Functional Value Iteration for Decision-Theoretic Planning with General Utility Functions

Yaxin Liu
Department of Computer Sciences
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

We study how to find plans that maximize the expected total utility for a given MDP, a planning objective that is important for decision making in high-stakes domains. The optimal actions can now depend on the total reward that has been accumulated so far in addition to the current state. We extend our previous work on functional value iteration from one-switch utility functions to all utility functions that can be approximated with piecewise linear utility functions (with and without exponential tails) by using functional value iteration to find a plan that maximizes the expected total utility for the approximate utility function. Functional value iteration does not maintain a value for every state but a value function that maps the total reward that has been accumulated so far into a value. We describe how functional value iteration represents these value functions in finite form, how it performs dynamic programming by manipulating these representations and what kinds of approximation guarantees it is able to make. We also apply it to a probabilistic blocksworld problem, a standard test domain for decision-theoretic planners.

Introduction

Decision-theoretic planning researchers believe that Markov decision process models (MDPs) provide a good foundation for decision-theoretic planning (Boutilier, Dean, & Hanks, 1999; Blythe, 1999). Typically, they find plans for MDPs that maximize the expected total reward. For this planning objective, the optimal actions depend on the current state only. However, decision-theoretic planning researchers also believe that it is sometimes important to maximize the expected utility of the total reward (= expected total utility) for a given monotonically nondecreasing utility function. For example, utility theory suggests that human decision makers maximize the expected total utility in single-instance high-stake planning situations, where their utility functions characterize their attitude toward risk (von Neumann & Morgenstern, 1944; Pratt, 1964). Examples of such situations include environmental crisis situations (Cohen *et al.*, 1989; Blythe, 1998), business decisions situations (Murthy *et al.*, 1999; Goodwin, Akkiraju, & Wu, 2002), and planning situations in space (Pell *et al.*, 1998; Zilberstein *et al.*, 2002), all of which are currently solved without taking risk attitudes into consideration. The question then arises how to find plans for MDPs that maximize the expected total utility. This is a challenge because the optimal actions can then depend on the total reward that has been accumulated so far (= the current wealth level) in addition to the current state (Liu & Koenig, 2005b). We showed in previous publications that, in principle, a version of value iteration can be used to find a plan that maximizes the expected total utility for an arbitrary utility function if it maps every pair of state and wealth level into a value (Liu & Koenig, 2005b). We then developed such a version of value iteration, functional value iteration, that maintains a value function for every state that maps wealth levels into values. Functional value iteration is practical only if there exist finite representations of these value functions. We applied functional value iteration to one-switch utility functions, for which the value functions are piecewise one-switch and thus can be represented in finite form (Liu & Koenig, 2005b). In this paper, we develop a more general approach that approximates a large class of utility functions with piecewise linear utility functions (with and without exponential tails). It then uses functional value iteration to find a plan that maximizes the expected total utility for the approximate utility function. We describe how functional value iteration can represent the value functions in finite form, how it performs dynamic programming by manipulating these representations and what kinds of approximation guarantees it is able to make. We then apply it to a probabilistic blocksworld problem to demonstrate how the plan that maximizes the expected total utility depends on the utility function.

Decision-Theoretic Planning

We perform decision-theoretic planning on MDPs with action costs and want to find a plan that maximizes the expected total utility until plan execution stops, which only happens when a goal state has been reached. We now define our MDPs and this planning objective more formally.

Our MDPs consist of a finite nonempty set of states S, a finite non-empty set of goal states $G \subseteq S$, and a finite nonempty set of actions A for each nongoal state $s \in S \setminus G$. An 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 executing actions, which implies that it no longer receives rewards in the future. Otherwise, it executes

an action $a_t \in A$ of its choice, receives a strictly negative finite reward $r(s_t, a_t, s_{t+1}) < 0$ in return, and transitions with probability $P(s_{t+1}|s_t, a_t)$ to state $s_{t+1} \in S$, where the process repeats at time step $t + 1$.

We denote the set of all possible behaviors of the agent (= plans; more precisely: randomized, history-dependent policies) by Π and now describe the objective of the agent when choosing a plan. The agent is given a monotonically nondecreasing utility function U. Assume for now that the time horizon T is finite. The wealth level of the agent at time step t is $w_t = \sum_{\tau=0}^{t-1} r(s_\tau, a_\tau, s_{\tau+1})$, the total reward that it has accumulated before time step t. Its wealth level thus starts at zero and then decreases as t increases. If the agent starts in state $s_0 \in S$ and follows plan $\pi \in \Pi$, then the expected utility of its final wealth (= 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 π in start state s_0. From now on, we assume that the time horizon T is infinite. If the agent starts in state $s_0 \in S$ and follows plan $\pi \in \Pi$, then the expected total utility is $v_U^{\pi}(s_0) = \lim_{T\to\infty} v_{U,T}^{\pi}(s_0)$. 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 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 $\pi^* \in \Pi$ with $v_U^{\pi^*}(s_0) = v_U^*(s_0)$ for all states $s_0 \in S$. In this paper, we assume that the optimal values $v_U^*(s_0)$ are finite for all states $s_0 \in S$ because otherwise the concept of an optimal plan is not well-defined (Liu & Koenig, 2005a).

Functional Value Iteration

In (Liu & Koenig, 2005b), we investigated one way of finding plans that maximize the expected total utility for a given MDP. We transformed the given MDP into an "augmented" MDP whose states are pairs (s, w), where s is a state of the given MDP and w is a wealth level that can be attained when executing actions in the given MDP. A plan that maximizes the expected total reward for the augmented MDP also maximizes the expected total utility for the given MDP. It is sufficient to consider stationary deterministic policies to maximize the expected total reward for the augmented MDP, resulting in a mapping from states and wealth levels to actions. In principle, one could use value iteration on the augmented MDP to find a plan that maximizes the expected total reward for the augmented MDP (Puterman, 1994) but this is not practical since the augmented MDP has an infinite number of states. Therefore, we developed a version of value iteration, called functional value iteration, that maintains a value function for each state of the given MDP (Liu & Koenig, 2005b). The value functions $V_U(s)$ map real-valued wealth levels w to real values $V_U(s)(w)$. (Our MDPs have strictly negative rewards, and all wealth levels are therefore nonpositive. We therefore use value functions that map nonpositive wealth levels to real values.) The value functions $V_U^t(s)$ at iteration t of functional value iteration are defined by the system of equations

