

# Probabilistic Planning with Nonlinear Utility Functions\*

Yaxin Liu

Department of Computer Sciences  
The University of Texas at Austin  
Austin, TX 78712-0233  
yxliu@cs.utexas.edu

Sven Koenig

Computer Science Department  
University of Southern California  
Los Angeles, CA 90089-0781  
skoenig@usc.edu

## Abstract

Researchers often express probabilistic planning problems as Markov decision process models and then maximize the expected total reward. However, it is often rational to maximize the expected utility of the total reward for a given nonlinear utility function, for example, to model attitudes towards risk in high-stake decision situations. In this paper, we give an overview of basic techniques for probabilistic planning with nonlinear utility functions, including functional value iteration and a backward induction method for one-switch utility functions.

## Introduction

In this paper, we give an overview of current research that addresses the issue of probabilistic planning with completely observable Markov decision process models (MDPs) where one wants to maximize the expected utility of the total reward (= the expected total utility) for a given monotonically nondecreasing utility function. Utility theory (von Neumann & Morgenstern, 1944) states that every rational human decision maker who accepts a small number of axioms has a monotonically nondecreasing utility function  $U$  that transforms their real-valued wealth levels  $w$  into finite real-valued utilities  $U(w)$  so that they always choose the course of action that maximizes their expected utility. Linear utility functions result in maximizing the expected total reward and characterize risk-neutral human decision makers, while nonlinear utility functions characterize risk-sensitive human decision makers. In particular, concave utility functions characterize risk-averse human decision makers (“insurance holders”), and convex utility functions characterize risk-seeking human decision makers (“lottery players”). Probabilistic planning with nonlinear utility functions is important since human decision makers are often risk-sensitive in single-instance decision situations with the possibility of large losses of money, equipment or human life (= high-stake decision situations) and their risk attitude affects their decisions. Probabilistic planning with nonlinear utility functions is therefore important for space applications (Zilberstein *et al.*, 2002), environmental applications (Blythe, 1998) and business applications (Goodwin, Akkiraju, & Wu, 2002), which are currently solved without taking risk attitudes into account.

\*We thank Craig Tovey and Anton Kleywegt, two operations researchers, for lots of advice. This research was partly supported by NSF awards to Sven Koenig under contracts IIS-9984827 and IIS-0098807 and an IBM fellowship to Yaxin Liu. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, companies or the U.S. government.  
Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

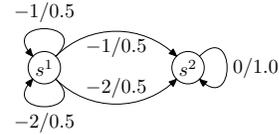


Figure 1: Importance of Finite Optimal Values

## Probabilistic Planning

We study probabilistic planning for MDPs that consist of a finite set of states  $S$ , a finite set of goal states  $G \subseteq S$ , and a finite set of actions  $A$  for each nongoal state  $s \in S \setminus G$ . The agent is given a time horizon  $1 \leq T \leq \infty$ . The initial time step is  $t = 0$ . Assume that the agent is in state  $s_t \in S$  at time step  $t$ . If  $t = T$  or  $s_t$  is a goal state, then the agent stops to execute actions, which means that it no longer receives rewards in the future. Otherwise, it executes an action  $a_t \in A$  of its choice. The execution of the action results in finite reward  $r(s_t, a_t, s_{t+1}) < 0$  (pure cost) and a transition to state  $s_{t+1} \in S$  in the next time step with probability  $P(s_{t+1}|s_t, a_t)$ . The wealth level of the agent at time step  $t$  is  $w_t = \sum_{\tau=0}^{t-1} r(s_\tau, a_\tau, s_{\tau+1})$ .

