Representations of Decision-Theoretic Planning Tasks

Sven Koenig and **Yaxin Liu**
College of Computing, Georgia Institute of Technology
Atlanta, Georgia 30332-0280
{skoenig, yxliu}@cc.gatech.edu

Abstract

Goal-directed Markov Decision Process models (GDMDPs) are good models for many decision-theoretic planning tasks. They have been used in conjunction with two different reward structures, namely the goal-reward representation and the action-penalty representation. We apply GDMDPs to planning tasks in the presence of traps such as steep slopes for outdoor robots or staircases for indoor robots, and study the differences between the two reward structures. In these situations, achieving the goal is often the primary objective while minimizing the travel time is only of secondary importance. We show that the action-penalty representation without discounting guarantees that the optimal plan achieves the goal for sure (if this is possible) but neither the action-penalty representation with discounting nor the goal-reward representation with discounting have this property. We then show exactly when this trapping phenomenon occurs, using a novel interpretation for discounting, namely that it models agents that use convex exponential utility functions and thus are optimistic in the face of uncertainty. Finally, we show how the trapping phenomenon can be eliminated with our Selective State-Deletion Method.

Introduction

Representations of planning tasks are studied in the planning literature often in the context of operator representations that ensure a good trade-off between being able to represent a wide range of planning tasks and being able to solve them efficiently. Decision theory provides a formal framework for choosing optimal plans from a set of viable plans. Therefore, in the context of decision-theoretic planning, it is also important to study how representations of planning tasks affect which plans are optimal. We study two different reward structures that have often been used in the decision-theoretic planning and reinforcement-learning literature[1]. The action-penalty representation penalizes the agent for every action that it executes, but does not reward or penalize it for stopping in a goal state. The goal-reward representation, on the other hand, rewards the agent for stopping in a goal state, but does not reward or penalize it for executing actions. We study what the two representations have in common and how they differ, by combining ideas from artificial intelligence planning, operations research, and utility theory and using robot-navigation tasks as examples.

We show that the two representations are equivalent if the discount factor (that is necessary for the goal-reward representation) is close to one. In actual implementations, however, the discount factor cannot be set arbitrarily closely to one. We show that, in this case, the action-penalty representation without discounting guarantees that the optimal plan achieves the goal for sure (if this is possible) but neither the action-penalty representation with discounting nor the goal-reward representation with discounting have this property. Thus, although it is often convenient to use discounting, it cannot be used to solve planning tasks for which achieving the goal is the primary objective while minimizing the execution cost is only of secondary importance. Planning tasks with this lexicographic preference ordering often include planning for robot navigation in the presence of traps such as steep slopes for outdoor robots or staircases for indoor robots. We show exactly when this trapping phenomenon occurs. As part of our analysis, we show that the expected total discounted reward of any plan for the goal-reward representation equals its expected total undiscounted *utility* for the action-penalty representation with a convex exponential utility function, that maps (total) rewards to (total) utilities. This result provides a novel interpretation for discounting in the context of the goal-reward representation, namely that it models agents that are optimistic in the face of uncertainty and thus do not avoid traps at all cost. It relates planning with the goal-reward representation to planning with exponential utility functions, that has been studied in the planning literature in the context of planning in high-stake decision situations, such as managing environmental crisis situations (Koenig and Simmons 1994; Koenig 1998), and enables us to use insights from utility theory to improve our understanding of planning with the goal-reward representation. Finally, we show how the trapping phenomenon can be eliminated with our Selective State-Deletion Method.

Representing Planning Tasks as GDMDPs

Goal-directed Markov decision process models (GDMDPs) are convenient and commonly used models of decision-theoretic planning tasks (Boutilier *et al.* 1999). GDMDPs are totally observable Markov decision process models with goal states in which execution stops. They consist of

- a finite set of states S;
- a start state $s_{start} \in S$;

[1] Reinforcement learning interleaves decision-theoretic planning, plan execution, and parameter estimation (Barto *et al.* 1989; Kaelbling *et al.* 1996). For the purpose of this paper, reinforcement learning can be treated as an on-line version of decision-theoretic planning.

- a set of goal states $G \subseteq S$;
- a finite set of actions $A(s) \neq \emptyset$ for each non-goal state s that can be executed in state s;
- a transition probability $p(s'|s,a)$ and a real-valued immediate reward $r(s,a,s')$ for each non-goal state s, state s', and action a that can be executed in state s, where $p(s'|s,a)$ denotes the probability that the agent transitions to state s' after it executed action a in state s, and $r(s,a,s')$ denotes the immediate reward that the agent receives for the transition;
- a real-valued goal reward $g(s)$ for each goal state s, where $g(s)$ denotes the goal reward that the agent receives whenever it is in state s and thus stops the plan execution.

The agent starts in the start state and selects actions for execution according to a given plan. We define plans to be mappings from non-goal states to actions that can be executed in those states, also known as "stationary, deterministic policies." Although the term "policy" originated in the field of stochastic dynamic programming, similar schemes have been proposed in the context of artificial intelligence planning, including universal plans (Schoppers 1987). The agent always executes the action that the plan assigns to its current state. It then receives the corresponding immediate reward and transitions to one of the successor states according to the corresponding transition probabilities. The agent always stops in goal states (but not otherwise), in which case it receives the goal reward and then does not receive any further rewards.

