Computers & Operations Research 43 (2014) 108–115

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Low-discrepancy sampling for approximate dynamic programming with local approximators

C. Cervellera*, M. Gaggero, D. Macciò

Institute of Intelligent Systems for Automation, National Research Council, Via De Marini 6, 16149 Genova, Italy

ARTICLE INFO

Available online 17 September 2013

Keywords:
Approximate dynamic programming
Low-discrepancy sampling
Local approximation
Nadaraya–Watson models
Inventory forecasting

ABSTRACT

Approximate dynamic programming (ADP) relies, in the continuous-state case, on both a flexible class of models for the approximation of the value functions and a smart sampling of the state space for the numerical solution of the recursive Bellman equations. In this paper, low-discrepancy sequences, commonly employed for number-theoretic methods, are investigated as a sampling scheme in the ADP context when local models, such as the Nadaraya–Watson (NW) ones, are employed for the approximation of the value function. The analysis is carried out both from a theoretical and a practical point of view. In particular, it is shown that the combined use of low-discrepancy sequences and NW models enables the convergence of the ADP procedure. Then, the regular structure of the low-discrepancy sampling is exploited to derive a method for automatic selection of the bandwidth of NW models, which yields a significant saving in the computational effort with respect to the standard cross validation approach. Simulation results concerning an inventory management problem are presented to show the effectiveness of the proposed techniques.

1. Introduction

Consider a Markovian Decision Problem (MDP) characterized by a continuous-state discrete-time dynamic system evolving according to the stochastic state equation

$$\boldsymbol{x}_{t+1}=\boldsymbol{f}(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\theta}_t),$$

where $t=0,\dots,T-1$, $\boldsymbol{f}$ is a smooth vectorial field, $\boldsymbol{x}_t\in X_t\subset\mathbb{R}^n$ is the state vector, $\boldsymbol{u}_t\in U_t\subset\mathbb{R}^m$ is a decision (or control) vector and $\boldsymbol{\theta}_t\in\Theta_t\subset\mathbb{R}^q$ is a random vector affecting the system. The vectors $\boldsymbol{\theta}_t$ are characterized by a probability measure $P(\boldsymbol{\theta}_t)$ and constitute an independent chain of vectors over the T stages.

The problem consists in finding optimal decision vectors that minimize a cost function depending on the evolution of the state, which has an additive form over T stages:

$$J=\sum_{t=0}^{T-1}h(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\theta}_t)+h_T(\boldsymbol{x}_T),$$

where h and h_T are suitable single-stage cost functions.

Since the system is affected by random quantities, we look for closed-loop decision vectors: $\boldsymbol{u}_t$ must be a function, commonly called policy, of the current state, i.e., $\boldsymbol{u}_t=\boldsymbol{\mu}_t(\boldsymbol{x}_t)$.

Many optimization problems coming from different fields such as economics, engineering, environment management, logistics, artificial intelligence can be formalized within this framework. Dynamic Programming (DP) is the standard mathematical tool to solve this kind of problems [1,2]. In general, the DP equations have to be solved numerically, leading to approximations of the value functions and, possibly, optimal policies. This popular procedure usually takes the name of Approximate Dynamic Programming (ADP) and is based, in the continuous-state case, on two main ingredients: (i) a class of models to be used in order to approximate the value functions and (ii) a proper sampling of the state space for the numerical solution of the DP equations (for an introduction see, e.g., [3] and the references therein).

Concerning (i), many different architectures have been proposed in the literature, among which we can cite polynomial approximators [4], splines [5], multivariate adaptive regression splines [6] and neural networks [7,8]. In this paper we focus on a choice based on local approximation, specifically on Nadaraya–Watson (NW) models (see, e.g., [9,10]). Local approximators based on kernel functions are routinely employed in the ADP framework [3,11] and in the reinforcement learning context [12], closely related to the ADP one. The main advantage, with respect to other popular models, lies on the small computational effort required to obtain the value function approximations, basically corresponding to the minimization of a one-dimensional cost. Concerning the other fundamental ingredient of ADP mentioned above at point (ii), i.e., sampling, the classic choice of a regular grid of points

* Corresponding author. Tel.: +39 0106475654; fax: +39 0106475600.
E-mail addresses: cervellera@ge.issia.cnr.it (C. Cervellera), mauro.gaggero@ge.issia.cnr.it (M. Gaggero), ddmach@ge.issia.cnr.it (D. Macciò).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.09.006

coming from a uniform sampling of all the state components is not feasible in high-dimensional contexts, due to the well-known curse of dimensionality phenomenon, i.e., the exponential growth of the number of grid points as the state dimension increases. Then, efficient alternatives have been proposed in the literature, like sampling techniques commonly used in statistics for design of experiments, such as orthogonal arrays [13] and Latin hypercubes [14]. Another kind of sequences that have been successfully employed in the general context of ADP are low-discrepancy sequences (see, e.g., [15–18]) commonly used for numerical integration and number-theoretic algorithms [19]. Here, the use of such sequences is investigated in the context of ADP in combination with NW models. The main motivation behind the use of low-discrepancy sampling is that the resulting sets of points provide an efficient uniform covering of the state space as the number of points grows. More formally, it is proved in the paper that the choice of low-discrepancy sampling endows the NW structure with the ability to approximate the value functions arbitrarily well, which is a fundamental assumption needed to guarantee the convergence of the whole ADP procedure.

From a practical point of view, another motivation to employ low-discrepancy sampling with NW models in the ADP context is that the very regular structure and uniformity of the former can be exploited to derive a bandwidth selection method that can be used as an alternative to the cross validation procedure, which is the standard way of optimizing the kernel widths. In fact, rules of thumb available in the literature for the choice of the bandwidth of NW models employed in density estimation problems in the one-dimensional case, such as Silverman's rule or plugin rules [20], cannot be applied in a straightforward way to the multidimensional approximation case considered here. Thus, in this paper we present a method, directly based on the notion of discrepancy, that allows one to find reasonable values for the bandwidth parameter without recurring to any optimization procedure. Having a quick bandwidth selection method leads to a saving in the overall computational requirements of the entire ADP procedure, which are then basically reduced, at each stage, only to the minimization efforts needed to compute the value function estimates.

To sum up, the main contributions of the paper are:

- the introduction and analysis of the use of low-discrepancy sampling in the context of continuous-state ADP when local models are used for value function estimation, as an efficient alternative to random sampling in high-dimensional settings;
- the definition of a new method for automatic selection of the kernel bandwidths for value function approximation, exploiting the regularity of the low-discrepancy sampling.

Then, the paper provides the ADP practitioner with tools to improve the accuracy and computational burden of the procedure when local models are employed. The proposed approaches have been tested in a 15-dimensional inventory forecasting problem, comparing them with random sampling and the standard cross validation for bandwidth selection.

The paper is organized as follows. In Section 2, a review of the ADP algorithm is reported. Section 3 describes the considered NW models and low-discrepancy sequences. The use of NW models and low-discrepancy sequences within the ADP algorithm is discussed in Section 4. Numerical results are reported in Section 5, and conclusions are given in Section 6.

