symmetry

MDPI

Article

An Iteration Algorithm for American Options Pricing Based on Reinforcement Learning

Nan Li

School of Mathematics, Jilin University, Changchun 130012, China; nli19@mails.jlu.edu.cn

Abstract: In this paper, we present an iteration algorithm for the pricing of American options based on reinforcement learning. At each iteration, the method approximates the expected discounted payoff of stopping times and produces those closer to optimal. In the convergence analysis, a finite sample bound of the algorithm is derived. The algorithm is evaluated on a multi-dimensional Black-Scholes model and a symmetric stochastic volatility model, the numerical results implied that our algorithm is accurate and efficient for pricing high-dimensional American options.

Keywords: American options; deep learning; Monte Carlo; optimal stopping; reinforcement learning

Citation: Li, N. An Iteration Algorithm for American Options Pricing Based on Reinforcement Learning. *Symmetry* **2022**, *14*, 1324. https://doi.org/10.3390/sym14071324

Academic Editors: Jinyu Li and Juan Luis García Guirao

Received: 30 May 2022
Accepted: 23 June 2022
Published: 27 June 2022

1. Introduction

The pricing of American options is an important issue in quantitative finance and stochastic processes [1,2]. Many popular derivative products in various financial sectors are of the American type, and can be exercised at any time before maturity. Therefore, considerable effort has been spent to obtain accurate and efficient methods for pricing American options (see, e.g., Hull [3]). When the dimension of the option is small, methods based on partial differential equations [4] and binomial trees [5] can be applied. However, the calculation costs of these methods increase exponentially as the dimension gets larger, thus making them inefficient for pricing options on many underlying assets, such as the widely used high-dimensional symmetric stochastic volatility models [6].

To treat American options on multi-dimensional underlying assets, many pricing methods based on Monte Carlo simulation have been proposed. The most popular are the regression-based methods proposed by Longstaff and Schwartz [7] and Tsitsiklis and Roy [8]. Through a backward iteration scheme, these methods can approximate the continuation value and a feasible exercise policy, such as linear regression [7], neural network [9], Gaussian process regression [10] and kernel ridge regression [11], all of which produce lower price bounds for American options. The dual approaches for American options were developed by Rogers [12] and Haugh and Kogan [13], these methods produce upper price bounds for options. However, in the computation of these methods, the continuation value at each date is approximated by different functions, and only the data at this date are used.

A different strand of the literature focuses on finding the optimal exercise policy by Monte Carlo sample [14,15]. These approaches consider a parametric class of exercise regions and maximize an estimate of the value function within the parametric class. Through optimization, all the sample data are used to approximate the optimal exercise policy. Recently, Bayer et al. [16] and Becker et al. [17] consider randomized stopping times in approximating the optimal exercise regions. However, in these approaches, the resulting loss function may still be non-concave and exhibit isolated local optima; thus, it is difficult to find the global optimum for the loss function [18,19].

Reinforcement learning, especially the policy iteration method, has achieved empirical success in high-dimensional control problems [20–22]. The basic idea of policy iteration is to compute the evaluation function of the policy in each iteration, after which an improved policy is computed from the function for the next iteration [23]. The pricing of American

options could be seen as a control problem, but there are only two possible actions, and these do not influence the underlying process. Reinforcement learning methods have already been applied to optimal stopping problems. Tsitsiklis and Roy [8] introduced the fitted Q-iteration for American option pricing based on the least-squares method. Yu and Bertsekas [24] proposed an algorithm based on projected value iteration, the convergence of this method for finite-state models was also obtained. Li et al. [25] considered a least squares policy iteration in the pricing of American options, and they give a finite-time bound for the algorithm. Becker et al. [26] developed an algorithm related to policy optimization for high-dimensional optimal stopping problems. Chen et al. [27] applied Zap-Q-learning for the optimal stopping problem, and established consistency of the algorithm for linear function approximation. Herrera et al. [28] considered a fitted Q-iteration based on randomized neural networks for the optimal stopping problems. However, most of these methods are a direct transformation of the method in reinforcement learning. Furthermore, there is a lack of analysis on the accuracy and efficiency of reinforcement learning in pricing high-dimensional American options.

In this paper, we propose an iteration algorithm for American options based on reinforcement learning. In each iteration, the expected discounted payoff of a family of stopping times is approximated by regression; thus, the data of all dates are used to improve the approximation of all dates. An improved family of stopping times was obtained based on the constructed function. After this procedure, an approximate optimal exercise policy was obtained. To provide theoretical guarantees, we developed a finite sample-error bound for the algorithm. In the numerical experiments, we considered the data generated by the multi-dimensional Black–Scholes model and a symmetric stochastic volatility model. The results showed that (a) our algorithm was accurate and efficient in pricing high-dimensional American options; (b) by using a function of time and underlying process, the continuation values can be approximated using a fraction of the parameters; (c) the methods based on reinforcement learning outperform the state-of-the-art methods in the pricing of American options.

The paper is organized as follows. In Section 2, we introduce the problem of pricing American options and illustrate the relationship of continuation values and stopping times. The efficient algorithm is described in Section 3. In Section 4, convergence rates of the algorithm are discussed. Numerical experiments of high-dimensional American options on multi-dimensional Black–Scholes model and a symmetric stochastic volatility model are given in Section 5. Finally, we conclude in Section 6. All proofs are found in the Appendix A.

2. Pricing of American Options and Stopping Times

In this section, we introduce to the pricing of American options. Let $\{X_t, 0 \le t \le T\}$ be a $\mathbb{R}^d$-valued Markov process, this process is defined on a filtered measurable probability space with a risk-neutral measure P. We assumed that the process records all relevant financial variables. In practice, the price of the American option was approximated by the price of a Bermudan option [11], which could be exercised at discrete time points $0 < t_1 < \cdots < t_N = T$. For $0 \le n \le N$, we represented t_n by n to simplify the notation in the following. We assumed that the risk-free discount factor between time points was constant, which was denoted by $\gamma \in (0, 1)$. The price $V_n(x)$ of the option at $n = 1, \ldots, N$ was given by the optimal stopping problems

$$V_n(x) = \sup_{\tau \in \mathcal{T}_n} \mathrm{E}\left[\gamma^{\tau - n} g(X_\tau) | X_n = x\right], \tag{1}$$

where $g(x)$ is the non-negative payoff function, $\mathcal{T}_n$ denotes the set of stopping times such that $n \leq \tau$. The price at time 0 is given by $V_0(x) = \mathrm{E}[\gamma V_1(X_1)|X_0 = x]$. We assume that g satisfies $\|g(X_n)\|_\infty \leq B$ for $n = 1, \ldots, N$, where $\|\cdot\|_p$ denotes the L^p-norm and $B > 0$. As we will see later, this assumption can be relaxed.

