Tropical numerical methods for solving stochastic control problems

Marianne Akian, Jean-Philippe Chancelier, Luz Pascal, Benoît Tran

Tropical numerical methods for solving stochastic control problems

Marianne Akian * **Jean-Philippe Chancelier** ** **Luz Pascal** ****
Benoît Tran ***

* *Inria and CMAP, École polytechnique CNRS IP Paris, France (e-mail: marianne.akian@inria.fr)*
** *CERMICS, École des Ponts ParisTech, France (jean-philippe.chancelier@enpc.fr)*
*** *FGV EMAp, Brazil (benoit.tran@tutanota.com)*
**** *Queensland University of Technology & CSIRO, Australia (luz.pascal96@gmail.com)*

Abstract: We consider Dynamic programming equations associated to discrete time stochastic control problems with continuous state space, which arise in particular from monotone time discretizations of Hamilton-Jacobi-Bellman equations. We develop and study several numerical algorithms for solving such equations, combining tropical numerical methods and stochastic dual dynamic programming methods. We also compare these algorithms with the point based methods for solving Partially Observable Markov Decision Processes (POMDP).

Keywords: Stochastic Control, Hamilton-Jacobi-Bellman equations, Stochastic Dual Dynamic Programming, Tropical algebra, Partially Observable Markov decision processes.

1. INTRODUCTION

We consider the following stochastic control problem with discrete time and a possibly discounted additive payoff, either with a finite or infinite horizon T. At each step $t \in [\![0, T]\!]$, the state $\mathbf{X_t} \in \mathbb{X} \subset \mathbb{R}^n$ follows the following dynamics

$$\mathbf{X_{t+1}} = f_t^{\mathbf{W_{t+1}}}(\mathbf{X_t}, \mathbf{U_t}) \ ,$$

where $(\mathbf{W_t})_{t\in[\![0,T]\!]}$ is a sequence of random variables with values in some measurable set $(\mathbb{W}, \mathcal{W})$, and $(\mathbf{U_t})_{t\in[\![0,T]\!]}$ is an *adapted* sequence of (random) decisions or controls with values in some measurable set $(\mathbb{U}, \mathcal{U})$. The state is fully observed, and we may be in the hazard-decision framework in which *adapted* means that, for all t, $\sigma(\mathbf{U_t}) \subset \sigma(\mathbf{X_0}, \mathbf{W_1}, \ldots, \mathbf{W_{t+1}})$. We may also be in the decision-hazard framework, in which *adapted* means that, for all t, $\sigma(\mathbf{U_t}) \subset \sigma(\mathbf{X_0}, \mathbf{W_1}, \ldots, \mathbf{W_t})$. At each time t, the decision maker is receiving the reward

$$r_t^{\mathbf{W_{t+1}}}(\mathbf{X_t}, \mathbf{U_t}) \ ,$$

and at the final time, if any, the decision maker receives the final reward $\psi(\mathbf{X_T})$. Then, the decision maker aims to maximize his total expected reward:

$$\mathbb{E}\left[\sum_{t=0}^{T-1} r_t^{\mathbf{W_{t+1}}}(\mathbf{X_t}, \mathbf{U_t}) + \psi(\mathbf{X_T})\right] \ .$$

Such a problem is also called a multi-stage optimization problem. In the sequel, we shall assume that the random variables $\mathbf{W_{t+1}}$ are independent and with finite support $\mathbb{W}$. The law of $\mathbf{W_{t+1}}$ may depend on t. In the decision-hazard framework, we may also consider the case where the law of $\mathbf{W_{t+1}}$ depends on $(\mathbf{X_t}, \mathbf{U_t})$, which is equivalent to consider the general framework of Markov decision processes, with some given transition probabilities $T_t^{u_t}(x_t, x_{t+1}) = P(\mathbf{X_{t+1}} = x_{t+1} \mid \mathbf{X_t} = x_t, \mathbf{U_t} = u_t)$, such that $T_t^{u_t}(x_t, \cdot)$ has a finite support. In the hazard-decision framework, we can assume that the law of $\mathbf{W_{t+1}}$ depends on $\mathbf{X_t}$, but it cannot depend on $\mathbf{U_t}$. The discrete probability law of $\mathbf{W_{t+1}}$ will be denoted by $p_t^{x_t,u_t}(w) = P(\mathbf{W_{t+1}} = w \mid \mathbf{X_t} = x_t, \mathbf{U_t} = u_t)$ in the first case, and by $p_t^{x_t}(w) = P(\mathbf{W_{t+1}} = w \mid \mathbf{X_t} = x_t)$ in the second case.

