HAL
open science

Policy Search: Any Local Optimum Enjoys a Global Performance Guarantee

Bruno Scherrer, Matthieu Geist

To cite this version:

Bruno Scherrer, Matthieu Geist. Policy Search: Any Local Optimum Enjoys a Global Performance Guarantee. 2013. hal-00829548

HAL Id: hal-00829548

https://inria.hal.science/hal-00829548v1

Preprint submitted on 6 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Policy Search: Any Local Optimum Enjoys a Global Performance Guarantee

Bruno Scherrer,
LORIA – MAIA project-team,
Nancy, France,
bruno.scherrer@inria.fr

Matthieu Geist,
Supélec – IMS-MaLIS Research Group,
Metz, France,
matthieu.geist@supelec.fr

June 6, 2013

Abstract

Local Policy Search is a popular reinforcement learning approach for handling large state spaces. Formally, it searches locally in a parameterized policy space in order to maximize the associated value function averaged over some predefined distribution. It is probably commonly believed that the best one can hope in general from such an approach is to get a local optimum of this criterion. In this article, we show the following surprising result: *any* (approximate) *local optimum* enjoys a *global performance guarantee*. We compare this guarantee with the one that is satisfied by Direct Policy Iteration, an approximate dynamic programming algorithm that does some form of Policy Search: if the approximation error of Local Policy Search may generally be bigger (because local search requires to consider a space of stochastic policies), we argue that the concentrability coefficient that appears in the performance bound is much nicer. Finally, we discuss several practical and theoretical consequences of our analysis.

1 Introduction

We consider the reinforcement learning problem formalized through Markov Decision Processes (MDP) (Sutton and Barto, 1998; Puterman, 1994), in the situation where the state space is large and approximation is required. On the one hand, Approximate Dynamic Programming (ADP) is a standard approach for handling large state spaces. It consists in mimicking in an approximate form the standard algorithms that were designed to optimize globally the policy (maximizing the associated value function for each state). On the other hand, Local Policy Search (LPS) consists in parameterizing the policy (the so-called "actor") and locally maximizing the associated expected value function, for example using a (natural) gradient ascent (Baxter and Bartlett, 2001; Kakade, 2001) (possibly with a critic (Sutton *et al.*, 1999; Peters and Schaal, 2008)), expectation-maximization (EM) (Kober and Peters, 2011), or even directly using some black-box optimization algorithm (Heidrich-Meisner and Igel, 2008).

The distinction we make here between ADP and LPS relies on the overall algorithmic scheme that is considered (dynamic programming or local expected value maximization). For example, we see the Direct (or Classification-based) Policy Iteration (DPI) algorithm (Lagoudakis and Parr, 2003; Fern *et al.*, 2006; Lazaric *et al.*, 2010) as belonging to the ADP family: even if it can be seen as a policy search method (since there is no representation for the value function), the algorithm follows the general approximate policy iteration (API) scheme. The Conservative Policy Iteration (CPI) algorithm (Kakade and Langford, 2002)—of which the analysis has close connections with what we are going to argue in this paper—might be considered at the frontier of ADP and LPS: it is based on a damped version of API, where each new policy is a convex mixture of the current policy and the greedy one, the precise combination being chosen such as guaranteeing an improvement in terms of the local fitness (the value function averaged over some predefined distribution).

Following the seminal works by Bertsekas and Tsitsiklis (1996), it has been shown that ADP algorithms enjoy global performance guarantees, bounding the loss of using the computed policy instead of using the optimal one as a function of the approximation errors involved along the iterations, for example for approximate policy iteration (API) (Munos, 2003), for approximate value iteration (AVI) (Munos,

2007), or more generally for approximate modified policy iteration (AMPI) (Scherrer *et al.*, 2012). To the best of our knowledge, similar general guarantees do not exist in the literature for LPS algorithms. Bounds have been derived for the CPI algorithm by Kakade and Langford (2002), and at first glance, one may think that this was due to its closeness to the ADP family. In general though, the best one can hope for LPS is to get a local optimum of the optimized fitness (that is, a local maximum of the averaged value function), and the important question of the loss with respect to the optimal policy remains open. As for instance mentionned as the main "future work" in Bhatnagar *et al.* (2007), where the convergence of a family of natural actor-critic algorithms is proven, "*[i]t is important to characterize the quality of converged solutions.*"

Experimentally, LPS methods seem to work pretty well. Applications to standard benchmarks (Baxter and Bartlett, 2001; Kakade, 2001; Peters and Schaal, 2008) and real applications such as robotics (Peters and Schaal, 2008; Kober and Peters, 2011) suggest that they are competitive with the ADP approach. Surprisingly, gradient-based and EM approaches, that are usually prone to be stuck in local optima, do not seem to be penalized in practice. Even more surprisingly, it was shown (Kakade, 2001) that a natural gradient ascent in the policy space can outperform ADP on the Tetris game. The motivation of this paper is to fill the theoretical gap on the LPS methods and to explain to some extent their empirical successes. Our main contribution (Theorem 3) is to show that *any (approximate) local optimum of the expected value function enjoys a global performance guarantee*, similar to the one provided by ADP algorithms. The proof technique we use is reminiscent of the one used for CPI, but our result is much more general and applies to a broad class of algorithms. Section 2 provides the necessary background and states formally what we mean by local policy search. Section 3 states and proves our main result. Section 4 discusses it. Notably, a comparison to similar bounds for ADP is proposed and the practical consequences of the result are discussed. Section 5 opens some perspectives.

