PATHWISE STOCHASTIC OPTIMAL CONTROL

L. C. G. Rogers

Statistical Laboratory, University of Cambridge

Stochastic optimal control as we know it.

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$.

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$.   The value function

$$V_n(x) \equiv \sup E\big[\sum_{j=n}^{T-1} f_j(X_j, a_j) + F(X_T) \,\big|\, X_n = x\big]$$

satisfies the Bellman equations:

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$. The value function

$$V_n(x) \equiv \sup E\Big[\sum_{j=n}^{T-1} f_j(X_j, a_j) + F(X_T) \;\Big|\; X_n = x\Big]$$

satisfies the Bellman equations:

$$V_n(x) = \sup_a \big[\, f_n(x,a) + PV_{n+1}(x,a)\,\big], \qquad V_T(x) = F(x).$$

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$. The value function

$$V_n(x) \equiv \sup E\big[\sum_{j=n}^{T-1} f_j(X_j, a_j) + F(X_T) \,\big|\, X_n = x\big]$$

satisfies the Bellman equations:

$$V_n(x) = \sup_a \big[\, f_n(x,a) + PV_{n+1}(x,a) \,\big], \qquad V_T(x) = F(x).$$

If no closed form solution exists, then numerics hit problems if the dimension of $\mathcal{X}$ is large (think of $\mathcal{X} = \mathbb{R}^{50}$):

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$. The value function

$$V_n(x) \equiv \sup E\big[\sum_{j=n}^{T-1} f_j(X_j, a_j) + F(X_T) \,\big|\, X_n = x\big]$$

satisfies the Bellman equations:

$$V_n(x) = \sup_a \big[\, f_n(x,a) + PV_{n+1}(x,a)\,\big], \qquad V_T(x) = F(x).$$

If no closed form solution exists, then numerics hit problems if the dimension of $\mathcal{X}$ is large (think of $\mathcal{X} = \mathbb{R}^{50}$):

- How do we store V_n?

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$. The value function

$$V_n(x) \equiv \sup E\big[\sum_{j=n}^{T-1} f_j(X_j, a_j) + F(X_T) \,\big|\, X_n = x\big]$$

satisfies the Bellman equations:

$$V_n(x) = \sup_a \big[\, f_n(x,a) + PV_{n+1}(x,a)\,\big], \qquad V_T(x) = F(x).$$

If no closed form solution exists, then numerics hit problems if the dimension of $\mathcal{X}$ is large (think of $\mathcal{X} = \mathbb{R}^{50}$):

- How do we store V_n?
- How do we compute integrals over $\mathcal{X}$?

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$. The value function

$$V_n(x) \equiv \sup E\big[\sum_{j=n}^{T-1} f_j(X_j, a_j) + F(X_T) \,\big|\, X_n = x\big]$$

satisfies the Bellman equations:

$$V_n(x) = \sup_a \big[\, f_n(x,a) + PV_{n+1}(x,a)\,\big], \qquad V_T(x) = F(x).$$

If no closed form solution exists, then numerics hit problems if the dimension of $\mathcal{X}$ is large (think of $\mathcal{X} = \mathbb{R}^{50}$):

- How do we store V_n?
- How do we compute integrals over $\mathcal{X}$?
- Use Monte Carlo for high-dimensional integration ....

Stochastic optimal control as we know it.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big]$$

to be maximized over adapted $a \in \mathcal{A}$. The value function

$$V_n(x) \equiv \sup E\big[\sum_{j=n}^{T-1} f_j(X_j, a_j) + F(X_T) \,\big|\, X_n = x\big]$$

satisfies the Bellman equations:

$$V_n(x) = \sup_a \big[\, f_n(x,a) + PV_{n+1}(x,a)\,\big], \qquad V_T(x) = F(x).$$

If no closed form solution exists, then numerics hit problems if the dimension of $\mathcal{X}$ is large (think of $\mathcal{X} = \mathbb{R}^{50}$):

- How do we store V_n?
- How do we compute integrals over $\mathcal{X}$?
- Use Monte Carlo for high-dimensional integration .... but what controls would we use along a simulated path?

Simplest example: optimal stopping.

