A discrete time DP approach on a tree structure for finite horizon optimal control problems

Maurizio Falcone

joint works with A. Alla (PUC, Rio) and L. Saluzzi (GSSI, L'Aquila)

Workshop "High Dimensional Hamilton-Jacobi PDEs"

IPAM, Los Angeles, March 31, 2020

Outline

1. Introduction
   - Dynamic programming principle and HJB equations
   - Classical Semi-Lagrangian approach
2. Tree Structure Algorithm
   - Algorithm
   - A priori error estimates
3. Numerical tests

Outline

1. Introduction
   - Dynamic programming principle and HJB equations
   - Classical Semi-Lagrangian approach
2. Tree Structure Algorithm
   - Algorithm
   - A priori error estimates
3. Numerical tests

Motivation

Feedback controls

The numerical solution of optimal control problems via the Dynamic Programming (DP) approach is mainly motivated by the search for feedback controls for general nonlinear optimal control problems with Lipschitz continuous dynamics and costs.

Motivation

Feedback controls

The numerical solution of optimal control problems via the Dynamic Programming (DP) approach is mainly motivated by the search for feedback controls for general nonlinear optimal control problems with Lipschitz continuous dynamics and costs.

Wide range of applications

This approach has been developed in the last 30 years in the framework of viscosity solutions for Hamilton-Jacobi equations for all the classical deterministic and stochastic optimal control problems. A similar approach has been recently applied also to Mean Field games.

Motivation

High computational cost and complexity

For all these problems the solution of the corresponding Bellman equation in high dimension is a computationally intensive task and this bottleneck has limited the applications of this theory to industrial cases, despite the many theoretical results available in any dimension and the numerical schemes that have been developed so far.

Motivation

High computational cost and complexity

For all these problems the solution of the corresponding Bellman equation in high dimension is a computationally intensive task and this bottleneck has limited the applications of this theory to industrial cases, despite the many theoretical results available in any dimension and the numerical schemes that have been developed so far.

Our goal

We want to reduce the computational costs still having a reliable approximation of feedback controls and trajectories.

Example 1: the deterministic infinite horizon problem

Controlled dynamical system

$$\begin{cases} \dot{y}(t) = f(y(t), u(t)) & t > 0, y \in \mathbb{R}^d \\ y(0) = x & x \in \mathbb{R}^d \end{cases}$$

Example 1: the deterministic infinite horizon problem

Controlled dynamical system

$$\begin{cases} \dot{y}(t) = f(y(t), u(t)) & t > 0, y \in \mathbb{R}^d \\ y(0) = x & x \in \mathbb{R}^d \end{cases}$$

Cost Functional

$$J(x, u) := \int_0^\infty L(y_x(t, u), u(t)) e^{-\lambda t} dt.$$

We are interested in minimizing this cost functional, so we introduce

Example 1: the deterministic infinite horizon problem

Controlled dynamical system

$$\begin{cases} \dot{y}(t) = f(y(t), u(t)) & t > 0, y \in \mathbb{R}^d \\ y(0) = x & x \in \mathbb{R}^d \end{cases}$$

Cost Functional

$$J(x, u) := \int_0^\infty L(y_x(t, u), u(t)) e^{-\lambda t} dt.$$

We are interested in minimizing this cost functional, so we introduce

Value function

$$v(x) := \inf_{u \in \mathcal{U}} J(x, u),$$

Infinite horizon problem for deterministic dynamics

Assumptions

- $u(\cdot) \in \mathcal{U}$: control, where

$$\mathcal{U} := \{u(\cdot) : [0, +\infty[\to U \text{ measurable}\}$$

- $f : \mathbb{R}^d \times U \to \mathbb{R}^d$ is the dynamic, which satisfies:
  - f is continuous with respect to (y, u)
  - f is locally bounded
  - f is Lipschitz continuous with respect to y

Carathèodory Theorem guarantees the existence and uniqueness of the trajectory $y_x(t, u)$ for every $u \in \mathcal{U}$

Example 2: Minimum time problem for SDE

Controlled Stochastic Differential Equation

$$(CSDE)\begin{cases} dY(t) = f(Y(t),\alpha(t))dt + \sigma(Y(t),\alpha(t))dW(t), & t > 0 \\ Y(0) = x \in \Omega \end{cases}$$

$\Omega \subset \mathbb{R}^d$, $Y(\cdot) \in \Omega$ is the state process, $u(\cdot) \in \mathcal{U}$ is the control process, with

$$\mathcal{U} = \{\alpha(\cdot) : [0,+\infty) \to U, \text{progressively measurable}\}$$

$U \subset \mathbb{R}^m$ is a compact set of admissible controls
$f : \Omega \times U \to \mathbb{R}^d$ is the dynamics
$\sigma : \Omega \times A \to \mathcal{L}(\mathbb{R}^k;\mathbb{R}^k)$ is the diffusion ($k \leq d$)
W is a k-dimensional Wiener process

Denote by $Y_x(t;\alpha(\cdot))$ the solution of $(CSDE)$ and define $t_x(u(\cdot))$ as the first time of arrival on $\partial\Omega$ starting at x using the strategy $u(\cdot)$:

$$t_x(u(\cdot)) = \inf\{t \geq 0 \ : \ Y_x(t;u(\cdot)) \in \partial\Omega\}$$

Second order Hamilton-Jacobi-Bellman equation

Stochastic optimal control problem

Given a running cost $\ell : \Omega \times U \to \mathbb{R}^+$ and a final cost $g : \partial\Omega \to \mathbb{R}^+$ find an optimal control $u^*(\cdot)$ minimizing

$$J\big(x, u(\cdot)\big) = \mathbb{E}\left\{ \int_0^{t_x(u(\cdot))} L\big(Y_x(s; u(\cdot)), u(s)\big)\, ds + g\big(Y_x(t_x(u(\cdot)); u(\cdot))\big) \right\}$$

among all the trajectories starting from x.