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

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

for all $w \in \mathbb{R}_0^-$ (the nonpositive real values) and $t \in \mathbb{N}_0$. The values $V_U^t(s)(w)$ converge as t increases to the optimal values $V_U^*(s)(w)$, which are the largest expected total utilities that the agent can obtain if it starts in state s with wealth level w, there fore $V_U^*(s)(0) = v_U^*(s)$. The agent then maximizes its expected total utility for every state by executing action $\arg\max_{a\in A} \sum_{s'\in S} P(s'|s,a) V_U^*(s')(w + r(s,a,s'))$ if its current state is nongoal state s and its current wealth level is w. (It stops if its current state is a goal state.) Additional details and proofs are given in (Liu & Koenig, 2005b).

Classes of Utility Functions

So far, functional value iteration has only been applied to restricted classes of utility functions, namely one-switch utility functions (Liu & Koenig, 2005b). We now develop a more general approach that approximates a large class of utility functions with piecewise linear utility functions (with and without exponential tails) and then uses functional value iteration to find a plan that maximizes the expected total utility for the approximate utility function. Note that such utility functions do not include one-switch utility functions. We assume for convenience of exposition that all utility functions are continuous. Our discussion, however, applies also to utility functions with a finite number of discontinuities with straightforward extensions.

Functional value iteration needs to calculate $\max_{a\in A} \sum_{s'\in S} P(s'|s,a) V_U^t(s')(w + r(s,a,s'))$. Thus, it needs to shift a value function $V_U^t(s')(w)$ by $r(s,a,s')$ to calculate $V_U(s')(w + r(s,a,s'))$. It needs to calculate the weighted average $\sum_{s'\in S} P(s'|s,a) V_U^t(s')(w + r(s,a,s'))$ of several value functions $V_U^t(s')(w + r(s,a,s'))$. Finally, it needs to calculate the maximum of several value functions $\sum_{s'\in S} P(s'|s,a) V_U^t(s')(w + r(s,a,s'))$. We restrict the class of utility functions to piecewise linear utility functions (with and without exponential tails) to allow functional value iteration to perform these three operations efficiently.[1] We then use these utility functions to approximate a large class of utility functions. A piecewise linear function f without an exponential tail (PWL function) can be represented as an ordered list of triples (w^i, k^i, b^i) for $i = 1, \ldots, n$ with the following properties: $-\infty = w^0 < w^1 < \cdots < w^n = 0$ and $f(w) = k^i w + b^i$ for all $w \in (w^{i-1}, w^i]$ and $i = 1, \ldots, n$. The values w^i are called breakpoints. A PWL utility function results in PWL value functions, as we show below. A piecewise linear function f with an exponential tail differs from a PWL function only for all $w \in (w^0, w^1]$ where $f(w) = -c^1\gamma^w + b^1$ for given constants $c^1 > 0$ and $0 < \gamma < 1$. A piecewise linear utility function with an

[1] We use operator and function overloading to implement these three operations in C++. The code of functional value iteration then is identical to the code of (regular) value iteration.

exponential tail results in so-called piecewise linex value functions, as we show below. A piecewise linex function (PWLinex function) f can be represented as an ordered list of quadruples (w^i, k^i, c^i, b^i) for $i = 1, \ldots, n$ with the following properties: $-\infty = w^0 < w^1 < \cdots < w^n = 0$ and $f(w) = k^i w - c^i \gamma^w + b^i$ for all $w \in (w^{i-1}, w^i]$ and $i = 1, \ldots, n$.

Shift Functional value iteration needs to shift a value function $V(w)$ by a given constant r to calculate the value function $V(w + r)$. Assume that a PWL value function $V(w)$ is represented as (w^i, k^i, b^i) for $i = 1, \ldots, n$. It holds that $V(w + r) = k^i(w + r) + b^i = k^i w + (k^i r + b^i)$ for $w + r \in (w^{i-1}, w^i]$ or, equivalently, $w \in (w^{i-1} - r, w^i - r]$. The new value function $V(w + r)$ is again PWL and can be represented as $(w^i - r, k^i, k^i r + b^i)$ for $i = 1, \ldots, n$. Similarly, assume that a PWLinex value function $V(w)$ is represented as (w^i, k^i, c^i, b^i) for $i = 1, \ldots, n$. Then, the value function $V(w+r)$ is again PWLinex and can be represented as $(w^i - r, k^i, c^i \gamma^r, k^i r + b^i)$ for $i = 1, \ldots, n$. In both cases, one simplifies the new value function (and speeds up future operations on it) by keeping only the part for $w \leq 0$.

Weighted Average Functional value iteration needs to calculate the weighted average of several value functions. Assume without loss of generality that there are two PWL value functions V and $\hat{V}$. One first introduces additional breakpoints, if needed, without changing the value functions to ensure that both value functions have the same breakpoints. Assume that the two resulting PWL value functions are represented as (w^i, k^i, b^i) and $(w^i, \hat{k}^i, \hat{b}^i)$ for $i = 1, \ldots, n$. Then, their weighted average $pV + q\hat{V}$ is again PWL and can be represented as $(w^i, pk^i + q\hat{k}^i, pb^i + q\hat{b}^i)$ for $i = 1, \ldots, n$. Similarly, assume that two PWLinex value functions V and $\hat{V}$ are represented as (w^i, k^i, c^i, b^i) and $(w^i, \hat{k}^i, \hat{c}^i, \hat{b}^i)$ for $i = 1, \ldots, n$. Then, their weighted average $pV + q\hat{V}$ is again PWLinex and can be represented as $(w^i, pk^i + q\hat{k}^i, pc^i + q\hat{c}^i, pb^i + q\hat{b}^i)$ for $i = 1, \ldots, n$.