2 Background and notations

Write Δ_X the set of probability distributions over a finite set X and Y^X the applications from X to the set Y. By convention, all vectors are column vectors, except distributions which are row vectors (for left multiplication). We consider a discounted MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, r, \gamma\}$ (Puterman, 1994; Bertsekas, 1995), with $\mathcal{S}$ the finite state space[1], $\mathcal{A}$ the finite action space, $P \in (\Delta_{\mathcal{S}})^{\mathcal{S}\times\mathcal{A}}$ the Markovian dynamics ($P(s'|s,a)$ denotes the probability of transiting to s' from the (s,a) couple), $r \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ the bounded reward function and $\gamma \in [0,1)$ the discount factor. A stochastic policy $\pi \in (\Delta_{\mathcal{A}})^{\mathcal{S}}$ associates to each state s a probability distribution $\pi(.|s)$ over the action space $\mathcal{A}$. For a given policy π, we write $r_\pi \in \mathbb{R}^{\mathcal{S}}$ defined as $r_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) r(s,a) = \mathbb{E}_{a\sim\pi(.|s)}[r(s,a)]$ and $P_\pi \in (\Delta_{\mathcal{S}})^{\mathcal{S}}$ defined as $P_\pi(s'|s) = \sum_{a\in\mathcal{A}} \pi(a|s) P(s'|s,a) = \mathbb{E}_{a\sim\pi(.|s)}[P(s'|s,a)]$. The value function v_π quantifies the quality of a policy π for each state s by measuring the expected cumulative reward received for starting in this state and then following the policy:

$$v_\pi(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r_\pi(s_t) | s_0 = s, s_{t+1} \sim P_\pi(.|s_t)\right].$$

The Bellman operator T_π of policy π associates to each function $v \in \mathbb{R}^{\mathcal{S}}$ the function defined as

$$[T_\pi v](s) = \mathbb{E}\left[r_\pi(s) + \gamma v(s') | s' \sim P_\pi(.|s)\right],$$

or more compactly $T_\pi v = r_\pi + \gamma P_\pi v$. The value function v_π is the unique fixed point of T_π.

It is well known that there exists a policy π_* that is optimal in the sense that it satisfies $v_{\pi_*}(s) \geq v_\pi(s)$ for all states s and policies π. The value function v_* is the unique fixed point of the following nonlinear Bellman equation:

$$v_* = T v_* \text{ with } T v = \max_{\pi\in\mathcal{A}^{\mathcal{S}}} T_\pi v$$

where the max is taken componentwise. Given any function $v \in \mathbb{R}^{\mathcal{S}}$, we say that a policy π' is greedy with respect to v if $T_{\pi'} v = T v$, and we write $\mathcal{G}(\pi)$ for the set of policies that are greedy with respect to the value v_π of some policy π. The notions of optimal value function and greedy policies are fundamental to

[1] It is straightforward to extend our results to the case of infinite state space and compact action space. We chose the finite space setting for the ease and clarity of exposition.

optimal control because of the following property: any policy π_* that is greedy with respect to the optimal value is an optimal policy and its value v_{π_*} is equal to v_*. Therefore, another equivalent characterization is that π is optimal if and only if it is greedy with respect to its value, that is if

$$\pi \in \mathcal{G}(\pi). \tag{1}$$

For any distribution μ, we define the γ-weighted occupancy measure[2] induced by the policy π when the initial state is sampled for μ as $d_{\mu,\pi} = (1-\gamma)\mu(I-\gamma P_\pi)^{-1}$ (we recall μ to be a row vector by convention) with $(I-\gamma P_\pi)^{-1} = \sum_{t\geq 0}(\gamma P_\pi)^t$. It can easily be seen that $\mu v_\pi = \frac{1}{1-\gamma} d_{\mu,\pi} r_\pi$. For any two distributions μ and ν, write $\left\|\frac{\mu}{\nu}\right\|_\infty$ the smallest constant C satisfying $\mu(s) \leq C\nu(s)$, for any $s \in S$ (this constant is actually the supremum norm of the componentwise ratio, thus the notation).

