# Max-Plus Linear Approximations for Deterministic Continuous-State Markov Decision Processes

DAG Seminar

Eloïse Berthier, Francis Bach

*IEEE Control Systems Letters 2020*

July 2, 2021




# Motivation: Application to Robotics

Controlling a robot is challenging:

- The dimensions of the system are (relatively) large
  $\Longrightarrow$ completely solving optimal control problems is hopeless.
- The dynamical system is nonlinear
  $\Longrightarrow$ we cannot directly use linear control methods.
- There are modeling uncertainties
  $\Longrightarrow$ exact solutions are somehow useless.
- Some computations are done in real-time, embedded systems
  $\Longrightarrow$ the available computing power/time is limited.

# General Idea

Consider a **continuous-state MDP** (discrete-time, discrete-control). We want to **discretize it into a finite MDP** (discrete-state), *e.g.* to approximate the value function with value iteration.

Problem: A naive discretization has no notion of spatial proximity, hence we would need a **very large state-discretization**, not even fitting in memory for problems of moderate dimensions.

Following the approach of [McE03, AGL08], adapted to finite MDPs in [CB14, Bac19], we compute a **max-plus linear approximation** of the value function.

# Setting

We consider a **deterministic**, time-homogeneous, infinite-horizon, discounted MDP defined by:

- a state space $\mathcal{S}$,
- an action space $\mathcal{A}$,
- a bounded reward function $r : \mathcal{S} \times \mathcal{A} \to [-R, R]$,
- a dynamics $\varphi_{\cdot}(.) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$,
- and a discount factor $0 \leq \gamma < 1$.

We make the following assumptions:

1. the state space $\mathcal{S}$ is a bounded subset of $\mathbb{R}^d$ ($d \geq 1$);
2. the action space $\mathcal{A}$ is finite.

# Value Iteration

The optimal value function $V^* : \mathcal{S} \to \mathbb{R}$ corresponds to an optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ maximizing the cumulative discounted reward. The greedy policy $\pi$ corresponding to a value function $V$ is then:

$$\pi(s) \in \operatorname*{argmax}_{a \in \mathcal{A}} r(s,a) + \gamma V(\varphi_a(s)).$$

The value iteration algorithm consists in computing $V^*$ as the unique fixed point of the Bellman operator $T : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$:

$$TV(s) := \max_{a \in \mathcal{A}} r(s,a) + \gamma V(\varphi_a(s)).$$

The value iteration algorithm iteratively computes the recursion $V_{k+1} = TV_k$ that converges to $V^*$, **with linear rate** since $T$ is strictly contractive with factor $\gamma < 1$. But if $\mathcal{S}$ is a finite set, it requires $O(|\mathcal{A}| \cdot |\mathcal{S}|)$ computations, and the storage of $O(|\mathcal{S}|)$ values of $V_k$ at each step.

# Max-Plus Linear Approximation

The max-plus semiring is defined as $(\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$, where $\oplus$ represents the maximum operator, and $\otimes$ represents the usual sum.

Let $\mathcal{W}$ be a finite dictionary of functions $w : \mathcal{S} \to \mathbb{R}$. The value function can be approximated by a max-plus linear combination of functions in $\mathcal{W}$.

For $\alpha \in \mathbb{R}^{\mathcal{W}}$, we define the **max-plus linear combinations**:

$$V(s) = \bigoplus_{w \in \mathcal{W}} \alpha(w) \otimes w(s) = \max_{w \in \mathcal{W}} \alpha(w) + w(s).$$

and we write it more compactly:

$$V = W\alpha$$

# Max-Plus Indicator Functions

Possible dictionaries of functions:

- Smooth: $w(s) = -c\|s - s_0\|^2$
- Lipschitz: $w(s) = -c\|s - s_0\|$
- Indicator: $w(s) = \begin{cases} 0 & \text{if } s \in A \\ -\infty & \text{otherwise} \end{cases}$
- Soft indicator: $w(s) = -c\text{dist}(s, A)^2$.

Smooth or Lipschitz basis functions are used to approximate value functions of the same regularity, controlled by $c$ [AGL08].

Piecewise constant value functions are good candidates for a discretization. They are used in [Bac19] to cluster similar states in discrete MDPs.

