Optimal Stopping under Partial Observation: Near-Value Iteration

Enlu Zhou

Abstract

We propose a new approximate value iteration method, namely near-value iteration (NVI), to solve continuous-state optimal stopping problems under partial observation, which in general cannot be solved analytically and also pose a great challenge to numerical solutions. NVI is motivated by the expression of the value function as the supremum over an uncountable set of linear functions in the belief state. After a smart manipulation of the operations in the updating equation for the value function, we reduce the set to only two functions at every time step, so as to achieve significant computational savings. NVI yields a value function approximation bounded by the tightest lower and upper bounds that can be achieved by existing algorithms in the same class, so the NVI approximation is closer to the true value function than at least one of these bounds. We demonstrate the effectiveness of our approach on an example of pricing American options under stochastic volatility.

I. Introduction

Optimal stopping under partial observation (OSPO) arises in a number of applications, such as quality control and reliability [7], optimal investment under partial information [9], and optimal stock selling [15]. Despite its broad applicability, OSPO often cannot be solved analytically and poses a great challenge to numerical approaches. OSPO is closely related with another mathematical model: partially observable Markov decision process (POMDP) [11]. As a POMDP, OSPO can be transformed to an equivalent fully observable optimal stopping problem by introducing the belief state, which is the conditional distribution of the unobserved state given

E. Zhou is with the Department of Industrial & Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, IL 61801 USA (e-mail: enluzhou@illinois.edu).

This work was supported by the National Science Foundation under Grants ECCS-0901543 and CMMI-1130273, and by the Air Force Office of Scientific Research under YIP Grant FA-9550-12-1-0250.

the observation history. The equivalent fully observable problem can then be approached by dynamic programming in principle, but in general is very hard to solve. The main difficulty arises from the infinite dimensionality of the belief-state space in most problems (except some limited cases) where the underlying unobserved state is continuous. To be numerically tractable, the problem has to be reduced to finite and preferably low dimension. Dimension reduction techniques that specifically target continuous-state POMDPs have been developed in recent years, such as [17, 13, 3, 18]. With some modification, most of these approaches can be adapted to solve OSPO. Meanwhile, approximation methods [10, 12, 4, 8, 14] were proposed in the specific context of solving OSPO.

Many of the aforementioned algorithms that focus on computing approximate value functions can be viewed as a combination of approximate filtering and approximate dynamic programming methods, where approximate filtering is used to construct a mesh on the belief space, and approximate dynamic programming is used to solve the fully observable problem on this discrete mesh. Due to the need to construct a mesh over the whole belief space, approximate filtering is often computationally expensive, and probably consumes much more computing time than approximate dynamic programming. Hence, we want to seek other value function approximation methods that can avoid approximate filtering. Please note that we only focus on the *offline* computation of value functions, whereas filtering is unavoidable when using the solved value functions and the induced policy for an *online* run, or in other words, a realization of the system.

In this paper, we propose an approximate value iteration approach by characterizing the structure of the value function. Our approach is motivated by [6], where Hauskrecht interpreted a number of value-function approximation methods as applications of Jensen's inequality to the exact value iteration in different ways. We extend his analysis to continuous-state OSPO to show a representation of the value function and the recursive iteration of the value function. This is only an intermediate step for us to derive a new value function approximation and its iteration, which we name "near-value iteration" (NVI). NVI is expected to yield better approximation to the true value function than most of the value-function approximation methods that are summarized in [6]. NVI is also extremely simple, requiring the updating of only one function per iteration; whereas the exact value iteration for the true value function would require the updating of an infinite number of functions, which is impossible to carry out in practice. In addition, the algorithm based on NVI only needs to construct a mesh over the state-observation space rather

than the belief space. Therefore, our algorithm is expected to be significantly faster than many of the algorithms that we mentioned above.

II. Optimal Stopping under Partial Observation

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space hosting a process $\{X_k, k = 0, 1, \ldots\}$ that is not directly observable and another process $\{Y_k, k = 0, 1, \ldots\}$ that is observable. The two processes satisfy the following equations:

$$X_k = f(X_{k-1}, W_k), \quad k = 1, 2, \ldots, \tag{1}$$

$$Y_k = h(X_k, Y_{k-1}, \widetilde{W}_k), \quad k = 1, 2, \ldots, \quad Y_0 = h(X_0, \widetilde{W}_0), \tag{2}$$

where k is the time index, the *unobserved* state X_k is in a continuous state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, the observation Y_k is in a continuous observation space $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$, and the random disturbance $W_k \in \mathbb{R}^{n_w}$ and $\widetilde{W}_k \in \mathbb{R}^{n_v}$ are sequences of independent and identically distributed (i.i.d.) random vectors with known distributions. Assume that $\{W_k\}$ and $\{\widetilde{W}_k\}$ are independent of each other, and also independent of the initial state X_0 and the initial observation Y_0. We also assume a prior distribution π on the initial state X_0. Eqn. (1) is often referred to as the state equation, and eqn. (2) as the observation equation.

Denote the filtration generated by $\{Y_k\}$ as $(\mathcal{F}_k^Y)$, where $\mathcal{F}_k^Y$ is the σ-algebra generated by $\{Y_s, 0 \leq s \leq k\}$. A random time $\tau : \Omega \to \{0, 1, \ldots\}$ is an $\mathcal{F}_k^Y$-*stopping time* if $\{\omega \in \Omega : \tau(\omega) \leq k\} \in \mathcal{F}_k^Y$ for every k. It intuitively means that the stopping time τ is completely determined by the observation history up to time k. We consider the finite-horizon optimal stopping problem under partial observation of the following form:

$$V_0(\pi, y) = \max_{\tau \in \{0,1,\ldots,T\}, \mathcal{F}^Y - adapted} E[g(\tau, X_\tau, Y_\tau) | X_0 \sim \pi, Y_0 = y], \tag{3}$$

where T is the time horizon, $g : \{0, \ldots, T\} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ is the reward function, and a stopping time τ^* that achieves V_0 is the optimal stopping time.

