Risk-Sensitive Planning with One-Switch Utility Functions: Value Iteration

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

Decision-theoretic planning with nonlinear utility functions is important since decision makers are often risk-sensitive in high-stake planning situations. One-switch utility functions are an important class of nonlinear utility functions that can model decision makers whose decisions change with their wealth level. We study how to maximize the expected utility of a Markov decision problem for a given one-switch utility function, which is difficult since the resulting planning problem is not decomposable. We first study an approach that augments the states of the Markov decision problem with the wealth level. The properties of the resulting infinite Markov decision problem then allow us to generalize the standard risk-neutral version of value iteration from manipulating values to manipulating functions that map wealth levels to values. We use a probabilistic blocks-world example to demonstrate that the resulting risk-sensitive version of value iteration is practical.

Introduction

Utility theory (von Neumann & Morgenstern, 1944) is a normative theory of decision making under uncertainty. It states that every rational decision maker who accepts a small number of axioms has a strictly monotonically increasing utility function that transforms their wealth level w into a utility $U(w)$ so that they always choose the course of action that maximizes their expected utility. The utility function models their risk attitudes. A decision maker is risk-neutral if their utility function is linear, risk-averse if their utility function is concave, and risk-seeking if their utility function is convex. Decision-theoretic planning with nonlinear utility functions is important since decision makers are often risk-sensitive in high-stake planning situations (= planning situations with the possibility of large wins or losses) and their risk attitude affects their decisions. For example, some decision makers buy insurance in business decision situations and some do not. Furthermore, their decisions often change with their wealth level. In particular, they 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 such decision makers. We model probabilistic planning problems as fully observable Goal-Directed Markov Decision Problems (GDMDPs) and investigate how to maximize their expected utility for a given one-switch utility function, which is difficult since the resulting planning problem is not decomposable. The optimal course of action now depends not only on the current state of the GDMDP but also the wealth level (= accumulated rewards). Thus, we first study an approach that transforms a risk-sensitive GDMDP into a risk-neutral one, basically by augmenting the states of the risk-sensitive GDMDP with the possible wealth levels. The resulting risk-neutral GDMDP has an infinite state space but its properties allow us to generalize the standard risk-neutral version of value iteration, which manipulates values (one for each state), to a risk-sensitive version of value iteration, which manipulates functions (one for each state) that map wealth levels to values. We use a probabilistic blocks-world example to demonstrate that the resulting risk-sensitive version of value iteration is practical. Our research is intended to be a first step toward better probabilistic planners for high-stake planning situations such as environmental crisis situations (Blythe, 1997), business decisions situations (Goodwin, Akkiraju, & Wu, 2002), and planning situations in space (Zilberstein *et al.*, 2002).

GDMDPs

We model probabilistic planning problems as finite Goal-Directed Markov Decision Problems (GDMDPs), which are characterized by a finite set of states S, a finite set of goal states $G \subseteq S$, and a finite set of actions A that can be executed in all non-goal states $s \in S \setminus G$. The decision maker always chooses which action $a \in A$ to execute in their current non-goal state $s \in S \setminus G$. Its execution results with probability $P(s'|s,a)$ in finite (immediate) reward $r(s,a,s') < 0$ and a transition to state $s' \in S$ in the next time step. The decision maker stops acting when they reach a goal state $s \in G$, which is modeled as them executing a dummy action whose execution results with probability 1.0 in reward 0.0 and leaves their current goal state unchanged. H_t denotes the set of all histories at time step $t \geq 0$. A history at time step t is any sequence $h_t = (s_0, a_0, \cdots, s_{t-1}, a_{t-1}, s_t) \in (S \times A)^t \times S$ of states and actions from the state at time step 0 to the current state at time step t that can occur with positive probability if the decision maker executes the corresponsing actions in sequence. The (planning) horizon of a

decision maker is the number of time steps $1 \leq T \leq \infty$ that they plan for. A trajectory is an element of H_T.

Decision-theoretic planners determine policies, where a policy π consists of a decision rule d_t for every time step $0 \leq t < T$ within the horizon. A decision rule determines which action the decision maker should execute in their current state. The most general policies are those that consist of potentially different decision rules for the time steps, where every decision rule is a mapping from histories at the current time step to probability distributions over actions, called randomized history-dependent (HR) decision rules. We denote the class of such policies as Π^{HR}. More restricted policies consist of the same decision rule for every time step, where the decision rule is a mapping from only the current state to actions, called deterministic stationary (SD) decision rules. We denote the class of such policies as Π^{SD}.