2. Review of the approximate dynamic programming scheme

The general scheme of the ADP procedure is briefly recalled in this section, together with some theoretical results on its convergence. The DP method relies on the backward computation of a value function that represents at each stage the optimal cost to be paid from that stage on. This leads to the recursive solution of the following well-known Bellman equations:

$$J_T^{\circ}(\boldsymbol{x}_T) = h_T(\boldsymbol{x}_T),$$
$$J_t^{\circ}(\boldsymbol{x}_t) = \min_{\boldsymbol{u}_t \in U_t \theta_t} \mathrm{E}[h(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t) + J_{t+1}^{\circ}(\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t))], \quad t = T-1, \dots, 0, \quad (1)$$

where the value function is represented by $J_t^{\circ}(\boldsymbol{x}_t)$ and $\mathrm{E}(\cdot)$ is the expectation operator. It is possible to prove (see, e.g., [2]) that $J_0^{\circ}(\boldsymbol{x}_0)$ corresponds to the optimal cost of the original MDP problem.

Since the value function is generally not available in the analytical form, in ADP an approximation, denoted by $\hat{J}_t$, has to be chosen inside a suitable class of models Γ at each stage t. To this purpose, a set Σ_t^L of L sampling points is chosen in X_t:

$$\Sigma_t^L = \{\boldsymbol{x}_{t,j} \in X_t : j = 1, \dots, L\}, \quad t = 1, \dots, T-1.$$

Then, using the approximate value function $\hat{J}_{t+1}(x_{t+1})$ defined at the previous stage, the basic ADP equation is written as

$$\tilde{J}_t^{\circ}(\boldsymbol{x}_{t,j}) = \min_{\boldsymbol{u}_t \in U_t} \frac{1}{S} \sum_{s=1}^{S} [h(\boldsymbol{x}_{t,j}, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}) + \hat{J}_{t+1}(\boldsymbol{f}(\boldsymbol{x}_{t,j}, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}))] \quad (2)$$

for each $\boldsymbol{x}_{t,j} \in \Sigma_t^L$. In Eq. (2), $\tilde{J}_t^{\circ}$ represents the estimated value of J_t° obtained by using $\hat{J}_{t+1}$, while the expected value with respect to $\boldsymbol{\theta}_t$ is estimated through an empirical mean computed over S realizations $\{\boldsymbol{\theta}_{t,1}, \dots, \boldsymbol{\theta}_{t,S}\}$ of the random variables drawn from their probability distribution.

Once the L values $\tilde{J}_t^{\circ}(\boldsymbol{x}_{t,j})$, $j = 1, \dots, L$, have been computed, the approximate value function $\hat{J}_t$ is obtained by exploiting the available observations. In general, we define for each stage t a class of models $\Gamma = \{\psi(\cdot, \boldsymbol{\alpha}_t) : \boldsymbol{\alpha}_t \in \Lambda \subset \mathbb{R}^k\}$ and set $\hat{J}_t(\boldsymbol{x}_t) = \psi(\boldsymbol{x}_t, \boldsymbol{\alpha}_t^*)$, where $\boldsymbol{\alpha}_t^*$ is obtained by minimizing an error criterion. A popular choice is the Mean Squared Error (MSE), which leads to the following optimization problem:

$$\boldsymbol{\alpha}_t^* = \arg\min_{\boldsymbol{\alpha} \in \Lambda} \frac{1}{L} \sum_{j=1}^{L} [\tilde{J}_t^{\circ}(\boldsymbol{x}_{t,j}) - \psi(\boldsymbol{x}_{t,j}, \boldsymbol{\alpha})]^2. \quad (3)$$

After this step, the value function can be estimated at each point of X_t and employed at stage $t-1$ for the computation of $\tilde{J}_{t-1}^{\circ}$.

Notice that the above described procedure to obtain the value function approximations is typically performed off line. Then, the approximations can be employed on line to compute the optimal decision vector in a given state, forward from $t=0$ to $t=T-1$. Specifically, the optimal vectors $\tilde{\boldsymbol{u}}_0^{\circ}, \dots, \tilde{\boldsymbol{u}}_{T-1}^{\circ}$ are computed on line, starting from the initial state $\tilde{\boldsymbol{x}}_0$, using at each stage t the ADP equations in the following way:

$$\tilde{\boldsymbol{u}}_t^{\circ} = \arg\min_{\boldsymbol{u}_t \in U_t} \frac{1}{S} \sum_{s=1}^{S} [h(\tilde{\boldsymbol{x}}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}) + \hat{J}_{t+1}(\boldsymbol{f}(\tilde{\boldsymbol{x}}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}))], \quad t = 0, \dots, T-1,$$

where $\tilde{\boldsymbol{x}}_t = \boldsymbol{f}(\tilde{\boldsymbol{x}}_{t-1}, \tilde{\boldsymbol{u}}_{t-1}^{\circ}, \tilde{\boldsymbol{\theta}}_{t-1})$, being $\tilde{\boldsymbol{\theta}}_{t-1}$ the realization of the random vector that affects the system in the on line phase. Eventually, the resulting states and decision vectors lead to the total cost actually paid in the on line phase from $\tilde{\boldsymbol{x}}_0$ to $\tilde{\boldsymbol{x}}_T$.

A key role for the convergence of the ADP algorithm is played by the approximation capabilities of the models in Γ. It is known that the convergence of the ADP solution, in terms of convergence of the total cost to the optimal one, is directly affected by the accuracy of the approximation of $\hat{J}_t$ in the various state points at stage t. To illustrate this, for a generic function $g : Z \to \mathbb{R}$, denote the infinite norm of g by $\|g\|_{\infty} = \sup_{\boldsymbol{z} \in Z} |g(\boldsymbol{z})|$. Then, we introduce the following assumption, which formalizes the universal approximation capability of the functions in the class Γ where we look for the approximation of the value functions.

Assumption 1. At each stage t, the value function approximation $\hat{J}_t$ is such that $\|\hat{J}_t - \tilde{J}_t^{\circ}\|_{\infty} \le \varepsilon$ for every $\varepsilon > 0$.

Under this assumption, it can be proved by induction on t (see, e.g., [7, p. 332]) that

$$\|J_0^{\circ} - \hat{J}_0\|_\infty \le T\varepsilon.$$

This result proves that we can annihilate the total error with respect to the true value function provided the approximating models are accurate enough. Notice that Assumption 1 is a condition on the class of models Γ and, in the case of NW models that depend structurally on data (as we will see in the following), also on the set of sampling points Σ_t^L. In particular, it states that there exists a model in Γ such that, after the optimization of the parameters based on the costs observed in the sampling points Σ_t^L, the error with respect to the function $\tilde{J}_t^{\circ}$ is arbitrarily small.

3. Nadaraya–Watson kernel models and low-discrepancy sampling

In this section we provide a brief overview of the two main elements considered here for the implementation of the ADP procedure: NW models for value function approximation and low-discrepancy sequences for efficient sampling of the state space.

3.1. Nadaraya–Watson models

Consider the value function $\tilde{J}_t^{\circ}$ computed at stage t by means of (2) and the set of sampling points $\Sigma_t^L = \{\boldsymbol{x}_{t,1}, \dots, \boldsymbol{x}_{t,L}\}$. The class Γ of NW models for the approximation of the value function is defined as

$$\Gamma = \left\{ \psi(\cdot, \alpha_t) = \frac{\sum_{j=1}^{L} \mathcal{K}(\cdot, \boldsymbol{x}_{t,j}, \alpha_t) \tilde{J}_t^{\circ}(\boldsymbol{x}_{t,j})}{\sum_{j=1}^{L} \mathcal{K}(\cdot, \boldsymbol{x}_{t,j}, \alpha_t)}, \alpha_t > 0 \right\},$$

where, for a generic point $\boldsymbol{z}_t \in X_t$, $\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)$ is a function, usually called kernel, that performs a weighted measure of the distance between $\boldsymbol{z}_t$ and $\boldsymbol{x}_{t,j}$. The main idea behind such an approximation scheme is that the value of the approximating function at the point $\boldsymbol{z}_t$ is a weighted average of the known function values in the neighborhood of $\boldsymbol{z}_t$, in such a way that closer points have higher weight.

