Extreme Risk Averse Policy for Goal-Directed Risk-Sensitive Markov Decision Process

Valdinei Freire
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
São Paulo, Brazil
e-mail: valdinei.freire@usp.br

Karina Valdivia Delgado
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
São Paulo, Brazil
e-mail: kvd@usp.br

***Abstract*—The Goal-Directed Risk-Sensitive Markov Decision Process allows arbitrary risk attitudes for the probabilistic planning problem to reach a goal state. In this problem, the risk attitude is modeled by an expected exponential utility and a risk factor λ. However, the problem is not well defined for every λ, posing the problem of defining the maximum (extreme) value for this factor. In this paper, we propose an algorithm to find this ϵ-extreme risk factor and the corresponding optimal policy.**

I. Introduction

Sequential decision problems can be modeled as a Markov Decision Processes (MDP). In MDPs, at each instant of discrete time, the agent observes a state, executes an action, transits to a next state following a probabilistic transition function and pays a cost. Most of the MDP solvers consider risk-neutral attitude, i.e., they seek policies that minimize the expected cumulative cost [1]. While such a policy is good in the expected case, it is known that in some cases, the decision makers prefer a risk averse attitude [2], [3].

A model that considers arbitrary attitude (neutral, prone and averse) towards risk is the Risk Sensitive Markov Decision Process (RSMDP) that minimizes the expected exponential utility and uses a risk factor λ. Patek [4] extends the notion of Goal Directed MDPs (an MDP that includes a set of goal states) to a model with an exponential risk-averse objective (called Goal Directed Risk Sensitive Markov Decision Process) and has already proved the conditions for the existence of a valid policy for this model. However, he has not showed how to build one and what is the maximum value for this risk factor.

In this paper, we study the problem of extreme risk averse in Goal Directed Risk Sensitive Markov Decision Process, and we propose an algorithm that will help to set the risk attitude arbitrarily, finding a solution, when it is possible; and finding an ϵ-extreme risk optimal solution otherwise.

The paper is organized as follows, in sections II and III we present an explanation about Markov Decision Processes and Risk Sensitive Markov Decision Processes. In section IV we present our algorithm. Finally in sections V, VI and VII, we present the related work, experiments and conclusion, respectively.

II. Goal-Directed Markov Decision Process

A Goal-Directed MDP [5], [6] (GD-MDP)is as a tuple $\mathcal{MDP} = \langle \mathcal{S}, \mathcal{A}, T, c, \mathcal{G}, \rangle$ where:

- $\mathcal{S}$ is a set of states;
- $\mathcal{A}$ is a set of actions that can be performed at each period of decision $t \in \{0, 1, 2, \ldots\}$;
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition function that represents the probability of the system transits to a state $s' \in \mathcal{S}$ after the agent executes an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$, i.e., $P(s_{t+1} = s' | s_t = s, a_t = a) = T(s, a, s')$;
- $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ is a cost function that represents the cost of taking an action $a \in \mathcal{A}$ when the process is in a state $s \in \mathcal{S}$, i.e., $c_t = c(s_t, a_t)$; and
- $\mathcal{G} \in \mathcal{S}$ is a set of absorbing goal states, i.e., $P(s_{t+1} \in \mathcal{G} | s_t \in \mathcal{G}, a_t = a) = 1$ and $c(s_t \in \mathcal{G}, a_t = a) = 0$ for any $a \in \mathcal{A}$.

The GD-MDP problem defines a discrete dynamic process. At any time t, the agent observes a state s_t, executes an action a_t, transits to a state s_{t+1} following T and pays a cost c_t. The process ends after reaching any goal state in $\mathcal{G}$.

The solution to GD-MDPs is a stationary policy defined by $\pi : \mathcal{S} \to \mathcal{A}$. The set of stationary policies is represented by Π. A policy gives the action to execute at any time t, i.e., if the process is in the state s_t, then action $a_t = \pi(s_t)$ is executed. The execution of a policy and the dynamic of a process defines a random variable C^π which stands for total cost for policy π and is defined by:

$$C^\pi = \lim_{M \to \infty} \sum_{t=0}^{M} c_t = \lim_{M \to \infty} \sum_{t=0}^{M} c(s_t, \pi(s_t)).$$

To find an optimal policy, a utility function $u : \mathbb{R}^+ \to \mathbb{R}$ must be defined and a value function of a policy, V^π. The value of a policy π at state s is given by the expected utility [7]:

$$V^\pi(s) = \mathrm{E}[u(C^\pi) | s_0 = s].$$

A policy π^* is optimal if and only if $V^{\pi^*}(s) \leq V^\pi(s)$ for every policy $\pi \in \Pi$ and every state $s \in \mathcal{S}$. In this paper, we will focus on GD-MDPs.

