# What is an Optimal Policy in Time-Average MDP?

Nicolas Gast, Bruno Gaujal, Kimang Khun

# What is an Optimal Policy in Time-Average MDP?

Nicolas Gast
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP
LIG, 38000 Grenoble, France
nicolas.gast@inria.fr

Bruno Gaujal
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP
LIG, 38000 Grenoble, France
bruno.gaujal@inria.fr

Kimang Khun
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP
LIG, 38000 Grenoble, France
kimang.khun@polytechnique.org

## ABSTRACT

This paper discusses the notion of optimality for time-average MDPs. We argue that while most authors claim to use the "average reward" criteria, the notion that is implicitly used is in fact the notion of what we call Bellman optimality. We show that it does not coincide with other existing notions of optimality, like gain-optimality and bias-optimality but has strong connection with canonical-policies (policies that are optimal for any finite horizons) as well as value iteration and policy iterations algorithms.

## 1. INTRODUCTION

Markov decision processes (MDPs) have been introduced in the 50s by Richard Bellman, and are still widely studied today with a huge regain of popularity as they are at the core of most reinforcement learning [6]. An MDP is a controlled Markov chain in which a decision maker wants to find the best "policy" in order to maximize a given reward criterion. Such a policy is called an optimal policy. The theory of MDPs provides very efficient algorithms to characterize and compute optimal policies. The definition of what is an optimal policy depends on the criterion that the decision maker wants to optimize. In this paper, we focus on the long-run average reward criterion, also called time-average, for which most authors define an optimal policy as a gain-optimal policy, that is, a policy that maximizes the long-time average reward. Yet, we argue that most authors studying time-average MDPs are using a stronger notion of optimality without acknowledging it. We call this notion *Bellman-optimality*. Essentially a policy is *Bellman-optimal* if it satisfies Bellman equation. We show that this notion does not coincide with gain-optimality nor $n$-bias-optimality. We recall the result of [7] that show that Bellman optimality coincides with optimality over all finite horizons (called *canonical-optimality*) and give arguments to show that classical algorithms (policy iteration or value iteration) output policies that are Bellman-optimal and not just gain-optimal. Note that this distinction only makes sense for undiscounted infinite-horizon MDPs: For finite-horizon or discounted MDPs, there is a unique notion of optimality.

## 2. INFINITE HORIZON MDPS

A MDP is a tuple $(\mathcal{S}, \mathcal{A}, p, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ is the transition kernel and $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the instantaneous reward. At any given time $t$, the controller observes the state $s_t \in \mathcal{S}$, chooses an action $a_t$ and earns an instantaneous reward $r(s_t, a_t)$. The system jumps to its next state $s_{t+1}$ according to the transition kernel. We assume $\mathcal{S}$ and $\mathcal{A}$ to be finite, and we denote by $p(j|s, a)$ the probability that the next state $s_{t+1}$ is $j$ given $s_t = s$ and $a_t = a$. A *policy* is a function $\pi : \mathcal{S} \to \mathcal{A}$ that specifies which action should be taken in the current state. For a given state $s$, we call $J_s^{\pi,T}$ the expected return over the first $T$ steps, when the initial state is $s$:

$$J_s^{\pi,T} = \mathbb{E}\left[\sum_{t=1}^{T} r(s_t, a_t) \mid s_1 = s, a_t = \pi(s_t)\right].$$

The gain of the policy when starting in $s$, $g_s^\pi$, is equal to the long-time average return:

$$g_s^\pi = \lim_{T\to\infty} \frac{1}{T} J_s^{\pi,T}. \tag{1}$$

We call the optimal gain, the quantity $g_s^* = \max_\pi g_s^\pi$. A policy $\pi$ is said to be *gain-optimal* if it maximizes the gain, *i.e*, if for all state $s$ and any policy $\pi'$, $g_s^\pi \geq g_s^{\pi'}$. When the state and action spaces are finite, such a policy exists [4].

### 2.1 Structure of an MDP

The long-term behavior of a MDP depends on the structure of the recurrent classes of its policies:
- A MDP is *recurrent* if all policies $\pi : \mathcal{S} \to \mathcal{A}$ induce a *recurrent* Markov chain.
- A MDP is *unichain* if all policies $\pi : \mathcal{S} \to \mathcal{A}$ induce a *unichain* Markov chain, *i.e.* a Markov chain with a unique recurrent class, plus a (possibly empty) set of transient states.
- A MDP is *weakly communicating* if there exists a subset of states $\mathcal{R} \subset \mathcal{S}$ such that (1) for all $s, s' \in \mathcal{R}$, there exists a path of positive probability of going from $s$ to $s'$, and (2) The states $\mathcal{S} \setminus \mathcal{R}$ are transient for all policies.

For a recurrent or unichain MDP, the gain of a policy does not depend on the initial state. For a weakly communicating MDP, the optimal gain does not depend on the initial state.

