Machine Learning, 1, 1–31 (2001)

Variable Resolution Discretization in Optimal Control

RÉMI MUNOS remi.munos@polytechnique.fr
Centre de Mathématiques Appliquées,
Ecole Polytechnique, 91128 Palaiseau, France

ANDREW MOORE awm@cs.cmu.edu
Robotics Institute, Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh, PA 15213, USA

Editor: Satinder Singh

Abstract. The problem of state abstraction is of central importance in optimal control, reinforcement learning and Markov decision processes. This paper studies the case of variable resolution state abstraction for continuous time and space, deterministic dynamic control problems in which near-optimal policies are required. We begin by defining a class of variable resolution policy and value function representations based on Kuhn triangulations embedded in a kd-trie. We then consider top-down approaches to choosing which cells to split in order to generate improved policies. The core of this paper is the introduction and evaluation of a wide variety of possible splitting criteria. We begin with local approaches based on value function and policy properties that use only features of individual cells in making split choices. Later, by introducing two new non-local measures, *influence* and *variance*, we derive splitting criteria that allow one cell to efficiently take into account its impact on other cells when deciding whether to split. Influence is an efficiently-calculable measure of the extent to which changes in some state effect the value function of some other states. Variance is an efficiently-calculable measure of how risky is some state in a Markov chain: a low variance state is one in which we would be very surprised if, during any one execution, the long-term reward attained from that state differed substantially from its expected value, given by the value function.

The paper proceeds by graphically demonstrating the various approaches to splitting on the familiar, non-linear, non-minimum phase, and two dimensional problem of the "Car on the hill". It then evaluates the performance of a variety of splitting criteria on many benchmark problems, paying careful attention to their number-of-cells versus closeness-to-optimality tradeoff curves.

Keywords: Optimal control, reinforcement learning, variable resolution discretization, adaptive mesh refinement

1. Introduction

This paper is about non-uniform discretization of state spaces when finding optimal controllers for continuous time and space Markov Processes.

There is an extensive literature in Numerical Analysis about solving numerically partial differential equations such as the famous Hamilton-Jacobi-Bellman (HJB) equations that arise in optimal control.

Discretization techniques (Kushner & Dupuis, 1992) using finite-element (FE) or finite-difference (FD) methods applied to *uniform* grids (and multi-grids) are widely used and provide convergence results and rates of convergence (using analytical

(Barles & Souganidis, 1991; Crandall, Ishii, & Lions, 1992; Crandall & Lions, 1983) or probabilistical (Kushner & Dupuis, 1992; Dupuis & James, 1998) approaches).

However, such uniform discretization suffer from impractical computational requirements when the size of the discretization step is small, especially when the state space is of high dimension. Indeed, since the symmetries of the control problem or the smoothness properties of the value function are not reflected in the structure of the grid, possible compact representations and computation are not exploited.

On the other hand, there is a growing interest for combining compact function representations (such as Neural Networks) with Dynamic Programming (Bertsekas & Tsitsiklis, 1996; Baird, 1995; Sutton, 1996) in order to handle high dimensionality. Successful applications include the game of backgammon (Tesauro, 1995) and a controller for elevator dispatching (Crites & Barto, 1996). However in general, there is no guarantee of convergence to the optimal solution (Boyan & Moore, 1995; Baird, 1995; Munos, 2000; Munos, Baird, & Moore, 1999). Some local convergence results are in (Gordon, 1995; Baird, 1998; Tsitsiklis & Van Roy, 1996; Bertsekas & Tsitsiklis, 1996).

The distinction between discretization and approximation methods is not simple. Usually we denote by discretization a way to decompose a function using a set of basis functions with *local* support (such as 'hat' functions used in finite-element methods) whereas approximation methods refer to using basis functions with *global* support (possibly the whole state space). However this distinction is not obvious since there exists some fancy grids (for example the sparse grids (Zenger, 1990)) that use extrapolation on large parts of the state space and some function approximators that use local basis functions (such as the Normalized Gaussian Networks (Moody & Darken, 1989)).

In this paper we consider variable resolution discretizations to approximate the value function and the optimal control and compare experimentally several splitting criteria. The ideas developed here are illustrated on a specific grid representation using kd-trees and Kuhn triangulation. However the same ideas can be used to implement variable resolution on other kinds of grids such as the sparse grids (Zenger, 1990; Griebel, 1998), the random and low-discrepancy grids (Niederreiter, 1992; Rust, 1996).

We consider a "general towards specific" approach where an initial coarse grid is successively refined at some areas of the state space according to a splitting criterion. In this work we evaluates and compare the performance of a variety of splitting criteria. We start (section 6) with two criteria - the *corner-value difference* and the *value non-linearity* - which consider splitting around the "singularities" of the value function. This is a refinement criterion commonly used in numerical resolution of partial differential equations using adaptive meshes (see for example (Grüne, 1997) for HJB equations).

This method approximates very accurately the value function, but it may be computationaly very expensive when the value function is discountinous.

Besides, the singularities of the value function are usually not located at the same areas as those of the optimal controller: a good approximation of the value function at some areas is not needed if this does not have any impact on the quality of the controller.

Next (section 7), we consider a splitting criterion - the *policy disagreement* - that takes into account the policy. This method split only where the optimal policy is expected to change. Unfortunately, the transition boundaries of the optimal control obtained are not optimally located, the reason for this being that the value function is not correctly approximated at the areas that have an "influence" on these boundaries. We illustrate the shortcomings of these *local* approaches that only consider features of individual cells in making split choices, and justify the need for *global* splitting criteria that take into account the non-local impact of the splitting process.

In section 8, we introduce the notion of *influence* as a measure of the non-local contribution of a state to the value function at other states. Then, in section 9, we define the *variance* of the expected future rewards. We show how to combine these two measures to derive efficient grid refinement techniques.

We describe an heuristic which intends to select the cells whose splitting will mostly increase the accuracy of the value function at the parts of the state space where there is a transition in the optimal control.

We illustrate the different splitting criteria on the "Car on the hill" problem described in section 4, and in section 11 we show the results for other control problems, including the 4-dimensional "Cart-pole", "Acrobot", "space-shuttle" and "airplane meeting" problems.

In this paper we make the assumption that we have a model of the dynamics and of the reinforcement function. For convenience we assume that the dynamics are deterministic; however the results are extendible to the stochastic case (provided that we remove the natural noise from the measure of variance, as suggested in the last remark of section 10).

2. Description of the optimal control problem

We consider discounted deterministic control problems. Let $x(t) \in X$ be the *state* of the system, with the *state space* X being a compact subset of $\mathbb{R}^d$. The evolution of the state depends on the *control* $u(t) \in U$ (with the *control space* U a finite set of possible actions) by the differential equation, called *state dynamics*:

$$\frac{dx(t)}{dt} = f(x(t), u(t)) \tag{1}$$

For an initial state x and a control function $u(t)$, this equation leads to a unique *trajectory* $x(t)$. Let τ be the *exit time* from the state space (with the convention that if $x(t)$ always stays in X, then $\tau = \infty$). Then, we define the *gain* J as the discounted cumulative reinforcement:

$$J(x; u(t)) = \int_0^\tau \gamma^t r(x(t), u(t))dt + \gamma^\tau r_b(x(\tau)) \tag{2}$$

where $r(x, u)$ is the *current reinforcement* and $r_b(x)$ the *boundary reinforcement.* γ is the *discount factor* $(0 \leq \gamma < 1)$.

The objective of the control problem is to find, for any initial condition x, the control $u^*(t)$ that maximizes the functional J.

Here, we use the method of *Dynamic Programming* (DP) that introduces the *value function* (VF), maximum of J as a function of initial state x:

$$V(x) = \sup_{u(t)} J(x; u(t)).$$

From the DP principle we know (see (Fleming & Soner, 1993) for example) that V satisfies a first-order non-linear differential equation, called the *Hamilton-Jacobi-Bellman* (HJB) equation:

THEOREM 1 *If V is differentiable at $x \in X$, let $DV(x)$ be the gradient of V at x, then the following HJB equation holds at x:*

$$V(x)\ln\gamma + \max_{u\in U}[DV(x).f(x,u) + r(x,u)] = 0 \tag{3}$$

DP computes the VF in order to define the optimal control with a feed-back control policy $\pi(x) : X \rightarrow U$ such that the optimal control $u^*(t)$ at time t only depends on current state $x(t)$: $u^*(t) = \pi(x(t))$. Indeed, from the value function, we deduce the following optimal feed-back control policy:

$$\pi(x) \in \arg\max_{u\in U}[DV(x).f(x,u) + r(x,u)] \tag{4}$$

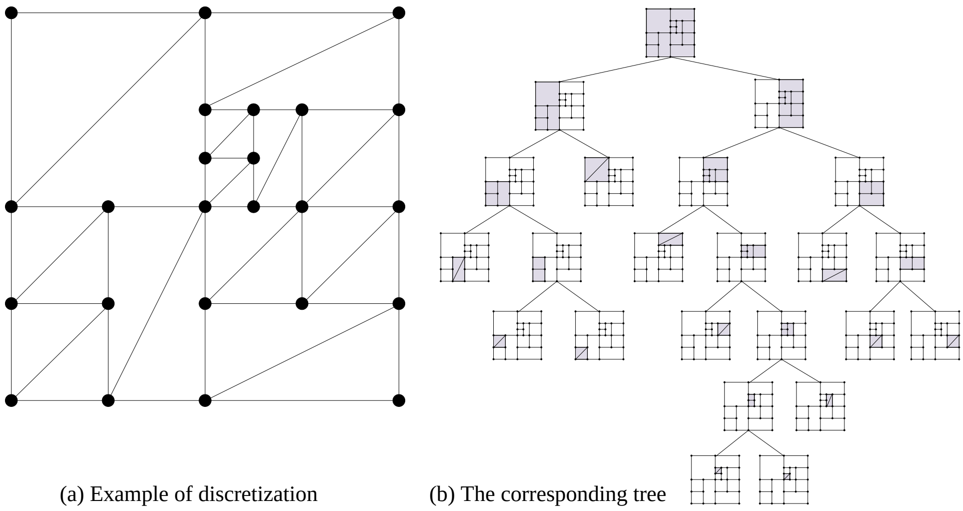
3. The discretization process

In order to discretize the continuous control problem described in the previous section, we use the numerical approximation scheme of (Kushner & Dupuis, 1992). We implement a class of functions known as *barycentric interpolators* (Munos & Moore, 1998), built from a triangulation of the state-space using a tree structure. This representation has been chosen for its very fast computational properties.

Here is a description of this class of functions. The state-space is discretized into a variable resolution grid using a structure of a tree. The root of the tree covers the whole state space, supposed to be a (hyper) rectangle. It has two branches which divide the state space into two smaller rectangles by means of a hyperplane perpendicular to the chosen splitting dimension. In the same way, each node (except for the leaves) splits in some direction $i = 1..d$ the rectangle it covers at its middle into two nodes of equal areas (see Figure 1). This kind of structure is known as a kd-trie (Knuth, 1973), and is a special kind of kd-tree (Friedman, Bentley, & Finkel, 1977) in which splits occur at the center of every cell.

On every leaf, we implement a Coxeter-Freudenthal-Kuhn triangulation (or simply the *Kuhn triangulation* (Moore, 1992)). In dimension 2 (Figure 1(b)) each rectangle is composed of 2 triangles. In dimension 3 (see Figure 2) they are composed of 6 pyramids, and in dimension d, of $d!$ simplexes.

The interpolated functions considered here are defined by their values at the corners of the rectangles. We use the Kuhn triangulation to linearly interpolate inside the rectangles. Thus, these functions are piecewise linear, continuous inside each rectangle, but may be discontinuous at the boundary between two rectangles.