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

It can be shown (R., Haugh & Kogan) that

$$Y_0^* = \inf_{M \in \mathcal{M}_0} E\Big[\sup_{0 \le s \le T} (Z_s - M_s) \Big].$$

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

It can be shown (R., Haugh & Kogan) that

$$Y_0^* = \inf_{M \in \mathcal{M}_0} E\Big[\sup_{0 \leq s \leq T} (Z_s - M_s) \Big].$$

- the inf is attained;

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

It can be shown (R., Haugh & Kogan) that

$$Y_0^* = \inf_{M \in \mathcal{M}_0} E\Big[\sup_{0 \le s \le T} (Z_s - M_s) \Big].$$

- the inf is attained;
- whatever M you use, you get an *upper* bound on the value of the option;

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

It can be shown (R., Haugh & Kogan) that

$$Y_0^* = \inf_{M \in \mathcal{M}_0} E\Big[\sup_{0 \le s \le T} (Z_s - M_s) \Big].$$

- the inf is attained;
- whatever M you use, you get an *upper* bound on the value of the option;
- effective method for bounding the price of high-dimensional options.

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

It can be shown (R., Haugh & Kogan) that

$$Y_0^* = \inf_{M \in \mathcal{M}_0} E\Big[\sup_{0 \le s \le T} (Z_s - M_s) \Big].$$

- the inf is attained;
- whatever M you use, you get an *upper* bound on the value of the option;
- effective method for bounding the price of high-dimensional options.

COULD A SIMILAR APPROACH WORK MORE GENERALLY??

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

It can be shown (R., Haugh & Kogan) that

$$Y_0^* = \inf_{M \in \mathcal{M}_0} E\Big[\sup_{0 \le s \le T} (Z_s - M_s) \Big].$$

- the inf is attained;
- whatever M you use, you get an *upper* bound on the value of the option;
- effective method for bounding the price of high-dimensional options.

COULD A SIMILAR APPROACH WORK MORE GENERALLY??

- optimising over a *much* larger class;

Simplest example: optimal stopping.

If we stop at time τ, we get reward Z_τ. The American option pricing problem (=optimal stopping problem) with horizon T is to find stopping time τ^* such that

$$Y_0^* \equiv \sup_{\tau \in \mathcal{T}} E[Z_\tau].$$

It can be shown (R., Haugh & Kogan) that

$$Y_0^* = \inf_{M \in \mathcal{M}_0} E\Big[\sup_{0 \le s \le T} (Z_s - M_s) \Big].$$

- the inf is attained;
- whatever M you use, you get an *upper* bound on the value of the option;
- effective method for bounding the price of high-dimensional options.

COULD A SIMILAR APPROACH WORK MORE GENERALLY??

- optimising over a *much* larger class;
- many possible laws to consider;

Problem formulation.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j,a_j) + F(X_T)\Big] \equiv E\Big[\sum_{j=0}^{T} f_j(X_j,a_j)\Big],$$

to be max'ed over adapted $a \in \mathcal{A}$.

Problem formulation.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big] \equiv E\Big[\sum_{j=0}^{T} f_j(X_j, a_j)\Big],$$

to be max'ed over adapted $a \in \mathcal{A}$. Transitions have density $\varphi(x, y; a)$ wrto Markovian reference measure P^*.

Problem formulation.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big] \equiv E\Big[\sum_{j=0}^{T} f_j(X_j, a_j)\Big],$$

to be max'ed over adapted $a \in \mathcal{A}$. Transitions have density $\varphi(x, y; a)$ wrto Markovian reference measure P^*. Set

$$\Lambda_t(a) \equiv \prod_{r=0}^{t-1} \varphi(X_r, X_{r+1}; a_r) \equiv \prod_{r=0}^{t-1} \varphi_{r+1}(a_r) \ \in L^1(\mathcal{F}_t);$$

Problem formulation.