A. Attitudes Regarding to Risk

Since C^π is a random variable, we may consider three general attitudes regarding to risk [7]: neutral, prone and averse. First, we need to define the certainty equivalent of a policy π. Intuitively a certainty equivalent is a guaranteed cost that the agent would prefer to pay, rather than taking a chance on a lower, but uncertain cost. If $V^\pi(s) < \infty$ and there exists the inverse function $u^{-1} : \mathbb{R} \to \mathbb{R}^+$, the certainty equivalent $\overline{C}^\pi(s)$ of a policy π is defined by:

$$\overline{C}^\pi(s) = u^{-1}(V^\pi(s)),$$

and the expected cost $\widetilde{C}^\pi(s)$ of a policy π is defined by:

$$\widetilde{C}^\pi(s) = \mathrm{E}[C^\pi | s_0 = s].$$

An agent is risk prone if $\overline{C}^\pi(s) < \widetilde{C}^\pi(s)$, risk averse if $\overline{C}^\pi(s) > \widetilde{C}^\pi(s)$ and risk neutral $\overline{C}^\pi(s) = \widetilde{C}^\pi(s)$ for every state $s \in \mathcal{S}$ and policy $\pi \in \Pi$. For example, a risk averse agent prefers to pay for sure a cost of $\overline{C}^\pi(s)$ even when it is expected to pay less if the process is followed.

In sections II-B and III, we describe the utility function used by GD-MDPs and Goal-Directed Risk-Sensitive MDPs (GD-RSMDPs), respectively. The first one considers the identity function as utility function, i.e., $u(x) = x$, characterizing a neutral attitude; whereas the latter considers an exponential function, i.e., $u(x) = -\mathrm{sgn}(\lambda)\exp(\lambda x)$ where sgn and exp are the signum and exponential function, respectively. GD-RSMDPs characterizes a risk-prone agent if $\lambda < 0$ and a risk-averse agent if $\lambda > 0$.

B. A GD-MDP Solution

A GD-MDP evaluates a policy π by considering the identity utility function $u(x) = x$ and usually defines the value function by:

$$V^\pi(s) = \lim_{M\to\infty} \mathrm{E}\left[\sum_{t=0}^{M} c(s_t, \pi(s_t))\right],$$

which can be found by solving the following system of equations:

$$V^\pi(s) = c(s, \pi(s)) + \sum_{s' \in \mathcal{S}} T(s, \pi(s), s') V^\pi(s'), \forall s \in \mathcal{S}, \quad (1)$$

or in its vector-matrix form:

$$\mathbf{V}^\pi = \mathbf{c}^\pi + \mathbf{T}^\pi \mathbf{V}^\pi, \quad (2)$$

where $\mathbf{V}^\pi$ is a $|\mathcal{S}| \times 1$ column vector, and $\mathbf{c}^\pi$ and $\mathbf{T}^\pi$ are defined bellow.

Definition 1: **(Policy Transition Matrices and Policy Cost Vector)** Let π be a stationary policy and an enumeration of every state in $\mathcal{S}$ as $1, 2, 3, \ldots, |\mathcal{S}|$. $\mathbf{T}^\pi$ is the $|\mathcal{S}| \times |\mathcal{S}|$ matrix, where each cell $(\mathbf{T}^\pi)_{ij}$ represents the transition probability from state i to state j when following policy π, i.e., $(\mathbf{T}^\pi)_{ij} = T(i, \pi(i), j)$. The $|\mathcal{S}| \times |\mathcal{S}|$ matrix $\mathbf{T}^\pi_{\mathcal{G}^c}$ is the matrix $\mathbf{T}^\pi$ where columns representing states in set $\mathcal{G}$ are set to 0, i.e.,

$$(\mathbf{T}^\pi_{\mathcal{G}^c})_{ij} = \begin{cases} 0 & \text{, if } j \in \mathcal{G} \\ T(i, \pi(i), j) & \text{, otherwise} \end{cases}.$$

$\mathbf{c}^\pi$ is the $|\mathcal{S}| \times 1$ column vector, where each cell $(\mathbf{c}^\pi)_i$ is the cost when following the policy π in state i, i.e., $(\mathbf{c}^\pi)_i = c(i, \pi(i))$.

Definition 2: **(Proper policy)** A policy π is proper if $\lim_{t\to\infty}(\mathbf{T}^\pi_{\mathcal{G}^c})^t = \mathbf{0}$, i.e., an absorbing state in $\mathcal{G}$ is reached with probability 1. Equivalently, a policy π is a proper policy if the spectral radius of $\mathbf{T}^\pi_{\mathcal{G}^c}$ is less than 1, i.e., $\rho(\mathbf{T}^\pi_{\mathcal{G}^c}) < 1$.

Policy Iteration (PI) (Algorithm 1) is one of the classical algorithms to find an optimal policy π^* [1] . It is an iterative algorithm that begins with an initial proper policy π_0 and at each iteration i executes two steps: *policy evaluation* and *policy improvement*. Policy evaluation step uses Equation 1 to compute the value of $V^{\pi_i}(\cdot)$ and policy improvement step improves π_i obtaining π_{i+1}.