Two Representations of Planning Tasks

When formulating decision-theoretic planning tasks as GDMDPs, one has to decide on the (positive or negative) immediate rewards that the agent receives after every action execution. If the agent receives immediate reward r_t during the $(t+1)$st action execution and reaches goal state s after n action executions, then its total reward is $\sum_{t=0}^{n-1}[\gamma^t r_t] + \gamma^n g(s)$, and the agent wants to maximize the expectation of this quantity. The discount factor $0 < \gamma \leq 1$ specifies the relative value of an immediate reward received after t action executions compared to the same immediate reward received one action execution earlier. If the discount factor is one, the total reward is called undiscounted, otherwise it is called discounted. Two different representations are often used in conjunction with GDMDPs, namely the action-penalty representation and the goal-reward representation (Koenig and Simmons 1996).

- The **action-penalty representation** penalizes the agent for every action that it executes, but does not reward or penalize it for stopping in a goal state. Formally, $r(s,a,s') = -1$ for each non-goal state s, state s', and action a that can be executed in state s, and $g(s) = 0$ for each goal state s. The planning objective then is to maximize the expected total undiscounted reward (or, equivalently, minimize the expected total undiscounted execution cost). The agent attempts to reach a goal state with as few action executions as possible to minimize the amount of penalty that it receives. The action-penalty representation has been used in (Barto *et al.* 1989; Barto *et al.* 1995; Dean *et al.* 1995) in the context of robot navigation tasks, among others.
- The **goal-reward representation** rewards the agent for stopping in a goal state, but does not reward or penalize it for executing actions. Formally, $r(s,a,s') = 0$ for each non-goal state s, state s', and action a that can be executed in state s, and $g(s) = 1$ for each goal state s. The planning objective then is to maximize the expected total discounted reward. Discounting is necessary with the goal-reward representation. Otherwise the agent would always receive a total undiscounted reward of one if it achieves the goal, and the agent could not distinguish among paths of different lengths. If discounting is used, then the goal reward gets discounted with every action execution, and the agent attempts to reach a goal state with as few action executions as possible to maximize the portion of the goal reward that it receives. The goal-reward representation has been used in (Sutton 1990; Whitehead 1991; Peng and Williams 1992; Thrun 1992; Lin 1993) in the context of robot-navigation tasks, among others.

Both representations have often been used in the decision-theoretic planning literature. It is easy to show that maximizing the expected total discounted or undiscounted reward for the action-penalty representation and maximizing the expected total discounted reward for the goal-reward representation are equivalent both for deterministic planning tasks and for decision-theoretic planning tasks where the discount factor approaches one (under appropriate assumptions). In actual implementations of decision-theoretic planning methods, however, the discount factor cannot be set arbitrarily close to one because, for example, the arithmetic precision is not sufficiently good, convergence is too slow (Kaelbling *et al.* 1996), or the expected total discounted rewards are systematically overestimated when function approximators are used (Thrun and Schwartz 1993). In this case, the various representations are not necessarily equivalent. We therefore study what the representations have in common and how they differ if the discount factor is not sufficiently close to one, starting with the goal-reward representation with discounting and the action-penalty representation without discounting.

The Trapping Phenomenon for the Goal-Reward Representation

There is an important difference between the goal-reward representation with discounting and the action-penalty representation without discounting. We say that a plan achieves the goal if the probability with which an agent starting in the start state reaches a goal state within a given number of action executions approaches one as this bound approaches infinity, otherwise the plan does not achieve the goal. The action-penalty representation has the desirable property that every plan that maximizes the expected total undiscounted reward for the action-penalty representation also achieves

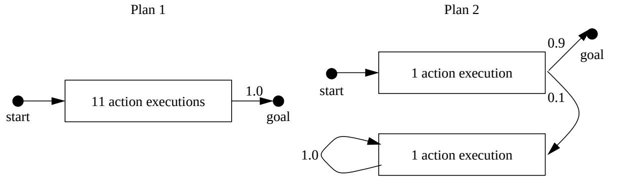

Figure 1: Two Plans that Illustrate the Trapping Phenomenon

the goal provided that the goal can be achieved, as the following theorem shows.

Theorem 1 *Every plan that maximizes the expected total undiscounted reward for the action-penalty representation achieves the goal provided that the goal can be achieved.*

Proof: Every plan that achieves the goal has a finite expected total undiscounted reward for the action-penalty representation. This is so, because plans assign actions to states. Let S' be the set of states that can be reached with positive probability during the execution of a given plan that achieves the goal, and y_s be the largest total undiscounted reward (that is, the non-positive total undiscounted reward closest to zero) that the agent can receive with positive probability when it starts in state $s \in S'$ and always selects the action for execution that the plan assigns to its current state. It holds that $y_s > -\infty$, since the plan achieves the goal and all immediate rewards are negative but finite. Let $p_s > 0$ be the probability that the agent receives this total undiscounted reward. Then, a lower bound on the expected total undiscounted reward of the plan is $(\min_{s \in S'} y_s)/(\min_{s \in S'} p_s) > -\infty$. On the other hand, every plan that does not achieve the goal has an expected total undiscounted reward that is minus infinity. This is so because the total undiscounted reward of every trajectory (that is, specification of the states of the world over time, representing one possible course of execution of the plan) is non-positive, and a total undiscounted reward of minus infinity is obtained with positive probability (since all immediate rewards are negative and plan execution does not reach a goal state with positive probability). ■

We now show by example that a plan that maximizes the expected total discounted reward for the goal-reward representation does not necessarily achieve the goal even if the goal can be achieved. We call this phenomenon the trapping phenomenon. Consider the following simple example, see Figure 1. Plan 1 always achieves the goal with 11 action executions. With probability 0.900, plan 2 achieves the goal with one action execution. With the complementary probability, plan 2 cycles forever and thus does not achieve the goal. Assume that we use the goal-reward representation with discount factor $\gamma = 0.900$. Then, plan 1 has a total discounted reward of $0.900^{11} = 0.3138$, and plan 2 has an expected total discounted reward of $0.900 \times 0.900^1 + 0.100 \times 0.900^\infty = 0.8100$. Thus, plan 2 is better than plan 1, but does not achieve the goal. In the next section, we present a more realistic example that shows that this can happen even if the discount factor is 0.999 and thus very close to one.