(a) Example of discretization

(b) The corresponding tree

Figure 1. (a) An example of discretization of the state space. There are 12 cells and 24 corners (the dots). (b) The corresponding tree structure. The area covered by each node is indicated in gray level. We implement a Kuhn triangulation on every leaf.

The approach of using Kuhn triangulations to interpolate the value function has been introduced to the reinforcement learning literature by (Davies, 1997).

Remark. As we are going to approximate the value function V with such piecewise linear functions, it is very easy to compute the gradient DV at (almost) any point of the state space, thus making it possible to use the feed-back equation (4) to deduce the corresponding optimal control.

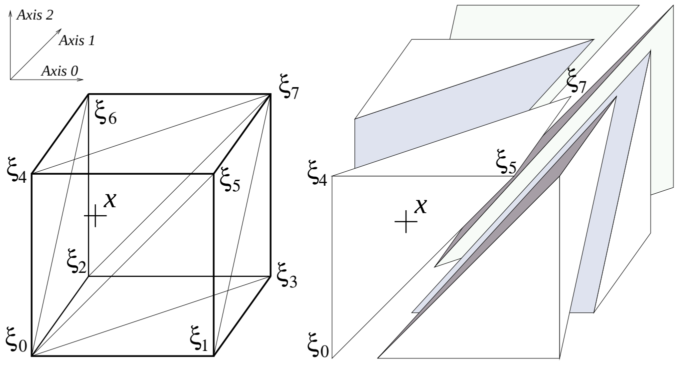


Figure 2. The Kuhn triangulation of a (3d) rectangle. The point x satisfying $1 \geq x_2 \geq x_0 \geq x_1 \geq 0$ is in the simplex $(\xi_0, \xi_4, \xi_5, \xi_7)$.

3.1. Computational issues

Although the number of simplexes inside a rectangle is factorial with the dimension d, the computation time for interpolating the value at any point inside a rectangle is only of order $(d \ln d)$, which corresponds to a sorting of the d relative coordinates $(x_0, ..., x_{d-1})$ of the point inside the rectangle.

Assume we want to compute the indexes $i_0, ..., i_d$ of the $(d+1)$ vertices of the simplex containing a point defined by its relative coordinates $(x_0, ..., x_{d-1})$ with respect to the rectangle in which it belongs to. Let $\{\xi_0, ..., \xi_{2^d}\}$ be the corners of this d-rectangle. The indexes of the corners use the binary decomposition in dimension d, as illustrated in Figure 2. Computing these indexes is achieved by sorting the coordinates from the highest to the smallest: there exist indices $j_0, ..., j_{d-1}$, permutation of $\{0, .., d-1\}$, such that $1 \geq x_{j_0} \geq x_{j_1} \geq ... \geq x_{j_{d-1}} \geq 0$. Then the indices $i_0, ..., i_d$ of the $(d+1)$ vertices of the simplex containing the point are: $i_0 = 0$, $i_1 = i_0 + 2^{j_0}$, ..., $i_k = i_{k-1} + 2^{j_{k-1}}$, ..., $i_d = i_{d-1} + 2^{j_{d-1}} = 2^d - 1$. For example, if the coordinates satisfy: $1 \geq x_2 \geq x_0 \geq x_1 \geq 0$ (illustrated by the point x in Figure 2) then the vertices are: ξ_0 (every simplex contains this vertex, as well as $\xi_{2^d - 1} = \xi_7$), ξ_4 (we added 2^2), ξ_5 (we added 2^0) and ξ_7 (we added 2^1).

Let us define the *barycentric coordinates* $\lambda_0, ..., \lambda_d$ of the point x inside the simplex $\xi_{i_0}, ..., \xi_{i_d}$ as the positive coefficients (uniquely) defined by: $\sum_{k=0}^{d} \lambda_k = 1$ and $\sum_{k=0}^{d} \lambda_k \xi_{i_k} = x$. Usually, these barycentric coordinates are expensive to compute; however, in the case of Kuhn triangulation these coefficients are simply: $\lambda_0 = 1 - x_{j_0}$, $\lambda_1 = x_{j_0} - x_{j_1}$, ..., $\lambda_k = x_{j_{k-1}} - x_{j_k}$, ..., $\lambda_d = x_{j_{d-1}} - 0 = x_{j_{d-1}}$. In the previous example, the barycentric coordinates are: $\lambda_0 = 1 - x_2$, $\lambda_1 = x_2 - x_0$, $\lambda_2 = x_0 - x_1$, $\lambda_3 = x_1$.

3.2. *Building the discretized MDP*

We refer to (Kushner & Dupuis, 1992) for the process of discretizing a continuous time and space optimal control problem into a finite Markov Decision Process (MDP), and to (Munos, 2000) for similar methods in reinforcement learning.

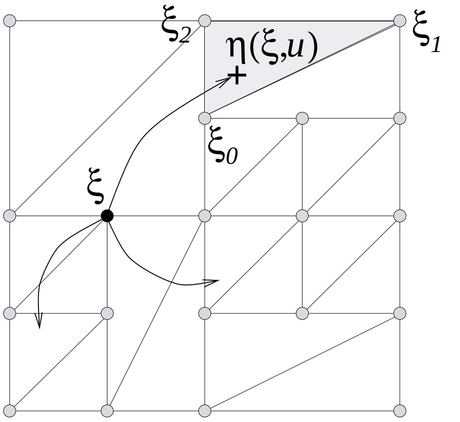


Figure 3. According to the current (variable resolution) grid, we build a discrete MDP. For every corner ξ (state of the MDP) and every control u, we integrate the corresponding trajectory until it enters a new cell at $\eta(\xi, u)$. The probabilities of transition of the MDP for (state ξ, control u) to (states $\{\xi_i\}_{i=0..2}$) are the barycentric coordinates $\lambda_{\xi_i}(\eta(\xi, u))$ of $\eta(\xi, u)$ inside (ξ_0, ξ_1, ξ_2).

For a given discretization, we build a corresponding MDP in the following way. The **state space** of the MDP is the set Ξ of corners of the cells. The **control space** is the finite set U. For every corner $\xi \in \Xi$ and control $u \in U$ we approximate a piece of a trajectory $x(t)$ (using Euler or Runge-Kuta method to integrate the state dynamics (1)) starting from initial state ξ, using a constant control u during some time $\tau(\xi, u)$ until the trajectory enters inside a new cell (which defines the point $\eta(\xi, u) = x(\tau(\xi, u))$ (see Figure 3). At the same time, we also compute the integral of the current reinforcement:

$$R(\xi, u) = \int_{t=0}^{\tau(\xi,u)} \gamma^t \cdot r(x(t), u)dt$$

which defines the **reward** of the MDP. Then we compute the vertices $(\xi_0, ..., \xi_d)$ of the simplex containing $\eta(\xi, u)$ and the corresponding barycentric coordinates $\lambda_{\xi_0}(\eta(\xi, u)), ..., \lambda_{\xi_d}(\eta(\xi, u))$. The **probabilities of transition** $p(\xi_i|\xi, u)$ of the MDP from state ξ and control u to states ξ_i are the barycentric coordinates: $p(\xi_i|\xi, u) = \lambda_{\xi_i}(\eta(\xi, u))$. The DP equation corresponding to this MDP is:

$$V(\xi) = \max_u \left[\gamma^{\tau(\xi,u)} \cdot \sum_{i=0}^{d} p(\xi_i|\xi, u)V(\xi_i) + R(\xi, u) \right] \tag{5}$$

Remark. If while integrating (1) from initial state ξ with the control u, the trajectory exits from the state space at some time $\tau(\xi, u)$, then in the MDP (ξ, u) will lead to a terminal state ξ_t (i.e. satisfying $p(\xi_t|\xi_t, v) = 1, p(\xi \neq \xi_t|\xi_t, v) = 0$ for all v) with probability 1 and with the reward: $R = \int_{t=0}^{\tau(\xi,u)} \gamma^t \cdot r(x(t), u)dt + \gamma^{\tau(\xi,u)} \cdot r_b(x(\tau(\xi, u)))$.

Remark. The interpolated value at $\eta(\xi, u)$ is a linear combination of the values of the vertices of the simplex it belongs to (simplex (ξ_0, ξ_1, ξ_2)) in figure 3), with positive coefficients that sum to one. **Doing this interpolation is thus mathematically equivalent to probabilistically jumping to a vertex: we approximate a *deterministic* continuous process by a *stochastic* discrete one**. The amount of stochasticity introduced by this interpolation process will be estimated by the measure of variance in section 9.

The DP equation (5) is a fixed-point equation satisfying a contraction property (in max-norm), thus it can be solved iteratively with any DP method like *value iteration*, *policy iteration*, or *modified policy iteration* (Puterman, 1994), (Bertsekas, 1987), (Barto, Bradtke, & Singh, 1995).

Remark. The main requirement to obtain the convergence of the approximate VF (solution to the DP equation (5)) to the VF of the continuous process (solution to the HJB equation (3)) is the property of *consistency* of the numerical scheme (Kushner & Dupuis, 1992; Barles & Souganidis, 1991). In the deterministic case, this property roughly means that the expected jump from a state ξ to next states ξ_i when choosing control u in the approximate MDP is a first-order approximation of the state dynamic vector $f(\xi, u)$:

$$\sum_{i=0}^{d} p(\xi_i|\xi, u) \cdot (\xi_i - \xi) = \tau(\xi, u) \cdot f(\xi, u) + o(\delta)$$

with δ being the resolution of the grid. The discretization method previously introduced satisfies this property, which implies that the VF of the discrete MDP converges to the VF of the continuous optimal control problem as the (maximal) size of the cells δ tends to zero.

4. Example: the "Car on the Hill" control problem

For a description of the dynamics of this problem, see (Moore & Atkeson, 1995). This problem is of dimension 2, the variables being the position and velocity of the car. In our experiments, we chose the reinforcement functions as follows: the current reinforcement $r(x,u)$ is zero everywhere. The boundary reinforcement $r_b(x)$ is -1 if the car exits from the left side of the state space, and varies linearly between $+1$ and -1 depending on the velocity of the car when it exits from the right side of the state space. The best reinforcement $+1$ occurs when the car reaches the right boundary (top of the hill) with zero velocity (figure 4). The control u has only 2 possible values: maximal positive or negative thrust.

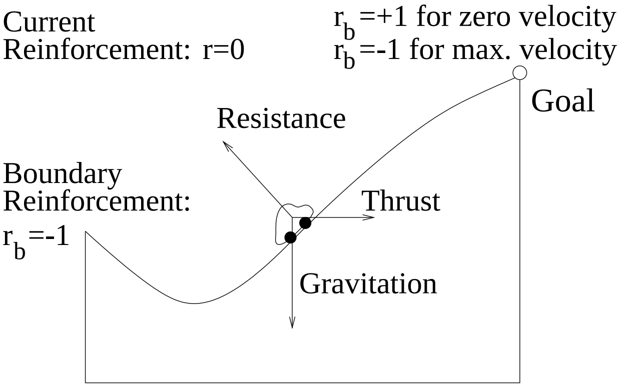


Figure 4. The "Car on the Hill" control problem. The car must reach the top of the hill as fast as possible and stop there. Of course, the car cannot climb the slope without initial speed. It must gain some momentum by first going backwards. It must also be careful not to hit the left boundary.

Figure 5 represents the approximate value function of the MDP obtained by a regular grid of 257 by 257 states (using a discount factor $\gamma = 0.6$).

We observe the following distinctive features of the value function:

- There is a discontinuity in the VF along the "Frontier 1" (see Figure 5) which results from the fact that given an initial point situated above this frontier, the optimal trajectory stays inside the state space (and eventually leads to a positive reward) so the value function at this point is positive. Whereas for a initial point below this frontier, any control lead the car to hit the left boundary (because the initial velocity is too much negative), thus the corresponding value function is negative (see some optimal trajectories in Figure 6). We observe that there is no change in the optimal control around this frontier.