Algorithm 1 Policy Iteration for GD-MDP

Require: An $\mathcal{MDP}$
Ensure: Optimal policy π

1: Choose an initial proper policy π_0 arbitrarily
2: $i \leftarrow 0$
3: **while** $\pi_i \neq \pi_{i-1}$ **do**
4: Policy evaluation: obtain the value of the current policy π_i for every $s \in \mathcal{S}$ by solving the system of equations in Equation 1.
5: Policy Improvement: improve the current policy by doing the following update for every $s \in \mathcal{S}$:

$$\pi_{i+1}(s) \leftarrow \arg\min_{a\in\mathcal{A}} \left[c(s,a) + \sum_{s'\in\mathcal{S}} T(s,a,s') V^{\pi_i}(s')\right].$$

6: $i \leftarrow i + 1$

If π_0 is a proper policy, PI algorithm finds an optimal policy [5].

C. Using the Discount Factor to Ensure the Existence of Optimal Policies

A common trick to ensure the existence of optimal policies and the convergence of Policy Iteration algorithm is to consider a discount factor $\gamma < 1$. In this case equation 2 becomes $\mathbf{V}^\pi = \mathbf{c}^\pi + \gamma \mathbf{T}^\pi \mathbf{V}^\pi$. Besides being a mathematical trick, the discount factor can be used in two ways: (i) Mode 1: as a discount in the future cost, or (ii) Mode 2: as a chance of being alive for another step, that is equivalent to reaching a goal state. If this last meaning is taking into account, the condition for the existence of an optimal policy becomes $\rho(\gamma \mathbf{T}^\pi_{\mathcal{G}^c}) < 1$, which is always true when $\gamma < 1$ since $\rho(\gamma \mathbf{T}^\pi_{\mathcal{G}^c}) = \gamma\rho(\mathbf{T}^\pi_{\mathcal{G}^c})$ and $\rho(\mathbf{T}^\pi_{\mathcal{G}^c}) \leq 1$.

Finally, if the cost is constant for any state not in the goal set $\mathcal{G}$, we can show that the optimal policy is optimal under a utility function given by [3]:

$$u(C^\pi) = -\mathrm{sgn}(\ln(\gamma))\exp(\ln(\gamma) C^\pi)$$

which is risk prone if $\gamma < 1$, but is risk averse if we allow $\gamma > 1$.

III. Goal-Directed Risk Sensitive Markov Decision Process

A GD-RSMDPs [8] is defined by the tuple $\mathcal{RSMDP} = \langle \mathcal{MDP}, \lambda \rangle$ where $\mathcal{MDP}$ is a GD-MDP and λ is the risk-attitude factor. GD-RSMDPs consider the utility function:

$$u(x) = -\text{sgn}(\lambda)\exp(\lambda x),$$

and model arbitrary risk attitude by considering a risk-attitude factor λ. If $\lambda < 0$ the agent considers a risk-prone attitude, if $\lambda > 0$ the agent considers a risk-averse attitude and in the limit if $\lambda \to 0$ the agent considers a risk-neutral attitude [9].

In GD-RSMDPs, the value function of a policy π is defined by:

$$V^{\pi}(s_0) = \lim_{M\to\infty} \text{E}\left[-\text{sgn}(\lambda)\exp\left(\lambda \sum_{t=0}^{M} c\left(s_t, \pi(s_t)\right)\right)\right],$$

Similar to GD-MDPs, the value of a policy π can be calculated by solving the following system of equations:

$$V^{\pi}(s) = \begin{cases} -\text{sgn}(\lambda), \text{ if } s \in \mathcal{G} \\ \exp(\lambda c(s,\pi(s))) \sum_{s' \in \mathcal{S}} T(s,\pi(s),s')V^{\pi}(s'), \\ \qquad\qquad \text{otherwise} \end{cases} \tag{3}$$

or in its vector-matrix form:

$$\mathbf{V}^{\pi} = (\mathbf{D}^{\pi})^{\lambda}(\mathbf{T}^{\pi}_{\mathcal{G}^c}\mathbf{V}^{\pi} - \text{sgn}(\lambda)(\mathbf{1} - \mathbf{T}^{\pi}_{\mathcal{G}^c}\mathbf{1})), \tag{4}$$

where $\mathbf{1}$ is a column vector with ones and $\mathbf{D}^{\pi}$ is a $|\mathcal{S}| \times |\mathcal{S}|$ diagonal matrix with elements in $\exp(\mathbf{c}^{\pi})$. Matrices $\mathbf{V}^{\pi}$, $\mathbf{c}^{\pi}$, and $\mathbf{T}^{\pi}_{\mathcal{G}^c}$ were previously defined.

Definition 3: **(λ-feasible policy)** A policy π is λ-feasible if the probability of not being in an absorbing state vanishes faster than the exponential accumulated cost, i.e., $\lim_{t\to\infty}((\mathbf{D}^{\pi})^{\lambda}\mathbf{T}^{\pi}_{\mathcal{G}^c})^t = \mathbf{0}$. Equivalently, a policy π is λ-feasible if the spectral radius of $(\mathbf{D}^{\pi})^{\lambda}\mathbf{T}^{\pi}_{\mathcal{G}^c}$ is less than 1, i.e., $\rho((\mathbf{D}^{\pi})^{\lambda}\mathbf{T}^{\pi}_{\mathcal{G}^c}) < 1$ [4].