# Max-Plus Projections

We define the following four operators:

$$W : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{S}}, \quad W\alpha(s) := \max_{w \in \mathcal{W}} \alpha(w) + w(s)$$

$$W^{+} : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{W}}, \quad W^{+}V(w) := \inf_{s \in \mathcal{S}} V(s) - w(s)$$

$$W^{\top} : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{W}}, \quad W^{\top}V(w) := \sup_{s \in \mathcal{S}} V(s) + w(s)$$

$$W^{\top +} : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{S}}, \quad W^{\top +}\alpha(s) := \min_{w \in \mathcal{W}} \alpha(w) - w(s).$$

$W^{+}$ is the **residuation** and acts as a pseudo-inverse:

$$W\alpha \leq V \Leftrightarrow \alpha \leq W^{+}V$$

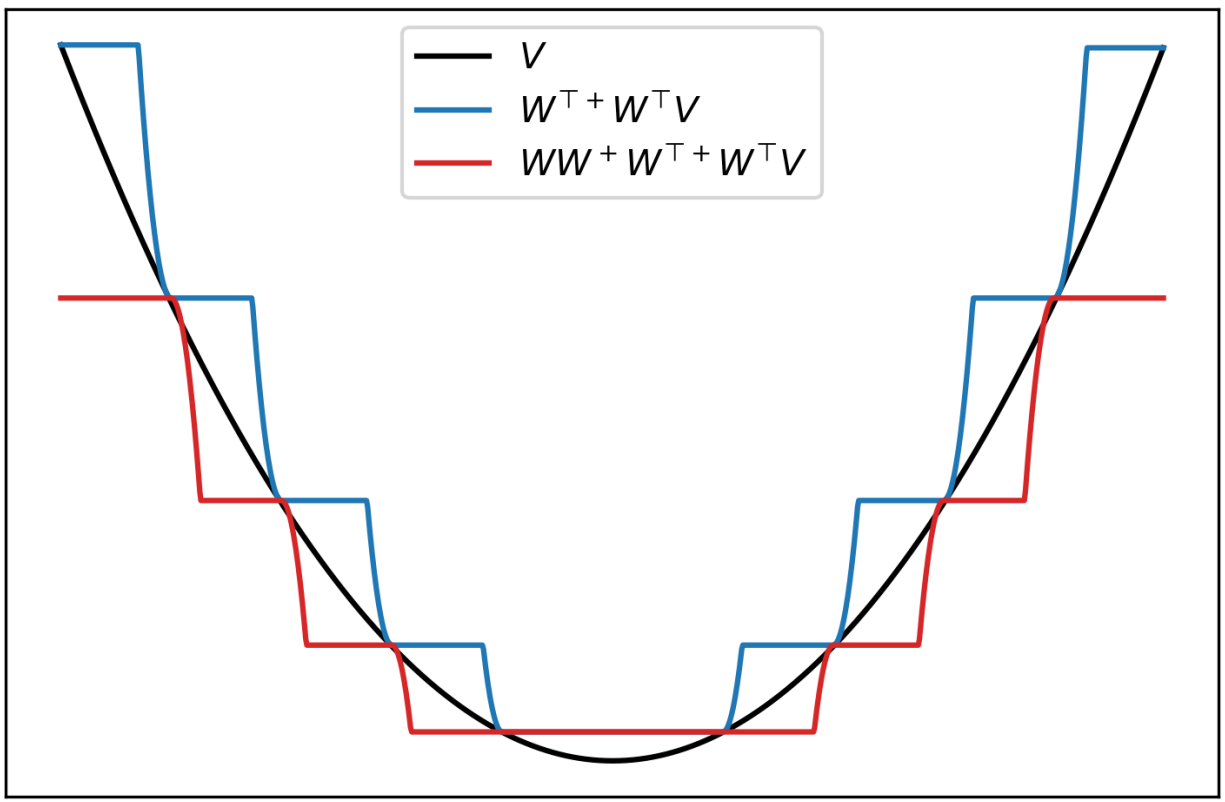
We also define a "dot product":

$$\forall z, w \in \mathbb{R}^{\mathcal{S}}, \quad \langle z, w \rangle := \sup_{s \in \mathcal{S}} z(s) + w(s).$$

# Max-Plus Projections

A function $V \in \mathbb{R}^{\mathcal{S}}$ can be lower- (or upper-) projected onto the basis $\mathcal{W}$:

$$P_{\mathcal{W}}(V) = WW^{+}V$$
$$P^{\mathcal{W}}(V) = W^{\top +}W^{\top}V$$



# Max-Plus Properties of the Bellman Operator

The structure of the Bellman operator

$$T : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$$
$$TV(s) = \max_{a \in \mathcal{A}} r(s,a) + \gamma V(\varphi_a(s))$$

is naturally compatible with max-plus algebra.
It is **max-plus additive and homogeneous**:

$$T(V \oplus V') = T(\max\{V, V'\}) = \max\{TV, TV'\} = TV \oplus TV'$$
$$T(c \otimes V) = T(c + V) = \gamma c + TV = c^{\otimes\gamma} TV.$$

This will be helpful to reduce the computational complexity of the subsequent approximation method. Importantly, **additivity no longer holds for stochastic MDPs**.

