Machine Learning, 1, 1–24 (1999)

Variable Resolution Discretization in Optimal Control

RÉMI MUNOS AND ANDREW MOORE {munos, awm}@cs.cmu.edu
Robotics Institute, Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh, PA 15213, USA

Received March 1, 1999

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 values of some other set of 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, but two dimensional problem of the "Car on the hill". It then evaluates the performance of a variety of splitting criteria on many benchmark problems (which we have published on the web), paying careful attention to their number-of-cells versus closeness-to-optimality tradeoff curves.

Keywords: Reinforcement learning, optimal control, variable resolution discretization

1. Introduction

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

Uniform discretizations (generally based on finite element or finite difference techniques (Kushner & Dupuis, 1992) provide us with important convergence results (see the analytical approach of (Barles & Souganidis, 1991; Crandall, Ishii, & Lions, 1992; Crandall & Lions, 1983; Munos, 1999) and the probabilistic results of (Kushner & Dupuis, 1992; Dupuis & James, 1998)), but suffer from impractical computational requirements when the size of the discretization step is small, especially when the state space is high dimensional. On the other hand, approximation methods (Bertsekas & Tsitsiklis, 1996; Baird, 1995; Sutton, 1996) can handle high

dimensionality but in general, have no guarantee of convergence to the optimal solution (Boyan & Moore, 1995; Baird, 1995; Munos, 1999). Some local convergence results are in (Gordon, 1995; Baird, 1998).

In this paper we try to keep the convergence properties of the discretized methods while introducing an approximation factor by the iterative designing of a variable resolution. In this paper we only consider the "general towards specific" approach : an initial coarse grid is successively refined at some areas of the state space by using a splitting process, until some desired approximation (of the value function or the optimal control) is reached.

First, we implement two splitting criteria based on the value function (see section 6), then we define a criterion of inconsistency between the value function and the policy (see section 7). In order to define the effect of the splitting of a state on others states, we define in section 8 the notion of *influence*. And we estimate the expected gain in the approximation of the value function when splitting states by defining in section 9 the *variance* of a Markov chain. By combining these two notions, we deduce, for a given discretization, the states whose splitting will mostly influence the parts of the state space where there is a change in the optimal control, leading to increase the resolution at those important areas.

We illustrate the different splitting criteria on the "Car on the hill" problem and in section 11 we show the results for other control problems, including the well known 4 dimensions "Cart-pole" and "Acrobot" problems.

In this paper we make the assumption that we have a model of the dynamics and of the reinforcement function. Moreover we assume that the dynamics is deterministic.

2. Description of the optimal control problem

We consider discounted deterministic control problems, which include the well-known reinforcement learning benchmarks of Car on the Hill (Moore, 1991), Cart-Pole (Barto, Sutton, & Anderson, 1983) and Acrobot (Sutton, 1996). Let $x(t) \in X$ be the *state* of the system, with the *state space* X be 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 discrete actions) by the differential equation, called *state dynamics* :

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

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 :

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

where $r(x, u)$ is the *current reinforcement* and $R(x)$ the *boundary reinforcement*. γ is the *discount factor* ($0 \leq \gamma < 1$). For convenience reasons, in what follows, we assume that $\gamma < 1$. However, most of the results apply to the undiscounted case $\gamma = 1$ as well, assuming that for any control $u(t)$, the trajectories do not loop (i.e. $x(t_1) \neq x(t_2)$ for $t_1 \neq t_2$).

The objective of the control problem is to find, for any initial condition x, the control $u^*(t)$ that optimizes 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 :

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

Following the DP principle, we can prove (Fleming & Soner, 1993) 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 $\pi(x) : X \to U$ 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 :

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

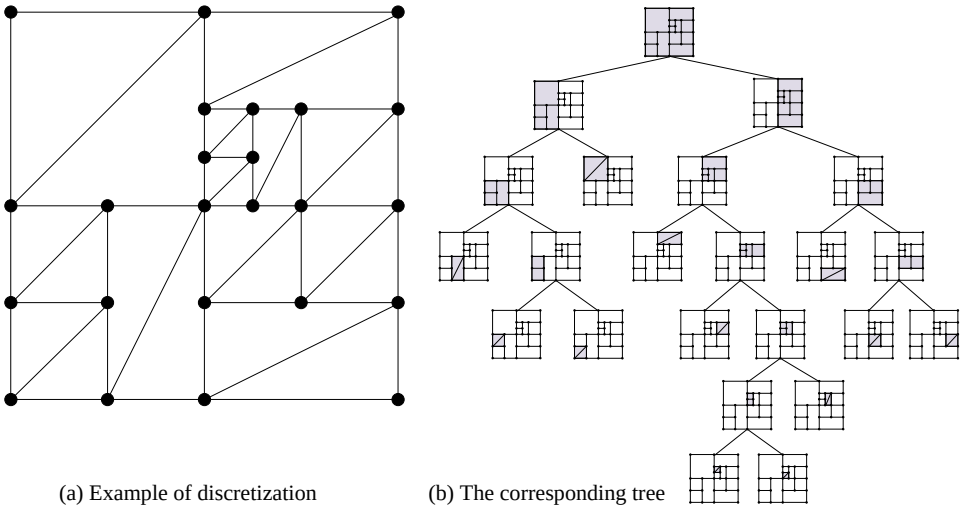
3. The discretization process

In order to discretize the continuous control problem described in the previous section, we use a process based on the finite element methods of (Kushner & Dupuis, 1992). We use a class of functions known as *barycentric interpolators* (Munos & Moore, 1998), built from a triangulation of the state-space. These functions are piecewise linear inside each simplex, but might be discontinuous at the boundary between two simplexes. 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 leaf ones) splits in some direction $i = 1..d$ the rectangle it covers at its middle into two nodes of half area (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.

For every leaf, we consider the 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 consider 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 rectangles 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

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 possible to use the feed-back rule 4 to deduce the corresponding optimal control.

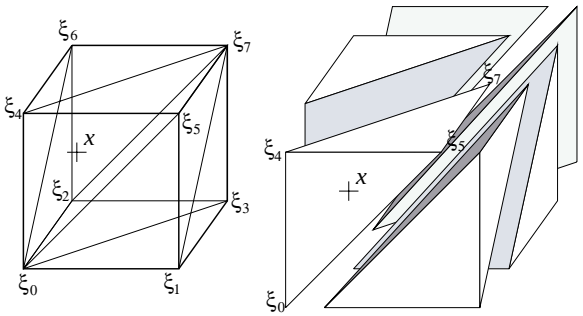


Figure 2. The Kuhn triangulation of a (3d) rectangle. The point x satisfying $1 \geq x_2 \geq x_0 \geq x_1 \geq 0$ is in the simplex $(\xi_0, \xi_4, \xi_5, \xi_7)$.

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 whose corners are $\{\xi_0, ..., \xi_{2^d}\}$. The indexes of the corners

uses 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 has this vertex, as well as $\xi_{2^d-1} = \xi_7$ in common), ξ_4 (we added 2^2), ξ_5 (we added 2^0) and ξ_7 (we added 2^1).

