserstein distance between states of the Markov process and states associated with lattice nodes (lines 1 – 10). Note that the algorithm described in the previous section does not provide conditional distributions of transitioning from a node in stage t to its successor in stage $t+1$. In a second step

1 Set number of nodes N_t for stages $t \in [T]$
2 Draw sample paths $(\tilde{\xi}_1^{-n}, \ldots, \tilde{\xi}_T^{-n})_{n \in [\max_t N_t]}$ from process ξ
3 $\bar{\xi}_{tn}^1 \leftarrow \tilde{\xi}_t^{-n}$, $n \in [N_t]$, $t \in [T]$
4 $H_{tn}^1 \leftarrow \operatorname{diag}((1, \ldots, 1)^\top)$, $n \in [N_t]$, $t \in [T]$
5 **for** $k \leftarrow 1$ **to** K **do**
6 | Draw sample path $\tilde{\xi}_1^k, \ldots, \tilde{\xi}_T^k$ from process ξ
7 | **for** $t \leftarrow 1$ **to** T **do**
8 | | Update centers according to (12) and (13)
9 | **end**
10 **end**
11 **for** $t \leftarrow 1$ **to** $T-1$ **do**
12 | **for** $n \leftarrow 1$ **to** N_t **do**
13 | | Draw L state transitions $\tilde{\xi}_{t+1}^l | \bar{\xi}_{tn}$, $l = 1, \ldots, L$
14 | | Estimate transition probabilities p_{tnm} according to (23)
15 | **end**
16 **end**

Algorithm 1: Algorithm for scenario lattice generation.

(lines 11 – 16), with nodes fixed, we therefore estimate transition probabilities between nodes of successive layers.

More specifically, let $\bar{\xi}_{tn}^K$ denote nodes that result from (12) after a predefined $K \in \mathbb{N}$ iterations. Denote $p_t,\ t \in [T-1]$, as the $|N_t| \times |N_{t+1}|$ transition matrix between layers t and $t+1$ with elements p_{tnm}. To estimate transition probabilities, we draw a sample of L state transitions for each node $(\tilde{\xi}_{t+1}^l | \bar{\xi}_{tn})_{l=1}^L$, $n = 1, \ldots, N_t$, $t = 1, \ldots, T-1$ and then count the frequency of transitions to cells in the Voronoi diagram Γ_{t+1}. The corresponding maximum likelihood estimate of the (conditional) probability of a state transition from $\bar{\xi}_{tn}$ to $\bar{\xi}_{t+1,m}$ is thus given by

$$p_{tnm} = L^{-1} \sum_{l=1}^{L} \mathbb{1}_{\Gamma_{tm}}(\tilde{\xi}_{t+1}^l | \bar{\xi}_{tn}^K),\ n \in [N_t],\ m \in [N_{t+1}],\ t \in [T-1]. \tag{23}$$

Bronstein et al. (2010) refer to this method as *spray method* and find that it offers a good trade-off between computational speed and accuracy. A welcome property of the spray method is that transition probabilities can be easily estimated in parallel as the loops in lines (11) and (12) do not share information.

Remark 3. As discussed in Remark 2, every single update in line 8 of the algorithm is computationally inexpensive and scales linearly with the number of dimensions of the stochastic process d. Analyzing Algorithm 1, it becomes clear that overall the proposed quasi-Newton method has a runtime complexity linear in d, the number of stages, and the number of updates.

As we will demonstrate in Section 3, in practice the computational effort of generating a lattice using Algorithm 1 is small as millions of updates can be performed in a matter of seconds on a standard computer.

Löhndorf and Wozabal (2021) propose a method that recursively shifts the location of nodes of a lattice such that the conditional expectation of each node's successor nodes exactly matches the conditional expectation of that node's next state under the true process. Correcting all nodes of the lattice in this fashion is also possible for the lattices generated in Algorithm 1 and ensures that the lattice is a martingale process, which can be important for discretizing price processes.

3. Numerical Results

In this section, we test the proposed method of scenario lattice generation using four examples: a single-period newsvendor problem, a multi-period newsvendor problem, a multi-period and multi-product inventory planning problem, and a gas storage valuation problem. The problems are chosen such that a true optimal solution for the respective underlying continuous processes can be calculated or at least closely approximated, which enables a direct assessment of the error introduced by the discretization of the process to a scenario lattice.

To generate lattices, we have to decide on the number of nodes per stage, the number of samples to fit the lattice, and the order r of the Wasserstein distance.

Existing theory offers limited guidance in selecting the appropriate number of nodes for a specific problem, as theoretical bounds linking solution quality to approximation quality are quite loose and therefore not suitable to choose the number of nodes for concrete problems. Therefore, in all numerical experiments, we vary the number of nodes and evaluate its impact on the optimality gap. This approach closely aligns with real-world practices, where users must balance solution quality and computation time.

Determining how to distribute the number of nodes across stages is a more subtle question. To the best of our knowledge, there are no theoretical results that address this issue in either scenario tree or lattice literature. Two distinct (and contradictory) approaches to this topic exist:

1. Distribute nodes in proportion to the stochastic process variability across stages. For instance, Bally and Pagès (2003) develop lattices for optimal stopping problems and suggest allocating the number of nodes in proportion to the distortion of the unconditional distributions (refer to Eq (41) in Section 3.3.2). This method typically results in a higher number of nodes in the problem's later stages.

2. Use the decision's importance as guiding principle. Early decisions have a more significant impact on the implemented solution, leading to higher bounds for the value functions of early stages compared to later stages. This consideration supports allocating more nodes in the earlier stages.

We experimented with varying the nodes and, in particular, implemented the method proposed by Bally and Pagès (2003), but observed no substantial benefit for the considered problems. As a result, we decided not to investigate this aspect further and opted for a constant number of nodes per stage across all problems.

Regarding the number of samples in Algorithm 1, we set $L = 1000$ and observed no noticeable improvement from increasing this value. For K, we consistently selected $K = 1000 \times N_t$, which implies that, on average, there are 1000 samples for each state transition.