Throughout the paper we assume all the probability distributions mentioned admit densities with respect to Lebesgue measure. The observable process $\{Y_k\}$ can be used to infer the unobservable process $\{X_k\}$ through a density estimate, which is the conditional density of X_k based on the history of observations $Y_{0:k} \triangleq \{Y_0, \ldots, Y_k\}$:

$$B_k(x) \triangleq p_{X_k}(x | Y_{0:k}).$$

This conditional distribution or density is often referred to as the *belief state*. Given a realization of the observations $Y_{0:k} = y_{0:k}$, the belief state is correspondingly denoted as b_k. Using Bayes' rule and the fact that $\{(X_k, Y_k)\}$ is a bivariate Markov process, we can show that b_k evolves as follows:

$$\begin{aligned} b_0(x_0) &= \frac{p(y_0|x_0)\pi(x_0)}{\int_{\mathcal{X}} p(y_0|x_0)\pi(x_0)dx_0}, \\ b_k(x_k) &= \frac{\int_{\mathcal{X}} p(x_k, y_k|x_{k-1}, y_{k-1})b_{k-1}(x_{k-1})dx_{k-1}}{\int_{\mathcal{X}} p(y_k|x_{k-1}, y_{k-1})b_{k-1}(x_{k-1})dx_{k-1}}, \quad k = 1, 2, \ldots, \end{aligned} \tag{4}$$

where the conditional densities $p(x_k, y_k|x_{k-1}, y_{k-1})$ and $p(y_k|x_{k-1}, y_{k-1})$ are induced by (1), (2) and distributions of W_k and $\widetilde{W}_k$. Noticing that the righthand side of (4) only depends on b_{k-1}, y_{k-1}, and y_k, and replacing the realization $y_{0:k}$ by its random variable $Y_{0:k}$, eqn. (4) can be abstractly rewritten as

$$B_k = \phi(B_{k-1}, Y_{k-1}, Y_k), \tag{5}$$

where (Y_{k-1}, Y_k) is characterized by the time-homogeneous conditional distribution $p(Y_{k-1}, Y_k|B_{k-1})$ that is induced by (1) and (2), and does not depend on $\{Y_0, \ldots, Y_{k-2}\}$.

By introducing the belief state, the partially observable optimal stopping problem can be transformed to a fully observable one, which is a well-known technique (for example, c.f. Chapter 5 in [1]). Define

$$\widetilde{g}(k, B_k, Y_k) \triangleq E[g(k, X_k, Y_k)|\mathcal{F}_k^Y] = \int g(k, x_k, Y_k)B_k(x_k)dx_k.$$

Then the problem (3) can be rewritten as

$$V_0(\pi, y) = \max_{\tau \in \{0,\ldots,T\}, \mathcal{F}^Y - adapted} E[\widetilde{g}(\tau, B_\tau, Y_\tau)|X_0 \sim \pi, Y_0 = y]. \tag{6}$$

Formulation (6) transforms (3) to an equivalent fully observable optimal stopping problem, where the state is the belief state B_k and its state equation is (5). The dynamic programming (DP) recursion for solving (6) is

$$\begin{aligned} V_T(B_T, Y_T) &= \widetilde{g}(T, B_T, Y_T), \\ V_k(B_k, Y_k) &= \max\left\{\widetilde{g}(k, B_k, Y_k), E[V_{k+1}(B_{k+1}, Y_{k+1})|B_k, Y_k]\right\}, \quad k = T-1, \ldots, 0. \end{aligned} \tag{7}$$

The second term on the righthand side in (7) is often referred to as the *continuation value*

$$C(B_k, Y_k) \triangleq E[V_{k+1}(B_{k+1}, Y_{k+1})|B_k, Y_k].$$

The exact computation of the DP recursion (7) is intractable for most problems. Besides the "curse of dimensionality" that is common to fully observable Markov decision processes (MDPs), (7) also suffers from the usual infinite-dimensionality of the belief state, since it is a density function of a continuous random variable X_k. The infinite dimensionality of the DP recursion prevents us borrowing directly from the existing vast body of approximate dynamic programming techniques that are intended for the usual setup of finite dimensionality.

III. Characterization of Value Function

A classical result for finite-state POMDPs is: provided that the one-step reward function is continuous and convex, after a finite number of DP recursions the value function can be represented as the maximum on a set of linear functions of the belief state, and hence is piecewise linear and convex in the belief state (for example, c.f. [16]). We extend this result and the analysis in [6] to the OSPO formulated above, as stated in the following theorem with its proof in the Appendix.

Theorem 1. *The value function can be expressed as*

$$V_k(b_k, y_k) = \sup_{\alpha_k \in \Gamma_k} \int_{\mathcal{X}} b_k(x_k)\alpha_k(x_k, y_k)dx_k, \quad \forall k \leq T-1. \tag{8}$$

where

$$\Gamma_{T-1} = \left\{ g(T-1, x_{T-1}, y_{T-1}), \int_{\mathcal{Y}} \int_{\mathcal{X}} g(T, x_T, y_T) p(x_T, y_T | x_{T-1}, y_{T-1}) dx_T dy_T \right\},$$

and Γ_k is uncountable for all $k < T-1$ and satisfies the following recursion:

$$\Gamma_k = \left\{ g(k, x_k, y_k), \int_{\mathcal{Y}} \int_{\mathcal{X}} \alpha_{k+1}^{*(y_{k+1})}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1} | x_k, y_k) dx_{k+1} dy_{k+1} \;\middle|\; \alpha_{k+1}^{*(y_{k+1})} \triangleq \arg \sup_{\alpha_{k+1} \in \Gamma_{k+1}} \int_{\mathcal{X}} \alpha_{k+1}(x_{k+1}, y_{k+1}) \left(\int_{\mathcal{X}} p(x_{k+1}, y_{k+1} | x_k, y_k) b_k(x_k) dx_k \right) dx_{k+1} \right\}.$$

Since the value function is expressed as the supremum on a set of linear functions of b_k, we immediately have the following corollary.