The optimal stopping problem (1) is solved by a family of optimal stopping times τ_n^*, $n = 1, \ldots, N$, that satisfies the consistency property $\tau_n^* > n \Longrightarrow \tau_n^* = \tau_{n+1}^*$ [16]. By a dynamic programming principle, τ_n^* can be determined by the continuation values [29]. The value in state x at time n is $C^*(N, x) \equiv 0$ for $n = N$ and

$$C^*(n, x) = \mathrm{E}[\gamma^{\tau_{n+1}^* - n} g(X_{\tau_{n+1}^*})|X_n = x], \tag{2}$$

for $n = 0, \ldots, N-1$. Then, τ_n^* can be written as

$$\tau_n^* = \inf\{i \geq n : g(X_i) \geq C^*(i, X_i)\}. \tag{3}$$

In other words, the option should be exercised when the current payoff is larger than the continuation value. A meaningful family of suboptimal stopping times should be obtained by replacing the continuation values by a good approximation.

Motivated by the fitted policy iteration method in reinforcement learning [23], we considered iteratively approximating the family of optimal stopping times. In this paper, we dealt with consistent families of stopping times $\tau_n, n = 1, \ldots, N$. These times satisfied $n \leq \tau_n \leq N$ with $\tau_N = N$ and $\tau_n > n \Longrightarrow \tau_n = \tau_{n+1}$. We define the function $C^\tau : \{0, 1, \ldots, N-1\} \times \mathbb{R}^d \to \mathbb{R}$ by

$$C^\tau(n, x) = \mathrm{E}[\gamma^{\tau_{n+1} - n} g(X_{\tau_{n+1}})|X_n = x]. \tag{4}$$

This function represents the expected discounted payoff achieved when $X_n = x$, and the option is not exercised at n after which the stopping time τ_{n+1} is followed. Conversely, given $C : \{0, 1, \ldots, N-1\} \times \mathbb{R}^d \to \mathbb{R}$, we defined a new family of stopping times by

$$\begin{aligned} \tau_N^{'} &= N, \\ \tau_n^{'} &= \begin{cases} n, & \text{if } g(X_n) \geq C(n, X_n), \\ \tau_{n+1}^{'}, & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

It was immediately seen that the obtained family of stopping times $\tau_n^{'}$, $1 \leq n \leq N$ was consistent. The following result shows that, the exercise policy $\tau_1^{'}$ constructed from C^τ yielded a higher than expected discounted payoff than the original policy τ_1.

Theorem 1. *For any family of consistent stopping times $\tau_n, n = 1, \ldots, N$, the stopping time $\tau_1^{'}$ constructed from C^τ by (5) satisfies*

$$\mathrm{E}\gamma^{\tau_1^{'}} g(X_{\tau_1^{'}}) \geq \mathrm{E}\gamma^{\tau_1} g(X_{\tau_1}). \tag{6}$$

By Theorem 1, if we can approximate $C^\tau(n, X_n)$ for a family of stopping times τ_n, $1 \leq n \leq N-1$, we can construct an improved family of stopping times closer to the optimal family.

3. Iteration Algorithm

In this section, we propose a two-step iteration algorithm for American options. In the evaluation step, $C^{\tau}(n,x)$ of a family of stopping times is estimated. In the improvement step, the estimated function is used to construct an improved family of stopping times by (5).

We first defined the approximation architecture used for estimating $C^{\tau}(n,x)$ in the algorithm. Contrary to the regression-based algorithms, we used a single function by taking the time as an argument in the computation throughout the paper, denoted by $\mathcal{F}:=\{f:\mathbb{R}^{d+1}\to\mathbb{R}\}$, the choosing set of real-valued functions. We also introduced the truncation operator for the approximation architecture. Let ψ_B denote the truncation operator with level B defined by

$$\psi_B f=\begin{cases} f, & \text{if } |f|\le B,\\ \operatorname{sign}(f)\cdot B, & \text{otherwise}. \end{cases} \tag{7}$$

For a set of functions $\mathcal{F}$, we set $\psi_B\mathcal{F}=\{\psi_B f: f\in\mathcal{F}\}$.

To obtain a good approximation of C^{τ}, it was a straightforward matter to consider

$$\frac{1}{N}\sum_{n=0}^{N-1}\mathrm{E}\big(C^{\tau}(n,X_n)-\mathrm{E}[\gamma^{\tau_{n+1}-n}g(X_{\tau_{n+1}})|X_n]\big)^2. \tag{8}$$

To obtain a practical procedure, we considered the sample-based approximation to (8) in the algorithm.

To approximate the optimal stopping times numerically, our method was initialized with arbitrary $C_0\in\mathcal{F}$ and corresponding stopping times τ_n^1 constructed from (5), $n=1,\dots,N$. For $j=1,\dots,J-1$, we generated a set of Monte Carlo paths $(x_0^i,\dots,x_N^i)$, $i=1,\dots,M$ of process X_n, this sample set was independent of all previously generated paths. If at time n, stopping time τ_n^j was applied, the discounted payoff along the i-th simulated path was denoted by $\gamma^{\tau_n^{j,i}-n}g(x^i_{\tau_n^{j,i}})$. To obtain the approximation of C^{τ^j}, we considered minimizing the empirical counterpart of (8). Let $\hat{f}_j\in\mathcal{F}$ satisfy

$$\hat{f}_j=\arg\min_{f\in\mathcal{F}}\frac{1}{NM}\sum_{i=1}^{M}\sum_{n=0}^{N-1}\left(f\left(n,x_n^i\right)-\gamma^{\tau_{n+1}^{j,i}-n}g(x^i_{\tau_{n+1}^{j,i}})\right)^2, \tag{9}$$

we used the truncation $\hat{C}^{\tau^j}=\psi_B\hat{f}_j$ as the approximation. In the next iteration, an improved family of stopping times τ_n^{j+1}, $n=1,\dots,N$ was obtained by (5). Starting from any family of consistent stopping times and computing inductively, we finally constructed the exercise policy τ_1^J. Note that the optimization problem (9) is easily solved for some linear function space such as polynomial basis functions. For other approximation architecture such as neural networks, gradient-based methods can be applied to find the infimum, since (9) is differentiable with respect to f.

To estimate V_0, we generated another independent Monte Carlo sample path $(x_0^i,\dots,x_N^i)$, $i=1,\dots,M'$ and approximate V_0 by the average

$$\widehat{V}_0=\sum_{i=1}^{M'}\gamma^{\tau_1^{J,i}}g(x^i_{\tau_1^{J,i}}). \tag{10}$$

Our method is summarized into Algorithm 1. In next section, we discuss the convergence of the algorithm and derive a finite sample bound.

Algorithm 1 Iteration algorithm for pricing American options.

