The Interaction of Representations and Planning Objectives for 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

We study decision-theoretic planning or reinforcement learning in the presence of traps such as steep slopes for outdoor robots or staircases for indoor robots. In this case, achieving the goal from the start is often the primary objective while minimizing the travel time is only of secondary importance. We study how this planning objective interacts with possible representations of the planning tasks, namely whether to use a discount factor that is one or smaller than one and whether to use the action-penalty or the goal-reward representation. We show that the action-penalty representation without discounting guarantees that the plan that maximizes the expected reward also achieves the goal from the start (provided that 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 of 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 our Selective State-Deletion Method can be used in conjunction with standard decision-theoretic planners to eliminate the trapping phenomenon.

1 Introduction

The planning literature studies representations of planning tasks 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 and reinforcement learning[1] provide 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.

In this paper, we point out the differences among common representations of decision-theoretic planning tasks, addressing practitioners of decision-theoretic planning, not theoreticians. While theoreticians often study decision-theoretic planning algorithms and their properties, practitioners have to apply them and, in this context, need to decide how to represent planning tasks. Often, there are several alternatives available that appear to be similar and practitioners need to understand their differences to ensure that decision-theoretic planners indeed determine those plans that fit the desired planning objectives.

We study this problem in the presence of traps such as steep slopes for outdoor robots or staircases for indoor robots. In this case, achieving the goal from the start is often the primary objective while minimizing the travel time is only of secondary importance. Thus, one often wants to find a plan that maximizes the expected reward among all plans that achieve the goal from the start. We study how this planning objective interacts with the possible

[1] Reinforcement learning adapts behavior in response to (possibly delayed) rewards and penalties. It 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.

representations of decision-theoretic planning tasks, namely whether to use a discount factor that is one or smaller than one and whether to use the action-penalty or the goal-reward representation. The action-penalty representation penalizes agents for every action that they execute and is a natural representation of resource consumptions in planning. The goal-reward representation, on the other hand, rewards agents for stopping in the goal and is a natural representation of positive reinforcements for completing tasks in reinforcement learning. We study what these representation alternatives have in common and how they differ, by combining ideas from artificial intelligence planning, operations research, and utility theory, using robot-navigation tasks as examples. For example, we show that the plan that maximizes the expected total reward for the action-penalty representation without discounting always achieves the goal from the start (provided that 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 of 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 our Selective State-Deletion Method can be used in conjunction with standard decision-theoretic planners to eliminate the trapping phenomenon.

2 Representing Planning Tasks with 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 goals in which execution stops. They consist of

- a finite set of states S;
- a start state $s_{start} \in S$;
- a set of goals $G \subseteq S$;
- 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 action reward $r(s,a,s')$ for each non-goal 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 (we say that action a leads from state s to state s' iff $p(s'|s,a) > 0$), and $r(s,a,s')$ denotes the (positive or negative) immediate reward that the agent receives for the transition;
- a real-valued goal reward $g(s)$ for each goal s, where $g(s)$ denotes the (positive or negative) goal reward that the agent receives whenever it is in state s and thus stops the execution of the plan.

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-goals to actions that can be executed in those states, also known as "stationary, deterministic policies" (short: 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 action reward and transitions to one of the successor states according to the corresponding transition probabilities. The agent always stops in goals (but not otherwise), in which case it receives the goal reward and then does not receive any further rewards.

3 Example: Robot-Navigation Tasks

We use outdoor robot-navigation tasks in a known terrain to illustrate how the planning tasks are modeled with GDMDPs. The task of the robot is to reach a given goal location from its start location. Movement of the robot

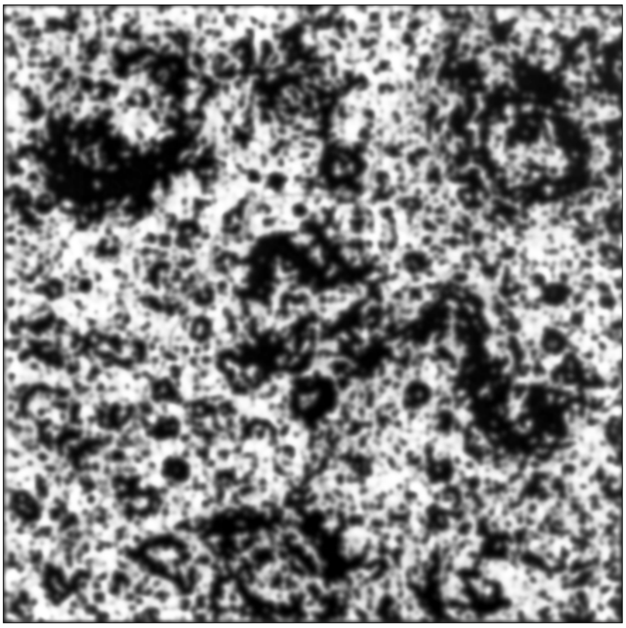

Figure 1: Terrain for the Navigation Task

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 with GDMDPs. The states of the GDMDPs naturally correspond to the possible locations of the robot. The actions of the GDMDPs correspond to moving in the four main compass directions. Figure 1 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 2 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 and thus achieve its goal. The actions that can tip over the robot ("hazardous actions") are indicated by arrows in Figure 2. Figure 3 gives an example of how the GDMDP is constructed from the terrain. Its left part shows the north-west corner of some terrain. In the corner location, the robot can move either north, east, south, or west. The right part of the figure shows how the east action is modeled for the corner location, including how the GDMDP models that the robot tips over, namely by transitioning to a new state that the robot cannot leave again, that is, where all movement actions lead to self transitions. In the following, we use this robot-navigation example to illustrate the interaction of representations and planning objectives for decision-theoretic planning tasks that are modeled with GDMDPs. In the appendix, we explain the transition probabilities of our robot-navigation example in detail, to allow the readers to reproduce our results.

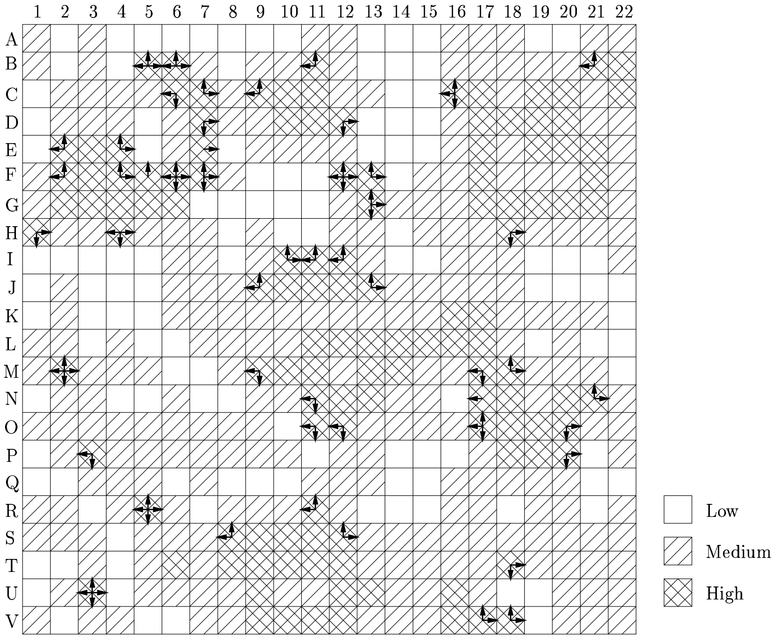

Figure 2: The Discretized Grid and Hazardous Actions (Patterns Indicate Different Elevations)

4 Planning for GDMDPs

The most common planning objective for GDMDPs is to maximize the expected total reward. If the agent receives action reward r_t during the $(t+1)$st action execution and reaches goal 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 a reward received after t action executions compared to the same reward received one action execution earlier. If the discount factor is smaller than one, one says that discounting is used and calls the total reward discounted. Otherwise, one says that discounting is not used and calls the total reward undiscounted.