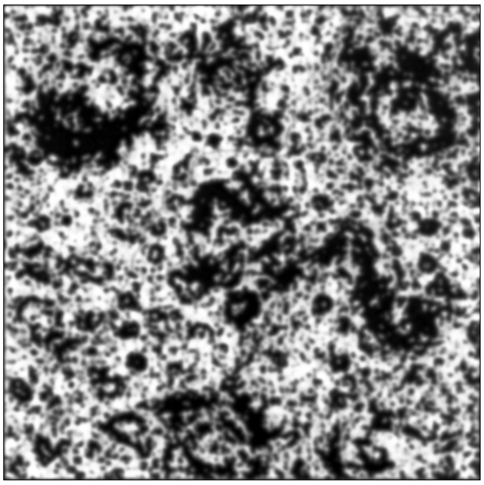

Figure 2: Terrain for the Navigation Task

Example: Robot-Navigation Tasks

To illustrate the trapping phenomenon, we use robot-navigation tasks in a known terrain. The task of the robot is to reach a given goal location from its start location. Movement of the robot is noisy but the robot observes its location at regular intervals with certainty, using a global positioning system. Following (Dean *et al.* 1995), we discretize the locations and model the robot-navigation tasks as GDMDPs. The states of the GDMDPs correspond to the locations. The actions of the GDMDPs correspond to moving in the four main compass directions. Figure 2 shows the terrain that the robot operates in, taken from (Stentz 1995). We assume that the darker the terrain, the higher it is. For simplicity, we distinguish only three elevation levels and discretize the terrain into 22x22 square locations. Figure 3 shows the resulting grid-world. The task of the robot is to navigate from the upper left location (A1) to the lower right location (V22). The robot can always move to any of its four neighboring locations but its movement is noisy since the robot can fail to move or stray off from its nominal direction by one square to the left or right due to movement noise and not facing precisely in the right direction. For example, if the robot is in location C2 and moves north, it can end up in location C2, B1, B2, or B3. The higher the elevation of the current location of the robot with respect to its intended destination, the less likely it is to stray off or slide back to its current location. However, if the elevation of its current location is high and the elevation of the location that it actually moves to is low, it can tip over. In this case, no matter in which direction it attempts to move subsequently, its wheels always spin in the air and it is thus no longer able to move to the goal location. The actions that can lead to the robot tipping over immediately ("hazardous actions") are indicated by arrows in Figure 3. They lead to a state that the robot cannot leave again. The transition probabilities are explained in more

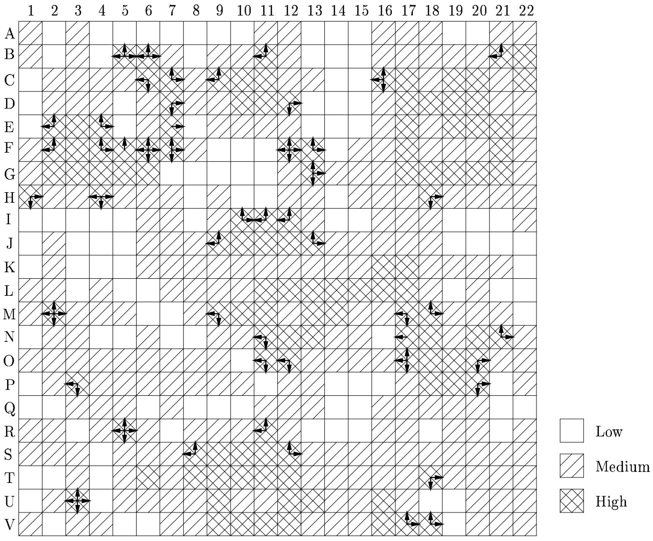

Figure 3: The Discretized Grid and Hazardous Actions (Patterns Indicate Different Elevations)

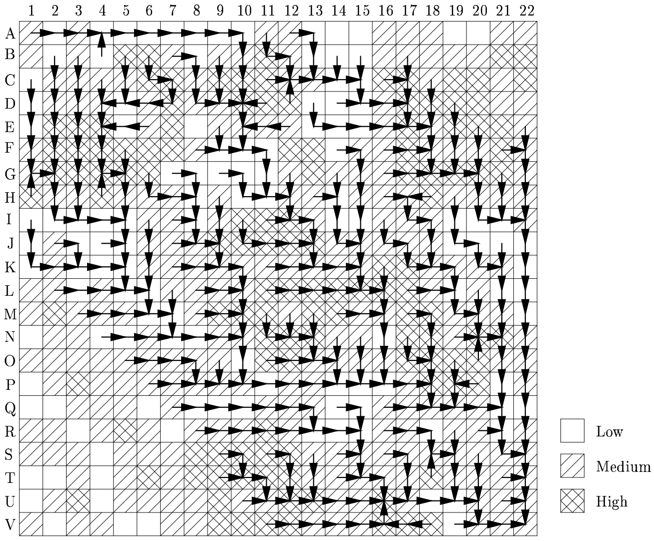

Figure 4: The Optimal Policy for the Action-Penalty Representation with Discount Factor 1.000

detail in the appendix. As expected, the optimal plan for the action-penalty representation avoids all of these actions and thus ensures that the robot does not tip over, see Figure 4. (The figure contains only the actions that the robot executes with positive probability.) However, the optimal plan for the goal-reward representation does not avoid all hazardous actions. This is true if the discount factor is 0.900, see Figure 5, and remains true even if the discount factor is 0.999 and thus very close to one, see Figure 6. (The hazardous actions that the robot executes with positive probability are circled in the figures.)