Consider a decision maker with an arbitrary utility function U. If the horizon T is finite and the decision maker starts in initial state $s \in S$, then the expected utility of their total reward under policy $\pi \in \Pi^{\text{HR}}$ is

$$v^{\pi}_{U,T}(s) = E^{s,\pi}\left[U\left(\sum_{t=0}^{T-1} r(s_t, a_t, s_{t+1})\right)\right],$$

where the expectation is taken over all possible trajectories. The expected utilities exist and are finite because the number of trajectories is finite. We refer to the value $w_t = \sum_{i=0}^{t-1} r(s_i, a_i, s_{i+1})$ as their wealth level at time step t. (Note that $w_0 = 0$.) If the horizon is infinite and the decision maker starts in initial state $s \in S$, then the expected utility of their total reward under policy $\pi \in \Pi^{\text{HR}}$ is

$$v^{\pi}_{U}(s) = \lim_{T\to\infty} v^{\pi}_{U,T}(s) = \lim_{T\to\infty} E^{s,\pi}\left[U\left(\sum_{t=0}^{T-1} r(s_t, a_t, s_{t+1})\right)\right].$$

The expected utilities exist since the values $v^{\pi}_{U,T}(s)$ exist and are decreasing in T. However, some or all expected utilities can be minus infinity. The maximal expected utilities of the total reward are $v^{*}_{U}(s) = \sup_{\pi\in\Pi^{\text{HR}}} v^{\pi}_{U}(s)$. They exist because the expected utilities exist under all policies. However, some or all maximal expected utilities can be minus infinity. To simplify our terminology, we refer to the expected utility $v^{\pi}_{U}(s)$ as the risk-sensitive value of state s under policy π and to the maximal expected utility $v^{*}_{U}(s)$ as the optimal risk-sensitive value of state s under policy π. A risk-sensitive optimal policy $\pi \in \Pi^{\text{HR}}$ is one for which $v^{\pi}_{U}(s) = v^{*}_{U}(s)$ for all states $s \in S$. It is important for all risk-sensitive values to be finite because the definition of a risk-sensitive optimal policy can otherwise be inconsistent with commonsense (Liu, 2005).

Transforming the GDMDP

A probabilistic planning problem is decomposable under utility function U if there exists a function f_U such that, for all random variables x (here: representing the reward at the current time step) and random variables y that depend on x (here: representing the total reward from the next time step on), it holds that

$$E[U(x+y)] = E\Big[f_U\big(x, E[U(y)]\big)\Big].$$

The expected utility of the total reward can then be obtained by combining the reward at the current time step with the expected utility of the total reward from the next time step on. Probabilistic planning problems under linear and exponential utility functions are decomposable. The optimal values and policies for GDMDPs under such utility functions can be determined with dynamic programming algorithms, such as value iteration (Bertsekas & Tsitsiklis, 1991; Patek, 2001), that manipulate one value for each state. Probabilistic planning problems under other utility functions are not decomposable. The optimal values and policies for GDMDPs under such utility functions thus cannot be determined with dynamic programming algorithms that manipulate one value for each state. Instead, we transform the problem of finding an optimal risk-sensitive policy for the original GDMDP into the problem of finding an optimal risk-neutral policy for an augmented GDMDP, where the augmented GDMDP is obtained by augmenting the states of the original GDMDP with the possible wealth levels. An optimal policy for the augmented GDMDP can then, in principle, be obtained with existing dynamic programming algorithms.

We first define the possible wealth levels W used in the construction of the augmented GDMDP. Let $R = \{0\} \cup \{r(s,a,s') \,|\, s, s' \in S, a \in A \text{ with } P(s'|s,a) > 0\}$. Let $W^0 = \{0\}$ and $W^{t+1} = \{r + w \,|\, r \in R, w \in W^t\}$ for all time steps $0 \leq t < T$. The finite set W^t includes the possible wealth levels at time step $t \geq 0$, and the countably infinite set $W = \bigcup_{t=0}^{\infty} W^t$ includes the possible wealth levels at any time step within the horizon.

We now define the augmented GDMDP for a given GDMDP and utility function U. We distinguish it from the original GDMDP by enclosing all of its elements in angular brackets. The augmented GDMDP 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 $\langle A\rangle = A$. (We continue to use the notation $a \in A$ instead of $\langle a\rangle \in \langle A\rangle$ for the augmented GDMDP.) The decision maker always chooses which action $a \in A$ to execute in their current non-goal state $\langle s\rangle \in \langle S\rangle \setminus \langle G\rangle$. Assume that $\langle s\rangle = (s, w)$ and $\langle s\rangle' = (s', w')$. The execution of the action then results with probability

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

in finite reward $\langle r_{|U}\rangle(\langle s\rangle, a, \langle s\rangle') = U(w') - U(w) < 0$ and a transition to state $\langle s\rangle' \in \langle S\rangle$ in the next time step. The decision maker stops acting when they reach a goal state $\langle s\rangle \in \langle G\rangle$.

We first relate the histories of the original and augmented GDMDP. A history $h_t = (s_0, a_0, s_1, \ldots, s_t) \in H_t$ in the original GDMDP corresponds to a set of histories $\langle h\rangle_t = \big((s_0, w + w_0), a_0, (s_1, w + w_1), \ldots, (s_t, w + w_t)\big) \in \langle H\rangle_t$ in the augmented GDMDP, where the initial wealth level $w \in W$ is arbitrary and $w_t = \sum_{i=0}^{t-1} r(s_i, a_i, s_{i+1})$ for all time steps t. Let ϕ_w be the mapping from histories of the original GDMDP to the corresponding histories of the augmented GDMDP with initial wealth level w. Also, let ψ be the mapping from histories of the aug-