Corollary 1. *The value function $V_k(b_k, y_k)$ is convex in b_k.*

Theoretically, if we could update and store all the α functions in the set Γ_k, then we could carry out the dynamic programming recursion and compute the value function exactly. However,

except $|\Gamma_{T-1}| = 2$, for all $k < T-1$ the set Γ_k is uncountable. Therefore, we need some approximation scheme that can guarantee a finite and small number of approximate α functions so that approximate dynamic programming can be done easily. Towards that goal, we first write down the exact DP recursion using the α-function expression of the value function. According to (17) (in the proof of Theorem 1, shown in the Appendix), for all $k < T$ we have

$$V_k(b_k, y_k) = \max\Big\{\widetilde{g}(k, b_k, y_k), \ldots \int_{\mathcal{Y}} \sup_{\alpha_{k+1}\in\Gamma_{k+1}} \int_{\mathcal{X}} b_k(x_k) \left(\int_{\mathcal{X}} \alpha_{k+1}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1}\right) dx_k dy_{k+1}\Big\}. \quad (9)$$

Using Jensen's inequality on the second term of (9), we obtain an upper bound on the value function as follows.

$$\begin{aligned} V_k(b_k, y_k) &\leq \max\Big\{\widetilde{g}(k, b_k, y_k), \ldots \\ &\qquad \int_{\mathcal{Y}}\int_{\mathcal{X}} b_k(x_k) \sup_{\alpha_{k+1}\in\Gamma_{k+1}} \left(\int_{\mathcal{X}} \alpha_{k+1}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1}\right) dx_k dy_{k+1}\Big\} \\ &= \max_{\overline{\alpha}_k\in\overline{\Gamma}_k} \int_{\mathcal{X}} b_k(x_k)\overline{\alpha}_k(y_k, x_k) dx_k \triangleq \overline{V}_k(b_k, y_k), \end{aligned} \quad (10)$$

where $\overline{\Gamma}_k = \{\overline{\alpha}_k^1, \overline{\alpha}_k^2\}$, and $\overline{\alpha}_k^1(x_k, y_k) = g(k, x_k, y_k)$,

$$\overline{\alpha}_k^2(x_k, y_k) = \int_{\mathcal{Y}} \sup_{\alpha_{k+1}\in\Gamma_{k+1}} \left(\int_{\mathcal{X}} \alpha_{k+1}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1}\right) dy_{k+1}. \quad (11)$$

Similarly, using Jensen's inequality on the second term of (9) in the other direction, we obtain a lower bound on the value function as follows.

$$\begin{aligned} V_k(b_k, y_k) &\geq \max\Big\{\widetilde{g}(k, b_k, y_k), \ldots \\ &\qquad \sup_{\alpha_{k+1}\in\Gamma_{k+1}} \int_{\mathcal{Y}}\int_{\mathcal{X}} b_k(x_k) \left(\int_{\mathcal{X}} \alpha_{k+1}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1}\right) dx_k dy_{k+1}\Big\} \\ &= \sup_{\underline{\alpha}_{k+1}\in\underline{\Gamma}_k} \int_{\mathcal{X}} b_k(x_k)\underline{\alpha}_k(x_k, y_k) dx_k \triangleq \underline{V}_k(b_k, y_k), \end{aligned} \quad (12)$$

where $\underline{\Gamma}_k = \{\underline{\alpha}_k^1, \underline{\Gamma}_k^-\}$, where $\underline{\alpha}_k^1 = g(k, x_k, y_k)$, and $\underline{\Gamma}_k^-$ consists of functions

$$\underline{\alpha}_k = \int_{\mathcal{Y}}\int_{\mathcal{X}} \alpha_{k+1}(y_{k+1}, x_{k+1}) p(y_{k+1}, x_{k+1}|x_k, y_k) dx_{k+1} dy_{k+1}, \quad \alpha_{k+1} \in \Gamma_{k+1}. \quad (13)$$

The two bounds $\underline{V}_k$ and $\overline{V}_k$ correspond to the tightest lower and upper bounds in [6], namely the unobservable MDP (UMDP) approximation and the fast informed bound, respectively (c.f. Fig. 15 there for a summary of all the existing bounds). In the following, we derive another

approximate value function $\widetilde{V}_k$ that is bounded by $\underline{V}_k$ and $\overline{V}_k$, so $\widetilde{V}_k$ is a better approximation than at least one of $\underline{V}_k$ and $\overline{V}_k$ to the true value function V_k.

From (10), we have

$$\begin{aligned}
&\overline{V}_k(b_k, y_k) \\
= \;& \max\Big\{\widetilde{g}(k, b_k, y_k), \ldots \\
&\int_{\mathcal{X}} b_k(x_k) \int_{\mathcal{Y}} \sup_{\alpha_{k+1}\in\Gamma_{k+1}} \left(\int_{\mathcal{X}} \alpha_{k+1}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1}\right) dy_{k+1} dx_k\Big\},
\end{aligned}$$

and applying Jensen's inequality to move the supremum to the left yields

$$\begin{aligned}
&\overline{V}_k(b_k, y_k) \\
\geq \;& \max\Big\{\widetilde{g}(k, b_k, y_k), \ldots \\
&\int_{\mathcal{X}} b_k(x_k) \sup_{\alpha_{k+1}\in\Gamma_{k+1}} \left(\int_{\mathcal{Y}}\int_{\mathcal{X}} \alpha_{k+1}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1}\right) dy_{k+1} dx_k\Big\} \\
= \;& \max_{\widetilde{\alpha}_k\in\widetilde{\Gamma}_k} \int_{\mathcal{X}} b_k(x_k)\widetilde{\alpha}_k(x_k, y_k) dx_k \;\triangleq\; \widetilde{V}_k(b_k, y_k), \qquad (14)
\end{aligned}$$

where $\widetilde{\Gamma}_k = \{\widetilde{\alpha}_k^1, \widetilde{\alpha}_k^2\}$, and $\widetilde{\alpha}_k^1 = g(k, x_k, y_k)$,

$$\widetilde{\alpha}_k^2 = \sup_{\alpha_{k+1}\in\Gamma_{k+1}} \int\int \alpha_{k+1}(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1} dy_{k+1}. \qquad (15)$$

Similarly, from (12), applying Jensen's inequality we obtain

$$\underline{V}_k(b_k, y_k) \;\leq\; \widetilde{V}_k(b_k, y_k),$$

where $\widetilde{V}_k(b_k, y_k)$ is exactly the same as that defined in (14). Summarizing the above the inequalities, we have the following proposition.