# Projection Method

Algorithm: alternative applications of the Bellman operator and projections onto $\mathcal{W}$:

$$\boxed{V_{k+1} = WW^+TV_k}.$$

Hence if $V_k$ is represented as $W\alpha_k$, then $\alpha_{k+1}$ is given by $\alpha_{k+1} = W^+TW\alpha_k$, where the operator $W^+TW : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{W}}$ is computed by:

$$\alpha_{k+1}(w) = \inf_{s\in\mathcal{S}} \max_{w'\in\mathcal{W}} \gamma\alpha_k(w') + Tw'(s) - w(s).$$

This computation is a min/max problem, which is not easy to solve in general. If $\mathcal{S}$ is finite, this requires to compute $|\mathcal{S}| \cdot |\mathcal{W}|$ values at each iteration.

# Variational Method

Let's define a second dictionary of test functions $\mathcal{Z}$. The value iteration recursion $V_{k+1} = TV_k$ is replaced by a variational formulation:

$$\langle z, V_{k+1} \rangle = \langle z, TV_k \rangle \quad \forall z \in \mathcal{Z},$$

of which we consider the **maximal solution** in $\mathrm{span}(W)$ [AGL08]:

$$\boxed{V_{k+1} = WW^+ Z^\top + Z^\top TV_k}.$$

If $V_k = W\alpha_k$, we have the following recursion:

$$\alpha_{k+1} = W^+ Z^\top + Z^\top TW\alpha_k.$$

# Variational Method

The operator $W^+ Z^\top + Z^\top T W : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{W}}$ decomposes as $M \circ K$, with $K = Z^\top T W : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{Z}}$ and $M = W^+ Z^{\top +} : \mathbb{R}^{\mathcal{Z}} \to \mathbb{R}^{\mathcal{W}}$.
The recursion may be recast as:

$$\begin{aligned}
\beta_{k+1}(z) = K\alpha_k(z) &= \sup_{s \in \mathcal{S}} z(s) + \max_{w \in \mathcal{W}} \gamma \alpha_k(w) + Tw(s) \\
&= \max_{w \in \mathcal{W}} \gamma \alpha_k(w) + \langle z, Tw \rangle \\
\alpha_{k+1}(w) = M\beta_{k+1}(w) &= \inf_{s \in \mathcal{S}} -w(s) + \min_{z \in \mathcal{Z}} \beta_{k+1}(z) - z(s) \\
&= \min_{z \in \mathcal{Z}} \beta_{k+1}(z) - \langle z, w \rangle.
\end{aligned}$$

$W^+ Z^{\top +} Z^\top T W$ is a $\gamma$-contraction, hence the recursion will converge with linear rate to the unique fixed point. The $|\mathcal{Z}| \cdot |\mathcal{W}|$ values $\langle z, Tw \rangle$ for $(z, w) \in \mathcal{Z} \times \mathcal{W}$ can be **precomputed** at a cost that is independent of the horizon $1/(1-\gamma)$ of the MDP.

# Approximate Value Iteration for Clustering

If $\mathcal{W} = \mathcal{Z}$ and the $(w_i)_{1 \leq i \leq n}$ are max-plus indicators, the approximate value iteration becomes:

$$\alpha_{k+1}(w) = \max_{w' \in \mathcal{W}} \langle w, Tw' \rangle + \gamma \alpha_k(w'),$$

which we interpret as classical value iteration on the **MDP formed with the clusters** $(A(w))_{w \in \mathcal{W}}$ as states, and as rewards the maximal achievable reward going from one cluster to the other:

$$\begin{aligned} R(w, w') = \langle w, Tw' \rangle &= \sup_{s \in \mathcal{S}} w(s) + Tw'(s) \\ &= \sup_{s \in A(w)} \max_{\substack{a \in \mathcal{A} \text{ s.t.} \\ \varphi_a(s) \in A(w')}} r(s, a). \end{aligned}$$

and $R(w, w') = -\infty$ if $A(w')$ cannot be reached from $A(w)$.

# Approximate Value Iteration for Clustering

Experiments taken from [Bac19] in a discrete MDP:

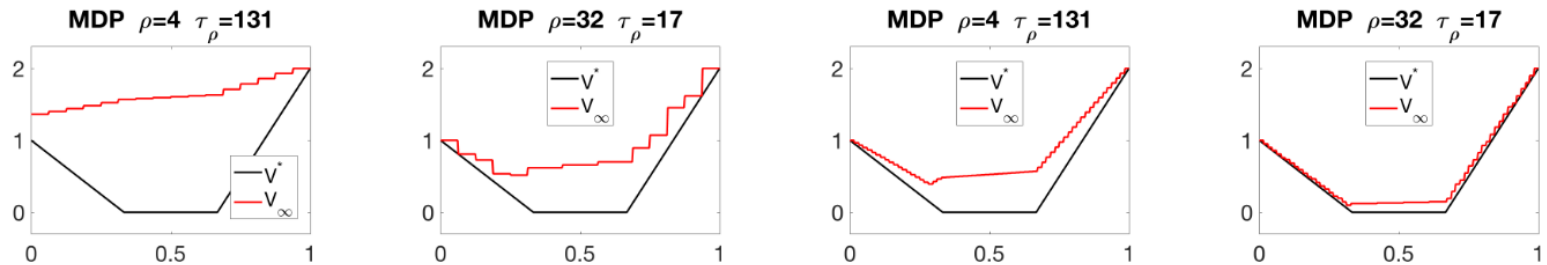



Figure 5: Approximation of a function $V$ with different finite basis with 16 or 64 elements, *within the MDP*, and with values of $\rho$ that are 4 and 32. One-dimensional case (with a convex optimal value function). From left to right: $(n = 16, \rho = 4)$, $(n = 16, \rho = 32)$, $(n = 64, \rho = 4)$, $(n = 16, \rho = 32)$.