mented GDMDP to the corresponding histories of the original GDMDP. Then, for all histories $h \in H_t$ of the original GDMDP and all wealth levels $w \in W$, it holds that $h = \psi(\phi_w(h))$. Also, for all histories $\langle h \rangle \in \langle H \rangle_t$ of the augmented GDMDP, there exists a wealth level $w \in W$ such that $\langle h \rangle = \phi_w\big(\psi(\langle h \rangle)\big)$. This correspondence in histories between the original and augmented GDMDP implies a correspondence in policies. Let Ψ_w be the mapping from policies $\pi = (d_0, d_1, \ldots, d_{T-1}) \in \Pi^{\mathrm{HR}}$ in the original GDMDP to policies $\Psi(\pi) = (\langle d \rangle_0, \langle d \rangle_1, \ldots, \langle d \rangle_{T-1}) \in \langle \Pi \rangle^{\mathrm{HR}}$ in the augmented GDMDP such that $\langle d \rangle_t(\langle h \rangle) = d_t\big(\psi(\langle h \rangle)\big)$ for all histories $\langle h \rangle \in \langle H \rangle_t$ of the augmented GDMDP. Also, let Φ_w be the mapping from policies $\langle \pi \rangle = (\langle d \rangle_0, \langle d \rangle_1, \ldots, \langle d \rangle_{T-1}) \in \langle \Pi \rangle^{\mathrm{HR}}$ in the augmented GDMDP to policies $\Phi_w(\langle \pi \rangle) = (d_0, d_1, \ldots, d_{T-1}) \in \Pi^{\mathrm{HR}}$ in the original GDMDP such that $d_t(h) = \langle d \rangle_t\big(\phi_w(h)\big)$ for all histories $h \in H_t$ in the original GDMDP. Then, for all wealth levels $w \in W$, it holds that $\Phi_w(\Psi(\pi)) = \pi$. However, it is not guaranteed that there exists a wealth level $w \in W$ such that $\Psi(\Phi_w(\langle \pi \rangle)) = \langle \pi \rangle$.

We now relate the values and policies of the original and augmented GDMDP. The following notation is helpful in this context: We use $v_U^\pi(s)$ to denote the risk-sensitive value of state s in the original GDMDP under policy π and for utility function U. We use $\langle v|_U \rangle^{\langle \pi \rangle}(\langle s \rangle) = \langle v|_U \rangle^{\langle \pi \rangle}(s, w)$ to denote the risk-neutral value of state $\langle s \rangle = (s, w)$ in the augmented GDMDP under policy $\langle \pi \rangle$, that is, for the identity function as utility function. We also introduce a means for transforming utility functions: We can consider the utility function $U(w + w')$ to be a utility function of wealth level w' obtained by shifting the utility function $U(w')$ to the left by wealth level w. $U \lll w$ denotes this utility function. It thus holds, for example, that $v^\pi_{U \lll w, T}(s) = E^{s,\pi}[U(w + w_T)]$.

Theorem 1. *Consider a policy $\pi \in \Pi^{\mathrm{HR}}$ for the original GDMDP. The risk-neutral values $\langle v|_U \rangle^{\Psi(\pi)}(s, w)$ exist for the augmented GDMDP and it holds that $\langle v|_U \rangle^{\Psi(\pi)}(s, w) = v^\pi_{U \lll w}(s) - U(w)$ for all states $s \in S$ and wealth levels $w \in W$.*

Proof. The risk-neutral values $v^\pi_{U \lll w}(s)$ exist for all states $s \in S$ and wealth levels $w \in W$ since the values of GDMDP states exist under all policies and utility functions. If $\langle s \rangle_t = (s_t, w + w_t)$ for all time steps $t \geq 0$, then it holds for all finite horizons T that

$$
\begin{aligned}
\langle v|_U \rangle_T^{\Psi(\pi)}(\langle s \rangle) &= E^{\langle s \rangle, \Psi(\pi)}\left[\sum_{t=0}^{T-1} \langle r|_U \rangle(\langle s \rangle_t, a_t, \langle s \rangle_{t+1})\right] \\
&= E^{\langle s \rangle, \Psi(\pi)}\left[\sum_{t=0}^{T-1} \big(U(w + w_{t+1}) - U(w + w_t)\big)\right] \\
&= E^{\langle s \rangle, \Psi(\pi)}\left[\sum_{t=1}^{T} U(w + w_t) - \sum_{t=0}^{T-1} U(w + w_t)\right] \\
&= E^{\langle s \rangle, \Psi(\pi)}\big[U(w + w_T) - U(w + w_0)\big] \\
&= E^{\langle s \rangle, \Psi(\pi)}\big[U(w + w_T) - U(w)\big] \\
&= E^{s,\pi}[U(w + w_T)] - U(w) \\
&= v^\pi_{U \lll w, T}(s) - U(w),
\end{aligned}
$$

where the second-to-last equality is due to the fact that policy π in the original GDMDP with initial state s and policy $\Psi(\pi)$ in the augmented GDMDP with initial state $\langle s \rangle = (s, w)$ induce the same random process of original states and actions. The theorem then follows by letting the horizon T approach infinity. □

The theorem implies that $\langle v|_U \rangle^{\Psi(\pi)}(s, 0) = v_U^\pi(s) - U(0)$ for all states $s \in S$.