The corresponding barycentric coordinates $\lambda_0, ..., \lambda_d$ of the point inside the simplex $\xi_{i_0}, ..., \xi_{i_d}$ are : $\lambda_0 = 1 - x_{j_0}$, $\lambda_1 = x_{j_0} - x_{j_1}$, ..., $\lambda_k = x_{j_{k-1}} - x_{j_k}$, ..., $\lambda_d = x_{j_{d-1}} - 0 = x_{j_{d-1}}$. In the previous example, the barycentric coordinates are : $\lambda_0 = 1 - x_2$, $\lambda_1 = x_2 - x_0$, $\lambda_2 = x_0 - x_1$, $\lambda_3 = x_1$.

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

3.2. Building the discretized MDP

For a given discretization, we build a corresponding Markov Decision Process (MDP) in the following way. The **state space** of the MDP is the set Ξ of corners of the tree. The **control space** is the finite set U. For every corner $\xi \in \Xi$ and control $u \in U$ we approximate a part of the corresponding trajectory $x(t)$ (with the Euler or Runge-Kuta method) by integrating the state dynamics (1) from initial state ξ for a constant control u, during some time $\tau(\xi, u)$ until it enters inside a new rectangle at some point $\eta(\xi, u)$ (see Figure 3). At the same time, we also compute the integral of the current reinforcement :

$$R_{MDP}(\xi, u) = \int_{t=0}^{\tau(\xi,u)} \gamma^t . r(x(t), u) . dt$$

which defines the **reinforcement function** 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 defined by these barycentric coordinates (see Figure 3) : $p(\xi_i|\xi, u) = \lambda_{\xi_i}(\eta(\xi, u))$. Thus, the DP equation corresponding to this MDP is :

$$V(\xi) = \max_u \left[\gamma^{\tau(\xi,u)} . \sum_{i=0}^{d} \lambda_{\xi_i}(\eta(\xi, u)) . V(\xi_i) + R_{MDP}(\xi, u) \right] \quad (5)$$

If while integrating (1) from initial state ξ with the control u, the trajectory exits from the state space at time $\tau(\xi, u)$, then (ξ, u) 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 reinforcement : $R_{MDP} = \int_{t=0}^{\tau(\xi,u)} \gamma^t . r(x(t), u) . dt + \gamma^{\tau(\xi,u)} . R(x(\tau(\xi, u)))$.

3.3. Resolution of the discretized MDP

We can use any of the classical methods to solve the discretized MDP, i.e. *value iteration*, *policy iteration*, *modified policy iteration* (Puterman, 1994), (Bertsekas,

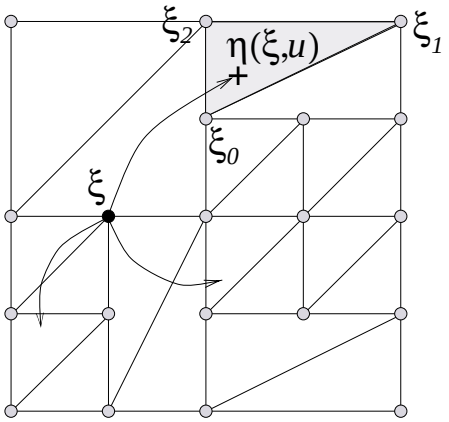


Figure 3. According to the current (variable resolution) grid, we build a discrete MDP. For every corner ξ (state of the MDP) and every control u, we integrate the corresponding trajectory until it enters a new rectangle at $\eta(\xi, u)$. The interpolated value at $\eta(\xi, u)$ is a linear combination of the values of the vertices of the simplex it is in (here (ξ_0, ξ_1, ξ_2)). Furthermore, it is a linear combination with positive coefficients that sum to one. Thus, doing this interpolation is mathematically equivalent to probabilistically jumping to a vertex. 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).

1987), (Barto, Bradtke, & Singh, 1995) or the *prioritized sweeping* (Moore & Atkeson, 1993).

4. An 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. In our experiments, we chose the reinforcement functions as follows : the current reinforcement $r(x, u)$ is zero everywhere. The terminal reinforcement $R(x)$ 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 with a null velocity (figure 4). The control u has only 2 possible values : maximal positive or negative thrust.

Figure 6 represents the interpolated value function of the MDP obtained by a regular discretization of 257 by 257 states.

We observe the following distinctive features of the value function :

- There is a discontinuity in the VF along the "Frontier 1" (see Figure 6) 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 exit from the left boundary (because the initial velocity is too negative), thus the corresponding value function is negative (see the optimal trajectories in Figure 5). 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 change in 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 down and do one loop to gain enough velocity to reach the top (see Figure 5). Moreover, we observe that around the lower part

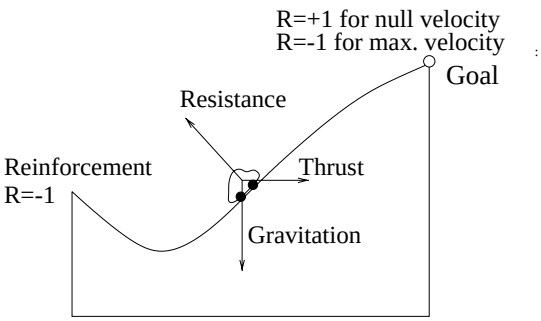


Figure 4. The "Car on the Hill" control problem.

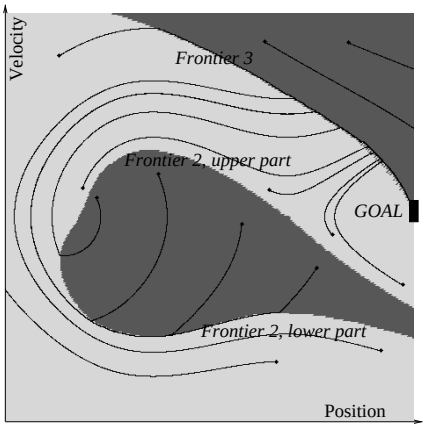


Figure 5. The optimal policy is indicated by different gray levels. Several optimal trajectories are drawn for different initial starting points.

of frontier 2 (see Figures 6), there is no visible discontinuity 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 reward as fast as possible, whereas above, it decelerates to reach the top of the hill with the lowest velocity and receive the highest reward).