Require: the number of sample path M, $M^{'}$, the number of iterations J and function space $\mathcal{F}$

Ensure: the approximating optimal stopping time τ_1^J, the price estimate $\widehat{V}_0$

1: Generate sample paths of the underlying process;
2: Generate a random function $C_0 \in \mathcal{F}$;
3: **for** $j = 1, \ldots, J-1$ **do**
4: Obtain τ_n^j, $n = 1, \ldots, N$ using $\hat{C}^{\tau^{j-1}}$ from (5);
5: Construct $\hat{f}_j$ by the regression optimization problem

$$\hat{f}_j = \arg\min_{f \in \mathcal{F}} \frac{1}{NM} \sum_{i=1}^{M} \sum_{n=0}^{N-1} \left(f\left(n, x_n^i\right) - \gamma^{\tau_{n+1}^{j,i} - n} g(x^i_{\tau_{n+1}^{j,i}}) \right)^2;$$

6: Obtain the approximation by $\hat{C}^{\tau^j} = \psi_B \hat{f}_j$;
7: **end for**
8: Obtain τ_1^J using $\hat{C}^{\tau^{J-1}}$ from (5);
9: Generate another independent sample path of the underlying process;
10: Calculate the option price by (10);
11: **return** τ_1^J and $\widehat{V}_0$;

4. Convergence Analysis

In this section, we consider the convergence of the algorithm introduced in Section 3. Before describing the main result, we present some necessary definitions. To measure the complexity of a functional class, we introduced the definition of covering numbers. For a class of functions $\mathcal{F}$ and points $\mathbf{z}_1^M := (z_1, \ldots, z_M)$, the covering number $\mathcal{N}_1(\epsilon, \mathcal{F}, \mathbf{z}_1^M)$ is the minimal number $Q \in \mathbb{N}$ such that there exist functions $f_1, \ldots, f_Q$ with the property that for every $f \in \mathcal{F}$ there is a $q \in \{1, \ldots, Q\}$ such that

$$\frac{1}{M} \sum_{i=1}^{M} |f(z_i) - f_q(z_i)| < \epsilon. \tag{11}$$

For $f : \mathbb{R}^{d+1} \to \mathbb{R}$, we introduce $\|\cdot\|$ by

$$\|f\|^2 = \frac{1}{N} \sum_{n=0}^{N-1} \|f(n, X_n)\|_2^2. \tag{12}$$

Denoted by $\tau(f)$, the family of stopping times was obtained from (5) with respect to f. Let E_M stands for the expectation conditioned by the samples used to approximate the function C^τ. We now state our main result about the convergence of our algorithm.

Theorem 2. *Assume that $B < \infty$. Fix the set of admissible functions $\mathcal{F}$ and positive integer M. For $j = 1, \ldots, J$, define τ^j by (5) and define $\hat{C}^{\tau^j}$ by (9). Then*

$$\begin{aligned} &\mathrm{E}_M \left\| C^*(n, X_n) - C^{\tau^J}(n, X_n) \right\| \\ &\quad \leq \frac{c_1 \log M \sup_{0 \leq n \leq N-1} \left(\sup_{\mathbf{x}_1^M \in (\{n\} \times \mathbb{R}^d)^M} \log^{1/2} \left(\mathcal{N}_1 \left(\frac{1}{MB}, \psi_B \mathcal{F}, \mathbf{x}_1^M \right) \right) \right)}{M^{1/2}} \\ &\qquad + c_2 \sup_{f' \in \mathcal{F}} \inf_{f \in \mathcal{F}} \left\| f - C^{\tau(f')} \right\| + c_3 \gamma^{J/2} B, \end{aligned} \tag{13}$$

where $c_1, c_2, c_3 > 0$.

There are three terms in the bound (13). The first is the estimation error caused by the sampling step in the approximation. The second is the approximation error of $\mathcal{F}$ with respect to the $C^{\tau^j}(n,x)$ appearing in the iteration. The third comes from the error remaining after running the iteration algorithm for J iterations. This term decays at a geometric rate.

Remark 1. $\log \mathcal{N}_1\left(\epsilon, \psi_B\mathcal{F}, \mathbf{x}_1^M\right)$ *is bounded by* $\log M \cdot \nu_{\psi_B\mathcal{F}+}$ *under some mild conditions,* $\nu_{\psi_B\mathcal{F}+}$ *is the VC-dimension of* $\psi_B\mathcal{F}+$ *(see the definition in Kohler and Langer [30]). Theorems 1 immediately apply for linear, finite-dimensional approximation architecture, since the corresponding VC-dimension is bounded [31]. For neural networks with L hidden layers,* λ *neurons per layer and ReLU activation function, a bound of* $c_4\lambda L \log \lambda$ *with* $c_4 > 0$ *on the corresponding VC-dimension is also known [30]. Hence, the Theorem applies for deep neural networks as well.*

The next corollary provides a bound on the difference between V_0 and $\mathrm{E}\widehat{V}_0$.

Corollary 1. *Assume that* $B < \infty$ *and* $X_0 = x_0$ *a.s. for some* $x_0 \in \mathbb{R}$. *Fix the set of admissible functions* $\mathcal{F}$ *and positive integer M. Define* τ_1^J *by (5) with respect to* $\hat{C}^{\tau^{J-1}}$ *and define* $\bar{V}_0 := \mathrm{E}\gamma^{\tau_1^J} g(X_{\tau_1^J})$. *Then*

$$\begin{aligned}
&\mathrm{E}_M|V_0 - \bar{V}_0| \\
&\quad \leq \frac{c_5 \log M \sup_{0\leq n\leq N-1}\left(\sup_{\mathbf{x}_1^M\in(\{n\}\times\mathbb{R}^d)^M} \log^{1/2}\left(\mathcal{N}_1\left(\frac{1}{MB}, \psi_B\mathcal{F}, \mathbf{x}_1^M\right)\right)\right)}{M^{1/2}} \\
&\qquad + c_6 \sup_{f'\in\mathcal{F}} \inf_{f\in\mathcal{F}} \left\| f - C^{\tau(f')} \right\| + c_7\gamma^{J/2}B,
\end{aligned} \tag{14}$$

where $c_5, c_6, c_7 > 0$ *and* $0 < \gamma < 1$.

5. Numerical Examples

In this section, the performances of the algorithm were tested on various American options for bounded and unbounded payoffs.

The computations were carried out on a laptop with an Intel i5-10300H 2.50 GHz CPU and a NVIDIA GeForce GTX 1650 GPU.

To evaluate our method, we considered two function spaces: the linear spaces of polynomials and neural networks. Polynomial basis functions have been used in Longstaff and Schwartz [7] and are a popular basis function for regression-based methods. To include interaction terms in the basis, we considered the classical polynomial basis functions up to the third order. Neural network approximates nonlinear functions by successive compositions of an affine transformation and non-linear activation function. This model showed good performance for pricing American options, especially in high dimensions [32].

We compared our method with two state-of-the-art methods: the least squares Monte Carlo (LSM) proposed in Longstaff and Schwartz [7] and deep optimal stopping (DOS) proposed in Becker et al. [26]. To have a fair comparison for accuracy and efficiency, we used the same number of sample paths and time steps for both methods. Furthermore, we used the same network architecture in DOS and in the method with the neural network except the activation function. There were 3 hidden layers and $40 + d$ neurons per hidden layer in the networks. The activation function in our method was a leaky ReLU function.