Controlled Markov process with values in $\mathcal{X}$, with finite horizon, and objective

$$E\Big[\sum_{j=0}^{T-1} f_j(X_j, a_j) + F(X_T)\Big] \equiv E\Big[\sum_{j=0}^{T} f_j(X_j, a_j)\Big],$$

to be max'ed over adapted $a \in \mathcal{A}$. Transitions have density $\varphi(x, y; a)$ wrto Markovian reference measure P^*. Set

$$\Lambda_t(a) \equiv \prod_{r=0}^{t-1} \varphi(X_r, X_{r+1}; a_r) \equiv \prod_{r=0}^{t-1} \varphi_{r+1}(a_r) \ \in L^1(\mathcal{F}_t);$$

so the problem is

$$V_0(X_0) = \sup_{a\in\mathcal{A}} v_0(X_0; a) \equiv \sup_{a\in\mathcal{A}} E^*\Big[\sum_{j=0}^{T} \Lambda_j(a) f_j(X_j, a_j)\Big].$$

Main result, 1.

Fixing $a \in \mathcal{A}$, for any martingale M,

$$\begin{aligned} v_0(X_0; a) &= E^*\Big[\sum_{j=0}^{T} \Lambda_j(a) f_j(X_j, a_j)\Big] \\ &= E^*\Big[\sum_{j=0}^{T} \Lambda_j(a)\{f_j(X_j, a_j) + \Delta M_{j+1}\}\Big] \end{aligned}$$

Main result, 1.

Fixing $a \in \mathcal{A}$, for any martingale M,

$$\begin{aligned} v_0(X_0; a) &= E^*\Big[\sum_{j=0}^{T} \Lambda_j(a) f_j(X_j, a_j)\Big] \\ &= E^*\Big[\sum_{j=0}^{T} \Lambda_j(a)\{f_j(X_j, a_j) + \Delta M_{j+1}\}\Big] \end{aligned}$$

where $\Delta M_{j+1} = Ph_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1})\varphi_{j+1}(a_j)$, $h_{T+1} \equiv 0$.

Main result, 1.

Fixing $a \in \mathcal{A}$, for any martingale M,

$$\begin{aligned} v_0(X_0; a) &= E^*\left[\sum_{j=0}^{T} \Lambda_j(a) f_j(X_j, a_j) \right] \\ &= E^*\left[\sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + \Delta M_{j+1} \} \right] \end{aligned}$$

where $\Delta M_{j+1} = P h_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1}) \varphi_{j+1}(a_j)$, $h_{T+1} \equiv 0$. Hence

$$\begin{aligned} V_0(X_0) &= \sup_{a \in \mathcal{A}} v_0(X_0; a) \\ &= \sup_{a \in \mathcal{A}} E^*\left[\sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + P h_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1}) \varphi_{j+1}(a_j) \} \right] \\ &= \sup_{a \in \mathcal{A}} E^*\left[h_0(X_0) + \sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + P h_{j+1}(X_j, a_j) - h_j(X_j) \} \right] \\ &\leq E^*\left[\sup_{a} \left\{ h_0(X_0) + \sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + P h_{j+1}(X_j, a_j) - h_j(X_j) \} \right\} \right] \end{aligned}$$

Main result, 2.

So

$$
\begin{aligned}
V_0(X_0) &\le h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \Lambda_j(a)\{f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\}\Big] \\
&\le h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \Lambda_j(a)\{\, f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\,\}^+\Big]
\end{aligned}
$$

Main result, 2.

So

$$
\begin{aligned}
V_0(X_0) &\le h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \; \Lambda_j(a)\{f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\} \Big] \\
&\le h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \; \Lambda_j(a)\{\, f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j) \,\}^+ \Big]
\end{aligned}
$$

Taking infimum over the functions h_j, we get

$$
V_0(X_0) \le \inf_{(h_j)} \Big\{ h_0(X_0) + E^*\big[\sum_{j=0}^{T} \sup_a \; \Lambda_j(a)\{\, f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j) \,\}^+ \,\big] \Big\}
$$

Main result, 2.

So

$$\begin{aligned} V_0(X_0) \quad &\leq \quad h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \; \Lambda_j(a)\{f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\}\Big] \\ &\leq \quad h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \; \Lambda_j(a)\{\; f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\;\}^+\Big] \end{aligned}$$

Taking infimum over the functions h_j, we get

$$V_0(X_0) \leq \inf_{(h_j)}\Big\{h_0(X_0)+E^*\big[\sum_{j=0}^{T} \sup_a \; \Lambda_j(a)\{\; f_j(X_j,a_j)+Ph_{j+1}(X_j,a_j)-h_j(X_j)\;\}^+\,\big]\Big\}$$

In fact, there is equality, and the equality is attained !!