- There is a discontinuity in the gradient of the VF along the upper part of "Frontier 2" which results from a frontier of transition of the optimal control. For example, a point above frontier 2 can reach directly the top of the hill, whereas a point below this frontier has to go backwards and do one loop to gain enough momentum to reach the top (see Figure 6). Moreover, we observe that around the lower part of frontier 2 (see Figures 5), there is no visible irregularity of the VF despite the fact that there is a change in the optimal control.

- There is a discontinuity in the gradient of the VF along the “Frontier 3” because of a change in the optimal control (below the frontier, the car accelerates in order to reach the goal as fast as possible, whereas above, it decelerates to reach the top of the hill with the lowest velocity and receive the highest reward).

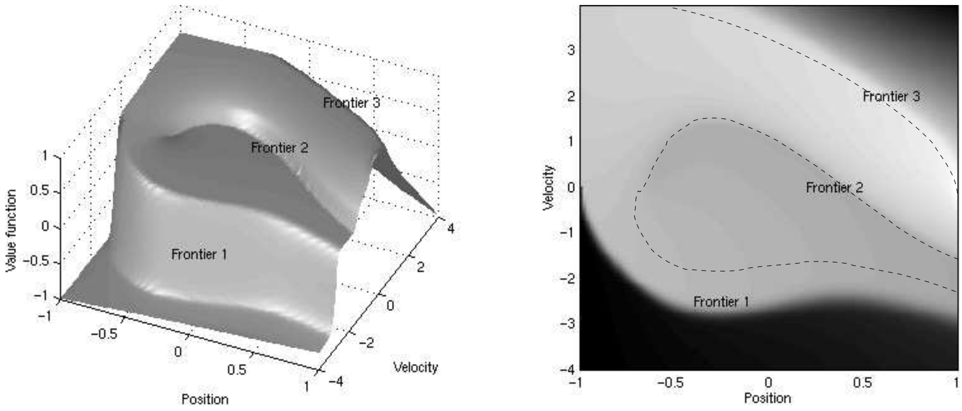


Figure 5. The value function of the Car-on-the-Hill problem obtained by a regular grid of 257 by 257 = 66049 states. The Frontier 1 (white line) illustrates the discontinuity of the VF, the Frontiers 2 and 3 (black lines) stands where there is a transition of the optimal control.

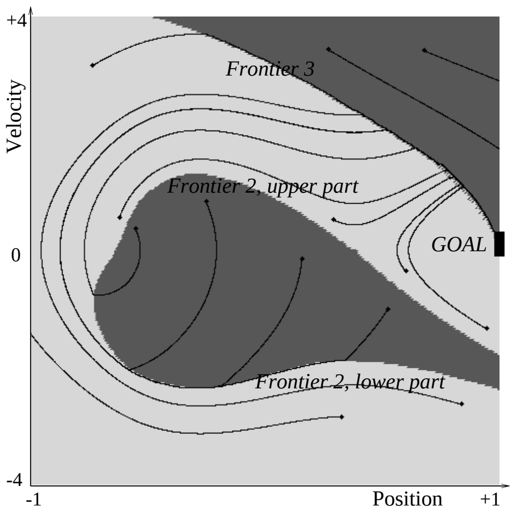


Figure 6. The optimal policy is indicated by different gray levels (light gray=positive thrust, dark gray=negative thrust). Several optimal trajectories are drawn for different initial starting points.

We deduce from these observations that a discontinuity in the value function (frontier 1) does not necessarily indicate that there is a transition in the optimal control, and that a discontinuity in the gradient of the value function (frontiers 2 and 3) may accompany a frontier of transition in the optimal control.

5. The variable resolution approach

We start with an initial coarse discretization and build the corresponding MDP. We solve it and obtain a initial (rough) approximation of the value function. Then, we choose which cells to split according to the process:

1. Score each cell for each direction i according to some splitting criterion.
2. Select the top $h\%$ (where h is a parameter) of the highest scoring couples (cell, direction).

Then, we locally refine the grid by splitting those cells in the corresponding direction. Next, we build the new discretized MDP, and we repeat this cycle (see the splitting process in Figure 7) until some estimation of the quality of approximation of the value function or the optimal control has been reached.

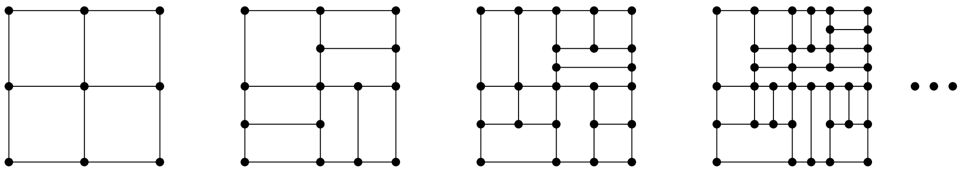


Figure 7. Several discretizations resulting of successive splitting operations.

Note that only the cells that were split, and those whose successive states involve a split cell need to have their state transition recomputed.

Remark. Here, we only consider a top-down process where the discretization is always refined. We could also consider a bottom-up process which would prune the tree and remove over-partitioned leaves.

The main goal of this paper is the study and comparison of several splitting criteria. In what follows, we illustrate the discretizations resulting from different splitting criteria on the "Car on the Hill" control problem previously introduced.

6. Criteria based on the value function

In order to minimize the approximation error of the value function, in the two splitting criteria that follow we choose to split the cells according to local irregularities of the approximate value function.

6.1. *First criterion: average corner-value difference*

For every cell, we compute the average of the absolute difference of the values at the corners of the edges for all directions $i = 0...d-1$. For example, this score on the cell shown in Figure 2 for direction $i = 0$ is $\frac{1}{4}[|V(\xi_1)-V(\xi_0)|+|V(\xi_3)-V(\xi_2)|+|V(\xi_5)-V(\xi_4)|+|V(\xi_7)-V(\xi_6)|]$.

Figure 8 represents the discretization obtained after 15 iterations of this procedure, starting with a 9 by 9 initial grid and using the *corner-value difference* criterion with a splitting rate of $h = 50\%$ of the cells at each iteration.

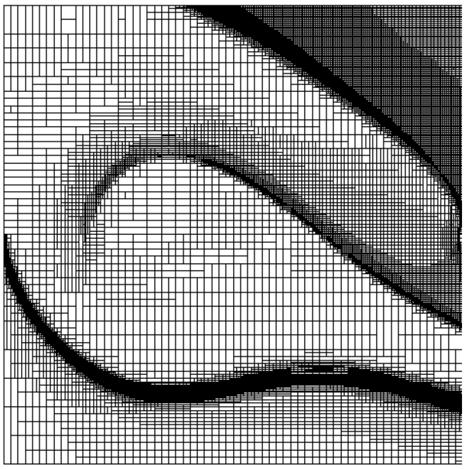


Figure 8. The discretization of the state space for the “Car on the Hill” problem using the *corner-value difference* criterion.

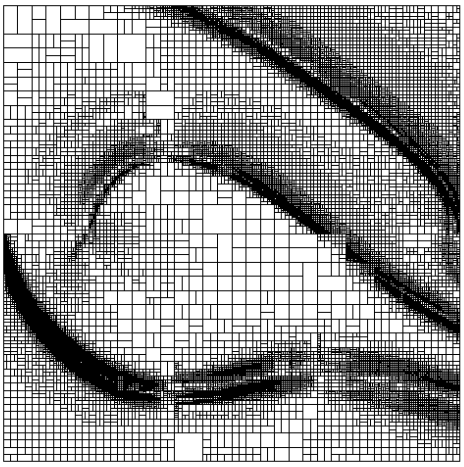


Figure 9. The discretization of the state space for the “Car on the Hill” problem using the *value non-linearity* criterion.

6.2. Second criterion: value non-linearity

For every cell, we compute the variance of the absolute increase of the values at the corners of the edges for all directions $i = 0...d$. This criterion is similar to the previous one except that it computes the variance instead of the average.

Figure 9 shows the corresponding discretization using the value non-linearity criterion with a splitting rate of 50% after 15 iterations.

Comments on these results:

- We observe that in both cases, the splitting occurs around the frontiers 1, 3 and the upper part of frontier 2, previously defined. In fact, the first criterion detects the cells with high average variation of the corner values, thus *splits wherever the value function is not constant.*
- The *value non-linearity* criterion detects the cells with high variance variation of the corner values, thus *splits wherever the value function is not linear.* So this criterion will also concentrate on similar irregularities but with two important differences compared to the *corner-value difference* criterion:
  - The *value non-linearity* criterion splits more parsimoniously than the *corner-value difference* (for a given accuracy of approximation). See, for example, the difference of splitting in the area above frontier 3.
  - The discretization around the discontinuity (frontier 1) are different (see Figure 10 for an explanation on a 1-dimensional problem). The *value non-linearity* criterion splits where the approximate function is the least linear. This explains the 2 parallel tails observed around frontier 1 in Figure 9.
- The refinement process spends a huge amount of resources to refine the grid around the discontinuity (frontier 1) in order to obtain a good approximation

of the VF. However, we notice that the optimal control is constant around this area.

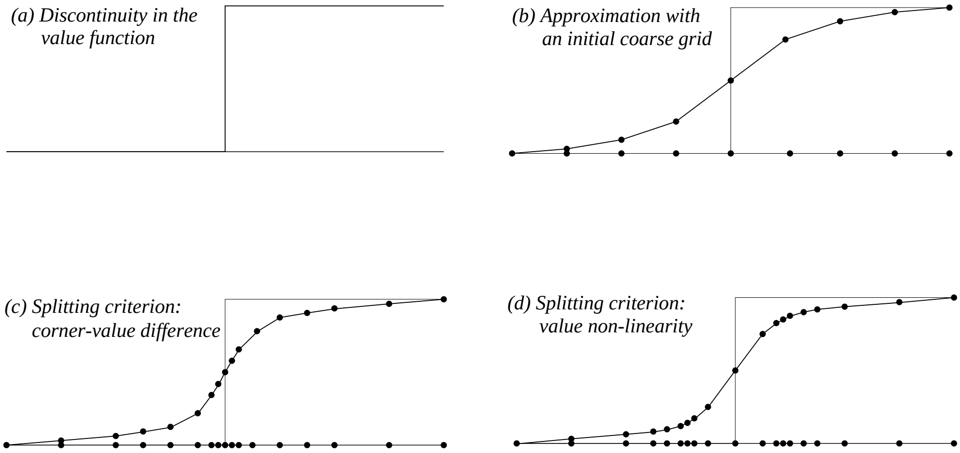


Figure 10. Cross-section of a discontinuous VF (a) and several approximations with a uniform grid (b) and variable resolution grids using the *corner-value difference* (c) and the *value non-linearity* (d) splitting criteria. Notice the different repartition in (c) and (d) of the grid points around the discontinuity.

These variable resolution methods (especially the *value non-linearity*) provide very accurate estimations of the value function compared to uniform discretizations (for a given number of states of the discretized MDP). However, in the end, we want to find the best controller and not so much a very good approximation of the VF, which is simply an artifact used in DP to generate the policy. Thus, we can question the efficiency of the previous splitting methods which spend too much effort around the discontinuity of the VF whereas the control is constant in this area.

In an attempt to spare some computational resources, we introduce in the next section some criteria that also take into account the policy.

Remark. The percentage h of the number of cells to be split at each iteration is a parameter acting on the uniformity of the resolution of the obtained grids. The choice of h allows a tradeoff between deriving almost uniform grids (for high values of h) which ensures convergence of the approximations but with possible high computational cost, and very non-uniform grids (low h), only refined at some critical parts of the state space, which save many computational resources but may potentially converge to sub-optimal solutions.

