Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Reinforced optimal control

Christian Bayer[1], Denis Belomestny[2], Paul Hager[3],

Paolo Pigato[4], John G. M. Schoenmakers[1], Vladimir Spokoiny[1]

submitted: December, 3, 2020

[1] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: christian.bayer@wias-berlin.de
john.schoenmakers@wias-berlin.de
vladimir.spokoiny@wias-berlin.de

[2] Duisburg-Essen University
Thea-Leymann-Str. 9
45127 Essen
Germany
E-Mail: denis.belomestny@uni-due.de

[3] Technische Universität Berlin
Straße des 17. Juni 136
10623 Berlin
Germany
E-Mail: phager@math.tu-berlin.de

[4] University of Rome Tor Vergata
Via Columbia 2
00133 Roma
Italy
E-Mail: paolo.pigato@uniroma2.it

No. 2792
Berlin 2020

2010 *Mathematics Subject Classification.* 91G20, 93E24.

Key words and phrases. Reinforced regression, least squares Monte Carlo, stochastic optimal control.

CB, PH, JS, and VS gratefully acknowledge financial support from the German science foundation (DFG) via the cluster of excellence MATH+, project AA4-2.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Reinforced optimal control

Christian Bayer, Denis Belomestny, Paul Hager,
Paolo Pigato, John G. M. Schoenmakers, Vladimir Spokoiny

Abstract

Least squares Monte Carlo methods are a popular numerical approximation method for solving stochastic control problems. Based on dynamic programming, their key feature is the approximation of the conditional expectation of future rewards by linear least squares regression. Hence, the choice of basis functions is crucial for the accuracy of the method. Earlier work by some of us [Belomestny, Schoenmakers, Spokoiny, Zharkynbay, Commun. Math. Sci., 18(1):109–121, 2020] proposes to *reinforce* the basis functions in the case of optimal stopping problems by already computed value functions for later times, thereby considerably improving the accuracy with limited additional computational cost. We extend the reinforced regression method to a general class of stochastic control problems, while considerably improving the method's efficiency, as demonstrated by substantial numerical examples as well as theoretical analysis.

1 Introduction

Stochastic control problems form an important class of stochastic optimization problems that find applications in a wide variety of fields, see [Pha09] for an overview. The general problem can be formulated as follows: How should a decision maker control a system with a stochastic component to maximize the expected reward? In the theory of stochastic control one distinguishes between problems with continuous and discrete sets of possible control values. While the first class of control problems contains, for example, energy storage problems, the second one includes stopping and multiple stopping problems. Furthermore one differentiates between discrete time and continuous time optimal control problems. (Neither of these distinctions is fundamental: for instance, many numerical methods will replace optimal control problems with a continuous set of control values in continuous time by a surrogate problem with discrete control values in discrete time. Moreover, many discrete optimal control problem may well be analyzed as continuous ones, if the number of possible control values or time-steps is finite, but very high.)

The range of applications of stochastic control problems is very wide. Originally, optimal stochastic continuous control problems were inspired by engineering problems in the continuous control of a dynamic system in the presence of random noise. In the last decades, problems in mathematical finance (portfolio optimization, options with variable exercise possibilities) and economics inspired many new developments, see [BR11] for some recent developments.

As a canonical general approach for solving a discrete time optimal control problem one may consider all possible future evolutions of the process at each time that a control choice is to be made. This method is well developed and may be effective in some special cases, but for more general problems such as optimal control of a diffusion in high dimensions, this approach is impractical. In [BKS09] a generic Monte Carlo approach combined with linear regression was proposed and studied, see also [BS18] for an overview. However, this approach has an important disadvantage – it may exhibit too

DOI 10.20347/WIAS.PREPRINT.2792 Berlin 2020

little flexibility for modeling highly non-linear behavior of the optimal value functions. Higher-degree polynomials or local polynomials (splines) can be used, but they may contain too many parameters and, therefore, either over-fit the Monte Carlo sample or prohibit parameter estimation because the number of parameters is too large. As an alternative to the polynomial bases, nonlinear approximation structures (e.g., artificial neural networks) can be used instead (see, e.g. [BCJ18]).

In [BSSZ20] a Monte Carlo based *reinforced regression* approach is developed for building sparse regression models at each backward step of the dynamic programming algorithm in the case of optimal stopping problems. In a nutshell, the idea is to start with a generic set of basis functions, which is systematically enlarged with highly problem dependent additional functions. The additional basis functions are constructed for the optimal stopping problem at hand without using a fixed predefined finite dictionary. Specifically, the new basis functions are learned during the backward induction via incorporating information from the preceding backward induction step. Thereby, basis functions highly specific to the problem at hand are constructed in a completely automatic way. [BSSZ20] report that the reinforced basis lead to increased precision over the starting set of basis functions, comparable to the standard regression algorithm based on a substantially increased set of basis functions. This improvement is obtained with limited increase of the computational cost.

In this work we carry over the approach of [BSSZ20] to a general class of discrete time optimal control problems including multiple stopping problems (allowing pricing of swing options) and a gas storage problem. This generalization turns out to be rather challenging as the complexity of using the previously constructed value function in regression basis at each step of the backward procedure becomes prohibitive when applying the original approach of [BSSZ20]. We overcome this computational bottleneck by introducing a novel version of the original reinforced regression algorithm where one uses a hierarchy of fixed time depth approximation of the optimal value function instead of a full depth approximation employed in [BSSZ20]. As a result we regain efficiency and are able to improve upon the standard linear regression algorithm in terms of achievable precision for a given computational budget.

More precisely, we construct a hierarchy $v^{(i)}$, $i = 0, \dots, I$, of (approximate) value functions with depth $I > 0$. Here, $v^{(0)}$ denotes the value functions obtained from the classical Monte Carlo regression algorithm. The higher levels $v^{(i)}$ are computed by regression based on a set of basis functions reinforced by the value function $v^{(i-1)}$ one level lower. This way, the added computational cost incurred from reinforcing the basis can be further decreased with minimal sacrifices of accuracy already for small values of I. In fact, we propose two versions of the algorithm. In the first version, the levels of the hierarchy of value functions are trained consecutively, allowing for an adaptive choice of the depth I of the hierarchy. In the second version, all the levels are trained concurrently, thereby improving the accuracy at each individual level. As a consequence, I needs to be fixed in advance and cannot be chosen adaptively in the second variant.

Outline of the paper

In Section 2 we describe a rather general setting for discrete stochastic control problems which we are going to use in this paper. The setting is based on [GHW11]. We recall the reinforced regression algorithm for optimal stopping problems by [BSSZ20] in detail in Section 3. There we also motivate the hierarchical construction of the new reinforced regression algorithm as restricted to the optimal stopping problem. The full algorithm – including both variants – is introduced in Section 4. A detailed analysis of computational costs is provided in Section 5. The next Section 6 provides a detailed convergence analysis for the standard and reinforced regression algorithms in the current setting. Extensive numerical examples including optimal stopping problems, multiple stopping problems and a gas storage optimization problem are provided in Section 7.

2 Setting

First, we present a proper setting for the construction and analysis of reinforced regression algorithms. The setting will be largely based on [GHW11]. We will consider stochastic control problems in discrete time with a finite action sets. We note that extensions to continuous action sets are certainly possible, but are left to future research.

We consider a filtration $\mathcal{F}_j$, $j = 0, \ldots, J$, which is extended by $\mathcal{F}_{-1} := \{\emptyset, \Omega\}$, $\mathcal{F}_{J+1} := \mathcal{F}_J$ for convenience. While the setting below could be (easily) formulated in a non-Markovian setting, we right away consider a Markov process X adapted to $(\mathcal{F}_j)_{j=0,\ldots,J}$. Note that we assume that the dynamics of the underlying process X does not depend on any control.

At time $0 \le j \le J$ we are given a *control* Y_j, which is $\mathcal{F}_{j-1}$-measurable, and an $\mathcal{F}_j$-measurable *cash-flow* $Z_j = H_j(a, Y_j, X_j)$ for some deterministic, measurable function H_j, where a is an *action* that we may choose at time j in some action space $\mathcal{K}$. Note that cash-flows may be positive or negative. We assume that the control Y_j takes values in a finite set $\mathcal{L}$, and that $\mathcal{K}$ is finite, as well. For a given value of the control $y \in \mathcal{L}$ and a given value x of the underlying process X_j, we are given a set

$$K_j(y, x) \subset \mathcal{K}, \quad j = 0, \ldots, J. \tag{2.1}$$

Only actions $a \in K_j(Y_j, X_j)$ are admissible at time j. Finally, if we choose $a \in K_j(Y_j, X_j)$, then the control is updated by

$$Y_{j+1} := \varphi_{j+1}(a, Y_j), \quad \varphi_{j+1} : \mathcal{K} \times \mathcal{L} \to \mathcal{L}. \tag{2.2}$$

Suppose that we are at time $0 \le j \le J$, the control and the underlying state process take values Y_j and X_j, respectively. For $\alpha = (a_j, \ldots, a_J) \in \mathcal{K}^{J-j+1}$ and $j \le \ell \le J-1$, we define

$$Y_{\ell+1}(\alpha; j, Y_j) := \varphi_{\ell+1}(a_\ell, Y_\ell(\alpha; j, Y_j)), \quad Y_j(\alpha; j, Y_j) := Y_j, \tag{2.3}$$

noting that $Y_\ell(\alpha; j, Y_j)$ only depends on $a_j, \ldots, a_{\ell-1}$. Additionally, we define $\mathbb{F}_{j,J}(\mathcal{K})$ to be the set of $(\mathcal{F}_\ell)_{\ell=j}^J$-adapted processes taking values in $\mathcal{K}$ indexed by $j, \ldots, J$. Clearly, if $\alpha \in \mathbb{F}_{j,J}(\mathcal{K})$ and $Y_j \in \mathcal{F}_{j-1}$, then the process $Y_\cdot(\alpha; j, Y_j)$ is previsible. The set of *admissible strategies or admissible policies* $\mathcal{A}_j$ is defined as follows:

$$\mathcal{A}_j(Y_j, X_{\ge j}) := \left\{ \alpha = (a_j, \ldots, a_J) \in \mathbb{F}_{j,J}(\mathcal{K}) \,\middle|\, a_\ell \in K_\ell(Y_\ell(\alpha; j, Y_j), X_\ell), \quad \ell = j, \ldots, J \right\}. \tag{2.4}$$

Now the central issue is the optimal control problem

$$V_j := \sup_{\alpha \in \mathcal{A}_j(Y_j, X_{\ge j})} \mathsf{E}_j \left[\sum_{\ell=j}^{J} H_\ell(a_\ell, Y_\ell(\alpha; j, Y_j), X_\ell) \right], \tag{2.5}$$

at a generic time $0 \le j \le J$, where E_j denotes the conditional expectation w.r.t. $\mathcal{F}_j$.

Taking advantage of the Markov property, we introduce the notation $\mathcal{A}_j(y, x) := \mathcal{A}_j(y, X^x_{\ge j})$, where $X^{j,x}$ denotes the Markov process X conditioned on $X_j = x$, and is defined for $j \le \ell \le J$. We may then define the *value function* as

$$v_j^*(y, x) := \sup_{\alpha \in \mathcal{A}_j(y,x)} \mathsf{E} \left[\sum_{\ell=j}^{J} H_\ell\left(a_\ell, Y_\ell(\alpha; j, y), X_\ell^{j,x}\right) \right], \tag{2.6}$$

which satisfies the *dynamic programming principle*:

$$v_j^*(y,x) = \sup_{a \in K_j(y,x)} \left(H_j(a,y,x) + \mathsf{E}\left[v_{j+1}^*(\varphi_{j+1}(a,y), X_{j+1}^{j;x}) \right] \right). \tag{2.7}$$

for $j = 0, \ldots, J$ (with $v_{J+1}^*(y,x) \coloneqq 0$).

Remark 2.1. *In light of portfolio optimization problems, we may also want to allow φ_{j+1} to additionally depend on X_j, X_{j+1}. Indeed, such a construction would allow Y_j to contain the value of the portfolio at time j. In such a case, the process Y would cease to be previsible. On the other hand, most of the following observations would continue to hold.*

Let us now give a few examples for classical stopping and control problems which fall into the above setup.

Example 2.2. *For a single optimal stopping problem with payoff $g_j \geq 0$ at time j, the set of possible control values is $\mathcal{L} = \{0, 1\}$, where a control state y denotes the number of remaining exercise opportunities. The action a takes the value 1 if we stop at the current time and 0 otherwise. Hence, we have*

$$K_j(y,x) = K(y) \coloneqq \begin{cases} \{0,1\}, & y = 1, \\ \{0\}, & y = 0, \end{cases}$$

implying that $\mathcal{K} = \{0, 1\}$. The cash-flow is defined by

$$H_j(a, Y_j, X_j) \coloneqq a\, g_j(X_j),$$

independent of the value of the control Y_j. Finally, the update function of the control is defined by

$$\varphi_{j+1}(a,y) \coloneqq \max(y - a, 0).$$

Example 2.3. *Let us now suppose that we have a multiple stopping problem with $L \in \mathbb{N}$ exercise rights. Again, the control state y signifies the remaining exercise opportunities, leading to $\mathcal{L} = \{0, 1, \ldots, L\}$. The admissible action set is now defined as*

$$K_j(y,x) = K(y) \coloneqq \begin{cases} \{0,1\}, & y \geq 1, \\ \{0\}, & y = 0. \end{cases}$$

Once again, we have $\mathcal{K} = \{0, 1\}$. The cash-flow H_j and the update function φ_{j+1} are defined as in Example 2.2.