Proposition 1. *For all $k < T$, the following inequalities hold:*

$$\underline{V}_k(b_k, y_k) \;\leq\; \widetilde{V}_k(b_k, y_k) \;\leq\; \overline{V}_k(b_k, y_k), \quad \underline{V}_k(b_k, y_k) \;\leq V_k(b_k, y_k) \;\leq\; \overline{V}_k(b_k, y_k),$$

where $\overline{V}_k$ is defined by the recursive equations of (10) and (11), $\underline{V}_k$ defined by (12) and (13), and $\widetilde{V}_k$ defined by (14) and (15).

In order to have implementable algorithms, we should use the approximate updating of α-functions iteratively, i.e., to replace the true α functions in (11), (13), and (15) by the approximate α-functions from the previous iteration. With a little abuse of notations, we use the same notations

$\overline{\alpha}$, $\underline{\alpha}$, $\widetilde{\alpha}$, $\overline{V}$, $\underline{V}$, $\widetilde{V}$ to denote the iterative approximations in the following. It is obvious that the iterative approximations preserve the directions of the inequalities, and hence the relation in Proposition 1 still holds. By a simple induction argument, we can see that the differences between $\underline{V}_k$, $\overline{V}_k$, and V_k enlarge as the number of iterations increases; hence, this class of methods are more suitable for problems with a small time horizon.

IV. Near-Value Iteration

As seen from Proposition 1, $\widetilde{V}_k$ is a better approximation to the true value function than at least one of the other two approximations $\underline{V}_k$ and $\overline{V}_k$, although the sign of $(\widetilde{V}_k - V_k)$ is undecided for a general problem. In addition, by examining the updating equations (15) for $\widetilde{\alpha}_k$, (11) for $\overline{\alpha}_k$, and (13) for $\underline{\alpha}_k$, we find the updating of $\widetilde{\alpha}_k$ the least computationally expensive: it has a constant size of two of $\widetilde{\alpha}_k$ functions for every iteration, whereas the set of $\underline{\alpha}_k$ has an increasing size $(T-k+1)$ as the recursion iterates backwards in time k; although the set of $\overline{\alpha}_k$ also has a constant size of two, the $\overline{\alpha}_k^2$ function involves a maximum within an integral, which can be very hard to compute numerically. Therefore, we propose "Near-Value Iteration (NVI)" to obtain $\widetilde{V}_k$ by recursively computing the $\widetilde{\alpha}_k$ functions according to (15). NVI is much more computationally effective than the algorithms based on $\underline{\alpha}_k$ (corresponding to UMDP) and $\overline{\alpha}_k$ (corresponding to the fast informed bound).

Near-Value Iteration (NVI)

- *Set* $\widetilde{\alpha}_T^l(x_T, y_T) = g(T, x_T, y_T), \quad l = 1, 2.$
- *For* $k = T-1, \ldots, 0$*, set*

$$\begin{aligned} \widetilde{\alpha}_k^1(x_k, y_k) &= g(k, x_k, y_k), \\ \widetilde{\alpha}_k^2(x_k, y_k) &= \max_{l=1,2} \int_{\mathcal{Y}} \int_{\mathcal{X}} \widetilde{\alpha}_{k+1}^l(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1} | x_k, y_k) dx_{k+1} dy_{k+1}. \end{aligned} \tag{16}$$

- *Approximate value function: set* $\widetilde{V}_0 = \max_{l=1,2} \int b_0(x_0) \widetilde{\alpha}_0^l(x_0, y_0) dx_0$.

NVI is not a readily implementable algorithm, because it involves integrals that cannot be evaluated exactly. One solution is to approximate the integrals numerically on a mesh of grid points. We construct a stochastic mesh by first simulating N sample paths of the $\{X_k\}$ and $\{Y_k\}$ processes to obtain grid points $\{(X_k^i, Y_k^i)\}$, and then computing the transition probabilities between the grid points. With this way of generating the mesh, as pointed out in [2], the grid points $\{(X_{k+1}^1, Y_{k+1}^1), \ldots, (X_{k+1}^N, Y_{k+1}^N)\}$ can be viewed as i.i.d. samples from a mixture

distribution of the transition kernels:

$$(X_{k+1}^j, Y_{k+1}^j) \overset{\text{iid}}{\sim} \frac{1}{N}\sum_{n=1}^{N} p(x_{k+1}, y_{k+1}|X_k^n, Y_k^n), \quad j = 1, \ldots, N.$$

In addition, assuming that $q(x_{k+1}, y_{k+1}|x_k, y_k)$ is a p.d.f. and p is absolutely continuous with respect to q, then the integral in NVI can be rewritten as

$$\begin{aligned} & \int_{\mathcal{Y}}\int_{\mathcal{X}} \widetilde{\alpha}_{k+1}^l(x_{k+1}, y_{k+1}) p(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1} dy_{k+1} \\ = & \int_{\mathcal{Y}}\int_{\mathcal{X}} \widetilde{\alpha}_{k+1}^l(x_{k+1}, y_{k+1}) \frac{p(x_{k+1}, y_{k+1}|x_k, y_k)}{q(x_{k+1}, y_{k+1}|x_k, y_k)} q(x_{k+1}, y_{k+1}|x_k, y_k) dx_{k+1} dy_{k+1} \\ = & E_q\left[\widetilde{\alpha}_{k+1}^l(X_{k+1}, Y_{k+1}) \frac{p(X_{k+1}, Y_{k+1}|x_k, y_k)}{q(X_{k+1}, Y_{k+1}|x_k, y_k)}\right], \end{aligned}$$