Similar to GD-MDP, there is also a Policy Iteration algorithm for GD-RSMDPs [4] (Algorithm 2).

Algorithm 2 Policy Iteration algorithm for GD-RSMDPs

Require: A $\mathcal{RSMDP}$

Ensure: Optimal policy π and the respective λ

1: Choose an initial λ-feasible policy π_0 arbitrarily
2: $i \leftarrow 0$
3: **while** $\pi_i \neq \pi_{i-1}$ **do**
4: Policy evaluation: obtain the value of the current policy π_i for every $s \in \mathcal{S}$ solving the system of equations in Equation 3.
5: Policy Improvement: improve the current policy by doing the following update for every $s \in \mathcal{S}$

$$\pi_{i+1}(s) \leftarrow \arg\min_{a\in\mathcal{A}}\left[\exp(\lambda c(s,a)) \sum_{s'\in\mathcal{S}} T(s,a,s')V^{\pi_i}(s')\right]$$

6: $i \leftarrow i + 1$

If there exists an optimal policy and π_0 is a λ-feasible policy, PI algorithm finds an optimal policy [4]. When $\lambda < 0$ (risk prone) and the policy π is proper, then π is also λ-feasible. However, this is not guaranteed for all policies when $\lambda > 0$ (risk averse). Given a GD-RSMDP, no result exists on how to determine the set of $\lambda > 0$ such that exists a λ-feasible policy.

In the next section, we show how to obtain a stationary extreme risk-averse policy for GD-MDPs with discount factor $\gamma > 1$ and for GD-RSMDP with risk factor $\lambda > 0$.

IV. Extreme Risk-Averse Algorithm

Although GD-RSMDPs allow modeling risk attitude, there is no work in the literature that explains how to set a risk-averse attitude appropriately. In this section, given a risk-averse attitude (modeled by the factor $\lambda > 0$ in a GD-RSMDP or factor $\gamma > 1$ in a GD-MDP), we would like algorithms to determine if there exists an optimal policy for such attitude. If there exists, the algorithm would find the optimal policy, and if there not exists, the algorithm would find the highest risk-averse policy. Those problems are defined bellow as the problem of founding γ-Extreme Risk-Averse Policy or λ-Extreme Risk-Averse Policy.

Definition 4: **(γ-Extreme Risk-Averse Policy)** Given any policy π, we can find an extreme value γ^{π} such that:

$$\rho(\gamma^{\pi}\mathbf{T}^{\pi}_{\mathcal{G}^c}) = 1$$

Then, the γ-Extreme Risk-Averse Policy is given by:

$$\pi^{\gamma^{\top}} = \arg\max_{\pi\in\Pi} \gamma^{\pi}.$$

Definition 5: **(λ-Extreme Risk-Averse Policy)** Given any policy π, we can find an extreme value λ^{π} such that:

$$\rho((\mathbf{D}^{\pi})^{\lambda^{\pi}}\mathbf{T}^{\pi}_{\mathcal{G}^c}) = 1.$$

Then, the λ-Extreme Risk-Averse Policy is given by:

$$\pi^{\lambda^{\top}} = \arg\max_{\pi\in\Pi} \lambda^{\pi}.$$

We present two algorithms, one for GD-MDP and other for GD-RSMDP; both algorithms depend on a parameter ϵ which represents the precision.

A. Algorithm for GD-MDP with $\gamma > 1$

Given a policy π it is possible calculate γ^{π} as in Definition 4 by:

$$\rho(\gamma^{\pi}\mathbf{T}^{\pi}_{\mathcal{G}^c}) = \gamma^{\pi}\rho(\mathbf{T}^{\pi}_{\mathcal{G}^c}) = 1 \Rightarrow \gamma^{\pi} = \frac{1}{\rho(\mathbf{T}^{\pi}_{\mathcal{G}^c})}.$$

Regarding π, we can formulate the following two opposite hypotheses: (*i*) $\pi = \pi^{\gamma^{\top}}$, and (*ii*) $\pi \neq \pi^{\gamma^{\top}}$. If the second hypothesis is true, there exists some $\epsilon > 0$ such that when we use $\gamma^{\pi,\epsilon} = \frac{1-\epsilon}{\rho(\mathbf{T}^{\pi}_{\mathcal{G}^c})}$, we found an optimal policy $\pi^{*}_{\gamma^{\pi,\epsilon}}$, where $\pi^{*}_{\gamma^{\pi,\epsilon}} \neq \pi$ and $\gamma^{\pi} < \gamma^{\pi^{*}_{\gamma^{\pi,\epsilon}}}$, i.e., $\pi^{*}_{\gamma^{\pi,\epsilon}}$ is γ-extremer than π.

Although we cannot define which ϵ will satisfy the condition to find an extremer policy, given a fixed value for ϵ, we can

improve γ while it is possible. We propose the following steps to find an approximated γ-extreme policy: (*i*) choose an initial policy, (*ii*) calculate a discount factor γ, (*iii*) evaluate such a policy; and (*iv*) improve on it. The last three steps are repeated until a better policy cannot be found (see Algorithm 3).