5.1. Multi-Dimensional Black–Scholes Model

In this subsection, we consider high-dimensional American options in the Black–Scholes model. Assume the risk-neutral dynamics of the assets prices $S_t = (S_t^1, \ldots, S_t^d)$ are given by

$$S_t = S_0 \exp\left(\left[r - \delta - \frac{1}{2}\sigma^2\right]t + \sigma W_t\right), \tag{15}$$

where S_0 is the initial value; r is the risk-free interest rate; δ is the dividend rate; and W_t is d-dimensional Brownian motions with covariance matrix ρ. The parameters are set as $T = 1$ and $N = 10$, and we used $r = 5\%$, $\delta = 0$ for each asset. We assumed that ρ was a diagonal matrix and all the assets had the same volatility $\sigma = 0.2$ and initial value $S_0 = 100$.

We considered three types of high-dimensional options: max call options with payoff $\left(\max_{1\le i\le d} S_t^i - K\right)^+$, arithmetic put options with payoff $(K - \frac{1}{d}\sum_{i=1}^d S_t^i)^+$ and geometric put options with payoff $(K - (\prod_{i=1}^d S_t^i)^{1/d})^+$. We considered $d = \{5, 10, 20, 30, 40, 60, 80, 100\}$, and set $K = 100$. In the computation we set M = 20,000; $M^{'}$ = 100,000; $J = 5$; and we used the payoff as a regressor. For American options with unbounded payoff, we omitted the truncation step.

Tables 1–3 report the pricing results. Column Ref. provides the benchmark value computed by Premia (https://www.rocq.inria.fr/mathfi/Premia, accessed date is 29 September 2021), a freely available software for derivative pricing and hedging. Column Time is the computational times in seconds. For large d the computation time of LSM and our method with a polynomial basis function exceeded a reasonable amount of time, so the results were omitted. The columns in tables labeled as LSM-2, LSM-3, LFPI and NNFPI correspond to, respectively, LSM with a second-order polynomial basis function, LSM with a third-order polynomial basis function, our method with a second-order polynomial basis function, and our method with a neural network. Because $\widehat{V}_0$ is a lower-biased price estimate, higher price estimates implied better experimental performance.

We observed that both LFPI and NNFPI provided accurate results. LFPI was relatively faster in low-dimensional cases, but the computation time of NNFPI increased little with d. For the linear approximation architecture, the results showed that our method outperformed the LSM with respect to the required polynomial degree in pricing. This phenomenon is crucial when the number of underlying assets is large. For the computation time, although LFPI is slower than the LSM algorithm with same polynomial degree, LSM needed larger polynomial degrees for accurate results so that our method returned accurate results faster than the LSM in high-dimensional examples. In the case of nonlinear architecture, NNFPI generally outperformed DOS, which has similar network architecture.

Table 1. Pricing results for d-dimensional max call options.

d	Ref.	LSM-2	Time	LSM-3	Time	LFPI	Time	DOS	Time	NNFPI	Time
5	29.63	29.635	2.3	29.722	2.0	**29.732**	2.6	27.483	12.8	29.683	19.1
10	38.96	39.062	5.9	39.131	9.2	**39.219**	7.1	35.772	11.9	39.097	18.9
20	47.84	47.279	14.4	47.912	91.6	**48.000**	23.2	45.839	12.8	47.968	22.8
30	52.91	50.082	28.7	52.786	667.2	52.965	53.8	51.301	13.1	**52.991**	23.6
40	56.37	56.175	90.6			56.347	138.7	56.132	14.4	**56.496**	24.3
60	61.24							60.416	16.7	**61.362**	32.0
80	64.72							63.457	19.3	**64.704**	35.7
100	67.28							66.333	20.4	**67.323**	40.4

Table 2. Pricing results for d-dimensional arithmetic put options.

d	Ref.	LSM-2	Time	LSM-3	Time	LFPI	Time	DOS	Time	NNFPI	Time
5	2.05	2.045	1.4	2.046	1.2	2.047	2.6	2.046	12.6	**2.048**	19.0
10	1.39	1.376	3.3	**1.378**	4.7	**1.378**	7.0	**1.378**	11.7	**1.378**	18.7
20	1.06	1.042	6.7	1.045	35.7	1.046	25.5	**1.047**	12.5	**1.047**	22.7
30	0.64	0.626	11.7	0.628	215.0	**0.630**	55.1	0.629	13.2	**0.630**	23.5
40	0.66	0.645	26.6			**0.646**	141.0	**0.646**	14.1	**0.646**	24.3
60	0.89							0.864	16.8	**0.865**	31.9
80	0.74							0.712	19.2	**0.713**	35.6
100	0.32							0.311	20.5	**0.312**	40.2

Table 3. Pricing results for d-dimensional geometric put options.

d	Ref.	LSM-2	Time	LSM-3	Time	LFPI	Time	DOS	Time	NNFPI	Time
5	2.05	2.033	1.3	2.036	1.5	**2.038**	2.6	2.036	12.6	**2.038**	19.1
10	1.39	1.368	3.8	1.376	5.1	**1.379**	6.9	1.373	11.9	**1.379**	18.8
20	1.06	1.027	7.7	1.043	40.7	1.049	25.2	1.051	12.6	**1.052**	22.8
30	0.64	0.604	13.7	0.616	271.5	**0.625**	54.9	**0.625**	13.2	**0.625**	23.7
40	0.66	0.645	35.1			0.654	135.8	0.646	14.3	**0.655**	24.4
60	0.89							0.869	16.6	**0.887**	32.0
80	0.74							0.719	19.3	**0.722**	35.6
100	0.32							0.308	20.8	**0.311**	40.3

5.2. Stochastic Volatility Model

This subsection is devoted to the American options on the Heston model [33], a well-known symmetric stochastic volatility model in options pricing. The evolution of underlying asset S_t and instantaneous variance ν_t is described by the following stochastic differential equation

$$\begin{aligned} \mathrm{d}S_t &= rS_t\mathrm{d}t + \sqrt{\nu_t}S_t\mathrm{d}W_t^1, \\ \mathrm{d}\nu_t &= \kappa(\theta - \nu_t)\mathrm{d}t + \xi\sqrt{\nu_t}\mathrm{d}W_t^2, \end{aligned} \tag{16}$$

where $r \geq 0, \kappa > 0, \theta > 0, \xi > 0$, W_t^1 and W_t^2 are d-dimensional Brownian motions. For a reliable price reference for high-dimensional American options, we used the same parameter settings as in the cases studied in Herrera et al. [28]. The max call options and geometric put options were considered in the experiments. Specifically, we choose the parameters $T = 1$; $N = 10$; $\kappa = 2$; $\theta = 0.01$; $\xi = 0.2$; the initial stock price $S_0 = 100$; the initial variance $\nu_0 = 0.01$; $r = 0\%$ for max call options; and $r = 2\%$ for geometric put options. We assumed that the dynamics of different assets were independent, the correlation between the Brownian motion driving the price process and variance process of single asset was $\rho = -0.3$. In the computation we used M = 20,000; $M^{'}$ = 100,000; and $J = 5$ and same network architecture as in last subsection.