Theorem 2. *Consider a policy $\langle \pi \rangle \in \langle \Pi \rangle^{\mathrm{HR}}$ for the augmented GDMDP. The risk-sensitive values $v^{\Phi_w(\langle \pi \rangle)}_{U \lll w}(s)$ exist for the original GDMDP and it holds that $v^{\Phi_w(\langle \pi \rangle)}_{U \lll w}(s) = \langle v|_U \rangle^{\langle \pi \rangle}(s, w) + U(w)$ for all states $s \in S$ and wealth levels $w \in W$.*

Proof. The risk-sensitive values $\langle v|_U \rangle^{\langle \pi \rangle}(s, w)$ exist for all states $s \in S$ and wealth levels $w \in W$ since the values of GDMDP states exist under all policies and utility functions. If $\langle s \rangle_t = (s_t, w + w_t)$ for all time steps $t \geq 0$, then it holds for all finite horizons T that

$$
\begin{aligned}
\langle v|_U \rangle_T^{\langle \pi \rangle}(s, w) &= E^{(s,w), \langle \pi \rangle}\left[\sum_{t=0}^{T-1} \langle r|_U \rangle(\langle s \rangle_t, a_t, \langle s \rangle_{t+1})\right] \\
&= E^{(s,w), \langle \pi \rangle}\left[\sum_{t=0}^{T-1} \big(U(w + w_{t+1}) - U(w + w_t)\big)\right] \\
&= E^{(s,w), \langle \pi \rangle}\left[\sum_{t=1}^{T} U(w + w_t) - \sum_{t=0}^{T-1} U(w + w_t)\right] \\
&= E^{(s,w), \langle \pi \rangle}\big[U(w + w_T) - U(w + w_0)\big] \\
&= E^{(s,w), \langle \pi \rangle}\big[U(w + w_T) - U(w)\big] \\
&= E^{s, \Phi_w(\langle \pi \rangle)}[U(w + w_T)] - U(w) \\
&= v^{\Phi_w(\langle \pi \rangle)}_{U \lll w, T}(s) - U(w),
\end{aligned}
$$

where the second-to-last equality is due to the fact that policy $\langle \pi \rangle$ in the augmented GDMDP with initial state $\langle s \rangle = (s, w)$ and policy $\Phi_w(\langle \pi \rangle)$ in the original GDMDP with initial state s induce the same random process of original states and actions. The theorem then follows by letting the horizon T approach infinity. □

The theorem implies $v_U^{\Phi_0(\langle \pi \rangle)}(s) = \langle v|_U \rangle^{\langle \pi \rangle}(s, 0) + U(0)$ for all states $s \in S$. Consequently, we can obtain an optimal risk-sensitive policy for the original GDMDP by obtaining an optimal risk-neutral policy for the augmented GDMDP, as stated in the following theorem.

Theorem 3. *There exists an optimal risk-neutral policy in $\langle \Pi \rangle^{SD}$ for the augmented GDMDP. If policy $\langle \pi \rangle$ is an optimal risk-neutral policy for the augmented GDMDP then policy $\Phi_0(\langle \pi \rangle)$ is an optimal risk-sensitive policy for the original GDMDP.*

Proof. There exists an optimal risk-neutral policy in $\langle \Pi \rangle^{\mathrm{SD}}$ for the augmented GDMDP according to Puterman (1994). Now assume that there exists a risk-sensitive policy π and a state s for the original GDMDP such that $v_U^\pi(s) >$

$v_U^{\Phi_0(\langle\pi\rangle)}(s)$. Then,

$$\begin{aligned}\langle v|_U\rangle^{\Psi(\pi)}(s,0) &= v_U^{\pi}(s) - U(0)\\ &> v_U^{\Phi_0(\langle\pi\rangle)}(s) - U(0) = \langle v|_U\rangle^{\langle\pi\rangle}(s,0)\end{aligned}$$

according to Theorem 1 and Theorem 2, which contradicts the fact that $\langle\pi\rangle$ is an optimal risk-neutral policy for the augmented GDMDP. □

It is known that there does not necessarily exist an optimal risk-sensitive policy in Π^{SD} for the original GDMDP (White, 1987). However, since there exists an optimal risk-neutral policy in $\langle\Pi\rangle^{\text{SD}}$ for the augmented GDMDP, there exists an optimal risk-sensitive policy for the original GDMDP that consists of the same decision rule for every time step, where the decision rule is a mapping from the current state and wealth level to actions. We refer to such a policy as augmented SD-optimal.