Algorithm 3 Policy Iteration for GD-MDP with $\gamma > 1$

Require: $(\mathcal{MDP}, \epsilon)$

Ensure: Optimal policy π and the respective γ

1: Choose an initial policy π_0 arbitrarily

2: $i \leftarrow 0$

3: **while** $\pi_i \neq \pi_{i-1}$ **do**

4: Update discount factor:

$$\gamma \leftarrow \frac{(1-\epsilon)}{\rho(\mathbf{T}^{\pi_i})}$$

5: Policy evaluation: obtain the value within the current policy π_i for every $s \in \mathcal{S}$ solving the following system of equations:

$$V^{\pi_i}(s) = c(s, \pi_i(s)) + \gamma \sum_{s' \in \mathcal{S}} T(s, \pi_i(s), s') V^{\pi_i}(s'), \forall s \in \mathcal{S}$$

6: Policy Improvement: improve the current policy by doing the following update for every $s \in \mathcal{S}$:

$$\pi_{i+1}(s) \leftarrow \arg\min_{a \in \mathcal{A}} \left[c(s,a) + \gamma \sum_{s' \in \mathcal{S}} T(s,a,s') V^{\pi_i}(s') \right]$$

7: $i \leftarrow i + 1$

B. Algorithm for GD-RSMDP with Risk Averse ($\lambda < 0$)

By following the same procedure in the previous section, given a policy π we must find $\lambda^{\pi,\epsilon}$ that solves:

$$\rho((\mathbf{D}^{\pi})^{\lambda^{\pi,\epsilon}} \mathbf{T}^{\pi}_{\mathcal{G}^c}) = 1 - \epsilon,$$

however, we cannot solve it analytically. Then, we consider a search procedure to find $\lambda^{\pi,\epsilon}$.

Consider that $\lambda^{\pi,\epsilon} = \lambda_0 + \delta_0$, where λ_0 is an initial guess and δ_0 is a positive variation in λ_0, we have:

$$\begin{aligned} \rho((\mathbf{D}^{\pi})^{\lambda_0+\delta_0} \mathbf{T}^{\pi}_{\mathcal{G}^c}) &= \rho((\mathbf{D}^{\pi})^{\lambda_0} (\mathbf{D}^{\pi})^{\delta_0} \mathbf{T}^{\pi}_{\mathcal{G}^c}) \\ &\leq \rho((\mathbf{D}^{\pi})^{\delta_0}) \rho((\mathbf{D}^{\pi})^{\lambda_0} \mathbf{T}^{\pi}_{\mathcal{G}^c}) \\ &= \max_{s \in \mathcal{S}} \exp(\delta_0 c(s)) \rho((\mathbf{D}^{\pi})^{\lambda_0} \mathbf{T}^{\pi}_{\mathcal{G}^c}) \end{aligned}$$

consider the following equality:

$$\max_{s \in \mathcal{S}} \exp(\delta_0 c(s)) \rho((\mathbf{D}^{\pi})^{\lambda_0} \mathbf{T}^{\pi}_{\mathcal{G}^c}) = (1 - \epsilon_1)$$

and take the logarithm on both sides of the equation, we have:

$$\begin{aligned} \max_{s \in \mathcal{S}} \delta_0 c(s) &= \ln(1-\epsilon_1) - \ln\rho((\mathbf{D}^{\pi})^{\lambda_0} \mathbf{T}^{\pi}_{\mathcal{G}^c}) \\ \delta_0 &= \frac{\ln(1-\epsilon_1) - \ln\rho((\mathbf{D}^{\pi})^{\lambda_0} \mathbf{T}^{\pi}_{\mathcal{G}^c})}{\max_{s \in \mathcal{S}} c(s)}. \end{aligned}$$

Let $\lambda_i = \lambda_{i-1} + \delta_{i-1}$ and set:

$$\delta_i = \frac{\ln(1-\epsilon_1) - \ln\rho((\mathbf{D}^{\pi})^{\lambda_i} \mathbf{T}^{\pi}_{\mathcal{G}^c})}{\max_{s \in \mathcal{S}} c(s)}.$$

then, we guarantee $\delta_i > 0$ and that:

$$\rho((\mathbf{D}^{\pi})^{\lambda_i+\delta_i} \mathbf{T}^{\pi}_{\mathcal{G}^c}) \leq \rho((\mathbf{D}^{\pi})^{\lambda_{i+1}+\delta_{i+1}} \mathbf{T}^{\pi}_{\mathcal{G}^c}) \leq 1 - \epsilon,$$

for all $i \geq 0$.

Given parameter $\beta > \epsilon$, we propose the following steps to find an approximated λ-extreme policy: (*i*) choose an initial $\lambda < 1$ and arbitrarily policy, (*ii*) updates λ while $\rho((\mathbf{D}^{\pi})^{\lambda} \mathbf{T}^{\pi}_{\mathcal{G}^c}) < 1 - \beta$, (*iii*) evaluate such a policy; and (*iv*) improve on it. The last three steps is repeated until a better policy cannot be found (see Algorithm 4).

