On the Computability of AIXI

Jan Leike
Australian National University
jan.leike@anu.edu.au

Marcus Hutter
Australian National University
marcus.hutter@anu.edu.au

Abstract

How could we solve the machine learning and the artificial intelligence problem if we had infinite computation? Solomonoff induction and the reinforcement learning agent AIXI are proposed answers to this question. Both are known to be incomputable. In this paper, we quantify this using the arithmetical hierarchy, and prove upper and corresponding lower bounds for incomputability. We show that AIXI is not limit computable, thus it cannot be approximated using finite computation. Our main result is a limit-computable ε-optimal version of AIXI with infinite horizon that maximizes expected rewards.

Keywords. AIXI, Solomonoff induction, general reinforcement learning, computability, complexity, arithmetical hierarchy, universal Turing machine.

1 INTRODUCTION

Given infinite computation power, many traditional AI problems become trivial: playing chess, go, or backgammon can be solved by exhaustive expansion of the game tree. Yet other problems seem difficult still; for example, predicting the stock market, driving a car, or babysitting your nephew. How can we solve these problems in theory? A proposed answer to this question is the agent AIXI [Hut00, Hut05]. As a *reinforcement learning agent*, its goal is to maximize cumulative (discounted) rewards obtained from the environment [SB98].

The basis of AIXI is Solomonoff's theory of learning [Sol64, Sol78, LV08], also called *Solomonoff induction*. It arguably solves the induction problem [RH11]: for data drawn from a computable measure μ, Solomonoff induction will converge to the correct belief about any hypothesis [BD62, RH11]. Moreover, convergence is extremely fast in the sense that Solomonoff induction will make a total of at most $E + O(\sqrt{E})$ errors when predicting the next data points, where E is the number of errors of the informed predictor that knows μ [Hut01]. While learning the environment according to Solomonoff's theory, AIXI selects actions by running an expectimax-search for maximum cumulative discounted rewards. It is clear that AIXI can only serve as an ideal, yet recently it has inspired some impressive applications [VNH$^+$11].

Both Solomonoff induction and AIXI are known to be incomputable. But not all incomputabilities are equal. The *arithmetical hierarchy* specifies different levels of computability based on *oracle machines*: each level in the arithmetical hierarchy is computed by a Turing machine which may query a halting oracle for the respective lower level.

We posit that any ideal for a 'perfect agent' needs to be *limit computable* (Δ^0_2). The class of limit computable functions is the class of functions that admit an *anytime algorithm*. It is the highest level of the arithmetical hierarchy which can be approximated using a regular Turing machine. If this criterion is not met, our model would be useless to guide practical research.

For MDPs, planning is already P-complete for finite and infinite horizons [PT87]. In POMDPs, planning is undecidable [MHC99, MHC03]. The existence of a policy whose expected value exceeds a given threshold is PSPACE-complete [MGLA00], even for purely epistemic POMDPs in which actions do not change the hidden state [SLR07]. In this paper we derive hardness results for planning in general semicomputable environments; this environment class is even more general than POMDPs. We show that finding an optimal policy is Π^0_2-hard and finding an ε-optimal policy is undecidable.

Moreover, we show that by default, AIXI is not limit computable. The reason is twofold: First, when picking the next action, two or more actions might have the same value (expected future rewards). The choice between them is easy, but determining whether such a tie exists is difficult. Second, in case of an infinite horizon (using discounting), the iterative definition of the value function [Hut05, Def. 5.30] conditions on surviving forever. The first problem

Model	γ	Optimal	ε-Optimal
Iterative AINU	DC	Δ^0_4, Σ^0_3-hard	Δ^0_3, Π^0_2-hard
	LT	Δ^0_3, Π^0_2-hard	Δ^0_2, Σ^0_1-hard
Iterative AIXI	DC	Δ^0_4, Π^0_2-hard	Δ^0_3, Π^0_2-hard
	LT	Δ^0_3, Σ^0_1-hard	Δ^0_2, Σ^0_1-hard
Iterative AIMU	DC	Δ^0_2	Δ^0_1
	LT	Δ^0_2	Δ^0_1
Recursive AINU	DC	Δ^0_3, Π^0_2-hard	Δ^0_2, Σ^0_1-hard
	LT	Δ^0_3, Π^0_2-hard	Δ^0_2, Σ^0_1-hard
Recursive AIXI	DC	Δ^0_3, Σ^0_1-hard	Δ^0_2, Σ^0_1-hard
	LT	Δ^0_3, Σ^0_1-hard	Δ^0_2, Σ^0_1-hard
Recursive AIMU	DC	Δ^0_2	Δ^0_1
	LT	Δ^0_2	Δ^0_1