From an algorithmic point of view, Dynamic Programming methods compute the optimal value policy pair (v_*, π_*) in an iterative way. When the problem is large and cannot be solved exactly, Approximate Dynamic Programming (ADP) refers to noisy implementations of these exact methods, where the noise is due to approximations at each iteration. For instance, Approximate Value and Policy Iteration respectively correspond to the following schemes:

$$v_{k+1} = Tv_k + \epsilon_k \quad \text{and} \quad \begin{cases} v_k = v_{\pi_k} + \epsilon_k \\ \pi_{k+1} \in \mathcal{G}(v_k) \end{cases}.$$

In the Local Policy Search (LPS) context on which we focus in this paper, we further need to specify the space where we are going to perform the search. We write Π this set and assume that it is a *convex* subset of $(\Delta_\mathcal{A})^\mathcal{S}$. For a predefined distribution ν of interest, the problem addressed by LPS can be cast as follows:

$$\text{find } \pi \in \Pi \text{ such that } \pi \text{ is a local maximum of } J_\nu(\pi) = \mathbb{E}_{s\sim\nu}[v_\pi(s)].$$

Assume that we are able to (approximately) find such a locally optiaml policy π. A natural question is: how close is v_π to $v_* = v_{\pi_*}$? Quite surprisingly, and in contrast with most optimization problems, we will provide a generic answer to this question; this is the aim of the next section.

3 Main result

In order to state our main result, we need to first define precisely what we mean by approximate local optimum. For any pair of policies π and π' and coefficient $\alpha \in (0,1)$, we write $(1-\alpha)\pi + \alpha\pi'$ the stochastic mixture of π and π' with weights $(1-\alpha)$ and α.

Definition 1 (ϵ-local optimum)**.** *We say that a policy $\pi \in \Pi$ is an ϵ-local optimum of $J_\nu(\pi)$ if:*

$$\forall \pi' \in \Pi, \quad \lim_{\alpha\to 0} \frac{\nu v_{(1-\alpha)\pi+\alpha\pi'} - \nu v_\pi}{\alpha} \leq \epsilon.$$

This condition is for instance satisfied when $\|\nabla_\pi J_\nu(\pi)\|_\infty \leq \epsilon$, it states that the gradient is "small enough". Notice that the assumption of a convex policy space is necessary for this definition.

Then, we define a relaxation of the set of policies that are greedy with respect to some given policy.

Definition 2 (μ-weighted ϵ-greedy policies)**.** *We write $\mathcal{G}_\Pi(\pi,\mu,\epsilon)$ the set of policies which are ϵ-greedy respectively to π (in μ-expectation). It is formally defined as*

$$\mathcal{G}_\Pi(\pi,\mu,\epsilon) = \{\pi' \in \Pi \textit{ such that } \forall \pi'' \in \Pi,\ \mu T_{\pi'} v_\pi + \epsilon \geq \mu T_{\pi''} v_\pi\}.$$

This is indeed a relaxation of $\mathcal{G}$, as it can be observed that for all policies π and π',

$$\pi' \in \mathcal{G}(\pi) \quad \Leftrightarrow \quad \forall \mu \in \Delta_\mathcal{S},\ \pi' \in \mathcal{G}_\Pi(\pi,\mu,0) \quad \Leftrightarrow \quad \exists \mu \in \Delta_\mathcal{S},\ \mu > 0,\ \pi' \in \mathcal{G}_\Pi(\pi,\mu,0).$$

We are now ready to state the first important result, which links the ϵ-local optimality of Definition 1 to some relaxed optimality characterization involving Definition 2.

[2] When it exists, this measure tends to the stationary distribution of P_π when the discount factor tends to 1.

Theorem 1. *The policy $\pi \in \Pi$ is an ϵ-local maximum of $J_\nu(\pi)$ if and only if it is $(1-\gamma)\epsilon$-greedy respectively to itself, in $d_{\nu,\pi}$-expectation:*

$$\forall \pi' \in \Pi, \quad \lim_{\alpha \to 0} \frac{\nu v_{(1-\alpha)\pi + \alpha\pi'} - \nu v_\pi}{\alpha} \leq \epsilon \qquad \Leftrightarrow \qquad \pi \in \mathcal{G}_\Pi(\pi, d_{\nu,\pi}, (1-\gamma)\epsilon).$$

The following technical (but simple) lemma will be useful for the proof.

Lemma 1. *For any policies π and π', we have*

$$v_{\pi'} - v_\pi = (I - \gamma P_{\pi'})^{-1}(T_{\pi'} v_\pi - v_\pi).$$

Proof. The proof uses the fact that the linear Bellman Equation $v_\pi = r_\pi + \gamma P_\pi v_\pi$ implies $v_\pi = (I - \gamma P_\pi)^{-1} r_\pi$. Then,

$$\begin{aligned} v_{\pi'} - v_\pi &= (I - \gamma P_{\pi'})^{-1} r_{\pi'} - v_\pi = (I - \gamma P_{\pi'})^{-1}(r_{\pi'} + \gamma P_{\pi'} v_\pi - v_\pi) \\ &= (I - \gamma P_{\pi'})^{-1}(T_{\pi'} v_\pi - v_\pi). \end{aligned}$$