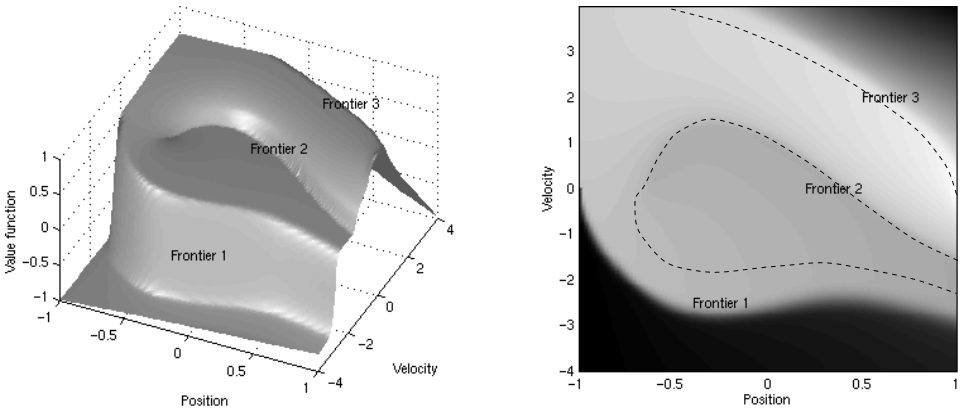


Figure 6. The value function of the Car-on-the-Hill problem obtained by a regular grid of 257 by 257 = 66049 states. The Frontier 1 illustrates the discontinuity of the VF, the Frontiers 2 and 3 (the dash lines) stands where there is a change in the optimal control.

We deduce from these observations that a good approximation of the value function does not necessarily mean a good approximation of the optimal control since :

- The approximation of the value function is not *sufficient* to predict the change in the optimal control around the lower part of frontier 2.
- A good approximation of the value function is not *necessary* around the frontier 1 since there is no change in the optimal control.

5. The variable resolution approach

The idea is to start with an initial coarse discretization, build the corresponding MDP, solve it in order to have a (coarse) approximation of the value function, then, locally refine the discretization by splitting some cells according to the process :

1. Score each cell and each direction i according to how promising it is to split according to some measure, called *split-criterion(i)*.
2. Pick the top f% (where f is a parameter) of the highest scoring cells.
3. Split them along the direction given by argmax_i split-criterion(i). Use the dynamics and reward model to create a new (larger) discretized MDP (see the splitting process in Figure 7). Note that only the cells that were split, and those whose successive states involve a split cell need to have their state transition recomputed.
4. Go to step 1 until we estimate that the approximation of the value function or the optimal control is precise enough.

Thus, the central purpose of this paper is the study of several splitting criteria.

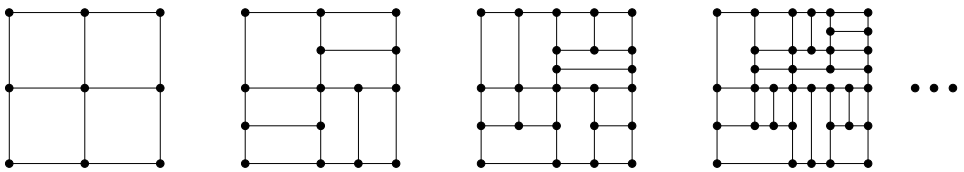


Figure 7. Several discretizations resulting of successive splitting operations.

Remark. In this paper, we only consider a "general towards specialized" process in the sense that the discretization is always refined. We could also consider some "generalization" process where, for example the tree coding for the discretization could be pruned, in order to avoid non relevant partitioning into too small subsets.

In what follows, we present several local splitting criteria and illustrate the resulting discretizations on the previous "Car on the Hill" control problem.

6. Criteria based on the value function

6.1. First criterion : average corner-value difference

For every rectangle, we compute the average of the absolute difference of the values at the corners of the edges for all directions $i = 0..d-1$. Let us denote $Ave(i)$ this criterion for direction i. For example, consider the square described in Figure 2. Then this *split-criterion* is :

$$\begin{aligned} Ave(0) &= \tfrac{1}{4}[|V(\xi_1) - V(\xi_0)| + |V(\xi_3) - V(\xi_2)| + |V(\xi_5) - V(\xi_4)| + |V(\xi_7) - V(\xi_6)|] \\ Ave(1) &= \tfrac{1}{4}[|V(\xi_2) - V(\xi_0)| + |V(\xi_3) - V(\xi_1)| + |V(\xi_6) - V(\xi_4)| + |V(\xi_7) - V(\xi_5)|] \\ Ave(2) &= \tfrac{1}{4}[|V(\xi_4) - V(\xi_0)| + |V(\xi_5) - V(\xi_1)| + |V(\xi_6) - V(\xi_2)| + |V(\xi_7) - V(\xi_3)|] \end{aligned}$$

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 50% of the rectangles at each iteration.

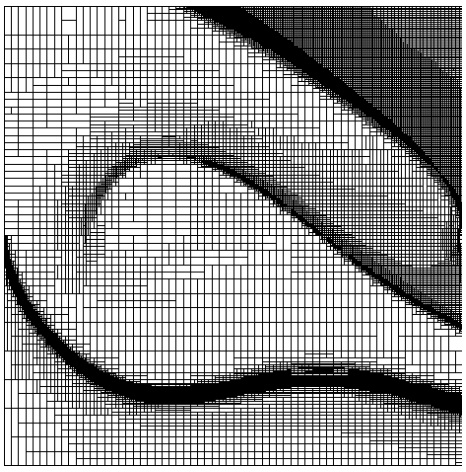


Figure 8. The discretization of the state space for the "Car on the Hill" problem using the *corner-value difference* criterion.

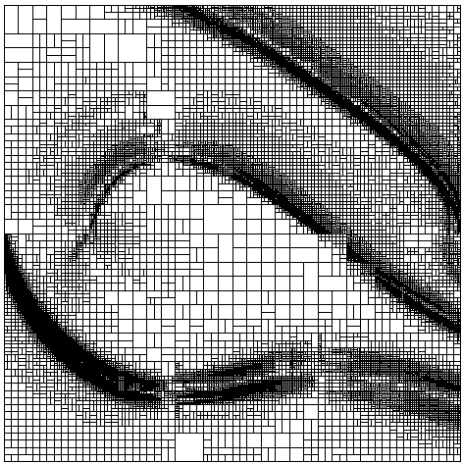


Figure 9. The discretization of the state space for the "Car on the Hill" problem using the *value non-linearity* criterion.

6.2. Second criterion : value non-linearity

For every rectangle, 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 leads to reduce the variation of the values, and *splits wherever the value function is not constant.* Figure 10(a)&(b) shows a (1-dimension) cut of a discontinuity and

the corresponding discretization and approximation obtained using the *corner-value difference* split criterion.

- The *value non-linearity* criterion leads to reduce the change of variation of the values, thus *splits wherever the value function is not linear*. So this criterion will also concentrate on the same 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*. See, for example, the difference of splitting in the area above the frontier 3.
  - The discretization resulting of the split of a discontinuity by the *corner-value difference* and the *value non-linearity* criteria are different (see Figure 10). The *value non-linearity* criterion splits where the approximated function (here some kind of sigmoid function whose slope depends on the density of the resolution) is the least linear (Figure 10(c)&(d)). This explains the 2 parallel tails observed around the frontiers (mainly the right part of frontier 1) in Figure 9.
- The refinement process does not split around the bottom part of frontier 2 although there is a change in the optimal control (because the VF is almost constant in this area). Moreover, there is a huge amount of memory spent for the approximation of the discontinuity (frontier 1) although the optimal control is constant in this area.