Main result, 2.

So

$$\begin{aligned} V_0(X_0) &\leq h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \ \Lambda_j(a)\{f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\}\Big] \\ &\leq h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \ \Lambda_j(a)\{\ f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\ \}^+\ \Big] \end{aligned}$$

Taking infimum over the functions h_j, we get

$$V_0(X_0) \leq \inf_{(h_j)} \Big\{ h_0(X_0) + E^*\big[\sum_{j=0}^{T} \sup_a \ \Lambda_j(a)\{\ f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\ \}^+\ \big] \Big\}$$

In fact, there is equality, and the equality is attained !!

To see this, use the Bellman equation:

$$\begin{aligned} V_j(x) &= \sup_a E\big[\ f_j(x,a) + V_{j+1}(X_{j+1})\ \big|\ X_j = x, a_j = a\ \big] \\ &= \sup_a \{\ f_j(x,a) + PV_{j+1}(x,a)\ \} \\ &\geq f_j(x,a) + PV_{j+1}(x,a) \end{aligned}$$

for any a.

Main result, 2.

So

$$\begin{aligned} V_0(X_0) &\le h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \ \Lambda_j(a)\{f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\} \Big] \\ &\le h_0(X_0) + E^*\Big[\sum_{j=0}^{T} \sup_a \ \Lambda_j(a)\{\ f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\ \}^+ \Big] \end{aligned}$$

Taking infimum over the functions h_j, we get

$$V_0(X_0) \le \inf_{(h_j)} \Big\{ h_0(X_0) + E^* \big[\sum_{j=0}^{T} \sup_a \ \Lambda_j(a)\{\ f_j(X_j,a_j) + Ph_{j+1}(X_j,a_j) - h_j(X_j)\ \}^+ \ \big] \Big\}$$

In fact, there is equality, and the equality is attained !!

To see this, use the Bellman equation:

$$\begin{aligned} V_j(x) &= \sup_a E\big[\ f_j(x,a) + V_{j+1}(X_{j+1}) \ \big|\ X_j = x, a_j = a \ \big] \\ &= \sup_a \{\ f_j(x,a) + PV_{j+1}(x,a)\ \} \\ &\ge f_j(x,a) + PV_{j+1}(x,a) \end{aligned}$$

for any a. *So use $h_j = V_j$.*

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^*\Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a)\{f_j(X_j, a_j) + Ph_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1})\varphi_{j+1}(a_j) \} \Big].$$

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^* \Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + P h_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1}) \varphi_{j+1}(a_j) \} \Big].$$

Remarks.

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^* \Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a)\{f_j(X_j, a_j) + Ph_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1})\varphi_{j+1}(a_j) \} \Big].$$

Remarks.

- Well suited to Monte Carlo; simulate X, and then maximise *pathwise* over a;

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^* \Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + P h_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1}) \varphi_{j+1}(a_j) \} \Big].$$

Remarks.

- Well suited to Monte Carlo; simulate X, and then maximise *pathwise* over a;
- The pathwise optimisation can be done recursively;

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^* \Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + P h_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1}) \varphi_{j+1}(a_j) \} \Big].$$

Remarks.

- Well suited to Monte Carlo; simulate X, and then maximise *pathwise* over a;
- The pathwise optimisation can be done recursively;
- There is an infinite-horizon version of the result.

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^*\left[\sup_a \sum_{j=0}^{T} \Lambda_j(a)\{f_j(X_j, a_j) + Ph_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1})\varphi_{j+1}(a_j)\ \} \right].$$

Remarks.

- Well suited to Monte Carlo; simulate X, and then maximise *pathwise* over a;
- The pathwise optimisation can be done recursively;
- There is an infinite-horizon version of the result.
- Rockafellar & Wets, Wets, Back & Pliska study maximisation of a concave path functional over adapted processes by absorbing adaptedness constraint into a Lagrangian term and then doing pathwise max ...