□

Proof of Theorem 1. For any α and any $\pi' \in \Pi$, write $\pi_\alpha = (1-\alpha)\pi + \alpha\pi'$. Using Lemma 1, we have:

$$\nu(v_{\pi_\alpha} - v_\pi) = \nu(I - \gamma P_{\pi_\alpha})^{-1}(T_{\pi_\alpha} v_\pi - v_\pi).$$

By observing that $r_{\pi_\alpha} = (1-\alpha) r_\pi + \alpha r_{\pi'}$ and $P_{\pi_\alpha} = (1-\alpha) P_\pi + \alpha P_{\pi'}$, it can be seen that $T_{\pi_\alpha} v_\pi = (1-\alpha) T_\pi v_\pi + \alpha T_{\pi'} v_\pi$. Thus, using the fact that $v_\pi = T_\pi v_\pi$, we get:

$$\begin{aligned} T_{\pi_\alpha} v_\pi - v_\pi &= (1-\alpha) T_\pi v_\pi + \alpha T_{\pi'} v_\pi - v_\pi \\ &= \alpha (T_{\pi'} v_\pi - v_\pi). \end{aligned}$$

In parallel, we have

$$\begin{aligned} (I - \gamma P_{\pi_\alpha})^{-1} &= (I - \gamma P_\pi + \alpha\gamma(P_{\pi'} - P_\pi))^{-1} \\ &= (I - \gamma P_\pi)^{-1}(I + \alpha M), \end{aligned}$$

where the exact form of the matrix M does not matter. Put together, we obtain

$$\nu(v_{\pi_\alpha} - v_\pi) = \alpha\nu(I - \gamma P_\pi)^{-1}(T_{\pi'} v_\pi - v_\pi) + o(\alpha^2).$$

Taking the limit, we obtain

$$\lim_{\alpha \to 0} \frac{\nu(v_{\pi_\alpha} - v_\pi)}{\alpha} = \nu(I - \gamma P_\pi)^{-1}(T_{\pi'} v_\pi - v_\pi) = (1-\gamma) d_{\nu,\pi}(T_{\pi'} v_\pi - v_\pi),$$

from which the stated result follows. □

With Theorem 1, we know that if the LPS algorithm has produced a policy π that is an ϵ-local maximum, then it satisfies for some distribution μ

$$\pi \in \mathcal{G}_\Pi(\pi, \mu, \epsilon), \tag{2}$$

that can be seen as a relaxed version of the original optimality characterization of Equation (1). As we are about to see, in the Theorem to come next, such a relaxed optimality characterization can be shown to imply a global guarantee. To state this result, we first need to define the "ν-greedy-complexity" of our policy space, which measure how good Π was designed so as to approximate the greedy operator, for a starting distribution ν.

Definition 3 (ν-greedy-complexity)**.** *We define $\mathcal{E}_\nu(\Pi)$ the ν-greedy-complexity of the policy space Π as*

$$\mathcal{E}_\nu(\Pi) = \max_{\pi \in \Pi} \min_{\pi' \in \Pi} \left(d_{\nu,\pi} \left(T v_\pi - T_{\pi'} v_\pi \right) \right).$$

It is clear that if Π contains any deterministic policy (a strong assumption), then $\mathcal{E}_\nu(\Pi) = 0$. Given this definition, we are ready to state our second important result.

Theorem 2. *If $\pi \in \mathcal{G}_\Pi(\pi, d_{\nu,\pi}, \epsilon)$, then for any policy π' and for any distribution μ over $\mathcal{S}$, we have*

$$\mu v_{\pi'} \leq \mu v_\pi + \frac{1}{(1-\gamma)^2} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty (\mathcal{E}_\nu(\Pi) + \epsilon).$$

Notice that this theorem is actually a slight[3] generalization of Theorem 6.2 of Kakade and Langford (2002). We provide the proof, that is essentially the same as that given in Kakade and Langford (2002), for the sake of completeness.

Proof. Using again Lemma 1 and the fact that $Tv_\pi \geq T_{\pi'} v_\pi$, we have

$$\mu(v_{\pi'} - v_\pi) = \mu(I - \gamma P_{\pi'})^{-1}(T_{\pi'} v_\pi - v_\pi) = \frac{1}{1-\gamma} d_{\mu,\pi'}(T_{\pi'} v_\pi - v_\pi) \leq \frac{1}{1-\gamma} d_{\mu,\pi'}(Tv_\pi - v_\pi).$$

Since $Tv_\pi - v_\pi \geq 0$ and $d_{\nu,\pi} \geq (1-\gamma)\nu$, we get

$$\mu(v_{\pi'} - v_\pi) \leq \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty \nu(Tv_\pi - v_\pi) \leq \frac{1}{(1-\gamma)^2} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty d_{\nu,\pi}(Tv_\pi - v_\pi).$$

Finally, using $d_{\nu,\pi}(Tv_\pi - v_\pi) = (d_{\nu,\pi} T v_\pi - d_{\nu,\pi} v_\pi)$, we obtain

$$\begin{aligned}
\mu(v_{\pi'} - v_\pi) &\leq \frac{1}{(1-\gamma)^2} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty (d_{\nu,\pi} T v_\pi - \max_{\pi' \in \Pi} d_{\nu,\pi} T_{\pi'} v_\pi + \max_{\pi' \in \Pi} d_{\nu,\pi} T_{\pi'} v_\pi - d_{\nu,\pi} v_\pi) \\
&\leq \frac{1}{(1-\gamma)^2} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty (\mathcal{E}_\nu(\Pi) + \epsilon)
\end{aligned}$$

□

The main result of the paper is a straightforward combination of the results of both Theorems.