Functional Value Iteration

The augmented GDMDP has a countably infinite set of states and a finite set of actions. In principle, value iteration can be used to find an optimal risk-neutral policy for the augmented GDMDP (Puterman, 1994). Its update equations are for all states $s \in S$, wealth levels $w \in W$, and time steps $t \geq 0$:

$$\begin{aligned}\langle v|_U\rangle^0(s,w) &= 0, && s \in S,\\ \langle v|_U\rangle^{t+1}(s,w) &= \max_{a\in A}\sum_{s'\in S}P(s'|s,a)\Big[U(w+r(s,a,s'))\\ &\quad - U(w) + \langle v|_U\rangle^t(s', w+r(s,a,s'))\Big], && s \notin G.\end{aligned}$$

The values $\langle v|_U\rangle^t(s,w)$ converge to the optimal risk-neutral values $\langle v|_U\rangle^*(s,w)$ as the number of time steps t approaches infinity. An optimal risk-neutral policy in $\langle\Pi\rangle^{\text{SD}}$ for the augmented GDMDP then is to execute action $\arg\max_{a\in A}\sum_{s'\in S}P(s'|s,a)[U(w+r(s,a,s'))-U(w)+\langle v|_U\rangle^*(s', w+r(s,a,s'))]$ in non-goal state $(s,w) \in \langle S\rangle \setminus \langle G\rangle$. Applying Φ_0 to this policy results in an augmented SD-optimal risk-sensitive policy for the original GDMDP.

Value iteration updates the values of augmented states with the same original state and different wealth levels in a similar way since they have the same transition probabilities. We exploit this similarity by defining the functional value functions $V_U : S \mapsto (W \mapsto \mathbb{R})$ that map original states to functions, namely functions from wealth levels to values. We define $V_U^t(s)(w) = \langle v|_U\rangle^t(s,w) + U(w)$ and $V_U^*(s)(w) = \langle v|_U\rangle^*(s,w)+U(w)$ for all states $s \in S$, wealth levels $w \in W$ and time steps $t \geq 0$. We then re-write the above version of value iteration for the augmented GDMDP so that it looks like value iteration for the original GDMDP except that it uses functional value functions for the original states instead of values. We refer to this version of value iteration as functional value iteration. Its update equations are for all states $s \in S$, wealth levels $w \in W$, and time steps $t \geq 0$:

$$\begin{aligned}V_U^0(s)(w) &= \langle v|_U\rangle^0(s,w) + U(w) = U(w), && s \in S\\ V_U^{t+1}(s)(w) &= \langle v|_U\rangle^{t+1}(s,w) + U(w)\end{aligned}$$

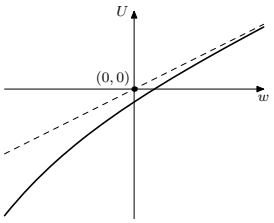


Figure 1: One-Switch Utility Function

$$\begin{aligned}&= \max_{a\in A}\sum_{s'\in S}P(s'|s,a)\big[U(w+r(s,a,s')) - U(w)\\ &\quad + \langle v|_U\rangle^t\big(s', w+r(s,a,s')\big)\big] + U(w)\\ &= \max_{a\in A}\sum_{s'\in S}P(s'|s,a)\big[U(w+r(s,a,s'))\\ &\quad + \langle v|_U\rangle^t\big(s', w+r(s,a,s')\big)\big]\\ &= \max_{a\in A}\sum_{s'\in S}P(s'|s,a)\cdot V_U^t(s')(w+r(s,a,s'))\\ &= \max_{a\in A}\sum_{s'\in S}P(s'|s,a)\cdot V^t_{U\ll r(s,a,s')}(s')(w), \quad s \notin G.\end{aligned}$$

The values $V_U^t(s)(w)$ converge to the values $V_U^*(s)(w)$ as the number of time steps t approaches infinity. An optimal risk-neutral policy in $\langle\Pi\rangle^{\text{SD}}$ for the augmented GDMDP then is to execute action $\arg\max_{a\in A}\sum_{s'\in S}P(s'|s,a)V^*_{U\ll r(s,a,s')}(s')(w)$ in non-goal state $(s,w) \in \langle S\rangle \setminus \langle G\rangle$. Again, applying Φ_0 to this policy results in an augmented SD-optimal risk-sensitive policy for the original GDMDP.

One needs to perform three operations on functional value functions to carry out functional value iteration. To calculate the expression $\max_{a\in A}\sum_{s'\in S}P(s'|s,a)\cdot V^t_{U\ll r(s,a,s')}(s')(\cdot)$, one needs to shift a functional value function to the left by reward $r(s,a,s')$, calculate the probabilistically weighted average of several functional value functions and calculate the maximum of several functional value functions. We need to restrict the class of utility functions to be able to perform these operations efficiently.

One-Switch Utility Functions

Most research on decision-theoretic planning in artificial intelligence has used linear utility functions and, to a much lesser degree, exponential utility functions (Koenig & Simmons, 1994; Koenig & Liu, 1999). However, linear and exponential utility functions have the property that the decisions of a decision maker depend only on the increase in wealth level but not the wealth level itself. This property is unrealistic since 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 such decision makers. Their name is due to the property that a decision maker with a one-switch utility function who is confronted with two courses of actions switches at most once from one course of action to another course of action as their wealth level increases. This property was proposed in (Bell,

1988) and studied in detail in (Bell, 1988; Nakamura, 1996; Gelles & Mitchell, 1999; Bell & Fishburn, 2001). The only one-switch utility functions that model decision makers who are risk averse but become risk-neutral in the limit as their wealth level increases have the form $U(w) = Cw - D\gamma^w$ for $C, D > 0$ and $0 < \gamma < 1$ (Bell, 1988). In the following, we assume that the utility function U has this particular form, which we refer to as a one-switch utility function. Figure 1 shows an example.