where E_q denotes the expectation taken with respect to q. Hence, taking

$$q(x_{k+1}, y_{k+1}|x_k, y_k) = \frac{1}{N}\sum_{n=1}^{N} p(x_{k+1}, y_{k+1}|X_k^n, Y_k^n),$$

the integral can be estimated by the i.i.d. samples $\{(X_{k+1}^1, Y_{k+1}^1), \ldots, (X_{k+1}^N, Y_{k+1}^N)\}$ from q using

$$\begin{aligned} & \frac{1}{N}\sum_{j=1}^{N} \widetilde{\alpha}_{k+1}^l(X_{k+1}^j, Y_{k+1}^j) \frac{p(X_{k+1}^j, Y_{k+1}^j|x_k, y_k)}{q(X_{k+1}^j, Y_{k+1}^j|x_k, y_k)} \\ = & \sum_{j=1}^{N} \widetilde{\alpha}_{k+1}^l(X_{k+1}^j, Y_{k+1}^j) \frac{p(X_{k+1}^j, Y_{k+1}^j|x_k, y_k)}{\sum_{n=1}^N p(X_{k+1}^j, Y_{k+1}^j|X_k^n, Y_k^n)}. \end{aligned}$$

Incorporating the above idea into NVI, we propose the following algorithm.

Algorithm 1. NVI on a Stochastic Mesh

- *Initialization: set a prior distribution π for the initial state X_0, and set number of sample paths N.*
- *Mesh Construction:*
  - *For $n = 1, 2, \ldots, N$, simulate a sample path of (1) and (2) to obtain $X^n = \{X_0^n, \ldots, X_T^n\}$ with $X_0^n \overset{\text{iid}}{\sim} \pi$ and $Y^n = \{Y_0^n, \ldots, Y_T^n\}$ with $Y_0^n = Y_0$.*
  - *For all $k = 0, \ldots, T-1$, $i = 1, \ldots, N$, $j = 1, \ldots, N$, compute the transition probabilities*

$$P_k^{ij} = \frac{p(X_{k+1}^j, Y_{k+1}^j|X_k^i, Y_k^i)}{\sum_{n=1}^N p(X_{k+1}^j, Y_{k+1}^j|X_k^n, Y_k^n)}.$$

- *Near-Value Iteration:*
  - *At $k = T$, set $\widehat{\alpha}_T^l(X_T^i, Y_T^i) = g(T, X_T^i, Y_T^i), \quad i = 1, \ldots, N,\ l = 1, 2.$*

– *For* $k = T-1, \ldots, 0$*, set*

$$\begin{aligned} \widehat{\alpha}_k^1(X_k^i, Y_k^i) &= g(k, X_k^i, Y_k^i), \quad i = 1, \ldots, N; \\ \widehat{\alpha}_k^2(X_k^i, Y_k^i) &= \max_{l=1,2} \sum_{j=1}^{N} P_k^{ij} \widehat{\alpha}_{k+1}^l(X_{k+1}^j, Y_{k+1}^j), \quad i = 1, \ldots, N. \end{aligned}$$

- *Approximate value function: set* $\widehat{V}_0 = \max_{l=1,2} \sum_{i=1}^{N} b_0(X_0^i) \widehat{\alpha}_0^l(X_0^i, Y_0^i)$*, where*

$$b_0(X_0^i) = \frac{p(Y_0|X_0^i)\pi(X_0^i)}{\sum_{j=1}^{N} p(Y_0|X_0^j)\pi(X_0^j)}, \quad i = 1, \ldots, N.$$

The computational time of Algorithm 1 is quadratic in the number of sample paths N and linear in the time horizon T. The algorithm converges to the ideal NVI almost surely as N goes to infinity, as stated in the following theorem with its proof in the Appendix.

Theorem 2. *Under the assumption that* $\int g(k, x_k, y_k) p(x_k, y_k|x_{k-1}, y_{k-1}) dx_k dy_k < \infty$ *for all* k*,*

$$\begin{aligned} \lim_{N\to\infty} \widehat{\alpha}_k^l(x_k, y_k) &= \widetilde{\alpha}_k^l(x_k, y_k) \quad w.p.1, \quad l = 1, 2, \\ \lim_{N\to\infty} \widehat{V}_0 &= \widetilde{V}_0 \quad w.p.1. \end{aligned}$$

Although our focus is to compute approximate value functions offline, we should mention that to use the induced policy from Algorithm 1 for an online run, one way is to estimate the belief state using a fixed-grid approximate filtering method on the mesh constructed in Algorithm 1 to obtain $\tilde{b}_k(x_k) = \sum_{i=1}^{N} w_k^i \delta(x_k - X_k^i)$, where w_k^i are the weights that sum up to one and $\delta(\cdot)$ is the Dirac delta function. By plugging such an approximate belief state into (14), we essentially compute $\arg\max_{l=1,2} \sum_{i=1}^{N} \tilde{w}_k^i \widehat{\alpha}_k^l(X_k^i, Y_k^i)$ to decide whether to stop (if $l = 1$) or continue (if $l = 2$). Since the induced policy is always suboptimal, the value associated with this policy provides a lower bound on the true value function. This lower bound can be estimated by simulating multiple sample paths with the induced policy.

V. Application: American Option Pricing under Stochastic Volatility

An application of optimal stopping is to price American options (c.f., for example, Chapter 8 in [5]). Stochastic volatility cannot be directly observed in reality, but can be inferred from the observed price of the asset. Hence, pricing American options under a stochastic volatility model falls into the framework of OSPO.