Theorem 3. *Assume that the policy π is an ϵ-local optimum of $J_\nu(\pi)$ over Π, that is*

$$\forall \pi' \in \Pi, \quad \lim_{\alpha \to 0} \frac{J_\nu(\pi_\alpha) - J_\nu(\pi)}{\alpha} \leq \epsilon \qquad (\textit{with } \pi_\alpha = (1-\alpha)\pi + \alpha\pi'),$$

then, π enjoys the following global performance guarantee:

$$0 \leq \mathbb{E}_{s \sim \mu}[v_*(s) - v_\pi(s)] \leq \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi_*}}{\nu} \right\|_\infty \left(\frac{\mathcal{E}_\nu(\Pi)}{1-\gamma} + \epsilon \right).$$

4 Discussion

We have just shown that *any policy search algorithm* that is able to estimate any ϵ-close local optimum of the fitness function $J_\nu(\pi) = \mathbb{E}_{s \sim \nu}[v_\pi(s)]$ *actually comes with a global performance guarantee.* In this section, we discuss the relations of our analyses with previous works, we compare this guarantee with the standard ones of approximate dynamic programming (focusing particularly on the DPI algorithm) and we discuss some practical and theoretical consequences of our analysis.

4.1 Closely related analysis

Though the main result of our paper is Theorem 3, and since Theorem 2 appears in a very close form in Kakade and Langford (2002), our main technical contribution is Theorem 1 that highlights a deep connection between local optimality and a relaxed Bellman optimality characterization. A result, that is similar in flavor, is derived by Kakade (2001) for the Natural Policy Gradient algorithm: Theorem 3 there shows that natural gradient updates are moving the policy towards the solution of a (DP) update. The author even writes: "The natural gradient could be efficient far from the maximum, in that it is pushing the policy toward choosing greedy optimal actions". Though there is an obvious connection with our work, the result there is limited since 1) it seems to be specific to the natural gradient approach

[3] Theorem 2 holds for any policy π', not only for the optimal one, and the error term is split up (which is necessary to provide a more general result).

(though our result is very general), and 2) it is not exploited so as to connect with a global performance guarantee.

A performance guarantee very similar to the one we provide in Theorem 2 was first derived for CPI in Kakade and Langford (2002). The main difference is that the term $(\mathcal{E}_\nu(\Pi) + \epsilon)$ of Theorem 2 is replaced there by some global precision ϵ, that corresponds to the error made by a classifier that is used as an approximate greedy policy chooser. Similarly to the work we have just mentioned on the Natural Policy Gradient, this result of the literature was certainly considered specific to the CPI algorithm, that has unfortunately not been used widely in practice probably because of its somewhat complex implementation. In contrast, we show in this paper that such a performance guarantee is valid for any method that finds a policy that satisfies a relaxed Bellman identity like that given Equation (2), among which one finds many widely used methods that do Local Policy Search.

4.2 Relations to bounds of approximate dynamic programming

The performance guarantee of any approximate dynamic programming algorithm implies *(i)* a (quadratic) dependency on the average horizon $\frac{1}{1-\gamma}$, *(ii)* a concentration coefficient (which quantifies the divergence between the worst discounted average future state distribution when starting from the measure of interest, and the distribution used to control the estimation errors), and *(iii)* an error term linked to the estimation error encountered at each iteration (which can be due to the approximation of value functions and/or policies). Depending on what quantity is estimated, a comparison of these estimation errors may be hard. To ease the comparison, the following discussion focuses on the Direct Policy Iteration algorithm that does some form of policy search. Note however that several aspects of our comparison holds for other ADP algorithms.

Direct Policy Iteration (DPI) (Lagoudakis and Parr, 2003; Lazaric *et al.*, 2010) is an approximate policy iteration algorithm where at each iteration, (i) the value function is estimated for a set of states using Monte Carlo rollouts and (ii) the greedy policy (respectively to the current value function) is approximately chosen in some predefined policy space (through a weighted classification problem). Write $\mathcal{P}$ this policy space, which is typically a set of *deterministic policies*. For an initial policy π_0 and a given distribution ν, the DPI algorithms iterates as follows:

$$\text{pick } \pi_{k+1} \in \mathcal{P} \text{ such as (approximately) minimizing } \nu(Tv_{\pi_k} - T_{\pi_{k+1}} v_{\pi_k}).$$