By the dynamic programming approach (see Bellman (1984)), the value function of the above problem is the function V_0 obtained from the solution to the following recurrence equation:

$$V_T = \psi \quad \text{and} \quad \forall t \in [\![0, T-1]\!], V_t = \mathfrak{B}_t\left(V_{t+1}\right) \ , \qquad (1)$$

where $\mathfrak{B}_t$ is the associated Bellman operator from the set of extended real functions over $\mathbb{X}$ ($\overline{\mathbb{R}}^{\mathbb{X}}$), to itself. This operator can be written as the composition of three different operators among the following ones which are operating on functions from either $\mathbb{X}$, $\mathbb{X} \times \mathbb{U}$, $\mathbb{X} \times \mathbb{U} \times \mathbb{W}$ or $\mathbb{X} \times \mathbb{W}$ to $\overline{\mathbb{R}}$:

⋆ Sponsor and financial support acknowledgment goes here. Paper titles should be written in uppercase and lowercase letters, not all uppercase.

$$\mathcal{Q}_t(\phi)(x,u,w) = r_t^w(x,u) + \phi\big(f_t^w(x,u)\big),$$
$$\mathcal{M}_t^{(1)}(Q)(x,w) = \max_{u\in\mathbb{U}} Q(x,u,w),$$
$$\mathcal{E}_t^{(2)}(Q)(x) = \mathbb{E}\Big[Q(x,\mathbf{W_{t+1}})\Big] = \sum_{w\in\mathbb{W}} p_t^x(w)Q(x,w),$$
$$\mathcal{E}_t^{(1)}(Q)(x,u) = \mathbb{E}\Big[Q(x,u,\mathbf{W_{t+1}}) \mid \mathbf{X}_t = x, \mathbf{U}_t = u\Big] = \sum_{w\in\mathbb{W}} p_t^{x,u}(w)Q(x,u,w),$$
$$\mathcal{M}_t^{(2)}(Q)(x) = \max_{u\in\mathbb{U}} Q(x,u) \ ,$$

in which we use the convention $+\infty - \infty = -\infty$. Indeed, in the hazard-decision case, we have $\mathfrak{B}_t = \mathcal{E}_t^{(2)} \circ \mathcal{M}_t^{(1)} \circ \mathcal{Q}_t$ and in the decision-hazard case, we have $\mathfrak{B}_t = \mathcal{M}_t^{(2)} \circ \mathcal{E}_t^{(1)} \circ \mathcal{Q}_t$.

The above discrete time Bellman equation (1) can also be obtained after some semi-Lagrangian time discretization of a Hamilton-Jacobi-Bellman equation, see Falcone and Ferretti (2014), or any monotone time discretization. The dynamic programming approach suffers from the "curse of dimensionality", since one would need to compute for all $t \in [\![0,T]\!]$ the value function V_t on all the state space $\mathbb{X}$, and any grid-based discretization would need a number of values exponential in the dimension n of the state space $\mathbb{X}$. Several methods have been proposed in the litterature to bypass the obstruction of curse of dimensionality. We shall only cite the ones related to the present work: the tropical numerical methods developped in the context of Hamilton-Jacobi equations (McEneaney (2007); McEneaney et al. (2011); Qu (2014); Akian and Fodjo (2018)), the tree-structured algorithm developped recently by Alla et al. (2019), and the stochastic dual dynamic programming method developped in the context of discrete time stochastic control (Pereira and Pinto (1991); Philpott et al. (2013)).

Here, we consider a general algorithm inspired by both tropical numerical methods and SDDP algorithm and which can be seen as a generalization of the algorithms proposed in Philpott et al. (2013); Baucke et al. (2018)). Moreover, we show that in the case of the dynamic programming equation associated to a partially observable Markov Decision Process (POMDP), it is similar to the so called point based algorithms developped in Pineau et al. (2003); Kurniawati et al. (2008); Shani et al. (2013).

2. TROPICAL NUMERICAL METHOD FOR LIPSCHITZ PRESERVING BELLMAN OPERATORS

The following algorithm is introduced in more details in Akian et al. (2020).