Relating the Representations

We now explain why the optimal plan for the goal-reward representation with discounting is not guaranteed to achieve the goal for sure. Discounting was originally motivated by collecting interest for resources. The discount factor γ can be interpreted as modeling agents that save or borrow resources at interest rate $(1-\gamma)/\gamma$. This interpretation can be relevant for money that agents save and borrow, but most agents cannot invest their resources and earn interest. For example, time often cannot be saved or invested. Sometimes the discount factor is also interpreted as the probability that agents do not die during each action execution. Often, however, the discount factor is only used as a mathematical convenience (because the expected total discounted reward of every plan is guaranteed to be finite). In the following, we provide a novel interpretation of discounting, namely that it models agents whose risk attitudes can be described using convex exponential utility functions, which implies that the agents are optimistic (risk seeking) in the face of uncertainty and thus do not avoid traps at all cost. The discount factor determines the shape of the utility function and thus the amount of optimism of the agents.

Our interpretation is based on the expected total undiscounted utility of plans. If the execution of a plan leads with probabilities p_i to total undiscounted rewards r_i, then its expected total undiscounted utility is $\sum_i [p_i u(r_i)]$ and its certainty equivalent is $u^{-1}(\sum_i [p_i u(r_i)])$, where u is a monotonically increasing utility function that maps (total) rewards r to their (total) utilities $u(r)$ (Bernoulli 1738; von Neumann and Morgenstern 1947). We show that the expected total discounted reward of any plan for the goal-reward representation equals its expected total undiscounted utility for the action-penalty representation and a convex exponential utility function. Convex exponential utility functions have the form $u(r) = \gamma^{-r}$ for $0 < \gamma < 1$. The following theorem relates the two representations.

Theorem 2 *Every plan that maximizes the expected total discounted reward for the goal-reward representation and discount factor γ also maximizes the expected total undiscounted utility for the action-penalty representation and the convex exponential utility function $u(r) = \gamma^{-r}$ ($0 < \gamma < 1$), and vice versa.*

Proof: Consider an arbitrary plan and any of its trajectory. If the trajectory needs i action executions to achieve the goal (i can be finite or infinite), then its discounted total reward for the goal-reward representation is γ^i. Its undiscounted total reward for the action-penalty representation is $-i$, and its total undiscounted utility is γ^i. This shows that the discounted total reward of every trajectory for the goal-reward representation equals its total undiscounted utility for the action-penalty representation. This means that the expected total discounted reward of every plan for the goal-reward representation equals its expected total undiscounted utility for the action-penalty representation. ■

Utility theory states that agents should select the plan for execution that maximizes the expected total undiscounted utility. Exponential utility functions are perhaps the most of-

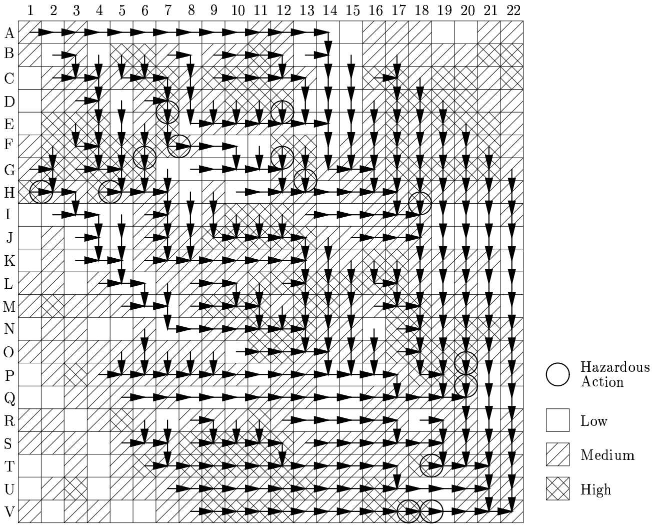

Figure 5: The Optimal Policy for the Goal-Reward Representation with Discount Factor 0.900

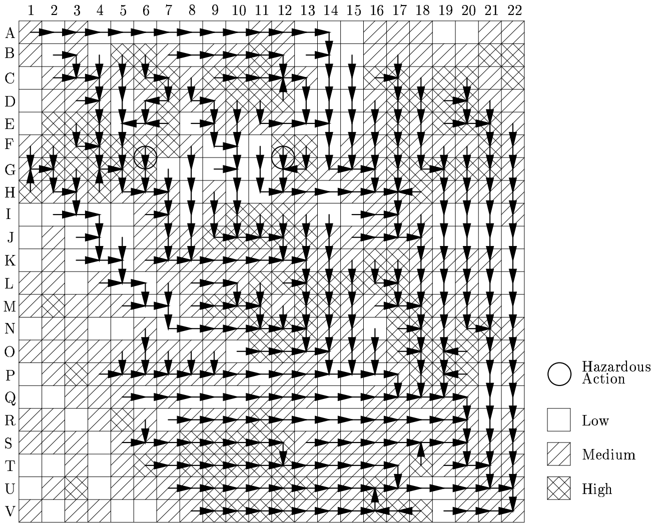

Figure 6: The Optimal Policy for the Goal-Reward Representation with Discount Factor 0.999