One can use dynamic programming methods and thus standard decision-theoretic planners to efficiently find plans that maximize the expected total reward for given GDMDPs, which explains why this planning objective is so common. These plans can be determined by solving a system of $|S|$ equations for the $|S|$ variables $v(s)$, where $v(s)$ is the largest expected total reward that an agent can obtain if plan execution starts in state s. These equations are called Bellman's equations [Bellman, 1957]:

$$v(s) = \begin{cases} g(s) & \text{for all } s \in G \\ \max_{a \in A(s)} \sum_{s' \in S}[p(s'|s,a)(r(s,a,s') + \gamma v(s'))] & \text{for all } s \in S \setminus G. \end{cases} \quad (1)$$

The optimal action to execute in non-goal s is $a(s) = \text{one-of} \arg\max_{a \in A(s)} \sum_{s' \in S}[p(s'|s,a)(r(s,a,s') + \gamma v(s'))]$. The system of equations can be solved with linear programming in polynomial time [Littman *et al.*, 1995]. They can also be solved with dynamic programming methods such as Value Iteration [Bellman, 1957], Policy Iteration [Howard, 1964], and Q Learning [Watkins and Dayan, 1992]. As an example, we describe Value Iteration:

1. Set $v_1(s) := 0$ for all $s \in S \setminus G$ and $v_1(s) := g(s)$ for all $s \in G$. Set $t := 1$.

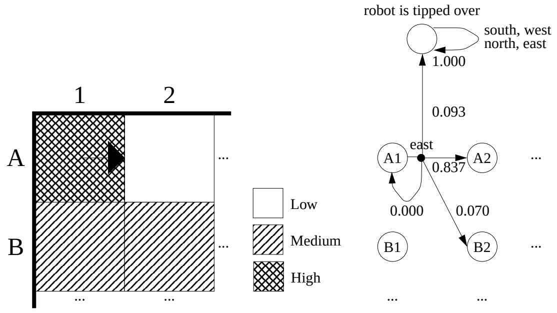

Figure 3: Modeling Robot-Navigation Tasks with GDMDPs

2. Set $v_{t+1}(s) := \max_{a \in A(s)} \sum_{s' \in S} [p(s'|s,a)(r(s,a,s') + \gamma v(s'))]$ for all $s \in S \setminus G$. Set $t := t + 1$.
3. Go to 2.

Here we leave the termination criterion unspecified. Then, $v(s) = \lim_{t \to \infty} v_t(s)$ for all $s \in S$ under some restrictive assumptions about the values of the various parameters. See [Bertsekas, 1987] for a good review of dynamic programming techniques for solving Bellman's equations.

5 Representation Alternatives

When modeling decision-theoretic planning tasks with GDMDPs, one has to make several decisions concerning their representation, for example, what the states are, what the discount factor is, and what the action and goal rewards are. There is often a natural choice for the states. In our robot-navigation example, for instance, the states of the GDMDP naturally correspond to the possible locations of the robot. However, the other two decisions are more difficult. For example, one can use a discount factor that is either one or smaller than one. Similarly, one can either use the action-penalty or the goal-reward representation to determine the action and goal rewards [Koenig and Simmons, 1996]. In the following, we first describe the alternatives in more detail and then study the resulting four combinations.

5.1 Discount Factor

One can use a discount factor that is either one or smaller than one. Discounting was originally motivated by collecting interest on resources. If an agent receives a reward $r \neq 0$ at time t and the interest rate is $i = (1-\gamma)/\gamma \geq 0$, then the reward is worth $(1+i)r = r/\gamma$ at time $t+1$. Thus, the discount factor is the relative value at time $t+1$ of a reward r received at time $t+1$ compared to the same reward received at time t:

$$0 < \frac{r}{r/\gamma} = \gamma \leq 1.$$

Consequently, a discount factor $\gamma < 1$ can be interpreted as modeling agents that can save or borrow resources at interest rate $(1-\gamma)/\gamma > 0$. 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.

A discount factor smaller than one can also be interpreted as positive probability with which the agent does not die during each action execution. When the agent dies, it cannot collect any further rewards. Thus, if it dies with probability $0 \leq 1-\gamma < 1$ between time t and time $t+1$, then it cannot collect reward r at time $t+1$ and thus the expected value of this reward at time t is $\gamma r + (1-\gamma)0 = \gamma r$. Thus, the discount factor is the relative value at time t of the reward r received at time $t+1$ compared to the same reward received at time t:

$$0 < \frac{\gamma r}{r} = \gamma \leq 1.$$

Consequently, a discount factor $\gamma < 1$ can be interpreted as modeling agents that die with probability $1-\gamma > 0$ after each action execution. However, it is rarely the case that agents die with the same probability after each action execution. The GDMDP thus best models the probability of dying, if any, with transition probabilities rather than the discount factor.

Often, therefore, discount factors $\gamma < 1$ are only used as a mathematical convenience because they guarantee that the expected total reward of every plan is finite, which simplifies the mathematics considerably. This explains why it is so popular to use discount factors smaller than one.

5.2 Action and Goal Rewards

One can use either the action-penalty or the goal-reward representation to determine the action and goal rewards.

- The **action-penalty representation** penalizes the agent for every action that it executes, but does not reward or penalize it for stopping in goals. Formally, $r(s,a,s') = -1$ for each non-goal s, state s', and action a that can be executed in state s, and $g(s) = 0$ for each goal s. The agent attempts to reach a goal with as few action executions as possible to minimize the amount of penalty that it receives.

  The action-penalty representation is often used in decision-theoretic planning since the consumption of a scarce resource (such as time, cost, or energy) can naturally be modeled as action penalties by making the action reward the negative of the resource consumption. The action-penalty representation has, for example, been used in [Barto *et al.*, 1989; Barto *et al.*, 1995; Dean *et al.*, 1995] in the context of robot navigation tasks.

- The **goal-reward representation** rewards the agent for stopping in a goals, but does not reward or penalize it for executing actions. Formally, $r(s,a,s') = 0$ for each non-goal s, state s', and action a that can be executed in state s, and $g(s) = 1$ for each goal s. Discounting is necessary with the goal-reward representation. Otherwise the agent would always receive a total reward of one when it reaches 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 with as few action executions as possible to maximize the portion of the goal reward that it receives.

  The goal-reward representation is often used in reinforcement learning since the positive reinforcement for completing a task can naturally be modeled as a goal reward. The goal-reward representation has, for example, been used in [Sutton, 1990; Whitehead, 1991; Peng and Williams, 1992; Thrun, 1992; Lin, 1993] in the context of robot-navigation tasks.

