Tuning the Discount Factor in Order to Reach Average Optimality on Deterministic MDPs

Filipo Studzinski Perotto(✉) and Laurent Vercouter

Normandy University / INSA / LITIS, Rouen, France
{filipo.perotto,laurent.vercouter}@litislab.fr

Abstract. Considering Markovian Decision Processes (MDPs), the meaning of an optimal policy depends on the optimality criterion chosen. The most common approach is to define the optimal policy as the one that maximizes the sum of discounted rewards. The intuitive alternative is to maximize the average reward per step. The former has strong convergence guarantees but suffers from the dependency on a discount factor. The latter has the additional inconvenience of being insensitive to different policies with equivalent average. This paper analyzes the impact of such different criteria on a series of experiments, and then provides a threshold for the discount factor in order to ensure average optimality for discounted-optimal policies in the deterministic case.

Keywords: Average dynamic programming
Sensitive dynamic programming · Reinforcement learning
Markovian decision process

1 Introduction

Dynamic Programming (DP) refers to a set of algorithms that can efficiently compute optimal policies for *Markovian Decision Processes* (MDPs), providing essential foundations for *Reinforcement Learning* (RL) methods [22,26]. DP and RL algorithms are fundamentally based on *discounted-optimality*. In this setting, an optimal policy maximizes the sum of discounted rewards over time using a discount factor γ.

When considering infinite time-horizon, the use of discounted rewards constitutes an important key on guarantying polynomial time convergence for such methods [2]. However, in many domains, the use of a discount factor does not present any relation to the optimization problem itself. Typically, when facing recurrent MDPs (where terminal states do not exist), discounting future rewards in favor of immediate rewards can introduce a kind of "distortion" on the real utility of a policy of actions [17,21,27].

The *crawling robot* problem [29] offers an illustrative example of such issue. The robot is endowed with a single articulated arm, and some of its movements

M. Bramer and M. Petridis (Eds.): SGAI-AI 2018, LNAI 11311, pp. 92–105, 2018.
https://doi.org/10.1007/978-3-030-04191-5_7

cause the displacement of the robot. The objective is finding the optimal cyclical sequence of actions in order to make the robot walk forward as fast as possible. Rewards correspond to immediate progressions. However, depending on the discount factor, a discounted-optimal policy can be unable to reach the maximum velocity. In other words, an intuitively optimal behavior can be seen as sub-optimal under the discounted framework. In fact, the robot reaches its maximum speed when it enters in the recurrent cycle of states that offers the highest average displacement per step. Other examples of this issue are given in [17]. For such scenarios, maximizing the average reward per step is, in some sense, more appropriate, but a key limitation of such approach is that *average-optimality* cannot distinguish among policies which have the same recurrent average reward per step, but which are not necessarily equivalent in terms of transient rewards [19].

The discussion about optimality is not new [4,8,14], and is summarized in Sect. 2. In practice, the discounted framework had been largely preferred. Such algorithms are easier to implement, and the polynomial convergence bounds are guaranteed for the general case [23,34].

In fact, if the discount factor γ is sufficiently high, *discounted-optimal* policies become also *average-optimal*. How high γ needs to be depends on each particular setting (topology and rewards), and cannot be calculated beforehand. For that reason, it is a hard-to-tune parameter. Without any other information, such average optimality is only guaranteed in the limit when $\gamma \to 1$ [3,4,12]. However, the higher the discount factor, the slower the convergence of iterative methods. When γ approaches 1, the necessary time for convergence approaches ∞ [34].

How often are discounted-optimal policies not average-optimal? The first contribution of this paper is an analysis on the difference between discounted-optimality and average-optimality in terms of total reward loss on the long run depending on how γ is tuned. Using a set of experiments with random MDPs we show that the impact of the use of too low γ values is not negligible.

How can the discount factor be optimally tuned in practice? The second contribution of this paper is a method for calculating a threshold for γ in order to ensure average-optimality to discounted-optimal policies. In this paper, as it consists on a first approach to the problem, only deterministic MDPs will be considered.

The rest of the paper is organized as follows. Section 2 reviews related concepts and methods on computational sequential decision-making. Section 3 presents our contributions: an analysis about the impact in terms of reward loss on choosing either discounted or average optimality, and the deduction of a formula for identifying average-optimal discount factors. Section 4 concludes the paper.

2 Background: Markovian Decision Processes

Markovian Decision Processes (MDPs) are in the center of a widely-used framework for approaching *automated control*, *sequential decision-making*, *planning*, and *computational reinforcement learning* problems [21,22,25,26,30,32].

An MDP works like a discrete stochastic finite state machine: at each time step the machine is in some state s, the agent observes that state and interacts with the process by choosing some action a to perform, then the machine changes into a new state s' and gives the agent a corresponding reward r.