Table 1: Computability results for different agent models derived in Section 3. DC means general discounting, a lower semicomputable discount function γ; LT means finite lifetime, undiscounted rewards up to a fixed lifetime m. Hardness results for AIXI are with respect to a specific universal Turing machine; hardness results for AINU are with respect to a specific environment $\nu \in \mathcal{M}$.

can be circumvented by settling for an ε-optimal agent. We show that the second problem can be solved by using the recursive instead of the iterative definition of the value function. With this we get a limit-computable agent with infinite horizon. Table 1 and Table 3 summarize our computability results.

2 PRELIMINARIES

2.1 THE ARITHMETICAL HIERARCHY

A set $A \subseteq \mathbb{N}$ is Σ^0_n iff there is a computable relation S such that

$$k \in A \iff \exists k_1 \forall k_2 \ldots Q_n k_n \; S(k, k_1, \ldots, k_n) \qquad (1)$$

where $Q_n = \forall$ if n is even, $Q_n = \exists$ if n is odd [Nie09, Def. 1.4.10]. A set $A \subseteq \mathbb{N}$ is Π^0_n iff its complement $\mathbb{N} \setminus A$ is Σ^0_n. We call the formula on the right hand side of (1) a Σ^0_n*-formula*, its negation is called Π^0_n*-formula*. It can be shown that we can add any bounded quantifiers and duplicate quantifiers of the same type without changing the classification of A. The set A is Δ^0_n iff A is Σ^0_n and A is Π^0_n. We get that Σ^0_1 as the class of recursively enumerable sets, Π^0_1 as the class of co-recursively enumerable sets and Δ^0_1 as the class of recursive sets.

We say the set $A \subseteq \mathbb{N}$ is Σ^0_n*-hard (*Π^0_n*-hard,* Δ^0_n*-hard)* iff for any set $B \in \Sigma^0_n$ ($B \in \Pi^0_n$, $B \in \Delta^0_n$), B is many-one reducible to A, i.e., there is a computable function f such that $k \in B \leftrightarrow f(k) \in A$ [Nie09, Def. 1.2.1]. We get $\Sigma^0_n \subset \Delta^0_{n+1} \subset \Sigma^0_{n+1} \subset \ldots$ and $\Pi^0_n \subset \Delta^0_{n+1} \subset \Pi^0_{n+1} \subset \ldots$. This hierarchy of subsets of natural numbers is known as the *arithmetical hierarchy*.

By Post's Theorem [Nie09, Thm. 1.4.13], a set is Σ^0_n if and only if it is recursively enumerable on an oracle machine with an oracle for a Σ^0_{n-1}-complete set.

2.2 STRINGS

Let $\mathcal{X}$ be some finite set called *alphabet*. The set $\mathcal{X}^* := \bigcup_{n=0}^{\infty} \mathcal{X}^n$ is the set of all finite strings over the alphabet $\mathcal{X}$, the set $\mathcal{X}^\infty$ is the set of all infinite strings over the alphabet $\mathcal{X}$, and the set $\mathcal{X}^\sharp := \mathcal{X}^* \cup \mathcal{X}^\infty$ is their union. The empty string is denoted by ϵ, not to be confused with the small positive real number ε. Given a string $x \in \mathcal{X}^*$, we denote its length by $|x|$. For a (finite or infinite) string x of length $\geq k$, we denote with $x_{1:k}$ the first k characters of x, and with $x_{<k}$ the first $k-1$ characters of x. The notation $x_{1:\infty}$ stresses that x is an infinite string. We write $x \sqsubseteq y$ iff x is a prefix of y, i.e., $x = y_{1:|x|}$.