We use  $\Pi$  to denote the set of all possible courses of action (= plans; more precisely: randomized, history-dependent policies). If the agent starts in state  $s_0 \in S$  and follows plan  $\pi \in \Pi$  for a finite time horizon  $T$ , then the expected utility of its final wealth level (= expected total utility) is  $v_{U,T}^\pi(s_0) = E^{s_0,\pi}[U(w_T)]$ , where the expectation is taken over all sequences of states and actions from time step 0 to time step  $T$  that can result with positive probability from executing plan  $\pi$  in start state  $s_0$  (= trajectories). If the agent starts in state  $s_0 \in S$  and follows plan  $\pi \in \Pi$  for an infinite time horizon  $T$ , then its expected total utility is  $v_U^\pi(s_0) = \lim_{T \rightarrow \infty} v_{U,T}^\pi(s_0)$ .<sup>1</sup> In this paper, we assume an infinite time horizon  $T$ . If the agent starts in state  $s_0 \in S$  and follows a plan that maximizes its expected total utility, then its expected total utility (= optimal value) is  $v_U^*(s_0) = \sup_{\pi \in \Pi} v_U^\pi(s_0)$ . The objective of the agent is to find a plan that maximizes its expected total utility for every start state (= optimal plan), that is, a plan  $\pi_U^* \in \Pi$  with  $v_U^{\pi_U^*}(s_0) = v_U^*(s_0)$  for all states  $s_0 \in S$ . This definition of an optimal plan is well-defined if all optimal values  $v_U^*(s_0)$  exist and are finite. Clearly, it is important that the optimal values exist because otherwise the expression

<sup>1</sup>We maximize the expected utility of the undiscounted total reward in this paper, which is more relevant for AI applications than the expected utility of the discounted total reward. However, operations researchers have studied how to maximize the expected utility of the discounted total reward (Jaquette, 1976; Bouakiz, 1985; Chung & Sobel, 1987; White, 1987).

that defines an optimal plan is not well-defined. Fortunately, the optimal values are guaranteed to exist since all rewards are strictly negative according to our assumptions (Liu & Koenig, 2005a). However, it is also important that the optimal values be finite. The MDP in Figure 1 has two mappings from states to actions (= stationary deterministic policies = SD policies).  $\pi_1$  assigns the top action to  $s^1$ , and  $\pi_2$  assigns the bottom action to  $s^1$ . Consider  $U(w) = -\left(\frac{1}{2}\right)^w$ . Then,

$$v_U^{\pi_1}(s^1) = \sum_{t=1}^{\infty} \left[ -\left(\frac{1}{2}\right)^{(-1)^t} \cdot (1/2)^t \right] = -\sum_{t=1}^{\infty} 1 = -\infty,$$

$$v_U^{\pi_2}(s^1) = \sum_{t=1}^{\infty} \left[ -\left(\frac{1}{2}\right)^{(-2)^t} \cdot (1/2)^t \right] = -\sum_{t=1}^{\infty} 2^t = -\infty,$$

and  $v_U^{\pi_1}(s^2) = v_U^{\pi_2}(s^2) = -1$ . Thus, the optimal values are  $v_U^*(s^1) = \max(-\infty, -\infty) = -\infty$  and  $v_U^*(s^2) = \max(-1, -1) = -1$ . All trajectories have identical probabilities for both policies, but the total reward and thus also the total utility of each trajectory is larger for policy  $\pi_1$  than policy  $\pi_2$ . Thus, policy  $\pi_1$  should be preferred over policy  $\pi_2$  for all monotonically strictly increasing utility functions. Policy  $\pi_2$  thus demonstrates that a policy that achieves the optimal values and thus is optimal according to our definition is not always the best one. The problem is that plans with infinite values are indistinguishable, and the optimal values thus need to be finite to compare plans in a meaningful way. We assume in this paper that the optimal values are finite. Sufficient conditions to guarantee this property are provided in (Liu & Koenig, 2005a).

## Linear Utility Functions

Assume that the given utility function is linear. Utility functions are defined only up to positive linear transformations. In other words, an agent with utility function  $U$  and an agent with utility function  $U' = b_1U + b_0$  with  $b_1 > 0$  always make the same choices if they break ties identically. Thus, we can assume without loss of generality that  $U(w) = w$ . We omit the subscript  $U$  for linear utility functions.