To obtain a Markovian model, we included price and variance as inputs. In practical, stochastic volatility models, they need to be calibrated from observed data; then, our algorithm can be applied to sample data generated from the models. In the experiments, we tested our method under max call options and geometric put options with $K = 100$. The results are reported in Tables 4 and 5. The results were similar to those under the multi-dimensional Black–Scholes model. The pricing results obtained from LFPI and NNFPI were close to the reference values. The LFPI computation time was smaller than that of NNFPI for low-dimensional situations, but the NNFPI was more efficient for $d \geq 10$. It could be seen that NNFPI is generally the most accurate method for high-dimensional American options, and the computation time of NNFPI is close to that of DOS, especially for large d. For the linear approximation architecture, the results showed that LFPI outperformed LSM. Note that our method had much fewer trainable parameters.

Table 4. Pricing results for d-dimensional max-call options in the Heston model.

d	Ref.	LSM-2	Time	LSM-3	Time	LFPI	Time	DOS	Time	NNFPI	Time
5	8.33	8.252	5.7	8.258	6.3	**8.262**	6.1	8.192	12.8	8.207	19.8
10	11.83	11.460	27.4	11.662	76.6	11.623	23.1	11.286	13.5	**11.796**	21.0
20		14.992	89.0			15.058	132.0	14.885	15.6	**15.330**	26.5
30								17.091	18.4	**17.465**	27.5
40								18.485	21.5	**18.974**	31.7
50	20.09							19.563	23.7	**20.100**	35.0
80								21.993	34.3	**22.487**	43.5
100	23.69							22.927	40.3	**23.613**	50.1

Table 5. Pricing results for d-dimensional geometric put options in the Heston model.

d	Ref.	LSM-2	Time	LSM-3	Time	LFPI	Time	DOS	Time	NNFPI	Time
5	2.43	2.366	4.9	2.368	5.0	2.420	6.0	2.309	12.8	**2.443**	19.7
10	2.01	1.773	21.1	1.959	52.6	1.986	22.5	1.931	13.6	**2.015**	21.0
20	1.71	1.631	67.5			1.658	134.0	1.643	15.4	**1.712**	26.5
30								1.292	18.1	**1.593**	27.5
40								1.304	20.8	**1.522**	31.6
50	1.48							1.291	22.5	**1.474**	34.9
80								1.194	33.3	**1.426**	43.3
100	1.40							1.141	39.9	**1.402**	49.8

5.3. Convergence with Respect to the Number of Iteration

In this subsection, we study numerically the convergence of our method with respect to the hyperparameter J. We considered max-call options under the Black–Scholes model with same parameters setting as in Section 5.1. Figure 1 presents the results for LFPI with $d = 5$ and NNFPI with $d = 20$. The errors were computed with respect to the final value. It could be seen that our method converged fast with respect to the number of iterations. The results confirmed that limiting the number of iterations below to 5 was reasonable.

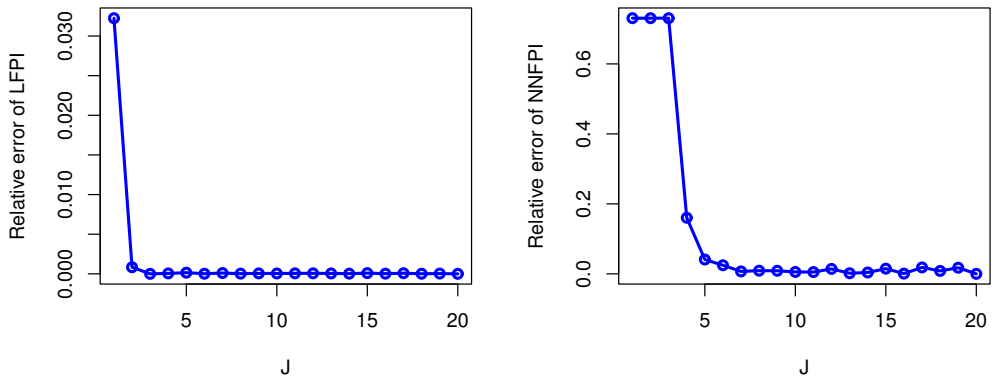

Figure 1. Convergence with respect to the number of iteration.

6. Conclusions

We introduced a novel method for American options based on reinforcement learning that was accurate and efficient in high-dimensional situations. We provided a convergence analysis for the algorithms in the number of training samples and iterations. We also considered the applicability of the algorithm and carried out comprehensive numerical experiments in multivariate Black-Scholes and Heston models. The results showed that (a) the NNFPI achieved good performance in high-dimensional situations, and the LFPI outperformed the LSM in the pricing of American options; (b) the algorithm had high efficiency and accuracy under different model assumptions; (c) our algorithm had a fast convergence rate with respect to the number of iterations; (d) the continuation values can be approximated with a fraction of the parameters by using a function of time and the underlying process. To summarize, the results reconfirmed that reinforcement learning methods surpass backward induction methods for pricing of high-dimensional American options.

There are several directions for future research. First, upper price bounds and confidence interval could be constructed based on the approximating the optimal stopping time. Furthermore, it would be desirable to remove the condition that the payoff function is bounded in L^∞. One idea is to use the truncation technique in Zanger [34]. The proofs are technically more challenging and are left for future research.

Funding: This work was partially supported by the Fundamental Research Funds for the Central Universities, JLU (93K172020K26).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the editor and reviewers for their valuable suggestions and comments which greatly improved the article.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Proofs

Proof of Theorem 1. Let I_A be the indicator function of set A. Because $\tau_n^{'} = nI_{\{g(X_n)\geq C^{\tau}(n,X_n)\}} + \tau_{n+1}^{'} I_{\{g(X_n)<C^{\tau}(n,X_n)\}}$, we have

$$\begin{aligned}
&\mathrm{E}[\gamma^{\tau_n} g(X_{\tau_n})|x_n = x] \\
=&\gamma^n g(x) I_{\{\tau_n=n\}} + \mathrm{E}[\gamma^{\tau_{n+1}} g(X_{\tau_{n+1}})|X_n = x] I_{\{\tau_n>n\}} \\
\leq&\gamma^n g(x) I_{\{\tau_n'=n\}} + \mathrm{E}[\gamma^{\tau_{n+1}} g(X_{\tau_{n+1}})|X_n = x] I_{\{\tau_n'>n\}} \\
=&\gamma^n g(x) I_{\{\tau_n'=n\}} + \mathrm{E}[\gamma^{\tau_{n+1}} g(X_{\tau_{n+1}}) I_{\{\tau_n'>n\}}|X_n = x] \\
\leq&\gamma^n g(x) I_{\{\tau_n'=n\}} + \mathrm{E}\Big[\gamma^{n+1} g(X_{n+1}) I_{\{\tau_n'=n+1\}} \\
&\qquad + \mathrm{E}[\gamma^{\tau_{n+2}} g(X_{\tau_{n+2}}) I_{\{\tau_n'>n+1\}}|X_{n+1}]|X_n = x\Big] \\
=&\gamma^n g(x) I_{\{\tau'=n\}} + \mathrm{E}\Big[\gamma^{n+1} g(X_{n+1}) I_{\{\tau_n'=n+1\}} \\
&\qquad + \gamma^{\tau_{n+2}} g(X_{\tau_{n+2}}) I_{\{\tau_n'>n+1\}}|X_n = x\Big].
\end{aligned}$$

