



**Table 1: Utility-Theoretic Analysis of the Game-Show Problem**

Utility Function	Leave	Guess			Utility Difference (Leave – Guess)	Optimal Decision
	Utility of \$500,000 (with prob. 1.0)	Utility of \$32,000 (with prob. 0.5)	Utility of \$1,000,000 (with prob. 0.5)	Expected Utility		
$U_\ell(w) = w$	$w_0 + 500,000$	$w_0 + 32,000$	$w_0 + 1,000,000$	$w_0 + 516,000$	-16,000	Guess
$U_e(w) = -\gamma^w$	$-0.6065\gamma^{w_0}$	$-0.9685\gamma^{w_0}$	$-0.3679\gamma^{w_0}$	$-0.6682\gamma^{w_0}$	$0.0617\gamma^{w_0}$	Leave
$U_{1s}(w) = w - D\gamma^w$	$w_0 + 500,000$ $-0.6065D\gamma^{w_0}$	$w_0 + 32,000$ $-0.9685D\gamma^{w_0}$	$w_0 + 1,000,000$ $-0.3679D\gamma^{w_0}$	$w_0 + 516,000$ $-0.6682D\gamma^{w_0}$	-16,000 $+0.0617D\gamma^{w_0}$	$w_0 > 1.35 \times 10^6$ : Guess $w_0 < 1.35 \times 10^6$ : Leave

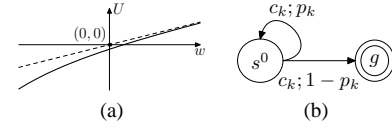
$w_0$  = the initial wealth level.  $\gamma = 0.999999$ .  $D = 10^6$ .

neutral risk attitude. A human being is risk-neutral iff they make decisions that maximize their expected future wealth level. The calculations for the game-show problem with the linear utility function  $U_\ell(w) = w$  (where  $w$  is the wealth level) are shown in the first row of Table 1. The optimal decision is to guess independent of the initial wealth level. Concave utility functions imply a risk-averse risk attitude. A human being is risk-averse iff they make decisions that do not maximize their expected future wealth level provided that the variance of their future wealth level is sufficiently reduced. Researchers often assume for mathematical convenience that risk-averse human beings have (concave) exponential utility functions, which are of the form  $U_e(w) = -\gamma^w$  for parameter  $0 < \gamma < 1$  [4]. The calculations for the game-show problem with the exponential utility function  $U_e(w) = -0.999999^w$  are shown in the second row of Table 1. The optimal decision is to leave independent of the initial wealth level, which is consistent with the decision of most contestants in this situation. In general, different human beings can have different utility functions and thus different risk attitudes. Thus, they can make different decisions.

The decisions of human beings with linear or exponential utility functions are independent of their initial wealth level and thus do not change as their wealth level increases, which is why these utility functions are also known as zero-switch utility functions [1]. In reality, the decisions of human beings often change with their wealth level, which is why it can be unrealistic to use linear and exponential utility functions. For example, the rewards in the game-show problem are high compared to the wealth level of average people, which explains why they are expected to be risk-averse in game shows. However, the rewards are low compared to the wealth level of billionaires, which explains why they are expected to be risk-neutral. It is more realistic to assume that human beings are always risk-averse but become risk-neutral in the limit as their wealth level increases, that the utility increases monotonically with their wealth level (since they can always give money away), and that their decision between any two alternatives changes at most once as their wealth level increases. Human beings whose behavior satisfies these assumptions have special kinds of one-switch utility functions which are of the form

$$U_{1s}(w) = w - D\gamma^w$$

for parameters  $D > 0$  and  $0 < \gamma < 1$ , as illustrated in Figure 1(a), where the dashed line indicates the linear utility function. The parameter  $D$  provides an adjustable tradeoff between risk-neutrality (linear term) and risk-aversion (exponential term). One-switch utility functions were proposed in [1] and have been studied extensively with many applications in the decision analysis community [1, 11, 5, 2]. The calculations for the game-show problem with the one-switch utility function  $U_{1s}(w) = w - 10^6 \times 0.999999^w$  are shown in the third row of Table 1. The optimal decision now depends on the initial wealth level. For example, the expected utility of leaving is  $-1.065 \times 10^5$ , while the expected utility of guessing is



**Figure 1: (a) One-Switch Utility Function (b) Two-State GDMDP**

Action	Description	Success probability	Cost
1	do-it-yourself	0.25	\$ 100
2	hire a professional	0.95	\$ 1,000
3	buy a termite-free house ... and sell the infested one!	1.00	\$10,000

**Figure 2: Termite Problem**

$-1.522 \times 10^5$  if your initial wealth level is zero. Thus, you would need to leave in order to maximize your expected utility. On the other hand, the exponential term of the one-switch utility function rapidly approaches zero as the initial wealth level increases and the linear term then dominates. Thus, you would eventually need to guess in order to maximize your expected utility. To determine at which initial wealth level  $w_0$  you should switch from leaving to guessing, we solve the equation

$$-16,000 + 0.0617 \times 10^6 \cdot 0.999999^{w_0} = 0 \Rightarrow w_0 = 1.35 \times 10^6.$$

Thus, you should switch from leaving to guessing as your wealth level increases beyond about \$1,350,000.

### 3. GOAL-DIRECTED MDPS

The game-show problem involves only one action rather than a sequence of actions, the hallmark of artificial intelligence planning. Imagine therefore that you own a termite-infested wooden house but would like to own a termite-free house. The actions are given in Figure 2. The outcome of each action is either still owning a termite-infested house or a termite-free house. You can thus achieve the goal state only with a sequence of actions. For example, you could attempt to exterminate the termites yourself twice in a row and then buy a termite-free house, stopping after your first success. This policy costs you \$100 with probability 0.25,  $\$100 + \$100 = \$200$  with probability  $[1 - 0.25] 0.25 = 0.19$ , and  $\$100 + \$100 + \$10,000 = \$10,200$  with probability  $[1 - 0.25] [1 - 0.25] = 0.56$ .

Sequential planning problems can be described with goal-directed Markov decision problems (GDMDPs). Formally, a GDMDP consists of a finite set of states  $S$ , a nonempty finite set of goal states  $G \subseteq S$  and a finite set of actions  $A_s$  for each non-goal state  $s \in S' = S \setminus G$ . The agent starts execution at time step  $t = 0$  and always chooses one action  $a \in A_s$  to execute in its current state  $s \in S'$ . Its execution results with probability  $P(s'|s, a)$  in a finite reward  $r(s, a, s') < 0$  at the current time step and a transition to successor state  $s' \in S$  at the next time step. The agent stops acting when it reaches a goal state and receives no more rewards

thereafter. We use  $s_t$  and  $a_t$  to denote the state and action at time step  $t$ , respectively. We use  $r_t = r(s_t, a_t, s_{t+1})$  to denote the reward for executing action  $a_t$ . (We define  $r_t = 0$  after the agent reaches a goal state.) Finally, we use  $w_t = w_0 + \sum_{i=0}^{t-1} r_i$  to denote the wealth level at time step  $t$  directly before executing action  $a_t$ . The wealth level starts at the initial wealth level  $w_0$  and then decreases over time since all rewards are negative. For example, Figure 1(b) shows how the termite problem can be modeled as a two-state GDMDP with three actions. One state is the start state  $s^0$  (owning an infested house), and the other state is the goal state  $g$  (owning a termite-free house). In general, the agent has to choose among  $n$  actions to execute in state  $s^0$ , numbered from 1 to  $n$ . The arcs indicate the state transitions for each action  $k = 1 \dots n$  and are annotated with their probabilities  $p_k$  and rewards  $c_k$ . For the termite GDMDP, there are three actions. Their probabilities are their failure probabilities, and their rewards are their negative costs. For example, probability  $p_1 = 0.75$  and reward  $c_1 = -100$  for action 1, namely to exterminate the termites yourself.