### 2.2 Characterization of gain-optimal policy

There exists a unique vector $g_s^* \in \mathbb{R}^{\mathcal{S}}$, equal to the optimal gain given in (2), and a non-unique bias vector $h \in \mathbb{R}^{\mathcal{S}}$ such that for all $s \in \mathcal{S}$ ([4, Chap. 9]):

$$g_s^* = \max_{a\in\mathcal{A}} \sum_j p(j|s, a) g_j^* \tag{2}$$

$$h_s + g_s^* = \max_{a\in\mathcal{A}} \{r(s, a) + \sum_{j\in\mathcal{S}} P(j|s, a) h_j\}. \tag{3}$$

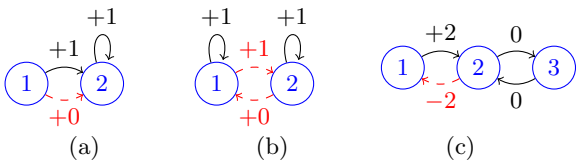


**Figure 1: Examples of MDPs. All transitions are deterministic and the labels on the edges indicate rewards. Each state has either only one action (solid black) or two actions (solid black / dashed red).**

The above system of equation is called the (modified) Bellman equation. It uniquely defines the gain $g^*$ but not $h$. We denote by $\mathcal{H}$ the set of functions $h$ that satisfy (2)-(3).

Most of the algorithms that compute gain-optimal policies starts by computing a (sometimes approximate) solution of (2)-(3) and then use this solution to define an optimal policy. Indeed, if $g^*, h$ are solutions of (2)-(3), the policy defined by

$$\pi(s) \in \arg\max_{a\in\mathcal{A}}\{r(s,a) + \sum_{j\in\mathcal{S}} P(j|s,a)h(j)\} \qquad (4)$$

is a gain-optimal policy.

## 2.3 Definition of Bellman–optimality

It is known that (4) is a sufficient but not necessary condition for being gain-optimal. Yet, when a paper mentions an "optimal policy" for the time-average reward, it is often implicitly assumed that such a policy should satisfy (4). The main reason for this is that it is (relatively) easy to design algorithms that provide a solution to (4).

Definition 1. *We say that a policy $\pi$ is* Bellman-optimal *if there exists $h \in \mathcal{H}$ such that $\pi$ satisfies (4) for this $h$.*

# 3. OTHER NOTIONS OF OPTIMALITY

## 3.1 Difference with gain-optimality

In Figure 1(a), we provide a simple example to illustrate the difference between gain-optimality and canonical-optimality. All the transitions of this MDP are deterministic. The MDP has two states (1 and 2). There are two possible actions in state 1: the "black" action earns +1 and transitions to state 2 while the action "red" earns +0 and also transitions to state 2. In state 2, the decision maker earns +1 and stays in state 2. For this example, all policies are gain-optimal because the choice of action in the transient state 1 does not affect the gain. The only canonical-optimal policy is the policy that takes the "black" action.

This example illustrates that a gain-optimal policy might take decision that are clearly suboptimal for transient states. In fact, as indicated by the following theorem, whose proof can be found in [3, Lemma 5.5], in [2] or in [5, Theorem 3.1e], a policy is gain-optimal if and only if it satisfies (4) for all of its recurrent state. The decision for the state that are transient under policy $\pi$ do not need to be canonical-optimal.

Theorem 1. *Let $\pi$ be a policy and let $\mathcal{R}^\pi$ be the set of recurrent states of the Markov chain induced by $\pi$. Then, $\pi$ is gain-optimal if and only if:*

- *For all $s \in \mathcal{S}$, $\pi(s)$ attains the max in (2).*
- *$\pi(s)$ satisfies (4) for all $s \in \mathcal{R}^\pi$.*

## 3.2 Difference with bias-optimality

Let $\pi$ be a policy. As indicated by the definition of the gain in (1), the expected return of the policy $\pi$ over the first $T$ time steps, $J_s^{\pi,T}$, is asymptotically equivalent to $Tg_s^\pi$. It can be shown in fact that for any initial state $s$, the Cesaro limit of $J_s^{\pi,T} - g_s^\pi$ converges to a quantity called the bias of the policy when starting in $s$, and that we denote by $h_s^\pi$:

$$h_s^\pi = \lim_{T\to\infty}\left(\frac{1}{T}\sum_{t=1}^{T} J_s^{\pi,t} - Tg_s^\pi\right).$$

In the above equation, the Cesaro-limit is necessary because of periodic Markov chains.

A policy is called *bias-optimal*[1] if it is gain-optimal and if for all state $s$ and all gain-optimal policy $\pi'$, we have $h_s^\pi \geq h_s^{\pi'}$. It is not hard to show that a bias-optimal policy is also Bellman-optimal. The converse is, however, not true.