ten used utility functions in utility theory (Watson and Buede 1987) and specialized assessment procedures are available that make it easy to elicit them from human decision makers (Farquhar 1984; Farquhar and Nakamura 1988). Convex exponential utility functions can express a continuum of optimistic risk attitudes. Optimistic (or, synonymously, risk-seeking) agents focus more on the best-case outcomes than the worst-case outcomes. From the theory of risk-sensitive Markov decision processes (Marcus *et al.* 1997) we know that, given that the execution of a plan leads with positive probability p_i to a trajectory with finite total undiscounted reward r_i (for all i), the certainty equivalent of any plan for the convex exponential utility function $u(r) = \gamma^{-r}$ $(0 < \gamma < 1)$ approaches its best-case total undiscounted reward as γ approaches zero (under appropriate assumptions). Thus, the agent becomes more and more optimistic as γ approaches zero. From the theory of risk-sensitive Markov decision processes we also know that, given that the execution of a plan leads with positive probability p_i to a trajectory with finite reward r_i (for all i), the certainty equivalent of any plan for the convex exponential utility function $u(r) = \gamma^{-r}$ $(0 < \gamma < 1)$ approaches the expected total undiscounted reward as γ approaches one (under appropriate assumptions). Thus, the agent becomes more and more risk-neutral as γ approaches one. Values of γ between zero and one trade off between maximizing the best-case and expected total undiscounted reward. Consequently, Theorem 2 relates the discount factor of the goal-reward representation with discounting to the parameter of convex exponential utility functions that expresses how optimistic an agent is. The smaller the discount factor, the more optimistic the agent and thus the more it pays attention to the outcomes in the best case, not the outcomes in the worst case. Thus, it is more likely to get trapped. For example, Figure 7 contains a log-log plot that shows for the robot-navigation example how the probability of tipping over while executing the plan that maximizes the expected total discounted reward for the goal-reward representation depends on the discount factor (ties among optimal plans are broken randomly). The figure confirms that the smaller the discount factor, the more pronounced the trapping phenomenon.

Eliminating the Trapping Phenomenon

So far, we have shown that a plan that maximizes the expected total discounted reward for the goal-reward representation does not necessarily achieve the goal even if the goal can be achieved. While it can be rational or even necessary to trade off a smaller probability of not achieving the goal and a smaller number of action executions in case the goal is achieved (for example, if the goal cannot be achieved with probability one), this is a problem when solving planning tasks for which goal achievement is the primary objective and cost minimization is only of secondary importance. For example, when solving robot-navigation tasks one often wants to rule out plans that risk the destruction of the robot such as getting too close to steep slopes for outdoor robots or staircases for indoor robots, as convincingly argued in (Dean *et al.* 1995). We now explain how to avoid the trapping phenomenon in these cases.

According to Theorem 1, one way of avoiding the trapping phenomenon is to use the action-penalty representation without discounting. This is not always an option. Reinforcement-learning researchers, for example, often prefer the goal-reward representation over the action-penalty representation because it fits the reinforcement-learning framework better. The goal-reward representation rewards an agent for completing a task, whereas the action-penalty representation neither rewards the agent for good results nor penalizes it for bad results of a behavior. One can avoid the trapping phenomenon for the goal-reward representation with discounting by choosing a plan that maximizes

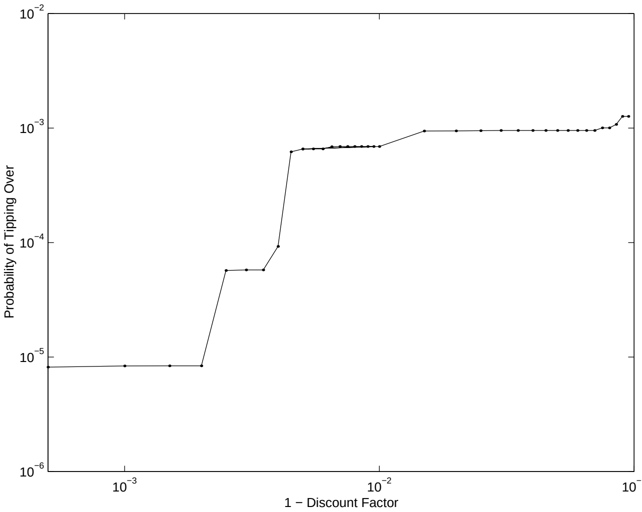

Figure 7: Probability of Tipping over when Following an Optimal Policy for the Goal-Reward Representation

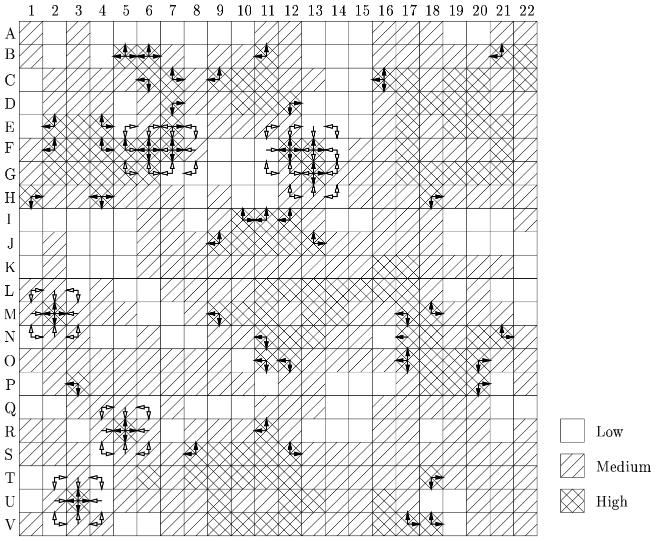

Figure 8: Actions Deleted after Selective State Deletion

the expected discounted total reward *among all plans that achieve the goal*. Since the number of these plans can be exponential in the number of states, we cannot enumerate all of them and thus have to investigate how one can use dynamic programming techniques instead. In the following, we show that one can reduce the problem of finding a plan that maximizes the expected discounted total reward among all plans that achieve the goal to a problem that we know how to solve, namely the problem of finding a plan that maximizes the expected discounted total reward. We say that the goal can be achieved from state s if a plan exists that achieves the goal when its execution starts in state s. We call all other states traps. We then use the following property.