Second order Hamilton-Jacobi-Bellman equation

Stochastic optimal control problem

Given a running cost $\ell : \Omega \times U \to \mathbb{R}^+$ and a final cost $g : \partial\Omega \to \mathbb{R}^+$ find an optimal control $u^*(\cdot)$ minimizing

$$J\big(x, u(\cdot)\big) = \mathbb{E}\left\{ \int_0^{t_x(u(\cdot))} L\big(Y_x(s; u(\cdot)), u(s)\big)\, ds + g\big(Y_x(t_x(u(\cdot)); u(\cdot))\big) \right\}$$

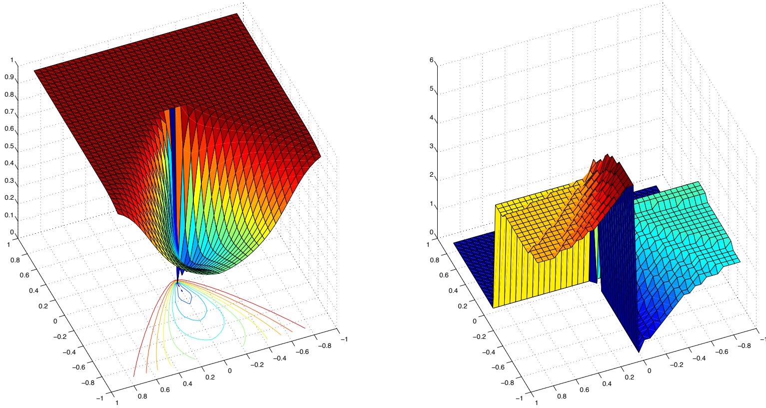
among all the trajectories starting from x.

2nd order Hamilton-Jacobi-Bellman equation

The value function $v(x) = \inf_{u(\cdot)\in\mathcal{U}} J(x, u(\cdot))$ is the viscosity solution of

$$\begin{cases} \max_{u\in U}\left\{ -\frac{1}{2} < \sigma(x,a)\sigma^T(x,a) : D^2u(x) > -f(x,a)\cdot\nabla u(x) - L(x,a) \right\} = 0 & x \in \Omega \\ v(x) = g(x) & x \in \partial\Omega \end{cases}$$

Value and feedback for Zermelo navigation problem



Value   Feedback

Our problem: the finite horizon problem

We focus on the following problem

Controlled Dynamics and Cost Functional

$$\begin{cases} \dot{y}(s,u) = f(y(s),u(s),s) & s \in (t,T] \\ y(t) = x \end{cases}$$

$$u(t) \in \mathcal{U} = \{u : [t,T] \to U \subset \mathbb{R}^m \, compact, measurable\},$$

$$J_{x,t}(u) = \int_t^T L(y(s,u),u(s),s)e^{-\lambda(s-t)}\,ds + g(y(T))e^{-\lambda(T-t)}$$

Our problem: the finite horizon problem

We focus on the following problem

Controlled Dynamics and Cost Functional

$$\begin{cases} \dot{y}(s,u) = f(y(s),u(s),s) & s \in (t,T] \\ y(t) = x \end{cases}$$

$$u(t) \in \mathcal{U} = \{u : [t,T] \to U \subset \mathbb{R}^m \, \textit{compact}, \textit{measurable}\},$$

$$J_{x,t}(u) = \int_t^T L(y(s,u),u(s),s)e^{-\lambda(s-t)}\,ds + g(y(T))e^{-\lambda(T-t)}$$

Value Function

$$v(x,t) := \inf_{u(\cdot)\in\mathcal{U}} J_{x,t}(u)$$

HJB equation for the finite horizon problem

Dynamic Programming Principle

$$v(x,t) = \min_{u \in \mathcal{U}} \left\{ \int_t^\tau L(y(s), u(s), s) e^{-\lambda(s-t)}\, ds + v(y(\tau), \tau) e^{-\lambda(\tau - t)} \right\}$$

HJB equation for the finite horizon problem

Dynamic Programming Principle

$$v(x,t) = \min_{u \in \mathcal{U}} \left\{ \int_t^\tau L(y(s), u(s), s) e^{-\lambda(s-t)} \, ds + v(y(\tau), \tau) e^{-\lambda(\tau - t)} \right\}$$

HJB equation

$$\begin{cases} -\dfrac{\partial v}{\partial t}(x,t) + \lambda v(x,t) = \min\limits_{u \in U} \{ L(x,u,t) + \nabla v(x,t) \cdot f(x,u,t) \} \\ v(x,T) = g(x) \ , x \in \mathbb{R}^d \end{cases}$$