7. Criteria based on the policy

Figure 6 shows the optimal policy and several optimal trajectories for different starting points. We would like to refine the grid only around the areas of transition of the optimal control: frontiers 2 and 3 but not around frontier 1. In what follows, we introduce such a criterion based on the inconsistency of the control derived from the value function and from the policy.

7.1. The policy disagreement criterion

When we solve the MDP and compute the value function of the DP equation (5), we deduce the following policy for any state $\xi \in \Xi$:

$$\pi(\xi) \in \arg\max_{u \in U} \left[\gamma^{\tau(\xi,u)} \sum_{i=0}^{d} p(\xi_i|\xi,u) V(\xi_i) + R(\xi,u) \right] \tag{6}$$

The policy disagreement criterion compares the control derived from the policy of the MDP (6) with the control derived from the local gradient of V (4).

Remark. Instead of computing the gradient DV for all the ($d!$) simplexes in the cells, we compute an approximated gradient $\tilde{DV}$ for all the (2^d) corners, based on a finite difference quotient. For the example of figure 2, the approximated gradient at corner ξ_0 is $\left(\frac{V(\xi_1)-V(\xi_0)}{||\xi_0-\xi_1||}, \frac{V(\xi_2)-V(\xi_0)}{||\xi_0-\xi_2||}, \frac{V(\xi_4)-V(\xi_0)}{||\xi_0-\xi_4||} \right)$.

Thus, for every corner we compute this approximate gradient and the corresponding optimal control from (4) and compare it to the optimal policy given by (6).

Figure 11 shows the discretization obtained by splitting all the cells where these two measures of the optimal control diverge (the parameter h is not used here).

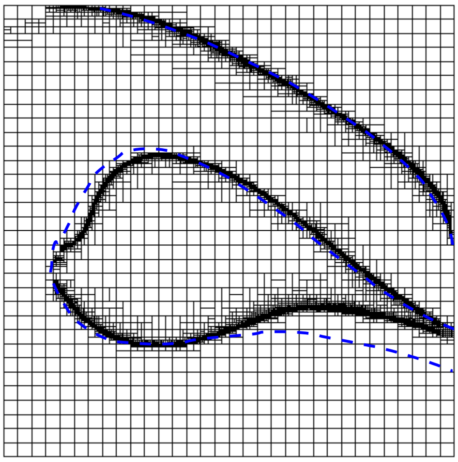


Figure 11. The discretization of the state space using the *policy disagreement* criterion. Here we used an initial grid of 33 × 33. The dash line shows the true frontiers of control transition.

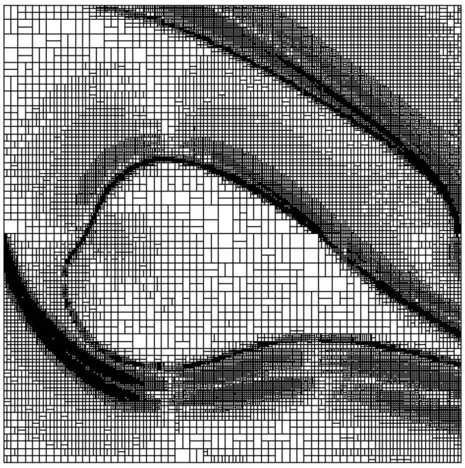


Figure 12. The discretization of the state space for the "Car on the Hill" problem using the combination of the *value non-linearity* and the *policy disagreement* criterion.

This criterion is interesting since it splits at the places where there is a change in the optimal control, thus refining the resolution at the most important parts of the state space for the approximation of the optimal control. However, as we can expect, if we only use this criterion, the value function will not be correctly approximated, and in turn, the policy may suffer from this approximation error. Indeed, we observe that on Figure 11, the bottom part of frontier 2 is (slightly) located higher than its optimal position, shown by the dash line. This error is due to an underestimation of the value function at that area, which is caused by the lack of precision around the discontinuity (frontier 1). Here, we clearly observe the non-local influences between the value function and the optimal control.

The performance of this splitting criterion is relatively weak (see section 7.3). However, this splitting criterion can be beneficially combined with previous ones based on the VF.

7.2. Combination of several criteria

We can combine the *policy disagreement* criterion with the *corner-value difference* or *value non-linearity* criterion in order to obtain the advantages of both methods: a good approximation of the value function on the whole state space and an increase of the resolution around the areas of transition of the optimal control. We can combine those criteria in several ways, for example by a weighted sum of the respective scores of each cells, by a logical operation (split if an and/or combination of these criteria is satisfied), or by an ordering of the criteria (first split with one criterion, then use another one).

Figure 12 shows the discretization obtained by alternatively, between iterations, using the *value non-linearity* criterion and the *policy disagreement* criterion. We observe an increased refinement at areas of singularities of both the value function and the optimal control.

7.3. Comparison of the performance

In order to compare the respective performance of the discretizations, we ran a set (here 256) of optimal trajectories starting from initial states regularly situated in the state space and using the feed-back controller (4). The *performance* of a discretization is the sum of the cumulated reinforcement (the gain defined by equation (2)) obtained along these trajectories, over the set of start positions.

Figure 13 shows the respective performances of several splitting criteria as a function of the number of states of the respective discrete MDPs.

For this 2-dimensional control problem, all the variable resolution approach performs better than uniform grids, except for the *policy disagreement* criterion used alone. However, as we will see later on, for higher dimensional problems, the resources allocated to approximate the VF-discontinuities around areas of the state space that are not useful for improving the optimal control might be prohibitively high.

Can we do better ?

So far, we have only considered local splitting criteria, in which we decide to split a cell according to information (value function and policy) relative to the cell itself. However, the effect of the splitting is not local: it has an influence on the whole state space.

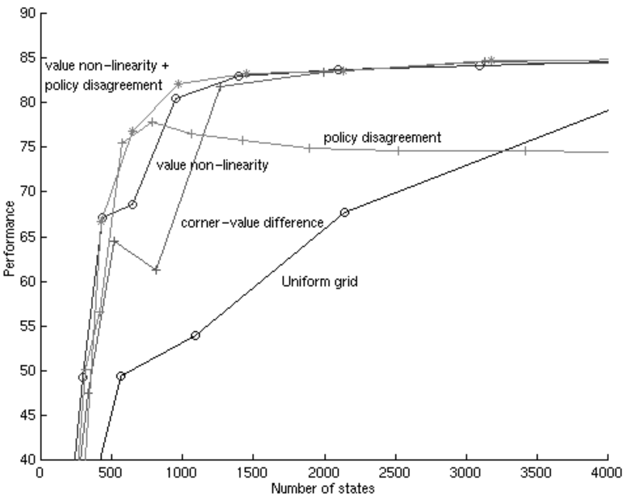


Figure 13. The performance for the uniform versus variable resolution grids for several splitting criterion. Both the *corner-value difference* and *value non-linearity* splitting processes perform better than the uniform grids. The *policy disagreement* splitting is very good for a small number of states but does not improve after, and thus leads to sub-optimal performance. The *policy disagreement* combined with the *value non-linearity* gives the best performances.

We would like to define a refinement process that would split cells only if it is useful to improve the performance. Sections that follow introduce two notions that will be useful for defining such *global* splitting methods: the **influence** measures the extent to which states affect globally the VF, and the **variance**, which measures the amount of interpolation introduced by the discretization process.

8. Notion of influence

Let us consider the Markov chain resulting from the discretized MDP in which we choose the optimal policy π. For convenience, we denote $R(\xi) = R(\xi, \pi(\xi))$, $p(\xi_i|\xi) = p(\xi_i|\xi, \pi(\xi))$, and $\tau(\xi) = \tau(\xi, \pi(\xi))$.

8.1. Intuitive idea

The influence $I(\xi_i|\xi)$ of a state ξ_i on another state ξ is defined as a measure of the extent to which the state ξ_i "contributes" to the VF of another state ξ. This can be done by estimating the infinitesimal variation of the VF at ξ resulting from a infinitesimal modification of the reward at ξ_i.

By considering the discounted transition probabilities $p_1(\xi_i|\xi) = \gamma^{\tau(\xi)} p(\xi_i|\xi)$ and by defining an additional jump to a "dead state" with a transition probability of $1 - \gamma^{\tau(\xi)}$, the influence $I(\xi_i|\xi)$ can be interpreted more intuitively as the expected number of visits of state ξ_i starting from state ξ when using the optimal policy, before the system dies.

8.2. Definition of the influence

Let us define the discounted cumulative $k-$chained probabilities $p_k(\xi_i|\xi)$, which represent the sum of the discounted transition probabilities of all sequences of k states from ξ to ξ_i:

$$\begin{aligned} p_0(\xi_i|\xi) &= 1 \text{ (if } \xi = \xi_i) \text{ or } 0 \text{ (if } \xi \neq \xi_i) \\ p_1(\xi_i|\xi) &= \gamma^{\tau(\xi)} p(\xi_i|\xi) \end{aligned}$$

$$\begin{aligned} p_2(\xi_i|\xi) &= \sum_{\xi_j \in \Xi} p_1(\xi_i|\xi_j) \cdot p_1(\xi_j|\xi) \\ &\cdots \\ p_k(\xi_i|\xi) &= \sum_{\xi_j \in \Xi} p_1(\xi_i|\xi_j) \cdot p_{k-1}(\xi_j|\xi) \end{aligned} \tag{7}$$

Definition 1. Let $\xi \in \Xi$. We define the **influence** of a state ξ_i on the state ξ as:

$$I(\xi_i|\xi) = \sum_{k=0}^{\infty} p_k(\xi_i|\xi)$$

Similarly, let Σ be a subset of Ξ. We define the influence of a state ξ_i on the subset Σ as $I(\xi_i|\Sigma) = \sum_{\xi \in \Sigma} I(\xi_i|\xi)$.

We call **influencers** of a state ξ (respectively *of a subset* Σ), the set of states ξ_i that have a non-zero influence on ξ (respectively on Σ) (note, by definition, that all influences are non-negative).

8.3. *Some properties of the influence*

First, we notice that if all the times $\tau(\xi)$ are > 0, then *the influence is well defined* and is bounded by: $I(\xi_i|\xi) \leq \frac{1}{1-\gamma^{\tau_{\min}}}$ with $\tau_{\min} = \min_\xi \tau(\xi)$. Indeed, from the definition of the discounted chained-probabilities, we have $p_k(\xi_i|\xi) \leq \gamma^{k \cdot \tau_{\min}}$ thus: $I(\xi_i|\xi) \leq \sum_{k=0}^{\infty} \gamma^{k \cdot \tau_{\min}} = \frac{1}{1-\gamma^{\tau_{\min}}}$.

Moreover, the definition of the influence is related to the intuitive idea expressed above that **the influence $I(\xi_i|\xi)$ is the partial derivative of $V(\xi)$ by $R(\xi_i)$**:

$$I(\xi_i|\xi) = \frac{\partial V(\xi)}{\partial R(\xi_i)} \tag{8}$$

Proof: The Bellman equation is: $V(\xi) = R(\xi) + \sum_{\xi_i} p_1(\xi_i|\xi) \cdot V(\xi_i)$. By applying the Bellman equation to $V(\xi_i)$, we have:

$$V(\xi) = R(\xi) + \textstyle\sum_{\xi_i} p_1(\xi_i|\xi) \left[R(\xi_i) + \sum_{\xi_j} p_1(\xi_j|\xi_i) \cdot V(\xi_j)\right]$$

From the definition of p_2, we can rewrite this equation as:

$$V(\xi) = R(\xi) + \textstyle\sum_{\xi_i} p_1(\xi_i|\xi) \cdot R(\xi_i) + \sum_{\xi_i} p_2(\xi_i|\xi) \cdot V(\xi_i)$$

Again, we can apply the Bellman equation to $V(\xi_i)$ and easily prove the convergence at the limit:

$$V(\xi) = \textstyle\sum_{k=0}^{\infty} \sum_{\xi_i} p_k(\xi_i|\xi) \cdot R(\xi_i)$$

from which we deduce that the contribution of the reward at ξ_i to the VF at ξ is the influence of ξ_i on ξ:

$$\frac{\partial V(\xi)}{\partial R(\xi_i)} = \sum_{k=0}^{\infty} p_k(\xi_i|\xi) = I(\xi_i|\xi) \qquad \blacksquare$$

The VF at ξ is expressed as a linear combination of the rewards at states ξ_i weighted by the influences $I(\xi_i|\xi)$.