Example 2.4. *Next we consider a more complicated multiple stopping problem. Consider a multiple stopping problem with refraction period $\overline{J}$, i.e., after exercising a right, we may not exercise another right for $\overline{J}$ periods. Moreover, the total set of possible exercise dates $\mathcal{J} \coloneqq \{0, 1, \ldots, J\}$ is divided into two subsets, $\mathcal{J} = \mathcal{J}_1 \sqcup \mathcal{J}_2$. When $j \in \mathcal{J}_1$, at most one optionality may be exercised, whereas at most two options may be exercised when $j \in \mathcal{J}_2$. Again, we have $L \in \mathbb{N}$ total exercise rights.*

In this case, the control y keeps track of the remaining time until the next optionality may be exercised. Hence, we set

$$\mathcal{L} \coloneqq \{0, \ldots, L\} \times \{0, \ldots, \overline{J}\},$$

where, for $y = (y^1, y^2) \in \mathcal{L}$, y^1 denotes the number of remaining exercise rights, whereas y^2 denotes the number of time periods, until we may exercise next. In other words, we may only exercise at time j if $Y_j^2 = 0$. Consequently, the set of admissible policies is given by

$$K_j(y) := \begin{cases} \{\,0\,\}, & y^1 = 0 \text{ or } y^2 > 0, \\ \{\,0,1\,\}, & y^1 > 0 \text{ and } y^2 = 0 \text{ and } j \in \mathcal{J}_1, \\ \{\,0,1,2\,\}, & y^1 \geq 2 \text{ and } y^2 = 0 \text{ and } j \in \mathcal{J}_2, \\ \{\,0,1\,\}, & y^1 = 1 \text{ and } y^2 = 0 \text{ and } j \in \mathcal{J}_2, \end{cases}$$

and $\mathcal{K} := \{\,0,1,2\,\}$. The cash-flow is defined by

$$H_j(a, Y_j, X_j) := a\, g_j(X_j),$$

and the update rule satisfies the following:

$$\varphi_{j+1}(a, y) := \begin{cases} (y^1, \max(y^2 - 1, 0)), & a = 0, \\ (\max(y^1 - a, 0), \overline{J}), & a \geq 1. \end{cases}$$

Example 2.5. *Consider a simple gas storage problem: given $N \in \mathbb{N}$ and $\Delta = 1/N$, we assume that the volume of gas in a storage can only be increased and decreased by a fraction Δ over a given time increment. Let the control y denote the status (fill level) of the storage at time j. Hence, we define*

$$\mathcal{L} := \{\,0, \Delta, 2\Delta, \dots, 1\,\}.$$

At time j, we may either sell Δ (volume of gas; $a = -1$), buy Δ ($a = +1$) – at the current market price X_j – or do nothing ($a = 0$). Hence, the admissible policy set is

$$K_j(y) := \begin{cases} \{\,0,1\,\}, & y = 0, \\ \{\,-1,0,1\,\}, & \Delta \leq y \leq 1 - \Delta, \\ \{\,-1,0\,\}, & y = 1, \end{cases}$$

with $\mathcal{K} = \{\,-1,0,1\,\}$, while the cash-flow is given by

$$H_j(a, Y_j, X_j) := -a \Delta X_j.$$

The update function in given by

$$\varphi_{j+1}(a, y) := ((y + a\Delta) \wedge 1) \vee 0.$$

3 Reinforced regression for optimal stopping

In this section, we recall the standard regression algorithm as well as the reinforced regression algorithm introduced in [BSSZ20] for optimal stopping problems. We will point out the drawbacks of the latter algorithm for more general control problems, and propose and motivate several modifications. However, for the purpose of a clear illustration, we will restrict ourselves in this section to the optimal stopping case.

Let us recall the optimal stopping setup from Example 2.2 and denote by $v_j^*(x)$ the value function at $j \in \{0, ..., J\}$ evaluated in $x \in \mathbb{R}^d$ and $y = 1$. Further recall that the dynamic programming principle is given by

$$v_j^*(x) = \max(g_j(x), c_j^*(x)), \quad 0 \le j \le J-1, \quad v_J^*(x) = g_J(x), \quad x \in \mathbb{R}^d,$$

where the continuation function is given by $c_j^*(x) = \mathsf{E}_j[v_{j+1}^*(X_{j+1}^{j,x})]$. For a set of basis function $\{\psi_1, ..., \psi_K\}$ with $\psi_i : \mathbb{R}^d \to \mathbb{R}$ and sample trajectories $(X_j^{(m)})_{0 \le j \le J, 1 \le m \le M}$ from the underlying Markov chain, the regression method due to Tsitsiklis-van Roy [TVR01], which we will refer to as the *standard regression method*, inductively constructs an approximation $v = (v_j)_{j=0,...,J}$ to the value function v^* as follows: For $j = J$ initialize $v_J = g_J$. For $j \in \{J-1, ..., 0\}$ set

$$v_j(x) := \max(g_j(x), c_j(x)), \quad c_j(x) = \sum_{k=1}^{K} \gamma_{j,k} \psi_k(x), \tag{3.1}$$

where the regression coefficients are given by the solution to the least squares problem

$$\gamma_{j,1}, ..., \gamma_{j,K} := \underset{\gamma_1, ..., \gamma_K}{\arg\min} \sum_{m=1}^{M} \left| v_{j+1}(X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) \right|^2. \tag{3.2}$$

The procedure is illustrated in Figure 1. Note that the costs of this algorithm are of the order $M \cdot J \cdot K^2$ (see, e.g., [BSSZ20] or Section 5).

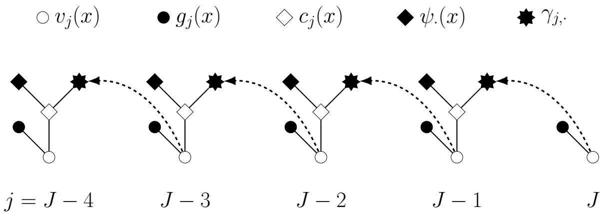

Figure 1: Illustration of standard regression approach due to Tsitsiklis-van Roy [TVR01]. The straight lines indicate the dependencies in the evaluation of c_j and v_j in (3.1). The dashed arrows start from the regression data v_{j+1} and symbolize the regression procedure (3.2).

One problem of the standard regression algorithm is that its performance strongly depends on the choice of basis functions. Indeed, while standard classes such as polynomials or splines usually form the backbone of the construction of basis functions, practitioners usually add customized basis functions, for instance the payoff function g_j and some functionals applied to it.

As a more systematic approach, the authors of [BSSZ20] proposed a *reinforced regression algorithm*. In this procedure the regression basis at each step of the backward induction is reinforced with the approximate value function from the previous step of the induction. The approximate continuation function at $j \in \{0, ..., J-1\}$ is then given by

$$c_j(x) := \sum_{k=1}^{K} \gamma_{j,k} \psi_k(x) + \gamma_{j,K+1} v_{j+1}(x),$$

where the regression coefficient are the solution to the least squares problem

$$\gamma_{j,1}, ..., \gamma_{j,K+1} := \underset{\gamma_1, ..., \gamma_{K+1}}{\arg\min} \sum_{m=1}^{M} \left| v_{j+1}(X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) - \gamma_{K+1} v_{j+1}(X_j^{(m)}) \right|^2.$$

Note that this procedure induces a recursion whenever an approximate value function is evaluated: in order to evaluate $v_j(x)$ we need to evaluate $c_j(x)$, which in turn requires an evaluation of $v_{j+1}(x)$ and so forth, until $v_J(x) = g_J(x)$ terminates the recursion. Figure 2 illustrates this procedure. The costs of the reinforced regression method are proportional to $M \cdot J \cdot K^2 + M \cdot J^2 \cdot K$ (see [BSSZ20]).

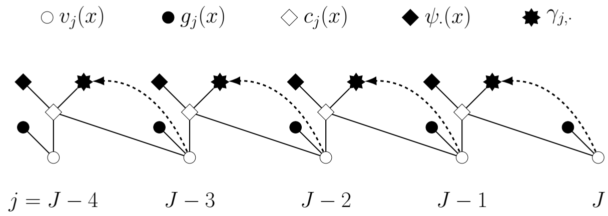

Figure 2: Illustration of the reinforced regression approach. Evaluation of v_j in the reinforced regression algorithm leads to a recursion with $J - j$ steps.

Despite the increased computational cost compared to the standard regression algorithm with the same set of basis functions $\psi_1, \dots, \psi_K$, the reinforced regression algorithm can improve the overall computational cost for a fixed error tolerance drastically. As a rule of thumb, [BSSZ20] report that the reinforced regression algorithm with a standard basis consisting of polynomials of a given degree leads to similar accuracy as the standard regression algorithm based on polynomials of one degree higher. In particular, the reinforced regression algorithm already outperforms the standard regression algorithm for small dimensions $d > 1$, as long as the number J of time-steps is not too large.

A direct generalization of the reinforced regression algorithm to more general control problems is certainly possible. The main difference to the optimal stopping problem is that for fixed time j we have to choose from many potential candidates to reinforce with, namely any $v_{j+1}(y, \cdot)$, $y \in \mathcal{L}$ is a candidate. Additionally, the dynamic programming principle (2.7) now entails a possibly non-trivial optimization problem in terms of the policy a. Especially the second point makes the recursion at the heart of the reinforced regression algorithm untenable for general control problems.

One solutions immediately comes to mind: If performing the recursion all the way to terminal time J is too costly, why not truncate at a certain recursion depth? This idea is, in principle, sound, and is the basis of the adaptations suggested below. However, some care is needed in the implementation of this idea. Indeed, if “truncation” simply were to mean “replace the reinforcing basis functions by 0 after a certain truncation step”, this would introduce a structural error in the procedure, as regression coefficients formerly computed in the presence of these basis functions would suddenly be incorrect. Instead, we propose to compute a *hierarchy* of reinforced regression solutions, corresponding to different “cut-off depths” of the recursion. This way, we can make sure that the coefficients are always consistent, that is, an error as mentioned above can be avoided. We introduce two versions, which both adhere to the same general idea, but differ in an important implementation detail.

The *hierarchical reinforced regression algorithm A* iteratively constructs approximations $(v^{(i)})_{i=0,1,\dots}$ to the true value function as follows: For $i = 0$ we construct $(v_j^{(0)})_{0 \le j \le J}$ using the standard regression method described above. Then for any $i \ge 1$, given that $v^{(l)}$ is already constructed for $0 \le l \le i-1$, define $v^{(i)}$ with the usual backwards induction, where the regression basis at step $j \in \{J-1, \dots, 0\}$ is reinforced with $v_j^{(i-1)}$. The approximate continuation function of the i^{th} iteration is given by

$$c_j^{(i)}(x) := \sum_{k=1}^{K} \gamma_{j,k}^{(i)} \psi_k(x) + \gamma_{j,K+1}^{(i)} v_{j+1}^{(i-1)}(x), \tag{3.3}$$

DOI 10.20347/WIAS.PREPRINT.2792

where the regression coefficients are the solutions to the least squares problem

$$\gamma_{j,1}^{(i)},...,\gamma_{j,K+1}^{(i)} := \underset{\gamma_1,...,\gamma_{K+1}}{\arg\min} \sum_{m=1}^{M} \left| v_{j+1}^{(i)}(X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) - \gamma_{K+1} v_{j+1}^{(i-1)}(X_j^{(m)}) \right|^2.$$

The procedure may be stopped after a fixed number of iterations, or using an adaptive criterion. An illustration of the method can be found in Figure 3. Note that the recursion that is started when evaluating $v_j^{(i)}(x)$ always terminates after at most i steps in the evaluation of $v_{j+i}^{(0)}(x)$ for $i \leq J-j$ or in $v_J^{(i-J-j)}(x) = g_J(x)$ for $J - j \leq i$.

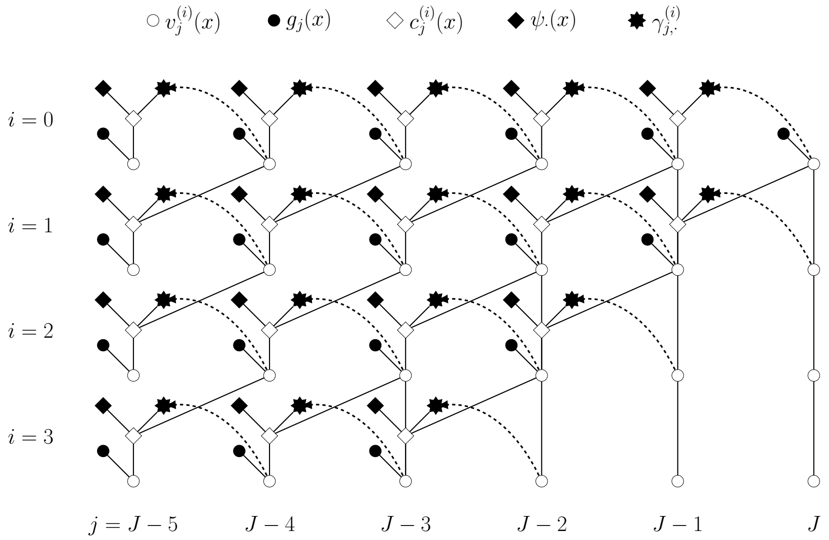

Figure 3: Illustration of the hierarchical reinforced regression algorithm A, for three iterations. In the lower right part of the diagram, the vertical lines indicate the equality $v_j^{(i)} \equiv v_j^{(l)}$ for $J - j \leq i$.

For a fixed number of iterations $i = \{0,...,I\}$ we can modify the structure of the previous method so that the primary iteration is the backwards induction over $j = \{J, J-1,...,0\}$ and the secondary iteration is over $i = \{0,...,I\}$. In this case we can further modify the algorithm by using $v_{j+1}^{(I)}$ as the regression target for the continuation functions $c_j^{(i)}$ for all $i \in \{0,...,I\}$. We name the resulting algorithm the *hierarchical reinforced regression algorithm B*. The approximate continuation function at step j and iteration i is then still given by (3.3) and the least squares problem is given by

$$\gamma_{j,1}^{(i)},...,\gamma_{j,K+1}^{(i)} := \underset{\gamma_1,...,\gamma_{K+1}}{\arg\min} \sum_{m=1}^{M} \left| v_{j+1}^{(I)}(X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) - \gamma_{K+1} v_{j+1}^{(i-1)}(X_j^{(m)}) \right|^2.$$

Also in this algorithm, the recursion that is started when evaluating $v_j^{(I)}$ stops after at most I steps. Figure 4 illustrates the procedure. The costs of the algorithm are discussed in Section 5.