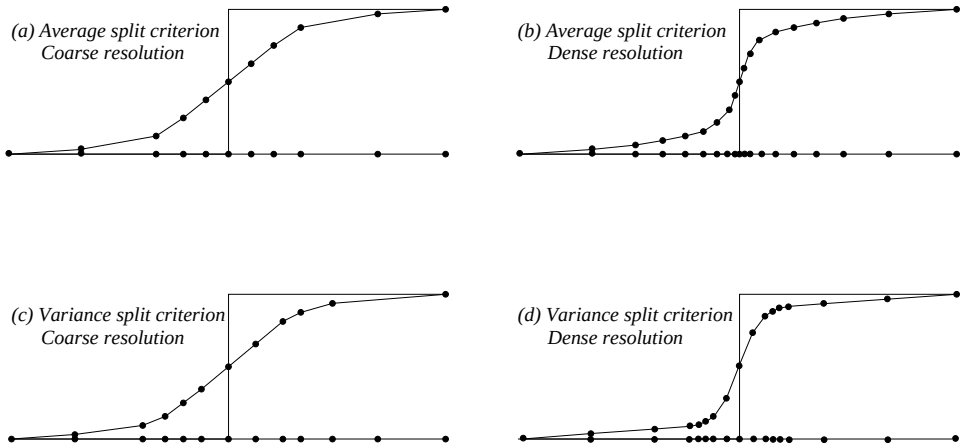


Figure 10. The discretization around a discontinuity resulting of the *corner-value difference* (a)&(b) and the *value non-linearity* (c)&(d) split criterion, for a coarse (a)&(c) and a dense (b)&(d) resolution.

Thus we can wonder if it is really useful to split so much around frontier 1, knowing that it will not result in an improved policy ?

Next section introduces a new criterion which takes into account the policy.

7. A criterion based on the policy

Figure 5 shows the optimal policy and several optimal trajectories for different starting points. We would like to define a refinement process that could refine around the areas of change in the optimal control, that is around frontier 2 (upper and lower parts) and 3, but not around frontier 1. In what follows, we propose 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 $\xi \in \Xi$:

$$\pi(\xi) \in \arg\max_{u \in U} \left[\gamma^{\tau(\xi,u)} . \sum_{i=0}^{d} \lambda_{\xi_i}(\eta(\xi,u)) . V(\xi_i) + R_{MDP}(\xi,u) \right] \tag{6}$$

and we can compare it with the optimal control law (4) derived from the gradient of V. The policy disagreement criterion compares the control derived from the local gradient of V (4) with the control derived from the policy of the MDP (6).

Remark. Instead of computing the gradient DV for all the $(d!)$ simplexes in the rectangles, 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 approximated 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 the rectangles where these two measures of the optimal control diverge.

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 well approximated, thus this process may converge to a sub-optimal performance. Indeed, we can observe that on Figure 11, the bottom part of frontier 2 is (lightly) situated higher than its optimal position, illustrated on Figure 5. This results in an underestimation of the value function at this area because of the lack of precision around the discontinuity (frontier 1). In section 7.3, we will observe that the performance of the discretization resulting from this splitting criterion is relatively weak.

However, this splitting criterion can be beneficially combined with the previous ones based on the approximation of 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 take 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 change in the optimal control. We can combine

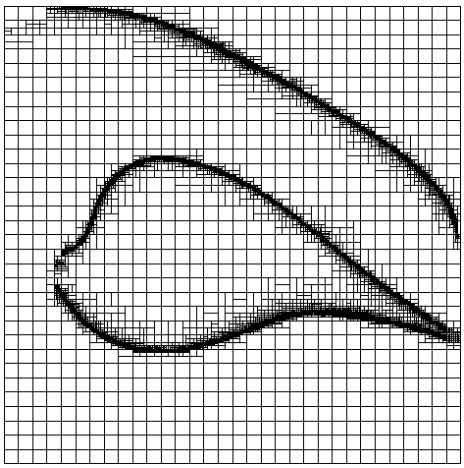


Figure 11. The discretization of the state space using the *policy disagreement* criterion. Here we used an initial grid of 33×33 and a splitting rate of 20%.

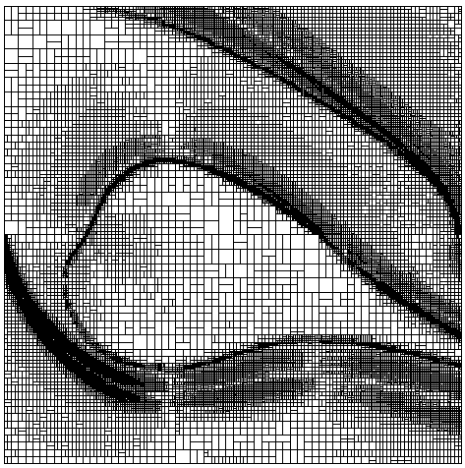


Figure 12. The discretization of the state space for the "Car on the Hill" problem using the combination of the *value non-linearity* and the *policy disagreement* criterion.

the previous criteria in several ways, for example by a weighted sum of the respective criteria, by a logical operation (split if an and/or combination of these criteria is satisfied), by an ordering of the criteria (first split with one criterion, then use an other one), etc.

Figure 12 shows the discretization obtained by alternatively, between iterations, using the *value non-linearity* criterion (to obtain a good approximation of the value function) and the *policy disagreement* criterion (to increasing the accuracy around the area of change in 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 (using the feed-back control law (4)) starting from initial states regularly situated in the state space. The *performance* of a discretization is the sum of the cumulated reinforcement (the gain defined by equation (2)) obtained by 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.

In this 2 dimensional control problem, the variable resolution approach perform much better (except for the *policy disagreement* criterion alone) than the uniform grids. However, as we will see later, for higher dimensional problems, the ressources allocated to approximate the discontinuities of the VF in areas not useful for improving the optimal control might be prohibitely high.

Can we do better ?

So far, we have only considered local splitting criteria, in the sense that we decide

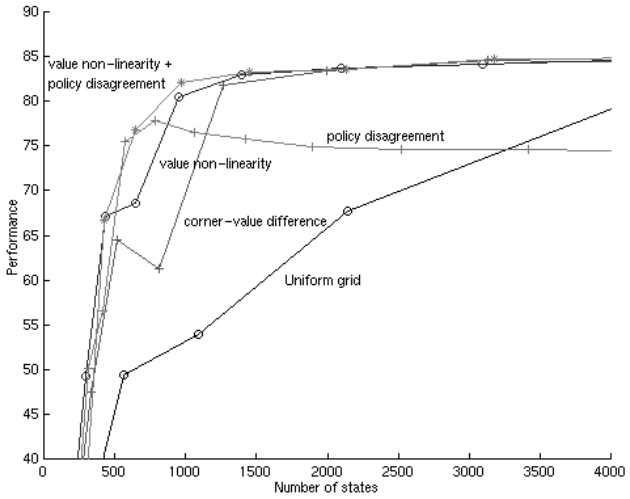


Figure 13. The performance for the uniform versus variable resolution grids for several splitting criterion. Both the *corner-value difference* and *value non-linearity* splitting processes perform better than the uniform grids. The *policy disagreement* splitting is very good for a small number of states 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.

whether or not to split a rectangle according to information (value function and policy) relative to the rectangle itself. However, the effect of the splitting is not local : it has an influence on the whole state space.