By induction, we have

$$\begin{aligned}
&\mathrm{E}\gamma^{\tau_1} g(X_{\tau_1}) \\
&\leq \mathrm{E}\left[\sum_{n=1}^{N-1} \gamma^n g(X_n) I_{\{\tau_n'=n\}} + \gamma^N g(X_N) I_{\{\tau_N>N-1\}}|X_0 = x_0\right] \\
&= \mathrm{E}\left[\sum_{n=1}^{N-1} \gamma^n g(X_n) I_{\{\tau_n'=n\}} + \gamma^N g(X_N) I_{\{\tau_N'=N\}}|X_0 = x_0\right] \\
&= \mathrm{E}\gamma^{\tau_1'} g(X_{\tau_1'}).
\end{aligned}$$

The last step follows from (5). □

Our aim is to derive a bound of C^{τ^J} and C^*. To this end, we define the operator T^{τ} by

$$(T^{\tau}C)(n,x) = \gamma \mathrm{E}[g(X_{n+1}) I_{\{\tau_{n+1}=n+1\}} + C(n+1, X_{n+1}) I_{\{\tau_{n+1}>n+1\}}|X_n = x].$$

It is easy to see that T^{τ} is a contraction operator with index γ, and hence has a unique fixed point $C^{\tau}(n,x)$,

$$T^{\tau}C^{\tau} = C^{\tau}.$$

For $j = 1, \ldots, J$, we define $\varepsilon_j = \hat{C}^{\tau^j} - T^{\tau^j}\hat{C}^{\tau^j}$.

Lemma A1. *Let J be a positive integer. Then, for the sequence of functions $\hat{C}^{\tau^j} \le B, 0 \le j < J$ and ε_j the following inequalities hold*

$$\left\|C^* - C^{\tau^J}\right\| \le c_8\left(\max_{1\le j<J}\|\varepsilon_j\| + \gamma^{J/2}B\right),$$

where $c_8 > 0$ and $0 < \gamma < 1$.

Proof. We interpret (n, X_n) as a random variable, where n is uniformly distributed on $0, \ldots, N-1$. We have

$$\begin{aligned}\left\|C^* - C^{\tau^J}\right\| &= \left(\frac{1}{N}\sum_{n=0}^{N-1}\mathrm{E}\left(C^*(n, X_n) - C^{\tau^J}(n, X_n)\right)^2\right)^{1/2}\\ &= \left\|C^*(n, X_n) - C^{\tau^J}(n, X_n)\right\|.\end{aligned}$$

By Lemma 12 in [23], the conclusions follows. □

Lemma A2. *Assume that $B < \infty$ and τ_n, $n = 1, \ldots, N$ is arbitrary family of consistent stopping times. $(x_0^i, \ldots, x_N^i)$, $i = 1, \ldots, M$ is a set of Monte Carlo paths of process X_n. Let $\hat{f}$ be defined by*

$$\hat{f} = \operatorname*{argmin}_{f\in\mathcal{F}} \frac{1}{NM}\sum_{i=1}^{M}\sum_{n=0}^{N-1}\left(f\left(n, x_n^i\right) - \gamma^{\tau_{n+1}^i - n} g\left(x^i_{\tau_{n+1}^i}\right)\right)^2,$$

and set $\hat{C}^\tau = \psi_B \hat{f}$. Then we have

$$\begin{aligned}&\mathrm{E}_M\left\|\hat{C}^\tau - T^\tau\hat{C}^\tau\right\|^2\\ &\le \frac{c_9(\log M)^2 \sup_{0\le n\le N-1}\left(\sup_{\mathbf{x}_1^M\in(\{n\}\times\mathbb{R}^d)^M}\log\left(\mathcal{N}_1\left(\frac{1}{MB}, \psi_B\mathcal{F}, \mathbf{x}_1^M\right)\right)\right)}{M}\\ &\quad + 2\inf_{f\in\mathcal{F}}\|f - C^\tau\|^2,\end{aligned}$$

where $c_9 > 0$.

Proof. For any $a, b \in \mathbb{R}$, we have $(a+b)^2 \le 2a^2 + 2b^2$. Thus, we have

$$\begin{aligned}&\left\|\hat{C}^\tau(n, X_n) - T^\tau\hat{C}^\tau(n, X_n)\right\|_2^2\\ &= \left\|\hat{C}^\tau(n, X_n) - \gamma\mathrm{E}[g(X_{n+1})I_{\{\tau_{n+1}=n+1\}} + \hat{C}^\tau(n+1, X_{n+1})I_{\{\tau_{n+1}>n+1\}}|X_n]\right\|_2^2\\ &= \Big\|\hat{C}^\tau(n, X_n) - \gamma\mathrm{E}[g(X_{n+1})I_{\{\tau_{n+1}=n+1\}} + C^\tau(n+1, X_{n+1})I_{\{\tau_{n+1}>n+1\}}|X_n]\\ &\qquad + \gamma\mathrm{E}[C^\tau(n+1, X_{n+1})I_{\{\tau_{n+1}>n+1\}} - \hat{C}^\tau(n+1, X_{n+1})I_{\{\tau_{n+1}>n+1\}}|X_n]\Big\|_2^2\\ &\le 2\left\|\hat{C}^\tau(n, X_n) - \gamma\mathrm{E}[g(X_{n+1})I_{\{\tau_{n+1}=n+1\}} + C^\tau(n+1, X_{n+1})I_{\{\tau_{n+1}>n+1\}}|X_n]\right\|_2^2\\ &\quad + 2\left\|\gamma\mathrm{E}[C^\tau(n+1, X_{n+1})I_{\{\tau_{n+1}>n+1\}} - \hat{C}^\tau(n+1, X_{n+1})I_{\{\tau_{n+1}>n+1\}}|X_n]\right\|_2^2\\ &\le 2\left\|\hat{C}^\tau(n, X_n) - C^\tau(n, X_n)\right\|_2^2 + 2\left\|C^\tau(n+1, X_{n+1}) - \hat{C}^\tau(n+1, X_{n+1})\right\|_2^2,\end{aligned}$$

the last inequality follows from Jensen's inequality. Thus by induction, we have

$$\left\|\hat{C}^\tau - T^\tau\hat{C}^\tau\right\|^2 = \frac{1}{N}\sum_{n=0}^{N-1}\left\|\hat{C}^\tau(n, X_n) - T^\tau\hat{C}^\tau(n, X_n)\right\|_2^2$$

$$\begin{aligned}
&\leq \frac{4}{N}\sum_{n=0}^{N-1}\left\|\hat{C}^{\tau}(n,X_n)-C^{\tau}(n,X_n)\right\|_2^2\\
&=4\left\|\hat{C}^{\tau}-C^{\tau}\right\|^2.
\end{aligned}$$