To provide the DPI bound, we need an alternative concentration coefficient as well as some new error characterizing $\mathcal{P}$. Let $C^*_{\mu,\nu}$ be the concentration coefficient defined as

$$C^*_{\mu,\nu} = (1-\gamma)^2 \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma^{i+j} \sup_{\pi \in \mathcal{A}^{\mathcal{S}}} \left\| \frac{\mu (P_{\pi_*})^i (P_\pi)^j}{\nu} \right\|_\infty .$$

We need also a measure of the complexity of the policy space $\mathcal{P}$, similar to $\mathcal{E}_\nu$:

$$\mathcal{E}'_\nu(\mathcal{P}) = \max_{\pi \in \mathcal{P}} \min_{\pi' \in \mathcal{P}} (\nu(Tv_\pi - T_{\pi'} v_\pi))$$

Let also e be an estimation error term, which depends on the number of samples and which tends to zero as the number of samples tends to infinity (at a rate depending on the chosen classifier). The performance guarantee of DPI (Lazaric *et al.*, 2010; Ghavamzadeh and Lazaric, 2012) can be expressed as follows:

$$\limsup_{k \to \infty} \mu(v^* - v_{\pi_k}) \leq \frac{C^*_{\mu,\nu}}{(1-\gamma)^2} (\mathcal{E}'_\nu(\mathcal{P}) + e).$$

This bound is to be compared with the result of Theorem 3, regarding the three terms involved: the average horizon, the concentration coefficient and the greedy error term. Each term is discussed now, a brief summary being provided in Table 1. As said in Section 4.1, the LPS bound is really similar to the CPI one, and the CPI and DPI bounds have been extensively compared in (Ghavamzadeh and Lazaric, 2012). Our discussion can be seen as complementary.

Horizon term. Both bounds have a quadratic dependency on the average horizon $\frac{1}{1-\gamma}$. For approximate dynamic programming, this bound can be shown to be tight (Scherrer *et al.*, 2012), the only known solution to improve this being to introduce non-stationary policies (Scherrer and Lesner, 2012).

Table 1: Comparison of the performance guarantees for LPS and DPI

	bounded term	horizon term	concentration term	error term
LPS	$\mu(v_* - v_\pi)$	$\frac{1}{(1-\gamma)^2}$	$\left\lVert \frac{d_{\mu,\pi_*}}{\nu} \right\rVert_\infty$	$\mathcal{E}_\nu(\Pi) + \epsilon(1-\gamma)$
DPI	$\limsup_{k\to\infty} \mu(v^* - v_{\pi_k})$	$\frac{1}{(1-\gamma)^2}$	$C^*_{\mu,\nu}$	$\mathcal{E}'_\nu(\mathcal{P}) + e$

The tightness of this bound for policy search is an open question. However, we suggest later in Section 4.3 a possible way to improve this.

Concentration coefficients. Both bounds involve a concentration coefficient. Even if they are different, they can be linked.

Theorem 4. *We always have that:* $\left\lVert \frac{d_{\mu,\pi_*}}{\nu} \right\rVert_\infty \leq \frac{1}{1-\gamma} C^*_{\mu,\nu}$. *Moreover, if there always exists a* ν *such that* $\left\lVert \frac{d_{\mu,\pi_*}}{\nu} \right\rVert_\infty < \infty$ *(with* $\nu = d_{\mu,\pi_*}$*), there might not exist a* ν *such that* $C^*_{\mu,\nu} < \infty$.

Proof. Let us first consider the inequality. By using the definition of d_{μ,π_*} and eventually the fact that $d_{\mu,\pi_*} \geq (1-\gamma)\nu$, we have

$$\begin{aligned}
C^*_{\mu,\nu} &= (1-\gamma)^2 \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \gamma^{i+j} \sup_{\pi \in (\Delta_{\mathcal{A}})^{\mathcal{S}}} \left\lVert \frac{\mu (P_{\pi_*})^i (P_\pi)^j}{\nu} \right\rVert_\infty \\
&\geq (1-\gamma)^2 \left\lVert \sum_{i,j=0}^{\infty} \gamma^{i+j} \frac{\mu (P_{\pi_*})^{i+j}}{\nu} \right\rVert_\infty = (1-\gamma) \left\lVert \sum_{i=0}^{\infty} \gamma^{i} \frac{d_{\mu,\pi_*} (P_{\pi_*})^{i}}{\nu} \right\rVert_\infty \\
&\geq (1-\gamma)^2 \left\lVert \sum_{i=0}^{\infty} \gamma^{i} \frac{\mu (P_{\pi_*})^{i}}{\nu} \right\rVert_\infty = (1-\gamma) \left\lVert \frac{d_{\mu,\pi_*}}{\nu} \right\rVert_\infty .
\end{aligned}$$

