ML THEORETICAL FOUNDATION: AN ANALYSIS OF 'DEEP OPTIMAL STOPPING'

Students
Plazzogna Marco
Zanirato Daniele
Amorotti Sean

Contents

1 Introduction

This paper deals with the optimal stopping problem, which involves determining a stopping rule for a stochastic process that maximizes a given utility function. This is a very general setting which can be applied to a wide variety of domains: from pricing American options, to deciding when to stop collecting data in a clinical trial or to manage investment timing in the market or in competitive games. More specifically, this paper focuses mostly on the financial setting via various applications. This paper merges classical results and algorithms in stopping time theory (like the Snell envelope and Monte-Carlo simulations) with deep learning, proposing a flexible method, able to approximate with high precision complex stopping rules in high-dimensional settings.

This paper was selected because it addresses a fundamental and widely applicable problem in stochastic control, with significant implications in finance, statistics, and machine learning. The optimal stopping problem lies at the intersection of probability theory and decision-making under uncertainty, two core areas of modern applied mathematics. Moreover, the integration of deep learning techniques with classical stopping time theory represents a promising and timely research direction, capable of tackling the increasing complexity of high-dimensional problems encountered in real-world applications.

2 Summary of the Paper

Outline

The paper is divided into four sections. In the *Introduction* section, where the authors outline the context and the stopping time problem, including the relevant literature; they also underline how a major contribution of this paper lies in the computational efficiency of this approach in high dimensional settings. The following section is *Deep Learning Optimal Stopping Rules*, where the authors recall the dynamic programming and reinforcement learning framework and propose the use of Neural Networks as a novel approach to directly estimate the optimal stopping policy. In particular, the authors define the mathematical setting of the problem and derive good mathematical properties of their approach. For instance, they prove that the stopping time derived with neural networks can approximate arbitrarily well the true optimal stopping time. In the third one, called *Bounds, Point Estimates and Confidence Intervals*, some statistical results are derived regarding our estimator of the optimal stopping time; in particular, point estimation, confidence interval, upper and lower bounds. Finally, in the *Example* section, the authors show that their method effectively handles three high-dimensional optimal stopping problems, as the 95% confidence intervals of the optimal stopping time are very tight even for high dimensions.

Deep Learning Optimal Stopping Rules

Firstly, they introduce the general framework. They are dealing with a d-dimensional Markov chain X defined on the usual probability space, equipped with the filtration generated by X, and they want to solve the optimization

$$V = \sup_{\tau \in \mathcal{T}} \mathbb{E}[g(\tau, X_\tau)],$$

which entails finding the stopping time that maximizes a given utility function.

The authors proceed by reducing the problem to the dynamic programming framework, recalling that any X-stopping time can be decomposed into a sequence of 0-1 stopping decisions: to optimally stop the Markov chain X, it is enough to chain the optimal 0-1 stopping decisions. This is formalized by considering the auxiliary stopping problems

$$V_n = \sup_{\tau_n \in \mathcal{T}_n} \mathbb{E}[g(\tau_n, X_{\tau_n})] \tag{1}$$

for $n = 0, 1, \ldots, N$, where $\mathcal{T}_n$ is the set of all X-stopping times satisfying $\tau_n \geq n$, which means that τ_n is the optimal stopping time given that they have not stopped at times $1, 2, \ldots, n-1$. It's relevant to notice the equivalence with the original optimization problem $V = V_0$. The idea is to estimate sequentially τ_N, τ_{N-1}, and so on until τ_0, which is the solution of our original optimization problem.

Theorem 1 has vast theoretical and practical implications, serving as a justification for the robustness of the dynamic programming framework of backward estimation developed in this paper. On the theoretical side, it proves that it is possible recover the optimal stopping time as a sequence of optimal binary decisions, therefore it is enough to consider stopping times of the form

$$\tau_n = \sum_{m=n}^{N} m f_m(X_m) \prod_{j=n}^{m-1} (1 - f_j(X_j)) \tag{2}$$

where $f_n(x)$ is the binary stopping decision function at time n. This is equivalent to say that they stop at time n if and only if $f_n(x) = 1$ and $f_1(x), ..., f_{n-1}(x) = 0$, which means deciding to stop at time n and move forward up until that point.

More importantly, in practice, the theorem guarantees that the approximation error they make while computing τ_{n+1}, propagates arithmetically and not geometrically to the subsequent estimation τ_n.