To fix ideas, we consider the asset price $\{S_t\}$ following a geometric Brownian motion and its volatility involves an *unobserved* mean-reverting process $\{X_t\}$, i.e.,

$$\begin{aligned} dS_t &= S_t\left(rdt + e^{X_t}dW_t\right), \\ dX_t &= \lambda(\theta - X_t)dt + \gamma d\widetilde{W}_t, \end{aligned}$$

where r represents the risk-free interest rate, λ is the mean-reversion rate, θ is the mean-reversion value, γ is the volatility of volatility, and $\{W_t\}$ and $\{\widetilde{W}_t\}$ are two independent standard Brownian motions. With a transformation $Y_t \triangleq \log(S_t)$, the log-price Y_t satisfies a Brownian motion. Suppose the observations of the price process occur at discrete time instants $\{0, \Delta, \ldots, k\Delta, \ldots, T\Delta\}$, simply denoted as $\{0, 1, \ldots, k, \ldots, T\}$ in the following. Based on the analytical solutions to the Brownian motion and the mean-reverting process, we apply Euler's scheme to discretize the original processes:

$$\begin{aligned} Y_{k+1} &= Y_k + \left(r - e^{2X_{k+1}}/2\right)\Delta + e^{X_{k+1}}\sqrt{\Delta}W_{k+1}, \\ X_{k+1} &= \theta + e^{-\lambda\Delta}(X_k - \theta) + \gamma\sqrt{\frac{1 - e^{-2\lambda\Delta}}{2\lambda}}\widetilde{W}_{k+1}, \end{aligned}$$

where $\{W_k\}$ and $\{\widetilde{W}_k\}$ are two independent sequences of i.i.d. standard Gaussian random variables. The price of an American put option (strictly speaking, Bermudan put option) is

$$V_0(x, y) = \max_{\tau \in \{0,\ldots,T\}, \mathcal{F}^Y adapted} E[e^{-r\tau}\max(K - S_\tau, 0)|X_0 = x, Y_0 = y].$$

In our numerical experiment, we use the following parameter values: $r = 0.05$, $\lambda = 1$, $\theta = 0.15$, $\gamma = 0.1$, $S_0 = 100$, $K = 100$, $X_0 = 0.15$, $\Delta = 0.1$ year, $T = 5, 10, 15$, and the number of sample paths in the stochastic mesh $N = [1000 : 1000 : 5000]$. For comparison, we adapted the UMDP and QMDP methods in [6] to continuous-state OSPO, and implemented them on the same stochastic mesh as NVI. We should note that the fast informed bound method on the constructed mesh here requires too much memory to run on our computer, so we resort to the QMDP method, which provides the next tightest upper bound other than the fast informed bound in this class of algorithms. In every parameter setting, each algorithm is run 50 times to obtain 50 replications of the option price.

In Table I, each entry shows the average and the standard error (in parentheses) of the 50 replications of the option price. It verifies that the output of NVI is bounded by the outputs of UMDP and QMDP. As we mentioned in the end of Section III, the differences between the three estimates of NVI, UMDP, and QMDP become larger as the number of exercise opportunities T

TABLE I

AMERICAN PUT OPTION PRICES

	$T=5$			$T=10$			$T=15$		
N	UMDP	NVI	QMDP	UMDP	NVI	QMDP	UMDP	NVI	QMDP
1000	30.39	30.71	31.54	40.43	42.17	43.90	46.93	50.14	52.69
	(0.12)	(0.11)	(0.11)	(0.13)	(0.12)	(0.12)	(0.14)	(0.12)	(0.13)
2000	30.37	30.65	31.25	40.52	41.71	42.96	46.90	49.27	51.12
	(0.09)	(0.09)	(0.08)	(0.10)	(0.08)	(0.07)	(0.11)	(0.09)	(0.09)
3000	30.30	30.55	31.03	40.27	41.38	42.45	46.93	48.96	50.46
	(0.07)	(0.07)	(0.07)	(0.08)	(0.07)	(0.07)	(0.08)	(0.07)	(0.07)
4000	30.23	30.47	30.93	40.45	41.43	42.40	46.92	48.77	50.12
	(0.07)	(0.06)	(0.06)	(0.07)	(0.06)	(0.06)	(0.08)	(0.06)	(0.06)
5000	30.17	30.37	30.80	40.35	41.22	42.07	46.96	48.68	49.96
	(0.06)	(0.05)	(0.05)	(0.07)	(0.06)	(0.06)	(0.07)	(0.06)	(0.07)

increases. In addition, the three estimates decrease as the number of sample paths N increases. That is because just as the stochastic mesh method in [2], the three estimates on stochastic mesh always have a positive bias that can be reduced by increasing the number of sample paths.

VI. CONCLUSION

We propose a new approximate value iteration, called near-value iteration, to solve the continuous-state optimal stopping problem under partial observation. This approach is computationally efficient in estimating the value function, and yields a better approximation than at least one of the two best existing approximations (UMDP and fast informed bound) in the same class of algorithms. We apply the algorithm to American option pricing under stochastic volatility. It should be mentioned that NVI can be easily extended to solve POMDPs in general by applying Jensen's inequality in the same fashion as here to the value function updating equation in a POMDP.

VII. APPENDIX

VII-A Proof for Theorem 1

We prove the result by induction. At final time T, the value function is

$$V_T(b_T, y_T) = \int_{\mathcal{X}} g(T, x_T, y_T) b_T(x_T) dx_T.$$

At $k = T-1$, first consider the continuation value

$$\begin{aligned}
& E[V_T(B_T, Y_T)|b_{T-1}, y_{T-1}] \\
= \;& E[V_T(\phi(b_{T-1}, y_{T-1}, Y_T), Y_T)|b_{T-1}, y_{T-1}] \\
= \;& \int_{\mathcal{Y}} \left(\int_{\mathcal{X}} g(T, x_T, y_T) \phi(b_{T-1}, y_{T-1} y_T)(x_T) dx_T \right) \left(\int_{\mathcal{X}} p(y_T|y_{T-1}, x_{T-1}) b_{T-1}(x_{T-1}) dx_{T-1} \right) dy_T \\
= \;& \int_{\mathcal{Y}} \left(\int_{\mathcal{X}} g(T, x_T, y_T) \int_{\mathcal{X}} p(x_T, y_T|x_{T-1}, y_{T-1}) b_{T-1}(x_{T-1}) dx_{T-1} dx_T \right) dy_T \\
= \;& \int_{\mathcal{X}} b_{T-1}(x_{T-1}) \alpha^2_{T-1}(x_{T-1}, y_{T-1}) dx_{T-1},
\end{aligned}$$