We will thus try to see if it is possible to find a better refinement process that could split an area if and only if it is useful to improve the performance. Sections that follow presents two notions which will be useful for defining such a global splitting criterion : the **influence**, which measures the extend to which local changes in some state effect the global VF, and the **variance**, which measures how accurate the current approximated VF is.

8. Notion of influence

Let us consider the Markov chain resulting from the choice of the control for the optimal policy $u^* = \pi^*(\xi)$ of the MDP. For convenience reasons, let us denote $R(\xi) = R_{MDP}(\xi, \pi^*(\xi))$, $p(\xi_i|x) = p(\xi_i|\xi, \pi^*(\xi))$, $\tau(\xi) = \tau(\xi, \pi^*(\xi))$.

8.1. Intuitive idea

The intuitive idea of the influence $I(\xi_i|\xi)$ of a state ξ_i on another state ξ is to give a measure of to what extend the VF of state ξ_i "contributes" to the VF of state ξ, i.e. the change in the VF at ξ resulting from a modification of the VF at ξ_i.

But the VF is the solution of a Bellman equation which depends on the structure of the Markov chain and the reinforcement values, thus we cannot modify the value at some state ξ_i, without violating Bellman equation.

However, we notice that the value function at state ξ_i is affected linearly by the reinforcement obtained at state ξ, thus we can compute the "contribution" of state ξ_i to state ξ by estimating the change in the VF at ξ resulting from a modification of the reinforcement $R(\xi_i)$.

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 :

$$\begin{aligned} p_0(\xi_i|\xi) &= 1 \text{ (if } \xi = \xi_i) \text{ or } 0 \text{ (if } \xi \neq \xi_i) \\ p_1(\xi_i|\xi) &= \gamma^{\tau(\xi)} p(\xi_i|\xi) \\ p_2(\xi_i|\xi) &= \textstyle\sum_{\xi_j \in \Xi} p_1(\xi_i|\xi_j).p_1(\xi_j|\xi) \\ &\dots \\ p_k(\xi_i|\xi) &= \textstyle\sum_{\xi_j \in \Xi} p_1(\xi_i|\xi_j).p_{k-1}(\xi_j|\xi) \end{aligned} \tag{7}$$

Definition 1. Let $\xi \in \Xi$. We define the **influence** of a state ξ_i on the state ξ being the quantity : $I(\xi_i|\xi) = \sum_{k=0}^{\infty} p_k(\xi_i|\xi)$. Let Σ be a subset of Ξ. We define the influence of a state ξ_i on the subset Σ being the quantity : $I(\xi_i|\Sigma) = \sum_{\xi \in \Sigma} I(\xi_i|\xi)$.

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 (resp. on Σ) (note, by definition, that all influences are non-negative).

8.3. Some properties of the influence

First, we notice that if 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 $p_k(\xi_i|\xi) \leq \gamma^{k.\tau_{\min}}$ thus : $I(\xi_i|\xi) \leq \sum_{k=0}^{\infty} \gamma^{k.\tau_{\min}} = \frac{1}{1-\gamma^{\tau_{\min}}}$.

Moreover, we can relate the definition to the intuitive idea previously stated and the following properties hold :

- The influence $I(\xi_i|\xi)$ is the partial derivative of $V(\xi)$ by $R(\xi_i)$: $I(\xi_i|\xi) = \frac{\partial V(\xi)}{\partial R(\xi_i)}$.
- For any states ξ and ξ_i, we have :

$$I(\xi_i|\xi) = \sum_{\xi_j} p_1(\xi_i|\xi_j).I(\xi_j|\xi) + \begin{cases} 1 & \text{if } \xi_i = \xi \\ 0 & \text{if } \xi_i \neq \xi \end{cases} \tag{8}$$

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

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

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

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

Again we can apply Bellman equation to $V(\xi_i)$ and finally deduce that :

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

Thus the contribution of $R(\xi_i)$ to $V(\xi)$ is : $\frac{\partial V(\xi)}{\partial R(\xi_i)} = \sum_{k=0}^{\infty} p_k(\xi_i|\xi) = I(\xi_i|\xi)$.

Property (8) is easily deduced from the very definition of the influence and the chained probability property (7), since for all ξ,

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

8.4. Computation of the influence

Equation (8) 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 :

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

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

$$\begin{aligned} \sum_{\xi_i} I(\xi_i|\xi) &= \sum_{\xi_i} \sum_{\xi_j} p_1(\xi_i|\xi_j).I(\xi_j|\xi) + 1 \\ &= \sum_{\xi_j} \gamma^{\tau(\xi_j)}.I(\xi_j|\xi) + 1 \end{aligned}$$

Thus, by denoting $I(\Xi|\xi)$ the vector whose components are the $I(\xi_i|\xi)$ and by introducing the 1-norm $||I(\Xi|\xi)||_1 = \sum_i |I(\xi_i|\xi)|$, we deduce that :

$$||I_{n+1}(\Xi|\xi) - I(\Xi|\xi)||_1 \leq \gamma^{\tau_{\min}}.||I_n(\Xi|\xi) - I(\Xi|\xi)||_1$$

and we have the contraction property in the 1-norm which insures convergence of the iterated $I_n(\xi_i|\xi)$ to the unique solution (the fixed point) $I(\xi_i|\xi)$ of (8).

8.5. Illustration 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 some 3 points.

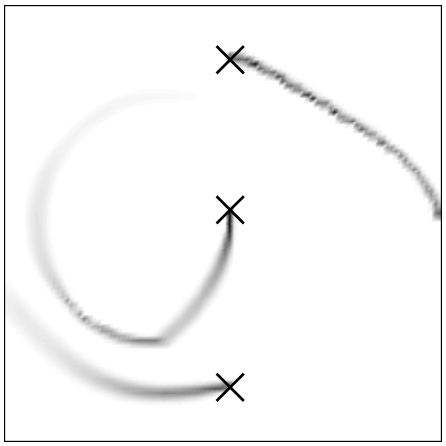
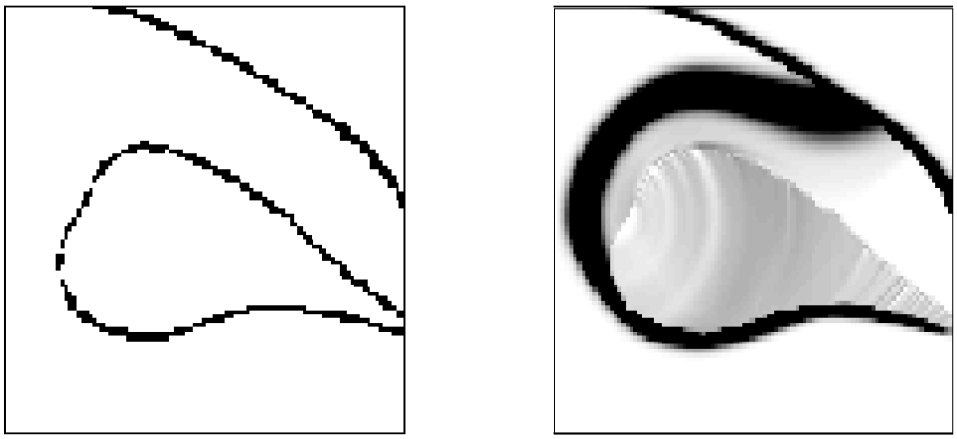


Figure 14. Influencers of 3 points (the crosses). The darker the gray level, the more important the influence. We notice that the influence of a state "follows" the direction of the optimal trajectory starting from that state (see figure 5) through some kind of "diffusion process".