Theorem 1 For a given $n \in \{0, 1, \ldots, N-1\}$, let τ_{n+1} be a stopping time in $\mathcal{T}_{n+1}$ of the form (2), then there exists a measurable function $f_n : \mathbb{R}^d \to \{0, 1\}$ such that the stopping time $\tau_n \in \mathcal{T}_n$ given by (2) satisfies

$$\mathbb{E}[g(\tau_n, X_{\tau_n})] \geq V_n - \left(V_{n+1} - \mathbb{E}[g(\tau_{n+1}, X_{\tau_{n+1}})]\right),$$

where V_n and V_{n+1} are the optimal values defined in equation (1).

Assuming it's always possible to find the best stopping time step by step, then it's guaranteed to reach a global optimal stopping time recursively, because the initial condition of the recursion is that τ_N is (trivially) optimal. However, in practice, the authors distinguish between the optimal stopping time at time n, τ_n, and the stopping rule that they manage to learn with neural networks, which necessarily includes an approximation error. This theorem guarantees that if we make a given estimation error ϵ on $\tau_n + 1$, the error that propagates on our estimate of τ_n is less than or equal to ϵ. This is an important theoretical guarantee, suggesting that the approach they develop in the paper is numerically stable.

In paragraph 2.2, the authors go into detail about how to recover the binary decision functions $f_1, \ldots, f_N$. They introduce N neural networks called $F_1, \ldots, F_N$ taking values in the open interval $(0, 1)$, from which they then recover the binary decision based on whether their value is smaller or greater than $\frac{1}{2}$. In the spirit of dynamic programming, in order to train the N neural networks, they proceed backward, imposing $F_{\theta_N} = 1$, and for all other n from $N-1$ to 0 they try to solve the optimization

$$\sup_{\theta_n} \mathbb{E}\left[g(n, X_n) F_{\theta_n}(X_n) + g(n+1, X_{n+1})(1 - F_{\theta_n}(X_n))\right].$$

Since the objective function is differentiable almost everywhere with respect to theta, the authors suggest that the optimization can be carried out by gradient ascent.

Paragraph 2.3 gives on one hand some theoretical guarantees on the approximation properties of neural networks, and on the other hand it explains in practice how to optimize their parameters so that the decision function they learn is closest to the true optimal stopping binary decision.

Proposition 4 and *Corollary 5* provide crucial theoretical guarantees for the deep learning approach to optimal stopping. *Proposition 4* states that, for any fixed future stopping time $\tau_{n+1} \in \mathcal{T}_{n+1}$, and for any given depth $I \geq 2$ and precision $\varepsilon > 0$, there exist θ such that a neural network f_θ can approximate the optimal binary stopping decision at time n up to an error of at most ε in expected reward. This result leverages the expressive power of neural networks as universal function approximators for measurable binary functions.

Building on this, *Corollary 5* follows immediately from *Theorem 1* and *Proposition 4*: since the optimal stopping time can be decomposed into a sequence of binary decisions, and since each binary decision can be approximated arbitrarily well by a neural network, it follows that the entire optimal stopping time can be approximated arbitrarily closely. More precisely, for any given optimal stopping problem, depth $I \geq 2$, and precision $\varepsilon > 0$, there exist appropriate network widths and parameters θ such that the resulting stopping time achieves an expected reward within ε of the true optimal value.

Once the authors established good theoretical properties, they moved on towards the more practical neural network optimization framework. As previously mentioned, it all revolves around working backward. First, they set $f_{\theta_N} = 1$, then they want to choose θ_{N-1} such that it optimizes the expected reward at time $N-1$, $\mathbb{E}[r_{N-1}]$, where the reward at time n is the random variable

$$r_n = F_{\theta_n}(X_n) g(X_n, n) + (1 - F_{\theta_n}(X_n)) g(X_{\tau_{n+1}}, \tau_{n+1}).$$

Once θ_{N-1} has been estimated, they proceed with estimating θ_{N-2} by optimizing r_{N-2}, and so on, until θ_0.

In order to estimate the expected value, the authors simulate K independent sample paths, denoted by $(x_n^k)_{n=0}^N$ for $k = 1, \ldots, K$. For each sample path k, they compute a corresponding realized reward r_n^k, defined according to the learned stopping policy.