2.3 COMPUTABILITY OF REAL-VALUED FUNCTIONS

We fix some encoding of rational numbers into binary strings and an encoding of binary strings into natural numbers. From now on, this encoding will be done implicitly wherever necessary.

Definition 1 (Σ^0_n-, Π^0_n-, Δ^0_n-computable)**.** A function $f : \mathcal{X}^* \to \mathbb{R}$ is called Σ^0_n*-computable (*Π^0_n*-computable,* Δ^0_n*-computable)* iff the set $\{(x, q) \in \mathcal{X}^* \times \mathbb{Q} \mid f(x) > q\}$ is Σ^0_n (Π^0_n, Δ^0_n).

A Δ^0_1-computable function is called *computable*, a Σ^0_1-computable function is called *lower semicomputable*, and a Π^0_1-computable function is called *upper semicomputable*. A Δ^0_2-computable function f is called *limit computable*, because there is a computable function ϕ such that

$$\lim_{k \to \infty} \phi(x, k) = f(x).$$

The program ϕ that limit computes f can be thought of as an *anytime algorithm* for f: we can stop ϕ at any time k and get a preliminary answer. If the program ϕ ran long enough (which we do not know), this preliminary answer will be close to the correct one.

Limit-computable sets are the highest level in the arithmetical hierarchy that can be approached by a regular Turing machine. Above limit-computable sets we necessarily need some form of halting oracle. See Table 2 for the definition of lower/upper semicomputable and limit-computable functions in terms of the arithmetical hierarchy.

Lemma 2 (Computability of Arithmetical Operations)**.** *Let* $n > 0$ *and let* $f, g : \mathcal{X}^* \to \mathbb{R}$ *be two* Δ^0_n*-computable functions. Then*

	$f_>$	$f_<$
f is computable	Δ^0_1	Δ^0_1
f is lower semicomputable	Σ^0_1	Π^0_1
f is upper semicomputable	Π^0_1	Σ^0_1
f is limit computable	Δ^0_2	Δ^0_2
f is Δ^0_n-computable	Δ^0_n	Δ^0_n
f is Σ^0_n-computable	Σ^0_n	Π^0_n
f is Π^0_n-computable	Π^0_n	Σ^0_n

Table 2: Connection between the computability of real-valued functions and the arithmetical hierarchy. We use the shorthand $f_> := \{(x,q) \mid f(x) > q\}$ and $f_< := \{(x,q) \mid f(x) < q\}$.

(i) $\{(x,y) \mid f(x) > g(y)\}$ *is* Σ^0_n,

(ii) $\{(x,y) \mid f(x) \leq g(y)\}$ *is* Π^0_n,

(iii) $f+g$, $f-g$, *and* $f \cdot g$ *are* Δ^0_n*-computable, and*

(iv) f/g *is* Δ^0_n*-computable if* $g(x) \neq 0$ *for all* x.

2.4 ALGORITHMIC INFORMATION THEORY

A *semimeasure* over the alphabet $\mathcal{X}$ is a function $\nu : \mathcal{X}^* \to [0,1]$ such that (i) $\nu(\epsilon) \leq 1$, and (ii) $\nu(x) \geq \sum_{a \in \mathcal{X}} \nu(xa)$ for all $x \in \mathcal{X}^*$. A semimeasure is called (probability) *measure* iff for all x equalities hold in (i) and (ii). *Solomonoff's prior* M [Sol64] assigns to a string x the probability that the reference universal monotone Turing machine U [LV08, Ch. 4.5.2] computes a string starting with x when fed with uniformly random bits as input. The *measure mixture* $\overline{M}$ [Gá83, p. 74] removes the contribution of programs that do not compute infinite strings; it is a measure except for a constant factor. Formally,

$$M(x) := \sum_{p:\, x \sqsubseteq U(p)} 2^{-|p|}, \quad \overline{M}(x) := \lim_{n\to\infty} \sum_{y \in \mathcal{X}^n} M(xy)$$