5.3 Combinations of Discount Factors and Rewards

We argued that there are three combinations of the alternatives: the action-penalty representation with and without discounting and the goal-reward representation with discounting. All three combinations have been used in the decision-theoretic planning literature and are considered to be approximately equivalent, for the following reason: We assume that all actions take unit time to execute. Then, the expected total reward for the action-penalty representation without discounting equals the negative expected plan-execution time since the goal rewards are zero and the action rewards equal the negative time needed to execute the actions. Consequently, a plan that maximizes the expected total reward for the action-penalty representation without discounting also minimizes the expected plan-execution time, which is often a desirable planning objective since one wants agents to reach the goal as quickly as possible. Furthermore, the following theorem shows that maximizing the expected total reward for the action-penalty representation with discounting and the goal-reward representation with discounting are equivalent to maximizing the expected total reward for the action-penalty representation without discounting both for deterministic planning tasks and for decision-theoretic planning tasks where the discount factor approaches one, under appropriate assumptions. Thus, all three combinations minimize the expected plan-execution time.

Theorem 1 *A plan that maximizes the (expected) total reward for the action-penalty representation without discounting, the action-penalty representation with discounting, or the goal-reward representation with the same discount factor also maximizes the (expected) total reward for the other two combinations provided that the GDMDP is deterministic.*

Proof: Suppose the discount factor is γ. Consider an arbitrary plan. If the plan needs i action executions to reach the goal from the start (i can be finite or infinite), then its total reward is $-i$ for the action-penalty representation without discounting, $-\sum_{j=0}^{i-1}\gamma^j = \frac{\gamma^i-1}{1-\gamma}$ for the action-penalty representation with discounting, and γ^i for the goal-reward representation with discounting. This shows that the total rewards are monotonic transformations of each other. Consequently, a plan that maximizes the total reward for the action-penalty representation without discounting, the action-penalty representation with discount factor γ, or the goal-reward representation with discount factor γ also maximizes the total reward for the other two combinations. ■

Theorem 2 *A plan that maximizes the expected total reward for the action-penalty representation without discounting, the action-penalty representation with a discount factor that approaches one, or the goal-reward representation with a discount factor that approaches one also nearly maximizes the expected total reward for the other two combinations.*[2]

Proof: As the discount factor approaches one, the expected total reward of any plan for the action-penalty representation with discounting trivially approaches its expected total reward for the action-penalty representation without discounting. Furthermore, a plan that maximizes the expected total reward for the action-penalty representation with discounting also maximizes the expected total reward for the goal-reward representation with the same discount factor, and vice versa, as we later show in Theorem 9. ■

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 three combinations are not necessarily equivalent. However, maximizing the expected total reward for the action-penalty representation with discounting and the goal-reward representation with discounting are often considered to be sufficiently similar to maximizing the expected total reward for the action-penalty representation without discounting that they are considered to minimize the expected plan-execution time approximately. This is the reason why they are used. However, there is a crucial difference between these two combinations and maximizing the expected total reward for the action-penalty representation without discounting, that we discuss in the following.

[2] To understand why such a plan only nearly maximizes the expected total reward for the other two combinations, consider the following synthetic example with only two plans. Plan 1 reaches the goal from the start with 2 action executions. Plan 2 reaches the goal from the start with probability 0.500 in 1 action execution and with probability 0.500 in 3 action executions. Then, both plans maximize the expected total reward for the action-penalty representation without discounting but only Plan 2 maximizes the expected total reward for the goal-reward representation (or action-penalty representation, respectively) with discounting for all discount factors smaller than one.

6 Achieving the Goal

We say that a plan achieves the goal from the start if the probability with which an agent that starts in the start state achieves a goal state within a given number of action executions approaches one as the bound approaches infinity, otherwise the plan does not achieve the goal from the start. We say that the goal can be achieved from the start if there exist at least one plan that achieves the goal from the start, otherwise the goal cannot be achieved from the start. While it can be rational or even necessary to trade off a smaller probability of not reaching the goal from the start and a smaller number of action executions in case the goal is reached (for example, if the goal cannot be achieved from the start), this is a problem when solving planning tasks for which achieving the goal from the start is the primary objective and minimizing the cost is only of secondary importance. Planning tasks with this lexicographic preference ordering often include robot navigation tasks in the presence of traps such as steep slopes for outdoor robots or staircases for indoor robots. In these cases, one often wants to rule out plans that risk the destruction of the robot, as convincingly argued in [Dean *et al.*, 1995]. One then often wants to find a plan that maximizes the expected total reward among all plans that achieve the goal from the start. In the following, we study how this can be done for the three combinations.

7 Action-Penalty Representation without Discounting

Every plan that maximizes the expected total reward for the action-penalty representation without discounting also achieves the goal from the start provided that the goal can be achieved from the start, as the following theorem shows.

Theorem 3 *Every plan that maximizes the expected total reward for the action-penalty representation without discounting achieves the goal from the start provided that the goal can be achieved from the start.*

Proof: Every plan that achieves the goal from the start has a finite expected total reward for the action-penalty representation without discounting. 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 from the start, 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 from the start and all action 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 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 from the start has an expected total 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 action rewards are negative and plan execution does not reach a goal with positive probability). ■

As an illustration, the plan that maximizes the expected total reward for the action-penalty representation without discounting avoids all hazardous actions for our robot-navigation example from Figure 2 and thus ensures that the robot does not tip over, see Figure 4. Each cell in the figure contains the action that that the robot executes when it is in that cell, provided that it can reach it under that plan. (Cells that the robot cannot reach under that plan do not contain actions.)

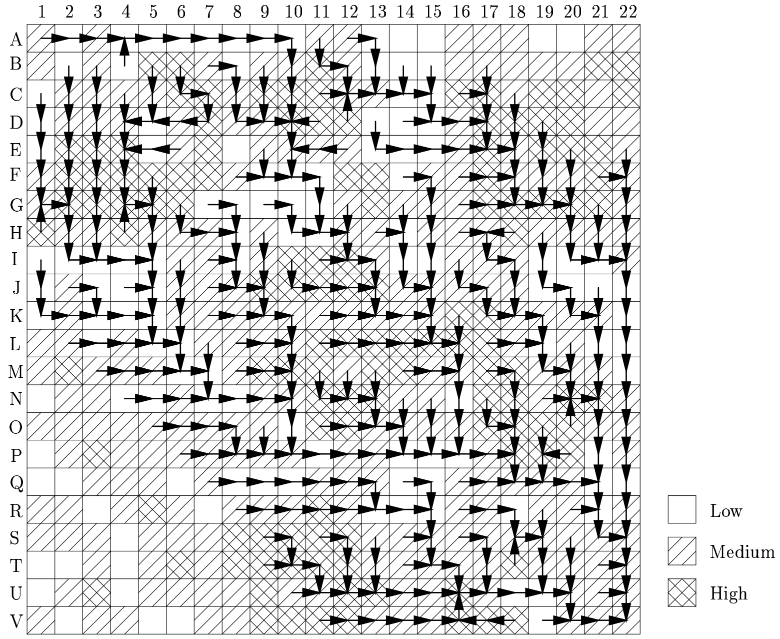

Figure 4: The Reward-Maximal Plan for the Action-Penalty Representation without Discounting

8 Goal-Reward Representation with Discounting

We just showed that a plan that maximizes the expected total reward for the action-penalty representation without discounting also achieves the goal from the start provided that the goal can be achieved from the start. Thus, one does not need to worry about achieving the goal from the start. One can use standard decision-theoretic planners to find a plan that maximizes the expected total reward, and the resulting plan achieves the goal from the start provided that the goal can be achieved from the start. Since the goal-reward representation with discounting is similar to the action-penalty representation without discounting, one could assume that the goal-penalty representation with discounting has the same property. We now show by example that, perhaps surprisingly, a plan that maximizes the expected total reward for the goal-reward representation with discounting does not necessarily achieve the goal from the start even if the goal can be achieved from the start. We call this phenomenon the trapping phenomenon. The trapping phenomenon thus differentiates between the goal-reward representation with discounting and the action-penalty representation without discounting.