HJB equation for the finite horizon problem

Dynamic Programming Principle

$$v(x,t) = \min_{u \in \mathcal{U}} \left\{ \int_t^{\tau} L(y(s), u(s), s) e^{-\lambda(s-t)}\, ds + v(y(\tau), \tau) e^{-\lambda(\tau - t)} \right\}$$

HJB equation

$$\begin{cases} -\dfrac{\partial v}{\partial t}(x,t) + \lambda v(x,t) = \min\limits_{u \in U} \{ L(x,u,t) + \nabla v(x,t) \cdot f(x,u,t) \} \\ v(x,T) = g(x)\ , x \in \mathbb{R}^d \end{cases}$$

Optimal Feedback Map

$$u^*(x,t) = \arg\min_{u \in U} \{ L(x,u,t) + \nabla v(x,t) \cdot f(x,u,t) \}$$

Classical approach

Semi-Lagrangian scheme ($\lambda = 0$)

$$\begin{cases} V_i^{n-1} = \min\limits_{u \in U} [\Delta t \, L(x_i, u, t_n) + V^n(x_i + \Delta t \, f(x_i, u, t_n))], \;\; n = N, \dots, 1 \\ \\ V_i^N = g(x_i) \; , \qquad x_i \in \Omega^{\Delta x} \; . \end{cases}$$

Classical approach

Semi-Lagrangian scheme ($\lambda = 0$)

$$\begin{cases} V_i^{n-1} = \min\limits_{u \in U} [\Delta t\, L(x_i, u, t_n) + V^n(x_i + \Delta t\, f(x_i, u, t_n))], \;\; n = N, \dots, 1 \\ \\ V_i^N = g(x_i)\,, \qquad x_i \in \Omega^{\Delta x}\,. \end{cases}$$

Cons of the approach

- $V^n(x_i + \Delta t\, f(x_i, u, t_n))$ is computed by interpolation operator.
- We need a numerical domain (not always given in the problem)
- Selection of boundary conditions (not always given in the problem)
- The curse of dimensionality makes the problem difficult to solve in high dimension (need e.g. model order reduction).

Other approaches and acceleration techniques

Several methods have been developed to accelerate the computation and/or mitigate the curse of dimensionality

- Domain decomposition (static or dynamic): F.-Lanucara-Seghini (1994-...), Krener-Navasca (2007-...), Cacace-Cristiani-F.-Picarelli (2012)
- Iteration in policy space: Bellman (1957), Howard (1960), Bokanowski- Maroso-Zidani (2009), Alla-F.-Kalise (2015), Bokanowki–Desilles-Zidani (2018)
- Max-plus algebra and Galerkin approximation: Akian-Gaubert-Lakhoua (2008), McEneaney (2009-...), Dower (2017)

Other approaches and acceleration techniques

- Model Order Reduction: Kunisch-Volkwein-Xie (2004), Alla-F-Volkwein (2017)
- Sparse grids: Bokanowski-Garke-Griebel-Klompmaker (2013), Garke-Kroner (2016)
- Spectral Methods and Tensor Calculus: Kalise-Kundu-Kunisch (2019), Dolgov-Kalise-Kunisch (2019)
- Hopf formulas: Osher-Darbon (2016- ...), Yegorov-Dower-Grüne (2018)
- DNN/DGM: Pham-Warin (2019)

Outline

Tree Structure Algorithm (Alla, F. , Saluzzi '18)

We start with an initial condition $x \in \mathbb{R}^d$ forming the first level $\mathcal{T}^0$.

$$x$$

Tree Structure Algorithm (Alla, F. , Saluzzi ’18)

We start with an initial condition $x \in \mathbb{R}^d$ forming the first level $\mathcal{T}^0$.

$$\textcircled{x}$$

Discretization: constant Δt for time and N_u discrete controls.

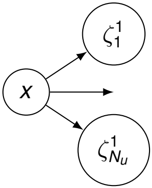
Tree Structure Algorithm (Alla, F. , Saluzzi ’18)

We start with an initial condition $x \in \mathbb{R}^d$ forming the first level $\mathcal{T}^0$.

Discretization: constant Δt for time and N_u discrete controls.

Starting with x, we follow the dynamics given by the discrete controls

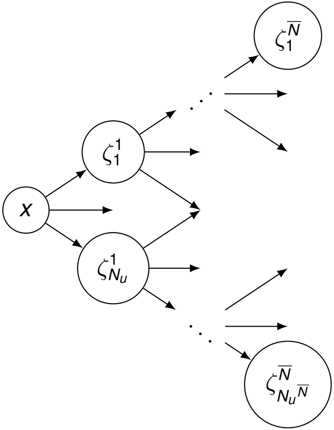
$$\mathcal{T}^1 = \{\zeta_i^1\}_i = \{x + \Delta t\, f(x, u_i, t_0)\}_i \ , \quad i = 1, ..., N_u$$



Tree Structure Algorithm

Given the nodes in the previous level, we construct the following one

$$\mathcal{T}^n = \{\zeta_i^{n-1} + \Delta t\, f(\zeta_i^{n-1}, u_j, t_{n-1})\}_{j=1}^{N_u} \quad i = 1, \ldots, N_u^n.$$