Policies specify which actions an agent should execute. In the most general case, these action can probabilistically depend on the current state as well as all previous states and actions [15]. The agent should follow the policy that maximizes its expected utility. For all utility functions  $U$  and all policies  $\pi$ , we define the value  $v_U^\pi(s, w) = \lim_{t \rightarrow \infty} E_{s,w}^\pi [U(w_t)]$  as the expected utility of an agent with initial state  $s = s_0$  and initial wealth level  $w = w_0$  that follows policy  $\pi$ . We also define the optimal value  $v_U^*(s, w) = \max_\pi v_U^\pi(s, w)$  as the highest possible expected utility of an agent with initial state  $s$  and initial wealth level  $w$ . We assume that value  $v_U^*(s, w)$  is finite for all states  $s \in S$  and wealth levels  $w$  since it is otherwise impossible to compare policies [8]. We define an optimal policy  $\pi_U^*$  to be a policy with  $v_U^{\pi_U^*}(s, w) = v_U^*(s, w)$  for all states  $s \in S$  and wealth levels  $w$ . For the utility functions  $U_\ell$ ,  $U_e$  and  $U_{1s}$ , we refer to the  $\text{MEU}_\ell$  (= MER),  $\text{MEU}_e$  and  $\text{MEU}_{1s}$  planning objectives, respectively, and replace the subscripts  $U$  in values  $v_U^\pi(s, w)$  and  $v_U^*(s, w)$  with  $\ell$ ,  $e$  and  $1s$ , respectively. We use the shorthands  $v_\ell^\pi(s) = v_\ell^\pi(s, 0)$  and  $v_\ell^*(s) = v_\ell^*(s, 0) = \max_\pi v_\ell^\pi(s, 0) = \max_\pi v_\ell^*(s)$ . Similarly, we use the shorthands  $v_e^\pi(s) = v_e^\pi(s, 0)$  and  $v_e^*(s) = v_e^*(s, 0) = \max_\pi v_e^\pi(s, 0) = \max_\pi v_e^*(s)$ . We exploit the relationships between these values in the next section.

## 4. PLANNING OBJECTIVES

One-switch utility functions are linear combinations of linear and exponential utility functions:

$$U_{1s}(w) = w - D\gamma^w = U_\ell(w) + D \cdot U_e(w).$$

For all one-switch utility functions  $U_\ell$  and all policies  $\pi$ , we thus have

$$\begin{aligned} v_{1s}^\pi(s, w) &= \lim_{t \rightarrow \infty} E_{s,w}^\pi [U_{1s}(w_t)] = \lim_{t \rightarrow \infty} E_{s,w}^\pi [U_\ell(w_t) + D \cdot U_e(w_t)] \\ &= \lim_{t \rightarrow \infty} E_{s,w}^\pi [U_\ell(w_t)] + D \cdot \lim_{t \rightarrow \infty} E_{s,w}^\pi [U_e(w_t)] \\ &= v_\ell^\pi(s, w) + D \cdot v_e^\pi(s, w). \end{aligned} \quad (1)$$

We therefore need to consider the  $\text{MEU}_\ell$  and  $\text{MEU}_e$  planning objectives. For these planning objectives, there always exists a stationary and deterministic (SD) policy that is optimal [3, 13]. An SD policy  $\pi$  maps every state  $s \in S'$  to the policy  $\pi(s) \in A_s$  that an agent in state  $s$  should execute independent of its wealth level. Thus, action  $a_t = \pi(s_t)$  and reward  $r_t$  are independent of its initial wealth level. For the termite GDMDP, there are only three SD policies, namely  $\pi_k(s^0) = k$  for actions  $k = 1, 2, 3$ .

First, we consider the  $\text{MEU}_\ell$  planning objective: For the linear utility function  $U_\ell$  and all SD policies  $\pi$ , we have

$$\begin{aligned} v_\ell^\pi(s, w) &= \lim_{t \rightarrow \infty} E_{s,w}^\pi [U_\ell(w_t)] = \lim_{t \rightarrow \infty} E_{s,w}^\pi [w_t] \\ &= \lim_{t \rightarrow \infty} E_{s,w}^\pi \left[ w + \sum_{i=0}^{t-1} r_i \right] = w + \lim_{t \rightarrow \infty} E_s^\pi \left[ \sum_{i=0}^{t-1} r_i \right] \\ &= w + v_\ell^\pi(s), \end{aligned} \quad (2)$$

where values  $v_\ell^\pi(s)$  are independent of the wealth level  $w$  and satisfy the policy-evaluation equations [3]

$$\begin{aligned} v_\ell^\pi(s) &= 0 & \forall s \in G \\ v_\ell^\pi(s) &= \sum_{s' \in S} P(s'|s, \pi(s)) [r(s, \pi(s), s') + v_\ell^\pi(s')] & \forall s \in S'. \end{aligned} \quad (3)$$

Thus,  $v_\ell^*(s, w) = w + v_\ell^*(s)$ , where the optimal values  $v_\ell^*(s)$  are also independent of the wealth level  $w$  and satisfy the optimality equations [3]

$$\begin{aligned} v_\ell^*(s) &= 0, & \forall s \in G \\ v_\ell^*(s) &= \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) [r(s, a, s') + v_\ell^*(s')] & \forall s \in S'. \end{aligned}$$

An agent in state  $s \in S'$  with wealth level  $w$  follows an  $\text{MEU}_\ell$ -optimal policy if it executes an action from  $\arg \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) [r(s, a, s') + v_\ell^*(s')]$ . For the termite GDMDP, the policy-evaluation equations are

$$v_\ell^{\pi_k}(g) = 0, \quad v_\ell^{\pi_k}(s^0) = p_k [c_k + v_\ell^{\pi_k}(s^0)] + [1 - p_k] [c_k + v_\ell^{\pi_k}(g)].$$

The values thus are

$$v_\ell^{\pi_k}(s^0) = \frac{c_k}{1 - p_k}.$$

The  $\text{MEU}_\ell$ -optimal policy is  $\pi_1$  since  $v_\ell^{\pi_1}(s^0) = -400$ ,  $v_\ell^{\pi_2}(s^0) = -1,052$  and  $v_\ell^{\pi_3}(s^0) = -10,000$ .