4 Iterated reinforced regression for optimal control

Following the ideas and motivations of Section 3 we now present hierarchical reinforced regression algorithms for optimal control based on the Bellman equation (2.7). The algorithms are based on M sample trajectories $(X_j^{(m)})_{j=0,...,J,m=1,...,M}$ from the underlying Markov chain X. and some initial set

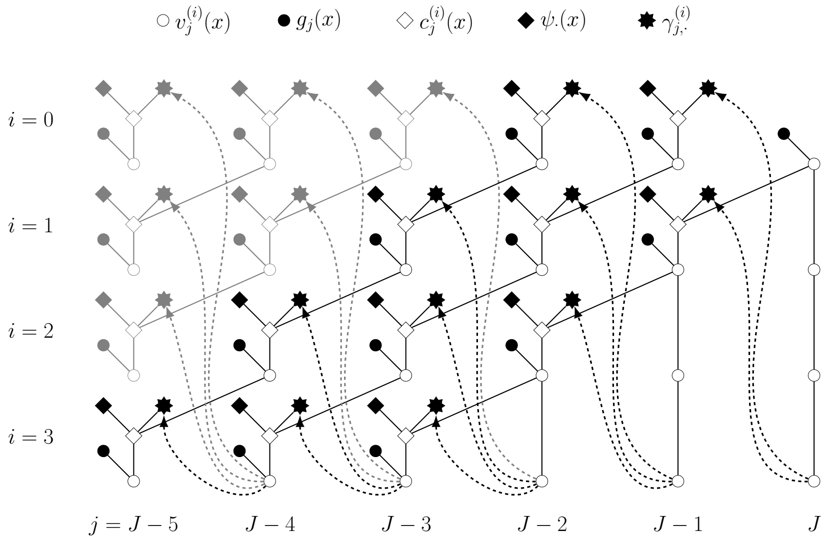

Figure 4: Illustration of the hierarchical reinforced regression algorithm B with $I=3$ iterations. In contrast to algorithm A illustrated in Figure 3, the regression is always applied to the approximation $v_{j+1}^{(I)}$, as indicated by the dashed arrows. The gray color of the elements in the upper left part of the diagram indicates that these elements are not needed for the construction and evaluation of $v^{(I)}$.

$\{\psi_1,...,\psi_K\}$ of basis functions $\psi_i:\mathbb{R}^d\to\mathbb{R}$. For each $y\in\mathcal{L}$ we will define a subset $\mathcal{L}^y\subset\mathcal{L}$ of cardinality $R^y:=|\mathcal{L}^y|$ and reinforce the basis $\{\psi_1,\ldots,\psi_K\}$ by $\{v_{j+1}(z,\cdot)|z\in\mathcal{L}^y\}$. The respective algorithms iteratively construct sequences of approximations to the value function

$$v^{(i)}=(v_j^{(i)})_{j=0,\ldots,J}\quad\text{with}\quad v_j^{(i)}:\mathcal{L}\times\mathbb{R}^d\to\mathbb{R},$$

for $i=\{0,1,...\}$ until the iteration is terminated.

4.1 Hierarchical reinforced regression algorithm A

For $i=0$ construct $v^{(0)}$ using the standard regression method inductively as follows: At the terminal time J initialize $v_J^{(0)}:=v_J$ where

$$v_J(y,x)=\max_{a\in K_J(y,x)}H_J(a,y,x),\quad\text{for all}\quad y\in\mathcal{L},\ x\in\mathbb{R}^d. \tag{4.1}$$

For a $j\in\{0,...,J-1\}$, assume that $v_l^{(0)}$ is already constructed for all $l\in\{j+1,...,J\}$. Then for each $y\in\mathcal{L}$ define the regression coefficients by solving the following least squares problem

$$\gamma_{j,1}^{(0),y},...,\gamma_{j,K}^{(0),y}\ :=\ \underset{\gamma_1,\ldots,\gamma_K}{\arg\min}\sum_{m=1}^{M}\left|v_{j+1}^{(0)}(y,X_{j+1}^{(m)})-\sum_{k=1}^{K}\gamma_k\psi_k(X_j^{(m)})\right|^2. \tag{4.2}$$

Next define the continuation function by

$$c_j^{(0)}(y,x):=\sum_{k=1}^{K}\gamma_{j,k}^{(0),y}\psi_k(x),\quad\text{for all}\quad y\in\mathcal{L},\ x\in\mathbb{R}^d \tag{4.3}$$

and the approximate value function $v_j^{(0)}$ through the dynamic programming principle

$$v_j^{(0)}(y,x):=\max_{a\in K_j(y,x)}\Big(H_j(a,y,x)+c_j^{(0)}(\varphi_j(a,y),x)\Big)\quad\text{for all}\quad y\in\mathcal{L},\ x\in\mathbb{R}^d. \tag{4.4}$$

Given the approximation $v^{(i)}$ for some $i \geq 0$ we construct a new approximation $v^{(i+1)}$ using reinforced regression inductively as follows: Initialize at the terminal time $v_J^{(i+1)} := v_J$. For $j \in \{0, ..., J-1\}$ assume that $v_l^{(i+1)}$ is already constructed for $l \in \{j+1, ..., J\}$. Then for each $y \in \mathcal{L}$ define the regression coefficients by solving the following least squares problem

$$\gamma_{j,1}^{(i+1),y}, ..., \gamma_{j,K+R^y}^{(i+1),y} := \underset{\gamma_1,...,\gamma_{K+R^y}}{\arg\min} \sum_{m=1}^{M} \Bigg| v_{j+1}^{(i+1)}(y, X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) - \sum_{k=1}^{R^y} \gamma_{K+k} v_{j+1}^{(i)}(y_k, X_j^{(m)}) \Bigg|^2, \tag{4.5}$$

where $\{y_k\}_{k=1,...,R^y} = \mathcal{L}^y$, and define the continuation function $c_j^{(i+1)}$ by

$$c_j^{(i+1)}(y, x) := \sum_{k=1}^{K} \gamma_{j,k}^{(i+1),y} \psi_k(x) + \sum_{k=1}^{R^y} \gamma_{j,K+k}^{(i+1),y} v_{j+1}^{(i)}(y_k, x), \tag{4.6}$$

for all $y \in \mathcal{L}$ and $x \in \mathbb{R}^d$. Finally define the approximation $v_j^{(i+1)}$ through the dynamic programming principle by

$$v_j^{(i+1)}(y, x) := \max_{a \in K_j(y,x)} \Big(H_j(a, y, x) + c_j^{(i+1)}(\varphi_j(a, y), x) \Big), \tag{4.7}$$

for all $y \in \mathcal{L}$ and $x \in \mathbb{R}^d$.

The iteration over $i \in \{0, 1, ...\}$ can be terminated after $I \in \mathbb{N}$ steps, yielding $v^{(I)}$ as an approximation to the true value function. Alternatively one can introduce an adaptive termination criterion, for example by comparing the relative change in the error of the least squares problem (4.5), terminating after the change falls under a given threshold.

Remark 4.1. *Recall that in the initialization we have $v_J^{(i)} = v_J^{(0)}$ for all $i \in \{1, ..., I\}$. It then follows inductively that*

$$v_j^{(i)} \equiv v_j^{(l)}, \quad \text{for all} \quad J - j \leq i \leq I, \quad l \geq i. \tag{4.8}$$

This identity can be used to reduce the costs of the algorithm, since the regression problem only needs to be solved for all (j, i) with $0 \leq j \leq J-1$ and $0 \leq i \leq (J-j) \wedge I$.

Remark 4.2. *More general or other forms of reinforced basis functions are certainly possible. The essential point is that they are based on the regression result from the preceding step in the backwards induction and the preceding iteration. Our specific choice may be seen as a natural primal choice. We left flexibility in the choice of the sets $\mathcal{L}^y$, for which depending on the cardinality of the set $\mathcal{L}$, possible choices are the trivial $\mathcal{L}^y = \mathcal{L}$ and $\mathcal{L}^y = \{y\}$, or $\mathcal{L}^y = \mathcal{L}'$ for some set $\mathcal{L}'$ independent of y, or more elaborately $\mathcal{L}_j^y = \{\varphi(a, y) \mid a \in K_j(y, x_j)\}$ for some $x_j \in \mathbb{R}^d$. Note that the use of a step dependent set $\mathcal{L}_j^y$ in the above method is straightforward.*

4.2 Hierarchical reinforced regression algorithm B

Fix a number of iterations $I \in \mathbb{N}$ and initialize the approximate value functions at the terminal time by $v_J^{(i)} \equiv v_J$ for all $i \in \{0, ..., I\}$, where v_J is given by (4.1). The approximate value functions at times previous to J are defined inductively as follows:

Let $j \in \{0, ..., J-1\}$ and assume that $v_l^{(i)}$ is already defined for all $l \in \{j+1, ..., J\}$ and $i \in \{0, ..., I\}$. For $i = 0$ and each $y \in \mathcal{L}$ determine the coefficients for the regression basis by solving the least squares problem

$$\gamma_{j,1}^{(0),y}, ..., \gamma_{j,K}^{(0),y} \ := \ \underset{\gamma_1,...,\gamma_K}{\arg\min} \sum_{m=1}^{M} \left| v_{j+1}^{(I)}(y, X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) \right|^2 \tag{4.9}$$

and define the approximate continuation function $c^{(0)}$ by (4.3). For $i \in \{1, ..., D\}$ and each $y \in \mathcal{L}$ determine the regression coefficients by solving the least squares problem

$$\begin{aligned} \gamma_{j,1}^{(i),y}, ..., \gamma_{j,K+R^y}^{(i),y} := \underset{\gamma_1,...,\gamma_{K+R^y}}{\arg\min} \sum_{m=1}^{M} \Bigg| \ & v_{j+1}^{(I)}(y, X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) \\ & - \sum_{k=1}^{R^y} \gamma_{K+k} v_{j+1}^{(i-1)}(y_k, X_j^{(m)}) \ \Bigg|^2, \end{aligned} \tag{4.10}$$

where $\{y_k\}_{k=1,...,R^y} = \mathcal{L}^y$, and define the continuation function $c_j^{(i)}$ by (4.6).

Finally, define the approximation to the value function $v_j^{(i)}$ for all $i = \{0, ..., I\}$ by (4.7). After ending the backwards induction use $(v_j^{(I)})_{j=0,...,J}$ as an approximation to the true value function.

Remark 4.3. *Note that the identity (4.8) also holds for the above algorithm. Moreover, since we are only interested in $v^{(I)}$, we can discard the computation of $c_j^{(i)}$ and $v_j^{(i)}$ for all $0 \leq j+i \leq I-1$, since they do not contribute to the construction of $v^{(I)}$. The least squares problem then only needs to be solved for $(j, i) \in \{0, ..., J-1\} \times \{0, ..., I\}$ with*

$$0 \leq j+i \leq I-1 \quad \textit{and} \quad 0 \leq i \leq (J-j) \wedge I.$$

Remark 4.4. *Choosing the number of iterations $I = J$ we then have from the previous remark that only the value functions on the diagonal $j = i$ need to be constructed. In this case, denote $v_j = v_j^{(j)}$, $c_j = c_j^{(j)}$ etc., and observe that the least squares problem which is solved in each step $j \in \{J-1, ..., 0\}$ of the backwards induction is given by*

$$\begin{aligned} \gamma_{j,1}^{y}, ..., \gamma_{j,K+R^y}^{y} := \underset{\gamma_1,...,\gamma_{K+1}}{\arg\min} \sum_{m=1}^{M} \Bigg| \ & v_{j+1}(y, X_{j+1}^{(m)}) - \sum_{k=1}^{K} \gamma_k \psi_k(X_j^{(m)}) \\ & - \sum_{k=1}^{R^y} \gamma_{K+k} v_{j+1}(y_k, X_j^{(m)}) \ \Bigg|^2. \end{aligned}$$

where $\{y_k\}_{k=1,...,R^y} = \mathcal{L}^y$. Hence, for $I = J$ the above algorithm represents a direct extension of the reinforced regression algorithm *in [BSSZ20] from optimal stopping to optimal control problems.*

5 Computational cost

We study the computational work of the modified reinforced regression algorithm of Section 4.2. In what follows, the following operations are considered to be performed at *constant* cost:

- Multiplications, additions and other primitive operations at cost c_*;

- Simulation from the distribution of the Markov process X_j at cost c_X;
- Evaluation of the standard basis functions ψ_i or of the payoff H_j at cost c_f;
- We set $R := \max_{y\in\mathcal{L}} R^y$.
- The cost of evaluating other non-trivial, but known functions φ (think of the value function when all the required regression coefficients are already known) will be denoted by $\mathrm{cost}(\varphi)$.

If an expression involves several such operations, then only the most expensive constant is reported. (E.g., evaluating a basis function and multiplying the value by a scalar constant is considered to incur a cost c_f.) We may also use constants c which do not depend on the specifics of the algorithm. We now go through the individual stages of the algorithm.

1 Simulating trajectories at cost $\mathrm{cost}_1 = c_X M(J+1)$.

2 Computing the terminal value function as in (4.1) at cost $\mathrm{cost}_2 = c_f\,|\mathcal{L}|\,|\mathcal{K}|$.

3 For fixed $0 \le j \le J-1$ and $y \in \mathcal{L}$ set up the least squares problem (4.9) at cost $M\left(c_f K + \mathrm{cost}\left(v_{j+1}^{(I)}\right)\right)$.

4 For fixed $0 \le j \le J-1$ and $y \in \mathcal{L}$, we solve the least squares problem (4.9) at cost $c_* M K^2$. The total cost is $\mathrm{cost}_4 = c_* J M K^2\,|\mathcal{L}|$.

5 For fixed $0 \le j \le J-1$, $y \in \mathcal{L}$, and $1 \le i \le I$ set up the least squares problem (4.10) at cost $M\left(c_f K + \mathrm{cost}\left(v_{j+1}^{(I)}\right) + R\,\mathrm{cost}\left(v_{j+1}^{(i-1)}\right)\right)$.

6 For fixed $0 \le j \le J-1$, $y \in \mathcal{L}$, and $1 \le i \le I$ solve the least squares problem (4.10) at cost $c_* M(K+R)^2$, leading to a total cost of $\mathrm{cost}_6 = c_* M(K+R)^2 J\,|\mathcal{L}|$.