Approximation of the value function

Computation of the value function on the tree

The tree structure defines $\mathcal{T} = \{\mathcal{T}^r\}_{r=0}^{\overline{N}}$, where we can compute the numerical value function:

$$\begin{cases} V^n(\zeta_i^n) = \min_{u \in U^{\Delta u}} \{ V^{n+1}(\zeta_i^n + \Delta t\, f(\zeta_i^n, u, t_n)) + \Delta t\, L(\zeta_i^n, u, t_n) \} & \zeta_i^n \in \mathcal{T}^n \\ V^{\overline{N}}(\zeta_i^{\overline{N}}) = g(\zeta_i^{\overline{N}}) & \zeta_i^{\overline{N}} \in \mathcal{T}^{\overline{N}} \end{cases}$$

Approximation of the value function

Computation of the value function on the tree

The tree structure defines $\mathcal{T} = \{\mathcal{T}^r\}_{r=0}^{\overline{N}}$, where we can compute the numerical value function:

$$\begin{cases} V^n(\zeta_i^n) = \min\limits_{u \in U^{\Delta u}} \{V^{n+1}(\zeta_i^n + \Delta t\, f(\zeta_i^n, u, t_n)) + \Delta t\, L(\zeta_i^n, u, t_n)\} & \zeta_i^n \in \mathcal{T}^n \\ V^{\overline{N}}(\zeta_i^{\overline{N}}) = g(\zeta_i^{\overline{N}}) & \zeta_i^{\overline{N}} \in \mathcal{T}^{\overline{N}} \end{cases}$$

Pros

- No need for interpolation since the nodes $x_i + \Delta t\, f(x_i, u, t_n)$ belong to the tree by construction.
- Mitigation of the curse of dimensionality (e.g. , $d \gg 10$).

Approximation of the value function

Computation of the value function on the tree

The tree structure defines $\mathcal{T} = \{\mathcal{T}^r\}_{r=0}^{\overline{N}}$, where we can compute the numerical value function:

$$\begin{cases} V^n(\zeta_i^n) = \min_{u \in U^{\Delta u}} \{V^{n+1}(\zeta_i^n + \Delta t\, f(\zeta_i^n, u, t_n)) + \Delta t\, L(\zeta_i^n, u, t_n)\} & \zeta_i^n \in \mathcal{T}^n \\ V^{\overline{N}}(\zeta_i^{\overline{N}}) = g(\zeta_i^{\overline{N}}) & \zeta_i^{\overline{N}} \in \mathcal{T}^{\overline{N}} \end{cases}$$

Pros

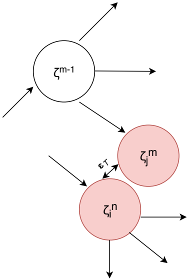
- No need for interpolation since the nodes $x_i + \Delta t\, f(x_i, u, t_n)$ belong to the tree by construction.
- Mitigation of the curse of dimensionality (e.g. , $d \gg 10$).

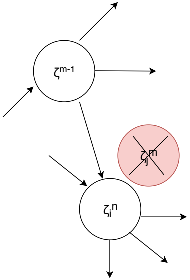
Cons

- Dimensionality problem. In fact, given N_u controls and $\overline{N}$ time steps, the cardinality of the tree is $O(N_u^{\overline{N}+1})$.

Solution: Pruning the tree

Solution: Pruning the tree





Pruning rule

Given a threshold $\varepsilon_{\mathcal{T}}$, two nodes ζ_i^n and ζ_j^n will be merged if

$$\|\zeta_i^n - \zeta_j^n\| \leq \varepsilon_{\mathcal{T}}$$

The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels

The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels

Pruning rule

Given a threshold $\varepsilon_{\mathcal{T}}$, two nodes ζ_i^n and ζ_j^m will be merged if

$$\|\zeta_i^n - \zeta_j^m\| \leq \varepsilon_{\mathcal{T}}$$

The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels

Pruning rule

Given a threshold $\varepsilon_{\mathcal{T}}$, two nodes ζ_i^n and ζ_j^m will be merged if

$$\|\zeta_i^n - \zeta_j^m\| \leq \varepsilon_{\mathcal{T}}$$

Computation of the value function on the tree

$$\begin{cases} V^n(\zeta) = \min\limits_{u \in U^{\Delta u}} \{ V^{n+1}(\zeta + \Delta t\, f(\zeta, u)) + \Delta t\, L(\zeta, u, t_n) \} & \zeta \in \cup_{k=0}^{n} \mathcal{T}^k \\ V^{\overline{N}}(\zeta) = g(\zeta) & \zeta \in \mathcal{T} \end{cases}$$

The case of an autonomous dynamics

The pruning rule and the computation of value function can be simplified, since we can extend the computation to the all previous tree levels

Pruning rule

Given a threshold $\varepsilon_{\mathcal{T}}$, two nodes ζ_i^n and ζ_j^m will be merged if

$$\|\zeta_i^n - \zeta_j^m\| \leq \varepsilon_{\mathcal{T}}$$

Computation of the value function on the tree

$$\begin{cases} V^n(\zeta) = \min\limits_{u \in U^{\Delta u}} \{V^{n+1}(\zeta + \Delta t\, f(\zeta, u)) + \Delta t\, L(\zeta, u, t_n)\} & \zeta \in \cup_{k=0}^n \mathcal{T}^k \\ V^{\overline{N}}(\zeta) = g(\zeta) & \zeta \in \mathcal{T} \end{cases}$$

Important reduction of the cardinality, we can get more information on V and this can be useful for the feedback reconstruction.