Second, we consider the  $\text{MEU}_e$  planning objective: For all exponential utility functions  $U_e$  and all SD policies  $\pi$ , we have similarly

$$\begin{aligned} v_e^\pi(s, w) &= \lim_{t \rightarrow \infty} E_{s,w}^\pi [U_e(w_t)] = \lim_{t \rightarrow \infty} E_{s,w}^\pi [-\gamma^{w_t}] \\ &= \lim_{t \rightarrow \infty} E_{s,w}^\pi [-\gamma^w \cdot \gamma^{\sum_{i=0}^{t-1} r_i}] = \gamma^w \cdot \lim_{t \rightarrow \infty} E_s^\pi [-\gamma^{\sum_{i=0}^{t-1} r_i}] \\ &= \gamma^w \cdot v_e^\pi(s), \end{aligned} \quad (4)$$

where the values  $v_e^\pi(s)$  are independent of the wealth level  $w$  and satisfy the policy-evaluation equations [13]

$$\begin{aligned} v_e^\pi(s) &= -1 & \forall s \in G \\ v_e^\pi(s) &= \sum_{s' \in S} P(s'|s, \pi(s)) \gamma^{r(s, \pi(s), s')} \cdot v_e^\pi(s') & \forall s \in S', \end{aligned} \quad (5)$$

provided that the values  $v_e^\pi(s)$  are finite for all states  $s \in S'$ . Thus,  $v_e^*(s, w) = \gamma^w \cdot v_e^*(s)$ , where the optimal values  $v_e^*(s)$  are also independent of the wealth level  $w$  and satisfy the optimality equations [13]

$$\begin{aligned} v_e^*(s) &= -1 & \forall s \in G \\ v_e^*(s) &= \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) \gamma^{r(s, a, s')} \cdot v_e^*(s') & \forall s \in S', \end{aligned}$$

provided that the optimal values  $v_e^*(s)$  are finite for all states  $s \in S'$ . An agent in state  $s \in S'$  with wealth level  $w$  follows an  $\text{MEU}_e$ -optimal policy if it executes an action from  $\arg \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) \gamma^{r(s, a, s')} \cdot v_e^*(s')$ . For the termite GDMDP, the policy-evaluation equations are

$$v_e^{\pi_k}(g) = -1 \text{ and } v_e^{\pi_k}(s^0) = p_k \gamma^{c_k} v_e^{\pi_k}(s^0) + [1 - p_k] \gamma^{c_k} v_e^{\pi_k}(g),$$

provided that the values  $v_e^{\pi_k}(s^0)$  are finite, which is the case if

$p_k \gamma^{c_k} < 1$ . (Otherwise,  $v_e^{\pi_k}(s^0) = -\infty$ .) The values thus are

$$v_e^{\pi_k}(s^0) = \begin{cases} -\frac{[1-p_k]\gamma^{c_k}}{1-p_k\gamma^{c_k}} & p_k\gamma^{c_k} < 1 \\ -\infty & p_k\gamma^{c_k} \geq 1. \end{cases}$$

Assume that your exponential utility function is  $U_e(w) = -0.997^w$ . Then, the MEU<sub>e</sub>-optimal policy is  $\pi_3$  since  $v_e^{\pi_1}(s^0) = v_e^{\pi_2}(s^0) = -\infty$  and  $v_e^{\pi_3}(s^0) = -1.1179 \times 10^{13}$ .

Third, we consider the MEU<sub>1s</sub> planning objective: For all one-switch utility functions  $U_{1s}$  and all SD policies  $\pi$ , we can decompose the values  $v_{1s}^\pi(s, w)$  according to Eq. (1), Eq. (2) and Eq. (4) as follows:

$$v_{1s}^\pi(s, w) = w + v_\ell^\pi(s) + D\gamma^w \cdot v_e^\pi(s). \quad (6)$$

For the termite GDMDP, these equations are

$$v_{1s}^{\pi_k}(g, w) = U_{1s}(w) = w - D\gamma^w$$

$$v_{1s}^{\pi_k}(s^0, w) = \begin{cases} w + \frac{c_k}{1-p_k} - D\gamma^w \cdot \frac{[1-p_k]\gamma^{c_k}}{1-p_k\gamma^{c_k}} & p_k\gamma^{c_k} < 1 \\ -\infty & p_k\gamma^{c_k} \geq 1. \end{cases}$$

Assume that your one-switch utility function is  $U_{1s}(w) = w - 10^{-9} \times 0.997^w$  and your initial wealth level is  $w = 0$  (due to your mortgage debt). Then, the MEU<sub>1s</sub>-optimal SD policy is  $\pi_3$  since  $v_{1s}^{\pi_1}(s^0, 0) = v_{1s}^{\pi_2}(s^0, 0) = -\infty$  and  $v_{1s}^{\pi_3}(s^0, 0) = -21, 179$ .

For the MEU<sub>1s</sub> planning objective, there does not necessarily exist an SD policy that is optimal but there always exists an augmented stationary and deterministic (ASD) policy that is optimal [9]. An ASD policy maps every combination of a state  $s \in S'$  and wealth level  $w$  to the action  $\pi(s, w) \in A_s$  that an agent in state  $s$  with wealth level  $w$  should execute. Thus, action  $a_t = \pi(s_t, w_t) = \pi(s_t, w_0 + \sum_{i=0}^{t-1} r_i)$  and reward  $r_t$  are no longer independent of its initial wealth level  $w_0$ . For the termite GDMDP, assume again that your one-switch utility function is  $U_{1s}(w) = w - 10^{-9} \times 0.997^w$  and your initial wealth level is zero. The policy of attempting to exterminate the termites yourself twice in a row and then buy a termite-free house, stopping after your first success, can be formulated as an ASD policy  $\pi$  with actions  $\pi(s^0, 0) = \pi(s^0, -100) = 1$  and  $\pi(s^0, -200) = 3$ . Its value is  $v_{1s}^\pi(s^0) = 0.25 \cdot U_{1s}(-100) + 0.19 \cdot U_{1s}(-200) + 0.56 \cdot U_{1s}(-10, 200) = -17, 193$ . Thus, this ASD policy is better than the MEU<sub>1s</sub>-optimal SD policy  $\pi_3$ .

We now generalize Eq. (6) to ASD policies by reformulating special cases of Theorems 1 and 2 from [9], which hopefully make them more accessible to the reader. For all ASD policies  $\pi$  and all wealth levels  $w$ , we define the policy  $\pi_w$  that satisfies  $\pi_w(s, w') = \pi(s, w + w')$  for all wealth levels  $w'$ . (If  $\pi$  is an SD policy, then  $\pi_w = \pi$ .) The example ASD policy for the termite GDMDP has actions  $\pi_{-100}(s^0, 0) = \pi(s^0, -100) = 1$  and  $\pi_{-100}(s^0, -100) = \pi(s^0, -200) = 3$ .