Since $C^{\tau}(n,x)=\mathrm{E}\left[\gamma^{\tau_{n+1}-n}g\left(X_{\tau_{n+1}}\right)|X_n=x\right]$, we have the following error decomposition

$$\begin{aligned}
&\left\|\hat{C}^{\tau}-C^{\tau}\right\|^2\\
&=\frac{1}{N}\sum_{n=0}^{N-1}\Bigg[\mathrm{E}\left|\hat{C}^{\tau}(n,X_n)-\gamma^{\tau_{n+1}-n}g\left(X_{\tau_{n+1}}\right)\right|^2\\
&\qquad-\mathrm{E}\left|C^{\tau}(n,X_n)-\gamma^{\tau_{n+1}-n}g\left(X_{\tau_{n+1}}\right)\right|^2\\
&\qquad-\frac{2}{M}\sum_{i=1}^{M}\left(\left|\hat{C}^{\tau}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\right.\\
&\qquad\left.-\left|C^{\tau}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\right)\Bigg]\\
&+\frac{2}{NM}\sum_{n=0}^{N-1}\sum_{i=1}^{M}\Bigg[\left|\hat{C}^{\tau}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\\
&\qquad-\left|C^{\tau}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\Bigg].
\end{aligned}\tag{A1}$$

Using Lemma 1 in [35], the first term in (A1) is bounded by

$$\frac{c_{10}(\log M)^2\sup_{0\leq n\leq N-1}\left(\sup_{\mathbf{x}_1^M\in(\{n\}\times\mathbb{R}^d)^M}\log\left(\mathcal{N}_1\left(\frac{1}{MB},\psi_B\mathcal{F},\mathbf{x}_1^M\right)\right)\right)}{M},\tag{A2}$$

for some $c_{10}>0$. Because $|\psi_B a-b|\leq|a-b|$ holds for $|b|\leq B$, the second term in (A1) is bounded by

$$\inf_{f\in\mathcal{F}}\frac{2}{NM}\sum_{n=0}^{N-1}\sum_{i=1}^{M}\left(\left|f(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2-\left|C^{\tau}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\right).$$

If we choose an $\tilde{f}\in\mathcal{F}$ such that

$$\left\|\tilde{f}-C^{\tau}\right\|^2\leq\inf_{f\in\mathcal{F}}\|f-C^{\tau}\|^2+\frac{1}{M},$$

we can conclude

$$\begin{aligned}
&\mathrm{E}_M\left[\inf_{f\in\mathcal{F}}\frac{1}{NM}\sum_{n=0}^{N-1}\sum_{i=1}^{M}\left|f(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\right]\\
&\quad-\mathrm{E}_M\left[\frac{1}{NM}\sum_{n=0}^{N-1}\sum_{i=1}^{M}\left|C^{\tau}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\right]\\
&\leq\mathrm{E}_M\left[\frac{1}{NM}\sum_{n=0}^{N-1}\sum_{i=1}^{M}\left|\tilde{f}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\right]\\
&\quad-\mathrm{E}_M\left[\frac{1}{NM}\sum_{n=0}^{N-1}\sum_{i=1}^{M}\left|C^{\tau}(n,x_n^i)-\gamma^{\tau_{n+1}^i-n}g\left(x^i_{\tau^i_{n+1}}\right)\right|^2\right]\\
&=\mathrm{E}_M\left[\frac{1}{N}\sum_{n=0}^{N-1}\left|\tilde{f}(n,X_n)-\gamma^{\tau_{n+1}-n}g\left(X_{\tau_{n+1}}\right)\right|^2\right]
\end{aligned}$$

$$
\begin{aligned}
&- \mathrm{E}_M\left[\frac{1}{N}\sum_{n=0}^{N-1}\left|C^{\tau}(n,X_n)-\gamma^{\tau_{n+1}-n}g\left(X_{\tau_{n+1}}\right)\right|^2\right]\\
=&\left\|\tilde{f}-C^{\tau}\right\|^2+\mathrm{E}_M\left[\frac{1}{N}\sum_{n=0}^{N-1}\left|C^{\tau}(n,X_n)-\gamma^{\tau_{n+1}-n}g\left(X_{\tau_{n+1}}\right)\right|^2\right]\\
&- \mathrm{E}_M\left[\frac{1}{N}\sum_{n=0}^{N-1}\left|C^{\tau}(n,X_n)-\gamma^{\tau_{n+1}-n}g\left(X_{\tau_{n+1}}\right)\right|^2\right]\\
\leq&\inf_{f\in\mathcal{F}}\|f-C^{\tau}\|^2+\frac{1}{M}. \qquad \text{(A3)}
\end{aligned}
$$

The conclusion follows from (A2) and (A3). □

Proof of Theorem 2. Fix $M, J > 0$. Lemma A1 gives

$$
\mathrm{E}_M\left\|C^*-C^{\tau^J}\right\| \leq c_{11}\left(\mathrm{E}_M\max_{0\leq j<J}\|\varepsilon_j\|+\gamma^{J/2}B\right), \qquad \text{(A4)}
$$

$c_{11} > 0$. For any $a, b > 0$, we have $\sqrt{a+b} \leq \sqrt{a}+\sqrt{b}$. By Jensen's inequality and Lemma A2, we conclude that for any fixed integer $0 \leq j < J$,

$$
\begin{aligned}
\mathrm{E}_M\|\varepsilon_j\| \leq&\left(\mathrm{E}_M\|\varepsilon_j\|^2\right)^{\frac{1}{2}}\\
\leq&\frac{c_{12}\log M\sup_{0\leq n\leq N-1}\left(\sup_{\mathbf{x}_1^M\in(\{n\}\times\mathbb{R}^d)^M}\log^{1/2}\left(\mathcal{N}_1\left(\frac{1}{MB},\psi_B\mathcal{F},\mathbf{x}_1^M\right)\right)\right)}{M^{1/2}}\\
&+c_{13}\inf_{f\in\mathcal{F}}\|f-C^{\tau}\|,
\end{aligned}
$$

for $c_{12} > 0$, $c_{13} > 0$. Combining this with (A4), we get

$$
\begin{aligned}
&\mathrm{E}_M\left\|C^*(n,X_n)-C^{\tau^J}(n,X_n)\right\|\\
\leq&\frac{c_1\log M\sup_{0\leq n\leq N-1}\left(\sup_{\mathbf{x}_1^M\in(\{n\}\times\mathbb{R}^d)^M}\log^{1/2}\left(\mathcal{N}_1\left(\frac{1}{MB},\psi_B\mathcal{F},\mathbf{x}_1^M\right)\right)\right)}{M^{1/2}}\\
&+c_2\sup_{f'\in\mathcal{F}}\inf_{f\in\mathcal{F}}\left\|f-C^{\tau(f')}\right\|+c_3\gamma^{J/2}B.
\end{aligned}
$$