Efficient pruning

Problem

The computation of the distances among all the nodes would be very expensive, especially for high dimensional problems.

Efficient pruning

Problem

The computation of the distances among all the nodes would be very expensive, especially for high dimensional problems.

One possible solution

We project the data onto a lower dimensional linear space such that the variance of the projected data is maximized. This can be done e.g. computing the Singular Value Decomposition of the data matrix and taking the first basis.

Efficient pruning

Problem

The computation of the distances among all the nodes would be very expensive, especially for high dimensional problems.

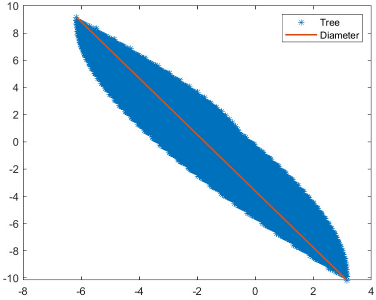
One possible solution

We project the data onto a lower dimensional linear space such that the variance of the projected data is maximized. This can be done e.g. computing the Singular Value Decomposition of the data matrix and taking the first basis.

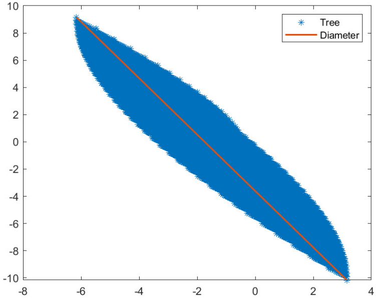
Reduced dynamics

The control problem can be solved in a reduced space, projecting the dynamics via Proper Orthogonal Decomposition.

Efficient pruning II

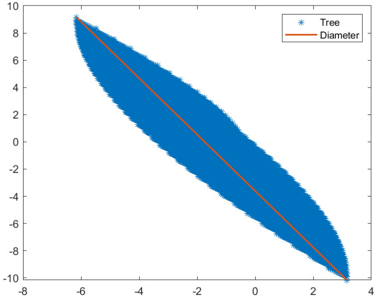


Efficient pruning II



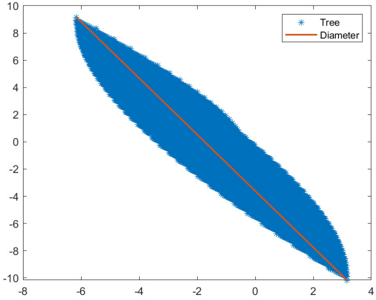
1. Construction of a rough full tree

Efficient pruning II



1. Construction of a rough full tree
2. Computation of the maximum variance direction and its subdivision in buckets of length equal to the tolerance.

Efficient pruning II



1. Construction of a rough full tree
2. Computation of the maximum variance direction and its subdivision in buckets of length equal to the tolerance.
3. Construction of the pruned tree comparing the nodes in the same bucket.

Error estimates for the approximate value V

Theorem (F.-Giorgi, '99)

Let f, L and g be Lipschitz continuous and bounded, then

$$\sup_{(x,t)\in\mathbb{R}^d\times[0,T]} |v(t,x) - V(t,x)| \leq C(T)\sqrt{\Delta t}\,.$$

Error estimates for the approximate value V

Theorem (F.-Giorgi, '99)

Let f, L and g be Lipschitz continuous and bounded, then

$$\sup_{(x,t)\in\mathbb{R}^d\times[0,T]} |v(t,x) - V(t,x)| \leq C(T)\sqrt{\Delta t}\,.$$

Theorem (Error estimate: first part)

Let f, L and g be Lipschitz continuous and bounded, then

$$\sup_{(x,t)\in\mathbb{R}^d\times[0,T]} (v(t,x) - V(t,x)) \leq C(T)\Delta t\,.$$

Error estimates for the approximate value V

Theorem (F.-Giorgi, '99)

Let f, L and g be Lipschitz continuous and bounded, then

$$\sup_{(x,t)\in\mathbb{R}^d\times[0,T]} |v(t,x) - V(t,x)| \le C(T)\sqrt{\Delta t}\ .$$

Theorem (Error estimate: first part)

Let f, L and g be Lipschitz continuous and bounded, then

$$\sup_{(x,t)\in\mathbb{R}^d\times[0,T]} (v(t,x) - V(t,x)) \le C(T)\Delta t\ .$$

The opposite inequality is based on the semiconcavity of the approximation V, *i.e.*

$$V(x+z,t+s) - 2V(x,s) + V(x-z,t-s) \le C(|z|^2 + s^2)\ .$$

Error estimates for the approximate value V

Proposition

Let f, L and g be Lipschitz continuous, bounded. Moreover let L and g be semiconcave and $f \in C^1$. Then the approximate solution V is semiconcave.

Error estimates for the approximate value V

Proposition

Let f, L and g be Lipschitz continuous, bounded. Moreover let L and g be semiconcave and $f \in C^1$. Then the approximate solution V is semiconcave.