We assume that the map r_t^w takes its values in $\mathbb{R}\cup\{-\infty\}$, to handle constraints in state and control. We also assume that r_t^w is bounded from above, which will imply that the Bellman operator preserves the set of upper bounded function from $\mathbb{X}$ to $\mathbb{R}\cup\{-\infty\}$. For any function ϕ from a set Y to $\mathbb{R}\cup\{-\infty\}$, the support of ϕ will be the set of $y \in Y$ such that $\phi(y) \in \mathbb{R}$. The constraints determined by the support of r_t^w and the support of the final reward ψ induce a sequence of sets $X_t \subset \mathbb{X}$, $t \in [\![0,T]\!]$, such that any sequence V_t, $t \in [\![0,T]\!]$, satisfying (1) is such that the support of V_t is included in X_t.

It is well known that Bellman operators are order preserving and nonexpansive for the sup-norm. Since $\mathbb{X}$ is an infinite set, we shall need the following stronger property:

Assumption 1. There exists a sequence $\mathcal{L}_t$, $t \in [\![0,T]\!]$, of compact subsets of the set of functions from $\mathbb{X}$ to $\mathbb{R}\cup\{-\infty\}$, endowed with the uniform convergence topology, such that, for all $t \in [\![0,T-1]\!]$, the Bellman operator $\mathfrak{B}_t$ sends $\mathcal{L}_{t+1}$ into $\mathcal{L}_t$.

Compact subsets $\mathcal{L}_t$ can be obtained by taking the set of functions from $\mathbb{X}$ to $\mathbb{R}\cup\{-\infty\}$, that are L_t-Lipschitz continuous (for some given norm of $\mathbb{R}^n$) on their compact support X_t, or any closed subset of this set. In Akian et al. (2020), Assumption 1 is proved to be satisfied for these particular sets $\mathcal{L}_t$, for some constants L_t, under some technical conditions similar to the ones that are generally assumed in the proofs of convergence of SDDP algorithm. Another important property assumed to apply SDDP algorithm (in the above context of a maximization problem) is that the control set is polyhedral, that the reward functions are polyhedral and concave with respect to state and control, and that the dynamics are affine with respect to state and control. In that case the value function can be approximated by the finite infimum of affine functions, and the computation of appropriate affine functions can be done by solving some Linear Programs. In what follows, we describe a general algorithm which does not necessarily need this property. What will be needed however is the following assumption which is satisfied again by the above particular sets.

Assumption 2. The subsets $\mathcal{L}_t$ of Assumption 1 are lattices for the pointwise partial order: for all $t \in [\![0,T]\!]$, and $\phi, \phi' \in \mathcal{L}_t$, there exists a supremum (least upper bound) of ϕ and ϕ' in $\mathcal{L}_t$, that we shall denote by $\phi \vee \phi'$ and an infimum (greatest lower bound) of ϕ and ϕ' in $\mathcal{L}_t$, that we shall denote by $\phi \wedge \phi'$.

Note that the set $\mathcal{C}_t$ of concave functions that are L_t-Lipschitz continuous on their compact support X_t is stable by the infimum operation, so that the pointwise infimum in the set of all functions coincides with the infimum in $\mathcal{C}_t$. It is not stable by the pointwise supremum, but the supremum in $\mathcal{C}_t$ exists and coincides with the concave hull of (the least concave map greater than) the pointwise supremum.

The Tropical Dynamic Programming (TDP) algorithm of Akian et al. (2020) (which generalizes Philpott et al. (2013); Baucke et al. (2018)) consists in the iterative construction of two approximations of the value function V_t, one from above and one from below. At each iteration k, the upper approximation, denoted $\overline{V}_t^k$, is obtained as the infimum (in $\mathcal{L}_t$) of a finite set $\overline{F}_t^k$ of basic functions and the lower approximation, denoted $\underline{V}_t^k$, is obtained as the supremum (in $\mathcal{L}_t$) of a finite set $\underline{F}_t^k$ of basic functions. Basic functions for the upper and lower approximations are taken respectively in subsets $\underline{\mathbf{F}}_t$ and $\overline{\mathbf{F}}_t$ of $\mathcal{L}_t$, that is we have $\overline{F}_t^k \subset \overline{\mathbf{F}}_t$ and $\underline{F}_t^k \subset \underline{\mathbf{F}}_t$. Note that the approximations $\overline{V}_t^k$ and $\underline{V}_t^k$ are parametrized by the sets $\overline{F}_t^k$ and $\underline{F}_t^k$, which means that we never store the values of these functions on a grid of $\mathbb{X}$. The sets of basic functions $\overline{F}_t^k$ and $\underline{F}_t^k$ are increasing with respect to iteration number

k, so that $\overline{V}_t^{k+1} \leq \overline{V}_t^k$ and $\underline{V}_t^{k+1} \geq \underline{V}_t^k$. These sets are computed using a sequence of state-action-noise, which is itself computed using the previous sequence of functions.