Typically, the kernel map assumes the form $\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t) = \mathcal{G}(\|\boldsymbol{z}_t - \boldsymbol{x}_{t,j}\| / \alpha_t)$, where $\mathcal{G}(y)$ is a nonincreasing function for $y > 0$. Different kernel forms have been employed in the literature: examples are the Gaussian, the Epanechnikov, the Poisson and the tricube kernel [10]. In this paper, we focus on the Gaussian kernel $\mathcal{G}(y) = e^{-\pi y^2}$, due to its simplicity and mathematical properties that will be useful to prove the approximation capabilities of NW models.

The bandwidth α_t is the only parameter that needs to be tuned to change the form of the NW model. It defines the range of influence of the kernel and has to be optimized according to some criterion. Most commonly, the bandwidth is derived through a cross validation approach using a MSE criterion. Eq. (3) for the selection of the bandwidth is then replaced, for $t = 0, \dots, T-1$, by the following:

$$\alpha_t^* = \arg\min_{\alpha} \frac{1}{M} \sum_{m=1}^{M} \left[\frac{M}{L} \sum_{\boldsymbol{x}_{t,l} \in \Sigma_{t,m}^M} \left(\tilde{J}_t^{\circ}(\boldsymbol{x}_{t,l}) - \frac{\sum_{\boldsymbol{x}_{t,j} \in \Sigma_t^L \setminus \Sigma_{t,m}^M} \mathcal{K}(\boldsymbol{x}_{t,l}, \boldsymbol{x}_{t,j}, \alpha) \tilde{J}_t^{\circ}(\boldsymbol{x}_{t,j})}{\sum_{\boldsymbol{x}_{t,j} \in \Sigma_t^L \setminus \Sigma_{t,m}^M} \mathcal{K}(\boldsymbol{x}_{t,l}, \boldsymbol{x}_{t,j}, \alpha)} \right)^2 \right], \tag{4}$$

where M is a positive integer and $\Sigma_{t,m}^M = \{\boldsymbol{x}_{(m-1)M+1}, \dots, \boldsymbol{x}_{mM}\} \subset \Sigma_t^L$. Roughly speaking, at each time t and for each $m = 1,\dots,M$, the set of available data samples Σ_t^L is divided into two subsets, namely $\Sigma_{t,m}^M$ and $\Sigma_t^L \setminus \Sigma_{t,m}^M$. Then, the data samples in the set $\Sigma_t^L \setminus \Sigma_{t,m}^M$ are used as centers of the kernels, whereas the data samples belonging to $\Sigma_{t,m}^M$ are employed to compute a quadratic error between $\tilde{J}_t^{\circ}$ and the NW approximator. By varying m from 1 to M, all the points in Σ_t^L are used in turn as kernel centers and as points for error evaluation.

3.2. Low-discrepancy sequences

As previously pointed out, the class of sampling algorithms considered here for the ADP solution relies on a special family of methods, called low-discrepancy sequences, that provide deterministic generation of points uniformly scattered over a multidimensional set. The notion of uniformity is given with respect to a measure, typically employed in number-theoretic methods and numerical integration, called discrepancy [21].

To avoid burdening the notation, in this section we refer to a generic stage t and drop the subscript t from all the quantities. From now on we also assume that the set X to be sampled is the unitary n-dimensional cube, i.e., $X = [0,1]^n$. This is not a limitation since one can extend the results presented in this section to any hypercube by a simple scaling, and to more complex domains by suitable transformations [19].

Given a set of L points Σ^L in X, let ζ be the family of all subintervals B of the form $\prod_{i=1}^{n} [a_i, b_i]$, where $a_i, b_i \in [0,1]$, and let $\mathcal{C}(B, \Sigma^L)$ be the counting function for the number of points of Σ^L that belong to B. Then, the discrepancy of Σ^L is defined as follows:

$$\mathcal{D}(\Sigma^L) = \sup_{B \in \zeta} \left| \frac{\mathcal{C}(B, \Sigma^L)}{L} - \lambda(B) \right|, \tag{5}$$

where $\lambda(B)$ is the Lebesgue measure of B.

A low-discrepancy sequence aims at keeping the discrepancy of the resulting points in X as small as possible, and provides a favourable asymptotical rate of convergence of the discrepancy itself. For a comparison, an i.i.d. sequence of L points drawn with uniform distribution can be proved [22] to yield a rate of convergence for the discrepancy of order $O(1/\sqrt{L})$.

According to the definition of discrepancy, we say that samples are uniformly well spread over the domain X if, by subdividing it into a number of basic subsets, each one contains a number of points that are as proportional as possible to the volume of the subset itself. This concept is implemented by (d,n)-sequences, which generalize the concept of low-discrepancy sequences.

First, the definition of a $(d,q,n)-$net is provided.

Definition 1. An *elementary interval in base b* (where $b \ge 2$ is an integer) is a subinterval E of X having the form

$$E = \prod_{i=1}^{d} [a_i b^{-p_i}, (a_i + 1) b^{-p_i}),$$

where $a_i, p_i \in \mathbb{Z}$, $p_i > 0$, $0 \le a_i \le b^{p_i}$ for $1 \le i \le n$.

Let d, q be two integers satisfying $0 \le d \le q$. A $(d,q,n)-$net in base b is a set F of b^q points in $X \subset \mathbb{R}^n$ such that $\mathcal{C}(E,F) = b^d$ for every elementary interval E in base b with $\lambda(E) = b^{d-q}$.

It is easy to verify that a $(d,q,n)-$net is endowed with the property of good uniform spreading in X mentioned above. In fact, if the sample Σ^L is a $(d,q,n)-$net in base b, every elementary interval in which we divide X must contain b^d points of Σ^L.

However, since the cardinality of $(d,q,n)-$nets is constrained to be equal to b^q, (d,n)*-sequences* are derived to provide a higher degree of freedom in choosing the sample size L.

Definition 2. Let $d \ge 0$ be an integer. A sequence $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_L\}$ of points in $X \subset \mathbb{R}^n$ is a *(d,n)-sequence in base b* if, for all the integers $k \ge 0$ and $q \ge d$ (with $(k+1)b^q \le L$), the point set consisting of $\{x_{kb^q}, \dots, x_{(k+1)b^q}\}$ is a $(d,q,n)-$net in base b.

The definition of $(d,q,n)-$nets and (d,n)-sequences is due to Sobol' [23] for the case $b=2$ and to Niederreiter [24] for the general case. The following result can be proved [21].

Theorem 1. *For every dimension $n \geq 1$ and every prime power v (i.e., an integer power of a prime number), there exists a $(\tau(v,n),n)-$sequence in base v, where $\tau(v,n)$ is a constant that depends only on v and n.*

In practice, the construction of (d,n)-sequences can be obtained in several ways. Then, different specific low-discrepancy sequences have been proposed in the literature, such as, e.g., the Halton, the Faure, the Hammersely, the Sobol and the Niederreiter sequence. The reader interested in the details of the construction of the cited sequences can consult [19,21].

Concerning discrepancy, the fundamental property of (d,n)-sequences is that they are proved to yield a convergence rate of order $O(L^{-1}(\log L)^{n-1})$ [21]. This means that the convergence of the discrepancy, ignoring the logarithmic factor, is asymptotically almost linear with respect to the number of points L, and thus better than the quadratic rate of i.i.d. uniform random sequences.