The optimal values  $v^*(s)$  can be achieved by an SD policy (Bertsekas & Tsitsiklis, 1991). Thus, the current state is the only piece of information needed for following an optimal course of action, which is called the Markov property of optimal policies. The optimal values satisfy the optimality equations (Bertsekas & Tsitsiklis, 1991)

$$v^*(s) = \begin{cases} 0 & s \in G \\ \max_{a \in A} \sum_{s' \in S} P(s'|s, a)(r(s, a, s') + v^*(s')) & s \in S \setminus G. \end{cases}$$

Value iteration assigns a value to every state and updates them iteratively until they (almost) converge. Value iteration calculates the values  $v^t(s)$  that are defined by the equations

$$v^0(s) = 0 \quad s \in S$$

$$v^{t+1}(s) = \begin{cases} 0 & s \in G \\ \max_{a \in A} \sum_{s' \in S} P(s'|s, a)(r(s, a, s') + v^t(s')) & s \in S \setminus G \end{cases}$$

for all  $t \in \mathbb{N}_0$  (Bertsekas & Tsitsiklis, 1991). It holds that  $\lim_{t \rightarrow \infty} v^t(s) = v^*(s)$  for all  $s \in S$ . An optimal SD policy is then greedy with respect to the optimal values. The agent thus maximizes its expected total reward when executing action  $\arg \max_{a \in A} \sum_{s' \in S} P(s'|s, a)(r(s, a, s') + v^*(s'))$  in nongoal state  $s \in S \setminus G$  and stopping in a goal state. Policy iteration can also be used to determine an optimal SD policy (Bertsekas & Tsitsiklis, 1991).

## Exponential Utility Functions

The choices of agents with linear utility functions imply a neutral attitude toward risk and thus often do not take the preferences of human decision makers sufficiently into account. Concave and convex exponential utility functions are the most often used nonlinear utility functions (Corner & Corner, 1995). Assume that the given utility function is convex exponential (that is,  $U_{\text{exp}}(w) = \gamma^w$  with  $\gamma > 1$ ) or concave exponential (that is,  $U_{\text{exp}}(w) = -\gamma^w$  with  $0 < \gamma < 1$ ). We write  $U_{\text{exp}}(w) = \iota \gamma^w$  where  $\iota = 1$  if  $\gamma > 1$  and  $\iota = -1$  if  $0 < \gamma < 1$ . We use the subscript  $\text{exp}$  for exponential utility functions.

Exponential utility functions have the following property: Assume that an agent receives total reward  $r_i$  with probability  $p_i$ . Then, its expected total utility is  $\sum_i p_i U_{\text{exp}}(r_i) = \iota \sum_i p_i \gamma^{r_i}$ . If all total rewards  $r_i$  are increased by a constant  $r$ , then its expected total utility increases to  $\sum_i p_i U_{\text{exp}}(r + r_i) = \iota \sum_i p_i \gamma^{r+r_i} = \gamma^r \cdot \iota \sum_i p_i \gamma^{r_i}$ , that is, by factor  $\gamma^r$ . One consequence of this property is that the choices of the agent do not depend on its wealth level when making the choices (= prior wealth level) because increasing its wealth level by an additive constant  $r$  increases the expected total utilities of all choices by the same multiplicative constant  $\gamma^r$ , which preserves the Markov property of optimal policies and implies that the optimal values can be achieved by an SD policy. The optimal values  $v_{\text{exp}}^*(s)$  satisfy the optimality equations (Patek, 2001)

$$v_{\text{exp}}^*(s) = \begin{cases} U(0) = \iota \gamma^0 = \iota & s \in G \\ \max_{a \in A} \sum_{s' \in S} P(s'|s, a) \gamma^{r(s, a, s')} v_{\text{exp}}^*(s') & s \in S \setminus G. \end{cases}$$