We assess the benefit of using second-order information to control the learning rate by comparing the proposed method with a standard stepsize rule typical of stochastic gradient descent methods. For our benchmark, we use the same samples $\tilde{\xi}_t^k$ as for the proposed quasi-Newton method and learn the centers by the following update rule

$$\bar{\xi}_{ti}^{k+1} = \begin{cases} \bar{\xi}_{ti}^k - \alpha^k \left(\operatorname{sign}(\bar{\xi}_{ti}^k - \tilde{\xi}_t^k) \cdot |\bar{\xi}_{ti}^k - \tilde{\xi}_t^k|^{r-1} \right), & \text{for } i = \arg\min_j \, ||\bar{\xi}_{ti}^k - \tilde{\xi}_t^k||_r, \\ \bar{\xi}_{ti}^k, & \text{otherwise,} \end{cases} \tag{24}$$

where we choose the learning rates α_k according to

$$\alpha_k = \frac{C_1}{C_2 + k^\beta}, \tag{25}$$

(e.g. Pagès and Printems 2003, Pflug and Pichler 2015). Assuming $C_1 > 0$, $C_2 > 0$, and $0 < \beta \leq 1$, the rule fulfills the Robbins-Monro conditions $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 < \infty$ that are necessary for convergence. In accordance with Pflug and Pichler (2015), we set $C_1 = 1$, $C_2 = 30$ and $\beta = 3/4$ for our numerical tests. Furthermore, we use (23) to estimate transition probabilities between nodes on consecutive stages (see Section 2.3).

We note that in line with Remark 3 the computation associated with the generation of the scenario lattices is negligible in all discussed problems: Even for the largest models in Section 3.3 with 100 stages and 100 nodes per stage the computation time for the quasi-Newton method described in Algorithm 1 was only 36 – 44 seconds as compared with the 485 – 508 seconds it took to solve the corresponding optimization problems that use these lattices. Also, the computation times do not significantly exceed those of the gradient descent benchmark.

3.1. Single-period Newsvendor

For our first experiment, we consider a simple single-period, multi-product newsvendor problem with uniformly distributed demand. This allows us to study how well the quasi-Newton method proposed in Section 2.2 discretizes multivariate distributions.

3.1.1. Stochastic programming model. Denote $b \in \mathbb{R}^d$ as the number of newspapers of each kind purchased at costs $c \in \mathbb{R}^d$, $s \in \mathbb{R}^d$ as the number of newspapers sold at prices $p \in \mathbb{R}^d$, and $D : \Omega \to \mathbb{R}^d$ as the random demand.

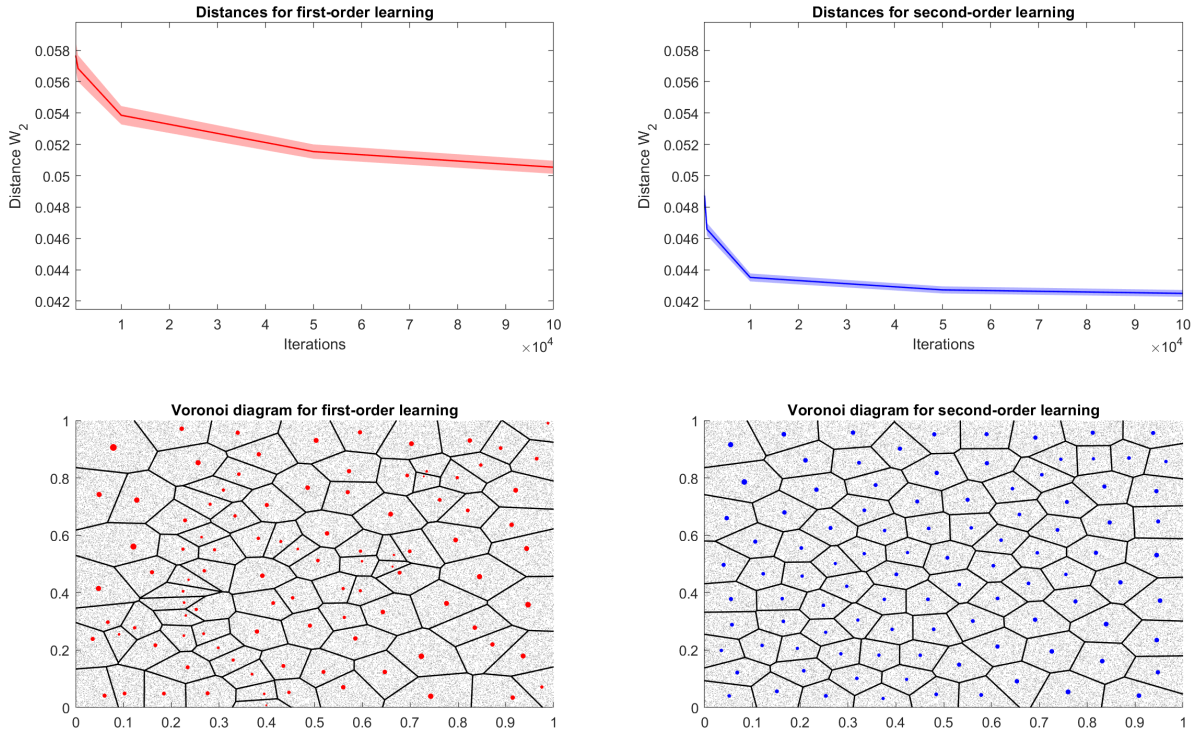


Figure 2 Comparison of first-order gradient descent and the quasi-Newton method; top: Wasserstein distances as a function of sample size (bold = average, shaded area = confidence interval); bottom: partition of outcomes space as Voronoi diagrams (gray = sampled points, red/blue = centers scaled by probability mass).

The newsvendor problem can be written as the following two-stage stochastic program,

$$\begin{aligned} \max_{b,s\geq 0} \; & -\langle c, b\rangle + \mathbb{E}[\langle p, s\rangle] \\ \text{s.t.} \quad & s \leq b,\; s \leq D \quad \text{a.s.} \end{aligned}$$

As there are no constraints that connect the individual buying (selling) decisions for the different types of newspapers, the problem can be decoupled into d individual decision problems.

It follows from elementary calculus (see Birge and Louveaux 2011) that the optimal solution and optimal value of the problem are

$$b_i^* = F_i^{-1}\left(\frac{p_i - c_i}{p_i}\right), \quad \Pi^* = -c_i b_i^* + p_i \int_{-\infty}^{\infty} \min(x, b_i^*)\, dF_i(x), \quad \forall i \in [d], \tag{26}$$

where F_i and F_i^{-1} are the distribution function and the inverse distribution function of the demand D_i for product i.

3.1.2. Comparison of learning methods. For ease of exposition, we focus on the two-dimensional case which lends itself to visual inspection. Further, to be able to assess the quality of the discretization, we fix the order of the Wasserstein distance to $r = 2$ for which properties of the optimal quantization are known for uniform distributions.