Fig. 1 shows a comparison between the sampling of a two-dimensional cube using 1000 samples obtained from a Sobol' low-discrepancy sequence (left) and a uniform i.i.d. random sequence (right). It can be clearly seen how the low-discrepancy sampling scheme is more uniformly spread and regular.

Finally, we introduce another quantity usually employed in number-theoretic methods to measure uniformity, i.e., the dispersion, since it will be used in the next section to prove the approximation capabilities of NW models with low-discrepancy sampling. The dispersion of the set Σ^L of L points is defined as

$$\delta^*(\Sigma^L) = \sup_{\boldsymbol{y} \in X} \min_{j=1,\dots,L} \|\boldsymbol{y} - \boldsymbol{x}_j\|. \tag{6}$$

Likewise for the discrepancy, from the definition it can be seen that the smaller is the dispersion $\delta^*(\Sigma^L)$ the more uniformly the points of Σ^L are spread over X. In fact, the dispersion is actually closely related to the discrepancy: it can be shown (see, e.g., [21]) that sets Σ^L coming from low-discrepancy sequences are also able to provide convergence to zero of the dispersion as the number of points L grows.

4. Nadaraya–Watson models and low-discrepancy sequences for approximate dynamic programming

In this section, we discuss the use of low-discrepancy sequences in the ADP framework when local approximating schemes such as the NW models are employed. In particular, we analyze the accuracy of the resulting value function approximations and describe a bandwidth selection method that is alternative to the standard cross validation approach.

4.1. Approximation via NW models

We present a result stating that NW models can approximate the value functions arbitrarily well when the sets Σ_t^L are low-discrepancy sequences.

Recall that, at stage t, the value function approximation $\hat{J}_t$ takes on the form of a NW model. With a little abuse of notation, we make the dependence on α_t explicit by rewriting the value function approximation as $\hat{J}_t(\cdot,\alpha_t)$.

Theorem 2. *Let us consider the following assumptions for each stage $t = 0,\dots,T-1$:*

(i) *the function $\tilde{J}_t^{\circ}$ is continuous over X_t;*
(ii) *the sample points in Σ_t^L come from a (d,n)-sequence.*

Then, for each $\varepsilon > 0$, there exist L^* and α_t^* such that, for all $t = 0,\dots,T-1$,

$$\sup_{\boldsymbol{z}_t \in X_t} |\tilde{J}_t^{\circ}(\boldsymbol{z}_t) - \hat{J}_t(\boldsymbol{z}_t, \alpha_t^*)| < \varepsilon$$

for every $L > L^*$.

Proof. For a given t, the following notation will be adopted: $\delta_j(\boldsymbol{z}_t) = \|\boldsymbol{z}_t - \boldsymbol{x}_{t,j}\|$, $\Delta_j(\boldsymbol{z}_t) = |\tilde{J}_t^{\circ}(\boldsymbol{z}_t) - \tilde{J}_t^{\circ}(\boldsymbol{x}_{t,j})|$, $\overline{\Delta}_t = \max_{\boldsymbol{z},\boldsymbol{y} \in X_t} |\tilde{J}_t^{\circ}(\boldsymbol{z}) - \tilde{J}_t^{\circ}(\boldsymbol{y})|$. Notice that, since by assumption $\tilde{J}_t^{\circ}$ is continuous on a compact domain, a generalization of the Weierstrass Theorem (see, e.g., [25]) guarantees that $\overline{\Delta}_t$ is well defined and $\overline{\Delta}_t < \infty$.

Since $\tilde{J}_t^{\circ}(\boldsymbol{x}_{t,j}) = \tilde{J}_t^{\circ}(\boldsymbol{z}_t) \pm \Delta_j(\boldsymbol{z}_t)$ for all $\boldsymbol{z}_t$ and all $j = 1,\dots,L$, we can write

$$\begin{aligned}\hat{J}_t(\boldsymbol{z}_t,\alpha_t) &= \frac{\sum_{j=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)[\tilde{J}_t^{\circ}(\boldsymbol{z}_t) \pm \Delta_j(\boldsymbol{z}_t)]}{\sum_{j=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)} \\ &= \tilde{J}_t^{\circ}(\boldsymbol{z}_t) \pm \frac{\sum_{j=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)\Delta_j(\boldsymbol{z}_t)}{\sum_{j=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)}.\end{aligned}$$

For a given α_t and L, consider the set $\Xi_t^P(\boldsymbol{z}_t)$ of the P points of Σ_t^L such that $\|\boldsymbol{z}_t - \boldsymbol{x}_{t,j}\| \leq 2\delta^*(\Sigma_t^L)$, where δ^* is the dispersion of Σ_t^L defined in (6). Thus, we have

$$\begin{aligned}&\sup_{\boldsymbol{z}_t \in X_t} \left|\tilde{J}_t^{\circ}(\boldsymbol{z}_t) - \hat{J}_t(\boldsymbol{z}_t,\alpha_t)\right| \\ &\quad = \sup_{\boldsymbol{z}_t \in X_t} \frac{\sum_{j=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)\Delta_j(\boldsymbol{z}_t)}{\sum_{j=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)} \\ &\quad = \sup_{\boldsymbol{z}_t \in X_t} \left(\sum_{\boldsymbol{x}_{t,j} \in \Xi_t^P(\boldsymbol{z}_t)} \frac{\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)\Delta_j(\boldsymbol{z}_t)}{\sum_{k=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,k}, \alpha_t)} + \sum_{\boldsymbol{x}_{t,j} \notin \Xi_t^P(\boldsymbol{z}_t)} \frac{\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j}, \alpha_t)\Delta_j(\boldsymbol{z}_t)}{\sum_{k=1}^L \mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,k}, \alpha_t)} \right) \\ &\quad \leq \sup_{\boldsymbol{z}_t \in X_t} \left(\max_{\boldsymbol{x}_{t,j} \in \Xi_t^P(\boldsymbol{z}_t)} \Delta_j(\boldsymbol{z}_t) \right) + \sup_{\boldsymbol{z}_t \in X_t} \frac{\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,\tilde{j}}, \alpha_t)}{\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j^{\min}}, \alpha_t)} (L-P)\overline{\Delta}_t,\end{aligned} \tag{7}$$

where $\boldsymbol{x}_{t,j^{\min}}$ is the point of Σ_t^L closest to $\boldsymbol{z}_t$ and $\boldsymbol{x}_{t,\tilde{j}}$ is the point outside $\Xi_t^P(\boldsymbol{z}_t)$ closest to $\boldsymbol{z}_t$.

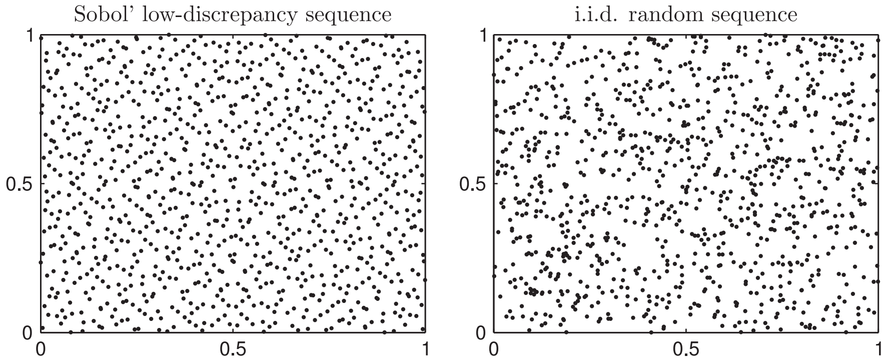

Fig. 1. Comparison between Sobol' and i.i.d. uniform samplings of the 2-dimensional unit cube.