Theorem 3 *Every plan that maximizes the expected total undiscounted utility for the action-penalty representation and the convex exponential utility function $u(r) = \gamma^{-r}$ ($0 < \gamma < 1$) achieves the goal provided that the goal can be achieved from all states.*

Proof by contradiction: Suppose that there exists a plan $plan$ that maximizes the expected total undiscounted utility for the action-penalty representation but does not achieve the goal. Since the goal can be achieved from all states, there must be some state s that is reached with positive probability during the execution of plan $plan$ such that plan $plan$ reaches a goal state with probability zero from state s, but there exists a plan $plan'$ that reaches a goal state with positive probability from state s, and both plans differ only in the action assigned to state s. To see this, consider the set of all states that are reached with positive probability during the execution of plan $plan$ and from which plan $plan$ reaches a goal state with probability zero. At least one such state exists and all of them are non-goal states. The statement then follows for one of these states, which we called s, since the goal can be achieved from all of those states. Now consider all trajectory of plan $plan$ that do not contain state s. Plan $plan'$ has the same trajectories with the same probabilities and total undiscounted utilities. The total undiscounted rewards of all trajectories of plan $plan$ that contain state s are minus infinity (since all immediate rewards are negative and the trajectories contain infinitely many actions) and thus their total undiscounted utilities are zero. On the other hand, at least one trajectory of plan $plan'$ that contains state s achieves the goal. Its probability is positive, its total undiscounted reward is finite, and its total undiscounted utility is positive. The total undiscounted utilities of the other trajectories of plan $plan'$ that contain state s are nonnegative. Therefore, the expected total undiscounted utility of plan $plan'$ is larger than the expected total undiscounted utility of plan $plan$. This, however, is a contradiction. ■

Corollary 1 *Every plan that maximizes the expected total discounted reward for the goal-reward representation also achieves the goal provided that the goal can be achieved from all states.*

To summarize, if there are no traps and the goal can thus be achieved from all states, then every plan that maximizes the expected total discounted reward for the goal-reward representation necessarily also achieves the goal. If the goal cannot be achieved from all states, one can delete all traps from the state space and all actions whose execution can lead to these states. The states and actions can be deleted, for example, with our Selective State-Deletion Method, that discovers the communicating structure of a GDMDP (Puterman 1994) to identify the states and actions that it needs to delete. In the following, we describe a simple but easy-to-understand version of the Selective State-Deletion Method for didactical reasons.

Selective State-Deletion Method

Repeat the following steps until no states have been

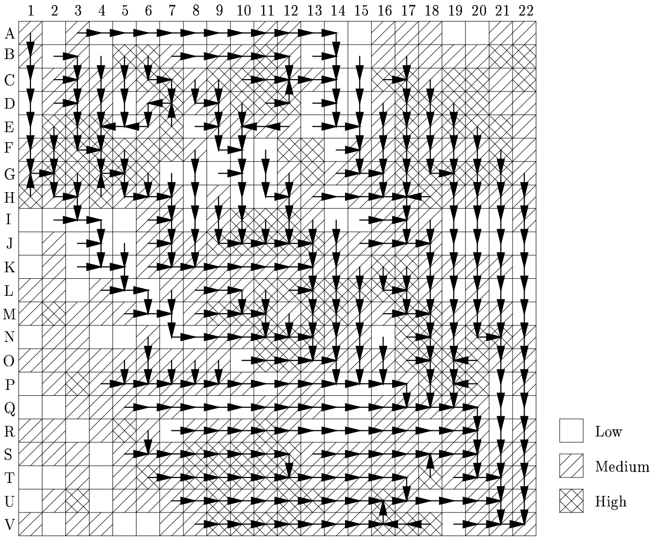

Figure 9: The Optimal Policy for the Goal-Reward Representation with Discount Factor 0.900 after Selective State Deletion

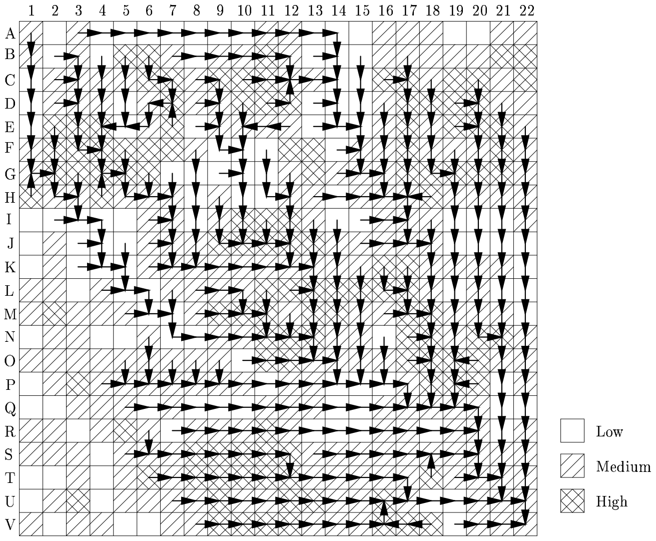

Figure 10: The Optimal Policy for the Goal-Reward Representation with Discount Factor 0.999 after Selective State Deletion

deleted during an iteration:

1. Construct the graph whose vertices are the states in the state space and that has a directed edge from vertex s to vertex s' if there exists an action that can be executed in state s and leads to state s' with positive probability.
2. Use standard methods from graph theory (Corman *et al.* 1990) to determine the strongly connected components of the graph.
3. Delete all states from the state space that are included in leaf components that do not contain vertices that correspond to goal states.
4. Delete all actions that lead to deleted states with positive probability.