where $\alpha^2_{T-1}(x_{T-1}, y_{T-1}) \triangleq \int_{\mathcal{Y}} \int_{\mathcal{X}} g(T, x_T, y_T) p(x_T, y_T|x_{T-1}, y_{T-1}) dx_T dy_T$. The fourth line follows by plugging (4) for $\phi(b_{T-1}, y_{T-1}, y_T)$ into the third line. Then the value function at time $T-1$ can be written as

$$\begin{aligned}
& V_{T-1}(b_{T-1}, y_{T-1}) \\
= \;& \max \left\{ \int_{\mathcal{X}} g(T-1, x_{T-1}, y_{T-1}) b_{T-1}(x_{T-1}) dx_{T-1}, \int_{\mathcal{X}} b_{T-1}(x_{T-1}) \alpha^2_{T-1}(x_{T-1}, y_{T-1}) dx_{T-1} \right\} \\
= \;& \sup_{\alpha_{T-1} \in \Gamma_{T-1}} \int_{\mathcal{X}} b_{T-1}(x_{T-1}) \alpha_{T-1}(x_{T-1}, y_{T-1}) dx_{T-1},
\end{aligned}$$

where $\Gamma_{T-1} = \{\alpha^1_{T-1}, \alpha^2_{T-1}\}$ and $\alpha^1_{T-1}(x_{T-1}, y_{T-1}) \triangleq g(T-1, x_{T-1}, y_{T-1})$.

For all $k \leq T-1$, the continuation value is

$$\begin{aligned}
& E[V_k(B_k, Y_k)|b_{k-1}, y_{k-1}] \\
= \;& \int_{\mathcal{Y}} \left(\sup_{\alpha_k \in \Gamma_k} \int_{\mathcal{X}} \alpha_k(x_k, y_k) b_k(x_k) dx_k \right) \left(\int_{\mathcal{X}} p(y_k|y_{k-1}, x_{k-1}) b_{k-1}(x_{k-1}) dx_{k-1} \right) dy_k \\
= \;& \int_{\mathcal{Y}} \left(\sup_{\alpha_k \in \Gamma_k} \int_{\mathcal{X}} \alpha_k(x_k, y_k) \left(\int_{\mathcal{X}} p(x_k, y_k|x_{k-1}, y_{k-1}) b_{k-1}(x_{k-1}) dx_{k-1} \right) dx_k \right) dy_k \qquad (17) \\
= \;& \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \int_{\mathcal{X}} \alpha^{*(y_k)}_k(x_k, y_k) p(x_k, y_k|x_{k-1}, y_{k-1}) dx_k dy_k \right) b_{k-1}(x_{k-1}) dx_{k-1}, \qquad (18)
\end{aligned}$$

where (17) follows by substituting (4) for $b_k(x_k)$ in the line above, and

$$\alpha^{*(y_k)}_k \triangleq \arg \sup_{\alpha_k \in \Gamma_k} \int_{\mathcal{X}} \alpha_k(x_k, y_k) \left(\int_{\mathcal{X}} p(x_k, y_k|x_{k-1}, y_{k-1}) b_{k-1}(x_{k-1}) dx_{k-1} \right) dx_k.$$

Note that since for each y_k there are at least $|\Gamma_k|$ candidates for $\alpha^{*(y_k)}_k$, there are a total of $|\Gamma_k|^{|\mathcal{Y}|}$ candidates for $\int_{\mathcal{Y}} \int_{\mathcal{X}} \alpha^{*(y_k)}_k(x_k, y_k) p(x_k, y_k|x_{k-1}, y_{k-1}) dx_k dy_k$. Denote the set of candidates by Γ^-_{k-1}. Since $\mathcal{Y}$ is uncountable, the set Γ^-_{k-1} is also uncountable. Following (18), we have

$$E[V_k(B_k, Y_k)|b_{k-1}, y_{k-1}] = \sup_{\alpha_{k-1} \in \Gamma^-_{k-1}} \int_{\mathcal{X}} \alpha_{k-1}(x_{k-1}, y_{k-1}) b_{k-1}(x_{k-1}) dx_{k-1},$$

Then for all $k < T-1$, the value function is

$$\begin{aligned} V_k(b_k, y_k) &= \max\left\{\widetilde{g}(k, b_k, y_k), E[V_{k+1}(B_{k+1}, Y_{k+1})|b_k, y_k]\right\} \\ &= \max\left\{\int_{\mathcal{X}} g(k, x_k, y_k) b_k(x_k) dx_k, \sup_{\alpha_k \in \Gamma_k^-} \int_{\mathcal{X}} \alpha_k(x_k, y_k) b_k(x_k) dx_k\right\} \\ &= \sup_{\alpha_k \in \Gamma_k} \int_{\mathcal{X}} \alpha_k(x_k, y_k) b_k(x_k) dx_k, \end{aligned}$$

where $\Gamma_k = \Gamma_k^- \cup \{g(k, x_k, y_k)\}$ is uncountable.