Lemma (Capuzzo-Ishii, '84)

Let ξ be semiconcave such that $\xi(0,0) = 0$ and $\limsup_{(x,t)\to(0,0)} \frac{\xi(x,t)}{|x|+|t|} \leq 0$, then

$$\xi(x,t) \leq \frac{C_\xi}{6}(|x|^2 + |t|^2) \quad \forall x \in \mathbb{R}^n, t \in [0,T].$$

Error estimates for the approximate value V

Proposition

Let f, L and g be Lipschitz continuous, bounded. Moreover let L and g be semiconcave and $f \in C^1$. Then the approximate solution V is semiconcave.

Lemma (Capuzzo-Ishii, '84)

Let ξ be semiconcave such that $\xi(0,0) = 0$ and $\limsup_{(x,t)\to(0,0)} \frac{\xi(x,t)}{|x|+|t|} \leq 0$, then

$$\xi(x,t) \leq \frac{C_\xi}{6}(|x|^2 + |t|^2) \quad \forall x \in \mathbb{R}^n, t \in [0,T].$$

Theorem (Error estimate: second part)

Under the above assumptions, the following estimate holds

$$\sup_{(x,t)\in\mathbb{R}^d\times[0,T]} (V(t,x) - v(t,x)) \leq C(T)\Delta t\ .$$

Error estimates with pruning

Let us define the *pruned trajectory:*

$$\eta_j^{n+1} = \eta^n + \Delta t\, f(\eta^n, u_j, t_n) + \mathcal{E}_{\varepsilon_{\mathcal{T}}}(\eta^n + \Delta t\, f(\eta^n, u_j, t_n), \{\eta_i^{n+1}\}_i),$$

where

$$\mathcal{E}_{\varepsilon_{\mathcal{T}}}(x, \{x_n\}_n) = \begin{cases} x_k - x & \text{if } \min\limits_n |x - x_n| = |x - x_k| \leq \varepsilon_{\mathcal{T}}, \\ 0 & \text{otherwise.} \end{cases}$$

Error estimates with pruning

Let us define the *pruned trajectory:*

$$\eta_j^{n+1} = \eta^n + \Delta t\, f(\eta^n, u_j, t_n) + \mathcal{E}_{\varepsilon_\mathcal{T}}(\eta^n + \Delta t\, f(\eta^n, u_j, t_n), \{\eta_i^{n+1}\}_i),$$

where

$$\mathcal{E}_{\varepsilon_\mathcal{T}}(x, \{x_n\}_n) = \begin{cases} x_k - x & \text{if } \min_n |x - x_n| = |x - x_k| \le \varepsilon_\mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition

Given the Euler approximation $\{y^n\}_n$ *and its perturbation* $\{\eta^n\}_n$ *, then*

$$|y^n - \eta^n| \le n\, \varepsilon_\mathcal{T} e^{L_f(t_n - t)}.$$

To guarantee first order convergence, the tolerance must be chosen such that

$$\varepsilon_\mathcal{T} \le C_{\varepsilon_\mathcal{T}} \Delta t^2.$$

Error estimates with pruning

Then we can define the *pruned* discrete cost functional and value function

$$J^{\Delta t,P}_{x,t_n}(u) = \Delta t \sum_{k=n}^{N-1} L(\eta^k, u, t_k) e^{-\lambda(t_k - s)} + g(\eta^{\overline{N}}) e^{-\lambda(t_N - s)},$$

$$V^P(x,t) := \inf_{u \in \mathcal{U}^\Delta} J^{\Delta t,P}_{x,t}(u)$$

Error estimates with pruning

Then we can define the *pruned* discrete cost functional and value function

$$J_{x,t_n}^{\Delta t,P}(u) = \Delta t \sum_{k=n}^{N-1} L(\eta^k, u, t_k) e^{-\lambda(t_k - s)} + g(\eta^{\overline{N}}) e^{-\lambda(t_N - s)},$$

$$V^P(x,t) := \inf_{u \in \mathcal{U}^\Delta} J_{x,t}^{\Delta t,P}(u)$$

Proposition

Choosing $\varepsilon_{\mathcal{T}} \leq C_{\varepsilon_{\mathcal{T}}} \Delta t^2$, *we have*

$$|V(x,t) - V^P(x,t)| \leq C(T)\Delta t,$$

and then

$$|v(x,t) - V^P(x,t)| \leq C(T)\Delta t.$$

Outline

Test 1: Comparison with exact solution

We consider the following dynamics

$$f(x,u) = \begin{pmatrix} u \\ x_1^2 \end{pmatrix},\ u \in U \equiv [-1,1].$$

where $x = (x_1, x_2) \in \mathbb{R}^2$, and the following cost functional:

$$J_{x,t}(u) = -x_2(T;u)\,.$$

We compare the approximations according to ℓ_2 relative error

$$\mathcal{E}_2(t_n) = \sqrt{\frac{\sum\limits_{x_i \in \mathcal{T}^n} |v(x_i,t_n) - V^n(x_i)|^2}{\sum\limits_{x_i \in \mathcal{T}^n} |v(x_i,t_n)|^2}}\,.$$