Figure 2 plots the Wasserstein distance as well as the Voronoi diagrams of a discretization for a sample of $K = 100,000$ common random variates drawn from a bivariate uniform distribution on $[0,1] \times [0,1]$. The two top panels graph the distance of $\bar{\xi}$ with 100 nodes to the true distribution, whereby $W_2(\xi, \tilde{\xi})$ is approximated by the average distance between the sample points and the atoms of $\bar{\xi}$. The bold line indicates the average distances over 30 repetitions of the experiment while the shaded area indicates the estimated normal 95% confidence interval. Clearly, the quasi-Newton method converges to distributions $\bar{\xi}$, which are on on average approximately 10-15% closer to ξ than the results obtained with the gradient descent method. Furthermore, we observe that the confidence intervals of the quasi-Newton method are significantly narrower than those of the gradient descent method, indicating a more stable convergence pattern.

The two panels on the bottom of Figure 2 show the Voronoi diagrams for one exemplary discretization. Each red and blue dot marks the center of a Voronoi cell with its size proportional to the estimated mass of the center. Newman (1982) show that the optimal quantization of the uniform distribution on a square results in a tessellation of regular hexagons. As can be seen from the bottom right panel, the quasi-Newton method produces a partition that fits this pattern nearly perfectly. The gradient descent method, by contrast, results in a rather irregular partition consisting of polygons with a varying number of faces, which is quite far from the optimum.

This finding is further supported by measuring the dispersion of the probabilities of the centers by the Gini index of the resulting discrete distributions. The theoretical results in Newman (1982) show that an optimal discretization leads to a uniform distribution on the centers and therefore a Gini index close to 0. In our experiment, the quasi-Newton method achieves a Gini index of 0.13, while the first-order method results in a distribution with a substantially higher Gini index of 0.25.

Figure 3 plots objective values for $r = 2$ and $r = 3$ for the newsvendor model with $p = 1$ and $c = 0.9$ for either learning method. Again, bold lines indicate averages and shaded areas indicate 95% confidence intervals.

Both methods lead to approximations that over-estimate expected profits, which is in accordance with theory on discretizations of stochastic programs (e.g., Birge and Louveaux 2011, Section 8.2). Although both methods yield close estimates of the objective value for this simple problem, for $r = 2$ the quasi-Newton method results in an optimality gap which is roughly half of that of the gradient descent method while exhibiting less variation in the estimate. For $r = 3$, both methods fare slightly worse, albeit with lower optimality gaps for the quasi-Newton method.

3.2. Multi-period Newsvendor

Since the problem treated in the last section is a two-period stochastic programming problem, the randomness can be represented by a distribution without any dependence on time. To

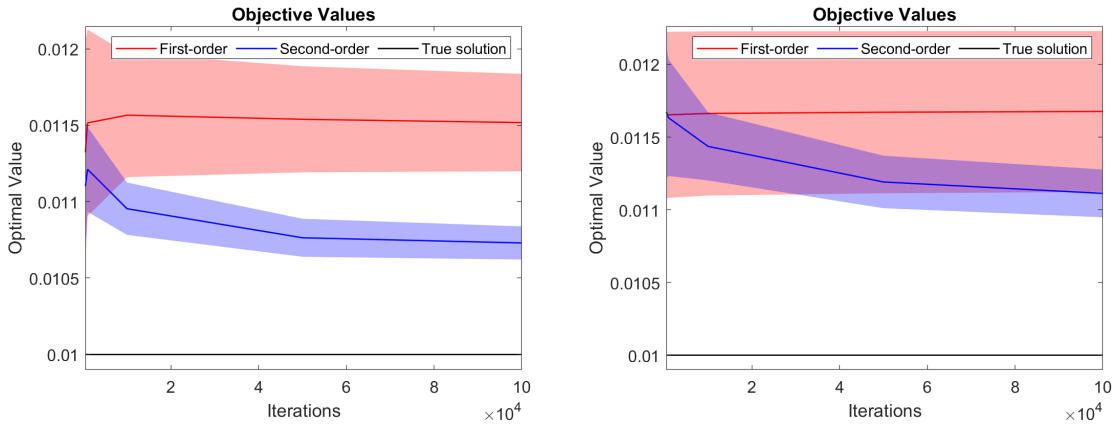


Figure 3 Optimal objective values for $r = 2$ (left) and $r = 3$ (right) right for stochastic gradient descent and the quasi-Newton method compared to the optimal values as a function of the number of training samples.

study the effectiveness of the quasi-Newton method for discretizing non-degenerate stochastic processes, we study a multi-stage extension of the problem. More specifically, we model a multi-period, multi-product newsvendor problem, where demand follows a random walk with normally distributed innovations. As with the single period model, the advantage of using this model is that we can calculate the optimal objective value analytically, which enables us to accurately measure the quality of the discretization.

3.2.1. Stochastic programming model. For the multi-stage case, we assume that demand follows a multivariate random walk,

$$D_t = D_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \operatorname{diag}((\sigma^2, \dots, \sigma^2)^\top)), \tag{27}$$

for $t = 1, ..., T$, with identical initial demand $D_{0i} = \mu$, $i \in [d]$, where d is the number of products, and $\text{cv} = \frac{\sigma}{\mu}$ is the coefficient of variation.

The corresponding multi-period, multi-product newsvendor model is given by the multi-stage stochastic program,

$$\begin{array}{ll} \max\limits_{b \geq 0, s} & \mathbb{E}\left[\sum_{t=1}^{T} \langle p, s_t \rangle - \langle c, b_t \rangle\right] \\ \text{s.t.} & s_t \leq b_{t-1},\ s_t \leq D_t,\ t = 1, \dots, T \quad \text{a.s.}, \end{array}$$

with $b_0 = 0$. Note that we allow sales s_t to be free variables to deal with negative demand, which has a positive probability for the arithmetic random walk (27).

To reduce the number of model parameters, we define each instance of the problem formulation by its critical fractile,

$$\text{cf} = \frac{p - c}{p},$$

which is the probability of the quantile that equals the optimal order quantity.

3.2.2. Analytical solution. The above variant of the newsvendor model has two attractive properties: (1) it can be decomposed by product, so that the joint optimal solution is merely the union of optimal solutions for each product; and (2) the optimal profit can be obtained analytically for a given product demand D_{ti}, $i \in [d]$, $t \in [T]$.