Notice that  $v_{\text{exp}}^*(s')$  is the expected total utility when starting in state  $s'$  and following an optimal course of action. All rewards are now increased by  $r(s, a, s')$  and the expected total utility thus increases to  $\gamma^{r(s, a, s')} v_{\text{exp}}^*(s')$ , which explains the second equation. Value iteration calculates the values  $v_{\text{exp}}^t(s)$  that are defined by the equations

$$v_{\text{exp}}^0(s) = \iota \quad s \in S$$

$$v_{\text{exp}}^{t+1}(s) = \begin{cases} \iota & s \in G \\ \max_{a \in A} \sum_{s' \in S} P(s'|s, a) \gamma^{r(s, a, s')} v_{\text{exp}}^t(s') & s \in S \setminus G \end{cases}$$

for all  $t \in \mathbb{N}_0$  (Patek, 2001). It holds that  $\lim_{t \rightarrow \infty} v_{\text{exp}}^t(s) = v_{\text{exp}}^*(s)$  for all  $s \in S$ . The agent then maximizes its expected total utility when executing action  $\arg \max_{a \in A} \sum_{s' \in S} P(s'|s, a) \gamma^{r(s, a, s')} v_{\text{exp}}^*(s')$  in nongoal state  $s \in S \setminus G$  and stopping in a goal state. Policy iteration can also be used to determine an optimal SD policy (Patek, 2001).

## Arbitrary Nonlinear Utility Functions

The choices of agents with linear and exponential utility functions do not depend on their prior wealth levels. They are called zero-switch utility functions. This name is due to the property that an agent with a zero-switch utility function who is confronted with two courses of action never switches from one course of action to the other one as its wealth level increases. Linear and exponential utility functions are the only zero-switch utility functions (Bell, 1988). The optimal values for other utility functions cannot necessarily be achieved by an SD policy (Liu & Koenig, 2005b), which makes it more difficult to determine an optimal plan for ar-

bitrary nonlinear utility functions.

**Transforming the MDP** To determine an optimal plan, we transform the given MDP into an MDP that we distinguish from the original MDP by enclosing all of its elements in angular brackets. Let  $W$  be the set of possible wealth levels after any number of action executions when starting in any state of the original MDP with wealth level zero. The transformed MDP is characterized by a countably infinite set of states  $\langle S \rangle = S \times W$ , a countably infinite set of goal states  $\langle G \rangle = G \times W$  and a finite set of actions  $\langle A \rangle = A$ . The agent executes an action  $\langle a \rangle \in \langle A \rangle$  of its choice in the current nongoal state  $\langle s \rangle \in \langle S \rangle \setminus \langle G \rangle$ . Assume that  $\langle s \rangle = (s, w)$ ,  $\langle s' \rangle = (s', w')$  and  $\langle a \rangle = a$ . The execution of the action then results in finite reward  $\langle r \rangle_{\langle U \rangle}(\langle s \rangle, \langle a \rangle, \langle s' \rangle) = U(w') - U(w) \leq 0$  and a transition to state  $\langle s' \rangle \in \langle S \rangle$  in the next time step with probability

$$\langle P \rangle_{\langle U \rangle}(\langle s' \rangle | \langle s \rangle, \langle a \rangle) = \begin{cases} P(s' | s, a) & \text{if } w' = w + r(s, a, s') \\ 0 & \text{otherwise.} \end{cases}$$

(Notice that we annotate those values with the utility function that depend on it, such as the rewards.) We can prove that a plan that maximizes the expected total reward for the transformed MDP also maximizes the expected total utility for the original MDP, that is,  $v_{\langle U \rangle}^*(s) = \langle v \rangle_{\langle U \rangle}^*((s, 0)) + U(0)$  for all  $s \in S$  (Liu & Koenig, 2005b).