Algorithm 4 Policy Iteration for GD-RSMDP with Risk Averse

Require: $\mathcal{RSMDP}$,ϵ,β)

Ensure: Optimal policy π and the respective λ

1: Choose an initial policy π_0 arbitrarily

2: $i \leftarrow 0$

3: **while** $\pi_i \neq \pi_{i-1}$ **do**

4: **while** $\rho((\mathbf{D}^{\pi})^{\lambda} \mathbf{T}^{\pi}_{\mathcal{G}^c}) \geq (1 - \beta)$ **do**

5: Update the risk-attitude factor:

$$\lambda \leftarrow \lambda + \frac{\ln(1-\epsilon) - \ln\rho((\mathbf{D}^{\pi})^{\lambda} \mathbf{T}^{\pi}_{\mathcal{G}^c})}{\max_{s \in S} c(s)}$$

6: Policy evaluation: obtain the value of the current policy π_i for every $s \in \mathcal{S}$ solving the system of equations in Equation 3.

7: Policy Improvement: improve the current policy by doing the following update for every $s \in \mathcal{S}$

$$\pi_{i+1}(s) \leftarrow \arg\min_{a \in \mathcal{A}} \left[\exp(\lambda c(s,a)) \sum_{s' \in \mathcal{S}} T(s,a,s') V^{\pi_i}(s') \right]$$

8: $i \leftarrow i + 1$

The algorithm does not stop if and only if there is a policy whose trajectories' lengh is less than a constant, i.e., exists $\tau < \infty$ such that $(\mathbf{T}^{\pi}_{\mathcal{G}^c})^{\tau} = 0$. However, this case is difficult to happen. In general, stochastic problems will always present some chance of taking one more step to accomplished a task, or, if there is a deterministic path, it rarely will be long.

V. Related Work

For solving risk-sensitive sequential decision problems, the objective is to maximize a risk-sensitive criterion such as: (i) expected exponential utility [8], [10], [11], [12], [4], (ii) variance-related measure [13], [14], (iii) percentil performance [15], or (iv) the probability that the cumulative cost is within some threshold [16], [17], [18]. In this work, we consider an expected exponential utility as a risk-sensitive criterion and as [4] we concentrate on risk aversion in Goal Directed Risk-Sensitive Markov Decision Process. [4] has already proved the conditions for the existence of a valid policy for this model. The contribution of our work is show how to build one valid policy and what is the maximum value for the risk factor.

An MDP where the objective is to maximize the criterion (iv) was recently revisited in the area of Planing in Artificial

Intelligence. In this model the objective is to find a policy that maximizes the probability that the cumulative cost of the policy is less or equal than some user-defined cost threshold [16]. In [16] a Value Iteration algorithm to solve this problem was introduced and the model was revisited by [17] for Goal-Directed MDPs. New algorithms that are faster than the Value Iteration algorithm proposed before were introduced in [17]. The same model with imperfect state information was addressed in [18]. In this work we also consider Goal-Directed MDPs as [17], however as we said before, we work with a different risk sensitive criterion.

In the area of reinforcement learning there are some works that consider expected exponential utility criterion [19], [20] and variance-related risk measure [21], [22]. Our algorithms work only when the model is known, i.e., we do not deal with reinforcement learning problems.

VI. Experiments

A. Driving License [3]

This scenario describes a candidate that wants to take his driving license, and he has two choices: take lessons or do the practical exam. However, the more lesson he takes, the greater is the chance to pass in the practical exam. The candidate wants to minimize his cost to take the driving license. The question for this problem is: how many hours of lessons he must take before taking the practical exam? The candidate pays a cost of $2 for each practical exam and the cost to have lessons is $1. The candidate can take at most 4 hours of lessons before each practical exam and can only accumulates a maximum of 10 hours of experience. The chance of being approved in the practical exam depends on the previous accumulated experience (x) and current number of lessons taken (y). The function that returns the probability to being approved in the practical exam is: $p(x, y) = 0.08x + 0.04y$.

To model this scenario, we use a GD-RSMDP with 11 states and 5 actions. The states $\{0, 1, ..., 10\}$ keep information of the number of hours accumulated before the current lessons and a goal state s_G which represents that the agent has been aproved. The actions $\{0, 1, ..., 4\}$ show the number of lessons to take before each practical exam. The cost function for any state $s \neq s_G$ is given by $c(s, a) = 2 + a$. The transition function for any state $s \neq s_G$ is given by:

$$T(s, a, s') = \begin{cases} 0.08s + 0.4a & , \text{ if } s' = s_G \\ 1 - (0.08s + 0.4a) & , \text{ if } s' = \min\{s + a, 10\} \\ 0 & , \text{ otherwise} \end{cases}$$

B. Values of γ updated by PI Algorithm for GD-MDP ($\gamma > 1$)

Figure 1 shows the values of γ at each iteration of the PI Algorithm for GD-MDP with $\gamma > 1$. The initial policy was the worst possible, i.e. do not do any lesson. In this case it was adopted $\gamma_0 = 0.99$ and $\epsilon = 0.01$. Note that, it is not necessary an initial proper policy. The algorithm returns the greatest value for γ, that was log(24.75)=3.2088 and the optimal policy 12. Although GD-MDP with $\gamma > 1$ are not risk-averse in general, the algorithm returns the most averse one, i.e., policy 12.