The first term of (7) comes from the convexity of the kernel structure. Now let ε be a positive quantity. Due to the fact that the domain X_t is compact, $\tilde{J}_t^{\circ}$ is uniformly continuous. Thus, there exists $\hat{\delta} > 0$ such that, for every $\boldsymbol{z}_t$ and every $\boldsymbol{x}_{t,j} \in \Xi_t^P(\boldsymbol{z}_t)$, $\Delta_j(\boldsymbol{z}_t) \leq \varepsilon/2$ when $\|\boldsymbol{z}_t - \boldsymbol{x}_{t,j}\| \leq \hat{\delta}$. Since the dispersion $\delta^*(\Sigma_t^L)$ tends to 0 as L grows, we can take a L^* sufficiently large such that $\|\boldsymbol{z}_t - \boldsymbol{x}_{t,j}\| < 2\delta^*(\Sigma_t^L) < \hat{\delta}$ when $L > L^*$, for all $\boldsymbol{z}_t \in X_t$ and all $\boldsymbol{x}_{t,j} \in \Xi_t^P(\boldsymbol{z}_t)$.

The second term of (7) comes from the fact that if $\{\kappa_1, \ldots, \kappa_Q\}$ is a set of positive numbers and $\kappa^* > 0$, we have

$$\frac{\kappa_1 + \cdots + \kappa_Q}{\kappa_1 + \cdots + \kappa_Q + \kappa^*} \leq \frac{\max_{i=1,\ldots,Q} \kappa_i}{\kappa^*} Q.$$

By defining $\tilde{\delta}(\boldsymbol{z}_t) = \|\boldsymbol{z}_t - \boldsymbol{x}_{t,\tilde{j}}\|$ and $\delta_{\min}(\boldsymbol{z}_t) = \|\boldsymbol{z}_t - \boldsymbol{x}_{t,j^{\min}}\|$ we have, for the Gaussian kernel,

$$\frac{\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,\tilde{j}}, \alpha_t)}{\mathcal{K}(\boldsymbol{z}_t, \boldsymbol{x}_{t,j^{\min}}, \alpha_t)} = \exp\left(-\frac{\pi}{\alpha_t}(\tilde{\delta}(\boldsymbol{z}_t)^2 - \delta_{\min}(\boldsymbol{z}_t)^2)\right) \leq \exp\left(-\frac{\pi\delta^*(\Sigma_t^L)^2}{\alpha_t}\right),$$

where the last inequality comes directly from the fact that $\tilde{\delta}(\boldsymbol{z}_t) - \delta_{\min}(\boldsymbol{z}_t) \geq \delta^*(\Sigma_t^L)$ by definitions of $\boldsymbol{x}_{t,j^{\min}}$ and $\boldsymbol{x}_{t,\tilde{j}}$. Then, (7) can be made smaller than $\varepsilon/2$ by properly decreasing α_t.

Finally, we obtain $\sup_{\boldsymbol{z}_t \in X_t} |\tilde{J}_t^{\circ}(\boldsymbol{z}_t) - \hat{J}_t(\boldsymbol{z}_t, \alpha_t^*)| < \varepsilon/2 + \varepsilon/2$, i.e., for any $\varepsilon > 0$ there exist L^* sufficiently large and α_t^* sufficiently small such that $\sup_{\boldsymbol{z}_t \in X_t} |\tilde{J}_t^{\circ}(\boldsymbol{z}_t) - \hat{J}_t(\boldsymbol{z}_t, \alpha_t^*)| < \varepsilon$ for $L > L^*$. □

Notice that Assumption (i) of Theorem 2 is always fulfilled under mild conditions on the involved functions. The interested reader can consult [26, Ch. 3] for a more detailed discussion.

Theorem 2 states that we can arbitrarily reduce the approximation error introduced by NW models by properly increasing the number of points in Σ_t^L. This shows that the combination of NW models and low-discrepancy sampling actually allows one to satisfy Assumption 1 and, eventually, guarantee covergence of the total cost to the true optimal one, according to the results discussed in Section 2.

4.2. Exploiting low-discrepancy regularity for bandwidth selection

After discussing the theoretical properties of NW models and low-discrepancy sequences in the ADP context, we focus on the advantages enabled from a practical point of view. In particular, we show how the very regular structure of low-discrepancy sampling can be exploited to provide a selection method for the bandwidth parameter α_t that can be used as an alternative to the standard cross validation procedure.

According to the notion of discrepancy given in Section 3.2, we can assume that, by definition of (d,n)-sequence, the difference between the volume of a subset of X_t and the frequency of the number of points of Σ_t^L contained therein is small. More specifically, assume that the set X_t corresponds to the hypercube of unitary volume, and that L is the number of sampling points in Σ_t^L. Then, a hypercube of volume $1/v$ is expected to contain about L/v samples when a (d,n)-sequence is employed to generate the points.

In order to exploit the regularity and uniformity of low-discrepancy sets, let us denote by $V(X_t)$ the volume of the sets X_t, $t = 0, \ldots, T-1$, each sampled with L points. Then, consider a percentage $p \in [0,1]$ of the total amount of points L, which corresponds to a certain number N of the total points.

Given a point $\boldsymbol{x} \in X_t$, let us denote by $W_{\boldsymbol{x},t}^p$ the hypercube centered in $\boldsymbol{x}$ containing N points, with $N = \lfloor L \cdot p \rfloor$, where, for any real number z, $\lfloor z \rfloor$ denotes the closest integer smaller than or equal to z. If the sampling is performed by means of a (d,n)-sequence, we can assume that the volume of this hypercube is about $V(X_t)p \simeq V(X_t)N/L$. Thus, the diagonal of the hypercube having this volume is equal to $l = (V(X_t)N/L)^{1/n} \cdot \sqrt{n}$.

The idea for bandwidth selection of the NW model is to constrain the kernel function $\mathcal{K}$ to take on a sufficiently small value ν at the boundary of $W_{\boldsymbol{x},t}^p$, whose distance from the point $\boldsymbol{x}$ is assumed to be equal to $l/2$. In this way, we can control a priori the range of influence of the kernel, i.e., determine exactly how much the kernel centers close to $\boldsymbol{x}$ affect the output of the value function estimator. Specifically, in the case of Gaussian kernels, we impose

$$\exp\left(-\frac{\pi(l/2)^2}{\alpha_t}\right) = \nu,$$

and thus the corresponding choice for α_t becomes

$$\alpha_t = -\frac{\pi(l/2)^2}{\log \nu}. \tag{8}$$

Obviously, the choice of p and ν is arbitrary, which makes the bandwidth selection method described above not fully automatic. However, the main advantage with respect to a direct guess on the bandwidth is that p and ν are "absolute" parameters. This means that they can be chosen independently on the volume of X_t and the number of sampling points, thus allowing one to have a direct and effective control over the bias-variance dilemma. Furthermore, some rules of thumb can be derived in practice for the choice of such parameters.

The parameter p is the one that has the least expected impact on the output, since the notion of uniformity based on discrepancy on which the method does not depend on the volume of the hypercube B in (5). However, large values should be avoided in order to follow the principle of locality on which the NW models are based. Thus, values in the 1%–5% range appear to be reasonable. Concerning ν, its value should not be too close to 1 in order to avoid a large bias but, at the same time, it should not be too small in order to keep the variance small. Therefore, values in the $10^{-3} - 10^{-1}$ range can be considered again as reasonable choices.