Denote ϕ as the density of the standard normal distribution Φ. Since the demand process is a martingale, D_{ti} is the expected value for demand in $D_{t+1,i}$. Then, the optimal expected newsvendor profit for normally distributed demand for product i is given by

$$(p-c)D_{ti} - p\sigma\phi\left(\Phi^{-1}(\mathrm{cf})\right). \tag{28}$$

To obtain the multi-stage profit, we calculate the expected profit for all possible $D_0, \ldots, D_{T-1}$,

$$\mathbb{E}\left[\sum_{t=1}^{T-1}\left((p-c)D_{ti} - p\sigma\phi\left(\Phi^{-1}(\mathrm{cf})\right)\right)\right] = (T-1)\left(D_{0i} - p\sigma\phi\left(\Phi^{-1}(\mathrm{cf})\right)\right). \tag{29}$$

Since we assume initial demand to be identical for all products, the total expected profit is given by

$$d(T-1)\left(\mu - p\sigma\phi\left(\Phi^{-1}(\mathrm{cf})\right)\right). \tag{30}$$

This expression enables us to calculate the optimal objective value of the problem and compare it to the solution obtained when using the discrete lattice.

3.2.3. Comparison of learning methods. For our computational experiment, we fix $p = 1$, $\mathrm{cf} \in \{0.1, 0.2, 0.5\}$, $d \in \{2, 5, 20\}$, $T \in \{5, 20, 100\}$, $D_{0i} = \mu = 10$, $i \in [d]$, and $\mathrm{cv} \in \{0.1, 0.2, 0.5\}$. Furthermore, we discretize the process to a scenario lattice with a constant number of nodes per stage, $N_t \equiv N \in \{10, 100\}$, $t = 2, \ldots, T$ with $L = 1000$, $K = LN_t$.

Table 1 reports the optimality gap for different instances of the multi-period, multi-product newsvendor problem for stochastic gradient descent (SGD) and the quasi-Newton (QN) method for orders $r \in \{2, 4\}$. The lowest gap for each instance is indicated in bold face. The standard error based on 10 repetitions is given in parentheses.

We can see that the quasi-Newton method achieves the lowest gap in all instances with the differences ranging roughly between 2% and 20%. Clearly, the harder instances with more stages or a higher coefficient of variation benefit more from the better quality of approximation provided by the quasi-Newton method. We also note that the gradient descent method fails to produce lattices for $r = 4$ in 12 out of 18 instances due to numerical problems. We attribute this to the scaling of the gradient by H_t which in case of $r = 4$ involve relatively large numbers.

3.3. Multi-product Inventory Control

The previous two models provide closed-form solutions and consequently can be solved without resorting to stochastic programming methods. We now study a variant of the previous problem

				Order	$r=2$		$r=4$	
				Method	SGD	QN	SGD	QN
T	d	CoefVar	CritFrac	Nodes				
5	5	0.20	0.20	10	8.23 (0.26)	7.25 (0.18)	n/a	**5.01** (0.38)
				100	0.83 (0.18)	**0.23** (0.05)	1.85 (0.12)	1.71 (0.09)
20	2	0.20	0.20	10	6.46 (1.36)	**4.10** (0.84)	n/a	6.78 (0.93)
				100	0.72 (0.23)	1.67 (0.33)	1.15 (0.22)	**1.02** (0.20)
	5	0.10	0.20	10	8.57 (0.38)	5.05 (0.24)	n/a	**4.71** (0.39)
				100	2.70 (0.25)	**1.37** (0.10)	1.63 (0.12)	2.30 (0.13)
		0.20	0.10	10	23.45 (0.55)	**13.44** (0.60)	n/a	13.52 (0.73)
				100	5.94 (0.38)	**5.86** (0.33)	12.09 (0.24)	8.86 (0.27)
			0.20	10	18.78 (0.51)	**10.42** (0.65)	n/a	10.46 (0.77)
				100	4.62 (0.49)	**3.86** (0.29)	9.10 (0.31)	5.55 (0.38)
			0.50	10	9.30 (0.62)	5.88 (0.69)	n/a	**5.28** (0.78)
				100	2.81 (0.38)	**1.33** (0.20)	3.72 (0.15)	2.36 (0.32)
		0.40	0.20	10	40.49 (1.25)	**25.18** (1.40)	n/a	27.04 (1.80)
				100	17.04 (2.33)	**13.83** (1.10)	n/a	19.26 (1.61)
	20	0.20	0.20	10	23.62 (0.24)	15.94 (0.33)	n/a	**14.99** (0.34)
				100	23.02 (0.22)	**4.34** (0.29)	n/a	5.90 (0.22)
100	5	0.20	0.20	10	28.45 (0.80)	**17.63** (1.47)	n/a	22.96 (2.80)
				100	23.68 (0.85)	4.86 (0.67)	n/a	**4.42** (0.85)

Table 1 Optimality gap as percentage of the optimal value for different model instances and method parameter settings. The numbers in gaps are the standard errors based on 10 repetitions.

which includes an inventory balance equation, more specifically a multi-product inventory control problem with lost sales for which there are neither analytical solutions nor good heuristic policies.

Inventory models with Markovian demand are discussed, for example, in Beyer et al. (2010), and solution approaches typically require demand to be modeled as a discrete Markov chain. As scenario lattices are effectively Markov chains, our framework provides an efficient approach to discretize Markovian demand processes to Markov chains. To solve these problems, Beyer et al. (2010) discusses dynamic programming solutions over discrete inventory states. Since this is prohibitive in higher dimensions, we instead resort to stochastic dual dynamic programming (SDDP) (Pereira and Pinto 1991). Since SDDP is a natural choice for problems where randomness is modeled by a scenario lattices, this example also demonstrates how the general class of multi-stage stochastic programming models can benefit from the proposed method.

3.3.1. Stochastic programming model. For this problem, we assume that demand follows a first-order vector-autoregressive (VAR) model,

$$D_t = \Theta D_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \operatorname{diag}((\sigma^2, \dots, \sigma^2)^\top)), \tag{31}$$

for $t = 1, ..., T$, with identical initial demand $D_{0i} = \mu$, $i \in [d]$, where d is the number of products, $\text{cv} = \frac{\sigma}{\mu}$ is the coefficient of variation, and $\Theta \equiv \operatorname{diag}(\theta)$ is the matrix of AR(1) parameters. While

the default error distribution is a multivariate normal distribution, we test uniform errors as well as a heavy-tailed t-distribution with $df = 5$, with scaling parameters set to match the given coefficient of variation.