List 5.1: Stages of the algorithm

For simplicity of the presentation, we shall only consider the following scenario:

Assumption 5.1. *The total set of reinforced basis functions contains all available value functions, i.e.,*

$$\bigcup_{y\in\mathcal{L}} \mathcal{L}^y = \mathcal{L}.$$

For fixed $0 \le i \le I$ and $0 \le j \le J$ let

$$\mathbf{v}_j^{(i)} := \left(v_j^{(i)}(y,\cdot)\right)_{y\in\mathcal{L}}, \quad \mathbf{c}_j^{(i)} := \left(c_j^{(i)}(y,\cdot)\right)_{y\in\mathcal{L}}. \tag{5.1}$$

The key step of the cost analysis is understanding the cost of evaluating the reinforced basis functions, which are, in turn, given in terms of reinforced basis functions at later time steps. We note that it is essential to analyze the cost of evaluating the full set of reinforced basis functions $\mathbf{v}_j^{(i)}$ rather than individual ones $v_j^{(i)}(y,\cdot)$, as the latter method would show us an apparent explosion of basis functions

DOI 10.20347/WIAS.PREPRINT.2792

as we increase time.[1] By (4.7), evaluating $\mathbf{v}_j^{(i)}$ requires evaluating the payoff functions for all combinations of controls $y \in \mathcal{L}$ and policies $a \in \mathcal{K}$, then evaluating $\mathbf{c}_j^{(i)}$, and taking the corresponding maxima. In total, this means

$$\operatorname{cost}\left(\mathbf{v}_j^{(i)}\right) \leq |\mathcal{K}|\,|\mathcal{L}|\,(c_f + c_*) + \operatorname{cost}\left(\mathbf{c}_j^{(i)}\right).$$

On the other hand, by (4.6) evaluating $\mathbf{c}_j^{(i)}$ requires K evaluations of standard basis functions, $K\,|\mathcal{L}|$ elementary operations for summing them, one evaluation of $\mathbf{v}_{j+1}^{(i-1)}$, and $|\mathcal{L}|^2$ elementary operations for their summation. In total, this means that

$$\operatorname{cost}\left(\mathbf{c}_j^{(i)}\right) \leq Kc_f + K\,|\mathcal{L}|\,c_* + \mathbb{1}_{i>0}\left(|\mathcal{L}|^2\,c_* + \operatorname{cost}\left(\mathbf{v}_{j+1}^{(i-1)}\right)\right).$$

This implies the cost estimate

$$\operatorname{cost}\left(\mathbf{v}_j^{(i)}\right) \leq |\mathcal{K}|\,|\mathcal{L}|\,(c_f + c_*) + Kc_f + K\,|\mathcal{L}|\,c_* + \mathbb{1}_{i>0}\left(|\mathcal{L}|^2\,c_* + \operatorname{cost}\left(\mathbf{v}_{j+1}^{(i-1)}\right)\right). \tag{5.2}$$

Lemma 5.2. *The cost of evaluating* $\mathbf{v}_j^{(i)}$, $i = 0, \dots, I$, $j = 0, \dots, J$ *can be bounded by*

$$\operatorname{cost}\left(\mathbf{v}_j^{(i)}\right) \leq \begin{cases} (i+1)\left(|\mathcal{K}|\,|\mathcal{L}|\,(c_f + c_*) + Kc_f + K\,|\mathcal{L}|\,c_*\right) + i\,|\mathcal{L}|^2\,c_*, & j+i \leq J, \\ |\mathcal{L}|\,|\mathcal{K}|\,(c_f + c_*) + (J-j)\left(|\mathcal{K}|\,|\mathcal{L}|\,(c_f + c_*) + Kc_f + (K+1)\,|\mathcal{L}|\,c_*\right), & j+i > J. \end{cases}$$

Proof. For $a \coloneqq |\mathcal{K}|\,|\mathcal{L}|\,(c_f + c_*) + Kc_f + K\,|\mathcal{L}|\,c_*$, consider the cost recursion

$$c(k+1) \leq a + |\mathcal{L}|\,c_* c(k), \quad k \geq 0.$$

Assuming that the recursion hits $i = 0$ before $j = J$, i.e., $i + j \leq J$, the cost $c(k) \coloneqq \operatorname{cost}\left(\mathbf{v}_{j+i-k}^{(k)}\right)$ satisfies the recursion with $c(0) \leq a$, and, hence, we obtain

$$c(k) \leq (k+1)a + k\,|\mathcal{L}|^2\,c_*.$$

This gives the first expression in the statement of the lemma with $k = i$.

On the other hand, if $i + j > J$, we hit $j = J$ before $i = 0$. In this case, $c(k) \coloneqq \operatorname{cost}\left(\mathbf{v}_{J-k}^{(i+j-J+k)}\right)$ satisfies the same recursion, but with initial value $c(0) \leq |\mathcal{L}|\,|\mathcal{K}|\,(c_f + c_*)$. ☐

In order to shorten notation, we introduce

$$\begin{aligned} a &\coloneqq |\mathcal{K}|\,|\mathcal{L}|\,(c_f + c_*) + Kc_f + K\,|\mathcal{L}|\,c_*, \\ b &\coloneqq |\mathcal{L}|^2\,c_*, \\ d &\coloneqq |\mathcal{L}|\,|\mathcal{K}|\,(c_f + c_*), \\ e &\coloneqq |\mathcal{K}|\,|\mathcal{L}|\,(c_f + c_*) + Kc_f + (K+1)\,|\mathcal{L}|\,c_*, \end{aligned}$$

so that the estimate of Lemma 5.2 shortens to

$$\operatorname{cost}\left(\mathbf{v}_j^{(i)}\right) \leq \begin{cases} (i+1)a + ib, & j+i \leq J, \\ d + (J-j)e, & j+i > J. \end{cases}$$

We next estimate the cost of setting up the regression problem (4.9).

[1] Suppose that each reinforced basis function $v_j^{(i)}(y, \cdot)$ depends on two reinforced basis functions $v_{j+1}^{(i-1)}(y', \cdot)$ and $v_{j+1}^{(i-1)}(y'', \cdot)$. If we follow this recursion for $l \leq i$ steps, we arrive at a total set of 2^l basis functions. The catch is that many, if not all, of these basis functions overlap with basis functions for other reinforced basis functions $v_j^{(i)}(\tilde{y}, \cdot)$.

Lemma 5.3. *The cost of setting up the regression problem for $c_j^{(0)}(y,\cdot)$, $j=0,\dots,J-1$, $y\in\mathcal{L}$, can be bounded by*

$$\mathrm{cost}_3 \le JMKc_f + M(J-I)\left((I+1)a + Ib\right) + MId + \frac{1}{2}MI(I+1)e.$$

Proof. We have

$$\begin{aligned}\mathrm{cost}_3 &\le \sum_{j=0}^{J-1} M\left(Kc_f + \mathrm{cost}\left(\mathbf{v}_{j+1}^{(I)}\right)\right)\\ &\le JMKc_f + M\sum_{j=0}^{J-I-1}\left[(I+1)a + Ib\right] + \sum_{\ell=1}^{I}\left[d+\ell e\right]\\ &= JMKc_f + M(J-I)\left((I+1)a+Ib\right) + MId + M\frac{1}{2}I(I+1)e. \qquad\square\end{aligned}$$

The cost for setting up the least squares problem (4.10) is computed in a similar way.

Lemma 5.4. *The cost of setting up the regression problem for $c_j^{(i)}(y,\cdot)$, $i=1,\dots,I$, $j=0,\dots,J-1$, $y\in\mathcal{L}$, can be bounded by*

$$\begin{aligned}\mathrm{cost}_5 \le JMKc_f + \frac{M}{2}I\left[(I+1)a + (I-1)b\right](J-I+2)+\\ + \frac{M}{6}I\left[11a + 2b - 9d + 5e + I(I+6)a + 3I(I+b) + 3I^2 d + I(I+6)e\right].\end{aligned}$$

Proof. A closer look at (4.10) reveals that the total cost of setting up all these least squares problems can be bounded by

$$\mathrm{cost}_5 \le \sum_{j=0}^{J-1} M\left(Kc_f + \sum_{i=1}^{I}\mathrm{cost}\left(\mathbf{v}_{j+1}^{(i-1)}\right)\right), \tag{5.3}$$

taking into account that $\mathbf{v}_{j+1}^{(I)}$ was already evaluated during the set-up of the least squares problem (4.9) and, hence, does not need to be evaluated again. Using Lemma 5.2, we obtain

$$\begin{aligned}\mathrm{cost}_5 &\le JMKc_f + M\sum_{j=0}^{J-1}\left[\sum_{i=0}^{(J-j)\wedge(I-1)}\left((i+1)a+ib\right) + \sum_{i=1+(J-j)\wedge(I-1)}^{I-1}\left(d+(J-j-1)e\right)\right]\\ &= JMKc_f + M\sum_{j=0}^{J-I+1}\left[\sum_{i=0}^{I-1}\left((i+1)a+ib\right)\right]+\\ &\quad + M\sum_{j=J-I}^{J-1}\left[\sum_{i=0}^{J-j}\left((i+1)a+ib\right) + \sum_{i=J-j+1}^{I-1}\left(d+(J-j-1)e\right)\right].\end{aligned}$$

Evaluating the double sums gives the estimate from the statement of the lemma. $\square$

Theorem 5.5. *The computational cost of the algorithm presented in Section 4.2 can be bounded by*

$$\mathrm{cost} \le \mathrm{const}\, MJ\left(c_X + I^2(K + |\mathcal{K}| + |\mathcal{L}|)\,|\mathcal{L}| + (K+R)^2\,|\mathcal{L}|\right),$$

where const *is a positive number independent of* $|\mathcal{K}|$, $|\mathcal{L}|$, K, J, *and* I.

Proof. We abandon the difference between c_f and c_*. Then,

$$\begin{gathered} a \leq \mathrm{const}\left(|\mathcal{K}|\,|\mathcal{L}| + K(|\mathcal{L}|+1)\right) \leq \mathrm{const}(K+|\mathcal{K}|)\,|\mathcal{L}|\,, \\ b \leq \mathrm{const}\,|\mathcal{L}|^2\,, \\ d \leq \mathrm{const}\,|\mathcal{K}|\,|\mathcal{L}|\,, \\ e \leq \mathrm{const}\,a \leq \mathrm{const}(K+|\mathcal{K}|)\,|\mathcal{L}|\,, \\ \max(a,b,e,d) \leq \mathrm{const}(K+|\mathcal{K}|+|\mathcal{L}|)\,|\mathcal{L}|\,. \end{gathered}$$

Similarly, we bound

$$\begin{aligned} \mathrm{cost}_3 &\leq \mathrm{const}\left[JMK + M(J-I)I(a+b) + MId + MI^2e\right] \\ &\leq \mathrm{const}\,MJ(I+1)(K+|\mathcal{K}|+|\mathcal{L}|)\,|\mathcal{L}|\,. \end{aligned}$$

For the second term in the cost estimate cost_5 given in Lemma 5.4, we bound

$$\frac{M}{2}I\left[(I+1)a + (I-1)b\right](J-I+2) \leq \mathrm{const}\,MI^2(J-I+2)(K+|\mathcal{K}|+|\mathcal{L}|)\,|\mathcal{L}|\,.$$

The third term in the cost estimate cost_5 is bounded by

$$\begin{aligned} \frac{M}{6}I\left[11a + 2b - 9d + 5e + I(I+6)a + 3I(I+b) + 3I^2d + I(I+6)e\right] \leq \\ \leq \mathrm{const}\,MI^3(K+|\mathcal{K}|+|\mathcal{L}|)\,|\mathcal{L}|\,. \end{aligned}$$

Hence, we obtain

$$\mathrm{cost}_5 \leq \mathrm{const}\,M\left[JK + I^2J(K+|\mathcal{K}|+|\mathcal{L}|)\,|\mathcal{L}|\right].$$

Further, note that $\mathrm{cost}_3 \leq \mathrm{const}\,\mathrm{cost}_5$, and similarly for cost_2. We also use that $\mathrm{cost}_4 \leq \mathrm{const}\,\mathrm{cost}_6$. The total cost can now be expressed as follows

$$\begin{aligned} \mathrm{cost} &\leq \mathrm{const}\left[\mathrm{cost}_1 + \mathrm{cost}_5 + \mathrm{cost}_6\right] \\ &\leq \mathrm{const}\left(MJc_X + M\left[JK + I^2J(K+|\mathcal{K}|+|\mathcal{L}|)\,|\mathcal{L}|\right] + M(K+R)^2J\,|\mathcal{L}|\right). \qquad \square \end{aligned}$$

Remark 5.6. *Recall that Remark 4.4 introduced a significantly cheaper variant of algorithm B for the case $I = J$. It is easy to see that the computational cost of this variant is bounded by*

$$\mathrm{cost} \leq \mathrm{const}\,MJ\left(c_X + J(K+|\mathcal{K}|+|\mathcal{L}|)\,|\mathcal{L}| + (K+R)^2\,|\mathcal{L}|\right),$$

i.e., the total cost is proportional to J^2 rather than J^3. Indeed, the main difference in the cost analysis as compared to the full modified algorithm is that (5.3) *can be replaced by*

$$\mathrm{cost}_5 \leq \sum_{j=0}^{J-1} M\left(Kc_f + \mathrm{cost}\left(\mathbf{v}_{j+1}^{(J-j-1)}\right)\right).$$

Note that this essentially corresponds to the algorithm of [BSSZ20] directly generalized to optimal control problems.

6 Convergence analysis

In this section we analyze the convergence properties of the standard and reinforced regression algorithms introduced in the previous sections. For related convergence analysis in the case of optimal stopping problems we refer the interested reader to [Zan13] and [BS20], see also [BRS20]. Henceforth we assume that

$$\max_{j=0,\ldots,J}\sup_{y\in\mathcal{L},a\in A}\sup_{x\in\mathcal{X}}|H_j(a,y,x)|\le C_H, \tag{6.1}$$

then all the value functions

$$v_j^*(y,x):=\sup_{\alpha\in\mathcal{A}_j(y,x)}\mathsf{E}\left[\sum_{\ell=j}^{J}H_\ell\left(a_\ell,Y_\ell(\alpha;j,y),X_\ell^{j,x}\right)\right]$$

are uniformly bounded by JC_H. Fix a sequence of spaces Ψ_j, $j=0,\ldots,J$, of functions defined on $\mathcal{X}$. We stress that these spaces must not be linear at this point. Construct the corresponding sequence of estimates $(v_{j,M}(y,x))_{j=0}^J$ via

$$\begin{aligned} v_{J,M}(y,x)&=\sup_{a\in K_j(y,x)}H_J(a,y,x)\quad\text{and}\\ v_{j,M}(y,x)&=\sup_{a\in K_j(y,x)}\left(H_j(a,y,x)+T_W\mathcal{P}_{j,M}[v_{j+1,M}](\varphi_{j+1}(a,y),x)\right),\quad j<J,\end{aligned} \tag{6.2}$$

where $\mathcal{P}_{j,M}[g](z,x)$ stands for the empirical projection of the conditional expectation $\mathsf{E}[g(z,X_{j+1}^{j,x})]$ on Ψ_j, based on a sample

$$\mathcal{D}_{M,j}=\left\{(X_j^{(m)},X_{j+1}^{(m)}),\quad m=1,\ldots,M\right\} \tag{6.3}$$

from the joint distribution of (X_j,X_{j+1}), that is,

$$\mathcal{P}_{j,M}[g](z,\cdot)\in\arg\inf_{\psi\in\Psi_j}\sum_{m=1}^{M}\left[\left|g(z,X_{j+1}^{(m)})-\psi(X_j^{(m)})\right|^2\right].$$

In (6.2) T_W is a truncation operator at level $W=JC_H$ defined by

$$T_Wf(x)=\begin{cases}f(x), & |f(x)|\le W,\\ W\operatorname{sign}(f(x)), & \text{otherwise.}\end{cases}$$