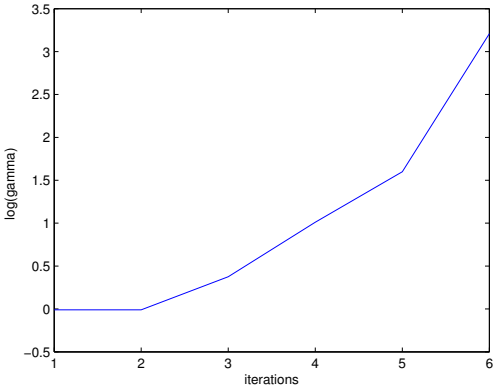

Fig. 1. Example of the execution o PI algorithm for MDP with $\gamma > 1$ in the driving license scenario.

C. Values of λ updated by PI Algorithm for GD-RSMDP with Risk Averse

Figure 2 shows the values of λ at each iteration of PI Algorithm for GD-RSMDP with Risk Averse. The initial policy was also the worst possible. In this case, it was adopted $\lambda_0 = -0.1$, $\epsilon = 0.001$ and $\beta = 0.0010000001$. Note that, for this algorithm it is not necessary a λ-feasible policy. The algorithm takes 221 iterations and returns the greatest value for λ that was 0.8042 and the optimal policy 6.

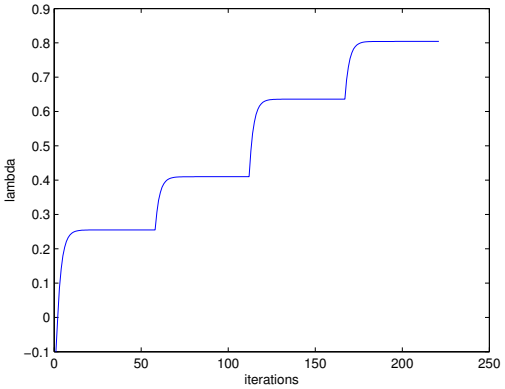

Fig. 2. Example of the execution of PI algorithm for GD-RSMDP with Risk Averse in the driving license scenario.

.

D. Behavior of Policies when λ gets Close to its Limit

To corroborate our results with the algorithm for GD-RSMDP with Risk Averse, we test systematically from riskier to safer policies. Table I shows each tested policy. For example, in policy 12, the action is to take the maximum number of hours of lesson permitted, i.e., 4 hours, for any state. This policy is the safest policy.

Figure 3 shows the certainty value of each policy for different values of the risk factor λ. We can see that each plateau in Figure 2 represents the limit for some policies in Figure 3, i.e., the maximum value of λ in both figures is also

the same. Figure 3 also shows the value goes to infinite when λ gets close to its limit, this is the typical behavior of policies. There are policies that are extremely risk averse, that are not optimal for any value of λ. For example, policies 7, 8, 9, 10, 11, 12 are never optimal.

TABLE I
EXAMPLES OF POLICIES FOR THE DRIVING LICENSE SCENARIO.

Policy States	0	1	2	3	4	5	6	7	8	9	10
Policy 1	4	4	3	2	1	0	0	0	0	0	0
Policy 2	4	4	4	3	2	1	0	0	0	0	0
Policy 3	4	4	4	4	3	2	1	0	0	0	0
Policy 4	4	4	4	4	4	3	2	1	0	0	0
Policy 5	4	4	4	4	4	4	3	2	1	0	0
Policy 6	4	4	4	4	4	4	4	3	2	1	0
Policy 7	4	4	4	4	4	4	4	4	3	2	1
Policy 8	4	4	4	4	4	4	4	4	4	3	2
Policy 9	4	4	4	4	4	4	4	4	4	3	1
Policy 10	4	4	4	4	4	4	4	4	4	3	2
Policy 11	4	4	4	4	4	4	4	4	4	4	3
Policy 12	4	4	4	4	4	4	4	4	4	4	4

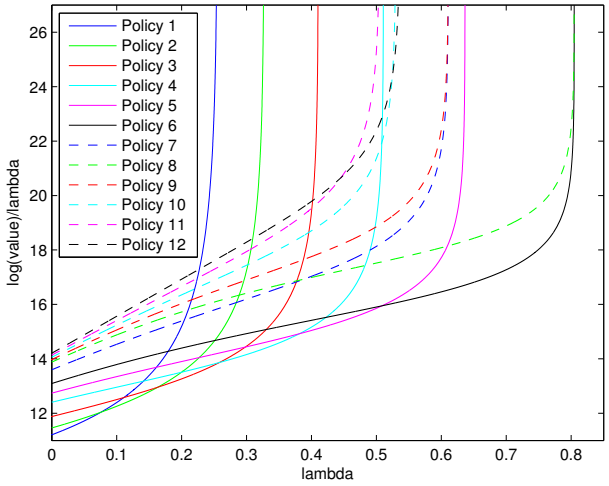

Fig. 3. The value of the 12 policies showed in Table I for different λ risk averse.