Probabilistic planning problems with one-switch utility functions are not decomposable but can be solved with functional value iteration. The functional value functions $V_U^t(s)(w)$ of functional value iteration are then piecewise one-switch value functions, that is, consist of segments of one-switch utility functions of the form $Cw - d\gamma^w - e$, where $C > 0$ and $0 < \gamma < 1$ are parameters of the one-switch utility function U and $d > 0$ and $e \geq 0$ are parameters of the segment. We represent piecewise one-switch value functions with ℓ segments as lists of triples $(w^1, d^1, e^1), (w^2, d^2, e^2), \ldots, (w^\ell, d^\ell, e^\ell)$, where $-\infty = w^0 < w^1 < w^2 < \cdots < w^\ell = 0$. We refer to the values w^i as breakpoints of the piecewise one-switch value function. The value of the piecewise one-switch value function is $Cw - d^i\gamma^w - e^i$ if $w^{i-1} < w \leq w^i$ for some $1 \leq i \leq \ell$. We need to show how to shift a piecewise one-switch value function to the left, calculate the probabilistically weighted average of several piecewise one-switch value functions and calculate the maximum of several piecewise one-switch value functions. We show in the remainder of this section that this can be done in a finite amount of time and with a finite amount of memory despite the number of augmented states being infinite.

If we have a piecewise one-switch value function V that is represented as $(w^1, d^1, e^1), (w^2, d^2, e^2), \ldots, (w^\ell, d^\ell, e^\ell)$, then shifting it to the left by reward r results also in a piecewise one-switch value function, which can be calculated in two steps: The first step is to calculate the new piecewise one-switch value function. Note that

$$\begin{aligned} V(w + r) &= C(w + r) - d^i\gamma^{w+r} - e^i \\ &= Cw - d^i\gamma^r\gamma^w - (e^i - Cr) \end{aligned}$$

for $w^{i-1} < w+r \leq w^i$ or, equivalently, $w^{i-1}-r < w \leq w^i-r$. The new piecewise one-switch value function $V \lll r$ can therefore be represented as $(w^1 - r, d^1\gamma^r, e^1 - Cr), (w^2 - r, d^2\gamma^r, e^2 - Cr), \ldots, (w^\ell - r, d^\ell\gamma^r, e^\ell - Cr)$. The second step is to simplify the new piecewise one-switch value function. The piecewise one-switch value function was actually shifted to the right since the reward r is negative. All wealth levels are non-positive and we can thus speed up subsequent operations on the new piecewise one-switch value function by keeping only the part to the left of $w = 0$.

If we have several piecewise one-switch value functions then their probabilistically weighted average is also a piecewise one-switch value function, which can be calculated in two steps: The first step is to introduce additional breakpoints to the representations of each piecewise one-switch value function (without changing the functions themselves), if needed, to make all of them have the same breakpoints. The second step is to calculate the parameters of each segment of the new piecewise one-switch value function from the corresponding segments of the given piecewise one-switch value functions. For example, if we have two piecewise one-switch value functions V' and V'' that are represented as $(w^1, d^1, e^1), (w^2, d^2, e^2), \ldots, (w^\ell, d^\ell, e^\ell)$ and $(w^1, \hat{d}^1, \hat{e}^1), (w^2, \hat{d}^2, \hat{e}^2), \ldots, (w^\ell, \hat{d}^\ell, \hat{e}^\ell)$, then their probabilistically weighted average $pV' + qV''$ for $p+q = 1$ can be represented as $(w^1, pd^1 + q\hat{d}^1, pe^1 + q\hat{e}^1), (w^2, pd^2 + q\hat{d}^2, pe^2 + q\hat{e}^2), \ldots, (w^\ell, pd^\ell + q\hat{d}^\ell, pe^\ell + q\hat{e}^\ell)$.

If we have several piecewise one-switch value functions then their maximum is also a piecewise one-switch value function, which can be calculated in three steps: The first step is identical to the step given above for calculating the probabilistically weighted average. The second step is more complicated since we might have to introduce additional breakpoints. For example, assume that we have two piecewise one-switch value functions V' and V'' that are represented as $(w^1, d^1, e^1), (w^2, d^2, e^2), \ldots, (w^\ell, d^\ell, e^\ell)$ and $(w^1, \hat{d}^1, \hat{e}^1), (w^2, \hat{d}^2, \hat{e}^2), \ldots, (w^\ell, \hat{d}^\ell, \hat{e}^\ell)$. Consider the segment of the new piecewise one-switch value function over the interval $w^{i-1} < w \leq w^i$. There are three cases: In the first case, the first piecewise one-switch value function dominates the other one over the whole segment ($V'(w^{i-1}) \geq V''(w^{i-1})$ and $V'(w^i) \geq V''(w^i)$) and the maximum of the two piecewise one-switch value functions over the segment is thus the first one. In the second case, the second piecewise one-switch value function dominates the other one over the whole segment and the maximum of the two piecewise one-switch value functions over the segment is thus the second one. In the third case, neither piecewise one-switch value function dominates the other one over the whole segment and their two segments thus intersect. There is only one intersection, and the intersection point w satisfies