Due to Theorem 11.5 in [GKKW02], one has for all g with $\|g\|_\infty\le W$, $j=0,\ldots,J-1$, and all $z\in\mathcal{L}$, that

$$\begin{aligned}&\mathsf{E}\left[\left\|T_W\mathcal{P}_{j,M}[g](z,\cdot)-\mathsf{E}\left[g(z,X_{j+1}^{j,\cdot})\right]\right\|_{L_2(\mu_j)}^2\right]\\ &\qquad\le\varepsilon_{j,M}^2+2\inf_{w\,\in\,\Psi_j}\|g(z,\cdot)-w\|_{L_2(\mu_j)}^2\quad\text{with}\quad\varepsilon_{j,M}^2:=cW^4\frac{1+\log M}{M}\mathtt{VC}(\Psi_j),\end{aligned} \tag{6.4}$$

where $\mathtt{VC}(\Psi_j)$ is the Vapnik-Chervonenkis dimension of Ψ_j (see Definition 9.6 in [GKKW02]), μ_j is the distribution of X_j, and c is an absolute constant. In order to keep the analysis tractable, we assume that the sets $\mathcal{D}_{M,j}$ are independent for different j. More specifically, we consider an algorithmic framework based on (6.2), where for every exercise date the samples (6.3) are simulated independently, and consider the information sets

$$\mathcal{G}_{j,M}:=\sigma\left\{\mathbf{X}^{j;M},\ldots,\mathbf{X}^{J;M}\right\}\text{ with }\mathbf{X}^{j;M}:=\left(X_j^{(m)},X_{j+1}^{(m)},\quad m=\,1,\ldots,M\right).$$

Let us define for $j < J$, $z \in \mathcal{L}$, $x \in \mathcal{X}$,

$$\widehat{C}_j(z,x) := T_W \mathcal{P}_{j,M}[v_{j+1,M}](z,x), \tag{6.5}$$

and for a generic (exact) dummy trajectory $(X_l)_{l=0,\dots,J}$ independent of $\mathcal{G}_{j,M}$, let

$$\widetilde{C}_j(z,x) := \mathsf{E}_{\mathcal{G}_{j+1,M}}\left[v_{j+1,M}(z, X^{j,x}_{j+1})\right]. \tag{6.6}$$

Note that $\widetilde{C}_j(\cdot,\cdot)$ is a $\mathcal{G}_{j+1,M}$-measurable random function while the estimate $\widehat{C}_j(\cdot,\cdot)$ is a $\mathcal{G}_j$-measurable one. We further define

$$C^*_j(z,x) = \mathsf{E}\left[v^*_{j+1}(z, X^{j,x}_{j+1})\right], \quad j < J. \tag{6.7}$$

The following lemma holds.

Lemma 6.1. *We have that,*

$$\mathsf{E}\left[\left\|\sup_{z\in\mathcal{L}}\left|\widetilde{C}_j(z,\cdot) - C^*_j(z,\cdot)\right|\right\|^2_{L_2(\mu_j)}\right] \leq \mathsf{E}\left[\left\|\sup_{z\in\mathcal{L}}\left|\widehat{C}_{j+1}(z,\cdot) - C^*_{j+1}(z,\cdot)\right|\right\|^2_{L_2(\mu_{j+1})}\right]. \tag{6.8}$$

Proof. Let X be a generic (exact) dummy trajectory independent of $\mathcal{G}_{j+1,M}$. Then from (6.6), and (6.7) we see that for $j < J$,

$$\left|\widetilde{C}_j(z,X_j) - C^*_j(z,X_j)\right| \leq \mathsf{E}_{\mathcal{G}_{j+1,M}}\left[\left|v_{j+1,M}(z,X_{j+1}) - v^*_{j+1}(z,X_{j+1})\right| \middle| X_j\right] \tag{6.9}$$

Next, by (2.7), (6.2), (6.5), and (6.7) we have that

$$\begin{aligned}\left|v_{j+1,M}(z,x) - v^*_{j+1}(z,x)\right| &\leq \sup_{a\in K_{j+1}(z,x)}\left|\widehat{C}_{j+1}(\varphi_{j+2}(a,z),x) - C^*_{j+1}(\varphi_{j+2}(a,z),x)\right| \\ &\leq \sup_{z'\in\mathcal{L}}\left|\widehat{C}_{j+1}(z',x) - C^*_{j+1}(z',x)\right|.\end{aligned} \tag{6.10}$$

Hence, by (6.9) one has that

$$\sup_{z\in\mathcal{L}}\left|\widetilde{C}_j(z,X_j) - C^*_j(z,X_j)\right| \leq \mathsf{E}_{\mathcal{G}_{j+1,M}}\left[\sup_{z\in\mathcal{L}}\left|\widehat{C}_{j+1}(z,X_{j+1}) - C^*_{j+1}(z,X_{j+1})\right| \middle| X_j\right].$$

Finally, by taking the "all-in expectation" w.r.t. the law $\mu_j \otimes \mathbb{P}_M$, we observe that

$$\begin{aligned}&\mathsf{E}\left[\sup_{z\in\mathcal{L}}\left|\widetilde{C}_j(z,X_j) - C^*_j(z,X_j)\right|^2\right] \\ &\qquad\leq \mathsf{E}\left\{\mathsf{E}_{\mathcal{G}_{j+1,M}}\left[\sup_{z\in\mathcal{L}}\left|\widehat{C}_{j+1}(z,X_{j+1}) - C^*_{j+1}(z,X_{j+1})\right| \middle| X_j\right]\right\}^2 \\ &\qquad\qquad\leq \mathsf{E}\left[\sup_{z\in\mathcal{L}}\left|\widehat{C}_{j+1}(z,X_{j+1}) - C^*_{j+1}(z,X_{j+1})\right|^2\right]\end{aligned}$$

by Jensen's inequality and the tower property. □

In fact, Lemma 6.1 is the key to the next proposition.

Proposition 6.2. *Set*

$$\mathcal{E}_j := \left\| \sup_{z\in\mathcal{L}} \left| \widehat{C}_j(z,\cdot) - C_j^*(z,\cdot) \right| \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)}, \quad j = 0,\ldots,J-1,$$

with $\mathbb{P}_M$ *being the law of the sample* $X_j^{(m)}$, $m = 1,\ldots,M$, $j = 1,\ldots,J$. *Then it holds*

$$\mathcal{E}_j \le |\mathcal{L}| \left(\varepsilon_{j,M} + \sqrt{2} \sup_{z\in\mathcal{L}} \inf_{w\in\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \right) + |\mathcal{L}|\, \mathcal{E}_{j+1} \tag{6.11}$$

for all $j = 0,\ldots,J-1$, *with* $\mathcal{E}_J = 0$ *by definition.*

Proof. The case $j = J-1$ follows from (6.4) and the fact that $\widetilde{C}_{J-1} = C_{J-1}^*$ and Due to (6.4) we have with probability 1,

$$\mathsf{E}_{\mathcal{G}_{j+1,M}} \left[\left\| \widehat{C}_j(z,\cdot) - \widetilde{C}_j(z,\cdot) \right\|_{L_2(\mu_j)}^2 \right] \le \varepsilon_{j,M}^2 + 2 \inf_{w\in\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)}^2 . \tag{6.12}$$

Hence

$$\left\| \widehat{C}_j(z,\cdot) - \widetilde{C}_j(z,\cdot) \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \le \varepsilon_{j,M} + \sqrt{2} \inf_{w\in\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} . \tag{6.13}$$

By applying (6.13) it follows that

$$\begin{aligned} \left\| \widehat{C}_j(z,\cdot) - C_j^*(z,\cdot) \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} &\le \varepsilon_{j,M} + \sqrt{2} \inf_{w\in\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \\ &\quad + \left\| \widetilde{C}_j(z,\cdot) - C_j^*(z,\cdot) \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} . \end{aligned}$$

From this and Lemma 6.1 we imply

$$\begin{aligned} &\sup_{z\in\mathcal{L}} \left\| \widehat{C}_j(z,\cdot) - C_j^*(z,\cdot) \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \\ &\qquad \le \varepsilon_{j,M} + \sqrt{2} \sup_{z\in\mathcal{L}} \inf_{w\in\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \\ &\qquad\qquad + \left\| \sup_{z\in\mathcal{L}} \left| \widehat{C}_{j+1}(z,\cdot) - C_{j+1}^*(z,\cdot) \right| \right\|_{L_2(\mu_{j+1}\otimes\mathbb{P}_M)} \end{aligned}$$

and then (6.11) follows. □

Corollary 6.3. *Suppose that*

$$\sup_{z\in\mathcal{L}} \inf_{w\in\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \le \delta, \quad \mathtt{VC}(\Psi_j) \le D, \quad 0 \le j \le J-1,$$

for some $\delta > 0$ *and* $D > 0$. *Proposition 6.2 then yields for* $j = 0,\ldots,J-1$, *by using* (6.10)*,*

$$\left\| \sup_{z\in\mathcal{L}} \left| v_{j,M}(z,\cdot) - v_j^*(z,\cdot) \right| \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \le \left(cW^4 \frac{1+\log M}{M} D + \sqrt{2}\delta \right) \frac{|\mathcal{L}|^{J-j+1} - |\mathcal{L}|}{|\mathcal{L}| - 1}. \tag{6.14}$$

Corollary 6.4. *By inserting the estimate*

$$\begin{aligned} &\inf_{w\in\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} \\ &\qquad \le \left\| \widetilde{C}_j(z,\cdot) - C_j^*(z,\cdot) \right\|_{L_2(\mu_j\otimes\mathbb{P}_M)} + \inf_{w\in\Psi_j} \left\| C_j^*(z,\cdot) - w \right\|_{L_2(\mu_j)} \end{aligned}$$

in Proposition 6.2, we get the alternative recursion

$$\mathcal{E}_j \leq |\mathcal{L}| \left(\varepsilon_{K,M} + \sqrt{2} \sup_{z\in\mathcal{L}} \inf_{w\,\in\,\Psi_j} \left\| C_j^*(z,\cdot) - w \right\|_{L_2(\mu_j)} \right) + |\mathcal{L}| \, (1+\sqrt{2}) \mathcal{E}_{j+1},$$

and under the alternative assumption

$$\sup_{z\in\mathcal{L}} \inf_{w\,\in\,\Psi_j} \left\| C_j^*(z,\cdot) - w \right\|_{L_2(\mu_j)} \leq \delta, \quad \mathtt{VC}(\Psi_j) \leq D, \quad 0 \leq j \leq J-1,$$

for some $\delta > 0$ and $D > 0$, we obtain for $j = 0, ..., J$ the bounds

$$\left\| \sup_{z\in\mathcal{L}} \left| v_{j,M}(z,\cdot) - v_j^*(z,\cdot) \right| \right\|_{L_2(\mu_j \otimes \mathbb{P}_M)} \leq \left(cW^4 \frac{1+\log M}{M} D + \sqrt{2}\delta \right) |\mathcal{L}| \, \frac{\left((1+\sqrt{2}) \, |\mathcal{L}| \right)^{J-j} - 1}{(1+\sqrt{2}) \, |\mathcal{L}| - 1}.$$

Discussion The proposed reinforced regression algorithm with $I = J$ uses linear approximation spaces of the form

$$\Psi_j = \operatorname{span}\{\psi_1(x), \ldots, \psi_K(x), v_{j+1,M}(y_1, x), \ldots, v_{j+1,M}(y_R, x)\}, \quad j = 0, \ldots, J-1,$$

for $\mathcal{L} = \{y_1, \ldots, y_R\}$, where $\psi_1(x), \ldots, \psi_K(x)$ are some fixed basis functions (e.g. polynomials) on $\mathcal{X}$. In this case $\mathtt{VC}(\Psi_j) \leq K + R, \quad 0 \leq j \leq J-1$. In order to see the advantage of adding additional basis functions more clearly, estimate

$$\sup_{z\in\mathcal{L}} \inf_{w\,\in\,\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j \otimes \mathbb{P}_M)} \leq \sup_{z\in\mathcal{L}} \left\| \mathsf{E}_{\mathcal{G}_{j+1,M}} \left[v_{j+1,M}(z, X_{j+1}^{j,\cdot}) - v_{j+1,M}(z,\cdot) \right] \right\|_{L_2(\mu_j \otimes \mathbb{P}_M)}.$$

Assume additionally that

$$\max_{j=1,\ldots,J} \sup_{y\in\mathcal{L}, a\in A} \sup_{x\in\mathcal{X}} |H_j(a,y,x_1) - H_j(a,y,x_2)| \leq L_H |x_1 - x_2|, \tag{6.15}$$

for all $x_1, x_2 \in \mathcal{X}$ and

$$\max_{j=1,\ldots,J} \max_{\ell=j,\ldots,J} \mathsf{E}[|X_\ell^{j,x_1} - X_\ell^{j,x_2}|] \leq L_X |x_1 - x_2|, \quad \forall x_1, x_2 \in \mathcal{X} \tag{6.16}$$

for some constants $L_H > 0$, $L_X > 0$. Under assumptions (6.15) and (6.16) we then have

$$\max_{j=1,\ldots,J} \sup_{y\in\mathcal{L}} |v_j^*(y,x_1) - v_j^*(y,x_2)| \leq J L_X L_H |x_1 - x_2|, \quad \forall x_1, x_2 \in \mathcal{X}.$$

By using an additional truncation, one can achieve that the Lipschitz constants of the estimates $v_{j,M}(z,\cdot)$, $j = 0, \ldots, J-1$, are all uniformly bounded by a constant $J L_X L_H$ with probability 1. Hence

$$\sup_{z\in\mathcal{L}} \inf_{w\,\in\,\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j \otimes \mathbb{P}_M)} \leq J L_X L_H \left[\mathsf{E} \int |X_{j+1}^{j,x} - x|^2 \mu_j(dx) \right]^{1/2}.$$

This implies that if JL_H stays bounded for $J \to \infty$, (for example if H scales as $1/J$), then the approximation error $\inf_{w\,\in\,\Psi_j} \left\| \widetilde{C}_j(z,\cdot) - w \right\|_{L_2(\mu_j \otimes \mathbb{P}_M)}$ becomes small as $J \to \infty$. Note that the latter property can not be guaranteed when using fixed (nonadaptive) linear spaces Ψ_j. Of course, the exponential in J factor in (6.14) will lead to explosion of the overall error as $J \to \infty$, but the above observation still indicates that the inclusion of the functions $v_{j+1,M}(y_1, x), \ldots, v_{j+1,M}(y_R, x)$ into Ψ_j can significantly improve the quality of the estimates $v_{j,M}(z,\cdot)$ especially in the case of large J. Concerning the dependence of the bound (6.14) on J, we note that this estimation is likely to be too pessimistic, see also a discussion in [Zan13].