An MDP can be defined as a set $\mathcal{M} = \{S, A, T, R\}$ in the form:

$$\mathcal{M} = \begin{cases} S = \{s_1, s_2, ..., s_n\} & \text{is the finite set of states} \\ A = \{a_1, a_2, ..., a_m\} & \text{is the finite set of actions} \\ T = \Pr(s'|s, a) & \text{is the transition function} \\ R = \Pr(r|s, a, s') & \text{is the reward function} \end{cases}$$

where $n = |S|$ is the number of states, and $m = |A|$ is the number of actions.

The transition function T defines the system dynamics by determining the next state s' given the current state s and the executed action a. The reward function R defines the immediate reward $r \in \mathbb{R}$ after moving from state s to s' with action a. *Deterministic MDPs* (D-MDPs) constitute the particular set of MDPs where the transitions are deterministic, in the form $T : S \times A \rightarrow S$.

Solving an MDP means finding a policy of actions that maximizes the rewards received by the agent, according to a given optimality criterion and a given time-horizon. The optimality criterion is defined by an utility function U. An optimal policy π^* is a policy that cannot be improved:

$$U(s, \pi^*) \geq U(s, \pi), \forall s \in S, \forall \pi \in \mathit{\Pi} \tag{1}$$

where $U(s, \pi)$ is the utility of following the policy π from the state s.

A deterministic stationary policy π is a mapping between states and actions in the form $\pi : S \rightarrow A$. The number of such policies contained in $\mathit{\Pi}$, the set of possible policies, is exponential, and corresponds to $|\mathit{\Pi}| = m^n$.

2.1 Discounted Optimality

When the stopping time h is finite and known, a simple solution consists in evaluating policies by estimating their *total rewards*. The utility function U is then equivalent to Z, the (undiscounted) sum of expected rewards:

$$Z_h(s, \pi) = \sum_{t=1}^{h} R_t(s, \pi) \tag{2}$$

where $R_t(s, \pi)$ corresponds to the expected reward in time t starting from state s and following policy π. In that case an exact optimal policy can be found through *backward recursion* [2] in polynomial time, $\mathcal{O}(nmh)$. However, such solution cannot be applied when the time-horizon is infinite, unbounded or unknown.

The standard approach to the infinite horizon setting consists in applying a *discount factor* $\{\gamma \in \mathbb{R} \mid 0 < \gamma < 1\}$ that reduces the weight of future rewards compared to immediate rewards in the sum. Such sum is always finite, which guarantees the convergence of iterative methods to an optimal solution [22,32].

The sum of discounted rewards $V_\gamma(s, \pi)$, starting in a given state s, following a given policy π, and for a given discount factor γ, is:

$$V_\gamma(s, \pi) = \lim_{h \to \infty} \sum_{t=1}^{h} \gamma^{t-1} R_t(s, \pi) \tag{3}$$

Tuning the discount factor implies a trade-off: the higher γ is (closer to 1), the better the chances of ensuring average optimality for discounted-optimal policies, but the bigger the computational costs for calculating the solution. The convergence time bound to compute discounted-optimal policies using iterative methods increases with rate $\mathcal{O}(\frac{1}{1-\gamma} \log \frac{1}{1-\gamma})$ [24,34]

Typical values of γ in the literature are 0.9 and 0.99. The inconvenience of using such generic suggestions is that, in certain circumstances, when γ is too low, discounted-optimality can lead the agent to a sub-optimal behavior in terms of average reward. We would like to call such phenomenon a "discount trap".

2.2 Average (or Gain) Optimality

In many domains, there is no grounded interpretation for the discount factor γ. In addition, the value corresponding to the sum of discounted rewards is less human readable (i.e. harder to interpret) than the average reward per step. Moreover, in recurrent domains (where later rewards are as important as earlier rewards) the use of low values of γ can "distort" the utility of some sequence of actions. For such reasons, maximizing the average reward received per time step can be preferable [9,28]. The *average reward* over an infinite time-horizon, called *gain*, of a policy π starting on state s, is:

$$G(s, \pi) = \lim_{h \to \infty} \frac{1}{h} \sum_{t=1}^{h} R_t(s, \pi) \tag{4}$$

The convenience of *average-optimality* compared to *discounted-optimality* can be observed regarding the MDPs shown in Fig. 1. On both problems, depending on the discount factor, the discounted-best policy can correspond to a clearly worse solution on the long run.

Considering unichain MDPs running over an infinite time-horizon, the average reward (or gain) of a given policy π converges to a single value g independently of the starting state [22], i.e. $G(s, \pi) = g, \forall s \in S$. Considering multichain MDPs, there is a convergent gain for each communicant subset of states (i.e. for each recurrent class within the process). The major drawback of average-optimal methods is that they have weaker convergence guarantees compared to discounted methods, even with the strong constraint of unichainess [11]. Worse yet, they are insensitive for distinguishing different policies with same average reward per step [15].