This reduced problem is appealing but hard to solve in a continuous state space. Even finding if $R(w, w')$ is finite is a reachability problem, whose solution is not straightforward.
We use soft indicators: $w(s) = -c\text{dist}(s, A(w))^2$, with $c \gg 1$.

# Precomputations

Subproblems: $\langle z, w \rangle$ is independent of the MDP and can be often computed in closed form, and:

$$\begin{aligned} \langle z, Tw \rangle &= \sup_{s \in \mathcal{S}} z(s) + Tw(s) \\ &= \sup_{s \in \mathcal{S},\ a \in \mathcal{A}} z(s) + r(s,a) + \gamma w(\varphi_a(s)). \end{aligned}$$

In [AGL08], $\langle z, Tw \rangle$ is approximated using the Hamiltonian of the control problem. For general MDPs that do not come from an underlying continuous-time control problem, this cannot be done.

# Precomputations

$$\langle z, Tw \rangle = \sup_{s \in \mathcal{S},\ a \in \mathcal{A}} z(s) + r(s,a) + \gamma w(\varphi_a(s))$$

$\langle z, Tw \rangle$ can be approximated by **gradient ascent** on

$$f_a(s) = z(s) + r(s,a) + \gamma w(\varphi_a(s))$$
$$\nabla f_a(s) = \nabla z(s) + \nabla r(s,a) + \gamma J_{\varphi_a}(s)^\top \nabla w(\varphi_a(s)).$$

for each $a \in \mathcal{A}$, and then taking the maximum on $a$.

Seeing this problem like [AGL08] as a perturbation of $\langle z, w \rangle$, an efficient initialization is given by

$$s_0 \in \operatorname*{argmax}_s z(s) + w(s).$$

Even though it **not a concave maximization** problem, choosing strongly concave basis functions $z$ and $w$ has a regularizing effect.

# Full Algorithm

**Input:** MDP, $\mathcal{W}$ and $\mathcal{Z}$, gradient steps $k$, step size $\xi$
**Output:** approximate value function $V$

*Precomputations*:
**for** $z \in \mathcal{Z}, w \in \mathcal{W}$ **do**
  $s, \langle z, w \rangle \leftarrow \operatorname{argmax}, \max_{s \in \mathcal{S}} z(s) + w(s)$
  **for** $a \in \mathcal{A}$ **do**
    $\langle z, Tw \rangle \leftarrow z(s) + r(s,a) + w(\varphi_a(s))$
    **for** $i = 1$ **to** $k$ **do**
      $g \leftarrow \nabla z(s) + \nabla r(s,a) + J_{\varphi_a}(s)^\top \nabla w(\varphi_a(s))$
      $s \leftarrow s + \xi g$
      $f \leftarrow z(s) + r(s,a) + w(\varphi_a(s))$
      $\langle z, Tw \rangle \leftarrow \max\{f, \langle z, Tw \rangle\}$
*Reduced value iteration*:
$\alpha \leftarrow 0$
**repeat**
  **for** $z \in \mathcal{Z}$ **do**
    $\beta(z) \leftarrow \max_{w \in \mathcal{W}} \gamma\alpha(w) + \langle z, Tw \rangle$
  **for** $w \in \mathcal{W}$ **do**
    $\alpha(w) \leftarrow \min_{z \in \mathcal{Z}} \beta(z) - \langle z, w \rangle$
**until** convergence
**return** $V = W\alpha$

# Extension: Multi-step Bellman operator

The Bellman operator $T$ can be replaced by $T^\rho$ for $\rho \geq 1$, replacing accordingly $\gamma$ by $\gamma^\rho$. This makes sense if one time step has a small effect compared to the scale of the basis functions, *e.g.* in clustering if one time step is not enough to cross different clusters.

This makes the computation of $\langle z, T^\rho w \rangle$ more complicated, as it requires to run $|\mathcal{A}|^\rho$ gradient ascents. A simplification is to consider only sequences of constant actions for $\rho$ steps.