7 Numerical examples

We now present various numerical examples which demonstrate the accuracy of the reinforced regression algorithm in practice. To allow for a direct comparison with the reinforced regression algorithm of [BSSZ20], we first consider a (single) optimal stopping problem, i.e., a Bermudan option. Our next example is a multiple stopping problem, for which the hyperparameters already become critical. Finally, our last example is an optimal control of a gas storage.

Before, let us also mention how a lower estimate to the value of a control problem in a Markovian setting is calculated using the result of a regression procedure. Let c be an approximation to the function c^* given by

$$c_j^*(x,y) = \mathsf{E}\left[v_{j+1}^*(y, X_{j+1}^{j,x})\right], \quad x \in \mathbb{R}^d,\ y \in \mathcal{L},\ j \in \{0, ..., J-1\},$$

with $c_J \equiv c_J^* \equiv 0$ by convention. Using the hierarchical reinforced regression method, such an approximation is given by $c^{(I)}$ defined in (4.6). Further let $(X^{(m)})_{1\leq m\leq M_{\text{test}}}$ be sample trajectories from the underlying Markov chain, generated independently from the samples used in the regression procedure. Then we can iteratively define a sequence of polices $(\alpha^{(m)})_{1\leq m\leq M_{\text{test}}}$ with $\alpha^{(m)} = (a_0^{(m)}, ..., a_J^{(m)})$ by

$$a_j^{(m)} := \underset{a\in K_j(Y_j^{(m)}, X_j^{(m)})}{\arg\max} \left(H_j(a, Y_j^{(m)}, X_j^{(m)}) + c_j(\varphi(a, Y_j^{(m)}), X_j^{(m)})\right),$$

or all $m = 1, ..., M_{\text{test}}$ and $j = 0, ..., J$, where $Y_0^{(m)} := y_0 \in \mathcal{L}$ and $Y_{j+1}^{(m)} := \varphi_{j+1}(a_j^m, Y_j^{(m)})$. It then follows from the definition, that each $\alpha^{(m)}$ is an admissible sequence of policies, i.e. $\alpha^{(m)} \in \mathcal{A}_0(y_0, X^{(m)})$. Therefore, a lower estimate to the value $\mathsf{E}(v(y_0, X_0))$ is given by

$$\frac{1}{M_{\text{test}}} \sum_{m=1}^{M_{\text{test}}} \sum_{j=0}^{J} H_j(a_j^{(m)}, Y_j^{(m)}, X_j^{(m)}).$$

7.1 Bermudan max-call option

In this section we evaluate the performance of the hierarchical reinforced regression (HRR) method from Section 4 on the valuation of a Bermudan max-call option. Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\leq t\leq T}, P)$ be a filtered probability space on which a d-dimensional Brownian motion $W = (W(t))_{0\leq t\leq T}$ is defined. Further let $X = (X(t))_{0\leq t\leq T}$ be the geometric Brownian motion defined by

$$dX^k(t) = (r-\delta)\, X^k(t)\, dt + X^k(t)\, \sigma\, dW^k(t), \quad X^k(0) = x_0, \quad 0 \leq t \leq T, \quad k \in \{1, ..., d\},$$

where $x_0, r, \delta, \sigma > 0$. Option rights can be exercised on a predefined set of possible exercise dates $\{t_0, t_1, ..., t_J\}$, where at most one right can be exercised on any given date. Assume that the exercise dates are equidistant $t_j := j \cdot \Delta t$ for all $j = \{0, ..., J\}$ with $\Delta t := T/J$ and define the underlying Markov chain $(X_j)_{j=0,...,J}$ by $X_j = X(j\Delta t)$. Recall from Example 2.3 that in order to model a multiple stopping problem in the optimal control framework we define the set of policies by $\mathcal{K} = \{0, 1\}$ and the set of controls by $\mathcal{L} = \{0, ..., y_{max}\}$, where y_{max} is the number of exercise rights. Further, we define

$$\varphi_j(a, y) := (y-a)_+, \quad K_j(x, y) := \{0, 1 \wedge y\}, \quad H_j(a, y, x) := a \cdot g_j(x)\, e^{-t_j r},$$

for all $y \in \mathcal{L}$, $a \in \mathcal{K}$ and $j \in \{0, ..., J\}$, where g is the max-call pay-off function defined by

$$g(x) := (\max\{x^1, ..., x^d\} - C)_+, \quad x \in \mathbb{R}^d,$$

where $C \in \mathbb{R}_+$ is the option strike. Then the value function $v_0^*(y_{max}, x_0)$ defined in (2.6) yields the value of the Bermudan max-call option with underlying X and data $(d, J, T, y_{max}, x_0, C, r, \delta, \sigma)$.

7.1.1 Single exercise right

We will first consider the case of a single exercise right $y_{max} = 1$. This is a standard example in the literature and we refer to [AB04] for reference values. The performance of the reinforced regression method for this example was already analyzed in [BSSZ20]. We revisit this example in order to demonstrate that even in the optimal stopping case our novel HRR method allows for improvements in computational costs without sacrificing the quality of the estimations.

d	Basis	Lower bounds			Upper bounds
		Regression	HRR	Reinf. Reg.	
		$I = 0$	$I = 1$	$I = 9$	
2	Ψ_1	13.002 (0.023)	13.762 (0.015)	13.793 (0.015)	14.006 (0.036)
	$\Psi_{1,g}$	13.670 (0.018)	-	-	
	Ψ_2	13.761 (0.017)	13.863 (0.014)	13.875 (0.015)	
	Ψ_3	13.859 (0.016)	-	-	
4	Ψ_1	21.881 (0.025)	22.550 (0.019)	22.548 (0.019)	22.789 (0.039)
	$\Psi_{1,g}$	22.385 (0.022)	-	-	
	Ψ_2	22.531 (0.020)	22.654 (0.018)	22.666 (0.019)	
	Ψ_3	22.666 (0.020)	-	-	
6	Ψ_1	28.592 (0.024)	29.013 (0.021)	29.002 (0.021)	29.240 (0.036)
	$\Psi_{1,g}$	28.831 (0.016)	-	-	
	Ψ_2	29.024 (0.022)	29.109 (0.019)	29.120 (0.020)	
	Ψ_3	29.126 (0.022)	-	-	
8	Ψ_1	33.819 (0.026)	34.095 (0.022)	34.083 (0.022)	34.291 (0.036)
	$\Psi_{1,g}$	33.907 (0.026)	-	-	
	Ψ_2	34.137 (0.022)	34.182 (0.021)	34.193 (0.021)	

Table 1: Bounds (with 99.7% confidence interval) for the value of the Bermudan max-call option with data $J = 9$, $T = 1$, $y_{max} = 1$, $x_0 = C = 100$, $r = 0.05$, $\delta = 0.1$, $\sigma = 0.2$ and different numbers of underlying assets $d \in \{2, 4, 6, 8\}$. For all methods we used $M = 10^6$ training sample paths and $M_{\text{test}} = 10^7$ paths for calculating the lower bound. The dual upper bounds are calculated using the HRR method with $I = 1$, basis Ψ_2 and 10^4 outer and 10^3 inner sample paths.

Define the functions $f_i : \mathbb{R}^d \to \mathbb{R}, x \mapsto \mathrm{sort}(x^1, \dots, x^d)^i$, the ith largest entry of x, for $i = 1, \dots, d$ and consider the following three sets of regression basis functions:

$$\Psi_1 := \{1, f_1, ..., f_d\}, \quad \Psi_{1,g} := \Psi_1 \cup \{g\}, \quad \Psi_2 := \Psi_1 \cup \{f_i \cdot f_j \mid 1 \le i \le j \le d\},$$

$$\Psi_3 := \Psi_1 \cup \Psi_2 \cup \{f_i \cdot f_j \cdot f_k \mid 1 \le i \le j \le k \le d\}.$$

Note that the cardinalities of these sets are given by $|\Psi_1| = d+1$, $|\Psi_{1,p}| = d+2$, $|\Psi_2| = \frac{1}{2}d^2+\frac{3}{2}d+1$, and $|\Psi_3| = \frac{1}{6}d^3 + d^2 + \frac{11}{6}d + 1$, respectively. Regarding the HRR method, we use the algorithm of the second type described in Section 4.2. Further note that in the optimal stopping case there is only one choice for the set of reinforced value function for the HRR method since $\mathcal{L} = \{1\}$ and therefore we always set $\mathcal{L}^1 = \{1\}$.

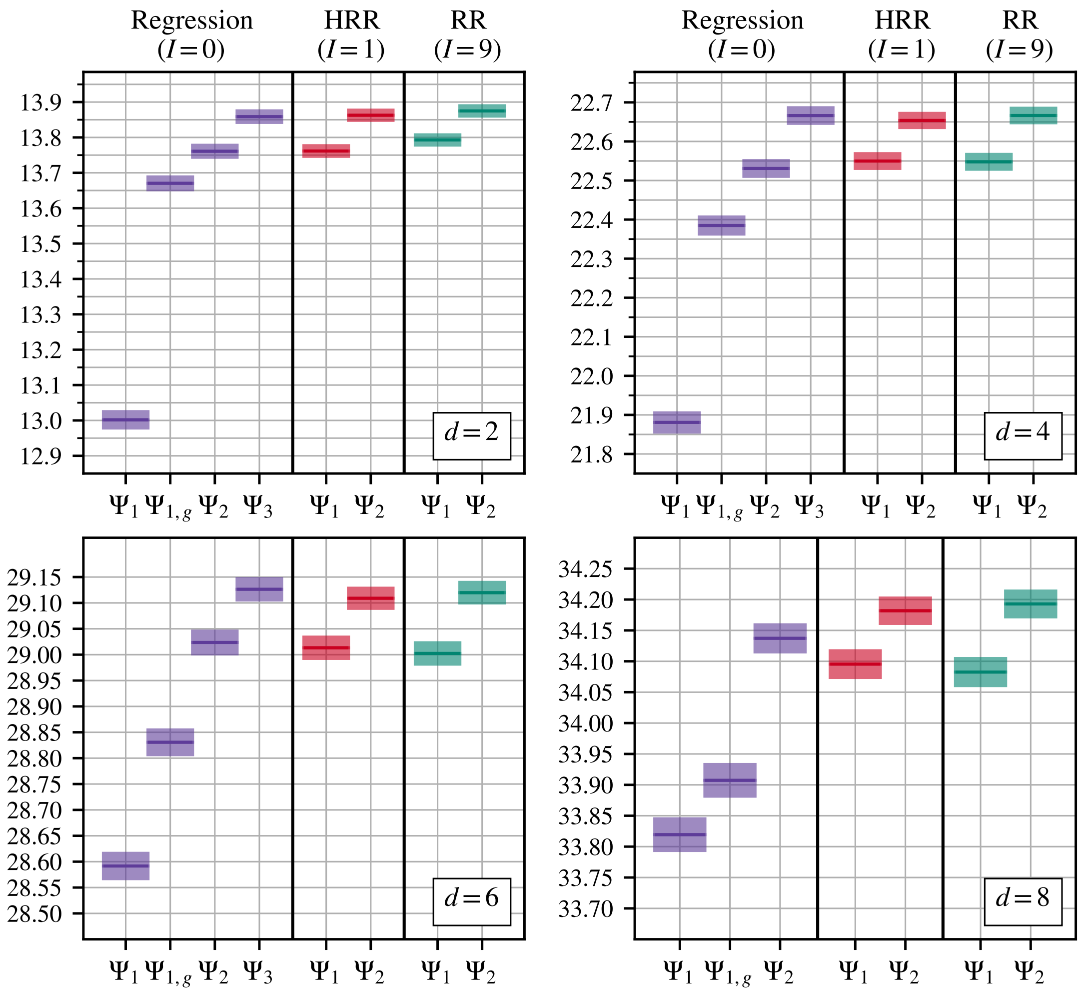

Figure 5: A visualization of the lower bounds from Table 1.

We considered two different set-ups for the comparison of the different methods:

- First we keep the number of exercises dates J fixed and vary the number of underlying assets d;
- Second we keep d fixed and vary J (while also keeping T fixed).

In Table 1 we present lower and upper bounds to the value of a Bermudan max-call option with a single exercise right for $J = 9$ and $d = 2, 4, 6, 8$. In the corresponding Figure 5 we have visualized the lower bounds for the comparison between the different regression methods. For each of the considered methods we used $M = 10^6$ simulated training samples paths to determine the regression coefficients and $M_{\text{test}} = 10^7$ paths for calculating the lower bounds.

We first observe that across all numbers of assets d the HRR method with the set of basis functions Ψ_1 performs significantly better then the standard regression method with the basis Ψ_1 and $\Psi_{1,g}$. The same holds true when comparing the methods using the regression basis Ψ_2. More importantly however, we observe that for $d \leq 6$ the HRR method with basis Ψ_1 yields lower bounds of the same quality as obtained with the standard method and the larger basis Ψ_2. The same holds true when comparing the HRR method with basis Ψ_2 against the standard method with the basis Ψ_3. In the case $d = 8$ assets, the lower bounds obtained with the HRR method and basis Ψ_1 lie just slightly below the values of the lower bounds obtained with standard method and the basis Ψ_2, however one has to keep in mind that in this case $|\Psi_1| = 9$ and $|\Psi_2| = 165$. Moreover, we see that the HRR method with a recursion depth $I = 1$ performs just as well as the (full depth) reinforced regression method $(I = J = 9)$.

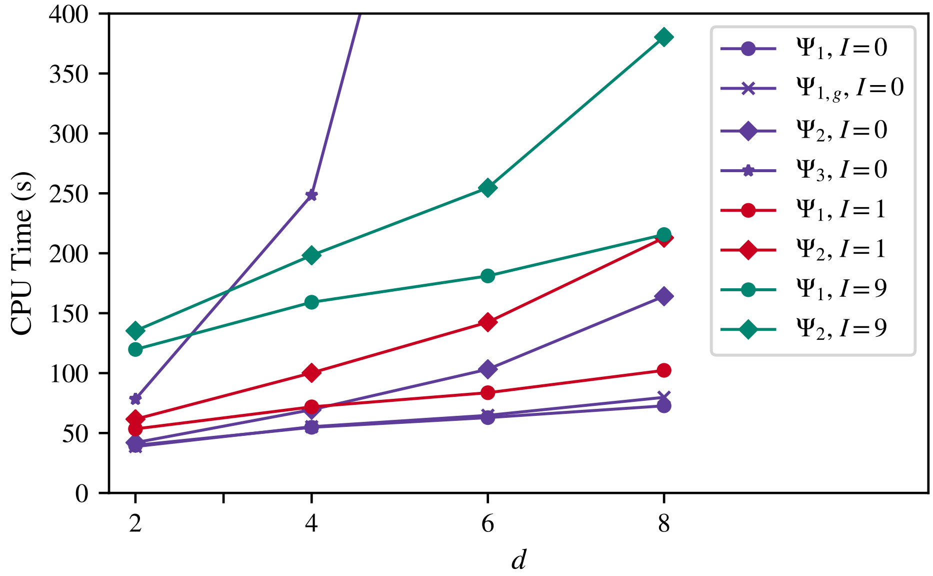

Figure 6: The elapsed CPU times during the backwards induction and calculation of the lower bounds from Table 1, plotted with respect to the number of underlying assets d.