The empirical average of these realized rewards,

$$\widehat{R}_{\theta_n} = \frac{1}{K} \sum_{k=1}^{K} r_{\theta_n}^k,$$

provides an unbiased estimator of the expected reward $\mathbb{E}[r_{\theta_n}]$. By the law of large numbers, as $K \to \infty$,

$$\widehat{R}_n \longrightarrow \mathbb{E}[r_n] \quad \text{almost surely.}$$

Statistical Results About the Optimal Stopping Time Estimator

With the deep optimal stopping approach, the authors are able to find an estimate for the lower bound of the true optimal stopping time. They also derive an estimate for the upper bound using traditional methods like the Snell envelope. Once the lower and upper bounds have been estimated, they use the average of the two as a point estimate for the optimal stopping time, and they can build asymptotic confidence intervals based on the Central Limit Theorem (CLT).

Lower Bound

Once the stopping decisions f_n have been trained, the stopping time τ_θ that the authors recover, yields a lower bound

$$L = \mathbb{E}[g(\tau_\theta, X_{\tau_\theta})]$$

for the optimal value

$$V_0 = \sup_{\tau \in \mathcal{T}} \mathbb{E}[g(\tau, X_\tau)].$$

It is possible to estimate L via Monte Carlo approximation:

$$\widehat{L} = \frac{1}{K_L} \sum_{k=1}^{K_L} g(\tau_\theta^k, y_{\tau_\theta^k}^k),$$

where $\widehat{L}$ gives an unbiased estimate of the lower bound L, and by the law of large numbers, $\widehat{L} \to L$ as $K_L \to \infty$.

Upper Bound

The Snell envelope, representing the smallest super-martingale dominating the reward process $g(n, X_n)$, serves as the theoretical foundation for deriving upper bounds in optimal stopping problems. It decomposes into a martingale component M_H, capturing the inherent randomness of future rewards, and a predictable non-decreasing process A_H, reflecting the diminishing optionality over time. A dual formulation (Proposition 7) leverages this decomposition to bound the optimal value V_0: the upper bound is expressed as the expectation of the maximum difference between the reward $g(n, X_n)$ and a martingale M_n perturbed by zero-mean error terms ε_n.

To approximate this bound, a candidate stopping strategy τ^*, trained via neural networks, generates a value process H_n^*. The martingale part M^* is estimated using simulated continuation paths. These paths, combined with Monte Carlo averaging of continuation values C_n^*, produce noisy increments ΔM_n^k, and these increments aggregate into a martingale approximating M_H.

An unbiased estimator $\widehat{U}$, derived from independent simulations, then computes the upper bound by maximizing the adjusted reward $g(n, z_n^k) - M_n^k$ across time steps. The tightness of this bound depends on how closely the learned strategy τ^* aligns with the true optimal stopping rule, with errors diminishing as simulation fidelity increases. By ensuring independence between training, validation, and estimation datasets, the method provides a statistically rigorous certificate of performance for the neural network's policy.

Point Estimate and Central Limit Theorem

The authors take the point estimate of the optimal stopping time value as the average between the estimated upper and lower bounds:

$$\widehat{V} = \frac{\widehat{L} + \widehat{U}}{2},$$

where $\widehat{L}$ is the estimated lower bound and $\widehat{U}$ is the estimated upper bound.

It is possible to derive asymptotic confidence intervals for $\widehat{L}$ and $\widehat{U}$ based on the Central Limit Theorem.

For every $\alpha \in (0, 1]$, they have:

$$\left[\widehat{L} - z_{\alpha/2}\frac{\widehat{\sigma}_L}{\sqrt{K_L}},\ \widehat{L} + z_{\alpha/2}\frac{\widehat{\sigma}_L}{\sqrt{K_L}}\right] \quad \text{and} \quad \left[\widehat{U} - z_{\alpha/2}\frac{\widehat{\sigma}_U}{\sqrt{K_U}},\ \widehat{U} + z_{\alpha/2}\frac{\widehat{\sigma}_U}{\sqrt{K_U}}\right],$$

where $z_{\alpha/2}$ is the $(1 - \alpha/2)$ quantile of the standard normal distribution, and $\widehat{\sigma}_L, \widehat{\sigma}_U$ are empirical standard deviations.

By putting them together, they recover the 1-α asymptotic confidence interval for the estimator of the optimal stopping time $\widehat{V}$:

$$\left[\widehat{L} - z_{\alpha/2}\frac{\widehat{\sigma}_L}{\sqrt{K_L}},\ \widehat{U} + z_{\alpha/2}\frac{\widehat{\sigma}_U}{\sqrt{K_U}}\right].$$

Examples

In this chapter, the authors exemplify their method with three stopping time problems, with the main focus of showcasing its effectiveness in very high dimensions.