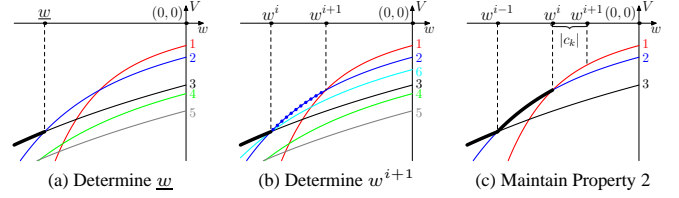
An agent with initial wealth level  $w$  that follows policy  $\pi$  executes action  $a_t = \pi(s_t, w + \sum_{i=0}^{t-1} r_i)$  at time step  $t$ , while an agent with initial wealth level 0 that follows policy  $\pi_w$  executes action  $a_t = \pi_w(s_t, \sum_{i=0}^{t-1} r_i) = \pi(s_t, w + \sum_{i=0}^{t-1} r_i)$  at time step  $t$ . These actions are the same and thus the probability distributions over the successor states and rewards are also the same. Thus, we have

$$v_\ell^\pi(s, w) = \lim_{t \rightarrow \infty} E_{s,w}^{\pi_w} \left[ w + \sum_{i=0}^{t-1} r_i \right] = w + \lim_{t \rightarrow \infty} E_s^{\pi_w} \left[ \sum_{i=0}^{t-1} r_i \right]$$

$$= w + v_\ell^{\pi_w}(s, 0) = w + v_\ell^{\pi_w}(s) \quad (7)$$

and

$$v_e^\pi(s, w) = \lim_{t \rightarrow \infty} E_{s,w}^{\pi_w} \left[ -\gamma^w \cdot \gamma^{\sum_{i=0}^{t-1} r_i} \right] = \gamma^w \cdot \lim_{t \rightarrow \infty} E_s^{\pi_w} \left[ -\gamma^{\sum_{i=0}^{t-1} r_i} \right]$$



**Figure 3: Illustration of Backward Induction**

$$= \gamma^w \cdot v_e^{\pi_w}(s, 0) = \gamma^w \cdot v_e^{\pi_w}(s). \quad (8)$$

For all one-switch utility functions  $U_{1s}$  and all ASD policies  $\pi$ , we can therefore decompose the value  $v_{1s}^\pi(s, w)$  according to Eq. (1), Eq. (7) and Eq. (8) as follows:

$$v_{1s}^\pi(s, w) = w + v_\ell^{\pi_w}(s) + D\gamma^w \cdot v_e^{\pi_w}(s). \quad (9)$$

The values  $v_{1s}^\pi(s, w)$  satisfy policy-evaluation equations that are implied by [9] but were not explicitly stated there

$$v_{1s}^\pi(s, w) = U_{1s}(w) = w - D\gamma^w \quad \forall s \in G, \forall w$$

$$v_{1s}^\pi(s, w) = \sum_{s' \in S} P(s'|s, \pi(s, w)) v_{1s}^\pi(s', w + r(s, \pi(s, w), s'))$$

$$\quad \forall s \in S', \forall w,$$

provided that the values  $v_{1s}^\pi(s, w)$  are finite for all states  $s \in S'$  and all wealth levels  $w$ . The optimal values  $v_{1s}^*(s, w)$  satisfy optimality equations that are again implied by [9]

$$v_{1s}^*(s, w) = U_{1s}(w) = w - D\gamma^w \quad \forall s \in G, \forall w$$

$$v_{1s}^*(s, w) = \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) v_{1s}^*(s', w + r(s, a, s'))$$

$$\quad \forall s \in S', \forall w, \quad (10)$$

provided that the optimal values  $v_{1s}^*(s, w)$  are finite for all state  $s \in S'$  and all wealth levels  $w$ . Then, an agent in state  $s \in S'$  with wealth level  $w$  follows an MEU<sub>1s</sub>-optimal policy if it executes an action from  $\arg \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) v_{1s}^*(s', w + r(s, a, s'))$ . Therefore, all that is left to do is to determine the optimal values  $v_{1s}^*(s, w)$ . The only previous approach for this purpose that we know of is based on functional value iteration [9] and typically determines the optimal values  $v_{1s}^*(s, w)$  only in the limit although it needs to terminate in finite time in practice. However, the approach does not provide a termination condition for improving the values nor an error bound on the resulting policies. We therefore introduce that backward-induction method that exploits the relationship between one-switch utility functions and exponential utility functions to determine the optimal values  $v_{1s}^*(s, w)$  and thus an MEU<sub>1s</sub>-optimal policy in finite time.

## 5. INDUCTIVE FOUNDATION

We now establish the induction foundation for the backward-induction method. We utilize the fact that one-switch utility functions are weighted sums of linear and exponential utility functions to prove that there exists an SD policy that is an MEU<sub>1s</sub>-optimal policy as the wealth level approaches negative infinity. The backward-induction method then starts with this policy and augments it for higher and higher wealth levels. We illustrate the backward-induction method in Figure 3 for a general two-state GDMDP with initial wealth level zero, where the graphs are the value functions  $v_{1s}^{\pi_k}(s^0, \cdot)$  of the policies  $\pi_k$ . The following lemma relates the optimal values  $v_{1s}^*(s, w)$  and  $v_e^*(s)$  and uses the fact

that we have for all MEU<sub>1s</sub>-optimal ASD policies  $\pi_{1s}^*$  according to Eq. (9)

$$v_{1s}^*(s, w) = v_{1s}^{\pi_{1s}^*}(s, w) = w + v_{\ell}^{\pi_{1s}^*}(s) + D\gamma^w v_e^{\pi_{1s}^*}(s). \quad (11)$$

LEMMA 1.  $\lim_{w \rightarrow -\infty} v_{1s}^*(s, w)\gamma^{-w} = Dv_e^*(s)$  for all states  $s \in S$ .

PROOF. For all MEU<sub>1s</sub>-optimal ASD policies  $\pi_{1s}^*$ , we have according to Eq. (11) and the fact that  $v_{\ell}^{\pi}(s) \leq v_{\ell}^*(s)$  and  $v_e^{\pi}(s) \leq v_e^*(s)$  for all policies  $\pi$

$$\begin{aligned} v_{1s}^*(s, w)\gamma^{-w} &= v_{1s}^{\pi_{1s}^*}(s, w)\gamma^{-w} = w\gamma^{-w} + v_{\ell}^{\pi_{1s}^*}(s)\gamma^{-w} + Dv_e^{\pi_{1s}^*}(s)\gamma^{-w} \\ &\leq w\gamma^{-w} + v_{\ell}^*(s)\gamma^{-w} + Dv_e^*(s). \end{aligned}$$

and thus

$$\limsup_{w \rightarrow -\infty} v_{1s}^*(s, w)\gamma^{-w} \leq \lim_{w \rightarrow -\infty} [w\gamma^{-w} + v_{\ell}^*(s)\gamma^{-w} + Dv_e^*(s)] = Dv_e^*(s).$$

On the other hand, for all MEU<sub>e</sub>-optimal SD policies  $\pi_e^*$ , we have according to Eq. (9) and the fact that  $v_{1s}^*(s, w) \geq v_{1s}^{\pi_e^*}(s, w)$  for all policies  $\pi$

$$\begin{aligned} v_{1s}^*(s, w)\gamma^{-w} &\geq v_{1s}^{\pi_e^*}(s, w)\gamma^{-w} = w\gamma^{-w} + v_{\ell}^{\pi_e^*}(s)\gamma^{-w} + Dv_e^{\pi_e^*}(s) \\ &= w\gamma^{-w} + v_{\ell}^{\pi_e^*}(s)\gamma^{-w} + Dv_e^*(s) \end{aligned}$$

and thus

$$\liminf_{w \rightarrow -\infty} v_{1s}^*(s, w)\gamma^{-w} \geq \lim_{w \rightarrow -\infty} [w\gamma^{-w} + v_{\ell}^{\pi_e^*}(s)\gamma^{-w} + Dv_e^*(s)] = Dv_e^*(s).$$