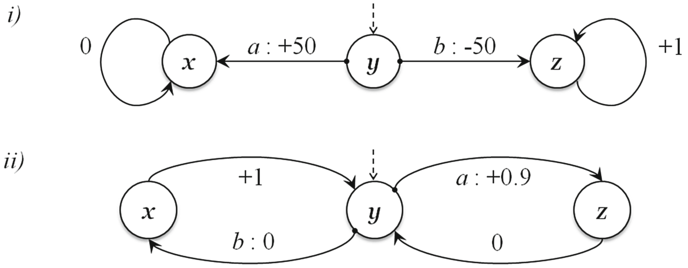

Fig. 1. In (*i*), there is a unique decision to be taken, on the middle state y. The gain of choosing the action a is $G(y, a) = 0$. It corresponds to the reward on the loop of the recurrent left state x. The gain of choosing the action b is $G(y, b) = +1$. It corresponds to the reward on the loop of the recurrent right state z. Because $G(y, b) > G(y, a)$, the action b constitutes the *average-optimal* policy. The action a earns a unique and immediate positive reward of 50 in time $t = 1$. At the same time, the action b loses 50, but then earns an additional reward of +1 per each subsequent time-step. In time $t = 101$, the policy b "reaches" the policy a, both having accumulated the same total rewards $Z_{101}(y, a) = Z_{101}(y, b) = +50$. Then, after 101 execution steps, b becomes better than a up to the infinity. So, considering an unbounded time-horizon (where the stopping time h of the process is likely to be greater than 100), the policy b would be preferred. However, given that $V_{0.99}(y, a) = +50$ and $V_{0.99}(y, b) = +49$, the discounted-optimal policy for any discount factor $\gamma \leq 0.99$ is a. **In (*ii*)**, the average of the policy starting on the middle state y and choosing the action a is $G(y, a) = +0.45$ (the average per step on the cycle $\{y, z\}$, on the right). The action b presents $G(y, b) = +0.5$ (the average per step on the cycle $\{y, x\}$, on the left) and is *gain-optimal*. However, given that $V_{0.9}(y, a) = V_{0.9}(y, b) \approx 4.737$, the discounted-optimal policy for any discount factor $\gamma < 0.9$ is a. In fact b becomes definitely better (i.e. get better total rewards) than a after 20 execution steps and up to the infinity.

2.3 Sensitive (or Blackwell) Optimality

For a same MDP there may be several *average-optimal* policies which are not necessarily equivalent. That is the case regarding the examples in Fig. 2. On both cases, two possible policies converge to a common average reward as time approaches infinity. They are, for that reason, indistinguishable from an average reward point of view.

The problem is that rewards obtained in the transient path toward the recurrent states disappear on the infinite averaging. In the same way, the position of each reward inside a sequence of cyclical rewards also disappears. However, such differences are important when considering an unbounded (but finite) time-horizon. Even though the gain of a policy π is mathematically independent of the starting state s on the infinite, the total expected reward in a given time h is not, i.e. $G(s, \pi) \neq \frac{1}{h} Z_h(s, \pi)$. Such differences are generally called *bias* [15,17,19,28].

In fact, for a given MDP, there is a discount factor γ^* from which the optimal policies do not change [3]. Such common "unanimous" optimal policies correspond to a *sensitive-optimality* [18,31].

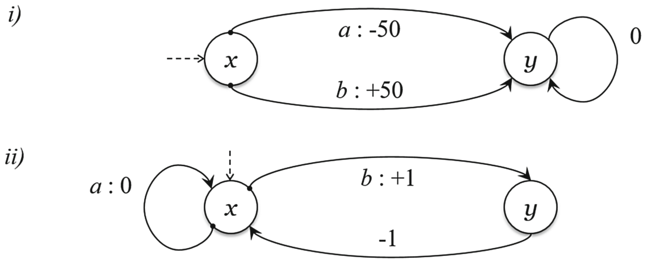

Fig. 2. In *(i)*, the gain of both possible policies is equivalent, $G(x,a) = G(x,b) = G(y) = 0$, but their initial steps (in $t = 1$) are not equivalently rewarded, $R_1(x,a) = -50$ and $R_1(x,b) = +50$. In this case, both policies a and b are *average-optimal*, but b offers a better transient reward and would be preferred over a. In fact, the total rewards accumulated by the policy b are greater than the rewards accumulated by a after the first step, i.e. $Z_h(x,b) > Z_h(x,a), \forall h \in \mathbb{Z} \mid h \geq 1$. **In *(ii)***, the gain of both possible policies is equivalent, $G(x,a) = G(x,b) = 0$, but the policy b presents a bigger total reward compared to a every time when t is odd. Then $Z_h(x,b) \geq Z_h(x,a), \forall h \in \mathbb{Z}^+$, and for such reason, b would be the preferred policy.

2.4 Dynamic Programming

Dynamic Programming (DP) refers to iterative optimization methods which can be used to efficiently compute optimal policies of Markovian Decision Processes (MDPs) when a model is given [2,22]. *Value-Iteration* (`VI`) [1] and *Policy-Iteration* (`PI`) [13] are the two fundamental and widely used DP algorithms for infinite time-horizon MDPs. It had been demonstrated that `PI` converges at least as quickly as `VI` [21], and, in practice, `PI` has been shown to be most effective [16].