Equivalently, the Solomonoff prior M can be defined as a mixture over all lower semicomputable semimeasures [WSH11]. The function M is a lower semicomputable semimeasure, but not computable and not a measure [LV08, Lem. 4.5.3]. A semimeasure ν can be turned into a measure ν_{norm} using *Solomonoff normalization*: $\nu_{\text{norm}}(\epsilon) := 1$ and for all $x \in \mathcal{X}^*$ and $a \in \mathcal{X}$,

$$\nu_{\text{norm}}(xa) := \nu_{\text{norm}}(x) \frac{\nu(xa)}{\sum_{b \in \mathcal{X}} \nu(xb)}. \quad (2)$$

2.5 GENERAL REINFORCEMENT LEARNING

In general reinforcement learning the agent interacts with an environment in cycles: at time step t the agent chooses an *action* $a_t \in \mathcal{A}$ and receives a *percept* $e_t = (o_t, r_t) \in \mathcal{E}$ consisting of an *observation* $o_t \in \mathcal{O}$ and a real-valued *reward* $r_t \in \mathbb{R}$; the cycle then repeats for $t+1$. A *history* is an element of $(\mathcal{A} \times \mathcal{E})^*$. We use $æ \in \mathcal{A} \times \mathcal{E}$ to denote one interaction cycle, and $æ_{1:t}$ to denote a history of length t. The goal in reinforcement learning is to maximize total discounted rewards. A *policy* is a function $\pi : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$ mapping each history to the action taken after seeing this history.

The environment can be stochastic, but is assumed to be semicomputable. In accordance with the AIXI literature [Hut05], we model environments as lower semicomputable *chronological conditional semimeasures* (LSCCCSs). A *conditional semimeasure* ν takes a sequence of actions $a_{1:t}$ as input and returns a semimeasure $\nu(\,\cdot \parallel a_{1:t})$ over $\mathcal{E}^\sharp$. A conditional semimeasure ν is *chronological* iff percepts at time t do not depend on future actions, i.e., $\nu(e_{1:t} \parallel a_{1:k}) = \nu(e_{1:t} \parallel a_{1:t})$ for all $k > t$. Despite their name, conditional semimeasures do *not* specify conditional probabilities; the environment ν is *not* a joint probability distribution on actions and percepts. Here we only care about the computability of the environment ν; for our purposes, chronological conditional semimeasures behave just like semimeasures.

2.6 THE UNIVERSAL AGENT AIXI

Our environment class $\mathcal{M}$ is the class of all LSCCCSs. Typically, Bayesian agents such as AIXI only function well if the true environment is in their hypothesis class. Since the hypothesis class $\mathcal{M}$ is extremely large, the assumption that it contains the true environment is rather weak. We fix the *universal prior* $(w_\nu)_{\nu \in \mathcal{M}}$ with $w_\nu > 0$ for all $\nu \in \mathcal{M}$ and $\sum_{\nu \in \mathcal{M}} w_\nu \leq 1$, given by the reference machine U. The universal prior w gives rise to the *universal mixture* ξ, which is a convex combination of all LSCCCSs $\mathcal{M}$:

$$\xi(e_{<t} \parallel a_{<t}) := \sum_{\nu \in \mathcal{M}} w_\nu \nu(e_{<t} \parallel a_{<t})$$

It is analogous to the Solomonoff prior M but defined for reactive environments. Like M, the universal mixture ξ is lower semicomputable [Hut05, Sec. 5.10].

We fix a *discount function* $\gamma : \mathbb{N} \to \mathbb{R}$ with $\gamma_t := \gamma(t) \geq 0$ and $\sum_{t=1}^\infty \gamma_t < \infty$ and make the following assumptions.

Assumption 3. *(a) The discount function* γ *is lower semicomputable.*

(b) Rewards are bounded between 0 *and* 1.

(c) The set of actions $\mathcal{A}$ *and the set of percepts* $\mathcal{E}$ *are both finite.*

Assumption 3 (b) could be relaxed to bounded rewards because we can rescale rewards $r \mapsto cr + d$ for any $c, d \in \mathbb{R}$ without changing optimal policies if the environment ν is a measure. However, for our value-related results, we require that rewards are nonnegative.