Therefore, the lemma holds.  $\square$

The lemma implies that an MEU<sub>1s</sub>-optimal policy is also MEU<sub>e</sub>-optimal in the limit, which is not surprising since the exponential term in Eq. (11) grows faster than the linear term. However, not every MEU<sub>e</sub>-optimal policy is also MEU<sub>1s</sub>-optimal in the limit. For a general two-state GDMDP, assume that there are only two actions and both of the corresponding SD policies  $\pi_1$  and  $\pi_2$  are MEU<sub>e</sub>-optimal. Thus, we have for actions  $k = 1, 2$

$$v_e^*(s^0) = v_e^{\pi_k}(s^0) = -\frac{[1-p_k]\gamma^{c_k}}{1-p_k\gamma^{c_k}} < -1,$$

provided that  $p_k\gamma^{c_k} < 1$ , which implies

$$p_k = \frac{1 + v_e^*(s^0)\gamma^{-c_k}}{1 + v_e^{\pi_k}(s^0)} \quad \text{and} \quad c_k \in [\log_{\gamma}(-v_e^*(s^0)), 0].$$

Thus, there are combinations of probabilities  $p_k$  and rewards  $c_k$  that achieve the same optimal value  $v_e^*(s^0)$  but differ in their values  $v_{\ell}^{\pi_k}(s^0) = \frac{c_k}{1-p_k}$ . Only the policy  $\pi_k$  with the highest value  $v_{\ell}^{\pi_k}(s^0)$  can possibly be MEU<sub>1s</sub>-optimal according to Eq. (11). The following lemma formalizes this observation. It defines an auxiliary GDMDP such that all policies for the auxiliary GDMDP are MEU<sub>e</sub>-optimal for the given GDMDP. The policy that is MEU<sub>e</sub>-optimal for the auxiliary GDMDP then is MEU<sub>1s</sub>-optimal for the given GDMDP for all wealth levels no higher than some wealth level threshold.

LEMMA 2. For all MEU<sub>1s</sub>-optimal policies  $\pi_{1s}^*$ , there exists a wealth level threshold  $\underline{w}$  such that it holds for all wealth levels  $w \leq \underline{w}$  that

1.  $v_e^{\pi_{1s}^*}(s) = v_e^*(s)$  for all states  $s \in S'$ .
2.  $v_{\ell}^{\pi_{1s}^*}(s) = v_{\ell}^{\pi_e^{**}}(s)$  for all states  $s \in S'$ , where  $\pi_e^{**}$  is any SD MEU<sub>e</sub>-optimal policy for the auxiliary GDMDP which

is the same as the original GDMDP except that the agent chooses its actions only from the MEU<sub>e</sub>-optimal actions, that is, the sets

$$A_e^*(s) = \arg \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a)\gamma^{r(s, a, s')}v_e^*(s').$$

3.  $v_{1s}^*(s, w) = v_{1s}^{\pi_e^{**}}(s, w)$  for all states  $s \in S'$ , meaning that  $\pi_e^{**}$  is an MEU<sub>1s</sub>-optimal policy for all wealth levels  $w \leq \underline{w}$ .

The proof of the lemma can be found in [7]. We need to determine the wealth level threshold  $\underline{w}$  to establish the induction foundation, as shown in Figure 3(a). For a general two-state GDMDP, assume that there are only two actions and that  $\pi_1$  is an MEU<sub>1s</sub>-optimal policy for all wealth levels  $w \leq \underline{w}$ , whereas  $\pi_2$  is not. Assume further that  $\pi_1$  is no longer an MEU<sub>1s</sub>-optimal policy for wealth levels that are higher than the wealth level threshold  $\underline{w}$  by a positive infinitesimal. Then, we have for all wealth levels  $w \leq \underline{w}$  according to Eq. (11) and the fact that  $(\pi_1)_w = \pi_1$  (since  $\pi_1$  is an SD policy)

$$v_{1s}^*(s^0, w) = v_{1s}^{\pi_1}(s^0, w) = w + v_{\ell}^{\pi_1}(s^0) + D\gamma^w v_e^{\pi_1}(s^0). \quad (12)$$

Now consider any wealth level  $w$  that is higher than the wealth level threshold  $\underline{w}$  by a positive infinitesimal but no higher than  $\underline{w} - c_k$  for all actions  $k = 1, 2$ . Then,  $v_{1s}^*(s^0, w + c_k) = v_{1s}^{\pi_1}(s^0, w + c_k)$  (since  $w + c_k \leq \underline{w}$ ) and we can rewrite Eq. (10) as follows:

$$\begin{aligned} v_{1s}^*(s^1, w) &= w - D\gamma^w \\ v_{1s}^*(s^0, w) &= \max_{k=1,2} [p_k v_{1s}^*(s^0, w + c_k) + [1-p_k] v_{1s}^*(s^1, w + c_k)] \\ &= \max_{k=1,2} [p_k v_{1s}^{\pi_1}(s^0, w + c_k) + [1-p_k] [w + c_k - D\gamma^{w+c_k}]] \\ &= \max_{k=1,2} [w + q_{\ell}^{\pi_1}(s^0, k) + D\gamma^w q_e^{\pi_1}(s^0, k)], \end{aligned}$$

where the last step uses both Eq. (12) and the definitions of the values  $q_{\ell}^{\pi_1}(s^0, k) = c_k + p_k v_{\ell}^{\pi_1}(s^0)$  and  $q_e^{\pi_1}(s^0, k) = \gamma^{c_k} [p_k - 1 + p_k v_e^{\pi_1}(s^0)]$  for  $k = 1, 2$ . It holds that  $q_{\ell}^{\pi_1}(s^0, 1) = v_{\ell}^{\pi_1}(s^0)$  according to Eq. (3) and  $q_e^{\pi_1}(s^0, 1) = v_e^{\pi_1}(s^0)$  according to Eq. (5).  $\pi_2$  is an MEU<sub>1s</sub>-optimal policy for wealth level  $w$  according to our assumptions, whereas  $\pi_1$  is not. Thus, we have

$$\begin{aligned} w + q_{\ell}^{\pi_1}(s^0, 2) + D\gamma^w q_e^{\pi_1}(s^0, 2) &> w + q_{\ell}^{\pi_1}(s^0, 1) + D\gamma^w q_e^{\pi_1}(s^0, 1) \\ &= w + v_{\ell}^{\pi_1}(s^0) + D\gamma^w v_e^{\pi_1}(s^0) \end{aligned}$$

or, equivalently,

$$\begin{aligned} q_{\ell}^{\pi_1}(s^0, 2) - v_{\ell}^{\pi_1}(s^0) &> D\gamma^w [v_e^{\pi_1}(s^0) - q_e^{\pi_1}(s^0, 2)] \\ w &> \log_{\gamma} \left( \frac{1}{D} \cdot \frac{q_{\ell}^{\pi_1}(s^0, 2) - v_{\ell}^{\pi_1}(s^0)}{v_e^{\pi_1}(s^0) - q_e^{\pi_1}(s^0, 2)} \right). \end{aligned}$$