For the example in Figure 1(b), the policy "(red,black)", that takes the action "red" in state 1 and "black" in state 2, is bias-optimal. It has a gain of 1 and a bias vector $(0, 0)$. The policy "(black,red)" is not bias-optimal because its bias is $(0, -1)$.However, this policy is still Bellman-optimal because it is a best-response to the bias-vector $(0, -1)$ that satisfies (2)-(3).

Note that the MDP is not unichain because it has two recurrent classes, $\{1\}$ and $\{2\}$. We argue that the distinction between Bellman-optimal and bias-optimal policies also occurs for unichain models, as shown in the example shown in Figure 1(c). There are exactly two policies, depending in which action is chosen in state 2. The model is unichain and the set $\mathcal{H}$ is the set of all vectors $(c + 2, c, c)$ for all $c \in \mathbb{R}$. Yet, the only bias-optimal policy is the policy that takes the action "black" in state 2. It has a bias $(2, 0, 0)$ whereas the policy "red" has a bias $(1, -1, -1)$ and is Bellman-optimal but not bias-optimal.

The examples in Figure 1 are all weakly communicating, and that examples (a) and (c) are unichain. Yet, none of these examples are recurrent. In fact, gain-optimality and bias-optimality coincide for recurrent MDPs (see [1, Sec. 3.1]). This implies that gain-optimality, canonical-optimality and bias-optimality, coincide in that case.

## 3.3 Finite Horizon Properties

In [7], policies that are uniformly optimal for any finite-time horizon (up a given final cost function) have been investigated and have been named *canonical-optimal* policies. More precisely, a policy $\pi$ is canonical-optimal if and only if there exists a final reward $f : \mathcal{S} \to \mathbb{R}$ such that for any finite-time horizon $T$, $\pi$ is optimal for the finite-horizon MDP $(\mathcal{S}, \mathcal{A}, p, r)$ with final cost $f$.

In the same paper [7], the authors show that canonical optimal policies exist and actually satisfy the Bellman equations for gain optimality (2)-(3). Using our definition their results actually translates into the following theorem:

Theorem 2 ([7]). *A policy $\pi$ is Bellman-optimal if and only if it is canonical-optimal.*

[1]The notion of bias-optimality is sometimes called 0-bias optimality or 0-sensitive optimality. There also exists stronger notions of optimality, called $n$-bias optimality, that are related to Blackwell-optimality. We refer to [4, Chapter 10] for a complete discussion.

# 4. CLASSICAL ALGORITHMS COMPUTE BELLMAN-OPTIMAL POLICIES

In this part, we explain that the two most popular generic-purpose algorithms to compute gain-optimal policies output policies that are Bellman-optimal and not just gain-optimal.

## 4.1 Value iteration

One of the most popular numerical algorithm to compute an optimal policy for MDP is value iteration (VI). This algorithm works as an iterative process. It starts by initializing a vector $V_s^{(0)}$ for all $s$. At iteration $k \geq 0$, it compute $V_s^{(k+1)}$ for all $s$ by using Bellman equation:

$$V_s^{(k+1)} = \max_{a \in \mathcal{A}} \left( r(s,a) + \sum_j p(j|s,a) V_j^{(k)} \right),$$

until some stopping criterion is met. For weakly communicating models, the stopping criteria is in general that the span of $V^{(k+1)} - V^{(k)}$ is smaller than some $\varepsilon > 0$ (see [4]):

$$\text{span}(V^{(k+1)} - V^{(k)}) \leq \varepsilon. \tag{5}$$

The algorithm outputs a policy such that for all $s \in \mathcal{S}$:

$$\pi_s \in \arg\max \left( r(s,a) + \sum_j p(j|s,a) V_j^{(k)} \right).$$

As shown below, value iteration outputs policies that are not just gain-optimal but satisfy a stronger optimality criterion.

THEOREM 3. *For any aperiodic unichain MDP, there exists $\varepsilon > 0$ such that if the stopping criteria of VI is (5):*
*- VI outputs a Bellman-optimal policy;*
*- Any Bellman-optimal policy can be the output of VI;*
*- If $V^{(0)}$ is set to 0, the output policy is bias-optimal.*

PROOF (SKETCH). If $V^{(0)} \in \mathcal{H}$, then value-iteration stops immediately and can output any Bellman-optimal policy by choosing the right value of $V^{(0)}$.

Let $V^{(0)} = 0$, then by construction, $V^{(k)} \geq J^{\pi,k}$ for any policy $\pi$, where $J^{\pi,k} = \sum_{i=0}^{k-1} P_\pi^i r^\pi$. Using the Bellman equations for $\pi$, $g^\pi = P_\pi g^\pi$ and $h^\pi + g^\pi = r^\pi + P_\pi h^\pi$, by multiplying by $P_\pi^i$ and summing, we get

$$J^{\pi,k} = k g^\pi + h^\pi - P_\pi^k h^\pi.$$

Note that (at least in the aperiodic case, that can be assumed w.l.o.g.) as $k$ goes to infinity, $P_\pi^k h^\pi \to P_\pi^\infty h^\pi = 0$.