8.4. Computation of the influence

First, let us prove the following property: for any states ξ and ξ_i, we have

$$I(\xi_i|\xi) = \sum_{\xi_j} p_1(\xi_i|\xi_j) \cdot I(\xi_j|\xi) + \begin{cases} 1 & \text{if } \xi_i = \xi \\ 0 & \text{if } \xi_i \neq \xi \end{cases} \tag{9}$$

Proof: This result is easily deduced from the definition of the influence and the chained transition probability property (7):

$$\begin{aligned} I(\xi_i|\xi) &= \textstyle\sum_{k=0}^{\infty} p_k(\xi_i|\xi) = \sum_{k=0}^{\infty} p_{k+1}(\xi_i|\xi) + p_0(\xi_i|\xi) \\ &= \textstyle\sum_{k=0}^{\infty} \sum_{\xi_j} p_1(\xi_i|\xi_j) \cdot p_k(\xi_j|\xi) + p_0(\xi_i|\xi) \\ &= \sum_{\xi_j} p_1(\xi_i|\xi_j) \cdot I(\xi_j|\xi) + \begin{cases} 1 & \text{if } \xi_i = \xi \\ 0 & \text{if } \xi_i \neq \xi \end{cases} \end{aligned} \qquad \blacksquare$$

For a given ξ, let us define the operator Γ_ξ that, applied to any function ψ (defined on Ξ), returns: $\Gamma_\xi \psi(\xi_i) = \sum_{\xi_j} p_1(\xi_i|\xi_j) \cdot \psi(\xi_j)$

Equation (9) is equivalent to: $I(\xi_i|\xi) = \Gamma_\xi I(\cdot|\xi)(\xi_i) + \begin{cases} 1 & \text{if } \xi_i = \xi \\ 0 & \text{if } \xi_i \neq \xi \end{cases}$. This is not a Bellman equation since the sum of the probabilities $\sum_{\xi_j} p_1(\xi_i|\xi_j)$ may be greater than 1, so we cannot deduce that the successive iterations:

$$I_{n+1}(\xi_i|\xi) = \Gamma_\xi I_n(\cdot|\xi)(\xi_i) + \begin{cases} 1 & \text{if } \xi_i = \xi \\ 0 & \text{if } \xi_i \neq \xi \end{cases} \tag{10}$$

converge to the influence by using the classical contraction property of the operator Γ_ξ in max-norm (Puterman, 1994). However, by using the 1-norm, we have:

$$\begin{aligned} ||\Gamma_\xi \psi||_1 = \sum_{\xi_i} |\Gamma_\xi \psi(\xi_i)| &\leq \sum_{\xi_i} \sum_{\xi_j} |p_1(\xi_i|\xi_j) \cdot \psi(\xi_j)| \\ &\leq \gamma^{\tau_{\min}} \sum_{\xi_j} |\psi(\xi_j)| \leq \gamma^{\tau_{\min}} ||\psi||_1 \end{aligned}$$

thus Γ_ξ is a contractant operator in 1-norm. We deduce that the iterated values $I_n(\xi_i|\xi)$ in (10) satisfy

$$\begin{aligned} ||I_{n+1}(\cdot|\xi) - I(\cdot|\xi)||_1 &= \sum_{\xi_i} |\Gamma_\xi I_n(\cdot|\xi)(\xi_i) - \Gamma_\xi I(\cdot|\xi)(\xi_i)| \\ &= \sum_{\xi_i} |\Gamma_\xi [I_n(\cdot|\xi) - I(\cdot|\xi)](\xi_i)| \leq \gamma^{\tau_{\min}} ||I_n(\cdot|\xi) - I(\cdot|\xi)||_1 \end{aligned}$$

thus converge to the influence $I(\xi_i|\xi)$, unique solution of (9).

Remark. In order to compute the influence $I(\xi_i|\Omega)$ on a subset Ω, we use the iteration:

$$I_{n+1}(\xi_i|\xi) = \sum_{\xi_j} p_1(\xi_i|\Omega) \cdot I_n(\xi_j|\Omega) + \begin{cases} 1 & \text{if } \xi_i \in \Omega \\ 0 & \text{if } \xi_i \notin \Omega \end{cases} \tag{11}$$

which converge (similar proof) to $I(\xi_i|\Omega)$. The computation of the influence is thus cheap: equivalent to computing the value function of a discounted Markov chain.

Remark. As pointed out by Geoffrey Gordon, the influence is closely related to the *dual variables* (or *shadow prices* in economics) of the Linear Program equivalent to the Bellman equation (Gordon, 1999). This property has already been used in (Trick & Zin, 1993) to derive an efficient adaptive grid generation.

Remark. A possible extension is to define the **influence of a MDP** as the *infinitesimal* change in the value function of a state resulting from an *infinitesimal* modification of the reward at another state. Since the value function is a maximum of linear expressions, the influence on states with multiple optimal actions (thus for which the value function is not differentiable) is defined (as a set-valued map) by taking the partial sub-gradient instead of the regular gradient (8).

8.5. A tool to select out the most important areas

We would like to use the influence as a tool to discover what are the areas of the state space where we need a high quality interpolation process to obtain an accurate controller, so we could focus our refinement process there and neglect other areas.

The idea is that we want a high quality estimation of the VF around the areas of transition of the optimal control so that those switching boundaries be accurately located. Thus, the relevant areas of the state space are those that have an influence on the states around these switching boundaries.

Let us illustrate this idea on the "Car on the Hill" problem.

For any subset Ω, we can compute its influencers. As an example, figure 14 shows the influencers of 3 points.

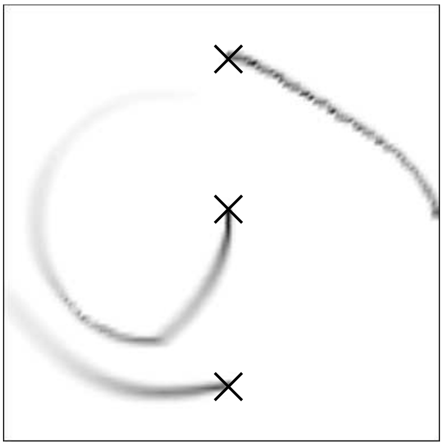
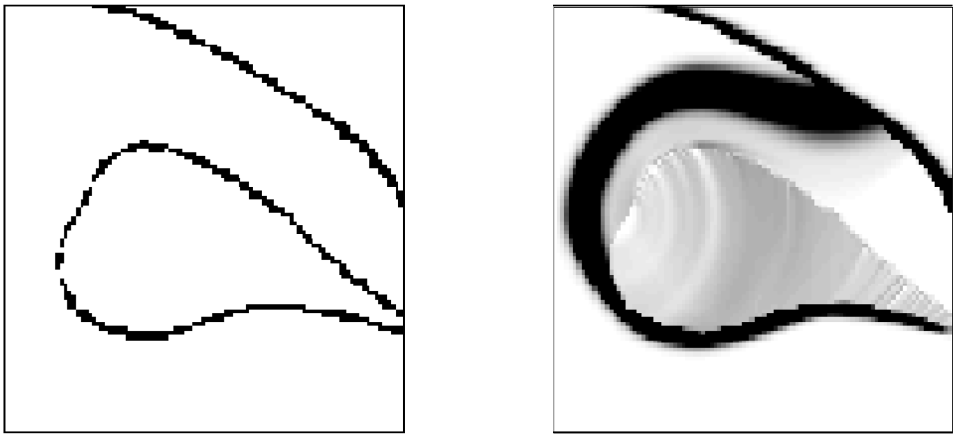


Figure 14. Influencers of 3 points (the crosses). The darker the gray level, the more important the influence. We notice that the influencers of a state "follow" some diffusion process in the direction of the optimal trajectory (see figure 6). This diffusion represents the stochasticity introduced by the discretization due to the averaging effect of the interpolation process.

First, for a given grid, let us define the subset Σ of the states of policy disagreement (in the sense of section 7.1). Figure 15(a) shows Σ for a regular grid of 129×129. Σ represents an estimation (given the current grid) of the optimal control switching boundaries.

Now we compute the influence on Σ (Figure 15(b)). The darkest zones show the states that influence the most the value function at Σ.

Consequently, if we were to increase the accuracy of the local interpolation process at the states illustrated by Figure 15(b) we would obtain a better approximation of the VF at the states shown in Figure 15(a), which would increase the precision of the switching boundaries, thus the performance of the controller.



(a) States of policy disagreement

(b) Influence on these states

Figure 15. The set of states of policy disagreement (a) and its influencers (b).

From this idea, we want to design a splitting heuristic that would take into account these non-local influences.

In order to decrease the local interpolation error, we first need to estimate, for a given grid, the VF approximation error caused by the accumulation of the interpolation errors due to the discretization.

In order to estimate this amount of stochasticity introduced by the interpolation process, we compute, in the next section, the *variance* of the future rewards for the discretized Markov chain.

9. Variance of a Markov chain

Again we consider the Markov chain resulting from the discretized MDP in which we choose the optimal policy π, and we use the same notations as in the previous section. Let $s(\xi) = (\xi(0) = \xi, \xi(1), \xi(2), ...)$ be an infinite sequence of states starting from an initial state ξ and generated by this Markov chain (the probability of transition from ξ to ξ' being $p(\xi'|\xi)$).

The *gain* $J(s(\xi))$ of a sequence $s(\xi)$ is the discounted cumulative rewards:

$$J(s(\xi)) = R(\xi) + \sum_{t\geq 1} \gamma^{\sum_{s=0}^{t-1} \tau(\xi(s))} R(\xi(t)) \tag{12}$$

and the VF of a state ξ is the expectation of this gain, for all possible sequences $s(\xi)$: $V(\xi) = E[J(s(\xi))]$.

The initial (continuous) control problem is deterministic, thus the VF of a state is simply the gain (defined by (2)) obtained along one optimal trajectory: the variance of the gain is zero. When this deterministic problem is discretized, the interpolation process produces an averaging effect that is mathematically equivalent to the introduction of stochasticity in the jumps from (discrete) state to state: the VF of the discretized MDP is an expectation of the gain (12) along all (discrete) optimal trajectories. Thus, the variance of the discrete MDP indicates the amount of averaging introduced during the discretization process.

The **variance** σ^2 of the gain is:

$$\sigma^2(\xi) = E\left[[J(s(\xi)) - V(\xi)]^2\right]$$

In order to compute this variance we first prove that the variance is solution to the Bellman equation:

$$\sigma^2(\xi) = \gamma^{2\tau(\xi)} \sum_{\xi'} p(\xi'|\xi) \cdot \sigma^2(\xi') + e(\xi) \tag{13}$$

with the one-step ahead contribution $e(\xi)$ defined as:

$$e(\xi) = \sum_{\xi'} p(\xi'|\xi) \cdot \left[\gamma^{\tau(\xi)} V(\xi') - V(\xi) + R(\xi)\right]^2 \tag{14}$$

Proof: The gain obtained along a sequence $s(\xi) = (\xi(0) = \xi, \xi(1), \xi(2), ...)$ satisfies $J(s(\xi)) = R(\xi) + \gamma^{\tau(\xi)} J(s(\xi(1)))$, with $s(\xi(1)) = (\xi(1), \xi(2), ...)$.