In each period t, the decision-maker decides about order quantities b_t, given an initial inventory level, I_{t-1}, and sells products s_t. We assume that the leadtime for ordering a product equals one period, so that demand must be satisfied from on-hand inventory. The deterministic equivalent of the multi-stage stochastic programming problem is given by

$$\max_{b,s,I \in \mathcal{X}(D_t, I_{t-1})} \mathbb{E}_{(D_1,\dots,D_T)} \left[\sum_{t=1}^{T} \langle p, s_t \rangle - \langle c, b_t \rangle \right], \tag{32}$$

with feasible set

$$\mathcal{X}(D_t, I_{t-1}) = \left\{ \begin{array}{ll} I_t = I_{t-1} + b_t - s_t, & \forall\, t = 1,\dots,T,\ a.s. \\ s_t \le I_{t-1}, & \forall\, t = 1,\dots,T,\ a.s. \\ s_t \le D_t, & \forall\, t = 1,\dots,T,\ a.s. \\ s_t \in \mathbb{R}^d, b_t \ge 0, I_t \ge 0 & \forall\, t = 1,\dots,T,\ a.s. \end{array} \right\}, \tag{33}$$

and $I_0 \equiv 0$ as the initial inventory position. As the VAR model is additive, demand can become negative. We therefore define the vector of sales, s_t, as free to allow for negative sales.

As opposed to the two previous examples, the above problem has no closed form solution. To assess the quality of the solution we compute an upper bound. In particular, since randomness only enters the model in the right hand side of the constraints and demand follows an (additive) VAR model, we can use the approach pioneered in Infanger and Morton (1996) to compute statistical upper bounds for the problem using SDDP. We calculate the bound 10 times, with 100 samples per stage and then report the average optimality gap as well the standard error. The sample average of the SAA SDDP upper bound provides an estimate of the true upper bound for each problem instance.

We then solve the multi-stage stochastic programming problem by replacing the stochastic process with a lattice obtained using the two learning methods. All scenario lattices are created with a constant number of nodes per stage, $N_t \equiv N \in \{10, 100, 1000\}$, $t = 2,\dots,T$ with $L = 1000$, $K = LN_t$. We solve the resulting optimization problem using Markov chain SDDP (Löhndorf and Shapiro 2019, Löhndorf and Wozabal 2021).

To be able to calculate an optimality gap, we run 10,000 forward simulations of the optimal policy after the algorithm converges. However, instead of sampling scenarios from the lattice, we use the original continuous stochastic process to simulate demand states in order to obtain out-of-sample results that be used to quantify the value of the policy trained on the lattice for realization of the true process. This allows us to compare the quality of the solutions for different lattice models. In particular, to obtain a solution for a given demand state of the stochastic process, we choose the value function of the nearest lattice node using the same distance measure that was used to create the lattice (see Terça and Wozabal 2020).

Having established upper and lower bounds for the optimization problem, we calculate an estimate of the optimality gap based on 10 independent model runs.

				Order / Method	SGD	$r=2$ QN	$r=4$ QN	$r=8$ QN	$r=16$ QN
T	d	Dist*	θ	Nodes					
5	10	ND	1.00	10	11.26 (0.41)	10.49 (0.44)	10.35 (0.42)	9.52 (0.47)	**8.33** (0.41)
				100	5.17 (0.13)	4.59 (0.11)	4.58 (0.12)	**4.36** (0.11)	4.60 (0.09)
20	5	ND	1.00	10	13.54 (0.50)	13.17 (0.66)	11.56 (0.57)	**10.40** (0.59)	11.14 (0.48)
				100	5.29 (0.50)	**4.76** (0.52)	4.86 (0.50)	5.29 (0.52)	6.69 (0.51)
	10	ND	-0.50	10	7.68 (1.61)	6.53 (2.81)	7.65 (1.67)	7.98 (2.19)	**4.62** (1.10)
				100	0.23 (0.02)	0.37 (0.09)	0.24 (0.02)	0.19 (0.02)	**0.17** (0.02)
			0.50	10	**4.55** (1.03)	9.93 (2.62)	9.93 (1.70)	9.86 (1.52)	8.02 (2.29)
				100	0.32 (0.10)	0.47 (0.10)	0.34 (0.11)	0.24 (0.10)	**0.19** (0.10)
			1.00	10	21.07 (0.44)	18.35 (0.47)	18.02 (0.45)	15.79 (0.39)	**13.13** (0.35)
				100	10.96 (0.33)	7.64 (0.38)	6.97 (0.35)	**6.47** (0.38)	7.06 (0.35)
		TD	1.00	10	31.17 (1.29)	31.32 (1.09)	**27.27** (1.13)	30.81 (1.18)	36.39 (1.12)
				100	22.04 (1.18)	**19.29** (1.16)	19.74 (1.18)	20.68 (1.11)	23.88 (1.17)
				1000	16.67 (1.17)	**14.16** (1.21)	14.81 (1.21)	15.34 (1.20)	17.84 (1.15)
		UD	1.00	10	41.50 (5.63)	**3.16** (0.61)	3.84 (0.83)	3.70 (0.84)	4.21 (1.11)
				100	1.30 (0.05)	0.95 (0.05)	0.85 (0.05)	0.71 (0.06)	**0.69** (0.06)
	20	ND	1.00	10	25.81 (0.55)	27.09 (0.33)	23.96 (0.48)	21.08 (0.42)	**16.86** (0.32)
				100	24.19 (0.43)	12.32 (0.41)	10.07 (0.36)	8.83 (0.34)	**8.58** (0.34)
				1000	31.74 (0.32)	7.82 (0.33)	7.12 (0.33)	**6.45** (0.33)	6.62 (0.33)
100	10	ND	1.00	10	191.77 (2.55)	53.18 (1.68)	50.37 (1.44)	48.99 (1.79)	**45.58** (1.86)
				100	272.39 (2.69)	26.24 (1.70)	26.46 (1.57)	**20.70** (1.84)	20.94 (1.82)
				1000	213.84 (2.13)	16.34 (1.87)	16.77 (1.81)	**14.04** (1.90)	16.02 (1.72)

*ND=normal distribution, UD=uniform distribution, TD=t-distribution with $df=5$

Table 2 Optimality gap (standard error) for different model instances and method parameter settings

3.3.2. Comparison of learning methods. The results of the numerical experiment are summarized in Table 2 for stochastic gradient descent (SGD) and the quasi-Newton (QN) method for orders $r \in \{2, 4, 8, 16\}$. As the SGD method was numerically unstable for $r > 2$, results are only shown for the quasi-Newton method. We report the average optimality gap (in %) from 10 independent repetitions of the experiment along with its standard error, which provides an indication of the accuracy of the measurement.