The resulting state space is the same as the original state space except that all traps and all actions whose execution can lead to these states have been removed. The deletion of these states and actions does not eliminate any plan that achieves the goal, and it does not change the expected total discounted rewards of the unaffected plans. It only eliminates some of the plans that do not achieve the goal. It does not eliminate all such plans (for example it does not necessarily eliminate an action that does not change a state if there is another action that can be executed in the same state and leaves it with positive probability), and thus Corollary 1 is non-trivial. However, every plan that maximizes the expected total discounted reward among the unaffected plans necessarily also achieves the goal, and consequently also maximizes the expected total discounted reward among all plans that achieve the goal before the deletion of the states and actions. This allows one to use the goal-reward representation and determine a plan that maximizes the expected total discounted reward among all plans that achieve the goal, without any changes to the planning method itself. If there is no plan that achieves the goal after the Selective State-Deletion Method has been applied, then there was no such plan before the Selective State-Deletion Methods was applied either, and the planning task has to be solved with a preference model that trades off the probability of achieving the goal and the execution cost of a plan, and thus gives up on the goal-oriented preference model of traditional artificial intelligence planning (Wellman 1990).

As an example, we use the Selective State-Deletion Method to determine a plan for the robot-navigation example that maximizes the expected total discounted reward for the goal-reward representation among all plans that avoid the hazardous actions. Figure 8 shows the actions that the Selective State-Deletion Method deletes. This includes all hazardous actions (filled arrows), see Figure 3, and all actions after whose execution the robot is eventually forced to execute a hazardous action (hollow arrows). Figures 9 and 10 show the resulting plans that avoid the hazardous actions and thus ensure that the robot does not tip over.

The Trapping Phenomenon for the Action-Penalty Representation

Discounting has also been used in the decision-theoretic planning literature in conjunction with the action-penalty representation. While it is not necessary to use discounting in this context, it is often convenient because then the expected total discounted reward of every plan is guaranteed to be finite. Unfortunately, we now show by example that a plan that maximizes the expected total discounted reward for the action-penalty representation does not necessarily achieve the goal even if the goal can be achieved. Thus, the trapping phenomenon can occur for the action-penalty representation with discounting. Consider again the simple example from Figure 1 and assume that we use the action-penalty representation with discount factor $\gamma = 0.900$. Then, plan 1 has a total discounted reward of $-(1-0.900^{11})/(1-0.900) = -6.8619$, and plan 2 has an expected total discounted reward of $-0.900\times 1-0.100\times 1/(1-0.900) = -1.9000$. Thus, plan 2 is better than plan 1, but does not achieve the goal. It is easy to show that the optimal plan for the action-penalty representation with discounting is the same as the optimal plan for the goal-reward representation with the same discount factor. Suppose the discount factor is γ. Consider an arbitrary plan and any of its trajectories. If the trajectory needs i action executions to achieve the goal (i can be finite or infinite), then its discounted total reward for the goal-reward representation is γ^i. Its discounted total reward for the action-penalty representation is $-\sum_{j=0}^{i-1}\gamma^j = \frac{\gamma^i-1}{1-\gamma}$. This shows that the discounted total reward of every trajectory for the goal-reward representation is a linear transformation of its discounted total reward for the action-penalty representation and the same discount factor. This means that the expected total discounted reward of every plan for the goal-reward representation is a linear transformation of its expected total discounted reward for the action-penalty representation and the same discount factor. Consequently, the optimal plan for the action-penalty representation with discounting is the same as the optimal plan for the goal-reward representation with the same discount factor. For example, Figure 5 shows that the optimal plan for the robot-navigation example with the action-penalty representation and discount factor 0.900 does not avoid all hazardous actions. Fortunately, the Selective State-Deletion method eliminates this version of the trapping phenomenon as well, as can be shown with a slightly modified version of the proof of Theorem 3. This allows one to use the action-penalty representation with discounting to determine a plan that maximizes the expected total discounted reward among all plans that achieve the goal, without any changes to the planning method. For example, Figure 9 shows the optimal plan for the robot-navigation example with the action-penalty representation and discount factor 0.900 after the Selective State-Deletion Method has been applied. This plan avoids the hazardous actions and thus ensures that the robot does not tip over.

Conclusion

We discussed two reward structures that have often been used in the decision-theoretic planning and reinforcement-learning literature: the goal-reward representation and the action-penalty representation. We showed that the action-penalty representation without discounting guarantees that the optimal plan achieves the goal for sure (if this is possible) but neither the action-penalty representation with discounting nor the goal-reward representation with discounting have this property. We explained this trapping phenomenon for the goal-reward representation using a novel interpretation for discounting, namely that it models agents that use convex exponential utility functions. These agents are optimistic in the face of uncertainty and thus do not avoid traps at all cost. This can be a problem for planning tasks where achieving the goal is the primary objective while minimizing the travel time is only of secondary importance. We then showed how the trapping phenomenon can be eliminated with our Selective State-Deletion Method, which deletes all states from the state space from which the goal cannot be achieved with probability one.