The three examples are estimating the optimal stopping rule of: Bermudan max-call options, Callable multi-barrier reverse convertibles (MBRCs), fractional Brownian motion (fBm).

Bermudan max-call options are one of the most studied examples in the numerics literature on optimal stopping problems. They are financial derivatives allowing the holder to exercise the option at predefined dates (e.g., monthly) to receive a payoff based on the maximum value of multiple underlying assets minus a strike price.

Considering d underlying assets, the derivative has payoff

$$\left(\max_{1\le i\le d} S_t^i - K\right)^+$$

and can be exercised at any point of a time grid $0 = t_0 < t_1 < \cdots < t_N$. Its price is given by

$$\sup_\tau \mathbb{E}\left[e^{-r\tau}\left(\max_{1\le i\le d} S_\tau^i - K\right)^+\right],$$

where the supremum is taken over all S-stopping times taking values in $\{t_0, t_1, \ldots, t_N\}$;

The price can be written as

$$\sup_{\tau\in\mathcal{T}} \mathbb{E}\left[g(\tau, X_\tau)\right],$$

for

$$g(n, x) = e^{-rt_n}\left(\max_{1\le i\le d} x_i - K\right)^+.$$

In the training phase, the authors simulate 8,192 independent full paths $(X_n^k)_{n=0}^N$ for $k = 1, \ldots, 8,192$, which are used to train the neural networks to approximate the optimal stopping decisions. The neural network trained in this example has depth $I = 3$ layers, and widths of the hidden layers $q_1 = q_2 = d + 40$, where d is the number of assets they consider.

To estimate the lower bound $\widehat{L}$, the authors simulate a much larger set of 4,096,000 independent full paths, apply the trained stopping rule to each path, and compute the corresponding rewards. The empirical average of these rewards provides an unbiased estimator of the expected payoff under the learned stopping policy. Similarly, to estimate the upper bound $\widehat{U}$, they simulate 1,024 base paths and generate 16,384 continuation paths from each one, allowing them to construct an estimate of the dual martingale and thereby obtain an upper bound for the optimal value.

Callable multi-barrier reverse convertibles (MBRCs) are financial products which pay coupons but convert into the worst-performing underlying asset if a trigger event (e.g., an asset falling below a barrier) occurs. The issuer (not the holder) decides when to "call" (terminate) the product, aiming to minimize losses. Optimal stopping here involves predicting when triggering the conversion is least costly, requiring analysis of multiple correlated assets and barrier conditions. MBRCs are relevant because they model real-world hybrid securities where stopping decisions directly impact financial outcomes, demanding sophisticated pricing models that account for path-dependent triggers and

multi-asset dynamics. The experimental setup is similar here with the previous one, except that the payoff function is more complex and the optimization problem is conceptually a minimization instead of a maximization, as the issuer of the derivative wants to minimize his loss instead of maximizing his gain. They observe similar results, with a tiny spread between $\widehat{L}$ and $\widehat{U}$. However, in this case, they don't go past 30 dimensions.

Finally, they perform the same experiment on a fractional Brownian motion (fBm). Such a stochastic process has a long-range dependence (governed by the Hurst parameter H), where increments are correlated over time. For $H \neq 0.5$, the process exhibits persistence ($H > 0.5$) or anti-persistence ($H < 0.5$), breaking the Markov property. This example is significant because stopping such a process optimally (e.g., to maximize the expected value of its terminal state) is challenging because traditional dynamic programming fails without Markovian structure. Indeed, the original process is unidimensional and non-Markovian, therefore they will transform it into a multidimensional Markov chain by incorporating all of the past information in the present.

More concretely, the authors want to approximate the supremum

$$\sup_{0 \leq \tau \leq 1} \mathbb{E}[W_\tau^H] \tag{3}$$

over all W^H-stopping times τ taking values between 0 and 1.