Maximum Functional value iteration needs to calculate the maximum of several value functions. Assume without loss of generality that there are two PWL value functions V and $\hat{V}$. One first introduces additional breakpoints, if needed, without changing the value functions to ensure that all value functions have the same breakpoints. Assume that the two resulting PWL value functions are represented as (w^i, k^i, b^i) and $(w^i, \hat{k}^i, \hat{b}^i)$ for $i = 1, \ldots, n$. Then, their maximum $\max(V, \hat{V})$ is again PWL. Consider any $i = 1, \ldots, n$ and assume without loss of generality that $V(w^{i-1}) \geq \hat{V}(w^{i-1})$ and $V(w) \neq V'(w)$ for some $w \in (w^{i-1}, w^i)$. The two value functions can intersect at zero or one intersection point $\bar{w}$ with $w^{i-1} \leq \bar{w} < w^i$ and

$$V(\bar{w}) = \hat{V}(\bar{w})$$
$$k^i \bar{w} + b^i = \hat{k}^i \bar{w} + \hat{b}^i.$$

We distinguish two cases:

- Value function V dominates the other one for all $w \in (w^{i-1}, w^i]$, which is the case iff $V(w^i) \geq \hat{V}(w^i)$. Then, value function V is the maximum of both value functions for all $w \in (w^{i-1}, w^i]$. The maximum can thus be represented as (w^i, k^i, b^i) for $w \in (w^{i-1}, w^i]$.
- The two value functions intersect at $\bar{w}$ with $w^{i-1} \leq \bar{w} < w^i$. One then adds the intersection point $\bar{w} = (\hat{b}^i - b^i)/(k^i - \hat{k}^i)$ as a new breakpoint to the new value function. Then, value function V is the maximum of both value functions for all $w \in (w^{i-1}, \bar{w}]$, and value function $\hat{V}$ is the maximum of both value functions for all $w \in (\bar{w}, w^i]$. The maximum can thus be represented as $(\bar{w}, k^i, b^i)$ for $w \in (w^{i-1}, \bar{w}]$ (this interval can be empty if $\bar{w} = w^{i-1}$) and $(w^i, \hat{k}^i, \hat{b}^i)$ for $w \in (\bar{w}, w^i]$.

Similarly, assume that two PWLinex value functions V and $\hat{V}$ are represented as (w^i, k^i, c^i, b^i) and $(w^i, \hat{k}^i, \hat{c}^i, \hat{b}^i)$ for $i = 1, \ldots, n$. Then, their maximum is again PWLinex. Consider any $i = 1, \ldots, n$. The two value functions can intersect at zero, one or two intersection points $\bar{w}$ with $w^{i-1} \leq \bar{w} < w^i$ and

$$V(\bar{w}) = \hat{V}(\bar{w})$$
$$k^i \bar{w} - c^i \gamma^{\bar{w}} + b^i = \hat{k}^i \bar{w} - \hat{c}^i \gamma^{\bar{w}} + \hat{b}^i$$
$$b^i - \hat{b}^i = \left(\hat{k}^i - k^i\right) \bar{w} - \left(\hat{c}^i - c^i\right) \gamma^{\bar{w}}.$$

In general, this equation can only be solved with numerical methods such as the Newton-Raphson method. One then adds the intersection points, if any, as new breakpoints to the new value function and then proceeds as for PWL value functions. In both cases, one can simplify the new value function (and speeds up future operations on it) by merging adjacent segments of it, if possible, to remove unnecessary breakpoints.

Termination of Functional Value Iteration We now show the convergence properties of the value functions for nongoal states as the number of iterations of functional value iteration increases. (The value functions for goal states are simply the utility function itself.) We calculate after how many iterations t of functional value iteration the error $|V_U^t(s)(w) - V_U^*(s)(w)|$ of all values $V_U^t(s)(w)$ is no larger than a given constant $\epsilon > 0$.

We first consider PWL value functions, starting with $w \leq w^1$. The values $v^t(s)$ at iteration t of (regular) value iteration for maximizing the expected total reward of the given MDP are defined by the system of equations

$$v^0(s) = 0 \qquad s \in S$$
$$v^{t+1}(s) = \begin{cases} 0 & s \in G \\ \max\limits_{a \in A} \sum\limits_{s' \in S} P(s'|s,a)(r(s,a,s') + v^t(s')) & s \notin G \end{cases}$$

for all $t \in \mathbb{N}_0$ (Puterman, 1994). We now show by induction on t that $V_U^t(s)(w) = k^1 w + k^1 v^t(s) + b^1$ for all $w \leq w^1$, $s \in S \setminus G$ and $t \in \mathbb{N}_0$. The property holds trivially for iteration $t = 0$ since $V_U^t(s)(w) = k^1 w + b^1 = k^1 w + k^1 0 + b^1 = k^1 w + k^1 v^t(s) + b^1$. Assume that it holds for some iteration t. Then,

$$V_U^{t+1}(s)(w)$$
$$= \max_{a \in A} \sum_{s' \in S} P(s'|s,a) V_U^t(s')(w + r(s,a,s'))$$

$$= \max_{a \in A} \sum_{s' \in S} P(s'|s,a) \left(k^1(w + r(s,a,s')) + k^1 v^t(s') + b^1\right)$$
$$= k^1 w + k^1 \left(\max_{a \in A} \sum_{s' \in S} P(s'|s,a)(r(s,a,s') + v^t(s'))\right) + b^1$$
$$= k^1 w + k^1 v^{t+1}(s) + b^1$$

for all $w \leq w^1$ and $s \in S \setminus G$, which proves the property. The values $v^t(s)$ are monotonically nonincreasing in t and converge as t increases to the optimal values $v^*(s)$. Thus, the values $V_U^t(s)(w)$ are monotonically nonincreasing in t and converge to

$$V_U^*(s)(w) = \lim_{t \to \infty} V_U^t(s)(w) = \lim_{t \to \infty} \left(k^1 w + k^1 v^t(s) + b^1\right)$$
$$= k^1 w + k^1 v^*(s) + b^1$$

for all $w \leq w^1$ and $s \in S \setminus G$. The error of value $V_U^t(s)(w)$ is therefore

$$|V_U^t(s)(w) - V_U^*(s)(w)| = V_U^t(s)(w) - V_U^*(s)(w)$$
$$= k^1 \left(v^t(s) - v^*(s)\right)$$

for all $w \leq w^1$ and $s \in S \setminus G$. This error is no larger than $\epsilon > 0$ for all $t \geq t^*$ provided that $k^1 \max_{s \in S}(v^{t^*}(s) - v^*(s)) \leq \epsilon$. If $k^1 = 0$, then we can simply use $t^* = 0$. Otherwise, one can easily find a t^* with the desired property since the values $v^t(s)$ and $v^*(s)$ can be calculated with (regular) value iteration and policy iteration, respectively, for maximizing the expected total reward of the given MDP (Puterman, 1994).