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^* \Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a)\{f_j(X_j, a_j) + Ph_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1})\varphi_{j+1}(a_j)\ \} \Big].$$

Remarks.

- Well suited to Monte Carlo; simulate X, and then maximise *pathwise* over a;
- The pathwise optimisation can be done recursively;
- There is an infinite-horizon version of the result.
- Rockafellar & Wets, Wets, Back & Pliska study maximisation of a concave path functional over adapted processes by absorbing adaptedness constraint into a Lagrangian term and then doing pathwise max ...

... but what we do here requires no concavity assumption;

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^* \Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a) \{ f_j(X_j, a_j) + P h_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1}) \varphi_{j+1}(a_j) \} \Big].$$

Remarks.

- Well suited to Monte Carlo; simulate X, and then maximise *pathwise* over a;
- The pathwise optimisation can be done recursively;
- There is an infinite-horizon version of the result.
- Rockafellar & Wets, Wets, Back & Pliska study maximisation of a concave path functional over adapted processes by absorbing adaptedness constraint into a Lagrangian term and then doing pathwise max ...

... but what we do here requires no concavity assumption;

- This approach is a new *strategy* for optimal control.

Main result: remarks.

$$V_0(X_0) = \min_{(h_j)} E^* \Big[\sup_a \sum_{j=0}^{T} \Lambda_j(a)\{f_j(X_j, a_j) + Ph_{j+1}(X_j, a_j) - h_{j+1}(X_{j+1})\varphi_{j+1}(a_j)\ \} \Big].$$

Remarks.

- Well suited to Monte Carlo; simulate X, and then maximise *pathwise* over a;
- The pathwise optimisation can be done recursively;
- There is an infinite-horizon version of the result.
- Rockafellar & Wets, Wets, Back & Pliska study maximisation of a concave path functional over adapted processes by absorbing adaptedness constraint into a Lagrangian term and then doing pathwise max ...

... but what we do here requires no concavity assumption;

- This approach is a new *strategy* for optimal control.

- In fact, we have an infinite-dimensional linear program, where the choice variable is the RCD for a given X.

How might we use this?

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$.

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$. Solution pushes out from the origin, so we really only need to know what is happening in a neighbourhood of the ray $\{\lambda x_0 : \lambda > 0\}$.

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$. Solution pushes out from the origin, so we really only need to know what is happening in a neighbourhood of the ray $\{\lambda x_0 : \lambda > 0\}$.

- Important to simulate paths which are 'like' optimal paths ...

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$. Solution pushes out from the origin, so we really only need to know what is happening in a neighbourhood of the ray $\{\lambda x_0 : \lambda > 0\}$.

- Important to simulate paths which are 'like' optimal paths ... so don't try to look too far ahead.

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$. Solution pushes out from the origin, so we really only need to know what is happening in a neighbourhood of the ray $\{\lambda x_0 : \lambda > 0\}$.

- Important to simulate paths which are 'like' optimal paths ... so don't try to look too far ahead. If

$$dX_t = dW_t - a_t dt,$$

with objective $\min E(|X_T|^2 + \gamma \int_0^T |a_s|^2 \, ds)$, the solution is a linear feedback; the solution is (nearly) an OU process, very different from the Brownian motion we might use to simulate paths ..

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$. Solution pushes out from the origin, so we really only need to know what is happening in a neighbourhood of the ray $\{\lambda x_0 : \lambda > 0\}$.

- Important to simulate paths which are 'like' optimal paths ... so don't try to look too far ahead. If

$$dX_t = dW_t - a_t dt,$$

with objective $\min E(|X_T|^2 + \gamma \int_0^T |a_s|^2 \, ds)$, the solution is a linear feedback; the solution is (nearly) an OU process, very different from the Brownian motion we might use to simulate paths ..

- Shorten the horizon, but think ahead!

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$. Solution pushes out from the origin, so we really only need to know what is happening in a neighbourhood of the ray $\{\lambda x_0 : \lambda > 0\}$.

- Important to simulate paths which are 'like' optimal paths ... so don't try to look too far ahead. If

$$dX_t = dW_t - a_t dt,$$

with objective $\min E(|X_T|^2 + \gamma \int_0^T |a_s|^2 \, ds)$, the solution is a linear feedback; the solution is (nearly) an OU process, very different from the Brownian motion we might use to simulate paths ..