**Functional Value Iteration** The optimal values  $\langle v \rangle_{\langle U \rangle}^*(\langle s \rangle)$  (= the expected total reward in the transformed MDP when starting in state  $\langle s \rangle$  and following an optimal course of action) can be achieved by an SD policy for the transformed MDP, which corresponds to a mapping from states and wealth levels of the original MDP to actions. They satisfy the optimality equations

$$\langle v \rangle_{\langle U \rangle}^*(\langle s \rangle) = \begin{cases} 0 & \langle s \rangle \in \langle G \rangle \\ \max_{\langle a \rangle \in \langle A \rangle} \sum_{\langle s' \rangle \in \langle S \rangle} \langle P \rangle_{\langle U \rangle}(\langle s' \rangle | \langle s \rangle, \langle a \rangle) \cdot (\langle r \rangle_{\langle U \rangle}(\langle s \rangle, \langle a \rangle, \langle s' \rangle) + \langle v \rangle_{\langle U \rangle}^*(\langle s' \rangle)) & \langle s \rangle \in \langle S \rangle \setminus \langle G \rangle. \end{cases}$$

In principle, value iteration can calculate the values  $\langle v \rangle_{\langle U \rangle}^t(\langle s \rangle)$  that are defined by the equations (Puterman, 1994)

$$\langle v \rangle_{\langle U \rangle}^0(\langle s \rangle) = 0 \quad \langle s \rangle \in \langle S \rangle$$

$$\langle v \rangle_{\langle U \rangle}^{t+1}(\langle s \rangle) = \begin{cases} 0 & \langle s \rangle \in \langle G \rangle \\ \max_{\langle a \rangle \in \langle A \rangle} \sum_{\langle s' \rangle \in \langle S \rangle} \langle P \rangle_{\langle U \rangle}(\langle s' \rangle | \langle s \rangle, \langle a \rangle) \cdot (\langle r \rangle_{\langle U \rangle}(\langle s \rangle, \langle a \rangle, \langle s' \rangle) + \langle v \rangle_{\langle U \rangle}^t(\langle s' \rangle)) & \langle s \rangle \in \langle S \rangle \setminus \langle G \rangle \end{cases}$$

for all  $t \in \mathbb{N}_0$ . However, the number of states of the transformed MDP is countably infinite because the number of wealth levels of the original MDP is countably infinite. Running this version of value iteration is thus not practical. Instead, we proceed as follows: Assume that  $\langle s \rangle = (s, w)$  and define  $V_U^*(s)(w) = \langle v \rangle_{\langle U \rangle}^*(\langle s \rangle) + U(w)$  (= the expected total utility in the original MDP when starting in state  $s$  with wealth level  $w$  and following an optimal course of action). We then rewrite the optimality equations (Liu & Koenig, 2005b)

$$V_U^*(s)(w) = \begin{cases} U(w) & s \in G \\ \max_{a \in A} \sum_{s' \in S} P(s' | s, a) V_U^*(s')(w + r(s, a, s')) & s \in S \setminus G \end{cases} \quad (*)$$

for all  $w \in W$ . Value iteration utilizes that the equations for the same state but different wealth levels are similar since the probabilities do not depend on the wealth level. It exploits this similarity by mapping states  $s$  to value functions  $V_U^*(s)$ , namely functions that map wealth levels  $w$  to values  $V_U^*(s)(w)$ , which is practical for all utility functions whose value functions can be represented and updated with finite representations. We therefore refer to this version of value

iteration as functional value iteration. Functional value iteration calculates the value functions  $V_U^t(s)$  that are defined by the equations (Liu & Koenig, 2005b)

$$V_U^0(s)(w) = U(w) \quad s \in S$$

$$V_U^{t+1}(s)(w) = \begin{cases} U(w) & s \in G \\ \max_{a \in A} \sum_{s' \in S} P(s' | s, a) V_U^t(s')(w + r(s, a, s')) & s \in S \setminus G \end{cases}$$

for all  $w \in W$  and  $t \in \mathbb{N}_0$ . It holds that  $\lim_{t \rightarrow \infty} V_U^t(s)(w) = V_U^*(s)(w)$  for all  $s \in S$  and  $w \in W$ . The agent then maximizes its expected total reward for the transformed MDP and also its expected total utility for the original MDP when executing action  $\arg \max_{a \in A} \sum_{s' \in S} P(s' | s, a) V_U^*(s')(w + r(s, a, s'))$  in nongoal state  $s \in S \setminus G$  if its current wealth level is  $w$  and stopping in a goal state.