There is a significant research effort for understanding the complexity of `PI`. The demonstration of its tight upper and lower bounds is still an open problem. Considering stochastic MDPs under discounted optimality, with a fixed discount rate $0 \leq \gamma < 1$, `PI` is proved to be *strongly polynomial* [10,23,34], i.e. the number of operations required to compute an optimal policy has an upper bound that is polynomial in the number of state-action pairs. Unfortunately, the convergence time increases with rate $\mathcal{O}(\frac{1}{1-\gamma}\log\frac{1}{1-\gamma})$ [24,34]. It constitutes a major impediment for using high discount factors ($\gamma \to 1$) in practice.

Typically, average optimization is a more difficult problem than discounted optimization [7]. `PI` can need an exponential number of iterations under average-optimality for stochastic MDPs in the general case [6]. In contrast, a deterministic MDP under average-optimality can be solved in strongly polynomial-time, $\mathcal{O}(n^2m)$ [11,20] as the well-known *Minimum-Mean-Cost-Cycle* problem. Experimental studies suggest that `PI` works very efficiently in this context [5]. Recent advances in average optimisation have been proposed in [33].

3 Contribution: Average-Optimal Discount Factor

Given that low discount factor values can lead to "discount traps", and that high discount factor values imply exponentially high computational costs, suggestions about how to tune such factor are needed. To the best of our knowledge, there is no method published in the literature for doing so.

The main contribution of this paper consists then in proposing a first threshold for γ in order to ensure average optimality to discounted-optimal policies, considering the case of deterministic MDPs, and based on simple characteristics of the target process. It means deducing a value for γ ensuring that discounted-optimal policies will correspond to average-optimal policies for any D-MDP which fits the given characteristics.

3.1 Tuning the Discount Factor

Let a *discount trap* be characterized by the situation where, for a given MDP $\mathcal{M}$, and for a given discount factor γ, there is a state s from where the gain of discounted-optimal policies is smaller than the gain of average-optimal policies. Formally, a *discount trap* exists if:

$$\exists \pi \in \mathit{\Pi}, \exists s \in S \begin{cases} V_\gamma(s, \pi^*) \geq V_\gamma(s, \pi) \\ G(s, \pi^*) < G(s, \pi) \end{cases} \tag{5}$$

Let a *family* $\mathcal{F}$ of deterministic MDPs be the set of all possible D-MDPs presenting the following identical characteristics:

$$\mathcal{F} = \begin{cases} n = |S| & \text{number of states} \\ m = |A| & \text{number of actions} \\ r_{\min} \in \mathbb{R} & \text{worst immediate reward value} \\ r_{\max} \in \mathbb{R} & \text{best immediate reward value} \\ \delta \in \mathbb{R}^+ & \text{the smallest non-zero immediate reward difference} \\ \mathit{\Delta} = r_{\max} - r_{\min} & \text{the range of the reward support} \end{cases}$$

In order to avoid discount traps, the discount factor must be tuned over a certain threshold. It is known that there exists an optimal discount factor γ^* from where the optimal policies do not change [3,12]. Such value can be called the *sensitive-optimal* discount factor. γ is guaranteed to be over such threshold on the limit when it approaches 1, i.e. $\lim \gamma \to 1 \implies \gamma > \gamma^*$. Another threshold, generally smaller than γ^*, ensures that discounted-optimal policies are also average-optimal. We would like to call it the *average-optimal* discount factor, and denote it $\gamma^{\bowtie}$.

When looking for the average-optimal discount factor $\gamma^{\bowtie}$ for a given family of D-MDPs $\mathcal{F}$, we must look for the worst case within the family, i.e. the process $\mathcal{M} \in \mathcal{F}$ which requires the highest value of γ to ensure average-optimality for discounted-optimal policies.

3.2 Deterministic Worst Case

Let an *optimal loop* L within a given D-MDP $\mathcal{M}$ be a cycle over a single state s, with some action a, presenting the maximum possible reward $r_{\max}$. Let an *almost optimal cycle* C be a cycle containing a subset of states, having a period $|C|$ that can vary from 1 (a single state) to $n-1$, and presenting a sequence of $|C|-1$ maximum rewards $r_{\max}$ followed by an almost maximum reward $r_{\max}-\delta$. Let a *maximally penalizing path* W be a path having a length $|W|$ that can vary from 1 (a single step) to $n-1$, and presenting the worst possible reward $r_{\min}$ on every step. When such 3 structures appear connected within a D-MDP, we discover a graph topology similar to the cases presented in Fig. 3.