The results provide evidence that the quasi-Newton method generates lattices that lead to better policies than the gradient descent method. Overall, the optimality gap is lowest when using lattices that were constructed using a Wasserstein distance of order $r = 16$, which indicates that it may generally pay off to train lattices with distance measures that penalize larger deviations more heavily.

The gaps are particularly wide for the problem with many decision stages ($T = 100$), which only the quasi-Newton method was able to solve with reasonable accuracy (14.04% gap with 1000 nodes and order $r = 8$). The gap is still considerable, but so is the difficulty of the problem with 10 independent dimensions for the random process and 100 stages in total. Most multi-stage stochastic programming problems from the literature study significantly smaller problems.

Another notable observation is that the quasi-Newton method decreases the optimality gap more quickly as the number of nodes in the lattice increases. This observation is particularly noticeable for the high-dimensional instance ($d = 20$), where the gradient descent method hardly manages to reduce the gap as more nodes become available, whereas the quasi-Newton method achieves a sizable reduction in optimality gap. A high-dimensional stochastic process is also subject of the next test case.

3.4. Gas Storage Valuation Problem

For our last test case, we consider a gas storage valuation problem in which randomness is characterized by a high-dimensional stochastic price process. The problem has been studied in Lai et al. (2010) and Löhndorf and Wozabal (2021) who show that the rolling intrinsic policy is near optimal under the risk-neutral measure. We therefore evaluate the rolling intrinsic policy on the scenario lattice and compare it with a information relaxation upper bound computed in Lai et al. (2010) based on the original process to assess the quality of the solution. The problem is an ideal candidate to study the effectiveness of the proposed method on decision problems with high-dimensional state spaces and randomness in the objective function.

3.4.1. Multi-factor price process. The stochastic process that characterizes the dynamics of futures prices is expressed by a multifactor model, e.g., see Clewlow and Strickland (2000), Eydeland and Wolyniec (2003). At time $t = 1, \dots, T$, let $F_t = (F_{tt}, \dots, F_{tT})$ be the vector of random prices, with F_{tt} being the spot price and F_{tj} being the price of the futures contract with maturity in period $j = t+1, \dots, T$. The price process in discrete-time is defined as

$$\ln F_{tj} = \ln F_{t-1,j} + Z_{tj},\ Z_{tj} \sim \mathcal{N}(\mu, \Sigma),\ j = t, \dots, T, t = 1, \dots, T. \tag{34}$$

We will work with the drift-less version of the process, which requires $\mu = -\frac{\sigma^2}{2}$.

3.4.2. Stochastic programming model. In each period t, the decision-maker has the option to schedule injection i_{tj} and withdrawal w_{tj} for period $j \geq T$, subject to feasibility of storage positions s_{tj}, operational limits $\bar{i}$, $\bar{w}$, $\bar{s}$, as well as in-kind losses, d^i and d^w. Any update to injections and withdrawals from period $t-1$ to t for period $j \geq t$ must be offset by trading future j at price F_{tj}. As money does not change hands until futures contracts mature, future profits are discounted at rate γ_j. The deterministic equivalent of the multi-stage stochastic programming problem is given by

$$\max_{i,s,w \in \mathcal{X}(F_t, s_{t-1})} \mathbb{E}_{(F_1, \dots, F_T)} \left[\sum_{t=1}^{T} \sum_{j=t}^{T} \gamma_j F_{tj} (w_{tj} - i_{tj} - w_{t-1,j} + i_{t-1,j}) \right], \tag{35}$$

with feasible set

$$\mathcal{X}(F_t, s_{t-1}) = \left\{ \begin{array}{ll} s_{tj} = s_{t,j-1} + d^i i_{tj} - d^w w_{tj}, & \forall\, t = 1, \dots, T,\ j = t, \dots, T \\ s_{t,t-1} = s_{t-1,t}, & \forall\, t = 1, \dots, T \\ i_{tj} \in [0, \bar{i}], w_{tj} \in [0, \bar{w}], s_{tj} \in [0, \bar{s}], & \forall\, t = 1, \dots, T,\ j = t, \dots, T \end{array} \right\}, \tag{36}$$

and $s_0 \equiv 0$ as the initial storage position. See Löhndorf and Wozabal (2021) for a more detailed discussion of the model.

3.4.3. Rolling intrinsic value. We can solve the above decision problem using the so-called rolling intrinsic approach – a simple reoptimization policy that is often used in practice. Instead of solving problem (35), the rolling intrinsic policy involves solution of a series of deterministic problems. The resulting rolling intrinsic value is then given by

$$\mathbb{E}\left[\sum_{t=1}^{T} \max_{i,s,w \in \mathcal{X}(F_t, s_{t-1})} \left\{ \sum_{j=t}^{T} \gamma_j F_{tj}(w_{tj} - i_{tj} - w_{t-1,j} + i_{t-1,j}) \right\}\right]. \tag{37}$$

The rolling intrinsic policy has a two attractive properties: (1) it is known to be near-optimal under a risk-neutral measure (Lai et al. 2010); and (2) it can be directly applied to simulations of the decision process.

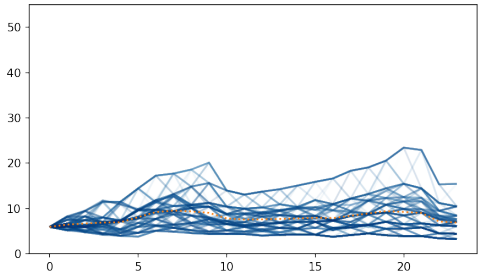
To assess discretization quality, we calculate the optimality gap between the dual upper bound provided in Lai et al. (2010) and the rolling intrinsic value obtained with the lattice process.

3.4.4. Case data. For our experiments, we use the model parameters reported in Lai et al. (2010) and Löhndorf and Wozabal (2021) who calibrate the multi-factor model using closing prices from NYMEX Henry Hub on 2006-03-01 (Spring), 2006-06-01 (Summer), 2006-09-01 (Fall), and 2006-12-01 (Winter). In line with the above papers, injection (withdrawal) limits are set to at 0.15 mmBtu/month and storage capacity to 1.0 mmBtu/month. The initial storage level is set to zero. The decision problem has 24 decision stages with monthly time increments.