Theorem 4 *A plan that maximizes the expected total reward for the goal-reward representation with discounting does not necessarily achieve the goal from the start even if the goal can be achieved from the start.*

Proof: Consider the following synthetic example, see Figure 5. Plan 1 reaches the goal from the start with 11 action executions. With probability 0.900, plan 2 reaches the goal from the start with one action execution. With the complementary probability, plan 2 cycles forever and thus does not achieve the goal from the start. 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 reward of $0.900 \times 0.900^{1} + 0.100 \times 0.900^{\infty} = 0.8100$. Thus, plan 2 has a larger expected total reward than plan 1, but does not achieve the goal from the start. ■

As an illustration, the plan that maximizes the expected total reward for the goal-reward representation with discounting does not avoid all hazardous actions for our robot-navigation example from Figure 2. This is true if the discount factor is 0.900, see Figure 6, and remains true even if the discount factor is 0.999 and thus very close to

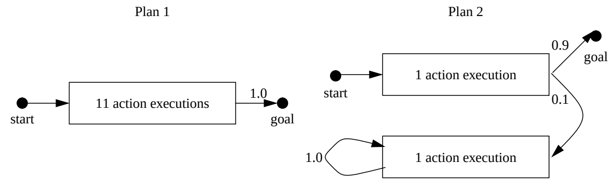

Figure 5: Two Plans that Illustrate the Trapping Phenomenon

one, see Figure 7. The hazardous actions that the robot executes with positive probability are circled in the figures. Their number decreases as the discount factor approaches one.

9 Relating the Representations

We now relate the goal-reward representation with discounting to the action-penalty representation without discounting, using 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 $u(r)$. We now show that the expected total reward of any plan for the goal-reward representation with discounting 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 $0 < \gamma < 1$.

Theorem 5 *Every plan that maximizes the expected total reward for the goal-reward representation with 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 trajectories. If the trajectory needs i action executions to reach the goal from the start (i can be finite or infinite), then its total reward for the goal-reward representation with discount factor γ is γ^i. Its total reward for the action-penalty representation without discounting is $-i$, and its total undiscounted utility is γ^i. This shows that the total reward of every trajectory for the goal-reward representation with discount factor γ equals its total undiscounted utility for the action-penalty representation and the convex exponential utility function $u(r) = \gamma^{-r}$. This means that the expected total reward of every plan for the goal-reward representation with discounting equals its expected total undiscounted utility for the action-penalty representation. ■

Consequently, maximizing the expected total reward for the goal-reward representation with discounting is the same as maximizing the expected total undiscounted utility for the action-penalty representation and a convex exponential utility function.

10 Explaining the Trapping Phenomenon

So far, we have related the expected total reward for the goal-reward representation with discounting to the expected total undiscounted utility for the action-penalty representation and a convex exponential utility function. We now explain why the plan that maximizes the expected total reward for the goal-reward representation with discounting is not guaranteed to achieve the goal from the start, using this relationship in conjunction with insights from utility theory. In the process, we provide a novel interpretation of discounting, namely that it models agents whose risk attitudes can be described with convex exponential utility functions, which implies that the agents are optimistic

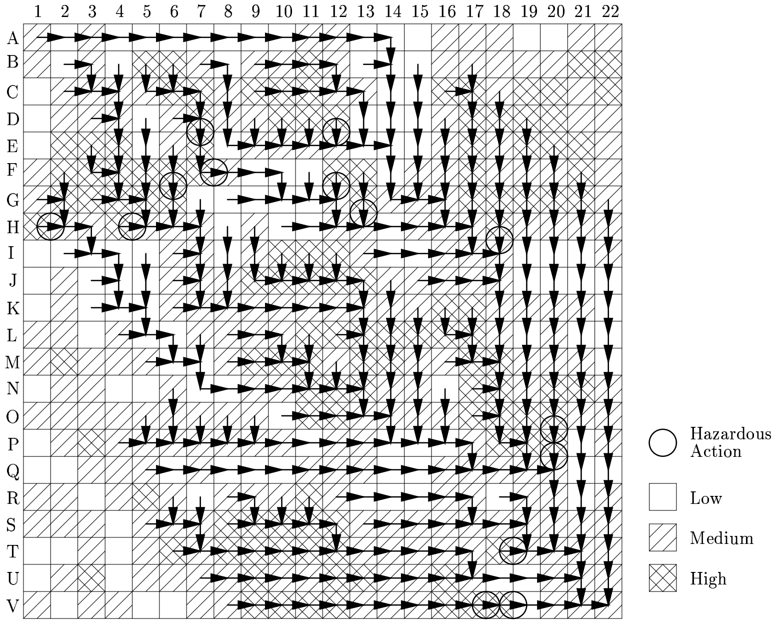

Figure 6: The Reward-Maximal Plan for the Goal-Reward Representation (and Action-Penalty Representation) with Discount Factor 0.900

(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 their optimism.

Utility theory investigates decision making in high-stake single-instance decision situations [Bernoulli, 1738; von Neumann and Morgenstern, 1947], such as environmental crisis situations [Koenig and Simmons, 1994; Koenig, 1998]. High-stake planning domains are domains in which huge wins or losses are possible. Single-instance planning domains are domains where plans can be executed only once (or a small number of times). In these situations, decision makers often do not maximize the expected total reward. Consider, for example, two alternatives with the following expected undiscounted payoffs:

	probability	payoff	expected payoff
alternative 1	1.0000000	0.00	0.00
alternative 2	0.9999999	-1.00	-0.90
	0.0000001	999,999.00	

This scenario corresponds to deciding whether to play the state lottery. The first alternative is to not play the lottery and thus neither win nor lose any money. The second alternative is to play the lottery and then to lose one dollar with 99.99999 percent probability (that is, buy a losing ticket for one dollar) and win 999,999 dollars with 0.00001 percent probability (that is, buy a winning ticket for one dollar and receive a payoff of 1,000,000 dollars). People who play the state lottery even though the expected payoff of playing is smaller than the one of abstaining are optimistic (or, synonymously, risk-seeking) and thus focus more on the best-case outcomes than the worst-case outcomes.

Utility theory explains why some decision makers do not maximize the expected payoff. It suggests that they choose plans for execution that maximizes the expected total undiscounted utility, where the utility function depends on the decision maker and has to be elicited from them. Convex utility functions account for the risk-seeking attitudes of some decision makers in the lottery example above. Assume, for example, that a decision maker has the convex exponential utility function $u(r) = 0.99999^{-r}$. Exponential utility functions