We now show by transfinite induction on w that the error of all values $V_U^t(s)(w)$ is no larger than ϵ for all $w \in \mathbb{R}_0^-$ and $s \in S \setminus G$ if $t \geq t^* + \left\lceil \frac{w^1 - \max(w, w^1)}{\underline{r}} \right\rceil$, where $\underline{r} = \max_{s \in S \setminus G, a \in A, s' \in S} r(s,a,s')$. We have already shown that this property holds for $w \in (-\infty, w^1]$ since $t \geq t^* + \left\lceil \frac{w^1 - \max(w, w^1)}{\underline{r}} \right\rceil$ implies $t \geq t^*$. Assume that it holds for $w \in (-\infty, \hat{w})$ with $\hat{w} \geq w^1$. We now show that it then also holds for $w = \hat{w}$. If $t \geq t^* + \left\lceil \frac{w^1 - \max(\hat{w}, w^1)}{\underline{r}} \right\rceil = t^* + \left\lceil \frac{w^1 - \hat{w}}{\underline{r}} \right\rceil$, then

$$V_U^t(s)(\hat{w}) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a) V_U^{t-1}(s')(\hat{w} + r(s,a,s'))$$
$$\leq \max_{a \in A} \sum_{s' \in S} P(s'|s,a) \left(V_U^*(s')(\hat{w} + r(s,a,s')) + \epsilon\right)$$
$$= \max_{a \in A} \sum_{s' \in S} P(s'|s,a) V_U^*(s')(\hat{w} + r(s,a,s')) + \epsilon$$
$$= V_U^*(s)(\hat{w}) + \epsilon,$$

where the preceeding inequality holds due to the induction assumption since, first, $\hat{w} + r(s,a,s') < \hat{w}$ and thus $\hat{w} + r(s,a,s') \in (-\infty, \hat{w})$ and, second,

$$t - 1 \geq t^* + \left\lceil \frac{w^1 - \hat{w}}{\underline{r}} \right\rceil - 1 = t^* + \left\lceil \frac{w^1 - \hat{w} - \underline{r}}{\underline{r}} \right\rceil$$
$$\geq t^* + \left\lceil \frac{w^1 - (\hat{w} + r(s,a,s'))}{\underline{r}} \right\rceil$$
$$\geq t^* + \left\lceil \frac{w^1 - \max(\hat{w} + r(s,a,s'), w^1)}{\underline{r}} \right\rceil.$$

This proves the property, which implies that the error of all values $V_U^t(s)(w)$ is no larger than ϵ for all $w \in \mathbb{R}_0^-$ and $s \in S \setminus G$ if $t \geq t^* + \left\lceil \frac{w^1}{\underline{r}} \right\rceil$. Thus, we have derived a termination condition of functional value iteration for PWL utility functions.

We now consider PWLinex value functions in a similar fashion, starting with $w \leq w^1$. The values $v_{\mathrm{e}}^t(s)$ at iteration t of (regular) value iteration for maximizing the expected total utility of the given MDP for the exponential utility function $U(w) = -\gamma^w$ are defined by the system of equations

$$v_{\mathrm{e}}^0(s) = -1 \qquad s \in S$$
$$v_{\mathrm{e}}^{t+1}(s) = \begin{cases} -1 & s \in G \\ \max_{a \in A} \sum_{s' \in S} P(s'|s,a) \gamma^{r(s,a,s')} v_{\mathrm{e}}^t(s') & s \notin G \end{cases}$$

for all $t \in \mathbb{N}_0$ (Patek, 2001). Then, one can show by induction on t that $V_U^t(s)(w) = c^1 v_{\mathrm{e}}^t(s) \gamma^w + b^1$ for all $w \leq w^1$, $s \in S \setminus G$ and $t \in \mathbb{N}_0$. The values $v_{\mathrm{e}}^t(s)$ are monotonically nonincreasing in t and converge as t increases to the optimal values $v_{\mathrm{e}}^*(s)$. Thus, the values $V_U^t(s)(w)$ are monotonically nonincreasing in t and converge to $V_U^*(s)(w) = c^1 v_{\mathrm{e}}^*(s) \gamma^w + b^1$ for all $w \leq w^1$ and $s \in S \setminus G$. The error of value $V_U^t(s)(w)$ is therefore $|V_U^t(s)(w) - V_U^*(s)(w)| = V_U^t(s)(w) - V_U^*(s)(w) = c^1 \left(v_{\mathrm{e}}^t(s) - v_{\mathrm{e}}^*(s)\right) \gamma^w$. It is difficult to find a termination condition for functional value iteration because γ^w increases unbounded as w decreases. We therefore simply run functional value iteration until the w^i, k^i, c^i and b^i parameters of the representations of the PWLinex value functions converge numerically, which implies that all values $V_U^t(s)(w)$ have converged numerically.

Approximating Value Functions

Consider two arbitrary monotonically nondecreasing utility functions U and U'. We now show that the two optimal value functions $V_U^*(s)$ and $V_{U'}^*(s)$ are close to each other if the two utility functions are close to each other. More formally, we show that $0 \leq V_U^*(s)(w) - V_{U'}^*(s)(w) \leq \epsilon$ for all $s \in S$ and $w \in \mathbb{R}_0^-$ if $0 \leq U(w) - U'(w) \leq \epsilon$ for all $w \in \mathbb{R}_0^-$. This implies that one can approximate the optimal value function $V_U^*(s)$ for a given utility function U by using functional value iteration to calculate the optimal value function $V_{U'}^*(s)$ for a utility function U' that approximates utility function U.