Starting with an initial state x_0 and emptysets, or appropriate singleton sets $\underline{\phi}_{t+1}^0$ and $\overline{\phi}_{t+1}^0$, the algorithm solving the Bellman equation in the hazard-decision framework consists at each step $k \geq 0$ in the following two phases:

- **Forward phase:** Compute a new (deterministic) trajectory $(x_t^k)_{t\in[\![0,T]\!]}$ starting in x_0 as follows. For $t = 0, \ldots, T-1$, do:

  For each $w \in \mathbb{W}$, compute an optimal control u_t^w for $\overline{V}_{t+1}^k$ at x_t^k:
  $$u_t^w \in \arg\max_{u\in\mathbb{U}} \mathcal{Q}_t(\overline{V}_{t+1}^k)(x_t^k, u, w) \ . \tag{2}$$
  Compute the noise $w_t \in \mathbb{W}$ which maximizes the future gap
  $$w_t \in \arg\max_{w\in\mathbb{W}} \big(\overline{V}_{t+1}^k - \underline{V}_{t+1}^k\big)\big(f_t^w(x_t, u_t^w)\big).$$
  Compute the next state associated to the above noise and optimal control:
  $$x_{t+1}^k = f_t^{w_t}(x_t^k, u_t^{w_t}) \ .$$
- **Backward phase:** For $t = T, T-1, \ldots, 0$, select for both upper and lower approximations, one new basic function $\overline{\phi}_t \in \overline{\mathbf{F}}_t$ (resp. $\underline{\phi}_t \in \underline{\mathbf{F}}_t$) and add it to the corresponding set: $\overline{F}_t^{k+1} := \overline{F}_t^k \cup \{\overline{\phi}_t\}$ and $\underline{F}_t^{k+1} := \underline{F}_t^k \cup \{\underline{\phi}_t\}$.

  If $t = T$, the new basic functions are chosen such that
  $$\overline{\phi}_T \geq \psi \quad \text{and} \quad \overline{\phi}_T(x_T^k) = \psi(x_T^k).$$
  and symmetrically
  $$\underline{\phi}_T \leq \psi \quad \text{and} \quad \underline{\phi}_T(x_T^k) = \psi(x_T^k).$$
  If $t < T$, the new basic functions are chosen such that
  $$\overline{\phi}_t \geq \mathfrak{B}_t\left(\overline{V}_{t+1}^{k+1}\right)$$
  $$\overline{\phi}_t(x_t^k) = \mathfrak{B}_t\left(\overline{V}_{t+1}^{k+1}\right)(x_t^k).$$
  and symmetrically
  $$\underline{\phi}_t \leq \mathfrak{B}_t\left(\underline{V}_{t+1}^{k+1}\right)$$
  $$\underline{\phi}_t(x_t^k) = \mathfrak{B}_t\left(\underline{V}_{t+1}^{k+1}\right)(x_t^k),$$
  where for all t, k, we denote $\overline{V}_t^k = \inf \overline{F}_t^k$ and $\underline{V}_t^k = \sup \underline{F}_t^k$.

For the decision-hazard framework, the only difference is in the forward phase, in which one computes an optimal control u_t independent of w:
$$u_t \in \arg\max_{u\in\mathbb{U}} \mathcal{E}_t^{(1)}(\mathcal{Q}_t(\overline{V}_{t+1}^k))(x_t^k, u) \ .$$
If $\mathcal{L}_t$ is the set of L_t-Lipschitz continuous functions on their compact support X_t, for some given norm $\|\cdot\|$ of $\mathbb{R}^n$, then a typical example of a set of basic functions $\underline{\mathbf{F}}_t$ is the set of functions $x \in X_t \mapsto a - L_t\|x - x_0\|$ with $a \in \mathbb{R}$ and $x_0 \in X_t$. Then $-\underline{\mathbf{F}}_t$ is also a good candidate for $\overline{\mathbf{F}}_t$. If $\mathcal{L}_t$ is the set of concave L_t-Lipschitz continuous functions on their compact support X_t, then one can replace $\overline{\mathbf{F}}_t$ by the set of L_t-Lipschitz continuous affine maps restricted to X_t. This is what is done in the SDDP like algorithms of Philpott et al. (2013); Baucke et al. (2018).