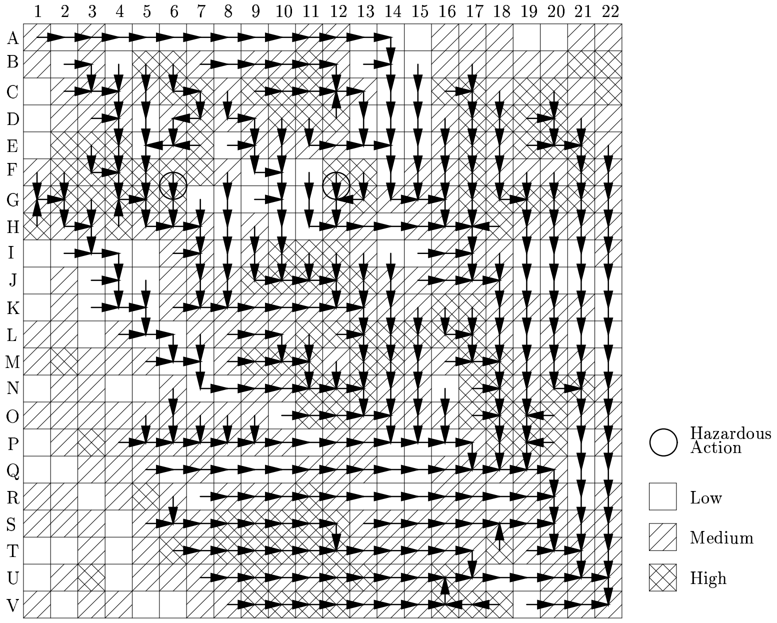

Figure 7: The Reward-Maximal Plan for the Goal-Reward Representation (and Action-Penalty Representation) with Discount Factor 0.999

are perhaps the most often used utility functions in utility theory [Watson and Buede, 1987] and specialized assessment procedures are available that make it easy to elicit them from decision makers [Farquhar, 1984; Farquhar and Nakamura, 1988]. Then, the two alternatives have the following expected total undiscounted utilities for this decision maker:

	probability	payoff	utility	expected utility
alternative 1	1.0000000	0.00	1.00000	1.00000
alternative 2	0.9999999	-1.00	0.99999	1.00219
	0.0000001	999,999.00	22,027.34686	

In this case, the expected total undiscounted utility of playing the lottery is larger than the one of abstaining, which explains why this decision maker chooses it over the one that maximizes the expected undiscounted payoff. Other decision makers can have different utility functions and thus make different decisions.

How optimistic a decision maker is depends on the parameter γ of the convex exponential utility function. Values of γ between zero and one trade off between maximizing the best-case and expected total undiscounted reward. In the context of GDMDPs, we know from the theory of risk-sensitive Markov decision processes [Marcus *et al.*, 1997] that the certainty equivalent of a 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. We also know from the theory of risk-sensitive Markov decision processes that the certainty equivalent of a 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.

Consequently, Theorem 5 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 is and thus the more it pays attention to the outcomes in the best case, not the outcomes in the worst case (tipping over). Thus, it is more likely to get trapped. For example, Figure 8 contains a log-log plot that shows for the robot-navigation example how the probability of tipping over

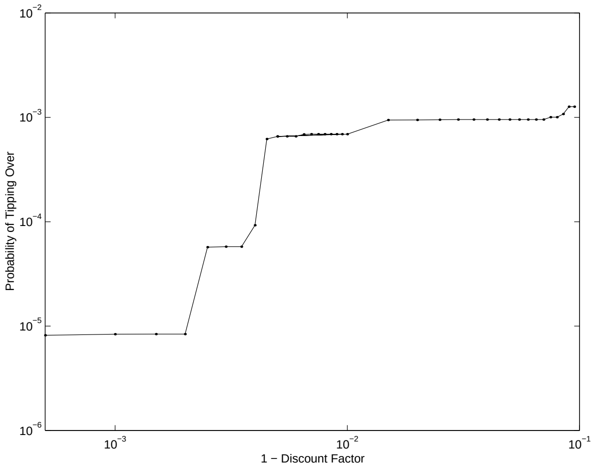

Figure 8: Probability of Tipping over when Executing a Reward-Maximal Plan for the Goal-Reward Representation

(and thus not achieving the goal) while executing the plan that maximizes the expected total reward for the goal-reward representation with discounting depends on the discount factor (ties among plans are broken randomly). The discount factor approaches one as one moves on the x axis to the left. The figure thus confirms that the trapping phenomenon becomes less pronounced as the discount factor approaches one, which explains our earlier observation that the number of hazardous actions that the robot executes with positive probability decreases as the discount factor approaches one. In fact, the combination of Theorems 2 and 3 shows that the number of hazardous actions that the robot executes with positive probability becomes zero and the trapping phenomenon thus disappears as the discount factor approaches one (provided that the goal can be achieved from the start).

11 Eliminating the Trapping Phenomenon

The trapping phenomenon can be avoided by finding a plan that maximizes the expected total reward among all plans that achieve the goal from the start. According to Theorem 3, one way of doing this is to use standard decision-theoretic planners to determine a plan that maximizes the expected total reward for the action-penalty representation without discounting. Unfortunately, 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 issue then is how to determine a plan for the goal-reward representation with discounting that maximizes the expected total reward among all plans that achieve the goal from the start. 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 total reward among all plans that achieve the goal from the start to a problem that we know how to solve with standard decision-theoretic planners, namely the problem of finding a plan that maximizes the expected 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. (Thus, traps are states from which the goal cannot be achieved.) We then use the following property.

Theorem 6 *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 from the start 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 from the start. 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 with probability zero from state s, but there exists a different plan $plan'$ that reaches a goal 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 with probability zero. At least one such state exists and all of them are non-goals. 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 trajectories 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 reaches the goal from the start. 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. ■

It is instructive to compare the following corollary of Theorem 6 to Theorem 3. Both statements are similar, except that one is conditioned on the goal being achievable from all states and the other is conditioned on the goal being achievable from only the start state.

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

To summarize Corollary 1, every plan that maximizes the expected total reward for the goal-reward representation with discounting necessarily also achieves the goal from the start provided that there are no traps and the goal can thus be achieved from all states. Thus, one does not need to worry about achieving the goal if the goal can be achieved from all states. One can use standard decision-theoretic planners to maximize the expected total reward. The resulting plan then achieves the goal from the start and thus also maximizes the expected total reward among all plans that achieve the goal from the start.

We now describe how one can find a plan that maximizes the expected total reward for the goal-reward representation with discounting among all plans that achieve the goal from the start even if the goal cannot be achieved from all states, namely by deleting all traps from the GDMDP (and all actions that lead to them). Our Selective State-Deletion Method, for example, uses the communicating structure of a GDMDP [Puterman, 1994] to delete the traps. In the following, we describe a simple version of the Selective State-Deletion Method.

Selective State-Deletion Method

Repeat the following steps until no states have been deleted during an iteration:

1. Construct the graph whose vertices are the states of the current GDMDP 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'.
2. Use standard methods from graph theory [Cormen *et al.*, 1990] to determine the strongly connected components of the graph (that is, the equivalence classes of vertices under the "are mutually reachable" relation).
3. Delete all states from the current GDMDP that are included in leaf components that do not contain vertices that correspond to goal states.
4. Delete all actions from the current GDMDP that lead to deleted states.