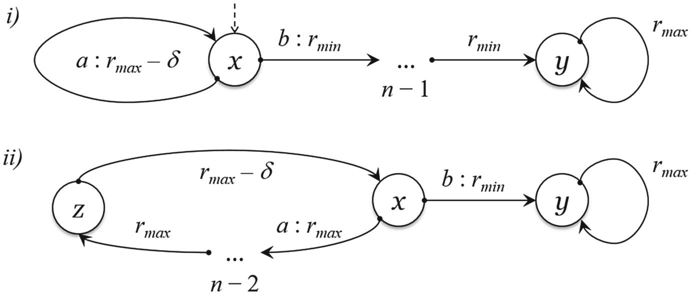

Fig. 3. In (*i*), the recurrent optimal loop on the state y can be reached by choosing the action b on state x. It offers a better gain than the almost optimal cycle reached with the action a in x, i.e. $G(x,b) = G(y) = R(y) = r_{\max} > G(x,a) = R(x,a) = r_{\max} - \delta$. However, a long and hardly penalizing transient path must be traversed in order to get from x to y. Such path counts the maximum possible distance between x and y, and is rewarded with the worst possible reward $r_{\min}$ at each step. The period of the almost optimal cycle is 1 (a loop), and the size of the maximally penalizing path is $n-1$. **In (*ii*)**, the almost optimal cycle constitutes the biggest possible cycle disjoint from the optimal loop in the MDP. Its period is $n-1$, which reduces the length of the path to 1. However, the gain difference is also smaller, $G(x,b) = r_{\max} > G(x,a) = r_{\max} - \frac{\delta}{n-1}$. Such are the two intuitively possible worst situations within a given MDP family for defining an average-optimal discount factor.

The infinite discounted sum of rewards on the almost optimal cycle C, denoted $V_\gamma(C)$, can be calculated by the difference between the infinite discounted sum of $r_{\max}$ and the infinite discounted sum of δ discounted by its position in the cycle:

$$V_\gamma(C) = \frac{r_{\max}}{1-\gamma} - \frac{\gamma^{|C|-1}\delta}{1-\gamma^{|C|}} \tag{6}$$

The infinite discounted sum of rewards on the path W followed by the loop L, denoted $V_\gamma(W)$, corresponds to the finite discounted sum of $r_{\min}$ on the maximally penalizing path plus the infinite discounted sum of $r_{\max}$ on the optimal loop discounted by the size of the path:

$$V_\gamma(W) = \frac{r_{\min}(1-\gamma^{|W|})}{1-\gamma} + \frac{r_{\max}\gamma^{|W|}}{1-\gamma} \tag{7}$$

When the discount factor is average-optimal, there is no discount trap.

Theorem 1. *Let $\mathcal{F}$ be a family of D-MDPs, corresponding to the set of processes presenting identical characteristics $n, m, r_{min}, r_{max}, \delta, \Delta$. The set of D-MDPs within such family which requires the highest average-optimal discount factor is characterized by: (i) an almost optimal cycle C disjoint from (ii) an optimal loop L, both separated by a unique (iii) maximally penalizing path W.*

The Theorem 1 necessarily holds because:

1. Shortening W increases $V_\gamma(W)$ without changing $V_\gamma(C)$.
2. Increasing $r_{\min}$ increases $V_\gamma(W)$ without changing $V_\gamma(C)$.
3. Increasing δ decreases $V_\gamma(C)$ without changing $V_\gamma(W)$.
4. Decreasing $r_{\max}$ makes $V_\gamma(C)$ decrease faster than $V_\gamma(W)$.

Developing the Theorem 1 results in n possible worst cases, from where the two extremes are illustrated in Fig. 3.

3.3 Deterministic Average-Optimal Discount Factor

In order to avoid a "discount trap", the utility of staying in an *almost optimal cycle* must be worse than the utility of traveling across all the states of the *maximally penalizing path* to the *optimal loop*. In the precedent section, two candidate worst cases have been presented. In this section the formula for an optimal discount factor is deduced for both examples. Such procedure allows to confirm what is effectively the worst case. An optimal discount factor for such worst case must necessarily be an optimal discount factor for any other case within the same MDP family. Such formulas can be deduced by simply developing the statement:

$$\gamma > \gamma^{\bowtie} \implies V_\gamma(W) > V_\gamma(C) \tag{8}$$

Firstly, we consider the case presented in Fig. 3 (i), which contains a long penalizing path. The value of the average-optimal discount factor $\gamma^{\bowtie}$ from which the discounted-optimal policies are also average-optimal for such D-MDP is:

$$\begin{aligned}
& V_\gamma(x,b) > V_\gamma(x,a) \\
\Longrightarrow \quad & \sum_{i=0}^{n-2} \gamma^i r_{\min} + \sum_{i=n-1}^{\infty} \gamma^i r_{\max} > \sum_{i=0}^{\infty} \gamma^i (r_{\max} - \delta) \\
\Longrightarrow \quad & \frac{(1-\gamma^{n-1}) r_{\min}}{1-\gamma} + \frac{\gamma^{n-1} r_{\max}}{1-\gamma} > \frac{r_{\max} - \delta}{1-\gamma} \\
\Longrightarrow \quad & r_{\min} - \gamma^{n-1} r_{\min} + \gamma^{n-1} r_{\max} > r_{\max} - \delta \\
\Longrightarrow \quad & \gamma^{n-1} r_{\max} - \gamma^{n-1} r_{\min} > r_{\max} - r_{\min} - \delta \\
\Longrightarrow \quad & \gamma^{n-1} \varDelta > \varDelta - \delta \\
\Longrightarrow \quad & \gamma > \sqrt[n-1]{1 - \frac{\delta}{\varDelta}}
\end{aligned}$$