We prove by induction on t that $0 \leq V_U^t(s)(w) - V_{U'}^t(s)(w) \leq \epsilon$ for all $s \in S$, $w \in \mathbb{R}_0^-$ and $t \in \mathbb{N}_0$ if $0 \leq U(w) - U'(w) \leq \epsilon$ for all $w \in \mathbb{R}_0^-$. This property holds trivially for $t = 0$ since $V_U^0(s)(w) = U(w)$ and $V_{U'}^0(s)(w) = U'(w)$ for all $s \in S$ and $w \in \mathbb{R}_0^-$. Assume that it holds for some iteration t. It holds trivially for $s \in G$ at iteration $t+1$ since $V_U^{t+1}(s)(w) = U(w)$ and $V_{U'}^{t+1}(s)(w) = U'(w)$ for all $s \in G$ and $w \in \mathbb{R}_0^-$. Furthermore,

$$V_U^{t+1}(s)(w) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a) V_U^t(s')(w + r(s,a,s'))$$
$$\geq \max_{a \in A} \sum_{s' \in S} P(s'|s,a) V_{U'}^t(s')(w + r(s,a,s'))$$
$$= V_{U'}^{t+1}(s)(w)$$

and

$$V_U^{t+1}(s)(w) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a) V_U^t(s')(w + r(s,a,s'))$$

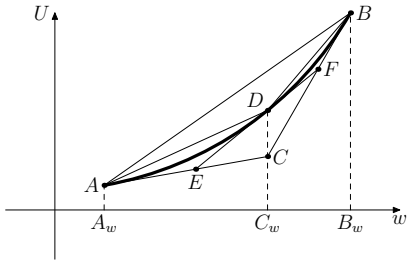


Figure 1: Sandwich method

$$\leq \max_{a \in A} \sum_{s' \in S} P(s'|s,a) \left(V^t_{U'}(s')(w + r(s,a,s')) + \epsilon \right)$$
$$= \max_{a \in A} \sum_{s' \in S} P(s'|s,a) V^t_{U'}(s')(w + r(s,a,s')) + \epsilon$$
$$= V^{t+1}_{U'}(s)(w) + \epsilon$$

for all $s \in S \setminus G$ and $w \in \mathbb{R}_0^-$. Thus, it holds that $0 \leq V_U^{t+1}(s)(w) - V_{U'}^{t+1}(s)(w) \leq \epsilon$ for all $s \in S$, $w \in \mathbb{R}_0^-$ and $t \in \mathbb{N}_0$. All values converge as t increases and it follows that $0 \leq V_U^*(s)(w) - V_{U'}^*(s)(w) \leq \epsilon$ for all $s \in S$ and $w \in \mathbb{R}_0^-$.

Approximating Utility Functions

We now show how to approximate given utility functions U from above with piecewise-linear utility functions $\overline{U}$ (with and without exponential tails) so that the error $\overline{U}(w) - U(w)$ is no larger than ϵ for all $w \in \mathbb{R}_0^-$. Utility functions U can be approximated closely by PWL utility functions if they are asymptotically linear, that is, if there exist constants $k_- \geq 0$ and b_- such that

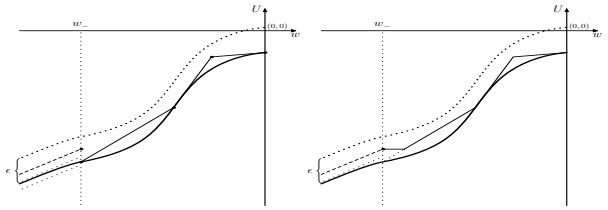
$$\lim_{w \to -\infty} (U(w) - k_- w) = b_-.$$

We first find an upper PWL approximation of utility function U for $w \leq w^1$ with a value w^1 to be determined. There exists a constant w_- such that for all $w \leq w_-$

$$b_- - \frac{\epsilon}{2} \leq U(w) - k_- w \leq b_- + \frac{\epsilon}{2}$$
$$U(w) \leq k_- w + b_- + \frac{\epsilon}{2} \leq U(w) + \epsilon.$$

For $w \leq w_-$, the linear function $\overline{U}_1(w) = k_- w + b_- + \frac{\epsilon}{2}$ is thus an upper PWL approximation of utility function U with an error that is no larger than ϵ. We set $w^1 = w_-$.

We now find an upper PWL approximation of utility function U for $w > w_-$. We first assume that utility function U for $w > w_-$ can be divided into a finite number of segments so that each segment is either convex or concave. The sandwich method can then be used to find an upper PWL approximation for each segment (Rote, 1992), as illustrated in Figure 1 for the segment AB of the utility function represented by the thick curve. The first upper PWL approximation consists of the line segment AB that connects the two endpoints of the segment. The first lower PWL approximation consists of the line segments AC and CB that are part of the tangent lines at the two endpoints, respectively, with intersection point C. If the error of the upper PWL approximation is greater than ϵ, the sandwich method is recursively applied to the two segments AD and DC of the utility function, where D is the point on the utility function that has the same w coordinate as point C. The second upper PWL approximation, for example, consists of the line segments AD and DB, and the second lower PWL approximation consists of the line segments AE, ED, DF and FB. (The line segments ED and DF can then be joined into one line segment EF.) The upper PWL approximation of the segment of the utility function shares its end points with the end points of the segment itself. Thus, for $w \geq w_-$, the upper PWL approximations of all segments of utility function U form a continuous upper PWL approximation $\overline{U}_2$ of utility function U of any desired error, including an error that is no larger than ϵ.