To do this, they discretize the time interval $[0, 1]$ into 100 parts and introduce the 100-dimensional Markov chain $(X_n)_{n=0}^{100}$ defined by

$$\begin{aligned} X_0 &= (0, 0, \ldots, 0), \\ X_1 &= (W_{t_1}^H, 0, \ldots, 0), \\ X_2 &= (W_{t_2}^H, W_{t_1}^H, 0, \ldots, 0), \\ &\vdots \\ X_{100} &= (W_{t_{100}}^H, W_{t_{99}}^H, \ldots, W_{t_1}^H). \end{aligned}$$

The corresponding discretized stopping problem

$$\sup_{\tau \in \mathcal{T}} \mathbb{E}[g(X_\tau)] \tag{4}$$

where $\mathcal{T}$ is the set of all X-stopping times and $g : \mathbb{R}^{100} \to \mathbb{R}$ is the projection $g(x_1, \ldots, x_{100}) = x_1$, provides an approximation of (3) from below.

They then computed estimates of (4) for different values of $H \in \{0.01, 0.05, 0.1, 0.15, \ldots, 1\}$ by training networks of the form (8), using a depth $I = 3$, input dimension $d = 100$, and hidden layer widths $q_1 = q_2 = 140$.

Observations

A few observations coming from the results of the experiments are that the expected reward $\widehat{L}$ coming from the learnt stopping rule is very close to the estimated theoretical upper bound, which implies that they have estimated the stopping rule with high precision, even in high dimensional setting: the upper bound is tight even up to dimension 500. Of course, it could be noticed that in general it is hardly surprising to find an extremely good estimate for the optimal stopping time which is indicated by the fact that $\widehat{L}$ and $\widehat{U}$ are so close. This is because it is possible to increase the number of epochs based on the desired precision for the estimation. What is really surprising is that the convergence is feasible and quite fast with such high epochs and dimensionality of the problem. For example, it takes less than 9 minutes to converge for the Bermudan max-call option problem with 500 dimensions. Another interesting observation is that the time required for the estimation increases linearly (and not exponentially!) with respect to the increase in the dimensions, because the computational complexity is mainly driven by the number of simulated paths and the size of the neural networks, both of which scale linearly with the input dimension.

3 Critical Analysis

Literature review and paper relevance

The review of the literature begins with the dynamic programming formulation via the Snell envelope, which provides an exact recursive characterization of the value process in discrete, finite-horizon settings. This framework translates the optimal stopping problem into a stochastic control problem, where the optimal global stopping time is obtained by

concatenating a sequence of local binary decisions: at each time step, the authors choose whether to stop or continue based on a comparison between immediate reward and expected future value. This theoretical foundation was laid in the 1950s–1960s with the contributions of Doob, Dynkin, and Snell [1, 2, 3], who established the formal probabilistic framework of the problem. The Snell envelope, introduced in this period, provides a martingale-based characterization of the value process as the smallest supermartingale dominating the payoff process. This result guarantees the existence of optimal stopping times and formalizes the problem in terms of conditional expectations.

In the 1970s–1990s, the focus shifted to constructive and numerical methods. Dynamic programming became the key algorithmic tool in the discrete-time setting, relying on Bellman recursions to compute value functions backward in time [4]. In continuous-time, low-dimensional models, particularly in finance, PDE methods were developed to solve optimal stopping problems through variational inequalities. Notably, in the Black-Scholes framework, the pricing of American options leads to a free-boundary PDE problem [5]. These methods were highly accurate for low-dimensional problems but suffered from the curse of dimensionality, making them computationally infeasible beyond two or three underlying state variables.

To address these limitations, the 1990s saw the rise of simulation-based methods, which aimed to approximate optimal stopping rules in high-dimensional spaces. The main idea was to use simulations to approximate the expectation

$$\mathbb{E}[g(\tau_{n+1}, X_{\tau_{n+1}}) \mid X_n]$$

called the continuation value function, which is the expected gain they get by deciding not to stop at time n. However, applying these Monte Carlo methods in a naive way also becomes intractable in high dimensions: at time n, it would require a batch of simulations for every possible value of our Markov chain X_n. In order to make the problem feasible, Monte Carlo simulations were associated with regression-based models of the type

$$X_n \mapsto \mathbb{E}[g(\tau_{n+1}, X_{\tau_{n+1}}) \mid X_n].$$

A landmark result in this direction was the Longstaff-Schwartz algorithm [6], which combined Monte Carlo simulation with least-squares regression to approximate the continuation value function in American option pricing. This allowed backward induction to be performed on simulated sample paths, projecting the continuation value onto a lower-dimensional space of basis functions (typically polynomials), and thus circumventing the need to discretize the entire state space.

In parallel, Tsitsiklis and Van Roy [7] proposed a more general approximate dynamic programming framework. Their approach also used simulation and function approximation but emphasized a broader class of approximators and established theoretical convergence guarantees. Both papers were critical in opening the door to solving high-dimensional optimal stopping problems.