References

Barto, A.; Sutton, R.; and Watkins, C. 1989. Learning and sequential decision making. Technical Report 89–95, Department of Computer Science, University of Massachusetts at Amherst, Amherst (Massachusetts).

Barto, A.; Bradtke, S.; and Singh, S. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence* 73(1):81–138.

Bernoulli, D. 1738. Specimen theoriae novae de mensura sortis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 5. Translated by L. Sommer, *Econometrica*, 22: 23–36, 1954.

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

Corman, T.; Leiserson, C.; and Rivest, R. 1990. *Introduction to Algorithms*. MIT Press.

Dean, T.; Kaelbling, L.; Kirman, J.; and Nicholson, A. 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76(1–2):35–74.

Farquhar, P. and Nakamura, Y. 1988. Utility assessment procedures for polynomial-exponential functions. *Naval Research Logistics* 35:597–613.

Farquhar, P. 1984. Utility assessment methods. *Management Science* 30(11):1283–1300.

Kaelbling, L.; Littman, M.; and Moore, A. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Koenig, S. and Simmons, R.G. 1994. How to make reactive planners risk-sensitive. In *Proceedings of the International Conference on Artificial Intelligence Planning Systems*. 293–298.

Koenig, S. and Simmons, R.G. 1996. The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. *Machine Learning* 22(1/3):227–250. Appeared also as a book chapter in: Recent Advances in Reinforcement Learning; L. Kaelbling (ed.); Kluwer Academic Publishers; 1996.

Koenig, S. 1998. Representation changes for planning with exponential utility functions. In *Proceedings of the Symposium on Abstraction, Reformulation, and Approximation*. 79–84.

Lin, L.-J. 1993. *Reinforcement Learning for Robots using Neural Networks*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh (Pennsylvania). Available as Technical Report CMU-CS-93-103.

Marcus, S.; Fernàndez-Gaucherand, E.; Hernàndez-Hernàndez, D.; Colaruppi, S.; and Fard, P. 1997. Risk-sensitive Markov decision processes. In al., C. Byrneset., editor 1997, *Systems and Control in the Twenty-First Century*. Birkhauser. 263–279.

Peng, J. and Williams, R. 1992. Efficient learning and planning within the DYNA framework. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*. 281–290.

Puterman, M. 1994. *Markov Decision Processes – Discrete Stochastic Dynamic Programming*. Wiley.

Schoppers, M. 1987. Universal plans for reactive robots in unpredictable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1039–1046.

Stentz, A. 1995. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation* 10(3).

Sutton, R. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the International Conference on Machine Learning*. 216–224.

Thrun, S. and Schwartz, A. 1993. Issues in using function approximation for reinforcement learning. In *Proceedings of the Connectionist Models Summer School*.

Thrun, S. 1992. The role of exploration in learning control with neural networks. In White, D. and Sofge, D., editors 1992, *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. Van Nostrand Reinhold. 527–559.

von Neumann, J. and Morgenstern, O. 1947. *Theory of games and economic behavior*. Princeton University Press, second edition.

Watson, S. and Buede, D. 1987. *Decision Synthesis*. Cambridge University Press.

Wellman, M. 1990. *Formulation of Tradeoffs in Planning under Uncertainty*. Pitman.

Whitehead, S. 1991. A complexity analysis of cooperative mechanisms in reinforcement learning. In *Proceedings of the National Conference on Artificial Intelligence*. 607–613.

Appendix

In this appendix, we explain the transition probabilities of our synthetic example in detail, to allow the readers to reproduce our results. Assume that the robot moves one square in some direction (say, north in C2). We let c denote the current location of the robot (C2), s the intended destination (B2), l the location to the left of the intended destination (B1), and r the location to the right of the intended destination (B3). The probabilities that the robot ends up at these locations are p_c, p_s, p_l, and p_r, respectively. If a location does not exist (because it is outside of the terrain), the corresponding probability is set to zero. The probability that the robot tips over is p_d. It always holds that $p_c + p_s + p_l + p_r + p_d = 1$.

If location s does not exist, then $p_c = 1$ and $p_s = p_l = p_r = p_d = 0$. For the remaining cases, we let $d_f \in \{-2, -1, 0, 1, 2\}$ denote the difference in elevation between location c and location f, where $f \in \{s, l, r\}$. For example, if the elevation of location c is high and the elevation of location s is low, then $d_s = 2$. The probabilities are calculated in five steps:

1. Calculate the probabilities p'_s, p'_l, and p'_r. If location l (or r) does not exist, set $p'_l = 0$ (or $p'_r = 0$, respectively). Otherwise set the probabilities as follows

$d_{\{l,r\}}$	-2	-1	0	1	2
$p'_{\{l,r\}}$	0.01	0.03	0.05	0.07	0.09

2. Set $p'_s = 1 - p'_l - p'_r$.
3. Set $p_c = P(c|d_l)p'_l + P(c|d_s)p'_s + P(c|d_r)p'_r$, where

$d_{\{l,s,r\}}$	-2	-1	0	1	2
$P(c\|d_{\{l,s,r\}})$	0.2	0.1	0	0	0

4. Set $p_d = P(d|d_l)p'_l + P(d|d_s)p'_s + P(d|d_r)p'_r$, where

$d_{\{l,s,r\}}$	-2	-1	0	1	2
$P(d\|d_{\{l,s,r\}})$	0	0	0	0	0.1

5. Set $p_{\{l,s,r\}} = p'_{\{l,s,r\}}\left(1 - P(c|d_{\{l,s,r\}}) - P(d|d_{\{l,s,r\}})\right)$.