The simulation results presented in Section 5 show that values of p and ν chosen in the above mentioned ranges actually yield very similar performances, confirming that their choice is not critical. Furthermore, all the above mentioned performances turn out to be comparable with those given by choosing the bandwidth α_t through standard cross validation.

Finally, notice that the proposed bandwidth selection method depends only on the sampling of the state space, and not on the form of the value function. On one hand, this may be non-optimal (indeed, most of the rules of thumb available for classic density estimation with NW kernels use also information from the output data, such as the variance) but, on the other hand, it avoids computationally intensive procedures to obtain parameters needed to estimate the value function smoothness, such as, e.g., second order derivatives.

5. Numerical results

In order to test the performance of low-discrepancy sampling in the ADP procedure when local models are employed, optimal management of an inventory has been chosen for a simulation analysis. The addressed problem is based on an inventory forecasting model that has been often employed as a testbed for ADP schemes (see, e.g., [6,14]). The aim is to control the inventory levels of a certain number of items based on forecasts of future demands.

The state variables of the system are the inventory levels and demand forecasts for each item. Specifically, we focus on a setting with five items and forecasts on item demands for the two stages ahead. Denote the inventory level of item $i = 1, \ldots, 5$ at a given time t as $w_{t,i}$, the demand forecast for item i at time $t+1$ as $r_{t,i}$, and the demand forecast for item i at time $t+2$ as $s_{t,i}$. Let us collect all the state variables in the 15-dimensional state vector $\boldsymbol{x}_t=(w_{t,1},\dots,w_{t,5},r_{t,1},\dots,r_{t,5},s_{t,1},\dots,s_{t,5})^\top \in \mathbb{R}^{15}$.

The decision vector $\boldsymbol{u}_t$ is made up by the order quantities. Specifically, let $u_{t,i}$ be the amount of item $i=1,\dots,5$ ordered at time t. The overall decision vector is $\boldsymbol{u}_t=(u_{t,1},\dots,u_{t,5})^\top \in \mathbb{R}^5$.

The inventory levels and corresponding forecasts evolve according to the following state equations:

$$\begin{aligned}
&w_{t+1,i}=w_{t,i}+u_{t,i}-r_{t,i}\theta_{t,i}, \quad i=1,\dots,5,\\
&r_{t+1,i}=s_{t,i}\theta_{t,i+5}, \quad i=1,\dots,5,\\
&s_{t+1,i}=q_i\theta_{t,i+10}, \quad i=1,\dots,5,
\end{aligned}$$

where $\theta_{t,i}$ is a random quantity that represents a correction between the true demand of item i and the corresponding forecasts, whereas q_i is a positive constant. Let us collect the random quantities in the vector $\boldsymbol{\theta}_t=(\theta_{t,1},\dots,\theta_{t,15})^\top \in \mathbb{R}^{15}$.

The goal is to select the decision vectors in an optimal way, i.e., to satisfy the demand while keeping the storage levels as small as possible. Following [6], we focus on a "V-shaped" cost function like the following:

$$h(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\theta}_t)=\sum_{i=1}^{5}(\omega_i \max(w_{t+1,i},0)-\gamma_i \min(0,-w_{t+1,i})), \tag{9}$$

where $\omega_i \ge 0$ is a holding cost coefficient for item i and $\gamma_i \ge 0$ is a backorder cost coefficient for item i. Again, following [6], in order to deal with differentiable costs and help the optimization routine to find a good minimum, we rely on a smooth approximation of the ideal V-shape of (9). Specifically, we introduce the following functions:

$$Q^+(z,\xi)=\begin{cases} 0 & \text{for } z\le 0,\\ \dfrac{1}{4\xi^2}z^3-\dfrac{1}{16\xi^3}z^4 & \text{for } 0<z<2\xi,\\ z-\xi & \text{for } z\ge 2\xi, \end{cases} \tag{10}$$

$$Q^-(z,\xi)=\begin{cases} -z-\xi & \text{for } z\le -2\xi.\\ -\dfrac{1}{4\xi^2}z^3-\dfrac{1}{16\xi^3}z^4 & \text{for } -2\xi<z<0\xi,\\ 0 & \text{for } z\ge 0, \end{cases} \tag{11}$$

where ξ is a positive constant such that $Q^+(z,\xi)\to\max(z,0)$ and $Q^-(z,\xi)\to\min(0,z)$ as $\xi\to 0$. The function h in (9) is then replaced by

$$h(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\theta}_t)=\sum_{i=1}^{5}(\omega_i Q^+(w_{t+1,i},\xi)-\gamma_i Q^-(-w_{t+1,i},\xi)).$$

We impose constraints on the decision vectors, i.e., we impose $\boldsymbol{u}_t \in [\boldsymbol{u}_t^{\min},\boldsymbol{u}_t^{\max}]$ for all t, where $\boldsymbol{u}_t^{\min}\in\mathbb{R}^5$ and $\boldsymbol{u}_t^{\max}\in\mathbb{R}^5$ are given constants. Moreover, for all t, we impose

$$\sum_{i=1}^{5} u_{t,i} < U^{\max}, \tag{12}$$

where $U^{\max}$ is a positive constant.

For the numerical tests, we focused on an instance of the previous problem with $T=50$ decision stages (corresponding to about 1 year, if orders are made weekly). The values of the variables $\theta_{t,i}$, $t=0,\dots,49$, $i=1,\dots,15$, were randomly chosen according to a lognormal distribution with mean equal to 1. Moreover, for all $i=1,\dots,5$, we fixed $\omega_i=1$, $\gamma_i=2$, and $q_i=2+(i-1)\cdot 0.5$. Concerning constraints, we fixed $U^{\max}=15$ and $\boldsymbol{u}_t^{\min}=[0,0,0,0,0]^\top$, $\boldsymbol{u}_t^{\max}=[5,5,5,5,5]^\top$ for all $t=0,\dots,49$. The constant ξ in (10) and (11) was taken equal to 1.

In order to keep the computational requirements to find an ADP solution feasible, and to mitigate propagation of the value function approximation error through the stages as well, a multistep lookahead approach was implemented, which is a standard in ADP when the number of stages is large (see, e.g., [2] for details on convergence and accuracy of the solution). Specifically, at every stage t, the on line optimal control vector was computed employing approximate value functions obtained as if the horizon was composed of $\overline{T}$ stages, with $\overline{T} \ll T$. In particular, the ADP solution procedure was applied as described in Section 2 for a $\overline{T}-$stage problem, leading to the definition of $\overline{T}$ approximate value functions. Then, since the state equation $\boldsymbol{f}$ and cost function h do not depend on the particular stage t, to obtain the optimal on line control vector $\tilde{\boldsymbol{u}}_t$ it is sufficient to employ only the first approximate value function by solving, for all $t=0,\dots,49$, the following optimization problem:

$$\tilde{\boldsymbol{u}}_t^{\circ}=\arg\min_{\boldsymbol{u}_t \in U_t}\frac{1}{S}\sum_{s=1}^{S}[h(\tilde{\boldsymbol{x}}_t,\boldsymbol{u}_t,\boldsymbol{\theta}_{t,s})+\hat{J}_1(\boldsymbol{f}(\tilde{\boldsymbol{x}}_t,\boldsymbol{u}_t,\boldsymbol{\theta}_{t,s}))].$$

The number of lookahead stages $\overline{T}$ was chosen equal to 5 for the tests (which is a good compromise between accuracy and computational effort; notice that in most lookahead solution approaches $\overline{T}$ is equal to 1 or 2). In order to solve the $\overline{T}-$step lookahead ADP problem, bounds for the state spaces X_k at the various stages of the lookahead horizon, $k=1,\dots,\overline{T}$ were defined. Specifically, we fixed $\boldsymbol{x}_k \in [\boldsymbol{x}_k^{\min},\boldsymbol{x}_k^{\max}]$, where $\boldsymbol{x}_k^{\min}=(-5-3k,-5-3k,-5-3k,-5-3k,-5-3k,0,0,0,0,0,0,0,0,0,0)^\top$ and $\boldsymbol{x}_k^{\max}=(5,5,5,5,5,3,3,3,3,3,3,3,3,3,3)^\top$.