(a) Approximating the two parts (b) Removing discontinuity

Figure 2: Approximating asymptotically linear utility functions

We now put the upper PWL approximations $\overline{U}_1$ (for $w \leq w_-$) and $\overline{U}_2$ (for $w \geq w_-$) together to an upper PWL approximation $\overline{U}$ of utility function U for all $w \in \mathbb{R}_0^-$. We cannot use

$$\overline{U}(w) = \begin{cases} \overline{U}_1(w) & w \in (-\infty, w_-] \\ \overline{U}_2(w) & w \in (w_-, 0] \end{cases}$$

as an upper PWL approximation with an error that is no larger than ϵ (given that its two parts have at most this error) because this approximation can be discontinuous and not monotonically nondecreasing at $w = w_-$, as Figure 2(a) shows. Thus, we use

$$\overline{U}(w) = \begin{cases} \overline{U}_1(w) & w \in (-\infty, w_-] \\ \max(\overline{U}_1(w_-), \overline{U}_2(w)) & w \in (w_-, 0] \end{cases}$$

as a PWL approximation that is continuous and monotonically nondecreasing. The only difference between this PWL approximation and the previous one is if $\max(\overline{U}_1(w_-), \overline{U}_2(w)) = \overline{U}_1(w_-)$ for some $w \in (w_-, 0]$. It then holds that $0 \leq \overline{U}_2(w) - U(w) \leq \max(\overline{U}_1(w_-), \overline{U}_2(w)) - U(w) = \overline{U}_1(w_-) - U(w) \leq U(w_-) + \epsilon - U(w) \leq \epsilon$ because the utility function U is monotonically nondecreasing. Thus, the PWL approximation $\overline{U}$ is indeed an upper PWL approximation of utility function U with an error that is no larger than ϵ.

Now assume that functional value iteration runs on the upper PWL approximation $\overline{U}$ of utility function U for a sufficient number of iterations (as determined earlier) so that the error is at most ϵ. Then,

$$0 \leq \overline{U}(w) - U(w) \leq \epsilon$$
$$0 \leq V^*_{\overline{U}}(s)(w) - V^*_U(s)(w) \leq \epsilon$$
$$0 \leq V^t_{\overline{U}}(s)(w) - V^*_{\overline{U}}(s)(w) \leq \epsilon$$

and thus

$$0 \leq V^t_{\overline{U}}(s)(w) - V^*_U(s)(w) \leq 2\epsilon.$$

for all $w \in \mathbb{R}_0^-$ and $s \in S$. Therefore, the resulting value functions for the upper PWL approximation $\overline{U}$ of utility function U are upper approximations of the optimal value functions for the utility function itself, with an error that is

Figure 3: Probabilistic blocksworld

no larger than 2ϵ. This is the main result of this paper.

One can also derive a lower PWL approximation $\underline{U}$ of utility function U of error ϵ in an analogous way (Liu, 2005) and then run functional value iteration on the lower PWL approximation $\underline{U}$ for a sufficient number of iterations so that the error is at most ϵ. Then,

$$-\epsilon \leq \underline{U}(w) - U(w) \leq 0$$
$$-\epsilon \leq V^*_{\underline{U}}(s)(w) - V^*_U(s)(w) \leq 0$$
$$0 \leq V^t_{\underline{U}}(s)(w) - V^*_{\underline{U}}(s)(w) \leq \epsilon$$
$$-\epsilon \leq V^t_{\underline{U}}(s)(w) - V^*_U(s)(w) \leq \epsilon$$

for all $w \in \mathbb{R}^-_0$ and $s \in S$. Thus, there is a trade-off. The resulting value functions for the lower PWL approximation of the utility function are not necessarily lower approximations of the optimal value functions for the utility function itself but have an error that is no larger than ϵ rather than 2ϵ.

Finally, utility functions U can be approximated closely by piecewise linear utility functions with exponential tails if they are asymptotically exponential, that is, if there exist constants $0 < \gamma < 1$, $k_- < 0$ and b_- such that

$$\lim_{w \to -\infty} (U(w) - k_- \gamma^w) = b_-.$$

We can find upper and lower PWL approximations $\overline{U}$ and $\underline{U}$ with exponential tails for a given asymptotically exponential utility function U so that the errors $\overline{U}(w) - U(w)$ and $U(w) - \underline{U}(w)$ are no larger than ϵ for all $w \in \mathbb{R}^-_0$ by proceeding in an analogous way as above. We can then run functional value iteration on the upper or lower PWL approximations with exponential tails to approximate the optimal value functions for the utility function itself, as before. The details can be found in (Liu, 2005, Chapter 4).

Probabilistic Blocksworld with Deadlines

We now apply functional value iteration to a probabilistic blocksworld, a standard test domain for decision-theoretic planners (Koenig & Simmons, 1994). Our probabilistic blocksworld is identical to the standard deterministic blocksworld except that one can execute two kinds of actions in each configuration of blocks and their effects can be nondeterministic. In particular, the move action succeeds only with probability 0.5. When it fails, the block drops directly onto the table. (Thus, moving a block to the table always succeeds.) There is also a paint action that changes the color of any one block and always succeeds. The move action takes one time unit to execute and thus has a reward of -1, and the paint action takes three time units to execute and thus has a reward of -3. Figure 3 shows the start configuration of blocks. The goal is to build a stack of three blocks: black (at the bottom), white, and black (on top). The remaining two blocks can be anywhere and can have any color. The probabilistic blocksworld example has 162 states, which we describe as a set of stacks by listing the blocks in each stack from bottom to top, using `W` for a white block and `B` for a black block. The start configuration of blocks, for example, is $\hat{s} = \{$`WBBW,B`$\}$. We use functional value iteration to find a plan that maximizes its expected total utility for every start state for utility functions that model a preference for completing the task by a given deadline (Haddawy & Hanks, 1992). Deadlines can be either hard or soft:

Table 1: Optimal values for hard deadlines d

d	$[0,-2)$	$[-2,-3)$	$[-3,-4)$	$[-4,-5)$	$[-5,-6)$	$[-6,-7)$	$[-7,-\infty)$
$v^*_U(\hat{s})$	0	0.25	0.5	0.6875	0.8125	0.890625	1.0

- If the deadline is hard, then the utility of completing the task after the deadline sharply drops to zero. Consider a hard deadline of x time units. The corresponding utility function has a step at $d = -x$, as shown in Figure 4(a). Table 1 shows the optimal value $v^*_U(\hat{s}) = V^*_U(\hat{s})(0)$ of the start configuration of blocks with wealth level zero for the probabilistic blocksworld as determined by functional value iteration. These values correspond to the probabilities of being able to complete the task by the deadline. For example, the optimal value is one for $d \leq -7$ because painting the two bottom blocks of the four-block stack and then removing its top block achieves the goal for sure with a total reward of -7. Similarly, the optimal value is zero for $0 < d < -2$ because there is no way of achieving the goal with a total reward of -2 or larger.
- If the deadline is soft, then the utility decreases gradually after the deadline. Soft deadlines can be modeled with a variety of utility functions, depending on how the utility decreases after the deadline. We consider the three utility functions shown in Figure 4, namely (b) the linear case, where the utility decreases linearly after the deadline d, (c) the exponential case, where the utility decreases exponentially after the deadline d, and (d) the mixed case, where the utility decreases linearly after the earlier deadline d and exponentially after the later deadline d''.