The wealth level threshold  $\underline{w}$  can thus be set to the right-hand side of the last inequality. We actually set it to the minimum of the right-hand side of the last inequality and the initial wealth level since the agent only encounters wealth levels no higher than its initial wealth level. If the argument of the logarithm is non-positive and the logarithm thus is undefined, then policy  $\pi_1$  is MEU<sub>1s</sub>-optimal for all wealth levels and we set the wealth level threshold to the initial wealth level. The general case is just slightly more complex than for the two-state GDMDP since one needs to minimize over all states  $s \in S'$  and all actions  $a \in A_s \setminus A_e^*(s)$ . The following theorem summarizes the discussion. It uses the following definitions for all states  $s \in S'$  and actions  $a \in A_s$ :

$$\begin{aligned} q_e^*(s, a) &= \sum_{s' \in S} P(s'|s, a)\gamma^{r(s, a, s')}v_e^*(s') \\ q_{\ell}^{\pi_e^{**}}(s, a) &= \sum_{s' \in S} P(s'|s, a) [r(s, a, s') + v_{\ell}^{\pi_e^{**}}(s')]. \end{aligned}$$

THEOREM 3. *The statements in Lemma 2 hold for*

$$\underline{w} = \min_{s \in S'} \min_{a \in A_s \setminus A_e^*(s)} \log_{\gamma} \left( \max \left[ \gamma^{w_0}, \frac{1}{D} \cdot \frac{q_{\ell}^{\pi_e^{**}}(s, a) - v_{\ell}^{\pi_e^{**}}(s)}{v_e^*(s) - q_e^*(s, a)} \right] \right).$$

The proof of the theorem can be found in [7]. The maximum in the definition of the wealth level threshold  $\underline{w}$  takes care of the case where the numerator is negative or the wealth level threshold would otherwise be higher than the initial wealth level  $w_0$ . One needs to determine  $A_e^*(s)$ ,  $v_e^*(s)$ ,  $\pi_e^{**}$  and  $v_{\ell}^{\pi_e^{**}}(s)$  to calculate the wealth level threshold. The action sets  $A_e^*(s)$  and the optimal values  $v_e^*(s)$  can be calculated with the MEU<sub>e</sub> version of policy iteration [13], and then the policy  $\pi_e^{**}$  and the optimal values  $v_{\ell}^{\pi_e^{**}}(s)$  can be calculated with the MEU<sub>ℓ</sub> version of policy iteration [3]. One should not use versions of value iteration for this purpose since they typically determine optimal values only approximately. For the termite GMDP, we have shown that  $\pi_3$  is the only SD MEU<sub>e</sub>-optimal policy and thus  $\pi_e^{**} = \pi_3$  and  $A_e^*(s^0) = \{3\}$ . Then, we have  $v_{\ell}^{\pi_e^{**}}(s^0) = -10,000$ ,  $q_{\ell}^{\pi_e^{**}}(s^0, 1) = -7,600$ ,  $q_{\ell}^{\pi_e^{**}}(s^0, 2) = -1,500$ ,  $v_e^*(s^0) = -1.1179 \times 10^{13}$ ,  $q_e^*(s^0, 1) = -1.1323 \times 10^{13}$  and  $q_e^*(s^0, 2) = -1.1278 \times 10^{13}$ . Therefore, the wealth level threshold is  $\underline{w} = -1482.0$ .

## 6. BACKWARD-INDUCTION METHOD

Algorithm 1 (BackwardInductionOneSwitch) shows our backward-induction method, which is based on the inductive foundation from the previous section and Eq. (10). There are two main differences to the common backward-induction method for solving GMDPs with finite planning horizons [15]. First, the inductive foundation of the backward-induction method for solving GMDPs with finite planning horizons is trivially provided by values that are all zero, while the inductive foundation of our backward-induction method is provided by Theorem 3. Thus, our backward-induction method first determines the wealth level threshold  $\underline{w}$  and, at the same time, the optimal values  $v_{1s}^*(s, w)$  for all states  $s \in S'$  and all wealth levels  $w$  with  $w \leq \underline{w}$ . Second, the backward-induction method for solving GMDPs with finite planning horizons starts at the planning horizon  $t = T$ , then decreases the time step  $t$  with a fixed step size of one, and ends at time step 0, while our backward-induction method starts at the wealth level threshold  $w = \underline{w}$ , then increases the wealth level  $w$  with a variable step size (that is controlled by a priority queue with the wealth level  $w$  as the key), and ends at the initial wealth level  $w_0$ .

We now explain the backward-induction method and show, at the same time, that the MEU<sub>1s</sub>-optimal value functions  $v_{1s}^*(s, \cdot)$  are piecewise one-switch functions with a finite number of segments. The backward-induction method represents the MEU<sub>1s</sub>-optimal value functions (one for each state) and an MEU<sub>1s</sub>-optimal policy as a finite list of tuples  $(w^i(s), v_{\ell}^i(s), v_e^i(s), a^i(s))$ , which represent that  $v_{1s}^*(s, w) = w + v_{\ell}^i(s) + D\gamma^w v_e^i(s)$  and that  $\pi_{1s}^*(s, w) = a^i(s)$  for all wealth levels  $w$  with  $w \in (w^i(s), w^{i+1}(s)]$ . In this case, we say that  $a^i(s)$  is an MEU<sub>1s</sub>-optimal action for wealth level  $w$ . *VAList* is a data structure that contains all of these tuples after the termination of the backward-induction method. *GetValues(VAList, s, w)* retrieves the values  $v_{\ell}^i(s)$  and  $v_e^i(s)$  from the tuple  $(w^i(s), v_{\ell}^i(s), v_e^i(s), a^i(s))$  in *VAList* with wealth level  $w \in (w^i(s), w^{i+1}(s)]$ .

The backward-induction method uses the inductive foundation from the previous section and thus sets  $w^0(s) = -\infty$ ,  $v_{\ell}^0(s) = v_{\ell}^{\pi_e^{**}}(s)$ ,  $v_e^0(s) = v_e^*(s)$  and  $a^0(s) = \pi_e^{**}(s)$  for all states  $s \in S'$  on Lines 1–4. It also initializes the priority queue *PQ*. The wealth level threshold  $\underline{w}$  is the first non-infinity wealth level  $w^i$

### Algorithm 1 Backward-Induction Method

We define  $\text{succ}(s, a) = \{s' \in S \mid a \in A_s, P(s'|s, a) > 0\}$  and  $\text{pred}(s, a) = \{s' \in S \mid a \in A_{s'}, P(s|s', a) > 0\}$ . We use the following operations on priority queues: *Insert(PQ, s, w)* inserts  $s$  into priority queue *PQ* with key  $w$ , *IsMember(PQ, s)* tests whether  $s$  is in *PQ*, *GetKey(PQ, s)* returns the key of  $s$  in *PQ* ( $s$  needs to be in *PQ*), *DecreaseKey(PQ, s, w)* decreases the current key of  $s$  in *PQ* to  $w$  ( $s$  needs to be in *PQ* with a key greater than  $w$ ), and *ExtractMin(PQ)* removes a state with the lowest key from *PQ* and returns both the state and its key.