$$\begin{aligned} V'(w) &= V''(w) \\ Cw - d^i\gamma^w - e^i &= Cw - \hat{d}^i\gamma^w - \hat{e}^i \\ \gamma^w &= \frac{\hat{e}^i - e^i}{d^i - \hat{d}^i} \end{aligned}$$

and thus

$$w = \log_\gamma \frac{\hat{e}^i - e^i}{d^i - \hat{d}^i}$$

with $w^{i-1} < w < w^i$. We add the intersection point w as a breakpoint. The maximum of the two piecewise one-switch value functions is just the first one on one side of the breakpoint and the second one on the other side of the breakpoint. The third step is to merge adjacent segments of the new piecewise one-switch value function, if possible, to remove unnecessary breakpoints and speed up subsequent operations on it.

Finiteness Property

We mentioned earlier that it is important that all optimal values be finite. The following theorem states a condition when this is the case for one-switch utility functions.

Theorem 4. *Consider any one-switch utility function $U(w) = Cw - D\gamma^w$ with $C, D > 0$ and $0 < \gamma < 1$.*

Figure 2: Probabilistic Blocks-World Example

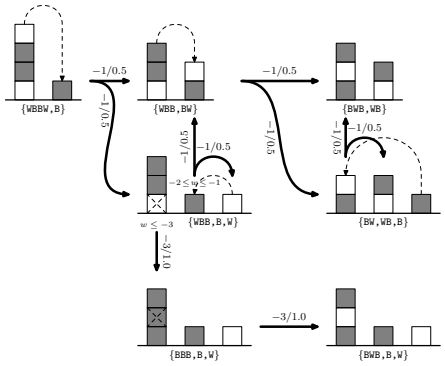


Figure 3: Optimal Policy for One-Switch Utility Function

The optimal values $v^*_U(s)$ for the one-switch utility function are finite for all states $s \in S$ if there exists a policy whose values $v^\pi_{U'}(s)$ for the concave exponential utility function $U'(w) = -\gamma^w$ are finite for all states $s \in S$.

Proof. There exist $c', d' > 0$ such that, for all wealth levels $w \leq 0$,

$$U(w) \geq -c' - d'\gamma^w = -c' + d'U'(w)$$

since the term $d'\gamma^w$ dominates $U(w)$. Therefore, it holds for the given policy π and all states $s \in S$ and horizons T that

$$\begin{aligned} v^\pi_{U,T}(s) &= E^{s,\pi}[U(w_T)] \\ &\geq E^{s,\pi}\left[-c' - d'\gamma^{w_T}\right] = -c' + d'v^\pi_{U',T}(s) \end{aligned}$$

and thus also

$$\begin{aligned} v^\pi_U(s) &= \lim_{T\to\infty} v^\pi_{U,T}(s) \\ &\geq \lim_{T\to\infty} \left(-c' + d'v^\pi_{U',T}(s)\right) = -c' + d'v^\pi_{U'}(s). \end{aligned}$$

It therefore holds that

$$\begin{aligned} v^*_U(s) &= \sup_{\pi' \in \Pi^{\text{HR}}} v^{\pi'}_U(s) \geq v^\pi_U(s) \\ &\geq -c' + d'v^\pi_{U'}(s) > -\infty. \end{aligned}$$

□

Example

We use a probabilistic blocks-world example to illustrate risk-sensitive planning with one-switch utility functions (Koenig & Simmons, 1994). The domain is a standard block-world 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. (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 has a reward of -1, and the paint action has a reward of -3. Figure 2 shows the initial configuration of the 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 blocks-world example has 162 states, which

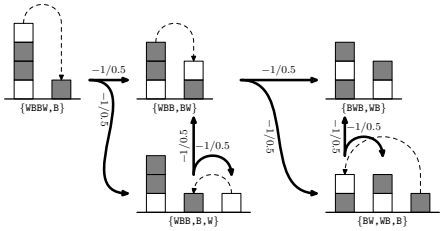


Figure 4: Optimal Policy for Linear Utility Function

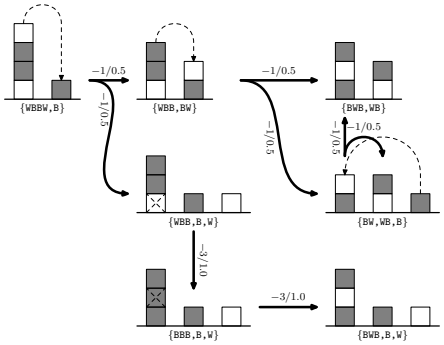
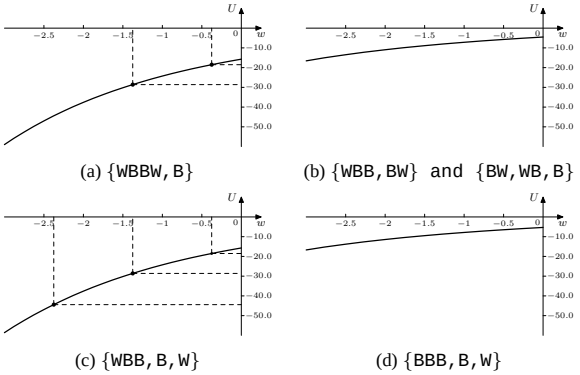