Three different types of simulations were performed. A first round of tests was devoted to verify the advantages of low-discrepancy sampling, comparing the results to those yielded by a pure random i.i.d. sampling of the state space. In a second round the empirical approach described in Section 4.2 for the selection of the bandwidth was applied in order to evaluate the impact of the related parameters. The performance of the empirical approach was evaluated in the final round of tests by comparing the results to those obtained by bandwidth selection with standard cross validation. All the simulations were performed in Matlab, on a PC equipped with an Intel Core 2 Duo with a 1.8 GHz CPU and 2 GB of RAM.

For the first round of tests, focused on the comparison of low-discrepancy sampling with random sampling, different random and quasi-random sequences were employed to sample the state space X_t at the various time instants. In particular, concerning low-discrepancy sampling, the Sobol', Niederreiter and Halton sequences were employed. As to random sampling, 20 different i.i.d. random sequences with uniform distribution were employed, and the average results were considered for the comparison. This choice was made in order to add robustness to the results, due to the random nature of the corresponding point sets, whereas the low-discrepancy ones are deterministic by construction. Two sampling sizes were tested for all the sequences: $L=2000$ and $L=5000$. For each sequence, a set of $\overline{T}=5$ approximate value functions was obtained by applying the procedure described in Section 2, using NW models for the approximation of the value functions. For these tests, standard 10-fold cross validation (4) was used for the selection of the bandwidths α_t.

In order to evaluate the goodness of a particular solution (i.e., of a particular sampling scheme), the total on line cost over the $T=50$ test stages yielded by the corresponding value function approximations was computed as described above in 100 simulation runs, each characterized by different initial conditions $\tilde{\boldsymbol{x}}_0$ and online random sequences $\tilde{\boldsymbol{\theta}}_t$, $t=0,\dots,49$, and the average value of the costs was taken as the performance index for that solution.

Fig. 2 summarizes the obtained results. In the figure, the average costs are reported for the three kinds of low-discrepancy sampling schemes mentioned above. Then, the grand average over these three low-discrepancy sequences is also reported, together with the grand average of the 20 uniform i.i.d. random sequences employed for the tests.

From the simulation results, it is evident that the low-discrepancy sampling schemes yield better results than those given by the random sequences, confirming in practice the theoretical advantages suggested by their asymptotic properties from Section 3.2. It is also interesting to notice that low-discrepancy sampling produces, averagely, an improvement of performance as the size L grows, while this is less evident for random sampling. Among the different low-discrepancy kinds of sequences, the Niederreiter one seems to guarantee the best results.

The second round of tests, as said, was focused on the use of the empirical rule (8) for bandwidth selection presented in Section 4.2, as an alternative to the minimization of the cross validation cost (4). Specifically, the tests were aimed at evaluating the effect of the parameters in (8) that have to be specified a priori. To this purpose, values of p equal to 0.01, 0.03, and 0.05 were employed and, for each value of p, five values of ν were considered, corresponding to increasing orders of magnitude (specifically, $\nu=10^{-4}$, 10^{-3}, 10^{-2}, 10^{-1} and 1). For the tests, a low-discrepancy sampling of the Sobol' kind was employed, with size equal to $L=2000$ and $L=5000$.

Fig. 3 reports, for the various values of the parameters, the average total costs computed over 100 simulation runs, each one with different initial conditions $\tilde{\boldsymbol{x}}_0$ and random vectors $\tilde{\boldsymbol{\theta}}_t$, $t=0,\ldots,49$. For a comparison, costs provided by a reasonable heuristic approach for the selection of the order quantities $u_{t,i}$ are also reported in the figure. Such an approach consists in ordering at each stage all the forecasted item quantities minus what is already in the store, up to the amounts allowed by the constraints. More in detail, the policy at stage t for all i is defined as

$$u_{t,i}=\min(u_{t,i}^{\max},\max(0,r_{t,i}+s_{t,i}-w_{t,i})). \tag{13}$$

Then, in order to fulfill the constraint (12), the order quantities given by (13) are replaced by the following quantities:

$$\tilde{u}_{t,i}=\frac{u_{t,i}}{\sum_{i=1}^{5}u_{t,i}}U^{\max}. \tag{14}$$

It can be seen that both in the cases of $L=2000$ and $L=5000$ data samples the mean value of the cost decreases as ν increases up to $\nu=10^{-1}$, while the mean of the cost corresponding to $\nu=1$ is higher than the case of $\nu=10^{-1}$. Roughly speaking, the larger the value of ν, the larger the influence of the kernel function centered at a given point in the domain. On the contrary, if ν is small, the kernel function gives importance only to the points that are very close to the center of the kernel. The best values of ν correspond then to a trade off between the two cases. Concerning the parameter p, it turns out that, in general, the performance of the empirical rule gets better when the percentage of points used for the selection of α_t increases. However, this parameter tends to become not critical when ν is chosen well. Overall, the behavior given by varying the two parameters confirms that there is a vast range of values of ν that yields good performance, which makes also the choice of p not a critical issue, as seen above. As expected, this behavior gets even better as the number of sampling points L grows. Almost in all cases, the results are better than those provided by the heuristic approach (13) and (14). Only when $\nu=10^{-4}$ and $L=2000$ the performances of the heuristic policy and of the empirical rule for the selection of α_t are comparable, suggesting that too small values of ν should be avoided.

In order to evaluate the actual performance of the empirical rule for bandwidth selection, a comparison with a selection based on cross validation was performed in a third series of tests. In particular, ADP solutions were computed by applying (8) using $p=0.03$ and two values of ν: 10^{-1} and 10^{-4} (the best and the worst values of ν in terms of final cost, respectively, according to Fig. 3). The low-discrepancy kind of sequence employed for these tests was again the Sobol' one, with both $L=2000$ and $L=5000$. Fig. 4 reports the average total costs obtained with the various solutions, computed over 100 simulation runs, each one with different initial conditions $\tilde{\boldsymbol{x}}_0$ and on line random sequences $\tilde{\boldsymbol{\theta}}_t$, $t=0,\ldots,49$. Notice that the simulation results reported for the cross validation approach are the same reported in Fig. 2 for the Sobol' sequence.

Looking at the simulation results, we can see that the best ones are obtained by the empirical rule with $\nu=10^{-1}$ for both $L=2000$ and $L=5000$, while the results obtained by cross validation are a little worse. The results obtained by the empirical rule with $\nu=10^{-4}$ and $L=2000$ or $L=5000$ are the worst (see also Fig. 3). Overall, the empirical rule (8) appears to yield results that are comparable to, and in some cases even better than, those yielded by the standard cross validation approach. A possible explanation

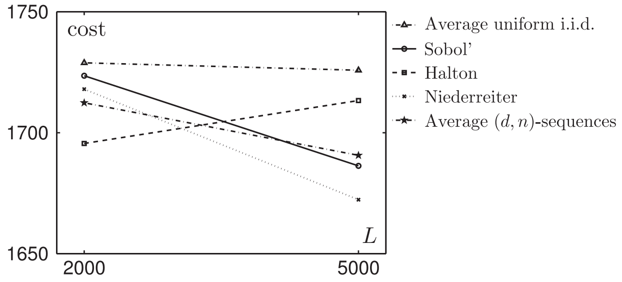

Fig. 2. Simulation results for the comparison of low-discrepancy and i.i.d. uniform sampling.