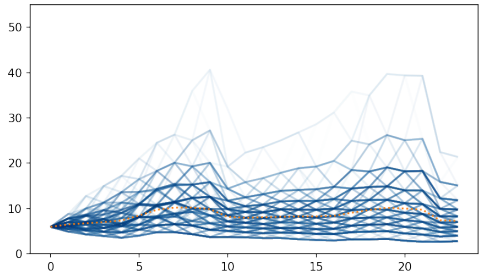
3.4.5. Comparison of learning methods. To gain a first intuition in the difference between the two learning methods, Figure 4 shows 10,000 simulated sample paths of spot price, F_{tt}, drawn from lattices with 10 and 100 nodes per stage for either method and $r = 2$. As we can see, and in line with the results in Section 3.1, the quasi-Newton method produces lattices which are more regular and exhibit a better coverage of the outcome space than the gradient descent method. For example, prices simulated with the 10-node lattice barely exceed 20 when using the gradient descent method but go up to 40 when using the quasi-Newton method.

Next, we compare the performance of the two methods by assessing the optimality gap of the rolling intrinsic value when using the discrete lattice instead of the continuous process. We discretize the multi-factor price process to a scenario lattice with a constant number of nodes per stage, $N_t \equiv N \in \{10, 100\}$, $t = 2, \ldots, T$ with $L = 1000$, $K = LN_t$.

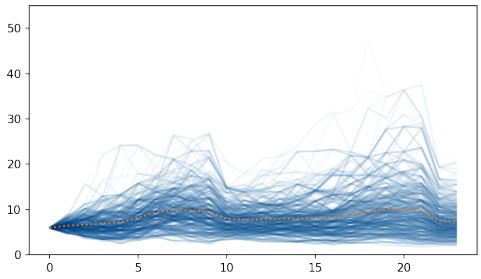
Table 3 summarizes the result of the numerical experiment for stochastic gradient descent (SGD) and the quasi-Newton (QN) method for orders $r \in \{2, 4\}$. We can see that the quasi-Newton method achieves substantially lower optimality gaps than the gradient descent method. It is worth highlighting that the proposed approach achieves optimality gaps below 3% for



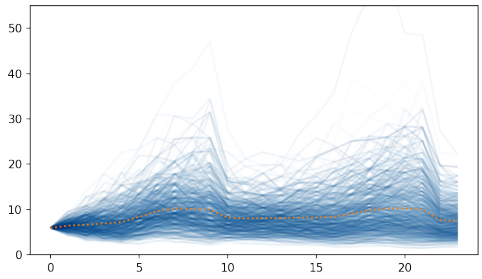
(a) 10-node lattice with gradient descent method



(b) 10-node lattice with quasi-Newton method



(c) 100-node lattice with gradient descent method



(d) 100-node lattice with quasi-Newton method

Figure 4 10,000 simulations of spot prices from the discrete lattice process. Dotted red lines mark the means and color intensity is proportional to the frequency of sampling so that the darker areas represent more probable outcomes.

	Order	$r=2$		$r=4$	
	Method	SGD	QN	SGD	QN
Season	Nodes				
Fall	10	8.15 (0.50)	11.04 (0.29)	n/a	**7.29** (0.36)
	100	10.86 (0.36)	3.05 (0.04)	15.81 (0.31)	**0.27** (0.06)
Spring	10	4.33 (0.58)	6.79 (0.34)	n/a	**2.81** (0.42)
	100	8.79 (0.29)	**0.77** (0.06)	13.05 (0.29)	1.02 (0.08)
Summer	10	3.94 (0.54)	7.09 (0.30)	n/a	**2.26** (0.32)
	100	8.45 (0.25)	**0.21** (0.04)	12.46 (0.28)	1.78 (0.05)
Winter	10	16.74 (1.87)	22.42 (0.74)	9.45 (1.85)	**6.74** (0.86)
	100	39.63 (0.70)	**1.76** (0.23)	61.68 (0.94)	4.67 (0.19)

Table 3 Optimality gap (std error) for different model instances of the gas storage valuation problem

either instance, and this is despite reducing a process with 24 dimensions to a lattice with only 100 nodes per stage. As in the case of the multi-variate newsvendor problem, for $r=4$, the gradient descent method fails to generate lattices for a number of instances due to numerical issues leading to n/a results.

Similar to the inventory problem, when the number of nodes is low ($N_t=10$), the higher-order distance measure yields the lowest gap, which confirms the finding of the previous experiment.

4. Conclusion

We propose a novel quasi-Newton method to train discrete scenario lattices that approximate continuous multi-dimensional stochastic processes and can be used to model randomness in multi-stage stochastic programs or stochastic-dynamic programs. Unlike alternative gradient descent approaches, the quasi-Newton method is parameter-free in the sense that stepsizes are chosen automatically.

Numerical experiments on a range of problems demonstrate that the new method performs markedly better than a benchmark gradient descent method. In contrast to gradient descent methods, the quasi-Newton approach is able to train lattices using higher-order probability metrics. For high-dimensional problems or those where the number of nodes per stage is relatively low, we observe that lower optimality gaps can be achieved with lattices that have been trained with higher-order metrics. While we cannot provide any theory to support this empirical result, we conjecture that the higher dispersion of discrete states over the outcome space leads to better policies.

Future work could study usage of this method on real-world multi-stage stochastic programs and explore the role of the order r of the employed Wasserstein metric on the quality of the approximation. Furthermore, it would be interesting to look into hierarchical partitions to speed-up nearest neighbor search of lattices with millions of nodes per stage.

References

Bally V, Pagès G (2003) A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli* 9(6):1003–1049.

Bardou O, Bouthemy S, Pagès G (2009) Optimal quantization for the pricing of swing options. *Applied Mathematical Finance* 16(2):183–217.

Bertsekas D (1981) Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control* 20(3):415–419.

Beyer D, Cheng F, Sethi SP, Taksar M, et al. (2010) *Markovian demand inventory models* (Springer).

Billingsley P (1986) *Probability and Measure*. Wiley Series in Probability and Statistics (Wiley).

Birge J, Louveaux F (2011) *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering (Springer New York).

Bonnans J, Cen Z, Christel T (2012) Energy contracts management by stochastic programming techniques. *Annals of Operations Research* 200(1):199–222.

Bottou L, Bengio Y (1994) Convergence properties of the k-means algorithms. *Proceedings of the 7th International Conference on Neural Information Processing Systems*, 585–592, NIPS'94 (Cambridge, MA, USA: MIT Press).

Broadie M, Glasserman P (2004) A stochastic mesh method for pricing high-dimensional american options. *Journal of Computational Finance* 7:35–72.

Bronstein AL, Pagès G, Wilbertz B (2010) How to speed up the quantization tree algorithm with an application to swing options. *Quantitative Finance* 10(9):995–1007.