Let us define the subset Σ to be those states of policy disagreement (in the sense of section 7.1). Figure 15(a) shows Σ for a regular grid of 129×129. The influencers of Σ is computed and plotted in Figure 15(b). The darkest zones in Figure 15(b) are the places whose splitting will most affect the value function at the places (illustrated in Figure 15(a)) of change in the optimal control.



(a) States of policy disagreement (b) Influence of these states

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

Now we would like to define the areas whose refinement could lead to the highest quantitative change on the value function. This is closely related to the quality of the approximation of the value function for a given discretization.

Indeed, the better the approximation, the lower a change in the value function may result from a splitting. In the following section, we introduce the *variance* of the Markov chain in order to estimate the quality of approximation of the VF for the current discretization, thus defining the areas whose splitting may lead to the highest change in the VF.

9. Variance of a Markov chain

By using the notation of the previous section, Bellman equation states that : $V(\xi) - R(\xi) = \gamma^{\tau(\xi)} \cdot \sum_{\xi_i \in X} p(\xi_i|\xi) V(\xi_i)$, thus $V - R$ at ξ is a discounted average of the next values $V(\xi_i)$ weighted by the probabilities of transition $p(\xi_i|\xi)$. We are interested in computing the *variance of the next values* in order to get an estimation of the range of the values averaged by $V(\xi)$. The first idea is to compute the one-step-ahead variance $e(\xi)$ of $V - R$:

$$e(x) = \sum_{\xi_i \in X} p(\xi_i|\xi) \left[\gamma^{\tau(\xi)} V(\xi_i) - V(\xi) + R(\xi)\right]^2 \qquad (10)$$

However, the values $V(\xi_i)$ also average some successive values $V(\xi_j)$, so we would like that the variance also takes into account this second-step-ahead average, as well as all the following ones. The next sections define the value function as an averager of the reinforcements and present the definition of the variance of a Markov chain.

9.1. The value function as an averager of the reinforcements

Let us denote $s_k(\xi)$ a sequence of $(k+1)$ states $(\xi_0 = \xi, \xi_1, \xi_2, ..., \xi_k)$ whose first one is ξ. Let $S_k(\xi)$ be the set of all possible sequences $s_k(\xi)$. Let $p(s_k(\xi)) = \prod_{i=1}^{k} p(\xi_i|\xi_{i-1})$, the product of the probabilities of transition of the successive states in a sequence, and for $i \leq k$, let $\tau_i(s_k(\xi)) = \sum_{j=0}^{i-1} \tau(\xi_j)$ the cumulated time of the i^{th} first states of the sequence (with by definition $\tau_0(s_k(\xi)) = 0$).

We have the following property : $\sum_{s_k(\xi) \in S_k(\xi)} p(s_k(\xi)) = 1$. We can prove that the value function satisfies the following equation (similar to the Bellman equation but for k-steps-ahead) : for any k,

$$V(\xi) = \sum_{s_k(\xi) \in S_k(\xi)} p(s_k(\xi)) \left[\gamma^{\tau_k(s_k(\xi))} V(\xi_k) + \sum_{i=0}^{k-1} \gamma^{\tau_i(s_k(\xi))} R(\xi_i) \right]$$

Let us denote $s_\infty(\xi)$ an infinite sequence of states starting with ξ, and $S_\infty(\xi)$ the set of all possible such sequences. Define $p(s_\infty(\xi)) = lim_{k \to \infty} p(s_k(\xi))$, and $\tau_i(s_\infty(\xi))$ is defined as previously for any $i \geq 0$. Then we still have the property : $\sum_{s_\infty(\xi) \in S_\infty(\xi)} p(s_\infty(\xi)) = 1$, and the value function satisfies :

$$V(\xi) = \sum_{s_\infty(\xi) \in S_\infty(\xi)} p(s_\infty(\xi)) \left[\sum_{i=0}^{\infty} \gamma^{\tau_i(s_\infty(\xi))} R(\xi_i) \right] \tag{11}$$

9.2. Definition of the Variance of a Markov chain

Intuitively, the variance of a state ξ is a measure of how dissimilar the cumulative future reward obtained along all possible trajectories starting from ξ are. More precisely, we define it as the variance of the quantities averaged in equation (11) :

Definition 2. Let $\xi \in \Xi$. We define the **variance** σ^2 of the Markov chain at ξ :

$$\sigma^2(\xi) = \sum_{s_\infty(\xi) \in S_\infty(\xi)} p(s_\infty(\xi)) \left[\sum_{i=0}^{\infty} \gamma^{\tau_i(s_\infty(\xi))} R(\xi_i) - V(\xi) \right]^2$$

Let us prove that the variance satisfies a Bellman equation. By selecting out the $i = 0$ case in the summation, we have :

$$\sigma^2(\xi) = \sum_{s_\infty(\xi) \in S_\infty(\xi)} p(s_\infty(\xi)) \left[\sum_{i=1}^{\infty} \gamma^{\tau_i(s_\infty(\xi))} R(\xi_i) - (V(\xi) - R(\xi)) \right]^2$$

and from (11), we deduce that :

$$\begin{aligned} \sigma^2(\xi) \;=\; & \sum_{s_\infty(\xi)} p(s_\infty(\xi)) \left\{ \left[\sum_{i=1}^{\infty} \gamma^{\tau_i(s_\infty(\xi))} R(\xi_i) \right]^2 - \left[\gamma^{\tau(\xi)} V(\xi_1) \right]^2 \right. \\ & \left. + \left[\gamma^{\tau(\xi)} V(\xi_1) \right]^2 - \left[V(\xi) - R(\xi) \right]^2 \right\} \end{aligned}$$