Let $\pi^v$ be the policy output of VI. The stopping condition of VI implies $V_s^{(k)} - J_s^{\pi^v,k} \leq \varepsilon$ for all state $s$. Therefore, $\pi^v$ enjoys the following property:

$$\begin{aligned} J^{\pi^v,k} &= k g^{\pi^v} + h^{\pi^v} - P_{\pi^v}^k h^{\pi^v} \\ &\geq k g^{\pi^*} + h^{\pi^*} - P_{\pi^*}^k h^{\pi^*} - \varepsilon \mathbf{1}, \end{aligned}$$

where $\pi^*$ is a bias-optimal policy and $\mathbf{1}$ the vector whose components are all 1.

When $\varepsilon$ goes to 0, then the number of steps $k$ grows so that $P_{\pi^v}^k h^{\pi^v}$ and $P_{\pi^*}^k h^{\pi^*}$ both become negligible, so that $\pi^v$, the output of VI becomes a bias-optimal policy. □

## 4.2 Policy iteration algorithm

As the name suggests, policy iteration [4] iterates on policies. It starts with an arbitrary policy $\pi^{(0)}$. At iteration $k \geq 0$, policy iteration first finds a pair $(g^{(k)}, h^{(k)})$ that satisfies the Bellman equations:

$$g_s^{(k)} = \sum_j p(j|s, \pi^{(k)}(s)) g_j^{(k)} \tag{6}$$

$$h_s^{(k)} + g_s^{(k)} = r(s, \pi^{(k)}(s)) + \sum_{j \in \mathcal{S}} P(j|s, \pi(s)) h_j^{(k)}. \tag{7}$$

The algorithm then first tries to find a policy that improves (6): if it finds a pair $(s,a)$ such that $\sum_j p(j|s,a) g_j^{(k)} > g_s^{(k)}$, then it sets $\pi_s^{(k+1)} = \arg\max_{a \in \mathcal{A}} \sum_j p(j|s,a) g_j^{(k)}$ for this $s$ and keeps the same actions in the other states: $\pi_s^{(k+1)} = \pi_s^{(k)}$. It then moves to iteration $k+1$.

If no such pair is available, the algorithm then tries to find a policy that improves (7): if it finds a pair $(s,a)$ such that $\sum_j p(j|s,a) g_j^{(k)} = g_s^{(k)}$ and $r(s,a) + \sum_{j \in \mathcal{S}} P(j|s,a) h_j^{(k)} > h_s^{(k)} + g_s^{(k)}$, then it sets $\pi_s^{(k+1)} = \arg\max_{a \in \mathcal{A}}$ for this $s$ and keeps the other states unchanged. If no such pair is available, then the algorithm stops and outputs $\pi^{(k)}$.

It is shown in [4] that policy iteration outputs a gain-optimal policy in a finite number of steps. This statement can be refined by using Bellman-optimality:

THEOREM 4. *Policy iteration outputs a Bellman-optimal policy. Moreover, any Bellman-optimal policy can be the output of policy iteration.*

# 5. CONCLUSION

In this paper, we show that Bellman-optimality lies strictly between gain-optimality and bias-optimality:

$$\text{Gain-optimal} \supsetneq \text{Bellman-optimal} \supsetneq n\text{-bias-optimal}.$$

We also believe that while gain-optimal policies are easy to define, there is no natural algorithmic method to compute gain-optimal but not Bellman-optimal policies. This is because the later allows for a simple characterization. For some problems (such as indexability of restless bandit discussed in [2]), having a precise (and easy to characterize) definition of what is an optimal policy is essential. For such cases, the notion of Bellman-optimality is very natural.

# 6. REFERENCES


[1] E. A. Feinberg and A Shwartz. *Handbook of Markov Decision Processes: Methods and Applications*, volume 40. Springer US, 2002.

[2] Nicolas Gast, Bruno Gaujal, and Kimang Khun. Computing Whittle (and Gittins) index in subcubic time. *arXiv preprint arXiv:2203.05207*, 2022.

[3] Kimang Khun. *Indexability and Learning Algorithms for Markovian Bandits*. PhD thesis, 2023.

[4] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 2nd edition, 2005.

[5] Paul J Schweitzer and Awi Federgruen. The functional equations of undiscounted markov renewal programming. *Mathematics of Operations Research*, 3(4):308–321, 1978.

[6] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[7] AA Yushkevich. On a class of strategies in general markov decision models. *Theory of Probability & Its Applications*, 18(4):777–779, 1974.