Then we consider the case presented in Fig. 3 (*ii*), which contains a short penalizing path. The value of the average-optimal discount factor $\gamma^{\bowtie}$ from which the discounted-optimal policies are also average-optimal for such D-MDP is:

$$\begin{aligned}
& V_\gamma(x,b) > V_\gamma(x,a) \\
\Longrightarrow \quad & r_{\min} + \gamma V_\gamma(y) > V_\gamma(y) - \frac{\gamma^{n-2}\delta}{1-\gamma^{n-1}} \\
\Longrightarrow \quad & V_\gamma(y) - r_{\max} + r_{\min} > V_\gamma(y) - \frac{\gamma^{n-2}\delta}{1-\gamma^{n-1}} \\
\Longrightarrow \quad & -r_{\max} + r_{\min} > -\frac{\gamma^{n-2}\delta}{1-\gamma^{n-1}} \\
\Longrightarrow \quad & \frac{\gamma^{n-2}\delta}{1-\gamma^{n-1}} > \varDelta \\
\Longrightarrow \quad & \gamma^{n-2}\delta > \varDelta(1-\gamma^{n-1}) \\
\Longrightarrow \quad & \gamma^{n-2} > \frac{-\delta \pm \sqrt{\delta^2 + 4\varDelta^2}}{2\varDelta} \\
\Longrightarrow \quad & \gamma > \sqrt[n-2]{\frac{-\delta + \sqrt{\delta^2 + 4\varDelta^2}}{2\varDelta}}
\end{aligned}$$

Effectively, the second case is the worst case, which can be algebraically confirmed:

$$\begin{aligned}
& \sqrt[n-2]{\frac{-\delta + \sqrt{\delta^2 + 4\varDelta^2}}{2\varDelta}} > \sqrt[n-1]{1 - \frac{\delta}{\varDelta}} \\
\Longrightarrow \quad & \left(\frac{-\delta + \sqrt{\delta^2 + 4\varDelta^2}}{2\varDelta}\right)^2 > 1 - \frac{\delta}{\varDelta} \\
\Longrightarrow \quad & \frac{2\delta^2 - 2\delta\sqrt{\delta^2 + 4\varDelta^2} + 4\varDelta^2}{4\varDelta^2} > 1 - \frac{\delta}{\varDelta} \\
\Longrightarrow \quad & \frac{2\delta^2 - 2\delta\sqrt{\delta^2 + 4\varDelta^2}}{4\varDelta^2} > -\frac{\delta}{\varDelta} \\
\Longrightarrow \quad & 2\delta^2 - 2\delta\sqrt{\delta^2 + 4\varDelta^2} > -4\varDelta\delta \\
\Longrightarrow \quad & \delta - \sqrt{\delta^2 + 4\varDelta^2} > -2\varDelta \\
\Longrightarrow \quad & \delta - \sqrt{\delta^2 + 4k^2\delta^2} > -2k\delta \\
\Longrightarrow \quad & k > 0 \\
\Longrightarrow \quad & \frac{\varDelta}{\delta} > 0
\end{aligned}$$

Hence, the formula for calculating the optimal discount factor is:

$$\gamma^{\bowtie} = \sqrt[n-2]{\frac{-\delta + \sqrt{\delta^2 + 4\varDelta^2}}{2\varDelta}} \tag{9}$$

Figure 4 plots the function $\gamma^{\bowtie}$ for two different settings. Parameter i indicates the number of states, $n = i$, and the two different series represent reward granularity $\delta = 1/i$ and $\delta = 1$ (binary rewards).

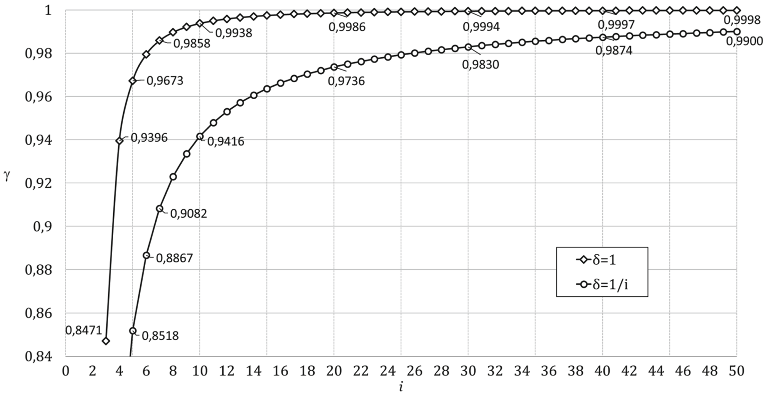

Fig. 4. The average-optimal discount factor for $n = i$ and reward support in $[0, 1]$. The two different series represent reward granularity $\delta = 1/i$ and $\delta = 1$ (Bernoulli).