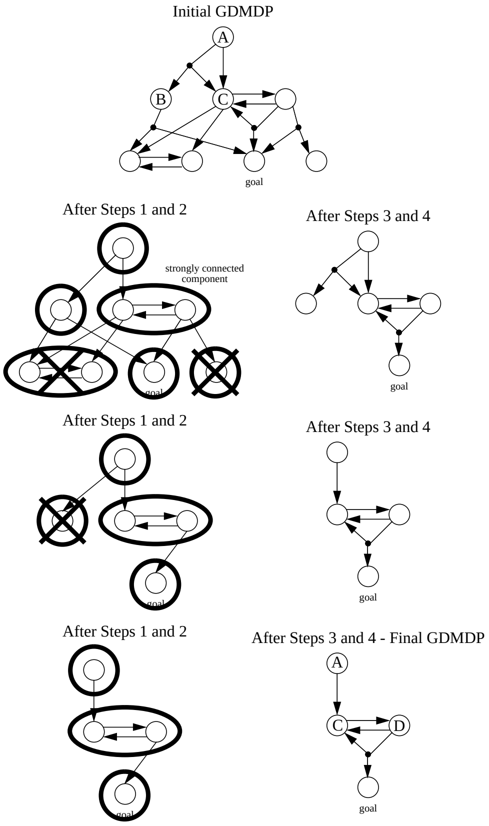

Figure 9: Example of Selective State Deletion

Figure 9 shows the Selective State-Deletion Method in action for a synthetic GDMDP, which is shown on top. Each hyper edge corresponds to an action in a state. Its destination vertices correspond to the states that the action leads to. The transition probabilities are not shown. For example, two actions can be executed in state A. One of them leads to states B and C. The other one leads with certainty to state C. The final GDMDP is shown at the bottom.

Theorem 7 *The Selective-State Deletion Method deletes exactly all traps from the GDMDP.*

Proof To simplify the description, we say in the following ”goal state” rather than ”goal states” and ”with probability one” rather than ”with a probability approaching one.” We prove both directions separately. Remember that a trap is a state from which the goal state cannot be reached with probability one.

- The Selective-State Deletion Method deletes each trap from the original GDMDP.

  Assume that this is not the case. Then, the Selective-State Deletion Method terminates without deleting at least one trap. Consider an arbitrary GDMDP for which the goal state can be reached from every state with positive probability. Then there is a policy for that GDMDP for which the goal state is reached from every state with positive probability. For this policy, the goal state can be reached from every state with probability one. Thus, if there is at least one trap in the GDMDP after termination of the Selective-State Deletion Method, then there is a state from which the goal state cannot be reached with positive probability. This means that the graph constructed from the final GDMDP (in Step 1 of the Selective-State Deletion

Method) has a leaf component that does not contain the goal state but the Selective-State Deletion Method would have deleted this component rather than stopped, which is a contradiction.

- The Selective-State Deletion Method deletes only traps from the original GDMDP.

  We prove the statement by induction. The statement is true for all states deleted during the first iteration of the Selective-State Deletion Method since the goal state cannot be reached with positive probability from them. Assume that the statement is true at the beginning of some iteration and consider a state deleted during that iteration. This state is part of a leaf component of the graph constructed from the GDMDP at the beginning of the iteration that does not contain the goal state. Thus, the goal state cannot be reached with positive probability from it for the GDMDP at the beginning of the iteration. It could be the case that there are additional states, including the goal state, that can be reached from it for the original GDMDP via actions that have already been deleted. However, these actions were deleted because they lead with positive probability to deleted states, that is, traps according to the induction assumption. Thus, the goal state cannot be reached from it for the original GDMDP other than via traps. This implies that the goal state cannot be reached from it with probability one for the original GDMDP and it is thus a trap. ■

Thus, the GDMDP that results from applying the Selective-State Deletion Method is the same as the original GDMDP except that all traps have been deleted. Instead of deleting the traps s, one can also clamp their values $v(s)$ to minus infinity. This way, the traps are effectively deleted. Corollary 1 applies to the GDMDP after the traps have been deleted. Consequently, every plan that maximizes the expected total reward also achieves the goal from the start. This property is non-trivial because deleting the traps only eliminates some of the plans that do not achieve the goal from the start but not all of them. For example, there exists a plan for the final GDMDP from Figure 9 that does not achieve the goal from the start, namely the plan that moves from state A to state C and then moves back and forth between states D and C.

The following theorem shows that every plan that maximizes the expected total reward after all traps have been deleted from the GDMDP not only achieves the goal from the start but also maximizes the expected reward of the original GDMDP among all plans that achieve the goal from the start.

Theorem 8 *Every plan that maximizes the expected reward for the goal-reward representation with discounting after all traps have been deleted from the GDMDP also maximizes the expected reward of the original GDMDP for the goal-reward representation with the same discount factor among all plans that achieve the goal from the start.*

Proof: Deleting the traps does not eliminate any plan that achieves the goal from the start, and it does not change the expected total rewards of the unaffected plans. Every plan that maximizes the expected total reward among the unaffected plans necessarily also achieves the goal from the start according to Corollary 1, and consequently also maximizes the expected total reward of the original GDMDP among all plans that achieve the goal from the start. ■

This theorem allows one to use standard decision-theoretic planners to determine a plan that maximizes the expected total reward for the goal-reward representation with discounting among all plans that achieve the goal from the start, simply by first deleting all traps from the GDMDP (for example, with the Selective-State Deletion Method) and then using a standard decision-theoretic planner to determine a plan for the resulting GDMDP that maximizes the expected total reward. If there is no plan that achieves the goal from the start after all traps have been deleted (that is, the start was a trap and consequently has been deleted), then there was no such plan for the original GDMDP either, and the planning task has to be solved with a preference model that trades off the probability of reaching the goal from the start and the execution cost of a plan, and thus gives up on the goal-oriented preference model of traditional artificial intelligence planning [Wellman, 1990].

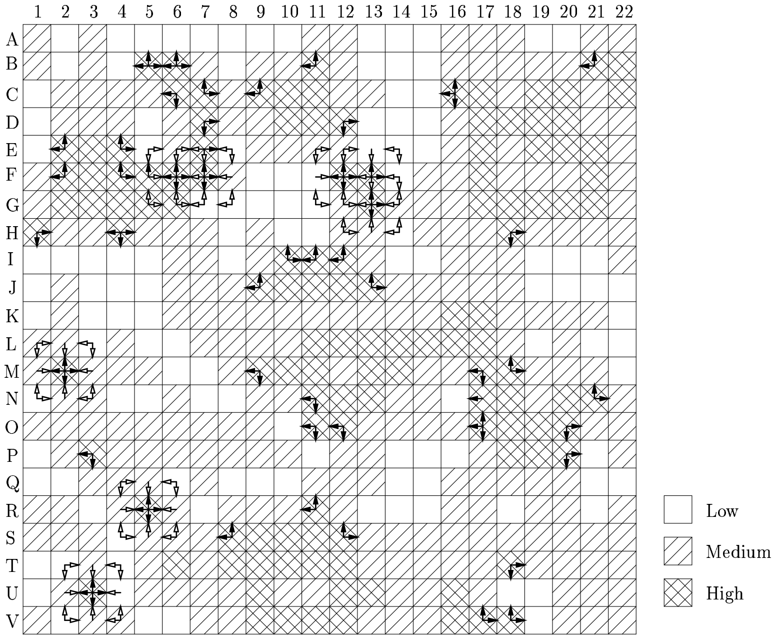

Figure 10: Actions Deleted after Selective State Deletion

As an illustration, we use the Selective State-Deletion Method to determine a plan that maximizes the expected total reward for the goal-reward representation with discounting for our robot-navigation example from Figure 2 among all plans that avoid the hazardous actions. Figure 10 shows the actions that the Selective State-Deletion Method deletes. This includes all hazardous actions (filled arrows), see Figure 2, and all actions after whose execution the robot can eventually be forced to execute a hazardous action (hollow arrows). Figure 11 shows the resulting plan that avoids the hazardous actions from Figure 6 and thus ensures that the robot does not tip over. Similarly, Figure 12 shows the resulting plan that avoids the hazardous actions from Figure 7.