For the second part of the theorem, consider an MDP with N states and N actions, with $\mu = \delta_1$ being a dirac on the first state, and such that from here action $a \in [1; N]$ leads in state a deterministically. Write $c = \sup_{\pi \in \mathcal{A}^{\mathcal{S}}} \lVert \frac{\mu P_\pi}{\nu} \rVert_\infty$ the first term defining $C^*_{\mu,\nu}$. For any π, we have $\mu P_\pi \leq c\nu$. Thus, for any action a we have $\delta_a \leq c\nu \Rightarrow 1 \leq c\nu(a)$. Consequently, $1 = \sum_{i=1}^N \nu(i) \geq \frac{1}{c}\sum_{i=1}^N 1 \Leftrightarrow c \geq N$. This being true for arbitrary $N \in \mathbb{N}$, we get $c = \infty$ and thus $C^*_{\mu,\nu} = \infty$. □

The first part of this result was already stated in Ghavamzadeh and Lazaric (2012), for the comparison of CPI (which involves the same concentration as LPS) and DPI. The second part is new: it tells that we may have $\left\lVert \frac{d_{\mu,\pi_*}}{\nu} \right\rVert_\infty \ll C^*_{\mu,\nu}$, which is clearly in favor of LPS (and CPI, as a side effect).

Error terms. Both bounds involve an error term. The terms ϵ (LPS) and e (DPI) can be made arbitrarily small by increasing the computational effort (the time devoted to run the algorithm and the amount of samples used), though not much more can be said without studying a specific algorithmic instance (*e.g.*, type of local search for LPS or type of classifier for DPI). The terms defining the "greedy complexity" of policy spaces can be more easily compared. Because they use different distributions that can be compared ($d_{\nu,\pi} \geq (1-\gamma)\nu$), we have for all policy spaces Π,

$$\mathcal{E}'_\nu(\Pi) \leq \frac{\mathcal{E}_\nu(\Pi)}{1-\gamma}.$$

However, this result (already stated in Ghavamzadeh and Lazaric (2012)) does not take into account the fact that LPS (or CPI for the discussion of Ghavamzadeh and Lazaric (2012)) works with *stochastic policies* while DPI works with *deterministic policies*. For example, assume that Π is the convex closure of $\mathcal{P}$. In this case, we would have $\mathcal{E}'_\nu(\mathcal{P}) \leq \mathcal{E}'_\nu(\Pi)$. Therefore, this error term would be more in favor of DPI than LPS: the search space is presumably smaller (while possibly allowing to represent the same deterministic greedy policies).

4.3 Practical and theoretical consequences of our analysis

Finally, this section provides a few important consequences of our analysis and of Theorem 3 in particular.

Rich policy and equivalence between local and global optimality. If the the policy space is very rich, one can easily show that any local optimum is actually global (this result being a direct corollary of Theorem 3).

Theorem 5. *Let $\nu > 0$ be a distribution. Assume that the policy space is rich in the sense that $\mathcal{E}_\nu(\Pi) = 0$, and that π is an (exact) local optimum of J_ν ($\epsilon = 0$). Then, we have $v_\pi = v_*$.*

If this result is well-known in the case of tabular policies, it is to our knowledge new in such a general case (acknowledging that $\mathcal{E}_\nu(\Pi) = 0$ is a rather strong assumption).

Choice of the sampling distribution. Provided the result of Theorem 3, and as also mentionned about CPI in Kakade and Langford (2002) since it satisfies a similar bound, if one wants to optimize the policy according to a distribution μ (that is, such that $\mu(v_* - v_\pi)$ is small), then one should optimize the fitness J_ν with the distribution $\nu \simeq d_{\mu,\pi_*}$ (so as to minimize the coefficient $\left\| \frac{d_{\mu,\pi_*}}{\nu} \right\|_\infty$). Ideally, one should sample states based on trajectories following the optimal policy π_* starting from states drawn according to μ (which is not surprising). This is in general not realistic since we do not know the optimal policy π_*, but practical solutions may be envisioned.

First, this means that one should sample states in the "interesting" part of the state space, that is where the optimal policy is believed to lead. This is a natural piece of information that a domain expert should be able to provide and this is in general much easier than actually controlling the system with the optimal policy (or with a policy that leads to these interesting parts of the state space). Also, though we leave the precise study of this idea for future research, a natural practical approach for setting the distribution ν would be to compute a sequence of policies $\pi_1, \pi_2, \ldots$ such that for all i, π_i is a local optimum of $\pi \mapsto J_{d_{\nu,\pi_{i-1}}}(\pi)$, that is of the criterion weighted by the region visited by the previous policy π_{i-1}. It may particularly be interesting to study whether the convergence of such an iterative process leads to interesting guarantees.

One may also notice that Theorem 3 may be straightforwardly written more generally for any policy. If π is an ϵ-local optimum of J_ν over Π, then for any stochastic policy π' we have

$$\mu v_{\pi'} \leq \mu v_\pi + \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty \left(\frac{\mathcal{E}_\nu(\Pi)}{1-\gamma} + \epsilon \right).$$

Therefore, one can sample trajectories according to an acceptable (and known) controller π' so as to get state samples to optimize $J_{d_{\nu,\pi'}}$. More generally, if we know where a good policy π' leads the system to from some initial distribution μ, we can learn a policy π that is guaranteed to be approximately as good (and potentially better).