Test 1: Comparison with exact solution

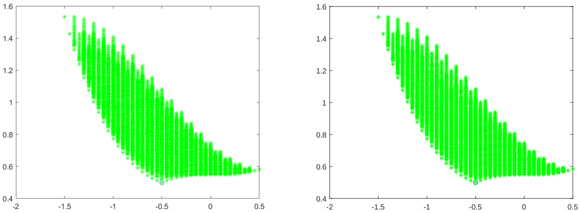


Figure: Full Tree ($|\mathcal{T}| = 2097151$) (left) and Pruned Tree with $\varepsilon_{\mathcal{T}} = \Delta t^2$ ($|\mathcal{T}| = 3151$) (right)

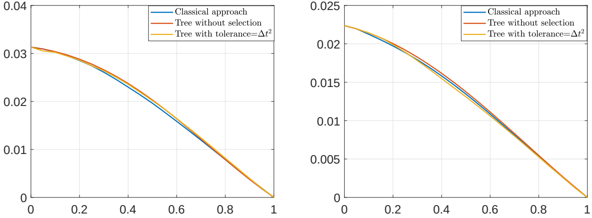


Figure: Error ℓ_2 with different initial conditions

Test 1: Comparison with exact solution

Δt	Nodes	CPU	$Err_{2,2}$	$Err_{\infty,2}$	$Order_{2,2}$	$Order_{\infty,2}$
0.2	63	0.05s	6.7e-02	0.18		
0.1	2047	0.35s	2.9e-02	0.09	1.16	0.98
0.05	2097151	1.1s	1.4e-02	0.05	1.08	0.99

Table: Table for Euler scheme for the Full Tree

Δt	Nodes	CPU	$Err_{2,2}$	$Err_{\infty,2}$	$Order_{2,2}$	$Order_{\infty,2}$
0.2	42	0.05s	9.1e-02	0.122		
0.1	324	0.08s	4.4e-02	0.062	1.05	0.98
0.05	3151	0.6s	2.1e-02	0.031	1.04	0.99
0.025	29248	2.5s	1.1e-02	0.016	1.005	0.994
0.0125	252620	150s	5.3e-03	0.008	1.004	0.997

Table: Table for Euler scheme with $\varepsilon_{\mathcal{T}} = \Delta t^2$

Test 1: Comparison with exact solution

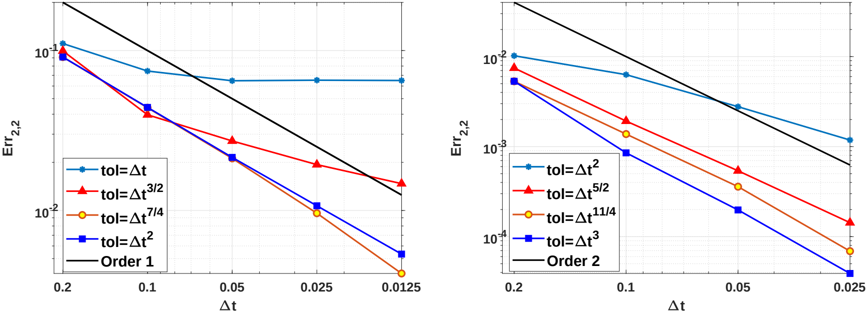


Figure: Comparison of the order of convergence for the pruned TSA with different tolerances (left) with Euler method and (right) with Heun's method.

Test 2: Heat Equation

We deal with the control of the heat equation with Dirichlet boundary conditions.
This test is unfeasible via a direct semi-Lagrangian approach.

Dynamics

$$
\begin{cases}
y_t = \sigma y_{xx} + y_0(x)u(t) & (x,t) \in (0,1) \times (0,T), \\
y(0,t) = y(1,t) = 0 & t \in (0,T), \\
y(x,0) = y_0(x) & x \in [0,1],
\end{cases}
$$

Semi-discretization in space

We set $T = 1$, $\sigma = 0.15$ and $y_0(x) = x - x^2$. and we apply a centered finite difference method in space getting dynamics

$$\begin{cases} \dot{y}(t) & = Ay(t) + Bu(t), \\ y(0) & = y_0 \end{cases}$$

where $A \in \mathbb{R}^{d \times d}$ is the stiffness matrix and $B \in \mathbb{R}^d$ is given by $B_i = y_0(x_i)$ for $i = 1, \dots, d$, x_i are the nodes.
We want to minimize the cost functional

$$J_{y_0,t}(u) = \int_t^T \left(\|y(s)\|_2^2 + \frac{1}{100}|u(s)|^2 \right) ds + \|y(T)\|_2^2.$$

Comparison with the exact solution (Riccati)

When the control is unconstrained, we can derive an exact solution solving the Riccati differential equation.
We compute the errors in L^2 and in L^∞

$$Err_2 := \frac{\sum_{n=0}^{N} |V(y_*^n, t_n) - v(y_R^n, t_n)|^2}{\sum_{n=0}^{N} |v(y_R^n, t_n)|^2}$$

$$Err_\infty := \frac{\max\limits_{n=0,\ldots,N} |V(y_*^n, t_n) - v(y_R^n, t_n)|}{\max\limits_{n=0,\ldots,N} |v(y_R^n, t_n)|}$$

where $\{y_*^n\}_n$ is the optimal trajectory computed via TSA and $\{y_R^n\}_n$ is obtained solving the Riccati equation.

TSA approximation

For $\Delta x = 10^{-2}$, we get a system of dimension $d = 100$.