Thus the variance is:

$$\sigma^2(\xi) = E\left[[\gamma^{\tau(\xi)} J(s(\xi(1))) - (V(\xi) - R(\xi))]^2\right]$$

From the definition of the VF, $V(\xi) - R(\xi) = \gamma^{\tau(\xi)} E[V(\xi(1))] = \gamma^{\tau(\xi)} E[J(s(\xi(1)))]$, thus:

$$\sigma^2(\xi) = E\left[[\gamma^{\tau(\xi)} J(s(\xi(1)))]^2 - [V(\xi) - R(\xi)]^2\right]$$

Now, let us decompose this expectation using an average for all possible second states ξ' in the sequence, weighted by the probability of occurrence $p(\xi'|\xi)$:

$$\begin{aligned}
\sigma^2(\xi) &= \sum_{\xi'} p(\xi'|\xi) \cdot E\left[[\gamma^{\tau(\xi)} J(s(\xi'))]^2 - [V(\xi) - R(\xi)]^2\right] \\
&= \sum_{\xi'} p(\xi'|\xi) \cdot E\left[[\gamma^{\tau(\xi)} J(s(\xi'))]^2 - [\gamma^{\tau(\xi)} V(\xi')]^2\right] \\
&+ \sum_{\xi'} p(\xi'|\xi) \cdot E\left[[\gamma^{\tau(\xi)} V(\xi')]^2 - [V(\xi) - R(\xi)]^2\right]
\end{aligned} \tag{15}$$

Now, from the Bellman equation $V(\xi) = R(\xi) + \sum_{\xi'} p(\xi'|\xi) \cdot \gamma^{\tau(\xi)} V(\xi')$ we deduce that:

$$\sum_{\xi'} p(\xi'|\xi) \cdot E\left[[\gamma^{\tau(\xi)}V(\xi')]^2 - [V(\xi) - R(\xi)]^2\right] = e(\xi) \tag{16}$$

with $e(\xi)$ defined in (14). Moreover, we have:

$$E\left[[\gamma^{\tau(\xi)}J(\xi')]^2 - [\gamma^{\tau(\xi)}V(\xi')]^2\right] = \gamma^{2\tau(\xi)}E\left[[J(\xi') - V(\xi')]^2\right] = \gamma^{2\tau(\xi)} \cdot \sigma^2(\xi')$$

Which, combined with (16) in (15) gives (13). ■

Thus the variance $\sigma^2(\xi)$ is equal to the immediate contribution $e(\xi)$ that takes into account the variation in the values of the immediate successors ξ' plus the discounted expected variance $\sigma^2(\xi')$ of these successors.

The equation (13) is a Bellman equation: it is a fixed-point equation of a contractant operator (in max-norm) (with a contraction coefficient of $\gamma^{2\tau_{\min}}$) and thus can be solved by value iteration.

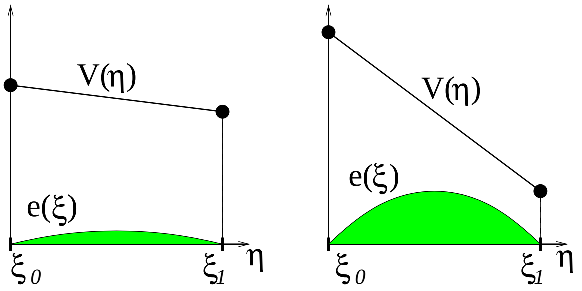


Figure 16. The term $e(\xi)$ as a function of the interpolated point η for low-(left) and high-(right) gradient value functions.

Remark. We can provide a geometrical interpretation of the term $e(\xi)$ related to the gradient of the value function at the iterated point $\eta = \eta(\xi, u^*)$ (see figure 3) and to the barycentric coordinates $\lambda_{\xi_i}(\eta)$. Indeed, from the definition of the discretized MDP (section 3.2), we have $V(\xi) = R(\xi) + \gamma^{\tau(\xi)}V(\eta)$ and from the piecewise linearity of the approximated functions we have $V(\xi_i) = V(\eta) + DV(\eta).(\xi_i - \eta)$, thus: $e(\xi) = \sum_{\xi_i} \lambda_{\xi_i}(\eta).\gamma^{2\tau(\xi)}[DV(\eta).(\xi_i - \eta)]^2$, which can be expressed as:

$$e(\xi) = \gamma^{2\tau(\xi)}.DV(\eta)^T.Q(\eta).DV(\eta)$$

with the matrix $Q(\eta)$ defined by its elements $q_{jk}(\eta) = \sum_{\xi_i} \lambda_{\xi_i}(\eta).(\xi_i - \eta)_j.(\xi_i - \eta)_k$. Thus, $e(\xi)$ is close to 0 in two specific cases: when the gradient at the iterated point η is low (i.e. the values are almost constant) and when η is close to a grid point ξ_i (then the barycentric coordinate λ_{ξ_i} is close to 1 and the other barycentric coordinates are close to 0, thus $Q(\eta)$ is low). In both cases, $e(\xi)$ is low and implies that the interpolation at ξ does not introduce a high degradation of the quality of approximation of the value function (the variance does not increase). Figure 16 shows $e(\xi)$ for a one-dimensional space.

Remark. The variance measures the amount of averaging accumulated by the interpolation process due to the discretization of the state space. Our basic assumption is that this measure is a good estimation of the approximation error of the VF, for a given grid. However, this may not be the case if the grid is too coarse so the policy of the discretized MDP differs too much from the optimal control of the continuous problem. Indeed, in that case, the variance would be computed along trajectories using a wrong policy. A detailed analysis of the estimation of the VF approximation error from local interpolation errors is initiated in (Munos & Moore, 2000).

Illustration of the variance for the "Car on the Hill"

Figure 17 shows the standard deviation $\sigma(\xi)$ for the "Car on the Hill" obtained with a uniform grid (of 257 by 257).

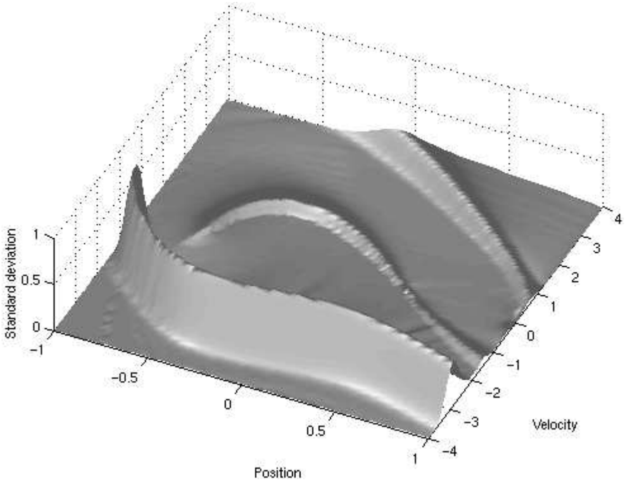


Figure 17. The standard deviation σ for the "Car on the Hill". We notice that it is very high around the frontier 1 (indeed, a discontinuity is impossible to approximate perfectly by discretization methods, whatever the resolution is) and noticeably high around frontiers 2 and 3, the discontinuities of the gradient of V (which correspond to boundaries of change in the optimal control, as shown in figure 6). Indeed, around these areas, the VF averages heterogeneous values of the discounted terminal rewards.

10. A global splitting heuristic

Now, we combine these notions of *influence* and *variance* in order to define a non-local splitting criterion. We have seen that:

- The states ξ of highest standard deviation $\sigma(\xi)$ are the states of lowest quality of approximation of the VF (figure 18(a)).
- The states ξ of highest influence on the set Σ of states of policy disagreement (figure 15(b)) are the states whose value function affects the area where there is a transition in the optimal control.

Thus, in order to improve the accuracy of approximation at the most relevant areas of the state space with respect to the controller (i.e. the optimal control switching boundaries), we split the states ξ of high standard deviation that have an influence on the areas of control transition, according to the *Stdev_Inf* criterion (see figure 18): $Stdev\_Inf(\xi) = \sigma(\xi).I(\xi|\Sigma)$. Figure 19 shows the discretization obtained by using this splitting criterion.

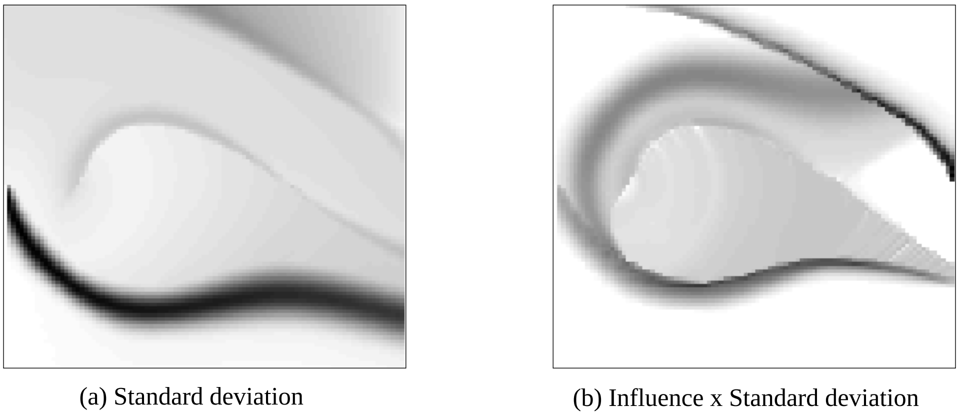


Figure 18. (a) The standard deviation $\sigma(\xi)$ for the "Car on the Hill" (equivalent to figure 17) and (b) The *Stdev_Inf* criterion, product of $\sigma(\xi)$ by the influence $I(\xi|\Sigma)$ (figure 15(b)).

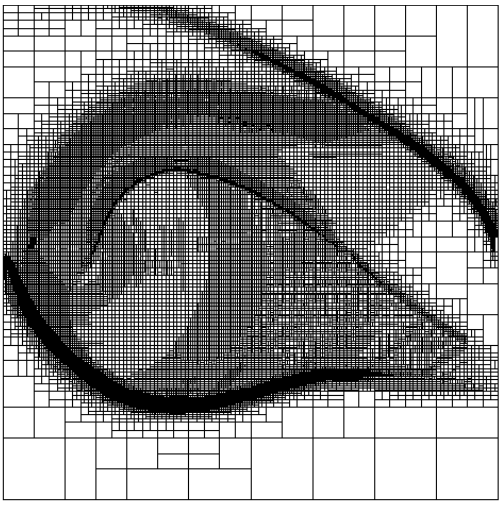


Figure 19. The discretization resulting from the *Stdev_Inf* split criterion. We observe that the upper part of frontier 1 is well refined. This refinement does not occur because we want to approximate the VF around its discontinuity (which was the case for the *corner-value difference* and *value non-linearity* criteria) but because the refinement there is needed to improve the quality of the controller at another area of the state space (the bottom part of frontier 2) where there is a switching boundary for the optimal control. We notice that the bottom part and the upper right part of the state space are not refined at all: it is not needed for the controller.

Remark. The performance of this criterion for the "Car on the Hill" problem are similar to those of combining the *value non-linearity* and the *policy disagreement* criterion. We didn't plot those performances in figure 13 for clarity reasons and because they do not represent a major improvement. However, the difference of performances between the local criteria and the *Stdev_Inf* criterion are much more significant in the case of higher dimensional problems, as illustrated in what follows.

It is important to notice the fact that the *Stdev_Inf* criterion does not split the areas where the VF is discontinuous unless some refinement is necessary to improve the quality of the controller (possibly at another part of the state space). As we will see in the simulations that follow, in higher dimensions, the cost to get an accurate approximation of a discontinuous VF is computationally very expensive, which explains why the splitting procedure using the *Stdev_Inf* criterion outperforms the previous refinement methods.