- Shorten the horizon, but think ahead! If we simply reduce T to $T_1 < T$, and use objective $\min E\gamma \int_0^{T_1} |a_s|^2 \, ds$, then we will get things very wrong!

How might we use this?

- Dynamic programming requires the solution from *every* starting point; this approach only cares about the given x_0.

$$dX_t = dW_t + a_t dt, \quad X_0 = x_0 \neq 0,$$

aiming to $\max E|X_T|^2$ subject to $|a_t| \leq 1$. Solution pushes out from the origin, so we really only need to know what is happening in a neighbourhood of the ray $\{\lambda x_0 : \lambda > 0\}$.

- Important to simulate paths which are 'like' optimal paths ... so don't try to look too far ahead. If

$$dX_t = dW_t - a_t dt,$$

with objective $\min E(|X_T|^2 + \gamma \int_0^T |a_s|^2 \, ds)$, the solution is a linear feedback; the solution is (nearly) an OU process, very different from the Brownian motion we might use to simulate paths ..

- Shorten the horizon, but think ahead! If we simply reduce T to $T_1 < T$, and use objective $\min E\gamma \int_0^{T_1} |a_s|^2 \, ds$, then we will get things very wrong! Include something to account (even approximately) for significant upcoming costs.

Implementation outline.

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions ‘nice’ in the sense that $P\psi(x, a; \theta)$ is simply expressed.

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions 'nice' in the sense that $P\psi(x, a; \theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions 'nice' in the sense that $P\psi(x,a;\theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Step 1: $n = 0$; select law $P^{(0)}$, and approximate value functions $V_t^{(0)} = h_t$;

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions `nice' in the sense that $P\psi(x, a; \theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Step 1: $n = 0$; select law $P^{(0)}$, and approximate value functions $V_t^{(0)} = h_t$;

Step 2: Simulate N paths $(X_t^i)_{t=0}^T$, $i = 1, \dots N$, according to $P^{(n)}$;

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions 'nice' in the sense that $P\psi(x, a; \theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Step 1: $n = 0$; select law $P^{(0)}$, and approximate value functions $V_t^{(0)} = h_t$;

Step 2: Simulate N paths $(X_t^i)_{t=0}^T$, $i = 1, \dots N$, according to $P^{(n)}$;

Step 3: Recursively compute

$$V_t^{(n+1)}(X_t^i) \equiv \sup_a \sum_{j=t}^T \Lambda_j(a)\{f_j(X_j^i, a_j) + Ph_{j+1}(X_j^i, a_j) - h_{j+1}(X_{j+1}^i)\varphi_{j+1}(a_j)\,\}/\Lambda_t(a),$$

with a_t^i the optimal action at time t on path i;

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions 'nice' in the sense that $P\psi(x, a; \theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Step 1: $n = 0$; select law $P^{(0)}$, and approximate value functions $V_t^{(0)} = h_t$;

Step 2: Simulate N paths $(X_t^i)_{t=0}^T$, $i = 1, \ldots N$, according to $P^{(n)}$;

Step 3: Recursively compute

$$V_t^{(n+1)}(X_t^i) \equiv \sup_a \sum_{j=t}^{T} \Lambda_j(a)\{f_j(X_j^i, a_j) + Ph_{j+1}(X_j^i, a_j) - h_{j+1}(X_{j+1}^i)\varphi_{j+1}(a_j)\}/\Lambda_t(a),$$

with a_t^i the optimal action at time t on path i;

Step 4: Regress $V_t^{(n+1)}$ onto the basis functions, to form (modified) $V_t^{(n+1)}$;

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions 'nice' in the sense that $P\psi(x, a; \theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Step 1: $n = 0$; select law $P^{(0)}$, and approximate value functions $V_t^{(0)} = h_t$;

Step 2: Simulate N paths $(X_t^i)_{t=0}^T$, $i = 1, \dots N$, according to $P^{(n)}$;