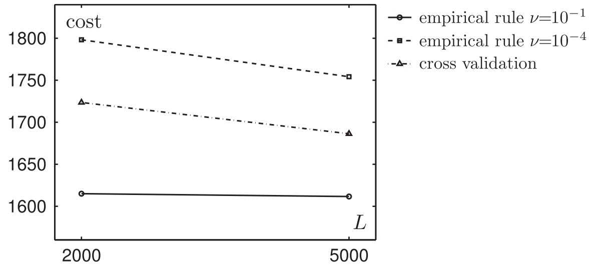

Fig. 4. Comparison between cross validation and the empirical selection of the bandwidths α_t.

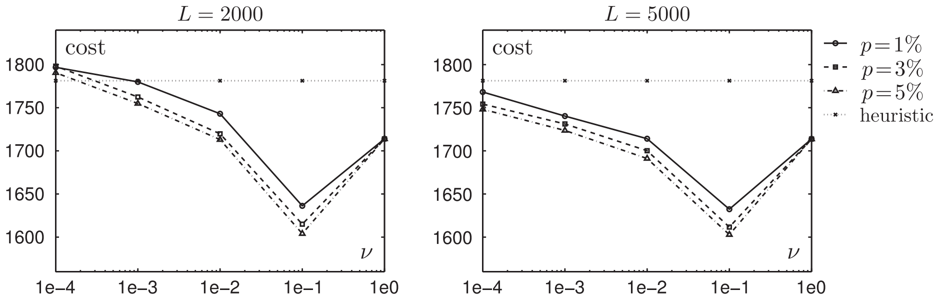

Fig. 3. Simulation results for the empirical selection of the bandwidths α_t.

may be the fact that the latter involves an optimization routine characterized by a cost function that can become very complex if L is large, and thus it may end up providing a sub-optimal bandwidth parameter.

Concerning computational times, it turns out that, as expected, the time required to find the bandwidth parameter α_t with the empirical rule is much smaller than the corresponding time required by cross validation. This is not surprising, since in the first case only Eq. (8) has to be applied, whereas in the second case one has to solve the optimization problem (4). In particular, in the tests the average time needed for the computation of α_t by the empirical rule turned out to be about 10^{-4} s, whereas cross validation required about 2.5 s.

6. Conclusions and further developments

In this paper, we have investigated how low-discrepancy sequences can be successfully exploited within the continuous-state ADP framework when local models, such as the NW ones, are chosen as approximators for the value functions. It has been proved that such sequences endow the NW structure with the approximation capabilities needed to guarantee the convergence of the ADP procedure. Then, due to its regular structure, the low-discrepancy sampling scheme has permitted to derive an empirical method for an automatic selection of the bandwidth of the NW model. Such a method can be employed as an alternative to the standard cross validation approach, which is characterized by an optimization procedure, thus leading to a significant saving in the total computational effort required by the entire ADP procedure. Simulation results obtained for a 15-dimensional inventory forecasting problem have shown that low-discrepancy sampling is superior to uniform i.i.d. random sampling in terms of performance. Furthermore, the results obtained with the proposed empirical method for automatic bandwidth selection have turned out to be comparable with those yielded by the standard cross validation approach. Future work will aim at investigating low-discrepancy sampling for a wider class of local and non-parametric models for value function approximation. Further simulation tests will also be carried out to evaluate the performance of the method on a vast range of applications from different fields.

References

[1] Bellman R. Dynamic programming. Princeton: Princeton University Press; 1957.

[2] Bertsekas D. Dynamic programming and optimal control, 2nd edition, vol. I. Belmont: Athena Scientific; 2000.

[3] Powell W. Approximate dynamic programming: solving the curses of dimensionality. 2nd edition. Wiley; 2011.

[4] Bellman R, Kalaba R, Kotkin B. Polynomial approximation—a new computational technique in dynamic programming allocation processes. Mathematics of Computation 1963;17:155–61.

[5] Johnson S, Stedinger J, Shoemaker C, Li Y, Tejada-Guibert J. Numerical solution of continuous-state dynamic programs using linear and spline interpolation. Operations Research 1993;41:484–500.

[6] Chen V, Ruppert D, Shoemaker C. Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming. Operations Research 1999;47(1):38–53.

[7] Bertsekas D, Tsitsiklis J. Neuro-Dynamic Programming. Belmont: Athena Scientific; 1996.

[8] Gaggero M, Gnecco G, Sanguineti M. Approximate dynamic programming for stochastic N-stage optimization with application to optimal consumption under uncertainty. Computational Optimization and Applications, in press.

[9] Cervellera C, Macciò D. Learning with kernel smoothing models and low discrepancy sampling. IEEE Transactions on Neural Networks and Learning Systems 2013;24:504–9.

[10] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference and prediction. New York: Springer; 2009.

[11] Cervellera C, Macciò D. A comparison of global and semi-local approximation in T-stage stochastic optimization. European Journal of Operational Research 2011;208:109–18.

[12] Ormoneit D, Sen S. Kernel-based reinforcement learning. Machine Learning 2002;49(2–3):161–78.

[13] Chen V. Measuring the goodness of orthogonal array discretizations for stochastic programming and stochastic dynamic programming. SIAM Journal of Optimization 2001;12:322–44.

[14] Cervellera C, Chen V, Wen A. Neural network and regression spline value function approximations for stochastic dynamic programming. Computers and Operations Research 2006;34:70–90.

[15] Cervellera C, Gaggero M, Macciò D. Efficient kernel models for learning and approximate minimization problems. Neurocomputing 2012;97:74–85.

[16] Cervellera C, Gaggero M, Macciò D, Marcialis R. Quasi-random sampling for approximate dynamic programming. In: Proceedings of the 2013 international joint conference on neural networks; 2013. p. 2567–74.

[17] Cervellera C, Muselli M. Efficient sampling in approximate dynamic programming algorithms. Computational Optimization and Applications 2007;38:417–43.

[18] Gaggero M, Gnecco G, Sanguineti M. Dynamic programming and value-function approximation in sequential decision problems: error analysis and numerical results. Journal of Optimization Theory and Applications 2013;156:380–416.

[19] Fang KT, Wang Y. Number-theoretic methods in statistics. London: Chapman & Hall; 1994.

[20] Schucany W. Kernel smoothers: an overview of curve estimators for the first graduate course in nonparametric statistics. Statistical Science 2004;19 (4):663–75.

[21] Niederreiter H. Random number generation and quasi-monte carlo methods. Philadelphia: SIAM; 1992.

[22] Chung KL. An estimate concerning the Kolmogoroff limit distribution. Transactions of the American Mathematical Society 1949;67:36–50.

[23] Sobol' IM. The distribution of points in a cube and the approximate evaluation of integrals. Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 1967;7:784–802.

[24] Niederreiter H. Point sets and sequences with small discrepancy. Monatshefte fur Mathematik 1987;104:273–337.

[25] Dudley RM. Real analysis and probability. 2nd edition. Cambridge University Press; 2002.

[26] Stokey N, Lucas R, Prescott E. Recursive methods in economic dynamic programming. Cambridge, Massachusetts: Harvard University Press; 1989.