Δt	Nodes	P/F ratio	CPU	Err_2	Err_∞	$Order_2$	$Order_\infty$
0.1	134	4.7e-09	0.14s	0.279	0.241		
0.05	863	1.2e-18	0.65s	0.144	0.118	0.95	1.03
0.025	15453	3.1e-38	12.88s	5.5e-2	5.3e-2	1.40	1.17
0.0125	849717	3.8e-78	1.1e+3s	1.6e-2	1.6e-2	1.77	1.42

Table: Test 2: Error analysis and order of convergence for forward Euler scheme of the TSA with $\varepsilon_{\mathcal{T}} = \Delta t^2$ and 11 discrete controls.

TSA with and without pruning

Δt	P/P ratio	F/F ratio
0.05	6.44	2.6e10
0.025	17.9	6.7e20
0.0125	984	4.5e41

Table: Test 2: Comparison between the ratio of cardinality for the full and the pruned tree for $\varepsilon_{\mathcal{T}} = \Delta t^2$ and 11 discrete controls.

TSA vs Riccati: 11 controls

We set $\Delta t = 10^{-4}$ for the Riccati equation to get an accurate solution. For a fair comparison, we first computed the LQR problem and then set the control space in the TSA, $U = [-1, 0]$

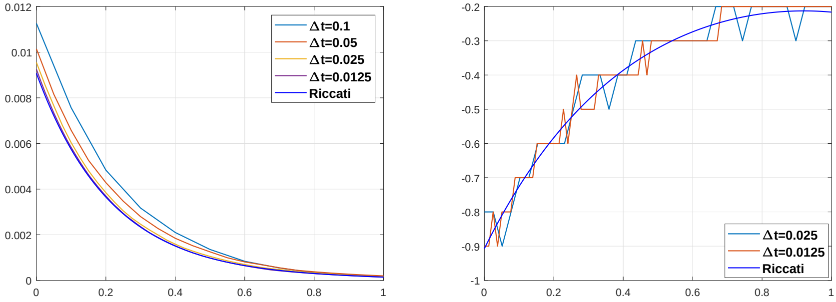


Figure: Test 2: Cost functional (left) and optimal control (right) with 11 discrete controls.

TSA vs Riccati: 100 controls

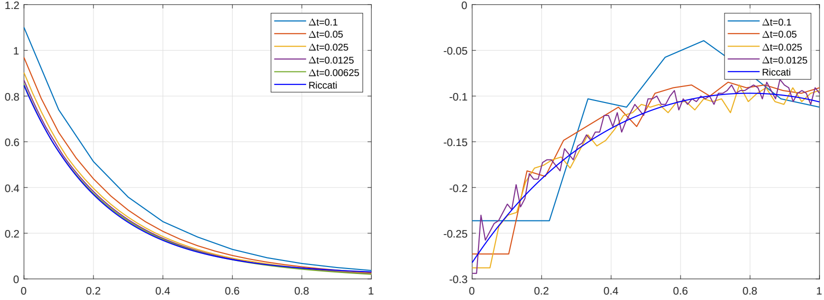


Figure: Test 2: Cost functional (left) and optimal control (right) with 100 discrete controls.

Conclusions and future works

Conclusions

- We presented a new algorithm to solve finite horizon optimal control problems using a **tree structure** with first order convergence.
- The pruning rule will mitigate the "curse of dimension"
- It can be easily extended to high-order methods
- It can be applied to general non linear control problems over a finite horizon.
- We can couple this method with POD to obtain a more efficient algorithms (e.g. PDEs in 2 or 3 dimensions)

Conclusions and future works

Conclusions

- We presented a new algorithm to solve finite horizon optimal control problems using a **tree structure** with first order convergence.
- The pruning rule will mitigate the "curse of dimension"
- It can be easily extended to high-order methods
- It can be applied to general non linear control problems over a finite horizon.
- We can couple this method with POD to obtain a more efficient algorithms (e.g. PDEs in 2 or 3 dimensions)

Future works

- Extension to stochastic control problems
- Efficient Feedback reconstruction.
- Algorithm improvements for the pruning

Thank you for the attention

1. A. Alla, M. Falcone, L. Saluzzi, *An efficient DP algorithm on a tree-structure for finite horizon optimal control problems*, SIAM J. Sci. Comput., 2019
2. A. Alla, M. Falcone, L. Saluzzi, *High-order Approximation of the Finite Horizon Control Problem via a Tree Structure Algorithm*, to appear in IFAC CPDE 2019.
3. A. Alla, L. Saluzzi, *A HJB-POD approach for the control of nonlinear PDEs on a tree structure*, Appl. Num. Math., 2020
4. M. Bardi, I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhäuser, Basel, 1997.
5. I. Capuzzo Dolcetta, H. Ishii, *Approximate solution of the Bellman equation of deterministic control theory*, Appl. Math. Optim., 11 (1984), 161-181.
6. M. Falcone, R. Ferretti, *Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations*, Numerische Mathematik, **67**, 1994, 315-344.
7. M. Falcone, T. Giorgi, *An approximation scheme for evolutive Hamilton-Jacobi equations*, in W.M. McEneaney, G. Yin and Q. Zhang (eds.), "Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming", Birkhäuser, 1999, 289-303.
8. L. Saluzzi, A. Alla, M. Falcone, *Error estimates for a tree structure algorithm on dynamic programming equations*, submitted, 2019.