We define the *discount normalization factor* $\Gamma_t := \sum_{i=t}^{\infty} \gamma_i$. There is no requirement that $\Gamma_t > 0$. In fact, we use γ for both, AIXI with discounted infinite horizon ($\Gamma_t > 0$ for all t), and AIXI with finite lifetime m. In the latter case we set

$$\gamma_{\mathrm{LT}m}(t) := \begin{cases} 1 & \text{if } t \leq m \\ 0 & \text{if } t > m. \end{cases}$$

If we knew the true environment $\nu \in \mathcal{M}$, we would choose the ν-optimal agent known as AINU that maximizes ν-expected value (if ν is a measure). Since we do not know the true environment, we use the universal mixture ξ over all environments in $\mathcal{M}$ instead. This yields the Bayesian agent AIXI: it weighs every environment $\nu \in \mathcal{M}$ according to its prior probability w_ν.

Definition 4 (Iterative Value Function [Hut05, Def. 5.30])**.** The *value* of a policy π in an environment ν given history $æ_{<t}$ is

$$V_\nu^\pi(æ_{<t}) := \frac{1}{\Gamma_t} \lim_{m\to\infty} \sum_{e_{t:m}} R(e_{t:m}) \nu(e_{1:m} \mid e_{<t} \parallel a_{1:m})$$

if $\Gamma_t > 0$ and $V_\nu^\pi(æ_{<t}) := 0$ if $\Gamma_t = 0$ where $a_i := \pi(e_{<i})$ for all $i \geq t$ and $R(e_{t:m}) := \sum_{k=t}^{m} \gamma_k r_k$. The *optimal value* is defined as $V_\nu^*(h) := \sup_\pi V_\nu^\pi(h)$.

Let $æ_{<t} \in (\mathcal{A} \times \mathcal{E})^*$ be some history. We extend the value functions V_ν^π to include initial interactions (in reinforcement learning literature on MDPs these are called Q-values), $V_\nu^\pi(æ_{<t} a_t) := V_\nu^{\pi'}(æ_{<t})$ where π' is the policy π except that it takes action a_t next, i.e., $\pi'(æ_{<t}) := a_t$ and $\pi'(h) := \pi(h)$ for all $h \neq æ_{<t}$. We define $V_\nu^*(æ_{<t} a_t) := \sup_\pi V_\nu^\pi(æ_{<t} a_t)$ analogously.

Definition 5 (Optimal Policy [Hut05, Def. 5.19 & 5.30])**.** A policy π is *optimal in environment ν (ν-optimal)* iff for all histories the policy π attains the optimal value: $V_\nu^\pi(h) = V_\nu^*(h)$ for all $h \in (\mathcal{A} \times \mathcal{E})^*$.

Since the discount function is summable, rewards are bounded (Assumption 3b), and actions and percepts spaces are both finite (Assumption 3c), an optimal policy exists for every environment $\nu \in \mathcal{M}$ [LH14, Thm. 10]. For a fixed environment ν, an explicit expression for the optimal value function is

$$V_\nu^*(æ_{<t}) = \frac{1}{\Gamma_t} \lim_{m\to\infty} \underset{æ_{t:m}}{\max\!\!\!\!\!\!\sum} R(e_{t:m}) \nu(e_{1:m} \mid e_{<t} \parallel a_{1:m}), \tag{3}$$

where $\max\!\!\!\!\!\!\sum$ denotes the expectimax operator:

$$\underset{æ_{t:m}}{\max\!\!\!\!\!\!\sum} := \max_{a_t \in \mathcal{A}} \sum_{e_t \in \mathcal{E}} \ldots \max_{a_m \in \mathcal{A}} \sum_{e_m \in \mathcal{E}}$$

For an environment $\nu \in \mathcal{M}$ (an LSCCCS), AINU is defined as a ν-optimal policy $\pi_\nu^* = \arg\max_\pi V_\nu^\pi(\epsilon)$. To stress that the environment is given by a measure $\mu \in \mathcal{M}$ (as opposed to a semimeasure), we use AIMU. AIXI is defined as a ξ-optimal policy π_ξ^* for the universal mixture ξ [Hut05, Ch. 5]. Since $\xi \in \mathcal{M}$ and every measure $\mu \in \mathcal{M}$ is also a semimeasure, both AIMU and AIXI are a special case of AINU. However, AIXI is not a special case of AIMU since the mixture ξ is not a measure.