Furthermore, in Figure 6 we have visualized the corresponding elapsed CPU times during the backwards induction and the calculation of the lower bounds. As foreshadowed in Section 5, we see that the computational costs of the HRR method are significantly reduced by choosing a small recursion depth I. In particular, we are able to state that for sufficiently large d ($d \geq 6$ respectively $d \geq 2$) the HRR method with recursions depth $I = 1$ and the basis Ψ_1 respectively Ψ_2 is more efficient then the standard method with the basis Ψ_2 respectively Ψ_3.

The results of the second set-up are visualized in Figure 7. In this case, we have approximated the value of Bermudan max-call options with a fixed number of assets $d = 4$ and different numbers of exercise dates $J = 9, 18, 36, 72$, while also keeping $T = 1$ fixed. When keeping all other parameters fixed, the value of the option clearly is non-decreasing in the number of exercise dates J. Our first observation is that for all considered methods there exists a threshold for J at which the performance worsens. Indeed, Corollary 6.3 shows that the approximation error depends exponentially on J.

However, in practice, some methods are less vulnerable to the error explosion in J than others. In this example, we see that the standard regression method with the basis $\Psi_{1,g}$ and Ψ_2, respectively, starts to perform worse for $J \geq 36$. The lower bounds calculated with the HRR method with the basis Ψ_1 and $I = 1$ stay approximately on the same level as the lower bounds calculated with the standard method and the basis Ψ_2, for all numbers of exercise dates. The lower bounds calculated with the standard method and the basis Ψ_3 first increase when moving from 9 to 18 exercise dates and decrease at last when moving from 36 to 72 exercise dates.

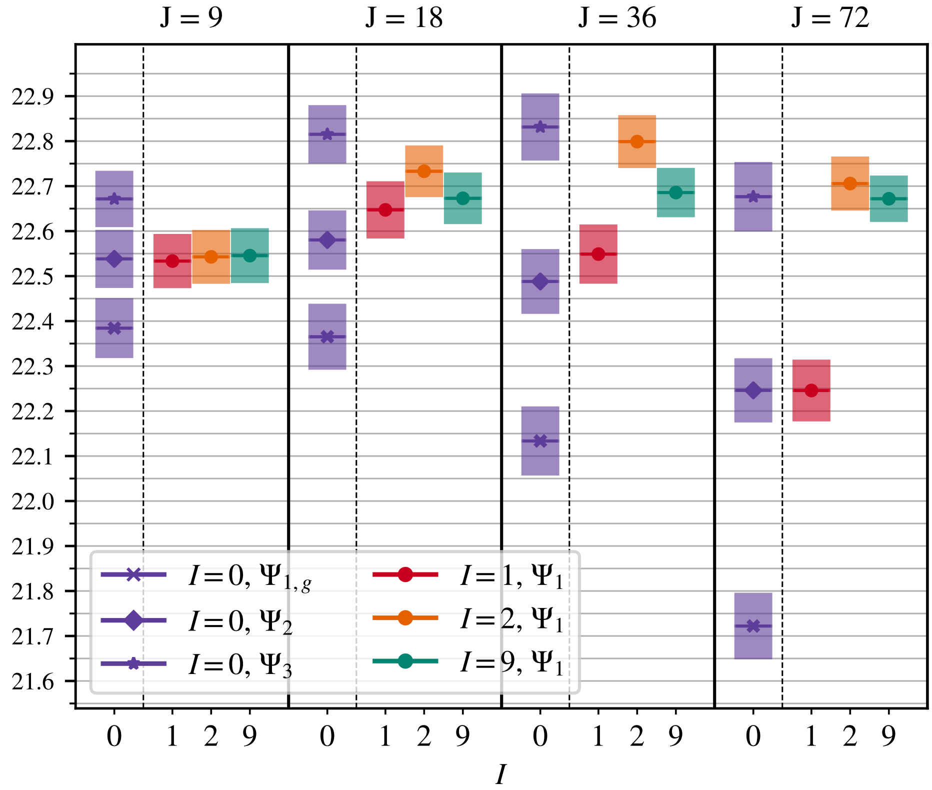

Figure 7: Visualization of the lower bounds (with 99.7% confidence intervals) of the values of Bermudan max-call options with $J = 9, 18, 36, 72$ exercise dates and $d = 4$, $T = 1$, $y_{max} = 1$, $x_0 = C = 100$, $r = 0.05$, $\delta = 0.1$, $\sigma = 0.2$. The values are obtained with the standard regression method $I = 0$ and the HRR method $I = 1, 2, 9$. For all methods we used $M = 10^6$ training sample paths and $M_{\text{test}} = 10^6$ sample paths for the calculation for the lower bounds.

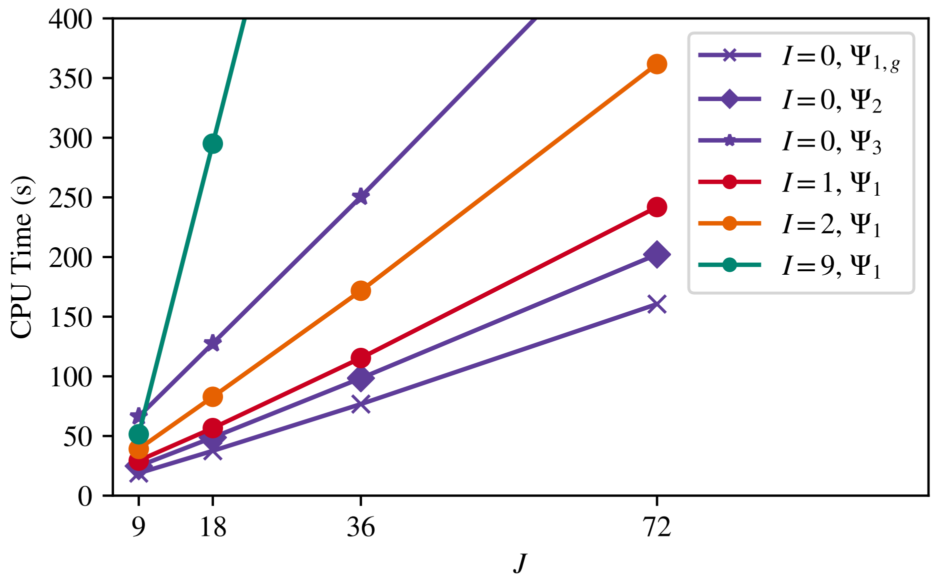

Figure 8: The elapsed CPU times during the backwards induction and calculation of the lower bounds in Figure 7, plotted with respect to the number of exercise dates J.

The main observation is that when we increase J, the HRR methods with $I = 2$ and $I = 9$ come closer to the lower bounds calculated with the standard method and the basis Ψ_3. This underlines the theoretical discussion for $J \to \infty$ from Section 6. Moreover, we see that the HRR method performs at least as well with $I = 2$ as with $I = 9$. The corresponding CPU times that elapsed during the regression procedure and the calculation of the lower bounds are presented in Figure 8. We see that for large enough J the HRR method with $I = 2$ is more efficient than the standard regression method with the basis Ψ_3. Comparing the HRR methods with $I = 2$ and $I = 9$, we see that choosing the parameter I small is necessary in order to obtain desirable efficiency. Since on the other hand, the HRR method with $I = 1$ performed significantly worse than with $I = 2$, we also see that in this case it was necessary to choose $I > 1$. These observations underline the relevance of the HRR in its full complexity even in the case of optimal stopping problems.

7.1.2 Multiple exercise rights

Next we consider a Bermudan max-call option with $y_{max} = 4$ exercise rights. In this case the HRR method allows for different possibilities of reinforced value functions depending on the choice of the sets $\mathcal{L}^y$ (recall Remark 4.2). Since y_{max} is small, we choose $\mathcal{L}^y \equiv \{1, 2, 3, 4\}$ for simplicity.

Basis	Regression	Hierarchical Reinforced Regression				
	$I = 0$	$I = 1$	$I = 2$	$I = 3$	$I = 4$	$I = 5$
Ψ_1	90.863 (0.072)	92.038	92.287	92.311	92.337	92.357
$\Psi_{1,g}$	91.837 (0.082)	-	-	-	-	-
Ψ_2	92.140 (0.070)	92.418	92.548	92.631	92.601	92.625
Ψ_3	92.571 (0.069)	-	-	-	-	-

Table 2: Lower bounds (with 99.7% confidence interval valid for each row) for the value of the Bermudan max-call option with data $J = 24$, $T = 2$, $y_{max} = 4$, $x_0 = C = 100$, $d = 5$, $r = 0.05$, $\delta = 0.1$, $\sigma = 0.2$. For all methods we used $M = 10^6$ training sample paths and $M_{\text{test}} = 10^7$ paths for calculating the lower bound. An upper bound to the value, calculated with the dual approach from [Sch12] and [BSZ13], is given by $92.971\ (0.043)$, with the HRR method with $I = 3$ and basis Ψ_2 using 10^5 outer and 10^3 inner sample paths.

In Table 3 and the corresponding Figure 9 we present lower bounds to the value of a Bermuda max-call option with $y_{max} = 4$ exercises rights, obtained with the standard regression method and HRR method for different choices of regression basis functions and the parameter I, with the implementation of the second type described in Section 4.2. We first observe that for a fixed set of basis functions Ψ_1 or Ψ_2 increasing the parameter I yields increased, and thus improved, lower bounds. This improvement is most significant when moving from $I = 0$ (standard regression) to $I = 1$ and from $I = 1$ to $I = 2$ and becomes less significant when further increasing I. Moreover, we observe that the HRR method with $I = 1$ and basis functions Ψ_1 yields better lower bounds then the standard regression method with the larger set of basis functions $\Psi_{1,g}$ and more importantly, for $I \geq 2$ the HRR with basis functions Ψ_1 method yields better lower bounds than the standard regression method with the even larger set of basis functions Ψ_2. This observation prevails when comparing the standard regression method with the basis Ψ_3 against the HRR method with the basis Ψ_2. We can therefore conclude that the HRR method yields results of better quality than standard regression using fewer regression basis functions. Moreover, we realize that up to changes that are insignificant with respect to the Monte Carlo error, the HRR reaches its best performance already for $I = 3$, thus further increasing I is not necessary.

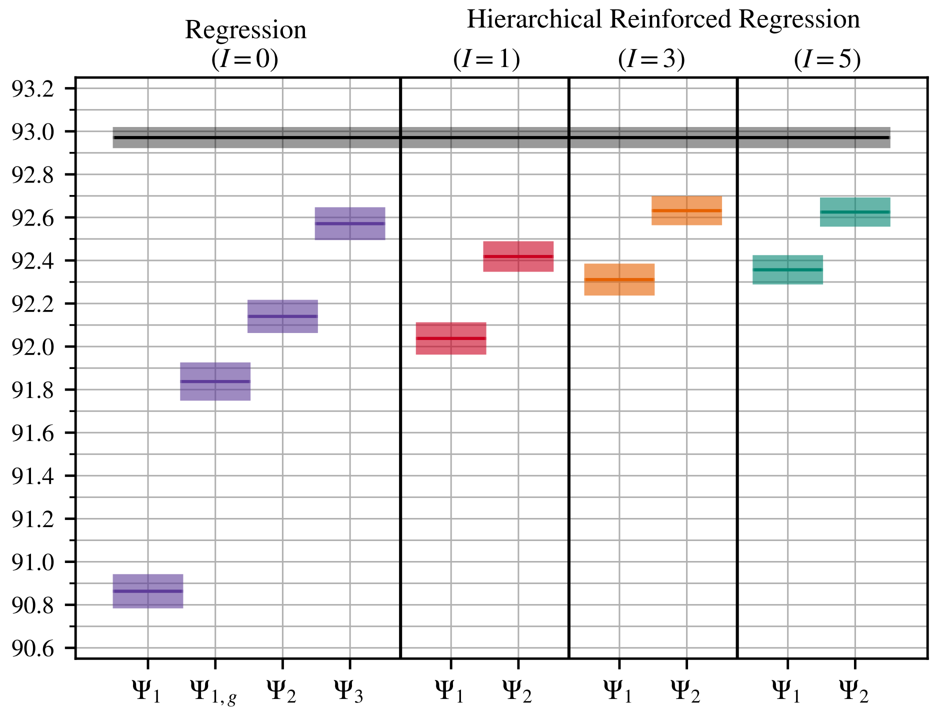

Figure 9: Visualization of the lower bounds with confidence intervals from Table 2. The black horizontal line represents the dual upper bound.

7.2 A gas storage problem

In this subsection we consider a gas-storage problem of the kind introduced in Example 2.5. In contrast to the example in the previous subsection, this optimal control problem is not of a multiple stopping type, which is a consequence of the anti-symmetry in the policy set: injection of gas into the facility $(a = 1)$, no action $(a = 0)$ and production of gas $(a = -1)$.