Figure 7 shows the optimal plans for various utility functions. The optimal plans include only configurations of blocks that are reachable from the start configuration of blocks. We chose the parameters of the utility functions so that different plans result: (a) is a hard deadline with $d = -5$, (b) is a soft deadline (linear case) with $d = -6.75$ and $d' = -7.75$, (c) is a soft deadline (exponential case) with $\gamma = 0.60$, $d = -6.90$ and $d' = -7.90$, and (d) is a soft deadline (mixed case) with $\gamma = 0.60$, $d = -6.50$, $d' = -7.50$ and $d'' = -10.50$. The optimal plan for case (d) shows that different actions can be optimal in the same configuration of blocks depending on the current wealth level (= time units spent). It includes only conditions that are reachable from the start configuration of blocks with zero wealth level under the optimal plan. Figure 5 shows the corresponding optimal value functions $V^*_U(s)$ for case (a), and Figure 6 shows the corresponding optimal value functions $V^*_U(s)$ for case (b).

Related Work

Most research is on decision-theoretic planners that maximize the expected total reward, which is equivalent to maximizing the expected total utility for linear utility functions.

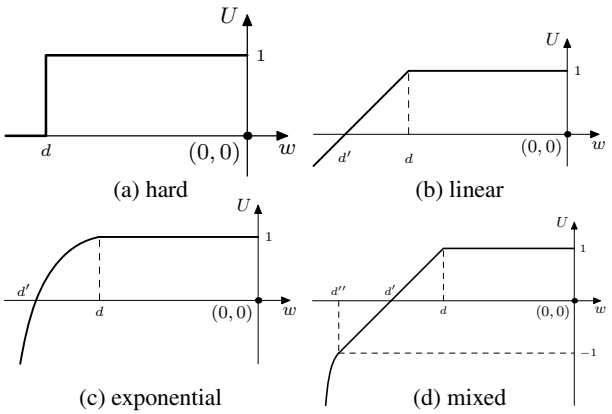


Figure 4: Different deadline utility functions

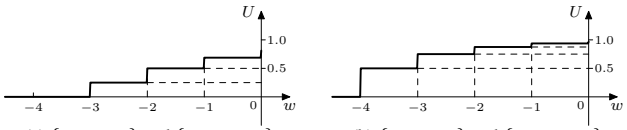


Figure 5: Optimal value functions for hard deadline ($d = -5$)

However, some decision-theoretic planners maximize the expected total utility: First, Loui (1983) discussed methods for maximizing the expected total utility but only for MDPs with deterministic actions (whose rewards are stochastic). Murthy & Sarkar (1996, 1997, 1998) discussed even more efficient methods for special cases of this scenario. Second, Koenig & Simmons (1994), Koenig & Liu (1999), Denardo & Rothblum (1979) and Patek (2001) discussed methods for maximizing the expected total utility but only for exponential utility functions. Third, Yu, Lin, & Yan (1998) discussed methods for maximizing the expected total utility but only for utility functions that are step functions. They also transformed a given MDP into an augmented MDP but their aug-

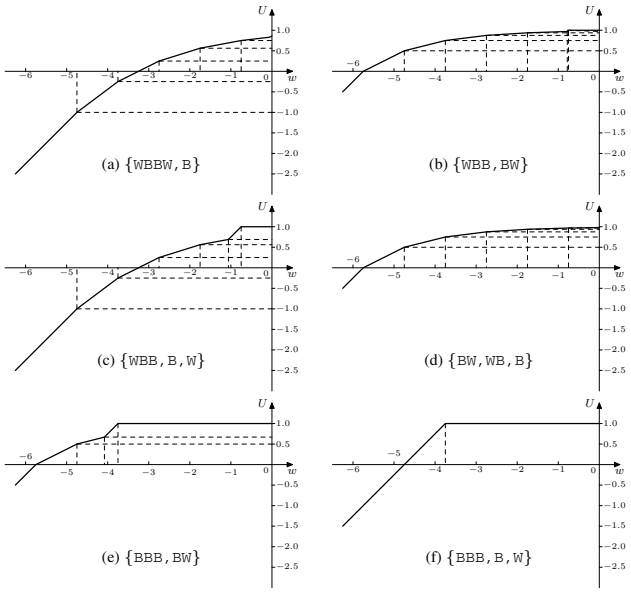


Figure 6: Optimal value functions for soft deadline (linear case)

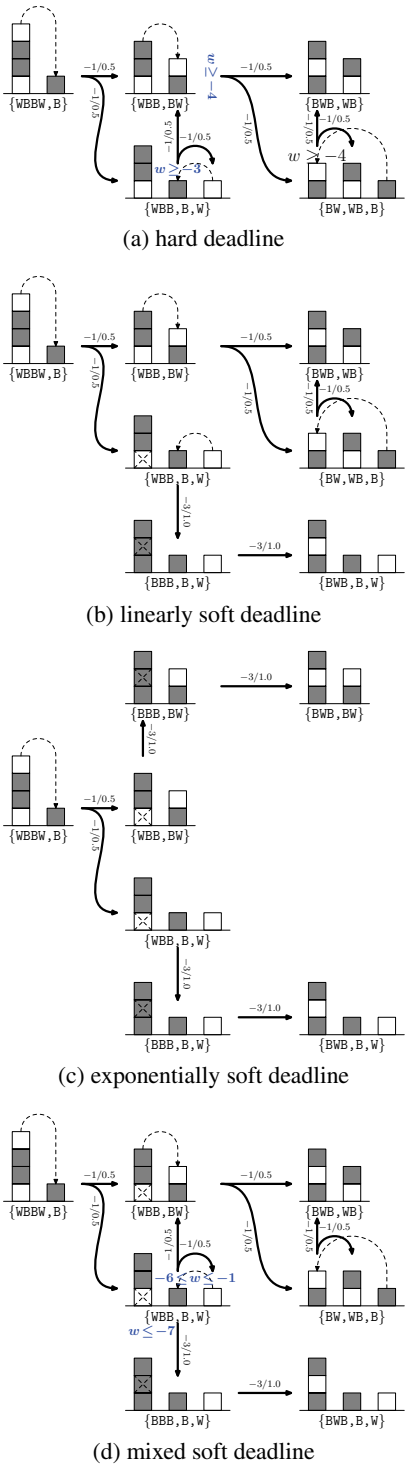


Figure 7: Optimal plans

mented MDP is different from ours. Fourth, White (1987) and Kerr (1999) discussed methods for maximizing the expected total utility for arbitrary utility functions but only for finite time horizons. Finally, Dolgov & Durfee (2004) discussed approximation methods for evaluating (rather than maximizing) the expected total utility of a given stationary

policy using Legendre polynomials.