12 Action-Penalty Representation with Discounting

We now show that the action-penalty representation with discounting is similar to the goal-reward representation with discounting. In fact, the following theorem shows that maximizing the expected total reward for the action-penalty representation with discounting is the same as maximizing the expected total reward for the goal-reward representation with the same discount factor.

Theorem 9 *A plan that maximizes the expected total reward for the action-penalty representation with discounting also maximizes the expected total reward for the goal-reward representation with the same discount factor, and vice versa.*

Proof: Suppose the discount factor is γ. Consider an arbitrary plan and any of its trajectories. If the trajectory needs i action executions to reach the goal from the start (i can be finite or infinite), then its total reward for the goal-reward representation with discounting is γ^i. Its total reward for the action-penalty representation with discounting is $-\sum_{j=0}^{i-1}\gamma^j = \frac{\gamma^i-1}{1-\gamma}$. This shows that the total reward of every trajectory for the goal-reward representation with discounting is a linear transformation of its total reward for the action-penalty representation with the same discount factor. This means that the expected total reward of every plan for the goal-reward representation with discounting

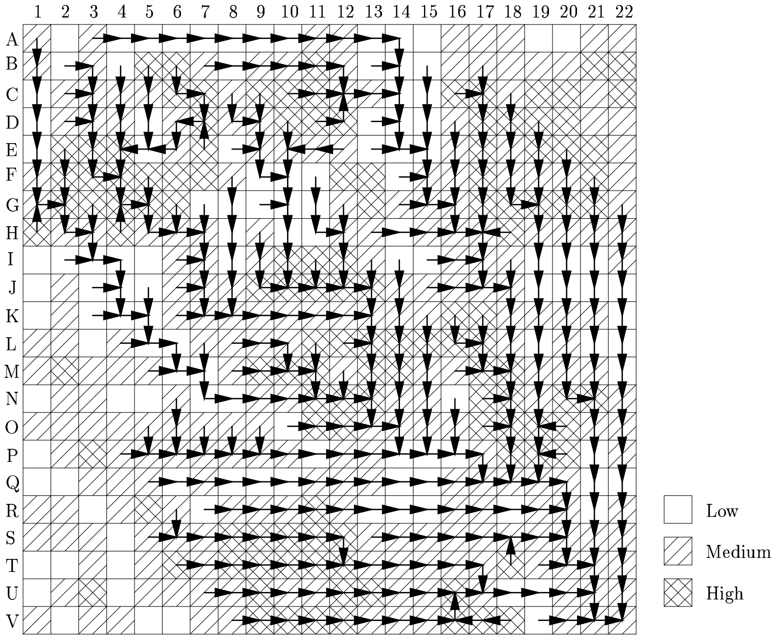

Figure 11: The Reward-Maximal Plan for the Goal-Reward Representation (and Action-Penalty Representation) with Discount Factor 0.900 after Selective State Deletion

is a linear transformation of its expected total reward for the action-penalty representation and the same discount factor. Consequently, a plan that maximizes the expected total reward for the action-penalty representation with discounting also maximizes the expected total reward for the goal-reward representation with the same discount factor, and vice versa. ■

We can use this similarity between the action-penalty representation with discounting and the goal-reward representation with discounting to show that a plan that maximizes the expected total reward for the action-penalty representation with discounting does not necessarily achieve the goal from the start even if the goal can be achieved from the start. Thus, the trapping phenomenon occurs not only for the goal-reward representation with discounting but also for the action-penalty representation with discounting.

Theorem 10 *A plan that maximizes the expected total reward for the action-penalty representation with discounting does not necessarily achieve the goal from the start even if the goal can be achieved from the start.*

Proof: Theorem 9 allows us to reuse the example from Theorem 4. Thus, consider again the synthetic example from Figure 5 and assume that we use the action-penalty representation with discount factor $\gamma = 0.900$. Then, plan 1 has a total reward of $(0.900^{11} - 1)/(1 - 0.900) = -6.8619$, and plan 2 has an expected total reward of $0.900 \times (-1) + 0.100 \times (-1)/(1 - 0.900) = -1.9000$. Thus, plan 2 has a larger expected total reward than plan 1, but does not achieve the goal from the start. ■

As an illustration, Figure 6 does not only show the plan that maximizes the expected reward for the goal-reward representation with discount factor 0.900 for our robot-navigation example from Figure 2 but also the plan that maximizes the expected reward for the action-penalty representation with discount factor 0.900. The plan does not avoid all hazardous actions.

Corollary 2 *Every plan that maximizes the expected total reward for the action-penalty representation with discounting achieves the goal from the start provided that the goal can be achieved from all states.*

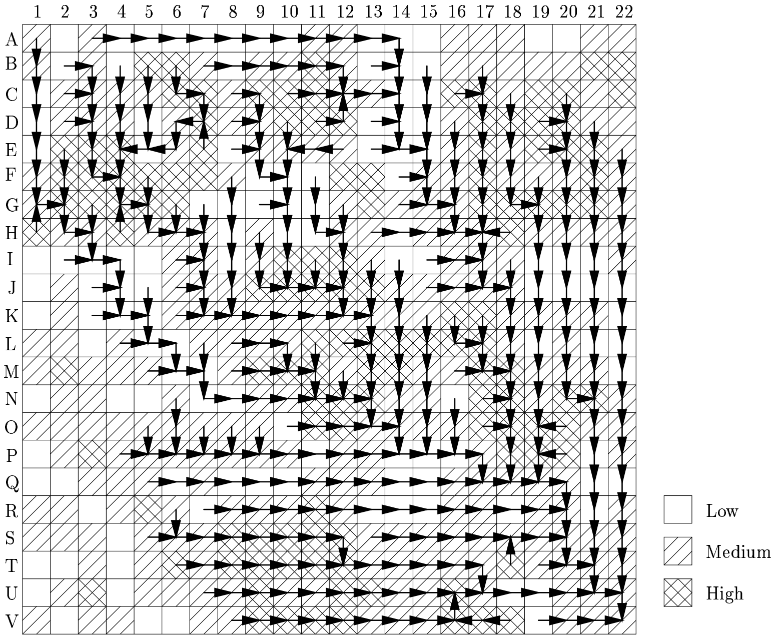

Figure 12: The Reward-Maximal Plan for the Goal-Reward Representation (and Action-Penalty Representation) with Discount Factor 0.999 after Selective State Deletion

Proof: The theorem follows directly from Theorem 9 and Corollary 1. ■

Theorem 11 *Every plan that maximizes the expected reward for the action-penalty representation with discounting after all traps have been deleted from the GDMDP also maximizes the expected reward of the original GDMDP for the action-penalty representation with the same discount factor among all plans that achieve the goal from the start.*

Proof: The proof follows the one of Theorem 8. ■

Thus, as for the goal-reward representation with discounting, one can use standard decision-theoretic planners to determine a plan for the action-penalty representation with discounting that maximizes the expected total reward among all plans that achieve the goal from the start, simply by first deleting all traps from the GDMDP (for example, with the Selective-State Deletion Method) and then using a standard decision-theoretic planner to determine a plan for the resulting GDMDP that maximizes the expected total reward.

As an illustration, Figure 11 shows the plan that maximizes the expected total reward for the action-penalty representation with discount factor 0.900 for our robot-navigation example from Figure 2 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.