# Decomposition of Errors

**Theorem (Approximation of the optimal value function)**

*Let $V^*$ be the optimal value function of the MDP, $\hat{V} = W\hat{\alpha}$, where $\hat{\alpha}$ is the fixed point of $M \circ \hat{K}$, and*

$$\|\hat{K} - K\|_\infty := \sup_{z \in \mathcal{Z}, w \in \mathcal{W}} |\hat{K}_{z,w} - K_{z,w}|.$$

Then:
$$\|\hat{V} - V^*\|_\infty \leq \frac{1}{1-\gamma}\Big(\|WW^+V^* - V^*\|_\infty + \|Z^\top{}^+ Z^\top V^* - V^*\|_\infty + \|\hat{K} - K\|_\infty\Big).$$

# Projection Error

**Proposition (*Approximation properties of soft-indicators*)**

*Let $c > 0$ and $(A_1, ..., A_n)$ a partition of $\mathcal{S}$ where each $A_i$ is convex, compact and non-empty, and let $D = \max_{1 \leq i \leq n} diam(A_i)$.*

*Let $\mathcal{W}_1 = \{w_1^1, ..., w_n^1\}$ and $\mathcal{W}_2 = \{w_1^2, ..., w_n^2\}$ defined by:*

$$\forall i \in \{1, ..., n\}, \forall s \in \mathcal{S}, \quad \begin{cases} w_i^1(s) & = -c_1 dist(s, A_i) \\ w_i^2(s) & = -c_2 dist(s, A_i)^2. \end{cases}$$

*If $V$ has Lipschitz constant $L$ and $c_1 \geq L$, $c_2 \geq \frac{L}{4D}$, then*

$$\|V - W_1 W_1^+ V\|_\infty \leq LD$$

$$\|V - W_2 W_2^+ V\|_\infty \leq LD + \frac{L^2}{4c_2} \leq 2LD.$$

No dependency in $c$ in the bound, for $c$ large enough.