Theorem 3. (Akian et al. (2020)). Let V_t be the solution of the Bellman equation (1). For all $t \in [\![0, T]\!]$, the sequences $\left(\underline{V}_t^k\right)_{k\in\mathbb{N}}$ and $\left(\overline{V}_t^k\right)_{k\in\mathbb{N}}$ converge uniformly to two functions $\underline{V}_t^*$ and $\overline{V}_t^*$ of $\mathcal{L}_t$ which satisfy $\underline{V}_t^* \leq V_t \leq \overline{V}_t^*$. Moreover, we have that $\overline{V}_t^*(x_t^*) = V_t(x_t^*) = \underline{V}_t^*(x_t^*)$ for every accumulation point x_t^* of the sequence $(x_t^k)_{k\in\mathbb{N}}$. In particular $\overline{V}_0^*(x_0) = V_0(x_0) = \underline{V}_0^*(x_0)$.

The above algorithm and theorem are defined and stated in Akian et al. (2020) under some technical assumptions which ensure that Assumptions 1 and 2 hold for the set $\mathcal{L}_t$ of L_t-Lipschitz continuous functions with compact support X_t. However, the algorithm and proof only use the properties stated in these assumptions.

3. POINT BASED ALGORITHMS FOR POMDP

One way to solve a partially observable Markov decision Problem (POMDP) is to introduce, for each time t, the belief state b_t that is a probability distribution among the elements of the state space, given the information available at time t. The value function and an optimal strategy can then be obtained by solving the dynamic programming equation of a Markov decision process over the belief state space with perfect information. This gives in particular an optimal strategy which only depend on the belief state at the current time.

We recall this dynamic programming equation in the discounted infinite horizon case, for which point based algorithm were introduced (see Pineau et al. (2003); Kurniawati et al. (2008); Shani et al. (2013)). Assume that the state space is equal to $[n] := \{1, \ldots, n\}$, so that the belief space is the simplex $\Delta_n = \{b \in \mathbb{R}_+^n \mid \sum_i b_i = 1\}$, which is a compact subset of $\mathbb{X} = \mathbb{R}^n$. Assume also that the observation space $\mathbb{O}$ and the control space $\mathbb{U}$ are finite sets. For all $o \in \mathbb{O}$ and $u \in \mathbb{U}$, let us denote by $\mathcal{M}^{u,o}$ the $n \times n$ matrix with entries
$$\mathcal{M}_{xx'}^{u,o} = P(\mathbf{o_{t+1}} = o, \mathbf{X_{t+1}} = x' \mid \mathbf{X_t} = x, \mathbf{U_t} = u) \ ,$$
and let us any belief state as a row $1 \times n$ vector. Then, the dynamics of the belief state is given by:
$$\mathbf{b_{t+1}} = \tau^{\mathbf{o_{t+1}}}(\mathbf{b_t}, \mathbf{u_t}) \quad \text{with} \quad \tau^o(b, u) = \frac{b\mathcal{M}^{u,o}}{b\mathcal{M}^{u,o}\mathbf{1}} \ .$$
We also have
$$P(\mathbf{o_{t+1}} = o \mid \mathbf{b_t} = b, \mathbf{U_t} = u) = p^{b,u}(o) := b\mathcal{M}^{u,o}\mathbf{1} \ .$$
Denoting $\gamma < 1$ the discount factor, the dynamic programming equation of the POMDP is the fixed point equation
$$V = \mathfrak{B}(V)$$
with $\mathfrak{B} = \mathcal{M} \circ \mathcal{E} \circ \mathcal{Q}$ for the following operators
$$\mathcal{Q}(\phi)(b, u, o) = R(b, u) + \gamma\phi\big(\tau^o(b, u)\big),$$
$$\mathcal{E}(Q)(b, u) = \sum_{o\in\mathbb{O}} p^{b,u}(o) Q(b, u, o),$$
$$\mathcal{M}(Q)(b) = \max_{u\in\mathbb{U}} Q(b, u) \ ,$$
in which $R(b, u) = b\, r(\cdot, u) = \sum_{x\in[n]} b_x r(x, u)$. The above Bellman operator has same form as the one of previous section in the decision-hazard framework, but for some special dynamics. The observation process $\mathbf{o_t}$ play the role