*VAList* = BackwardInductionOneSwitch( $S, \{A_s\}, P, r, D, \gamma, w_0$ )

```

1: determine  $v_e^*, v_{\ell}^{\pi_e^{**}}$ , and  $\pi_e^{**}$ ;
2: for all  $s \in S'$  do
3:   AddList(VAList,  $s, -\infty, v_{\ell}^{\pi_e^{**}}(s), v_e^*(s), \pi_e^{**}(s)$ );
4:   Insert(PQ,  $s, -\infty$ );
5:  $A \leftarrow \bigcup_{s \in S'} A_s$ ;
6: while  $\neg \text{IsEmpty}(\text{PQ})$  do
7:    $s, w^i \leftarrow \text{ExtractMin}(\text{PQ})$ ;
8:    $v_{1s}^i \leftarrow -\infty$ ;
9:   for all  $a \in A_s$  do
10:     $q_{\ell}^i(s, a), q_e^i(s, a) \leftarrow 0$ ;
11:    for all  $s' \in \text{succ}(s, a)$  do
12:      $v_{\ell}^j(s'), v_e^j(s') \leftarrow \text{GetValues}(\text{VAList}, s', w^i + r(s, a, s'))$ ;
13:      $q_{\ell}^i(s, a) \leftarrow q_{\ell}^i(s, a) + P(s'|s, a) [r(s, a, s') + v_{\ell}^j(s')]$ ;
14:      $q_e^i(s, a) \leftarrow q_e^i(s, a) + P(s'|s, a) \gamma^{r(s, a, s')} v_e^j(s')$ ;
15:     if  $w^i \neq -\infty$  then
16:       $q_{1s}^i(s, a) \leftarrow w^i + q_{\ell}^i(s, a) + D\gamma^{w^i} q_e^i(s, a)$ ;
17:      if  $q_{1s}^i(s, a) > v_{1s}^i$  or  $(q_{1s}^i(s, a) = v_{1s}^i$  and  $q_e^i(s, a) < v_e^i)$  then
18:        $v_{1s}^i \leftarrow q_{1s}^i(s, a)$ ;
19:        $a^i, v_{\ell}^i, v_e^i \leftarrow a, q_{\ell}^i(s, a), q_e^i(s, a)$ ;
20:     if  $w^i \neq -\infty$  then
21:      AddList(VAList,  $s, w^i, v_{\ell}^i, v_e^i, a^i$ );
22:     for all  $a \in A_s$  do
23:      InsertSP(PQ,  $s, w^i, v_{\ell}^i, v_e^i, q_{\ell}^i(s, a), q_e^i(s, a), \gamma, w_0$ );
24:     if  $w^i \neq -\infty$  then
25:      for all  $a \in A$  do
26:       for all  $s' \in \text{pred}(s, a)$  do
27:        InsertNeg(PQ,  $s', w^i - r(s', a, s), w_0$ );

```

*InsertSP(PQ, s, w, v<sub>ℓ</sub>, v<sub>e</sub>, q<sub>ℓ</sub>, q<sub>e</sub>, γ, w<sub>0</sub>)* *InsertNeg(PQ, s, w, w<sub>0</sub>)*

```

1: if  $v_e \neq q_e$  then
2:    $\text{tmp} \leftarrow \frac{q_{\ell} - v_{\ell}}{v_e - q_e}$ ;
3:   if  $\text{tmp} > D\gamma^{w_0}$  then
4:      $\hat{w} \leftarrow \log_{\gamma} \left( \frac{1}{D} \cdot \text{tmp} \right)$ ;
5:     if  $\hat{w} > w$  then
6:       InsertNeg(PQ,  $s, \hat{w}, w_0$ );
1: if  $w < w_0$  then
2:   if IsMember(PQ,  $s$ ) then
3:     if GetKey(PQ,  $s$ ) >  $w$  then
4:       DecreaseKey(PQ,  $s, w$ );
5:     else
6:       Insert(PQ,  $s, w$ );

```

removed from the priority queue in the main loop. The backward-induction method then proceeds in a backward fashion. Consider any state  $s \in S'$  and any segment  $(w^i(s), w^{i+1}(s)]$ , where the wealth level  $w^{i+1}(s)$  is still to be determined. We demand that two properties hold for this segment: First, the same action  $a^i(s)$  has to be MEU<sub>1s</sub>-optimal for all wealth levels  $w \in (w^i(s), w^{i+1}(s)]$  (Property 1). Second, for all wealth levels  $w \in (w^i(s), w^{i+1}(s)]$ , actions  $a \in A_s$  and states  $s' \in S$  with  $P(s'|s, a) > 0$ , there exists a  $j_{a, s'}^{s, i}$  such that  $w^{j_{a, s'}^{s, i} + 1}(s') \leq w^i(s)$  and  $w + r(s, a, s') \in (w^{j_{a, s'}^{s, i}}(s'), w^{j_{a, s'}^{s, i} + 1}(s'))$ , that is, all possible wealth levels after the action execution should be in the same segment (Property 2). Therefore, we have for all wealth levels  $w \in (w^i(s), w^{i+1}(s)]$  according to Eq. (10) and Eq. (11)

$$\begin{aligned} v_{1s}^*(s, w) &= \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) v_{1s}^*(s', w + r(s, a, s')) \\ &= \max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) [w + r(s, a, s') + v_{\ell}^{j_{a, s'}^{s, i}}(s') \\ &\quad + D\gamma^{w + r(s, a, s')} v_e^{j_{a, s'}^{s, i}}(s')] \end{aligned}$$

$$\begin{aligned}
&= \max_{a \in A_s} \left[ w + \sum_{s' \in S} P(s'|s, a) \left[ r(s, a, s') + v_{\ell}^{j^{s,i}, i}(s') \right] \right. \\
&\quad \left. + D\gamma^w \sum_{s' \in S} P(s'|s, a) \gamma^{r(s, a, s')} v_{\ell}^{j^{s,i}, i}(s') \right] \\
&= \max_{a \in A_s} \left[ w + q_{\ell}^i(s, a) + D\gamma^w q_{\ell}^i(s, a) \right], \tag{13}
\end{aligned}$$

where we defined the values  $q_{\ell}^i(s, a)$  and  $q_{\ell}^i(s, a)$  in the last step for use below.

First, the backward-induction method calculates action  $a^i(s)$  on Lines 8–21. Since the optimal value function  $v_{1s}^*(s, \cdot)$  is continuous in the wealth level  $w$  and action  $a^i(s)$  is  $\text{MEU}_{1s}$ -optimal for all wealth levels  $w \in (w^i(s), w^{i+1}(s))$ , it is also  $\text{MEU}_{1s}$ -optimal for wealth level  $w^i(s)$ , which implies according to Eq. (13)

$$a^i(s) \in \hat{A}(s) = \arg \max_{a \in A_s} \left[ w^i(s) + q_{\ell}^i(s, a) + D\gamma^{w^i(s)} q_{\ell}^i(s, a) \right],$$

where we defined the action sets  $\hat{A}(s)$  for use below. The action sets  $\hat{A}(s)$  can contain actions that are not  $\text{MEU}_{1s}$ -optimal for all wealth levels  $w \in (w^i(s), w^{i+1}(s))$ . Figure 3(b) illustrates such actions. The action set  $\hat{A}(s^0)$  includes actions 2, 3, 6 (the blue, black, and cyan graphs, respectively) for the segment  $(w^i(s^0), w^{i+1}(s^0))$ , all of which intersect at the same point. Only action 2 is  $\text{MEU}_{1s}$ -optimal for this segment, action 3 is  $\text{MEU}_{1s}$ -optimal for the previous segment, and action 6 is  $\text{MEU}_{1s}$ -optimal for neither segment. In general, the spurious actions can be eliminated by comparing the actions in the action set  $\hat{A}(s)$  for wealth level  $w^i(s) + \delta$ , where  $\delta$  is a positive infinitesimal. Consider any actions  $a, a' \in \hat{A}(s)$ . Then, we have

$$\begin{aligned}
&w^i(s) + q_{\ell}^i(s, a) + D\gamma^{w^i(s)} q_{\ell}^i(s, a) \\
&= w^i(s) + q_{\ell}^i(s, a') + D\gamma^{w^i(s)} q_{\ell}^i(s, a'). \tag{14}
\end{aligned}$$