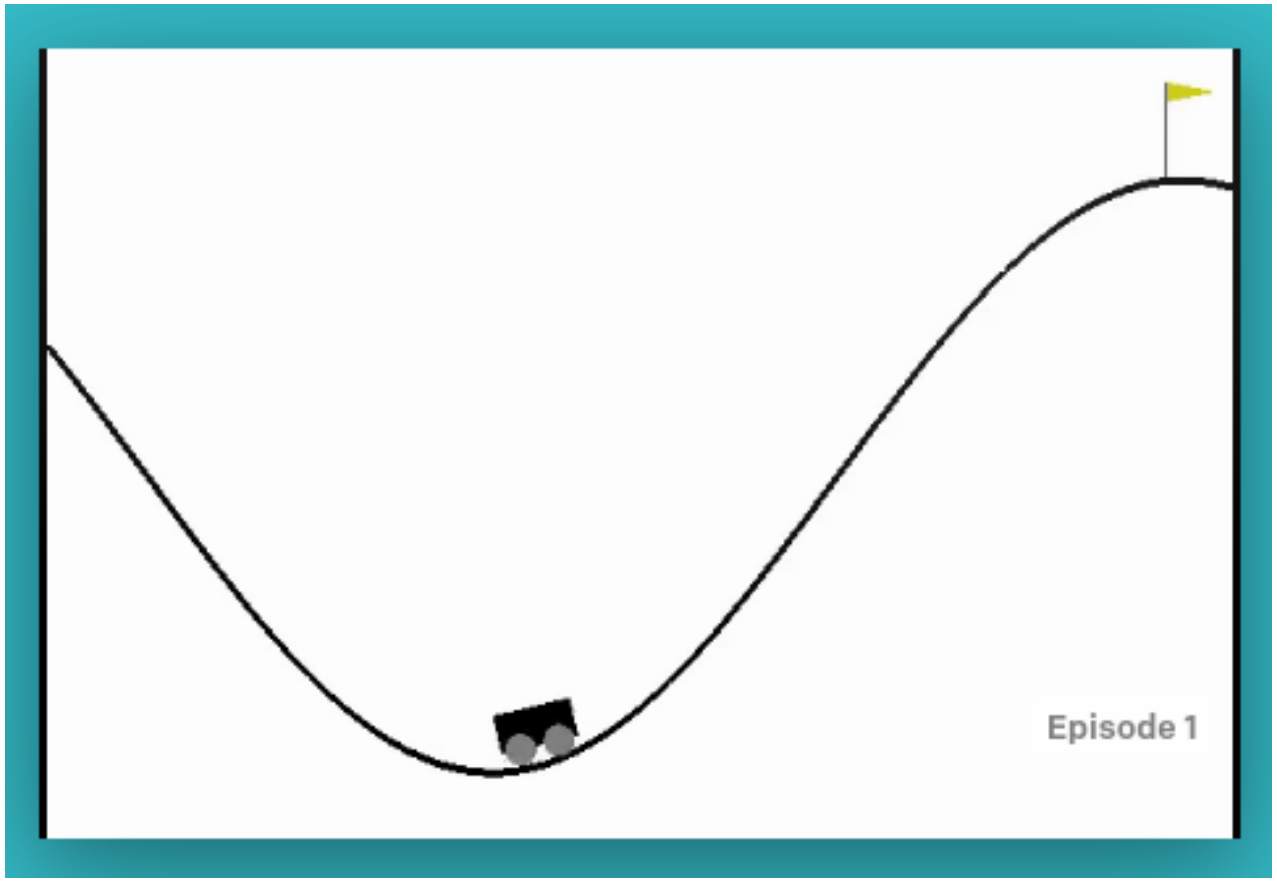
# Greedy Basis Selection

From a partition $(A_1, ..., A_n)$ of the state space, we define a dictionary $\mathcal{W} = \mathcal{Z}$ of soft-indicators $w_i(.) = -c\text{dist}(., A_i)^2$.
Starting from a coarse partition, we compute the approximate value function, and then we select one of the $(A_i)_{1 \leq i \leq n}$ that we want to refine. Then we split this cluster into new sub-clusters.

A simple splitting strategy is to subdivide it into $2^d$ smaller parallelepipeds, by a middle cut along each dimension. This corresponds to building a quadtree.

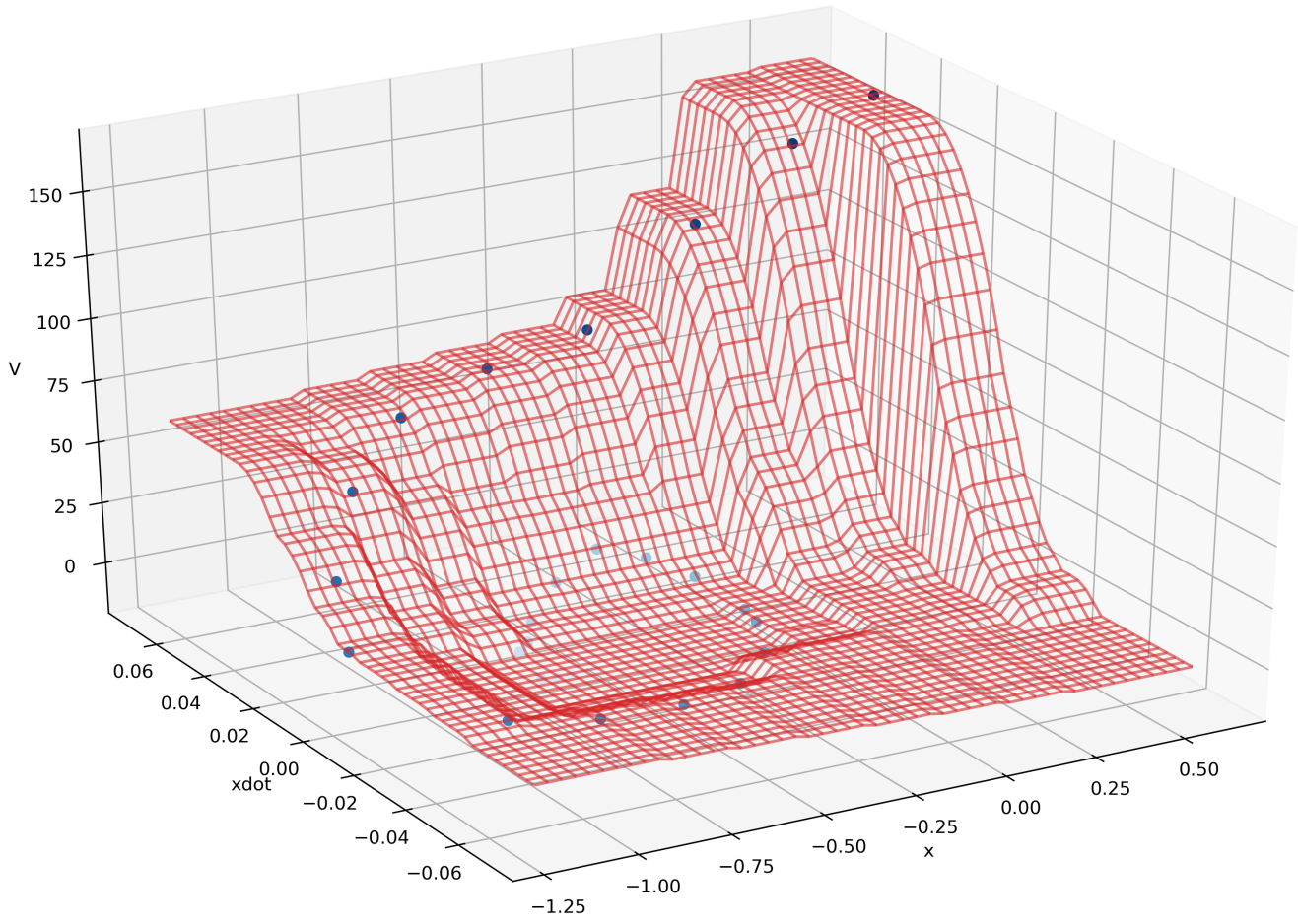
Following the idea of matching pursuit, a simple heuristic is to split the cluster with highest Bellman error $|TV(s) - V(s)|$.