Some decision-theoretic planners use piecewise approximations of the value functions but so far only for maximizing the expected total reward: First, Li & Littman (2005) discussed methods that use piecewise constant functions to approximate the value functions of finite horizon continuous state MDPs. Second, both Poupart *et al.* (2002) and Pynadath & Marsella (2004) discussed methods that use piecewise linear functions to approximate the value functions of factored MDPs but their piecewise linear functions are decision trees with linear functions at the leaves.

Conclusions and Future Work

Functional value iteration is a powerful tool for finding plans that maximize the expected total utility for MDPs. A variety of decision-theoretic planners use (regular) value iteration in combination with other techniques to maximize the expected total reward for MDPs efficiently. Examples include LAO* (Hansen & Zilberstein, 2001) and SPUDD (Hoey *et al.*, 1999). Functional value iteration, together with the approximation techniques presented in this paper, can replace (regular) value iteration in these decision-theoretic planners to create versions of them that efficiently maximize the expected total utility for a large class of utility functions. We are currently in the process of implementing such decision-theoretic planners.

Acknowledgments

We thank Craig Tovey and Anton Kleywegt, two operations researchers, for lots of advice. Without their expertise, it would have been impossible to write this paper. 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.

References

Blythe, J. 1998. *Planning under Uncertainty in Dynamic Domains*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University.

Blythe, J. 1999. Decision-theoretic planning. *AI Magazine* 20(2):37–54.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Cohen, P. R.; Greenberg, M. L.; Hart, D. M.; and Howe, A. E. 1989. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine* 10(3):32–48.

Denardo, E. V., and Rothblum, U. G. 1979. Optimal stopping, exponential utility, and linear programming. *Mathematical Programming* 16:228–244.

Dolgov, D., and Durfee, E. 2004. Approximate probabilistic constraints and risk-sensitive optimization criteria in Markov decision processes. In *Proceedings of the Eighth International Symposium on Artificial Intelligence and Mathematics*.

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.

Haddawy, P., and Hanks, S. 1992. Representations for decision-theoretic planning: Utility functions for deadline goals. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR-92)*, 71–82.

Hansen, E. A., and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence* 129:35–62.

Hoey, J.; St. Aubin, R.; Hu, A. J.; and Boutilier, C. 1999. SPUDD: Stochastic planning using decision diagrams. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, 279–288.

Kerr, A. L. 1999. Utility maximising stochastic dynamic programming: An overview. In *Proceedings of the 34th Annual Conference of the Operational Research Society of New Zealand (ORSNZ-99), December 10-11, Hamilton, New Zealand*.

Koenig, S., and Liu, Y. 1999. Sensor planning with non-linear utility functions. In *Proceedings of the Fifth European Conference on Planning (ECP-99)*, 265–277.

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.

Li, L., and Littman, M. L. 2005. Lazy approximation for solving continuous finite-horizon MDPs. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, 1175–1180.

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.

Liu, Y. 2005. *Decision Theoretic Planning under Risk-Sensitive Planning Objectives*. Ph.D. Dissertation, College of Computing, Georgia Institute of Technology.

Loui, R. P. 1983. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM* 26(9):670–676.

Murthy, I., and Sarkar, S. 1996. A relaxation-based pruning technique for a class of stochastic shortest path problems. *Transportation Science* 30(3):220–236.

Murthy, I., and Sarkar, S. 1997. Exact algorithms for the stochastic shortest path problem with a decreasing deadline utility function. *European Journal of Operational Research* 103:209–229.

Murthy, I., and Sarkar, S. 1998. Stochastic shortest path problems with piecewise-linear concave utility functions. *Management Science* 44(11):S125–S136.

Murthy, S.; Akkiraju, R.; Goodwin, R.; Keskinocak, P.; Rachlin, J.; Wu, F.; Kumaran, S.; Yeh, J.; Fuhrer, R.; Aggarwal, A.; Sturzenbecker, M.; Jayaraman, R.; and Daigle, B. 1999. Cooperative multi-objective decision-support for the paper industry. *Interface* 29(5):5–30.

Patek, S. D. 2001. On terminating Markov decision processes with a risk averse objective function. *Automatica* 37(9):1379–1386.

Pell, B.; Bernard, D.; Chien, S.; Gat, E.; Muscettola, N.; Nayak, P. P.; Wagner, M.; and Williams, B. 1998. An autonomous spacecraft agent prototype. *Autonomous Robotics* 5:1–27.

Poupart, P.; Boutilier, C.; Schuurmans, D.; and Patrascu, R. 2002. Piecewise linear value function approximation for factored MDPs. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 292–299.

Pratt, J. W. 1964. Risk aversion in the small and in the large. *Econometrica* 32(1-2):122–136.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Pynadath, D. V., and Marsella, S. C. 2004. Fitting and compilation of multiagent models through piecewise linear functions. In *Proceedings of the Third International Joint Conference on Autonomuous Agents and Multiagent Systems (AAMAS-04)*, 1197–1204.

Rote, G. 1992. The convergence rate of the sandwich algorithm for approximating convex functions. *Computing* 48:337–361.

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.

Yu, S. X.; Lin, Y.; and Yan, P. 1998. Optimization models for the first arrival target distribution function in discrete time. *Journal of Mathematical Analysis and Applications* 225:193–223.

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.