By successively applying (11) to ξ_1 and by selecting out the state ξ_i in the sequence s_∞, we obtain :

$$\sigma^2(\xi) = \sum_{s_\infty(\xi)} p(s_\infty(\xi)) \left\{ \left[\sum_{i=1}^{\infty} \gamma^{\tau_i(s_\infty(\xi))} R(\xi_i) - \gamma^{\tau(\xi)} V(\xi_1) \right]^2 + \left[\gamma^{\tau(\xi)} V(\xi_1) - V(\xi) + R(\xi) \right]^2 \right\}$$

$$\sigma^2(\xi) = \sum_{\xi_1} p(\xi_1|\xi) \left\{ \gamma^{2\tau(\xi)} \sum_{s_\infty(\xi_1)} p(s_\infty(\xi_1)) \left[\sum_{i=0}^{\infty} \gamma^{\tau_i(s_\infty(\xi_1))} R(\xi_i) \right]^2 + \left[\gamma^{\tau(\xi)} V(\xi_1) - V(\xi) + R(\xi) \right]^2 \right\}$$

$$\text{Thus :} \qquad \sigma^2(\xi) = e(\xi) + \gamma^{2\tau(\xi)} \sum_{\xi_i} p(\xi_i|\xi)\sigma^2(\xi_i)$$

with $e(\xi)$ satisfying (10). Thus the variance $\sigma^2(\xi)$ is the sum of the immediate contribution $e(\xi)$ that takes into account the variation in the values of the immediate successors ξ_i, and the discounted average of the variance $\sigma^2(\xi_i)$ of these successors.

This is a Bellman equation and it can be solved by value iteration.

Remark. We can give a geometric 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 : either if the gradient at the iterated point η is very low (i.e. the values are almost constant) or if η is very close to one vertex ξ_i (then the barycentric coordinate λ_{ξ_i} is close to 1 and the λ_{ξ_j} (for $j \neq i$) are close to 0, thus $Q(\eta)$ is low). In both of these cases, $e(\xi)$ is low and implies that the iteration of ξ does not lead to a degradation of the quality of approximation of the value function (the variance does not increase).

9.3. Example : variance of the "Car on the Hill"

Figure 16 shows the standard deviation $\sigma(\xi)$ for the Car-on-the-Hill problem for a uniform grid (of 257 by 257).

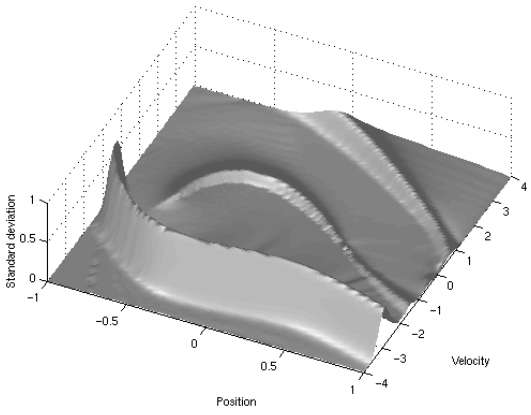


Figure 16. The standard deviation σ for the "Car on the Hill". The standard deviation is very high around the frontier 1 : indeed, a discontinuity is impossible to approximate perfectly by discretization techniques, whatever the resolution is. We can observe this fact on figure 10 where the maximal error of approximation is equal to half the step of the discontinuity. However, the higher the resolution is, the lower the integral of the error is (compare figure 10(a)&(c) versus (b)&(d)). There is a noticeable positive standard deviation around frontier 3 and the upper part of frontier 2 because the value function is an average of different values of the discounted terminal reinforcement.

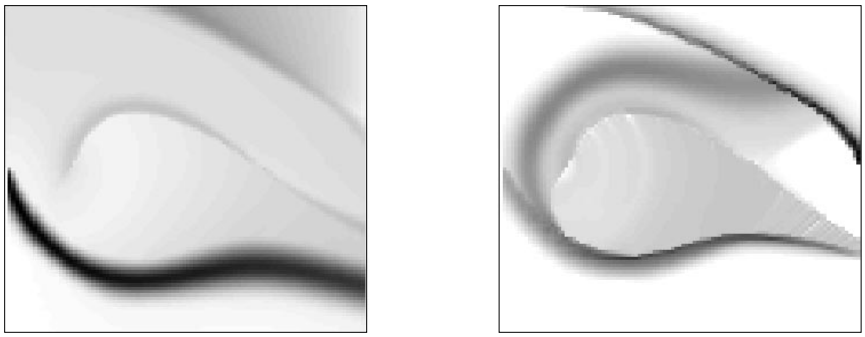
A refinement of the resolution in the areas where the standard deviation is low has no chance of producing an important change in the value function. Thus it appears that the areas where a splitting might affect the most the approximation of the value function are the rectangles of highest surface whose corners have the highest standard deviations.

10. A global splitting criterion

Now we are going to combine the notions of *influence* and *variance* in order to define a new, 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, thus the states that could improve the most their approximation accuracy when split (figure 17(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 change in the optimal control.

Thus in order to improve the precision of approximation at the most relevant areas of the state space we split the states ξ of highest standard deviation that have an influence on the areas of change in the optimal control, according to the *Stdev_Inf* criterion (see figure 17) : $Stdev\_Inf(\xi) = \sigma(\xi).I(\xi|\Sigma)$. Figure 18 shows the discretization obtained by using this splitting criterion.



(a) Standard deviation (b) Influence x Standard deviation

Figure 17. (a) The standard deviation $\sigma(\xi)$ for the "Car on the Hill" (equivalent to figure 16) and (b) The *Stdev_Inf* criterion, product of $\sigma(\xi)$ by the influence $I(\xi|\Sigma)$ (figure 15(b)).