Remark. In the case of a stochastic process (Markov Diffusion Processes), we will need to reconsider this splitting heuristic since in that case the variance would reflect two components: the interpolation error introduced by the grid-approximation but also the intrinsic stochasticity of the continuous process. The latter is not relevant to our splitting method since a refinement around areas of high variance of the process will not result in an improvement of the approximations. This case will be further developed in future work.

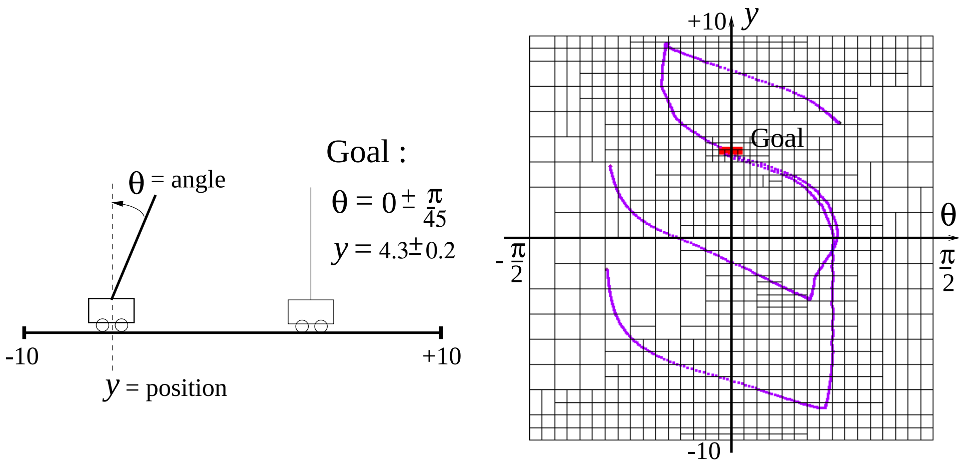
11. Illustration on more complex control problems

11.1. The Cart-Pole problem

The dynamics of this 4-dimensional physical system (illustrated in figure 20(a)) are described in (Barto, Sutton, & Anderson, 1983). In our experiments, we chose the following parameters as follows: the state space is defined by the position $y \in [-10, +10]$, angle $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, and velocities restricted to $\dot{y} \in [-4, 4]$, $\dot{\theta} \in [-2, 2]$. The control consists in applying a strength of ±10 Newton. The goal is defined by the area: $y = 4.3 \pm 0.2$, $\theta = 0 \pm \frac{\pi}{45}$, (and no limits on $\dot{y}$ and $\dot{\theta}$). This is a notably narrow goal to try to hit (see the projection of the state space and the goal on the 2d plan (y, θ) in figure 20). Notice that our task of "minimum time maneuver to a small goal region" from an arbitrary start state is much harder than merely balancing the pole without falling (Barto et al., 1983). The current reinforcement r is zero everywhere and the boundary reinforcement r_b is -1 if the system exits from the state space ($|y| > 10$ or $|\theta| > \frac{\pi}{2}$), and $+1$ if the system reaches the goal.

Figure 21 shows the performance obtained for several splitting criteria previously defined for this 4-dimensional control problem. We observe the following points:

- The local splitting criteria do not perform better than the uniform grids. The problem is that the VF is discontinuous at several parts of the state space (areas of high $|\theta|$ for which it is too late to re-balance the pole, which is similar to the frontier 1 of the "Car on the Hill" problem) and the value-based criteria spend too many resources on approximating these useless areas.



(a) The ‘‘Cart-pole’’

(b) The projection of the state space

Figure 20. (a) Description of the Cart-pole. (b) The projection of the discretization (onto the plane (θ, y)) obtained by the *Stdev_Inf* criterion and some trajectories for several initial points.

- The *Stdev_Inf* criterion performs very well. We observe that the trajectories (see figure 20(b)) are nearly optimal (the angle $|\theta|$ is maximized in order to reach the goal as fast as possible, and very close to its limit value, for which it is no more possible to recover the balance).

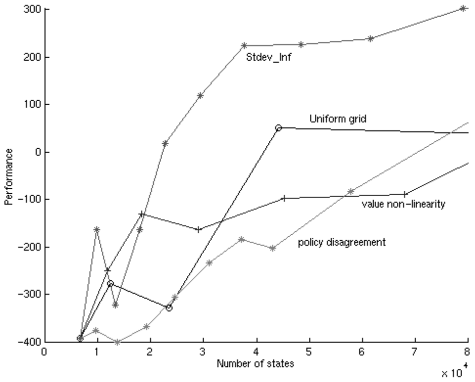


Figure 21. Performance on the “Cart-pole”.

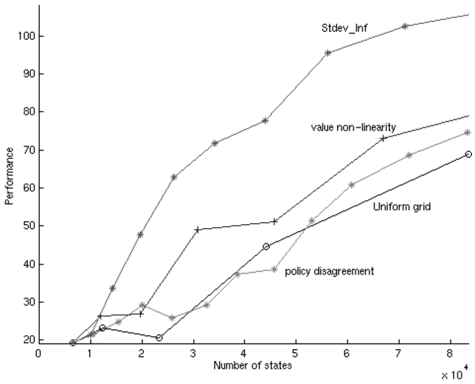


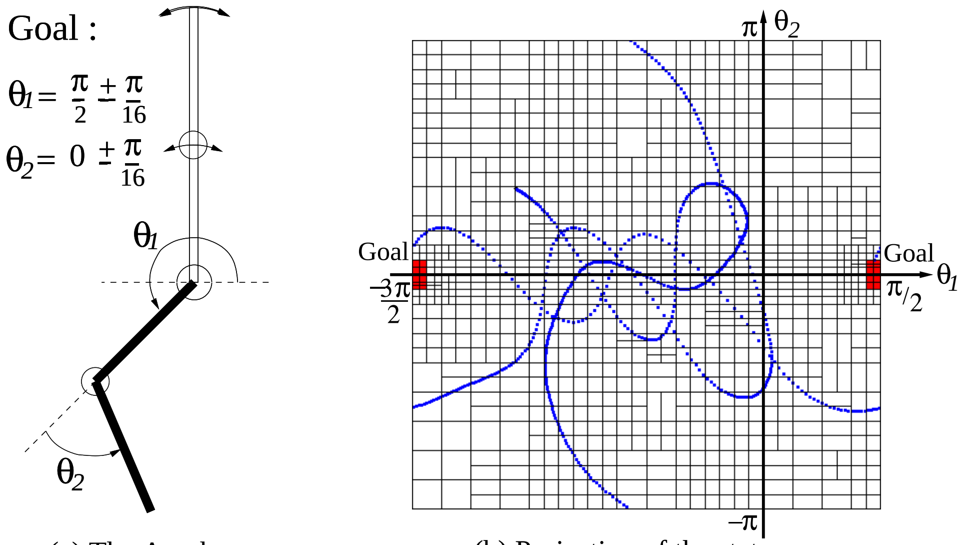
Figure 22. Performance on the Acrobot.

11.2. The Acrobot

The Acrobot is a 4-dimensional control problem which consists of a two-link arm with one single actuator at the elbow. This actuator exerts a torque between the links (see figure 23(a)). It has dynamics similar to a gymnast on a high bar, where Link 1 is analogous to the gymnast’s hands, arms and torso, Link 2 represents the legs, and the joint between the links is the gymnast's waist (Sutton, 1996). Here, the goal of the controller is to balance the Acrobot at its unstable, inverted vertical position, in the minimum time (Boone, 1997). The goal is defined by a very narrow range of $\frac{\pi}{16}$ on both angles around the vertical position $\theta_1 = \frac{\pi}{2}, \theta_2 = 0$ (figure 23(b)), for which the system receives a reinforcement of $r_b = +1$. Anywhere else, the reinforcement is zero. The two first dimensions (θ_1, θ_2) of the state space have a structure of a torus (because of the 2π modulo on the angles), which is implemented in our structure by having the vertices of 2 first dimensions being angle 0 and 2π pointing to the same entry for the value function in the interpolated kd-trie.

Figure 22 shows the performance obtained for several splitting criteria previously defined. The respective performance of the different criteria are similar to the "Cart-pole" problem above: the local criteria are no better than the uniform grids ; the *Stdev_Inf* criterion performs much better.

Figure 23(b) shows the projection of the discretization obtained by the *Stdev_Inf* criterion and one trajectory onto the 2d-plane (θ_1,θ_2).



(a) The Acrobot

(b) Projection of the state space

Figure 23. (a) Description of the Acrobot physical system. (b) Projection of the discretization (onto the plane (θ_1,θ_2)) obtained by the *Stdev_Inf* criterion, and one trajectory.

11.3. Brief description of two other control problems

The "space-shuttle" control problem

This is a 4-dimensional "space-shuttle" control problem defined by the position (x, y) and velocity (v_x, v_y) of a point (the shuttle) in a 2d-plane. There are 5 possible controls : do nothing or thrust to one of the 4 cardinal directions. The dynamics follow the laws of Newtonian physics where the shuttle is attracted by the gravitation of a planet (dark gray circle in figure 11.3) and some intergalactic dust (light gray circle). The goal is to reach some position in space (the square) by minimizing a cost (function of the time to reach the target and the fuel consumption). Figure 11.3 shows some trajectories.

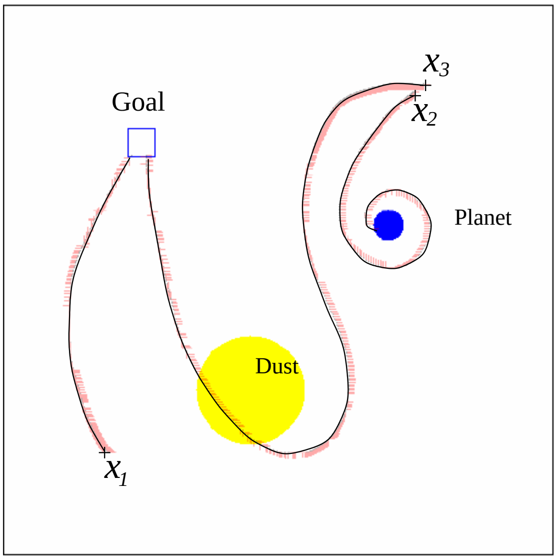
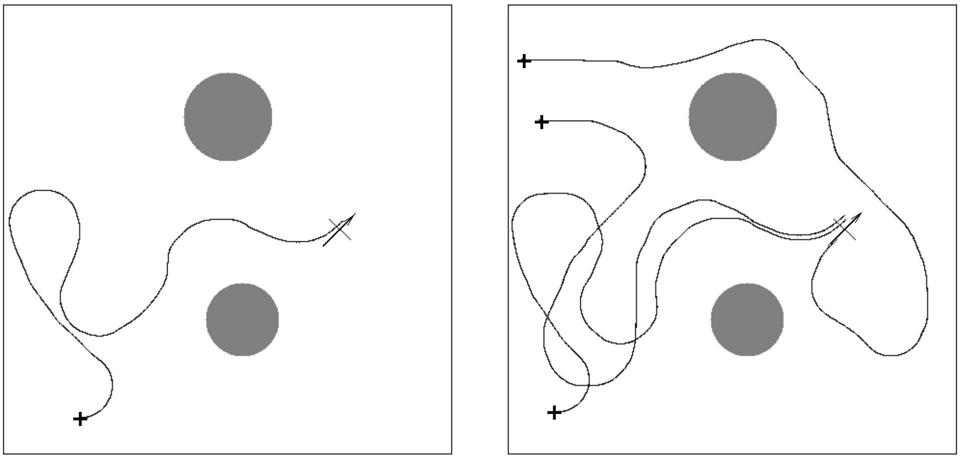


Figure 24. The "space-shuttle" trajectories for 3 different starting positions. From x_1 the goal is directly reachable (the gravitation is low). From x_2 the collision is unavoidable whatever the thrust (represented by small gray segments) to avoid the planet is. From x_3 the controller uses the gravitation forces to reach the goal.