of the noise process $\mathbf{W_t}$. Both belong to finite sets. So we can apply the TDP algorithm as soon as we found some appropriate sets $\mathcal{L}_t$. It is well known that the value function of a POMDP is a bounded convex Lipschitz continuous function over the simplex Δ_n. The bound and the Lipschitz constant (with respect to the ℓ_1 norm on the simplex) are both equal to $L = R_{\max}/(1-\gamma)$, where $R_{\max}$ is the sup-norm of the reward function r. The point based algorithms developped in (Pineau et al. (2003); Kurniawati et al. (2008); Shani et al. (2013)) consist in approximating the value function from below by a supremum of linear maps and from above by either an infimum of functions of the form $b \mapsto a + L\|b - b_0\|_1$ or by the convex hull of such functions. Both methods can be seen as a particular case of the TDP algorithm, up to some improvements, and a generalization to the infinite horizon case. Such a generalization consists in gathering (at each iteration k) all the improvements into the same approximate value function $\overline{V}^k$ or $\underline{V}^k$, and in stopping the trajectory (x_t^k) at a time T such that $\left(\overline{V}^k - \underline{V}^k\right)(x_T^k) \leq \epsilon$.

In Smith and Simmons (2005), an analysis of the point based algorithm is done, only under the assumption that the algorithm stops. Moreover, in Fehr et al. (2018), it is proved that the Bellman operator $\mathfrak{B}$ preserves the set of Lipschitz continuous functions over the simplex, but the Lipschitz constant can become very large for some $\gamma > 1/2$.

We can show however that the set $\mathcal{L}$ of functions on the simplex which are bounded by L and can be extended in a positively homogenous and L-Lipschitz continuous map on the positive cone $\mathbb{R}^n_+$ is preserved by the Bellman operator of the POMDP. This set is compact for the uniform convergence topology, so it satisfies Assumption 1. It also satisfies Assumption 2. This allows us to construct a variant of point based algorithm for which a convergence result similar to Theorem 3 up to ϵ is possible.

REFERENCES

Akian, M., Chancelier, J.P., and Tran, B. (2020). Tropical dynamic programming for lipschitz multistage stochastic programming. ArXiv:2010.10619.

Akian, M. and Fodjo, E. (2018). From a monotone probabilistic scheme to a probabilistic max-plus algorithm for solving Hamilton-Jacobi-Bellman equations. In *Hamilton-Jacobi-Bellman equations*, volume 21 of *Radon Ser. Comput. Appl. Math.*, 1–23. De Gruyter, Berlin.

Alla, A., Falcone, M., and Saluzzi, L. (2019). An efficient DP algorithm on a tree-structure for finite horizon optimal control problems. *SIAM J. Sci. Comput.*, 41(4), A2384–A2406. doi:10.1137/18M1203900.

Baucke, R., Downward, A., and Zakeri, G. (2018). A deterministic algorithm for solving stochastic minimax dynamic programmes. *Preprint, available on Optimization Online*, 36.

Bellman, R. (1984). *Dynamic Programming*. Princeton Univ. Pr, Princeton, NJ.

Falcone, M. and Ferretti, R. (2014). *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Fehr, M., Buffet, O., Thomas, V., and Dibangoye, J. (2018). rho-pomdps have lipschitz-continuous epsilon-optimal value functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Kurniawati, H., Hsu, D., and Lee, W.S. (2008). Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Zurich, Switzerland.

McEneaney, W.M. (2007). A curse-of-dimensionality-free numerical method for solution of certain HJB PDEs. *SIAM J. Control Optim.*, 46(4), 1239–1276. doi: 10.1137/040610830.

McEneaney, W.M., Kaise, H., and Han, S.H. (2011). Idempotent method for continuous-time stochastic control and complexity attenuation. In *Proceedings of the 18th IFAC World Congress, 2011*, 3216–3221. Milano, Italie.

Pereira, M.V.F. and Pinto, L.M.V.G. (1991). Multi-stage stochastic optimization applied to energy planning. *Math. Programming*, 52(2, Ser. B), 359–375. doi: 10.1007/BF01582895.

Philpott, A., de Matos, V., and Finardi, E. (2013). On Solving Multistage Stochastic Programs with Coherent Risk Measures. *Operations Research*, 61(4), 957–970. doi:10.1287/opre.2013.1175.

Pineau, J., Gordon, G., Thrun, S., et al. (2003). Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, 1025–1032.

Qu, Z. (2014). A max-plus based randomized algorithm for solving a class of HJB PDEs. In *53rd IEEE Conference on Decision and Control*, 1575–1580. doi: 10.1109/CDC.2014.7039624.

Shani, G., Pineau, J., and Kaplow, R. (2013). A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1), 1–51.

Smith, T. and Simmons, R. (2005). Point-based pomdp algorithms: Improved analysis and implementation. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI-05)*.