3.4 Discount Trap Frequency

The choice of the optimality criterion has an impact on the calculated optimal policies. When using the discounted framework, if the discount factor γ is not sufficiently high, discounted-optimal policies could not correspond to gain-optimal policies, and it means a worst performance in terms of total rewards in the long run. In order to be able to measure the impact of such choice, we made a series of experiments, verifying how often a "discount trap" is detected.

Each experiment consists in generating 10000 random D-MDPs for a given setting (or family) i, varying the number of states $n = i$ and the reward granularity $\delta = {}^1/_i$. It means that the MDP size and the reward granularity are both gradually incremented. The number of actions is fixed to $m = 2$, as well as the minimum reward $r_{\min} = 0$ and the maximum reward $r_{\max} = 1$. We make the parameter i vary from 2 to 50. The results presented in Fig. 5 confirm that, for standard "naive" values of γ, like 0.9 and 0.99, the frequency of "discount traps" is not negligible, even for such small MDPs.

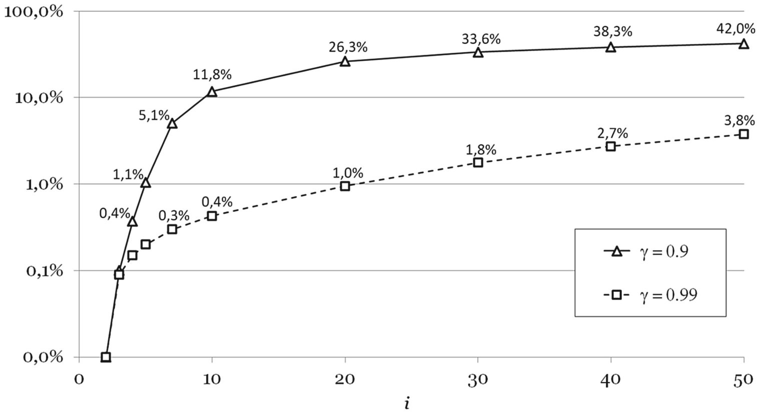

Fig. 5. The frequency of discount traps, when the discounted-optimal policy is not gain-optimal, considering $\gamma = 0.9$ and $\gamma = 0.99$. The parameter i indicates the number of states $n = i$ and the reward granularity $\delta = {}^1/_i$.

4 Conclusion

Using a set of experiences with randomly generated MDPs, we demonstrated that the occurrence of *discount traps*, inherent to all mechanisms that calculate utility functions using a discount factor, can cause sub-optimal behaviors on several recurrent MDPs in terms of total (undiscounted) rewards, and can be observed more often than usually suspected. In our experiments, we show that the use of "naive" but classical values for γ can result in discounted-optimal policies which are not average-optimal in 40% of the simulations when $\gamma = 0.9$, and almost 4% when $\gamma = 0.99$, which is far from being negligible.

In this paper, a formula for calculating an average-optimal discount factor is deduced, given the target family of deterministic MDPs characterized by n (the number of states), $r_{\max}$ and $r_{\min}$ (the reward bounds), and δ (the "reward granularity", equivalent to the smallest difference between any two rewards into the reward function). It represents an upper bound that could be improved by taking other characteristics into account. This paper was limited to deterministic MDPs. The next step of the work is understanding how such γ threshold can be defined on the stochastic case.