**One-Switch Utility Functions** Human decision makers are often risk-averse but become risk-neutral in the limit as their wealth level increases. One-switch utility functions are an important class of nonlinear utility functions that can model risk attitudes of this kind (Bell, 1988). Their name is due to the property that an agent with a one-switch utility function who is confronted with two courses of action switches at most once from one course of action to the other one as its wealth level increases. The only one-switch utility functions that model risk attitudes with the above properties have the form  $U_{\text{one}}(w) = Cw - D\gamma^w$  for  $C, D > 0$  and  $0 < \gamma < 1$  (Bell, 1988). Assume that the given utility function is of this form, which we simply refer to as one-switch utility function in the following.

**Functional Value Iteration:** Functional value iteration can be used to determine the optimal values since the value functions for one-switch utility functions are piecewise one-switch and thus can be represented and updated with finite representations (Liu & Koenig, 2005b). One disadvantage of functional value iteration is that the value functions converge only in the limit and thus are only approximately equal to the optimal values after a finite runtime. It is an open problem to determine the approximation error for functional value iteration and the resulting greedy policy.

**Backward Induction:** A backward-induction method can be used to determine the optimal values exactly but is specific to one-switch utility functions. It is based on the following idea: One-switch utility functions are linear combinations of the identity function and a concave exponential utility function. Thus, there exists a wealth level threshold so that, for every wealth level below the wealth level threshold, the exponential term dominates the other one and the action that maximizes the expected total utility for the concave exponential utility function  $U_{\text{exp}}(w) = -\gamma^w$  also maximizes the expected total utility for the one-switch utility function. This allows us to use policy iteration for concave exponential utility functions to determine the optimal value function for all wealth levels below the wealth level threshold, which provides the inductive foundation of the backward-induction method. The backward-induction method then determines the optimal values for larger and larger wealth levels by calculating first the wealth level at which the currently optimal action might no longer be optimal and then the optimal value functions between this and the previous wealth level using the optimality equations (\*). It terminates when it reaches wealth level zero.