For the gas price we use a similar but slightly more elaborate model to the one proposed in [TDR09] (and also used in [GHW11]). More specifically, we use the following joint dynamics to model the *price of crude oil* X^1 and the *price of natural gas* X^2

$$\begin{aligned} \mathrm{d}X^1(t) &= \alpha_1(\beta - X^1(t))\mathrm{d}t + \sigma_1 X^1(t)\mathrm{d}W^1(t) + \left(J^1_{N(t-)+1} - X^1(t)\right)\mathrm{d}N(t) \\ \mathrm{d}X^2(t) &= \alpha_2(X^1(t) - X^2(t))\mathrm{d}t + \sigma_2 X^2(t)\mathrm{d}W^2(t) + \left(J^2_{N(t-)+1} - X^2(t)\right)\mathrm{d}N(t), \end{aligned} \tag{7.1}$$

for $0 \le t \le T$, where $\beta, \alpha_i, \sigma_i > 0$ for $i = 1, 2$, W^1 and W^2 are Brownian motions with correlation $\rho_W \in [0, 1]$, N is a Poisson process with intensity $\lambda > 0$ and $(J_k)_{k=1,\ldots}$ are i.i.d. normal distributed random vectors with $J^i_1 \sim \mathcal{N}(\mu_i, \eta_i^2)$, $\mu_i, \eta_i > 0$ and $\rho_J = \mathrm{Cor}(J^1_1, J^2_1) \in [0, 1]$. Moreover we assume that (W^1, W^2), N and (J^1, J^2) are independent. Note that both X^1 and X^2 are mean reverting processes with jump contributions. The oil price process X^1 reverts to the long-term constant mean β and the gas price process X^2 reverts towards the oil price X^1, which is aiming to model the well known strong correlation between crude oil and natural gas prices. Note also that we have assumed for simplicity that the jump signal, which has the purpose of modeling price peaks, is the same Poisson process for both oil and gas prices, however the magnitude of the jumps is given by different (but correlated) normal distributed random variables.

Denote by $(\widetilde{X}_j)_{j=1,\ldots,365}$ the 2-dimensional Markov chain that is obtained by discretizing the above SDE (7.1) with an Euler-scheme on the time interval interval $[0, 1]$. We assume that the manager of the gas storage facility has the possibility to buy and sell gas on a predefined set of dates in the year $\{t_j\}_{j=1,\dots,J} \subset \{1,\dots,365\}$ with $t_j = j \cdot \delta t$ and some $\delta t, J \in \mathbb{N}$ such that $\delta t \cdot J \leq 365$. The 2-dimensional Markov chain underlying the optimal control problem is then given by $X = (X_j)_{j=0,\dots,J}$ with $X_j := \widetilde{X}_{t_j}$.

Recall from Example 2.5 that we assume that the volume of gas in the storage can only be increased or decreased by a fraction $\Delta = 1/N$ for some $N \in \mathbb{N}$ over the time interval of δt days. The state space of the control variable is then given by $\mathcal{L} = \{0, \Delta, 2\Delta, \dots, 1\}$. Also recall the definition of the space of policies $\mathcal{K}$, the constraint sets K_j and the function φ_j from Example 2.5. We assume that there is no trading at $j = 0$ hence $K_0 \equiv \{0\}$. The cash-flow underlying to the optimal control problem only depends on the second component of the Markov chain X_j and is given by

$$H_j(a, y, X_j) = -a \cdot \Delta \cdot X_j^2 \cdot e^{-rj(\delta t/365)}, \quad a \in \mathcal{K},\ y \in \mathcal{L},\ j = 1, \dots, J,$$

where $r > 0$ is the interest rate.

We do not pay attention to the physical units of the parameters quantifying the gas storage capacity and the quotation of the gas price, since the linearity of the pay-off with respect to the parameter Δ and the gas price X_j^2 allows to properly scale the resulting value of the optimal control problem. The following specific choice of the price model parameters are oriented at the values in [TDR09]

$$\begin{gathered} \beta = 45, \quad \alpha_1 = 0.25, \quad \alpha_2 = 0.5, \quad \sigma_1 = \sigma_2 = 0.2, \quad \rho_W = 0.6, \\ \lambda = 2, \quad \mu_1 = \mu_2 = 100, \quad \eta_1 = \eta_2 = 30, \quad \rho_J = 0.6. \end{gathered} \tag{7.2}$$

Figure 10 shows a sample trajectory of the Markov chain X with the above parameters.

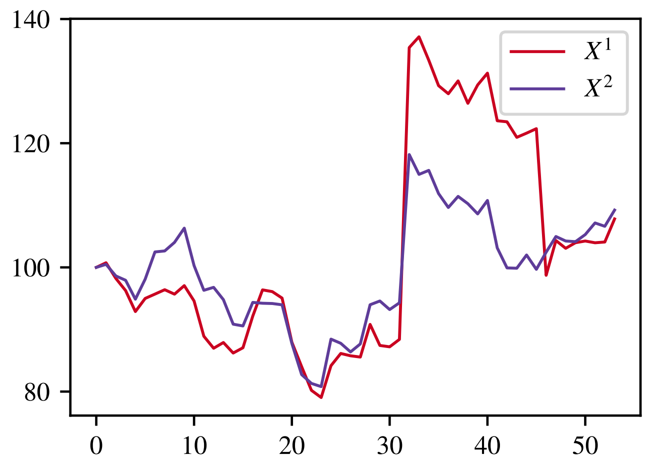

Figure 10: A sample path of the Markov chain $(X_j)_{j=0,\dots,J} = (\widetilde{X}_{t_j})_{j=0,\dots,J}$ where $t_j = j \cdot 7$ and $J = 52$. The approximation $\widetilde{X} = (\widetilde{X}_j)_{j=1,\dots,365}$ to the SDE (7.1) is simulated with the parameters given in (7.2) and $\widetilde{X}_0 = (100, 100)$. X^1 and X^2 serve as models for the prices of crude oil and natural gas.

Further we define the following sets of polynomial regression basis functions

$$\begin{aligned} P_i(X^2) &:= \left\{(x_1, x_2) \mapsto (x_2)^p \,\middle|\, p = 0, \dots, i\right\} \\ P_i(X^1, X^2) &:= \left\{(x_1, x_2) \mapsto (x_1)^p (x_2)^q \,\middle|\, p, q = 0, \dots, i, \ \ p + q \leq i\right\}. \end{aligned} \tag{7.3}$$

We have approximated the value of the gas storage problem with the following parameters

$$\delta t = 7, \quad J = 52, \quad \Delta = 1/8, \quad X_0 = (100, 100), \quad Y_0 = 4/8, \quad r = 0.1. \tag{7.4}$$

DOI 10.20347/WIAS.PREPRINT.2792

I	Basis	$v_0(Y_0, X_0)$	Lower bounds
0	$P_1(X^2)$	78.381	70.489 (0.066)
	$P_1(X^1, X^2)$	78.575	70.635 (0.068)
	$P_2(X^2)$	73.072	71.253 (0.068)
	$P_2(X^1, X^2)$	73.207	71.402 (0.068)
	$P_3(X^1, X^2)$	72.929	71.333 (0.081)
	$P_4(X^1, X^2)$	72.595	71.498 (0.068)
1	$P_1(X^1, X^2)$	71.991	71.579 (0.070)

Table 3: Approximate values and lower bounds for the gas storage problem with parameters given in (7.4) and price model parameters given in (7.2). The quantities were obtained with the standard regression method ($I = 0$) and the HRR method ($I = 1$), the different sets of basis functions (7.3), $M = 10^5$ training sample paths and $M_{\text{test}} = 10^6$ paths for calculating the lower bounds.

In this configuration the gas storage facility is initially loaded with half its capacity and the gas storage manager has the possibility to trade gas every seven days, and the amount by which the manager can inject or produce gas is one eighth of the total capacity. In Table 3 and the corresponding Figure 11 we present the numerical results that were obtained with the standard regression method and the HRR method. We used $M = 10^5$ training sample paths and $M_{test} = 10^6$ sample paths for the calculation of the lower bounds. For the set of reinforced basis functions in the HRR method we have chosen $\mathcal{L}^y \equiv \{Y_0\}$, i.e. in each step of the backwards induction, the regression basis was reinforced with only one function.

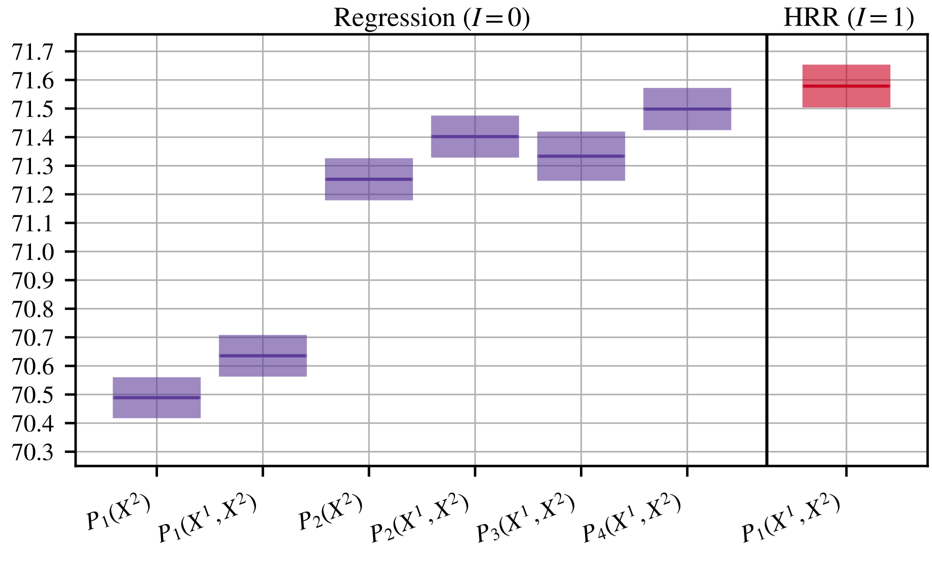

Figure 11: Visualization of the lower bounds with confidence intervals from Table 3.

We observe at first, that the lower bounds obtained with the standard regression method are improved when using polynomials in both variables (X^1, X^2) instead of just in the second variable X^2 (gas price) and are also improved when using polynomials of increasing order (with the only exception of the third degree polynomials). Moreover, we observe that the lower bound obtained with the HRR method, using the set of basis functions $P_1(X^1, X^2)$ and $I = 1$, lies above all lower bounds that were obtained with the standard regression, in particular the bound obtained with the regression basis $P_4(X^1, X^2)$.

8 Conclusions

In this paper, we present a new reinforced regression algorithm (HRR) for stochastic control problem, extending and improving the algorithm (RR) developed in [BSSZ20] for optimal stopping problems. The new algorithm provides a flexible hierarchical construction of approximate value functions, based on enlarging the set of basis functions used with approximate value functions at one level lower in the hierarchy. As value functions at level 0 correspond to value functions obtained by standard regression, the new hierarchical construction significantly reduces the computational footprint of reinforced regression compared with the algorithm of [BSSZ20], while keeping the improvement in accuracy.

In fact, for a hierarchy of value functions of depth $I+1$, the computational cost of HRR is proportional to JI^2 compared to J^2 for RR and J for standard regression (SR), see Theorem 5.5.

We provide a generic convergence analysis of regression type algorithms for the considered class of stochastic optimal control problems. In particular, we show that mean-squared error of the corresponding estimate for the value function $v^*(z,\cdot)$ is proportional to $\frac{\log M}{M}$ plus an approximation error. Our analysis further indicates that the use of adaptive (data dependent) approximation spaces as, for example, in HRR can significantly reduce the approximation error especially in the case of big J. These theoretical findings are underlined by careful numerical examples. In particular, we observe that HRR based on polynomial basis functions of a certain degree $\deg$ tend to produce results comparable to SR based on polynomial basis functions of degree $\deg+1$ or even higher, see Figures 5, 7, 9, and, most impressively, Figure 11.

The numerical results also indicate that, indeed, HRR with low depth of the hierarchy I already performs very well, even if $I \ll J$, see Figures 5, 7, 9. Hence, HRR performs with similar accuracy to RR, but much improved cost. Additionally, when comparing HRR with SRR at fixed accuracy, the computational cost of HRR is usually much smaller, especially for d large, see Figures 6, 8.

Finally, we note that the accuracy of the HRR method increases substantially when the time discretization is refined, i.e., when J is increased for fixed time horizon T. This theoretically very plausible observation (see Section 6) is backed up by numerical experiments, see Figure 7.

The present paper considers stochastic control problems in discrete time with finite action spaces. Of course, continuous time problems can be easily covered by discretizing time, a standard approach also considered in our theoretical analysis of Section 6. The same is, in principle, true for continuous action spaces, even though such problems lead to interesting new questions for the proposed HRR algorithms, in particular regarding to choice of (reinforced) basis functions in light of infinitely many possible choices. We postpone this problem to future research.

References

[AB04] L. Andersen and M. Broadie. A Primal-Dual Simulation Algorithm for Pricing Multi-Dimensional American Options. *Management Science*, 50(9):1222–1234, 2004.

[BCJ18] Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *arXiv preprint arXiv:1804.05394*, 2018.

[BKS09] D. Belomestny, A. Kolodko, and J. Schoenmakers. Regression methods for stochastic control problems and their convergence analysis. *SIAM Journal on Control and Optimization*, 48:3562–3588, 01 2009.

[BR11] Nicole Bäuerle and Ulrich Rieder. *Markov decision processes with applications to finance*. Springer Science & Business Media, 2011.

[BRS20] Christian Bayer, Martin Redmann, and John Schoenmakers. Dynamic programming for optimal stopping via pseudo-regression. *Quantitative Finance*, pages 1–16, 2020.

[BS18] Denis Belomestny and John Schoenmakers. *Advanced simulation-based methods for optimal stopping and control*. Palgrave Macmillan, London, 2018. With applications in finance.

[BS20] Denis Belomestny and John Schoenmakers. Optimal stopping of McKean–Vlasov diffusions via regression on particle systems. *SIAM Journal on Control and Optimization*, 58(1):529–550, 2020.

[BSSZ20] Denis Belomestny, John Schoenmakers, Vladimir Spokoiny, and Bakhyt Zharkynbay. Optimal stopping via reinforced regression. *Communications in Mathematical Sciences*, 16(1):109–121, 2020.

[BSZ13] Christian Bender, John Schoenmakers, and Jianing Zhang. Dual representations for general multiple stopping problems. *Math. Fin., to appear*, 2013.

[GHW11] Lajos Gyurko, B. Hambly, and Jan Witte. Monte Carlo methods via a dual approach for some discrete time stochastic control problems. *Mathematical Methods of Operations Research*, 81, 12 2011.

[GKKW02] László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Series in Statistics. Springer-Verlag, New York, 2002.

[Pha09] Huyên Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media, 2009.

[Sch12] J. Schoenmakers. A pure martingale dual for multiple stopping. *Finance Stoch.*, 16:319–334, 2012.

[TDR09] Matt Thompson, Matt Davison, and Henning Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics (NRL)*, 56(3):226–238, 2009.

[TVR01] John Tsitsiklis and Benjamin Van Roy. Regression methods for pricing complex American style options. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 12:694–703, 02 2001.

[Zan13] Daniel Z. Zanger. Quantitative error estimates for a least-squares Monte Carlo algorithm for American option pricing. *Finance Stoch.*, 17(3):503–534, 2013.