The "airplane meeting" control problem

This is also a 4-dimensional control problem in which we consider one (or several) airplane(s) flying at constant altitude and velocity. They try to reach a target defined by a position x_G, y_G and an angle θ_G (the arrow in figure 25) at a *precise time* t_G. Each plane is defined at any time t by its position $x(t), y(t)$ and angle $\theta(t)$. There are 3 possible controls for each plane : turn left, right, or go straight. The state space is of dimension 4 : the position x, y, the angle θ and the time t. The dynamics are : $\frac{dx}{dt} = \cos(\theta)$, $\frac{dy}{dt} = \sin(\theta)$, $\frac{d\theta}{dt} = \{-1, 0, +1\}.v_\theta$ and $\frac{dt}{dt} = 1$. Here, the terminal cost is : $(x - x_G)^2 + (y - y_G)^2 + k_\theta(\theta - \theta_G)^2 + k_t(t - t_G)^2$ and there is a small constant current cost if a plane is in a gray area (some clouds that the planes should avoid). Figure 25 shows some trajectories for one and 3 planes when there is more time than necessary to reach the target directly (the planes have to loop).

Interpretation of the results: We notice that for the previous 4d problems, the local splitting criteria fail to improve the performance of the uniform grids because they spend too many resources on trying to approximate the discontinuities of the VF. For example, for the "Cart-pole" problem, the *value non-linearity* criterion focuses on approximating the VF mostly at parts of the state space where there is already no chance to re-balance the pole. And the areas around the vertical position (low θ), which are the most important areas, will not be refined in time (however, if we continue the simulations after about 90000 states, the local splitting criteria perform better than the uniform grids, because these important areas are eventually refined).



(a) One airplaine

(b) Meeting of 3 airplanes

Figure 25. The “airplane meeting” control problem

The *Stdev_Inf* criterion, which takes into account global consideration for the splitting, provides an accurate controller for all the tasks described above.

12. Conclusion and Future work

In this paper we proposed a variable resolution discretization approach to solve continuous time and space control problems. We described several local splitting criteria, based on the VF or the policy approximation. We observed that this approach works well for 2d problems like the “Car on the Hill”. However, for more complex problems, these local methods fail to perform better than uniform grids.

Local value-based splitting is an efficient, model-based, relative of the Q-learning-based tree splitting criteria used, for example, by (Chapman & Kaelbling, 1991; Simons, Van Brussel, De Schutter, & Verhaert, 1982; McCallum, 1995). But it is only when combined with new non-local measures that we are able to get truly effective, near-optimal performance on difficult control problems. The tree-based, state-space partitions in (Moore, 1991; Moore & Atkeson, 1995) were produced by different criteria (of empirical performance), and produced far more parsimonious trees, but no attempt was made to minimize cost: merely to find a valid path.

In order to design a global criterion, we introduced two useful measures of a Markov chain: the *influence* estimates the non-local dependencies in the VF, the *variance* estimates the VF error of approximation for a given grid. By combining these measures, we proposed an efficient splitting heuristic that exhibit good performance (in comparison to the uniform grids) on all the problems studied. These measures could also be used to solve large (discrete) MDPs by selecting which initial features (or categories) one has to refine to provide a relevant partition of the state space.

Another extension of these measures could be to learn them through interactions with the environment in order to design efficient exploration policies in reinforcement learning. Our notion of variance could be used with "Interval Estimation" heuristic (Kaelbling, 1993), to permit "optimism-in-the-face-of-uncertainty" exploration, or with the "back-propagation of exploration bonuses" of (Meuleau & Bourgine, 1999) for exploration in continuous state-spaces. Indeed, if we observe that the learned variance of a state ξ is high, then a good exploration strategy could be to inspect the states that have a high expected influence on ξ.

Even more parsimonious grid can be obtained if we only consider a controller for a specific area Ω of initial states. Indeed, the *Stdev_Inf* criterion can be computed with respect to $\Sigma_{|\Omega} = \{\xi \in \Sigma, I(\xi|\Omega) > 0\}$ (the areas of transition in the optimal control that have some influence on Ω) instead of Σ, in order to restrict the refinement process to the areas of the state space actually used by the trajectories.

Also, the notion of variance might be useful to provide a safe controller for which choosing a sub-optimal action would be preferable if it leads to states of lower variance than when taking the optimal action.

The more severe limitation to these discretization techniques (even with the variable resolution approach developed here) is still the curse of dimensionality. Currently, we were able to solve all 4-dimensional problems considered and a few 5-dimensional ones.

In the future, it seems important to develop the following points:

- A generalization process that could implement a bottom-up process for regrouping the areas (for example by pruning the tree) that have been over-refined.
- Consider the stochastic case, for which the computation of the VF approximation error (obtained by the measure of variance in the deterministic case) should only take into account the interpolation error and not the intrinsic noise of the process.
- Implement the same ideas on sparse representations that can handle high dimensions (and even in some case are able to break the curse of dimensionality), such as the sparse grids (Zenger, 1990; Griebel, 1998), the random and low-discrepancy grids (Niederreiter, 1992; Rust, 1996). In some early experiments using variable resolution random grids, we were able to solve stochastic problems in dimension six.

Acknowledgements to Jeff Schneider, Geoffrey Gordon, Nicolas Meuleau, Paul Bourgine, Csaba Szepesvári, Ron Parr, Andy Barto, and Leslie Kaelbling for their useful suggestions and advice.

References

Baird, L. C. (1995). Residual algorithms : Reinforcement learning with function approximation. *Machine Learning : proceedings of the Twelfth International Conference.*

Baird, L. C. (1998). Gradient descent for general reinforcement learning. *Neural InformationProcessing Systems, 11.*

Barles, G., & Souganidis, P. (1991). Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis, 4*, 271–283.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike Adaptive elements that that can learn difficult Control Problems. *IEEE Trans. on Systems Man and Cybernetics*, *13*(5), 835–846.

Barto, A. G., Bradtke, S. J., & Singh, S. P. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, pp. 81–138.

Bertsekas, D. P. (1987). *Dynamic Programming: Deterministic and Stochastic Models*. Prentice Hall.

Bertsekas, D. P., & Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Boone, G. (1997). Minimum-time control of the acrobot. *International Conference on Robotics and Automation*.

Boyan, J., & Moore, A. (1995). Generalization in reinforcement learning : Safely approximating the value function. *Advances in Neural Information Processing Systems*, *7*.

Chapman, D., & Kaelbling, L. P. (1991). Learning from Delayed Reinforcement In a Complex Domain. In *IJCAI-91*.

Crandall, M., Ishii, H., & Lions, P. (1992). User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, *27*(1).

Crandall, M., & Lions, P. (1983). Viscosity solutions of hamilton-jacobi equations. *Trans. of the American Mathematical Society*, *277*.

Crites, B., & Barto, A. (1996). Improving elevator performance using reinforcement learning. *Advances in Neural Information Processing Systems*, *8*.

Davies, S. (1997). Multidimensional Triangulation and Interpolation for Reinforcement Learning. In *Neural Information Processing Systems 9, 1996*. Morgan Kaufmann.

Dupuis, P., & James, M. R. (1998). Rates of convergence for approximation schemes in optimal control. *SIAM Journal Control and Optimization*, *360*(2).

Fleming, W. H., & Soner, H. M. (1993). *Controlled Markov Processes and Viscosity Solutions*. Applications of Mathematics. Springer-Verlag.

Friedman, J. H., Bentley, J. L., & Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, *3*(3), 209–226.

Gordon, G. (1995). Stable function approximation in dynamic programming. *Proceedings of the International Conference on Machine Learning*.

Gordon, G. J. (1999). *Approximate solutions to Markov Decision Processes*. Ph.D. thesis, CS department, Carnegie Mellon University, Pittsburgh, PA.

Griebel, M. (1998). Adaptive sparse grid multilevel methods for elliptic pdes based on finite differences. *Notes on Numerical Fluid Mechanics, Proceedings Large Scale Scientific Computations*.

Grüne, L. (1997). An adaptive grid scheme for the discrete hamilton-jacobi-bellman equation. *Numerische Mathematik*, *75-3*.

Kaelbling, L. P. (1993). *Learning in Embedded Systems*. MIT Press, Cambridge MA.

Knuth, D. E. (1973). *Sorting and Searching*. Addison Wesley.

Kushner, H. J., & Dupuis (1992). *Numerical Methods for Stochastic Control Problems in Continuous Time*. Applications of Mathematics. Springer-Verlag.

McCallum, A. (1995). Instance-Based Utile Distinctions for Reiforcement Learning with Hidden State. In *Machine Learning (proceedings of the twelfth international conference)* San Francisco, CA. Morgan Kaufmann.

Meuleau, N., & Bourgine, P. (1999). Exploration of multi-state environments: Local measures and back-propagation of uncertainty. *Machine Learning Journal, vol 35(2)*.

Moody, J., & Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, *1*, 289–303.

Moore, A. W. (1991). Variable Resolution Dynamic Programming: Efficiently Learning Action Maps in Multivariate Real-valued State-spaces. In Birnbaum, L., & Collins, G. (Eds.), *Machine Learning: Proceedings of the Eighth International Workshop*. Morgan Kaufmann.

Moore, A. W., & Atkeson, C. (1995). The parti-game algorithm for variable resolution reinforcement learning in multidimensional state space. *Machine Learning Journal*, *21*.

Moore, D. W. (1992). *Simplical Mesh Generation with Applications*. Ph.D. thesis, Cornell University.

Munos, R. (2000). A study of reinforcement learning in the continuous case by the means of viscosity solutions. *Machine Learning Journal*.

Munos, R., Baird, L., & Moore, A. (1999). Gradient descent approaches to neural-net-based solutions of the hamilton-jacobi-bellman equation. *International Joint Conference on Neural Networks*.

Munos, R., & Moore, A. (1998). Barycentric interpolators for continuous space and time reinforcement learning. *Advances in Neural Information Processing Systems*.

Munos, R., & Moore, A. W. (2000). Rates of convergence for variable resolution schemes in optimal control. *International Conference on Machine Learning*.

Niederreiter, H. (1992). Random number generation and quasi-monte carlo methods. *SIAM CBMS-NSF Conference Series in Applied Mathematics, Philadelphia*, *63*.

Puterman, M. L. (1994). *Markov Decision Processes, Discrete Stochastic Dynamic Programming*. A Wiley-Interscience Publication.

Rust, J. (1996). *Numerical Dynamic Programming in Economics*. In Handbook of Computational Economics. Elsevier, North Holland.

Simons, J., Van Brussel, H., De Schutter, J., & Verhaert, J. (1982). A Self-Learning Automaton with Variable Resolution for High Precision Assembly by Industrial Robots. *IEEE Trans. on Automatic Control*, *27*(5), 1109–1113.

Sutton, R. S. (1996). Generalization in reinforcement learning : Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*, *8*.

Tesauro, G. (1995). Temporal difference learning and td-gammon. *Communication of the ACM*, *38*, 58–68.

Trick, M. A., & Zin, S. E. (1993). A linear programming approach to solving stochastic dynamic programs. *Unpublished manuscript*.

Tsitsiklis, J., & Van Roy, B. (1996). An analysis of temporal difference learning with function approximation. *Technical report LIDS-P-2322, MIT*.

Zenger, C. (1990). Sparse grids. *Parallel Algorithms for Partial Differential Equations, Proceedings of the sixth GAMM-Seminar*.