Step 3: Recursively compute

$$V_t^{(n+1)}(X_t^i) \equiv \sup_a \sum_{j=t}^{T} \Lambda_j(a)\{f_j(X_j^i, a_j) + Ph_{j+1}(X_j^i, a_j) - h_{j+1}(X_{j+1}^i)\varphi_{j+1}(a_j)\}/\Lambda_t(a),$$

with a_t^i the optimal action at time t on path i;

Step 4: Regress $V_t^{(n+1)}$ onto the basis functions, to form (modified) $V_t^{(n+1)}$;

Step 5: extend $(a_t^i)_{i=1}^N$ to a function $a_t^{(n+1)}(\cdot)$ by choosing action a_t^i when in x with probability proportional to $g(d(x, X_t^i))$ (some g decreasing);

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions 'nice' in the sense that $P\psi(x,a;\theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Step 1: $n = 0$; select law $P^{(0)}$, and approximate value functions $V_t^{(0)} = h_t$;

Step 2: Simulate N paths $(X_t^i)_{t=0}^T$, $i = 1, \dots N$, according to $P^{(n)}$;

Step 3: Recursively compute

$$V_t^{(n+1)}(X_t^i) \equiv \sup_a \sum_{j=t}^{T} \Lambda_j(a)\{f_j(X_j^i, a_j) + Ph_{j+1}(X_j^i, a_j) - h_{j+1}(X_{j+1}^i)\varphi_{j+1}(a_j)\ \}/\Lambda_t(a),$$

with a_t^i the optimal action at time t on path i;

Step 4: Regress $V_t^{(n+1)}$ onto the basis functions, to form (modified) $V_t^{(n+1)}$;

Step 5: extend $(a_t^i)_{i=1}^N$ to a function $a_t^{(n+1)}(\cdot)$ by choosing action a_t^i when in x with probability proportional to $g(d(x, X_t^i))$ (some g decreasing);

Step 6: Set $P^{(n+1)}$ to be the law of the process controlled according to $a_t^{(n+1)}$;

Implementation outline.

Probably very difficult unless there is some rich enough family $\{\psi(\,\cdot\,;\theta) : \theta \in \Theta\}$ of functions 'nice' in the sense that $P\psi(x, a; \theta)$ is simply expressed. If such exists, we will only try (h_j) which are linear combinations of $\psi(\,\cdot\,;\theta)$.

Step 1: $n = 0$; select law $P^{(0)}$, and approximate value functions $V_t^{(0)} = h_t$;

Step 2: Simulate N paths $(X_t^i)_{t=0}^T$, $i = 1, \dots N$, according to $P^{(n)}$;

Step 3: Recursively compute

$$V_t^{(n+1)}(X_t^i) \equiv \sup_a \sum_{j=t}^{T} \Lambda_j(a)\{f_j(X_j^i, a_j) + Ph_{j+1}(X_j^i, a_j) - h_{j+1}(X_{j+1}^i)\varphi_{j+1}(a_j)\,\}/\Lambda_t(a),$$

with a_t^i the optimal action at time t on path i;

Step 4: Regress $V_t^{(n+1)}$ onto the basis functions, to form (modified) $V_t^{(n+1)}$;

Step 5: extend $(a_t^i)_{i=1}^N$ to a function $a_t^{(n+1)}(\cdot)$ by choosing action a_t^i when in x with probability proportional to $g(d(x, X_t^i))$ (some g decreasing);

Step 6: Set $P^{(n+1)}$ to be the law of the process controlled according to $a_t^{(n+1)}$;

Step 7: Increase n by 1 and return to Step 2.

Conclusions

Conclusions

- Genuinely new strategy for dynamic optimal control

Conclusions

- Genuinely new strategy for dynamic optimal control
- Works well with Monte Carlo

Conclusions

- Genuinely new strategy for dynamic optimal control
- Works well with Monte Carlo
- Particularly useful in high dimensions

Conclusions

- Genuinely new strategy for dynamic optimal control
- Works well with Monte Carlo
- Particularly useful in high dimensions
- Numerical examples of optimal stopping are promising

Conclusions

- Genuinely new strategy for dynamic optimal control
- Works well with Monte Carlo
- Particularly useful in high dimensions
- Numerical examples of optimal stopping are promising
- More experience required of wider range of applications