# MDP Example with state dimension 2



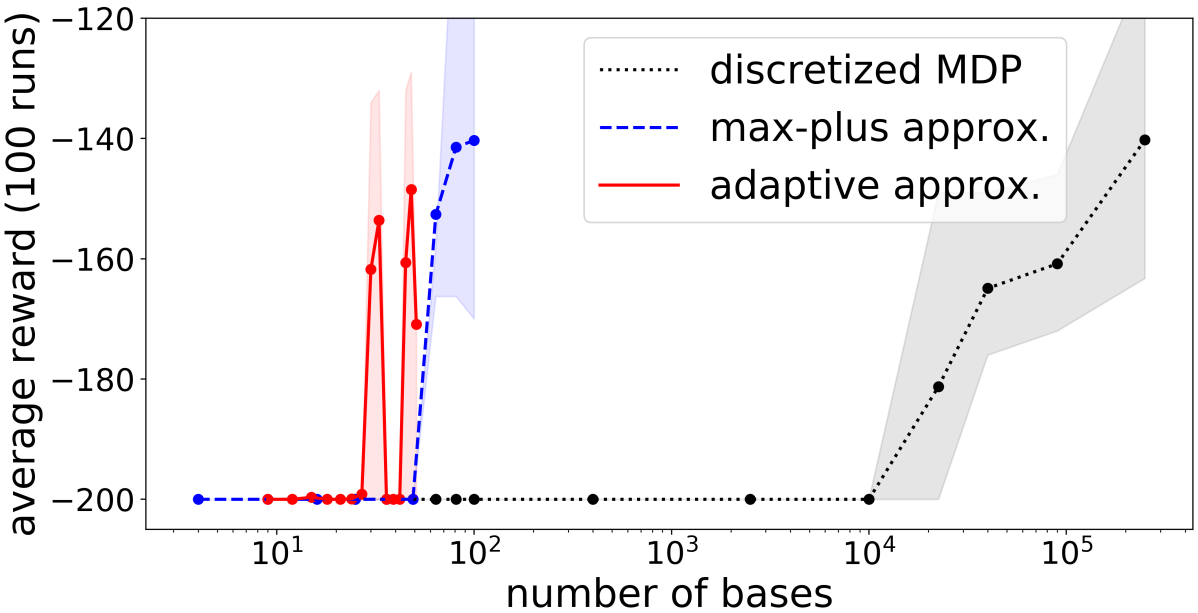

`Mountain` MDP ($d = 2$)

# Approximate value function




Approximate value function obtained with the algorithm

# Results

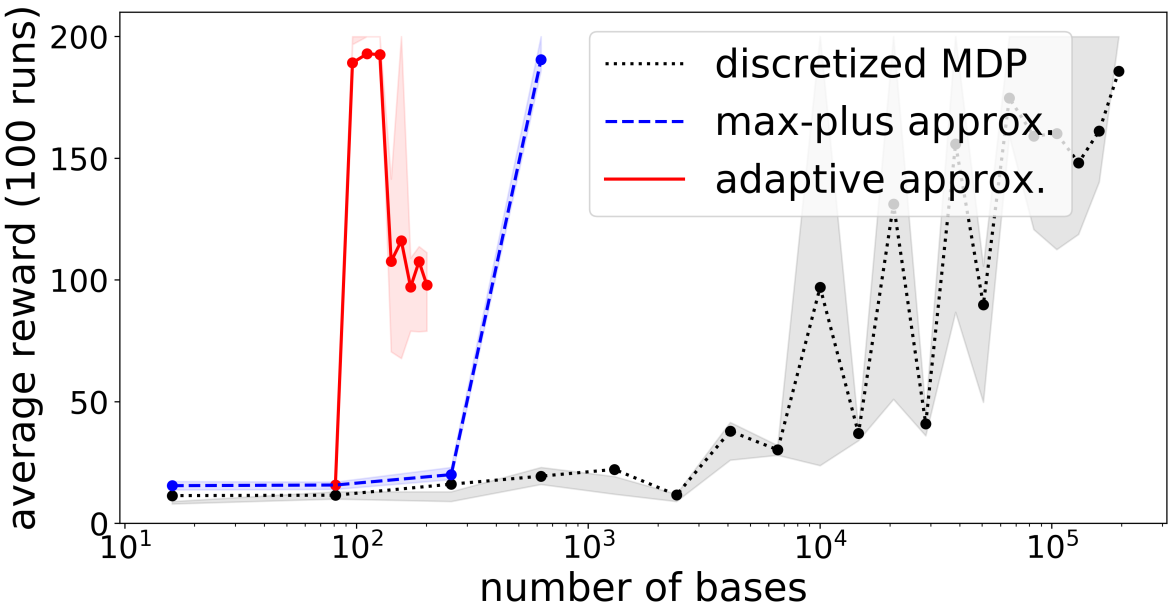



Average performance of the three approximation methods on `Mountain` as a function of the number of parameters.