VII. CONCLUSION

We have proposed two versions of the PI algorithm, one for MDP with $\gamma > 1$ and one for GD-RSMDP with risk-averse. For the first one, it is no longer necessary an initial proper policy. For the second, it is no longer necessary an initial λ-feasible policy and now we can use any risk-averse attitude, i.e. any lambda value. In this case the algorithm can find the optimal policy if it exists, otherwise, the algorithm returs the highest feasible risk-averse policy. Both of the proposed algorithms depend on a parameter ϵ that guides the precision to which the task is accomplished. Note that, although the algorithm converges in general, the algorithm does not converge for the most risk averse feasible policy. However, we believe it does in most real problems.

A direction for future research is to consider the distribution for the initial state. It is possible that some initial states do not converge or converge slowly to optimal policies, but after an appropriate choice of λ they are never met after all feasible initial states. In this case, if we discard these states we can have policies with higher values of λ.

ACKNOWLEDGMENT

The authors would like to thank the Sao Paulo Research Foundation (FAPESP) for the financial support (grant #2015/01587-0).

REFERENCES

[1] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY: John Wiley and Sons, 1994.

[2] A. Ruszczyński, "Risk-averse dynamic programming for Markov decision processes," *Mathematical Programming*, vol. 125, no. 2, pp. 235–261, 2010.

[3] R. Minami and V. F. da Silva, "Shortest stochastic path with risk sensitive evaluation," in *Proc. of the eleventh Mexican Int. Conf. on Artif. Intell. (MICAI'12)*. Springer, 2012, pp. 370–381.

[4] S. D. Patek, "On terminating Markov decision processes with a risk averse objective function," *Automatica*, vol. 37, pp. 1379–1386, 2001.

[5] D. P. Bertsekas and J. N. Tsitsiklis, "An analysis of stochastic shortest path problems," *Mathematics of Operations Research*, vol. 16, no. 3, pp. 580–595, Aug. 1991. [Online]. Available: http://dx.doi.org/10.1287/moor.16.3.580

[6] H. Geffner and B. Bonet, "A concise introduction to models and methods for automated planning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 1, pp. 1–141, 2013.

[7] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York: Wiley, 1976.

[8] R. A. Howard and J. E. Matheson, "Risk-sensitive Markov decision processes," *Management Science*, vol. 18, no. 7, pp. 356–369, 1972.

[9] K. Sladký, "Growth rates and average optimality in risk-sensitive Markov decision chains," *Kybernetika*, vol. 44, no. 2, pp. 205–226, 2008. [Online]. Available: http://hdl.handle.net/10338.dmlcz/135844

[10] S. C. Jaquette, "A utility criterion for Markov decision processes," *Management Science*, vol. 23, no. 1, pp. 43–49, 1976.

[11] E. V. Denardo and U. G. Rothblum, "Optimal stopping, exponential utility, and linear programming," *Mathematical Programming*, vol. 16, no. 1, pp. 228–244, 1979.

[12] U. G. Rothblum, "Multiplicative Markov decision chains," *Mathematics of Operations Research*, vol. 9, no. 1, pp. 6–24, 1984.

[13] M. J. Sobel, "The variance of discounted Markov decision processes," *Journal of Applied Probability*, vol. 19, no. 4, pp. 794–802, 1982.

[14] J. A. Filar, L. C. M. Kallenberg, and H.-M. Lee, "Variance-penalized markov decision processes," *Mathematics of Operations Research*, vol. 14, no. 1, pp. 147–161, 1989.

[15] J. A. Filar, D. Krass, K. W. Ross, and K. W. Ross, "Percentile performance criteria for limiting average Markov decision processes," *IEEE Transactions on Automatic Control*, vol. 40, no. 1, pp. 2–10, 1995.

[16] S. X. Yu, Y. Lin, and P. Yan, "Optimization models for the first arrival target distribution function in discrete time," *Journal of Mathematical Analysis and Applications*, vol. 225, no. 1, pp. 193 – 223, 1998.

[17] P. Hou, W. Yeoh, and P. Varakantham, "Revisiting risk-sensitive MDPs: New algorithms and results," in *International Conference on Automated Planning and Scheduling*. AAAI Press, 2014.

[18] ——, "Solving risk-sensitive POMDPs with and without cost observations," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2016, pp. 3138–3144.

[19] V. S. Borkar, "A sensitivity formula for risk-sensitive cost and the actor–critic algorithm," *Systems & Control Letters*, vol. 44, no. 5, pp. 339 – 346, 2001.

[20] ——, "Q-learning for risk-sensitive control," *Mathematics of Operations Research*, vol. 27, no. 2, pp. 294–311, 2002.

[21] A. Tamar, D. Di Castro, and S. Mannor, "Policy gradients with variance related risk criteria," in *International conference on machine learning*, 2012, pp. 387–396.

[22] P. L.A. and M. Ghavamzadeh, "Actor-critic algorithms for risk-sensitive mdps," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 252–260.