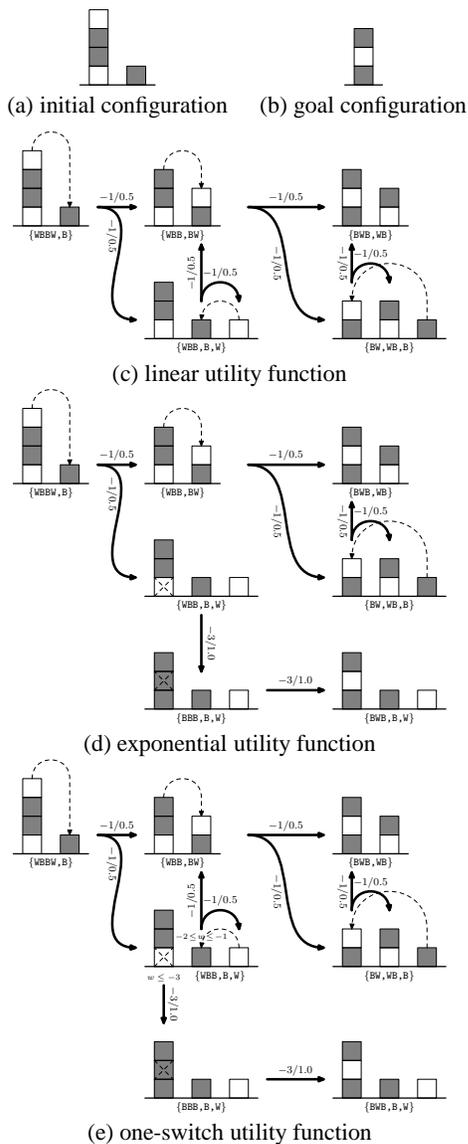


Figure 2: A Blocksworld Problem

### Example

We use a probabilistic blocksworld example (Koenig & Simmons, 1994) to illustrate probabilistic planning with MDPs where one wants to maximize the expected total utility for a given monotonically nondecreasing utility function. The domain is a standard blocksworld domain with five blocks that are either white (W) or black (B). However, the move action succeeds only with probability 0.5. When it fails, the block drops directly onto the table. One can also execute a paint action that changes the color of any one block and always succeeds. The move action has a reward of  $-1$ , and the paint action has a reward of  $-3$ . The initial configuration of blocks is shown in Figure 2(a). The goal is to build a stack of three blocks as shown in Figure 2(b). Figure 2(c)-(e) show the optimal plans, all of which are different, for the linear utility function  $U(w) = w$ , the convex exponential utility function  $U(w) = -0.6^w$  and the one-switch utility function  $U(w) = w - 0.5 \times 0.6^w$ . Notice that the optimal action in one of the states of the latter case depends on the wealth

level. It is a move action for wealth levels  $-1$  and  $-2$  and a paint action for wealth level  $-3$ .

### References

- Bell, D. E. 1988. One-switch utility functions and a measure of risk. *Management Science* 34(12):1416–1424.
- Bertsekas, D. P., and Tsitsiklis, J. N. 1991. An analysis of stochastic shortest path problems. *Mathematics of Operations Research* 16(3):580–595.
- Blythe, J. 1998. *Planning under Uncertainty in Dynamic Domains*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University.
- Bouakiz, M. 1985. *Risk-Sensitivity in Stochastic Optimization with Applications*. Ph.D. Dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology.
- Chung, K.-J., and Sobel, M. J. 1987. Discounted MDP's: Distribution functions and exponential utility maximization. *SIAM Journal of Control and Optimization* 35(1):49–62.
- Corner, J. L., and Corner, P. D. 1995. Characteristics of decisions in decision analysis practice. *The Journal of Operational Research Society* 46:304–314.
- Goodwin, R. T.; Akkiraju, R.; and Wu, F. 2002. A decision-support system for quote-generation. In *Proceedings of the Fourteenth Conference on Innovative Applications of Artificial Intelligence (IAAI-02)*, 830–837.
- Jaquette, S. C. 1976. A utility criterion for Markov decision processes. *Management Science* 23(1):43–49.
- Koenig, S., and Simmons, R. G. 1994. Risk-sensitive planning with probabilistic decision graphs. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR-94)*, 2301–2308.
- Liu, Y., and Koenig, S. 2005a. Existence and finiteness conditions for risk-sensitive planning: Results and conjectures. In *Proceedings of the Twentieth Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*.
- Liu, Y., and Koenig, S. 2005b. Risk-sensitive planning with one-switch utility functions: Value iteration. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, 993–999.
- Patek, S. D. 2001. On terminating Markov decision processes with a risk averse objective function. *Automatica* 37(9):1379–1386.
- Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.
- von Neumann, J., and Morgenstern, O. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.
- White, D. J. 1987. Utility, probabilistic constraints, mean and variance of discounted rewards in Markov decision processes. *OR Spektrum* 9:13–22.
- Zilberstein, S.; Washington, R.; Bernstein, D. S.; and Mouaddib, A.-I. 2002. Decision-theoretic control of planetary rovers. In Beetz, M.; Hertzberg, J.; Ghallab, M.; and Pollack, M. E., eds., *Advances in Plan-Based Control of Robotic Agents*, volume 2466 of *Lecture Notes in Computer Science*. Springer. 270–289.