Byrd RH, Hansen SL, Nocedal J, Singer Y (2016) A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization* 26(2):1008–1031.

Chourdakis K (2004) Non-affine option pricing. *The Journal of derivatives* 11(3):10–25.

Chow CS, Tsitsiklis JN (1991) An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE Transactions on Automatic Control* 36(8):898–914.

Clewlow L, Strickland C (2000) *Energy Derivatives: Pricing and Risk Management* (Lacima Publications).

Cox J, Ross S, Rubinstein M (1979) Option pricing: A simplified approach. *Journal of financial Economics* 7(3):229–263.

Dufour F, Prieto-Rumeau T (2013) Finite linear programming approximations of constrained discounted markov decision processes. *SIAM Journal on Control and Optimization* 51(2):1298–1324.

Dupačová J, Gröwe-Kuska N, Römisch W (2003) Scenario reduction in stochastic programming. *Mathematical Programming* 95(3):493–511.

Eydeland A, Wolyniec K (2003) *Energy and Power Risk Management: New Developments in Modeling, Pricing, and Hedging*. Wiley Finance (Wiley).

Felix B, Weber C (2012) Gas storage valuation applying numerically constructed recombining trees. *European Journal of Operational Research* 216(1):178–187.

Feller W (1966) *An introduction to probability theory and its applications*. Number volume 2 in Wiley mathematical statistics series (Wiley).

Graf S, Luschgy H (2000) *Foundations of Quantization for Probability Distributions*. Lecture Notes in Mathematics (Springer Berlin Heidelberg).

Hahn WJ, Dyer JS (2008) Discrete time modeling of mean-reverting stochastic processes for real option valuation. *European journal of operational research* 184(2):534–548.

Hanasusanto G, Kuhn D, Wiesemann W (2016) A comment on "computational complexity of stochastic programming problems". *Mathematical Programming* 159(1):557–569.

Harikae S, Dyer J, Wang T (2021) Valuing real options in the volatile real world. *Production and Operations Management* 30(1):171–189.

Heitsch H, Römisch W (2009) Scenario tree modeling for multistage stochastic programs. *Mathematical Programming* 118:371–406.

Homem-de Mello T, Bayraksan G (2014) Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science* 19(1):56–85.

Høyland K, Kaut M, Wallace SW (2003) A heuristic for moment-matching scenario generation. *Computational optimization and applications* 24(2):169–185.

Infanger G, Morton DP (1996) Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming* 75(2):241–256.

Kall P, Mayer J (2006) *Stochastic Linear Programming: Models, Theory, and Computation*. International Series in Operations Research & Management Science (Springer US).

Kaut M, Wallace S (2007) Evaluation of scenario generation methods for stochastic programming. *Pacific Journal of Optimization* 3(2):257–271.

Kiszka A, Wozabal D (2020) A stability result for linear Markovian stochastic optimization problems. *Mathematical Programming* .

Lai G, Margot F, Secomandi N (2010) An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations Research* 58(3):564–582.

Langen HJ (1981) Convergence of dynamic programming models. *Mathematics of Operations Research* 6(4):493–512.

LeCun Y, Bottou L, Orr GB, Müller KR (2002) Efficient backprop. *Neural networks: Tricks of the trade*, 9–50 (Springer).

Löhndorf N (2016) An empirical analysis of scenario generation methods for stochastic optimization. *European Journal of Operational Research* 255(1):121 – 132.

Löhndorf N, Shapiro A (2019) Modeling time-dependent randomness in stochastic dual dynamic programming. *European Journal of Operational Research* 273(2):650 – 661.

Löhndorf N, Wozabal D, Minner S (2013) Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. *Operations Research* 61(4):810–823.

Löhndorf N, Wozabal D (2021) Gas storage valuation in incomplete markets. *European Journal of Operational Research* 288(1):318–330.

Martens J (2010) Deep learning via hessian-free optimization. *ICML*, volume 27, 735–742.

Newman D (1982) The hexagon theorem. *IEEE Transactions on information theory* 28(2):137–139.

Pagès G, Printems J (2003) Optimal quadratic quantization for numerics: the gaussian case. *Monte Carlo Methods and Applications* 9:135–166.

Pennanen T (2005) Epi-convergent discretizations of multistage stochastic programs. *Mathematics of Operations Research* 30(1):245–256.

Pennanen T (2009) Epi-convergent discretizations of multistage stochastic programs via integration quadratures. *Mathematical Programming* 116(1):461–479.

Pennanen T, Koivu M (2005) Epi-convergent discretizations of stochastic programs via integration quadratures. *Numerische mathematik* 100(1):141–163.

Pereira M, Pinto L (1991) Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming* 52(2):359–375.

Pflug G (2001) Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming, Series B* 89(2):251–271.

Pflug G, Pichler A (2012) A distance for multistage stochastic optimization models. *SIAM Journal on Optimization* 22(1):1–23.

Pflug GC, Pichler A (2014) *Multistage Stochastic Optimization* (Springer Series in Operations Research and Financial Engineering).

Pflug GC, Pichler A (2015) Dynamic generation of scenario trees. *Computational Optimization and Applications* 62(3):641–668.

Powell W (2011) *Approximate dynamic programming. Solving the curses of dimensionality* (Wiley).

Rubinstein M (1994) Implied binomial trees. *The journal of finance* 49(3):771–818.

Saldi N, Yüksel S, Linder T (2017) On the asymptotic optimality of finite approximations to markov decision processes with borel spaces. *Mathematics of Operations Research* 42(4):945–978.

Shapiro A (2011) Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research* 209(1):63 – 72.

Shapiro A, Dentcheva D, Ruszczyński A (2009) *Lectures on Stochastic Programming: Modeling and Theory.* MOS-SIAM series on optimization (Siam).

Terça G, Wozabal D (2020) Envelope Theorems for Multi-Stage Linear Stochastic Optimization. *Operations Research* Accepted for Publication.

Villani C (2003) *Topics in optimal transportation*, volume 58 of *Graduate Studies in Mathematics* (Providence, RI: American Mathematical Society).

Wang T, Dyer J (2010) Valuing multifactor real options using an implied binomial tree. *Decision Analysis* 7(2):185–195.

Wang X, Ma S, Goldfarb D, Liu W (2017) Stochastic quasi-newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization* 27(2):927–956.

White D (1982) Finite state approximations for denumerable state infinite horizon discounted markov decision processes with unbounded rewards. *Journal of Mathematical Analysis and Applications* 86(1):292 – 306.