To get an efficient controller, the max-plus discretization does not need to be as sharp as the naive discretization. The adaptive discretization gives an even sparser representation of the MDP.

# Results

On an MDP with state dimension 4:



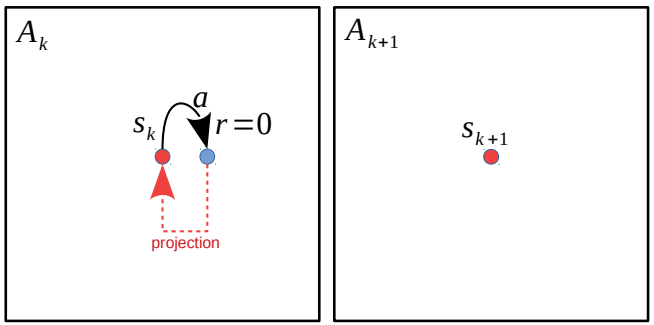

Average performance of the three approximation methods on `Cartpole` as a function of the number of parameters.

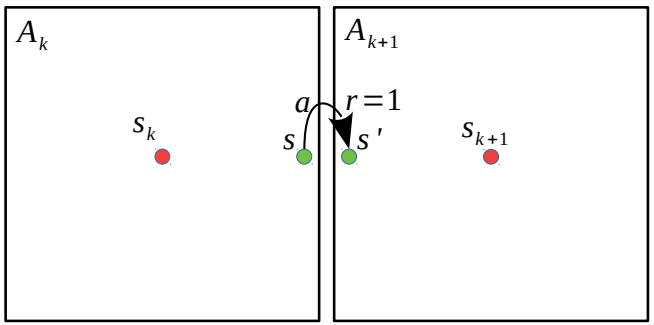
# Naïve vs max-plus discretization

Naive Discretization

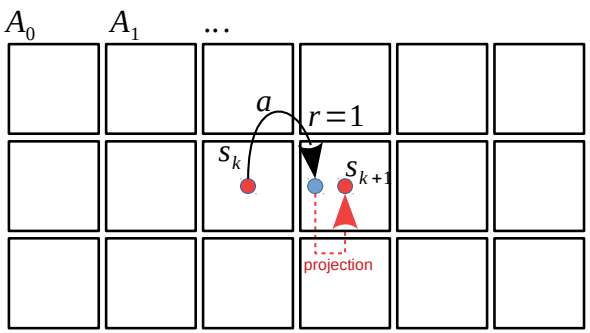
Max-Plus Discretization

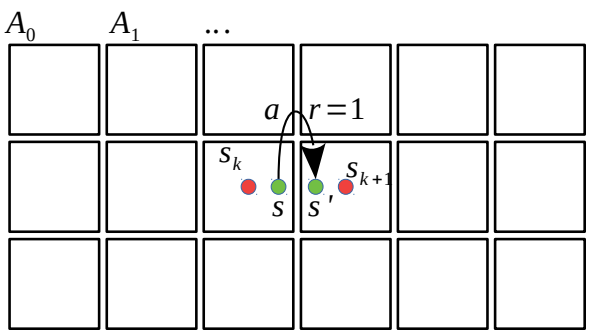
Coarse Discretization




Tight Discretization



# Conclusion

We adapted the approximation method of [AGL08] designed for control systems to MDPs. It provides intuitive state-space discretizations with a reasonable number of parameters.

Possible **future directions**:

- generalization to Q-function approximation.
- a more efficient adaptive algorithm, with some exploration mechanism, *e.g.* with upper confidence bounds?
- how to deal with stochastic MDPs, without becoming computationally intractable?
- how to extend to Q-learning (model-free reinforcement learning)?
  $\rightarrow$ [Gon21] proposed a first online learning approach, "*following the philosophy of reinforcement learning: explore the environment, receive the rewards and use this information to improve the knowledge of the value function*."

# References I



Marianne Akian, Stéphane Gaubert, and Asma Lakhoua, *The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis*, SIAM Journal on Control and Optimization **47** (2008), no. 2, 817–848.


Francis Bach, *Max-plus matching pursuit for deterministic Markov decision processes*, working paper or preprint, June 2019.


L Chandrashekar and Shalabh Bhatnagar, *Approximate dynamic programming with (min;+) linear function approximation for markov decision processes*, 53rd IEEE Conference on Decision and Control, IEEE, 2014, pp. 1588–1593.


Vinicius Mariano Gonçalves, *Max-plus approximation for reinforcement learning*, Automatica **129** (2021), 109623.


William M McEneaney, *Max-plus eigenvector representations for solution of nonlinear H infinity problems: basic concepts*, IEEE Transactions on Automatic Control **48** (2003), no. 7, 1150–1163.