VII-B Proof for Theorem 2

To lighten the notations, we use Z to denote the pair (X, Y). For example, $z_k = (x_k, y_k)$, $Z_k^i = (X_k^i, Y_k^i)$, $dz_k = dx_k dy_k$, etc. For $l = 1$, it is trivial since $\widehat{\alpha}_k^1(z_k) = \widetilde{\alpha}_k^1(z_k) = g(k, z)$ for all k. For $l = 2$, we prove the result by induction. Notice that for scalars a, b, c, and d,

$$\begin{aligned} & |\max(a, b) - \max(c, d)| \\ \leq\ & |\max(a, b) - \max(a, d)| + |\max(a, d) - \max(c, d)| \\ \leq\ & |b - d| + |a - c|. \end{aligned}$$

Hence,

$$\begin{aligned} & \left|\widetilde{\alpha}_k(Z_k^i) - \widehat{\alpha}_k(Z_k^i)\right| \\ \leq\ & \left|\int\int g(k+1, z_{k+1}) p(z_{k+1}|Z_k^i) dz_{k+1} - \sum_{j=1}^N P_k^{ij} g(k+1, Z_{k+1}^j)\right| ... \\ & + \left|\int\int \widetilde{\alpha}_{k+1}^2(z_{k+1}) p(z_{k+1}|Z_k^i) dz_{k+1} - \sum_{j=1}^N P_k^{ij} \widehat{\alpha}_{k+1}^2(Z_{k+1}^j)\right| \end{aligned} \tag{19}$$

Consider the first term on the righthand side of (19). As mentioned earlier, letting

$$q(z_{k+1}|Z_k^i) = \frac{1}{N} \sum_{n=1}^N p(z_{k+1}|Z_k^n),$$

then $\{Z_k^j, j = 1, \ldots, N\}$ are i.i.d. samples from q and according to the strong law of large numbers we have

$$\begin{aligned} & \left|\int\int g(k+1, z_{k+1}) p(z_{k+1}|Z_k^i) dz_{k+1} - \sum_{j=1}^N P_k^{ij} g(k+1, Z_{k+1}^j)\right| \\ =\ & \left|E_q\left[g(k+1, Z_{k+1}) \frac{p(Z_{k+1}|Z_k^i)}{q(Z_{k+1}|Z_k^i)}\right] - \frac{1}{N} \sum_{j=1}^N g(k+1, Z_{k+1}^j) \frac{p(Z_{k+1}^j|Z_k^i)}{q(Z_{k+1}^j|Z_k^i)}\right| \\ \to\ & 0 \quad w.p.1, \quad \text{as } N \to \infty. \end{aligned}$$

Consider the second term on the righthand side of (19).

$$\begin{aligned}
&\left| \int \widetilde{\alpha}^2_{k+1}(z_{k+1}) p(z_{k+1}|Z^i_k) dz_{k+1} - \sum_{j=1}^{N} P^{ij}_k \widehat{\alpha}^2_{k+1}(Z^j_{k+1}) \right| \\
\leq \quad &\left| \int \widetilde{\alpha}^2_{k+1}(z_{k+1}) p(z_{k+1}|Z^i_k) dz_{k+1} - \sum_{j=1}^{N} P^{ij}_k \widetilde{\alpha}^2_{k+1}(Z^j_{k+1}) \right| \ldots \\
&+ \left| \sum_{j=1}^{N} P^{ij}_k \widetilde{\alpha}^2_{k+1}(Z^j_{k+1}) - \sum_{j=1}^{N} P^{ij}_k \widehat{\alpha}^2_{k+1}(Z^j_{k+1}) \right|,
\end{aligned}$$

where the first term vanishes asymptotically w.p.1 due to the same argument as above, and the second term is upper bounded by

$$\sum_{j=1}^{N} P^{ij}_k \left| \widetilde{\alpha}^2_{k+1}(Z^j_{k+1}) - \widehat{\alpha}^2_{k+1}(Z^j_{k+1}) \right|,$$

which also vanishes asymptotically w.p.1 by the induction argument. Therefore, $|\widetilde{\alpha}_k(Z^i_k) - \widehat{\alpha}_k(Z^i_k)| \rightarrow 0 \quad w.p.1$. Using a similar approach we can show that $\left|\widetilde{V}_0 - \widehat{V}_0\right| \rightarrow 0 \quad w.p.1$ as $N \rightarrow \infty$.

REFERENCES

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[2] M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. *The Journal of Computational Finance*, 7(4):35 – 72, 2004.

[3] A. Brooks and S. Williams. A Monte Carlo update for parametric POMDPs. *International Symposium of Robotics Research*, Nov. 2007.

[4] I. Florescu and F. G. Viens. Stochastic volatility: option pricing using a multinomial recombining tree. *Applied Mathematical Finance*, 15(2):151 – 181, 2008.

[5] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2003.

[6] M. Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 13:33–95, 2000.

[7] U. Jensen and G.-H. Hsu. Optimal stopping by means of point process observations with applications in reliability. *Mathematics of Operations Research*, 18(3):645 – 657, 1993.

[8] M. Ludkovski. A simulation approach to optimal stopping under partial information. *Stochastic Processes and Applications*, 119(12):2071 – 2087, 2009.

[9] J.-P. Décamps T. Mariotti and S. Villeneuve. Investment timing under incomplete information. *Mathematics of Operations Research*, 30(2):472 – 500, 2005.

[10] G. Mazziotto. Approximations of the optimal stopping problem in partial observation. *Journal of Applied Probability*, 23(2):341 – 354, 1986.

[11] G.E. Monahan. A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science*, 28(1):1 – 16, 1982.

[12] H. Pham, W. Runggaldier, and A. Sellami. Approximation by quantization of the filter process and applications to optimal stopping problems under partial observation. *Monte Carlo Methods and Applicaitons*, 11(1):57 – 81, 2005.

[13] J. M. Porta, N. Vlassis, and M. T.J. Spaan amd P. Poupart. Point-based value iteration for continuous POMDPs. *Journal of Machine Learning Research*, 7:2329–2367, 2006.

[14] B. R. Rambharat and A. E. Brockwell. Sequential Monte Carlo pricing of American-style options under stochastic volatility models. *The Annals of Applied Statistics*, 4, No. 1, 222-265(1):222 – 265, 2010.

[15] R. Rishel and K. Helmes. A variational inequality sufficient condition for optimal stopping with application to an optimal stock selling problem. *SIAM Journal on Control and Optimization*, 45:580 – 598, 2006.

[16] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[17] S. Thrun. Monte Carlo POMDPs. *Advances in Neural Information Processing Systems*, 12:1064–1070, 2000.

[18] E. Zhou, M. C. Fu, and S. I. Marcus. Solving continuous-state POMDPs via density projection. *IEEE Transactions on Automatic Control*, 55(5):1101 – 1116, 2010.