Assume that action  $a$  is  $\text{MEU}_{1s}$ -optimal for all wealth levels  $w \in (w^i(s), w^{i+1}(s))$ , whereas action  $a'$  is not. Then, action  $a$  is also  $\text{MEU}_{1s}$ -optimal for wealth level  $w^i(s) + \delta$ , whereas action  $a'$  is not. Thus, we have

$$\begin{aligned}
&w^i(s) + \delta + q_{\ell}^i(s, a) + D\gamma^{w^i(s)+\delta} q_{\ell}^i(s, a) \\
&> w^i(s) + \delta + q_{\ell}^i(s, a') + D\gamma^{w^i(s)+\delta} q_{\ell}^i(s, a').
\end{aligned}$$

By subtracting Eq. (14) from the inequality, we have

$$\begin{aligned}
&\delta + D\gamma^{w^i(s)} \left[ \gamma^{\delta} - 1 \right] q_{\ell}^i(s, a) > \delta + D\gamma^{w^i(s)} \left[ \gamma^{\delta} - 1 \right] q_{\ell}^i(s, a') \\
&q_{\ell}^i(s, a) < q_{\ell}^i(s, a'),
\end{aligned}$$

Thus, the backward-induction method chooses any action  $a^i(s) \in \arg \min_{a \in \hat{A}(s)} q_{\ell}^i(s, a)$ .

Second, the backward-induction method calculates the values  $v_{\ell}^i(s) = q_{\ell}^i(s, a^i(s))$  and  $v_{\ell}^i(s) = q_{\ell}^i(s, a^i(s))$  on Lines 8–21 since  $v_{1s}^*(s, w) = w + q_{\ell}^i(s, a^i(s)) + D\gamma^w q_{\ell}^i(s, a^i(s))$  according to Eq. (13) and  $v_{1s}^*(s, w) = w + v_{\ell}^i(s) + D\gamma^w v_{\ell}^i(s)$  according to our representation of the piecewise one-switch functions for all wealth levels  $w \in (w^i(s), w^{i+1}(s))$ .

Third, the backward-induction method calculates the wealth levels  $w^{i+1}(s)$ . Property 1 implies that action  $a^i(s)$  is  $\text{MEU}_{1s}$ -optimal for wealth level  $w^{i+1}(s)$ . We thus have for all actions  $a \in A_s$

$$\begin{aligned}
&w^{i+1}(s) + q_{\ell}^i(s, a^i(s)) + D\gamma^{w^{i+1}(s)} q_{\ell}^i(s, a^i(s)) \\
&\geq w^{i+1}(s) + q_{\ell}^i(s, a) + D\gamma^{w^{i+1}(s)} q_{\ell}^i(s, a) \\
&q_{\ell}^i(s, a^i(s)) + D\gamma^{w^{i+1}(s)} q_{\ell}^i(s, a^i(s))
\end{aligned}$$

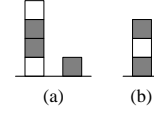


Figure 4: Painted Blocksworld Problem

$$\begin{aligned}
&\geq q_{\ell}^i(s, a) + D\gamma^{w^{i+1}(s)} q_{\ell}^i(s, a) \\
&v_{\ell}^i(s) + D\gamma^{w^{i+1}(s)} v_{\ell}^i(s) \\
&\geq q_{\ell}^i(s, a) + D\gamma^{w^{i+1}(s)} q_{\ell}^i(s, a) \\
&w^{i+1}(s) \leq \log_{\gamma} \left( \frac{1}{D} \cdot \frac{q_{\ell}^i(s, a) - v_{\ell}^i(s)}{v_{\ell}^i(s) - q_{\ell}^i(s, a)} \right),
\end{aligned}$$

where the right-hand side of the last inequality defines a potential switching point for the  $\text{MEU}_{1s}$ -optimal action and thus a potential value for the wealth level  $w^{i+1}(s)$  if the argument of the logarithm is positive and the logarithm thus is defined, as illustrated in Figure 3(b). The determination of these potential switching points is essentially the same as the determination of the wealth level threshold (with the only difference being whether  $w^i = -\infty$ ), which allows the backward-induction method to combine their calculation in the main loop on Lines 22–23 using `InsertSP`. Property 2 implies that  $w + r(s, a, s') \leq w^{j^{s,i}, i+1}(s')$  and thus that  $w \leq w^{j^{s,i}, i+1}(s') - r(s, a, s')$ , where the right-hand side of this inequality defines another potential switching point for the  $\text{MEU}_{1s}$ -optimal action and thus another potential value for the wealth level  $w^{i+1}(s)$ , as illustrated in Figure 3(c) where wealth level  $w^i$  takes the role of wealth level  $w^{j^{s,i}, i+1}$  and reward  $c_k$  takes the role of reward  $r(s, a, s')$ . The backward-induction method calculates the potential switching points on Lines 25–27 using `InsertNeg`.

The backward-induction method stores the potential switching points in the priority queue  $PQ$  and processes them in order of increasing wealth levels since the values of segment  $(w^i(s), w^{i+1}(s))$  depend on the values of segments  $(w^j(s'), w^{j+1}(s'))$  for one or more  $j$  with  $w^{j+1}(s') \leq w^i(s)$ , which need to have been calculated already. The backward-induction method terminates when the priority queue  $PQ$  is empty, which happens in a finite amount of time since the wealth level threshold  $\underline{w}$  is a finite negative value and the values of the processed switching points are monotonically increasing by at least a positive constant and guaranteed to be non-positive [7]. Thus, the  $\text{MEU}_{1s}$ -optimal value functions are indeed piecewise one-switch functions with a finite number of segments. For the termite GMDMP, the backward-induction method finds the following  $\text{MEU}_{1s}$ -optimal policy for the initial wealth level  $w_0 = 0$ :

$$\pi_{1s}^*(s^0, w) = \begin{cases} 1 & w \in (-316.4, 0] \\ 2 & w \in (-1482.0, -316.4] \\ 3 & w \in (-\infty, -1482.0] \end{cases}$$

## 7. EXAMPLE

We use the painted blocksworld problem from [9] to illustrate risk-sensitive planning with one-switch utility functions. 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 (M) has a reward of  $-1$ , and the paint action (P) has a reward of  $-3$ . Figure 4(a) shows the initial state. The goal is to build a stack of three blocks as shown in Figure 4(b). The painted