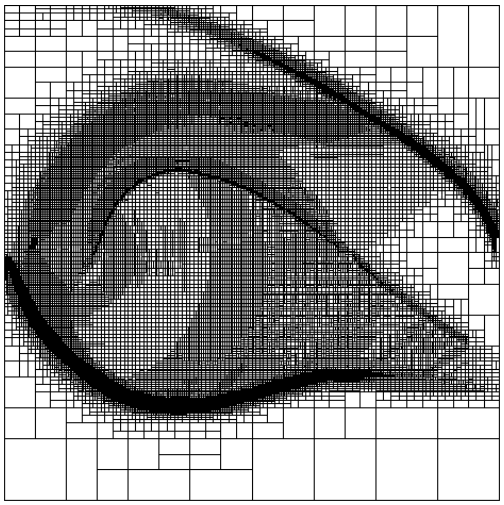


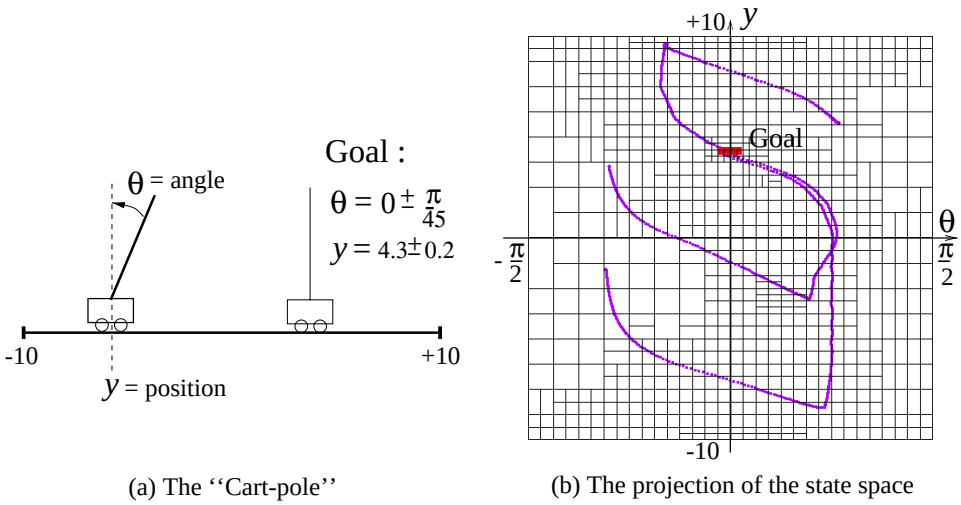
Figure 18. The discretization resulting of the *Stdev_Inf* split criterion. We observe that the upper part of frontier 1 is well refined. This refinement occurred not because we split according to the value function (such as the *corner-value difference* or *value non-linearity* criterion) but because the splitting there is necessary to have a good approximation of the value function around the bottom part of frontier 2 (and even the upper part of frontier 2) where there is a change in the optimal control.
The fact that the *Stdev_Inf* criterion does not split the areas where the VF is discontinuous unless some refinement is necessary to get a better approximation of the optimal control, is very important since, as we will see in the simulations that follow, in higher dimensions, the cost to get an accurate approximation of the VF is too high.

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 more difficult problems (the Acrobot, the Cart-pole) as illustrated in what follows.

11. Illustration on other control problems

11.1. The Cart-Pole problem

The dynamics of this 4-dimensional physical system (illustrated in figure 19(a)) are described in (Barto et al., 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 19). Notice that our task of "minimum time manoever 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 terminal reinforcement R 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.



(a) The ''Cart-pole'' (b) The projection of the state space

Figure 19. (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.

Figure 20 shows the performance obtained for several splitting criteria previously defined for this 4 dimensional control problem. We observe the following points :

- The local 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 rebalance 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.
- The *Stdev_Inf* criterion performs very well. We observe that the trajectories (see figure 19(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).

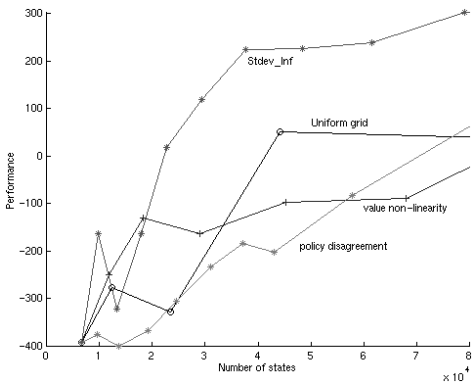


Figure 20. Performance on the "Cart-pole".

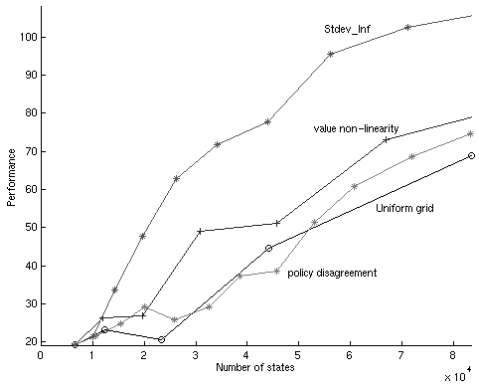


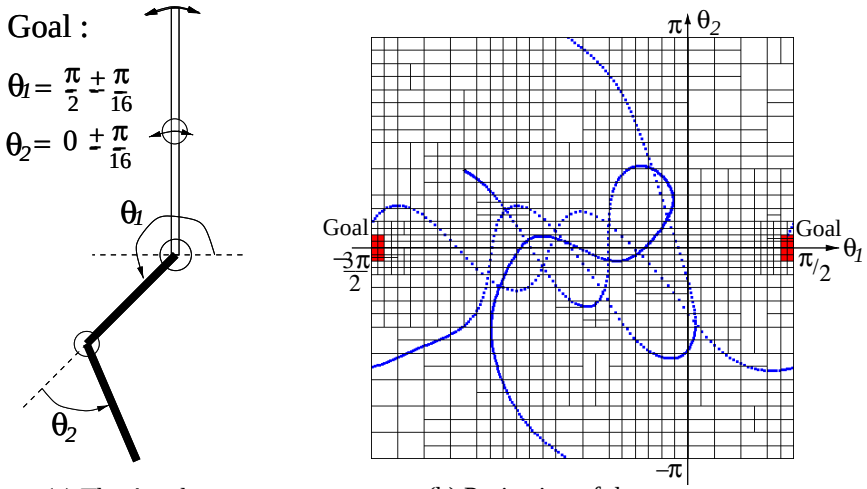
Figure 21. Performance on the Acrobot.

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 22(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 22(b)), for which the system receives a reinforcement of $R = +1$. 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 21 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 22(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 22. (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.

Interpretation of the results : As we noticed for the two previous 4d problems, the local splitting criteria fail to improve the performance of the uniform grids because they spend too many resources on local considerations (either approximating the value function or the optimal policy). For example, on the "Cart-pole" problem, the *value non-linearity* criterion will concentrate on approximating the VF mostly at parts of the state space where there is already no chance to rebalance 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 criteria start to perform better than the uniform grids, because these areas get eventually refined).

The *Stdev_Inf* criterion, which takes into account global consideration for the splitting, performs very well for all the problems 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 our 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 the notions of *influence* which estimates the impact of states over others, and of *variance* of a Markov chain, which measure the quality of the current approximation. By combining these notions, we defined an interesting splitting criterion that gives very good performance (in comparison to the uniform grids) on all the problems studied.

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.

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

- A generalization process, in order to have also a "specific towards general" grouping of areas (for example by pruning the tree) that have been over-refined.
- Suppose that we only want to solve the problem for a specific area Ω of initial start states. Then we can restrict our refinement process to the areas used by the trajectories. The notion of influence introduced in this paper can be used for that purpose by computing the *Stdev_Inf* criterion with respect to $\Sigma_{|\Omega} = \{\xi \in \Sigma, I(\xi|\Omega) > 0\}$ (the set of states of *policy disagreement* that have an influence on the area of initial states Ω) instead of Σ, and can improve drastically the performance observed when starting from this specific area.
- We would like to deal with the stochastic case. If we assume that we have a model of the noise, then the only change will be in the process of building the MDP (Kushner & Dupuis, 1992; Munos & Bourgine, 1997).
- Release the assumption that we have a model and build an approximation of the dynamics and the reinforcement, and deal with the exploration problem.

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.*

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 Trans. on Mathematical Software, 3*(3), 209–226.

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

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. *Journal of Machine Learning.*

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. (1993). Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning, 13.*

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. (1999). A study of reinforcement learning in the continuous case by the means of viscosity solutions. *To appear in Machine Learning Journal.*

Munos, R., & Bourgine, P. (1997). Reinforcement learning for continuous stochastic control problems. *Neural Information Processing Systems.*

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

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

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.*