References

1. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton (1957)
2. Bertsekas, D.P.: Dynamic Programming and Optimal Control, 3rd edn. Athena Scientific, Belmont (2005)
3. Blackwell, D.: Discrete dynamic programming. Ann. Math. Stat. **33**(2), 719–726 (1962)
4. Cao, X.R., Zhang, J.: The n^{th}-order bias optimality for multichain Markov decision processes. Trans. Autom. Control **53**(2), 496–508 (2008)
5. Dasdan, A.: Experimental analysis of the fastest optimum cycle ratio and mean algorithms. Trans. Des. Autom. Electr. Syst. **9**(4), 385–418 (2004)
6. Fearnley, J.: Exponential lower bounds for policy iteration. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6199, pp. 551–562. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14162-1_46
7. Feinberg, E.A., Huang, J.: Strong polynomiality of policy iterations for average-cost MDPs modeling replacement and maintenance problems. Oper. Res. Lett. **41**(3), 249–251 (2013)
8. Feinberg, E.A., Huang, J.: Reduction of total-cost and average-cost MDPs with weakly continuous transition probabilities to discounted MDPs. Oper. Res. Lett. **46**(2), 179–184 (2018)
9. Gosavi, A.: A reinforcement learning algorithm based on policy iteration for average reward: empirical results with yield management and convergence analysis. Mach. Learn. **55**(1), 5–29 (2004)
10. Hansen, T.D., Miltersen, P.B., Zwick, U.: Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. J. ACM **60**(1), 1–16 (2013)
11. Hansen, T.D., Zwick, U.: Lower bounds for Howard's algorithm for finding minimum mean-cost cycles. In: Cheong, O., Chwa, K.-Y., Park, K. (eds.) ISAAC 2010. LNCS, vol. 6506, pp. 415–426. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17517-6_37
12. Hordijk, A., Yushkevich, A.: Blackwell optimality. In: Feinberg, E.A., Shwartz, A. (eds.) The Handbook of Markov Decision Processes: Methods and Applications, chap. 8, pp. 231–268. Kluwer (2002)
13. Howard, R.: Dynamic Programming and Markov Processes. MIT Press, Cambridge (1960)

14. Kallenberg, L.: Finite state and action MDPS. In: Feinberg, E.A., Shwartz, A. (eds.) Handbook of Markov Decision Processes. International Series in Operations Research and Management Science, vol. 40, pp. 21–87. Springer, Boston (2003). https://doi.org/10.1007/978-1-4615-0805-2_2
15. Lewis, M.E., Puterman, M.L.: Bias optimality. In: Feinberg, E.A., Shwartz, A. (eds.) The Handbook of Markov Decision Processes: Methods and Applications, chap. 3, pp. 89–111. Kluwer (2002)
16. Littman, M.L., Dean, T.L., Kaelbling, L.P.: On the complexity of solving Markov decision problems. In: Proceedings of the 11th UAI, p. 394402 (1994)
17. Mahadevan, S.: Average reward reinforcement learning: foundations, algorithms, and empirical results. Mach. Learn. **22**(1–3), 159–195 (1996)
18. Mahadevan, S.: Sensitive discount optimality: unifying discounted and average reward reinforcement learning. In: Saitta, L. (ed.) Proceedings of the 13th ICML, pp. 328–336. Morgan Kaufmann (1996)
19. Mahadevan, S.: Learning representation and control in Markov decision processes: new frontiers. Found. Trends Mach. Learn. **1**(4), 403–565 (2009)
20. Papadimitriou, C., Tsitsiklis, J.N.: The complexity of Markov decision processes. Math. Oper. Res. **12**(3), 441–450 (1987)
21. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (1994)
22. Puterman, M., Patrick, J.: Dynamic programming. In: Sammut, C., Webb, G. (eds.) Encyclopedia of Machine Learning, pp. 298–308. Springer (2010)
23. Kalyanakrishnan, S., Mall, U., Goyal, R.: Batch-switching policy iteration. In: Proceedings of the 25th IJCAI. AAAI Press (2016)
24. Scherrer, B.: Improved and generalized upper bounds on the complexity of policy iteration. Math. Oper. Res. **41**(3), 758–774 (2016)
25. Sigaud, O., Buffet, O. (eds.): Markov Decision Processes in Artificial Intelligence. iSTE - Wiley (2010)
26. Sutton, R., Barto, A.: Introduction to Reinforcement Learning. MIT Press, Cambridge (1998)
27. Tadepalli, P.: Average-reward reinforcement learning. In: Sammut, C., Webb, G. (eds.) Encyclopedia of Machine Learning, pp. 64–68. Springer (2010)
28. Tadepalli, P., Ok, D.: Model-based average reward reinforcement learning. Artif. Int. **100**(1–2), 177–224 (1998)
29. Tokic, M., Fessler, J., Ertel, W.: The crawler, a class room demonstrator for reinforcement learning. In: Lane, C., Guesgen, H. (eds.) Proceedings of the 22th FLAIRS, pp. 160–165. AAAI Press, Menlo Park (2009)
30. Uther, W.: Markov decision processes. In: Sammut, C., Webb, G. (eds.) Encyclopedia of Machine Learning, pp. 642–646. Springer (2010)
31. Veinott, A.: Discrete dynamic programming with sensitive discount optimality criteria. Ann. Math. Stat. **40**(5), 1635–1660 (1969)
32. van Otterlo, M., Wiering, M.: Reinforcement learning and Markov decision processes. In: Wiering, M., van Otterlo, M. (eds.) Reinforcement Learning. Adaptation, Learning, and Optimization, vol. 12, pp. 3–42. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27645-3_1
33. Yang, S., Gao, Y., An, B., Wang, H., Chen, X.: Efficient average reward reinforcement learning using constant shifting values. In: Proceedings of the 30th AAAI. AAAI Press/The MIT Press (2016)
34. Ye, Y.: The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. Math. Oper. Res. **36**(4), 593603 (2011)