Figure 5: Optimal Policy for Exponential Utility Function

we describe as a set of stacks by listing the blocks in each stack from bottom to top. For example, the initial configuration is {`WBBW,B`}. We use functional value iteration to find the augmented SD-optimal policy for the one-switch utility function $U(w) = w - 0.5 \times 0.6^w$. Functional value iteration terminates quickly. Figure 3 depicts the augmented SD-optimal policy and Figure 6 shows the optimal functional value functions for the five non-goal states that are reachable from the initial configuration under the augmented SD-optimal policy. Table 1 shows the breakpoints of the functional value function and the values of the piecewise one-switch value function at the breakpoints in parentheses followed by the d and e values of the segments to their immediate left.

We compare the augmented SD-optimal policy for the one-switch utility function $U(w) = w - 0.5 \times 0.6^w$ against two other policies that are easier to calculate. Figure 5 shows the optimal policy for the concave exponential utility function $U(w) = -0.6^w$, and Figure 4 shows the op-



(a) {`WBBW,B`} (b) {`WBB,BW`} and {`BW,WB,B`}

(c) {`WBB,B,W`} (d) {`BBB,B,W`}

Figure 6: Optimal Functional Value Function

Table 1: Segments of the Value Functions from Figure 6

{WBB,BW} and {BW,WB,B}

(w^i, u^i)	d^i	e^i
$(0.00, -4.50)$	2.50	2.00

{BBB,B,W}

(w^i, u^i)	d^i	e^i
$(0.00, -5.31)$	2.32	3.00

{WBBW,B}

(w^i, u^i)	d^i	e^i
$(0.00, -15.72)$	11.47	4.25
$(-0.38, -18.52)$	11.26	4.50
$(-1.38, -28.61)$	11.02	5.00

{WBB,B,W}

(w^i, u^i)	d^i	e^i
$(0.00, -15.72)$	11.47	4.25
$(-0.38, -18.52)$	11.26	4.50
$(-1.38, -28.61)$	11.02	5.00
$(-2.38, -44.43)$	10.72	6.00

timal policy for the linear utility function $U(w) = w$, that is, the optimal risk-neutral policy. Both of these policies can be calculated with standard versions of value iteration since the optimal policies for linear and exponential utility functions are known to be in Π^{SD} (Bertsekas & Tsitsiklis, 1991; Patek, 2001). The three policies differ in the actions that they execute in state {WBB,B,W}. The optimal policy for the concave exponential utility function always executes a move action in state {WBB,B,W}, and the optimal policy for the linear utility function always executes a paint action in state {WBB,B,W}. The optimal policy for the one-switch utility function, on the other hand, executes an action in state {WBB,B,W} that depends on the wealth level. It is a move action for wealth level -1 or -2 and a paint action for wealth level -3 or smaller. The augmented SD-optimal policy executes the same move action in the initial configuration until it succeeds or has tried it three times. If it succeeds, it then solves the problem by repeatedly executing another move action until it succeeds, otherwise it solves the problem by executing two different paint actions. The optimal policies for the concave exponential and linear utility functions are not optimal for decision makers with the given one-switch utility function. If we evaluate the policies with the one-switch utility function, then the expected utility of the initial configuration is -16.01 for the concave exponential utility function and -16.50 for the linear utility function, whereas the maximal expected utility is -15.72 for the one-switch utility function.

Conclusions

In this paper, we studied how to maximize the expected utility of a Markov decision problem for a given one-switch utility function. We first studied an approach that augments the states of the Markov decision problem with the wealth level. The properties of the resulting infinite Markov decision problem then allowed us to generalize the standard risk-neutral version of value iteration from manipulating values to manipulating functions that map wealth levels to values. Large parts of this paper applied to arbitrary utility functions, only the part about how to represent the functional value functions used by the new version of value iteration was specific to one-switch utility functions. We therefore expect our ideas to apply to additional nonlinear utility functions as well. In the meantime, we have been able to use properties of one-switch utility functions to derive an exact algorithm for them, similar to backward induction, but this algorithm is specific to one-switch utility functions (Liu & Koenig, 2005).

Acknowledgments

We apologize that the very short period for revisions coincided with the end of the semester and did not allow us to submit a final version of the paper that satisfies our own quality standards. 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

Bell, D. E., and Fishburn, P. C. 2001. Strong one-switch utility. *Management Science* 47(4):601–604.

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. 1997. *Planning under Uncertainty in Dynamic Domains*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University.

Gelles, G. M., and Mitchell, D. W. 1999. Broadly decreasing risk aversion. *Management Science* 45:1432–1439.

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.

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.

Liu, Y., and Koenig, S. 2005. Risk-sensitive planning with one-switch utility functions: Backward induction. Submitted.

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

Nakamura, Y. 1996. Sumex utility functions. *Mathematical Social Sciences* 31:39–47.

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.