A better learning problem? With the result of Theorem 3, we have a squared dependency of the bound on the effective average horizon $\frac{1}{1-\gamma}$. For approximate dynamic programming, it is known that this dependency is tight (Scherrer and Lesner, 2012). At the current time, this is an open question for policy search. However, we can somehow improve the bound. We have shown that the ϵ-local optimality of a policy π implies that it satisfies a relaxed Bellman global optimality characterization, $\pi \in \mathcal{G}_\Pi(\pi, d_{\nu,\pi}, \epsilon)$, which in turns implies Theorem 3. The following result, involving a slightly simpler relaxed Bellman equation, can be proved similarly to Theorem 2:

$$\pi \in \mathcal{G}_\Pi(\pi, \nu, \epsilon) \quad \Leftrightarrow \quad \mu v_{\pi'} \leq \mu v_\pi + \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty (\mathcal{E}_\nu(\Pi) + \epsilon).$$

A policy satisfying the left hand side would have an improved dependency on the horizon ($\frac{1}{1-\gamma}$ instead of $\frac{1}{(1-\gamma)^2}$). At the current time, we do not know whether there exists an efficient algorithm for computing a policy satisfying $\pi \in \mathcal{G}_\Pi(\pi, \nu, \epsilon)$. The above guarantee suggests that solving such a problem may improve over traditional policy search approaches.

5 Conclusion

In the past years, local policy search algorithms have been shown to be practical viable alternatives to the more traditional approximate dynamic programming field. The derivation of global performance guaran-

tees for such approaches, probably considered as a desperate case, was to our knowledge never considered in the literature. In this article, we have shown a surprising result: *any Local Policy Search algorithm*, as long as it is able to *provide an approximate local optimum* of $J_\nu(\pi)$, *enjoys a global performance guarantee* similar to the ones of approximate dynamic programming algorithms. From a theoretical viewpoint, there is thus no reason to prefer approximate dynamic programming over policy search (practical reasons – *e.g.*, necessity of a simulator – or other theoretical reasons – *e.g.*, rate of convergence – may come in line).

Since the bounds of ADP are known to be tight, the question whether the guarantee we have provided is tight constitutes an interesting future research direction. We suggested that it may be a better learning strategy to look for a policy π satisfying $\pi \in \mathcal{G}_\Pi(\pi, \nu, \epsilon)$ instead of searching for a local maximum of J_ν, as it leads to a better bound. Designing an algorithm that would do so efficiently is another interesting perspective. Finally, we here only considered pure actor algorithms, with only a parameterization of the policy. The extension of our analysis to situations where one also uses a critic (a parameterization of the value function) is a natural track to explore.

References

Baxter, J. and Bartlett, P. L. (2001). Infinite-horizon gradient-based policy search. *Journal of Artificial Intelligence Research*, **15**, 319–350.

Bertsekas, D. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control*. Athena Scientific.

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. (2007). Incremental natural actor-critic algorithms. In *Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada.

Fern, A., Yoon, S., and Givan, R. (2006). Approximate Policy Iteration with a Policy Language Bias: Solving Relational Markov Decision Processes. *Journal of Artificial Intelligence Research*, **25**, 75–118.

Ghavamzadeh, M. and Lazaric, A. (2012). Conservative and Greedy Approaches to Classification-based Policy Iteration. In *Conference on Artificial Intelligence (AAAI)*.

Heidrich-Meisner, V. and Igel, C. (2008). Evolution strategies for direct policy search. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*, pages 428–437.

Kakade, S. (2001). A Natural Policy Gradient. In *Neural Information Processing Systems (NIPS)*, pages 1531–1538.

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*.

Kober, J. and Peters, J. (2011). Policy Search for Motor Primitives in Robotics. pages 171–203.

Lagoudakis, M. G. and Parr, R. (2003). Reinforcement learning as classification: Leveraging modern classifiers. In *International Conference on Machine Learning*, pages 424–431.

Lazaric, A., Ghavamzadeh, M., and Munos, R. (2010). Analysis of a classification-based policy iteration algorithm. In *International Conference on Machine Learning*, pages 607–614.

Munos, R. (2003). Error bounds for approximate policy iteration. In *International Conference on Machine Learning*, pages 560–567.

Munos, R. (2007). Performance bounds in Lp norm for approximate value iteration. *SIAM J. Control and Optimization*.

Peters, J. and Schaal, S. (2008). Natural Actor-Critic. *Neurocomputing*, **71**, 1180–1190.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience.

Scherrer, B. and Lesner, B. (2012). On the Use of Non-Stationary Policies for Stationary Infinite-Horizon Markov Decision Processes. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1835–1843.

Scherrer, B., Gabillon, V., Ghavamzadeh, M., and Geist, M. (2012). Approximate Modified Policy Iteration. In *International Conference on Machine Learning (ICML)*.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning, An introduction*. BradFord Book. The MIT Press.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999). Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Neural Information Processing Systems (NIPS)*, pages 1057–1063.