13 Conclusion

Planning tasks are often modeled with Goal-directed Markov Decision Process models (GDMDPs). One then has to decide whether to use a discount factor that is one or smaller than one and whether to use the action-penalty or the goal-reward representation. The action-penalty representation penalizes the agent for every action that it

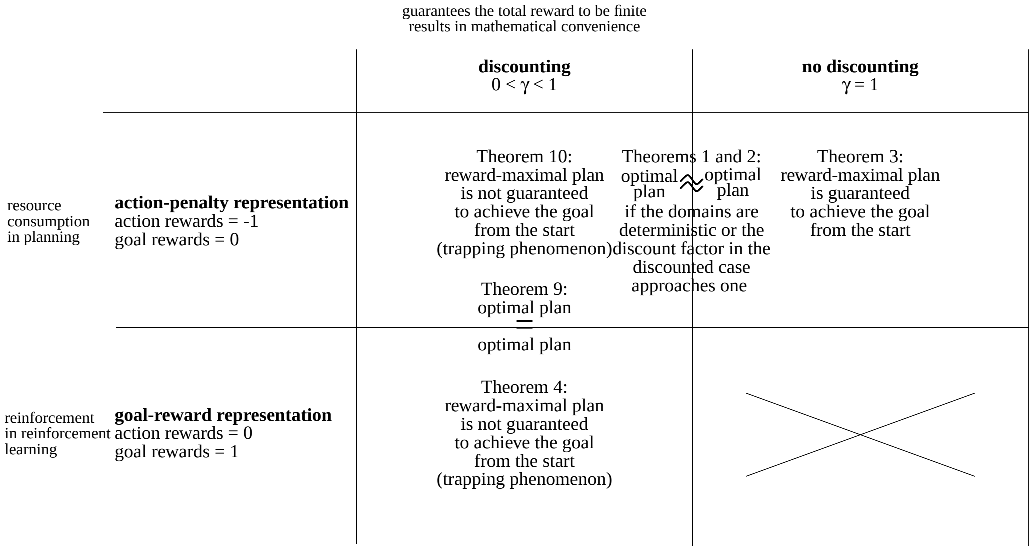
guarantees the total reward to be finite
results in mathematical convenience
discounting
$0 < \gamma < 1$
no discounting
$\gamma = 1$
resource consumption in planning
action-penalty representation
action rewards = -1
goal rewards = 0
Theorem 10: reward-maximal plan is not guaranteed to achieve the goal from the start (trapping phenomenon)
Theorems 1 and 2: optimal plan ≈ optimal plan if the domains are deterministic or the discount factor in the discounted case approaches one
Theorem 3: reward-maximal plan is guaranteed to achieve the goal from the start
Theorem 9: optimal plan = optimal plan
reinforcement in reinforcement learning
goal-reward representation
action rewards = 0
goal rewards = 1
Theorem 4: reward-maximal plan is not guaranteed to achieve the goal from the start (trapping phenomenon)

Figure 13: Summary of Properties

executes and is a natural representation of resource consumptions in planning. The goal-reward representation, on the other hand, rewards the agent for stopping in a goal and is a natural representation of positive reinforcements for completing tasks in reinforcement learning. Since the goal-reward representation has to be used with a discount factor that is smaller than one, there are three combinations, all of which have been used in the literature on decision-theoretic planning.

GDMDPs can be solved efficiently with standard decision-theoretic planners that maximize the expected total reward. Researchers, however, often want to find plans that maximize the expected total reward among all plans that achieve the goal from the start. For example, they want to move a robot as quickly as possible from the start to the goal but guarantee that the robot does not tip over in the presence of traps such as steep slopes for outdoor robots or staircases for indoor robots.

Maximizing the expected total reward for the action-penalty representation with or without discounting and maximizing the expected total reward for the goal-reward representation with discounting are equivalent both for deterministic planning tasks and for decision-theoretic planning tasks where the discount factor approaches one. In all three cases, each plan that maximizes the expected total reward minimizes the expected plan-execution time if all actions have unit cost and achieves the goal from the start provided that this is possible. Some researchers prefer to use a discount factor smaller than one, for example, because it guarantees that the total reward is finite, which greatly simplifies the mathematics. Maximizing the expected total reward for the action-penalty representation with discounting and the goal-reward representation with discounting remain equivalent for the same discount factor. However, it is no longer guaranteed that each plan that maximizes the expected total reward also achieves the goal from the start even if this is possible. We called this phenomenon the trapping phenomenon. Figure 13 summarizes the properties that we discussed in this paper.

We explained the trapping phenomenon using a novel interpretation of 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. In particular, we showed that the expected total reward of any plan for the goal-reward representation with discounting equals its expected total undiscounted utility for the action-penalty representation with a convex exponential utility function. This novel interpretation of discounting enabled us to use insights from utility theory to improve our understanding of discounting.

We then showed how the trapping phenomenon can be eliminated with the Selective State-Deletion Method, which deletes all traps, that is, states from which the goal cannot be reached with probability one from the start. Each plan that maximizes the expected total reward for the resulting planning task then also maximizes the expected total reward for the original planning task among all plans that achieve the goal from the start. Thus, the resulting planning task can again be solved with standard decision-theoretic planners.

Acknowledgments

The Intelligent Decision-Making Group is partly supported by an IBM faculty fellowship and NSF awards under contracts IIS-9984827, IIS-0098807, and ITR/AP-0113881. 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 and agencies or the U.S. government. Yaxin Liu performed all of the experiments.

A Appendix

In this appendix, we explain the transition probabilities of our robot-navigation 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 $p_{\{l,s,r\}} = p'_{\{l,s,r\}} \left(1 - P(c|d_{\{l,s,r\}}) - P(d|d_{\{l,s,r\}})\right)$.

References

(Barto *et al.*, 1989) 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 *et al.*, 1995) Barto, A.; Bradtke, S.; and Singh, S. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence* 73(1):81–138.

(Bellman, 1957) Bellman, R. 1957. *Dynamic Programming*. Princeton University Press.

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

(Bertsekas, 1987) Bertsekas, D. 1987. *Dynamic Programming, Deterministic and Stochastic Models*. Prentice Hall.

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

(Cormen *et al.*, 1990) Cormen, T.; Leiserson, C.; and Rivest, R. 1990. *Introduction to Algorithms*. MIT Press.

(Dean *et al.*, 1995) 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 and Nakamura, 1988) Farquhar, P. and Nakamura, Y. 1988. Utility assessment procedures for polynomial-exponential functions. *Naval Research Logistics* 35:597–613.

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

(Howard, 1964) Howard, R. 1964. *Dynamic Programming and Markov Processes*. MIT Press, third edition.

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

(Koenig and Simmons, 1994) 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 and Simmons, 1996) 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, 1998) Koenig, S. 1998. Representation changes for planning with exponential utility functions. In *Proceedings of the Symposium on Abstraction, Reformulation, and Approximation*. 79–84.

(Lin, 1993) 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.

(Littman *et al.*, 1995) Littman, M.; Dean, T.; and Kaelbling, L. 1995. On the complexity of solving Markov decision problems. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*. 394–402.

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

(Peng and Williams, 1992) 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, 1994) Puterman, M. 1994. *Markov Decision Processes – Discrete Stochastic Dynamic Programming*. Wiley.

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

(Stentz, 1995) Stentz, A. 1995. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation* 10(3):89–100.

(Sutton, 1990) 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 and Schwartz, 1993) Thrun, S. and Schwartz, A. 1993. Issues in using function approximation for reinforcement learning. In *Proceedings of the Connectionist Models Summer School*.

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

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

(Watkins and Dayan, 1992) Watkins, C. and Dayan, P. 1992. Q-learning. *Machine Learning* 8(3-4):279–292.

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

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

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