□

Proof of Corollary 1. By dynamic programming principle, we have

$$
V_0 = C^*(0,x_0).
$$

By the definition, we have

$$
\bar{V}_0 = C^{\tau^J}(0,x_0).
$$

We have the following error bound,

$$\begin{aligned}
&\mathrm{E}_M|V_0 - \bar{V}_0| \\
&\quad = \mathrm{E}_M\left|C^*(0,x_0) - C^{\tau^J}(0,x_0)\right| \\
&\quad \le c_{14}\mathrm{E}_M\left\|C^* - C^{\tau^J}\right\| \\
&\quad \le \frac{c_5 \log M \sup_{0\le n\le N-1}\left(\sup_{\mathbf{x}_1^M\in(\{n\}\times\mathbb{R}^d)^M}\log^{1/2}\left(\mathcal{N}_1\left(\frac{1}{MB},\psi_B\mathcal{F},\mathbf{x}_1^M\right)\right)\right)}{M^{1/2}} \\
&\qquad + c_6 \sup_{f'\in\mathcal{F}}\inf_{f\in\mathcal{F}}\left\|f - C^{\tau(f')}\right\| + c_7\gamma^{J/2}B,
\end{aligned}$$

where $c_{14} > 0$ and $0 < \gamma < 1$. □

References

1. Andriyanov, N. Forming a taxi service order price using neural networks with multi-parameter training. *J. Phys. Conf. Ser.* **2020**, *1661*, 012165 . [CrossRef]
2. Ullrich, T. On the Autoregressive Time Series Model Using Real and Complex Analysis. *Forecasting* **2021**, *3*, 44. [CrossRef]
3. Hull, J.C. *Options, Futures, and Other Derivatives*; Pearson Education: New York, NY, USA, 2018.
4. Achdou, Y.; Pironneau, O. *Computational Methods for Option Pricing*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2005.
5. Cox, J.C.; Ross, S.A.; Rubinstein, M. Option pricing: A simplified approach. *J. Financ. Econ.* **1979**, *7*, 229–263. [CrossRef]
6. Casas, I.; Veiga, H. Exploring option pricing and hedging via volatility asymmetry. *Comput. Econ.* **2021**, *57*, 1015–1039. [CrossRef]
7. Longstaff, F.A.; Schwartz, E.S. Valuing American Options by Simulation: A Simple Least-Squares Approach. *Rev. Financ. Stud.* **2001**, *14*, 113–147. [CrossRef]
8. Tsitsiklis, J.; Roy, B.V. Regression methods for pricing complex American-style options. *IEEE Trans. Neural Netw.* **2001**, *12*, 694–703. [CrossRef] [PubMed]
9. Kohler, M.; Krzyżak, A.; Todorovic, N. Pricing of High-Dimensional American Options by Neural Networks. *Math. Financ.* **2010**, *20*, 383–410. [CrossRef]
10. Goudenege, L.; Molent, A.; Zanette, A. Machine learning for pricing American options in high-dimensional Markovian and non-Markovian models. *Quant. Financ.* **2020**, *20*, 573–591. [CrossRef]
11. Hu, W.; Zastawniak, T. Pricing high-dimensional American options by kernel ridge regression. *Quant. Financ.* **2020**, *20*, 851–865. [CrossRef]
12. Rogers, L.C. Monte Carlo valuation of American options. *Math. Financ.* **2002**, *12*, 271–286. [CrossRef]
13. Haugh, M.B.; Kogan, L. Pricing American options: A duality approach. *Oper. Res.* **2004**, *52*, 258–270. [CrossRef]
14. Andersen, L. A simple approach to the pricing of Bermudan swaptions in the multifactor LIBOR market model. *J. Comput. Financ.* **2000**, *3*, 5–32. [CrossRef]
15. Belomestny, D. On the rates of convergence of simulation-based optimization algorithms for optimal stopping problems. *Ann. Appl. Probab.* **2011**, *21*, 215–239. [CrossRef]
16. Bayer, C.; Belomestny, D.; Hager, P.; Pigato, P.; Schoenmakers, J. Randomized optimal stopping algorithms and their convergence analysis. *SIAM J. Financ. Math.* **2021**, *12*, 1201–1225. [CrossRef]
17. Becker, S.; Cheridito, P.; Jentzen, A.; Welti, T. Solving high-dimensional optimal stopping problems using deep learning. *Eur. J. Appl. Math.* **2021**, *32*, 470–514. [CrossRef]
18. Garcıa, D. Convergence and biases of Monte Carlo estimates of American option prices using a parametric exercise rule. *J. Econ. Dyn. Control* **2003**, *27*, 1855–1879. [CrossRef]
19. Bayer, C.; Tempone, R.; Wolfers, S. Pricing American options by exercise rate optimization. *Quant. Financ.* **2020**, *20*, 1749–1760. [CrossRef]
20. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, UK, 2018.
21. Wang, R.; Salakhutdinov, R.R.; Yang, L. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6123–6135.
22. Wang, R.; Du, S.S.; Yang, L.; Salakhutdinov, R.R. On reward-free reinforcement learning with linear function approximation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17816–17826.
23. Antos, A.; Szepesvári, C.; Munos, R. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Mach. Learn.* **2008**, *71*, 89–129. [CrossRef]
24. Yu, H.; Bertsekas, D.P. Q-learning algorithms for optimal stopping based on least squares. In Proceedings of the European Control Conference, Kos, Greece, 2–5 July 2007.

25. Li, Y.; Szepesvari, C.; Schuurmans, D. Learning policies for American options. In Proceedings of the Conference on Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, 16–18 April 2009.
26. Becker, S.; Cheridito, P.; Jentzen, A. Deep optimal stopping. *J. Mach. Learn. Res.* **2019**, *20*, 74.
27. Chen, S.; Devraj, A.M.; Bušić, A.; Meyn, S. Zap Q-Learning for optimal stopping. In Proceedings of the American Control Conference, Denver, CO, USA, 1–3 July 2020.
28. Herrera, C.; Krach, F.; Ruyssen, P.; Teichmann, J. Optimal Stopping via Randomized Neural Networks. *arXiv* **2021**, arXiv:2104.13669.
29. Glasserman, P. *Monte Carlo Methods in Financial Engineering*; Springer Science & Business Media: New York, NY, USA, 2003; Volume 53.
30. Kohler, M.; Langer, S. On the rate of convergence of fully connected very deep neural network regression estimates. *arXiv* **2019**, arXiv:1908.11133.
31. Zanger, D.Z. Quantitative error estimates for a least-squares Monte Carlo algorithm for American option pricing. *Financ. Stoch.* **2013**, *17*, 503–534. [CrossRef]
32. Beck, C.; Weinan, E.; Jentzen, A. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *J. Nonlinear Sci.* **2019**, *29*, 1563–1619. [CrossRef]
33. Heston, S.L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Stud.* **1993**, *6*, 327–343. [CrossRef]
34. Zanger, D.Z. General error estimates for the Longstaff–Schwartz least-squares Monte Carlo algorithm. *Math. Oper. Res.* **2020**, *45*, 923–946. [CrossRef]
35. Bauer, B.; Kohler, M. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Ann. Stat.* **2019**, *47*, 2261–2285. [CrossRef]