However, these methods are prone to bias and instability because they depend on the choice of the function class used for regression, which may not be flexible enough to accurately capture the true continuation value, especially in high-dimensional or highly non-linear settings. Therefore, this paper utilizes techniques from deep reinforcement learning to bypass explicit value function estimation, directly learning optimal stopping policies (that is, the sequence of 0-1 decisions) by training neural networks on simulated trajectories to approximate the optimal decision boundaries at each time step. This approach is inspired by policy-based reinforcement learning, where the agent learns a mapping from states to actions (here: stop or continue) to maximize expected reward, without explicitly modelling or estimating the continuation value function [8].

As is well known, neural networks are universal function approximators [9], as they are expressive enough to approximate arbitrarily well any (measurable) decision function. This expressiveness is particularly valuable when the optimal policy does not correspond to a simple threshold or linear rule, which is often the case in realistic financial or operational settings. Therefore, this method provides a flexible and efficient way to approximate the optimal stopping time in high-dimensional settings, making it possible to solve problems previously out of reach. For example, classical approaches to optimal stopping, such as the Longstaff-Schwartz algorithm [6] and the approximate dynamic programming method of Tsitsiklis and Van Roy [7], were typically able to handle problems in up to 5–10 dimensions, mainly due to limitations in the flexibility of basis functions and the instability of regression in high dimensions. Dual methods such as those of Broadie and Glasserman [10] also suffered from scalability issues. This paper has pushed the dimensionality barrier significantly further, solving optimal stopping problems in up to more than 100 dimensions.

Limitations and Improvements

While the proposed deep learning approach to optimal stopping shows strong performance, several limitations remain. First, the method relies on the availability of a large number of simulated paths, which may become computationally expensive for very complex models or when high simulation fidelity is required. Second, although neural networks

are universal function approximators, the choice of architecture, activation functions, and optimization algorithms can significantly affect performance and may require careful tuning on a case-by-case basis. Future improvements could include the development of adaptive sampling techniques to reduce computational costs and the integration of uncertainty quantification to better control estimation errors. Moreover, extending the method to handle continuous-time stopping problems without explicit time discretization remains an open and promising direction for further research.

Finally, as a side note on the paper's structure, it is worth noting that the paper does not include a dedicated conclusion section and offers limited discussion of the results. This choice leaves the interpretation and broader implications of the findings largely to the reader, which, while encouraging independent reflection, also makes the work feel somewhat inconclusive. A more thorough synthesis of the results and a clearer articulation of the main takeaways would have strengthened the paper's impact and provided greater guidance for future research.

4 Conclusion

On a more personal level, as a group, we particularly appreciated this paper because we were intrigued by the general framework in which the authors operate: this helped us to develop our knowledge on multiple fronts. Firstly, we gained a deeper understanding of optimal stopping time theory and practice, gaining a precious tool that we will be able to apply not only in finance, but possibly in a variety of scientific settings. On top of this, we appreciated delving at the intersection of multiple fields that we encountered in our master and seeing how they were interacting with each other in this context, namely martingale theory, deep learning, dynamic programming, and reinforcement learning. Connecting the dots in the context of stopping times helped to solidify our knowledge in these area.

References

References

[1] J. L. Doob, *Stochastic Processes*, John Wiley & Sons, 1953.

[2] E. B. Dynkin, *Markov Processes*, Springer, 1963.

[3] J. L. Snell, Applications of martingale system theorems, *Transactions of the American Mathematical Society*, 73(2):293–312, 1952.

[4] R. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[5] H. P. McKean, A free boundary problem for the heat equation arising from a problem of optimal stopping, *Industrial Management Review*, 1965.

[6] F. A. Longstaff and E. S. Schwartz, Valuing American options by simulation: A simple least-squares approach, *The Review of Financial Studies*, 14(1):113–147, 2001.

[7] J. N. Tsitsiklis and B. Van Roy, Regression methods for pricing complex American-style options, *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.

[9] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks*, 4(2):251–257, 1991.

[10] M. Broadie and P. Glasserman, A stochastic mesh method for pricing high-dimensional American options, *Journal of Finance*, 59(4):2071–2112, 2004.

[11] D. Belomestny, Solving optimal stopping problems via dual optimization, *Annals of Applied Probability*, 21(5):1912–1931, 2011.