	Plain	Conditional
M	$\Sigma_1^0 \setminus \Delta_1^0$	$\Delta_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$
M_{norm}	$\Delta_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	$\Delta_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$
$\overline{M}$	$\Pi_2^0 \setminus \Delta_2^0$	$\Delta_3^0 \setminus (\Sigma_2^0 \cup \Pi_2^0)$
$\overline{M}_{\mathrm{norm}}$	$\Delta_3^0 \setminus (\Sigma_2^0 \cup \Pi_2^0)$	$\Delta_3^0 \setminus (\Sigma_2^0 \cup \Pi_2^0)$

Table 3: The complexity of the set $\{(x, q) \in \mathcal{X}^* \times \mathbb{Q} \mid f(x) > q\}$ where $f \in \{M, M_{\mathrm{norm}}, \overline{M}, \overline{M}_{\mathrm{norm}}\}$ is one of the various versions of Solomonoff's prior. Lower bounds on the complexity of $\overline{M}$ and $\overline{M}_{\mathrm{norm}}$ hold only for specific universal Turing machines.

Because there can be more than one optimal policy, the definitions of AINU, AIMU and AIXI are not unique. More specifically, a ν-optimal policy maps a history h to

$$\pi_\nu^*(h) :\in \arg\max_{a \in \mathcal{A}} V_\nu^*(ha). \tag{4}$$

If there are multiple actions $\alpha, \beta \in \mathcal{A}$ that attain the optimal value, $V_\nu^*(h\alpha) = V_\nu^*(h\beta)$, we say there is an *argmax tie*. Which action we settle on in case of a tie (how we break the tie) is irrelevant and can be arbitrary.

3 THE COMPLEXITY OF AINU, AIMU, AND AIXI

3.1 THE COMPLEXITY OF SOLOMONOFF INDUCTION

AIXI uses an analogue to Solomonoff's prior on all possible environments $\mathcal{M}$. Therefore we first state computability results for *Solomonoff's prior* M and the *measure mixture* $\overline{M}$ in Table 3 [LH15b]. Notably, M is lower semicomputable and its conditional is limit computable. However, neither the measure mixture $\overline{M}$ nor any of its variants are limit computable.

3.2 UPPER BOUNDS

In this section, we derive upper bounds on the computability of AINU, AIMU, and AIXI. Except for Corollary 13, all results in this section apply generally to any LSCCCS $\nu \in \mathcal{M}$, hence they apply to AIXI even though they are stated for AINU.

For a fixed lifetime m, only the first m interactions matter. There is a finite number of policies that are different for the first m interactions, and the optimal policy π^*_ξ can be encoded in a finite number of bits and is thus computable. To make a meaningful statement about the computability of AINU_{LT}, we have to consider it as the function that takes the lifetime m and outputs a policy π^*_ξ that is optimal in the environment ξ using the discount function $\gamma_{\text{LT}m}$. In contrast, for infinite lifetime discounting we just consider the function $\pi^*_\xi : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$.

In order to position AINU in the arithmetical hierarchy, we need to identify these functions with sets of natural numbers. In both cases, finite and infinite lifetime, we represent these functions as relations over $\mathbb{N} \times (\mathcal{A} \times \mathcal{E})^* \times \mathcal{A}$ and $(\mathcal{A} \times \mathcal{E})^* \times \mathcal{A}$ respectively. These relations are easily identified with sets of natural numbers by encoding the tuple with their arguments into one natural number. From now on this translation of policies (and m) into sets of natural numbers will be done implicitly wherever necessary.

Lemma 6 (Policies are in Δ^0_n)**.** *If a policy π is Σ^0_n or Π^0_n, then π is Δ^0_n.*

Proof. Let φ be a Σ^0_n-formula (Π^0_n-formula) defining π, i.e., $\varphi(h, a)$ holds iff $\pi(h) = a$. We define the formula φ',

$$\varphi'(h, a) := \bigwedge_{a' \in \mathcal{A} \setminus \{a\}} \neg\varphi(h, a').$$

The set of actions $\mathcal{A}$ is finite, hence φ' is a Π^0_n-formula (Σ^0_n-formula). Moreover, φ' is equivalent to φ. □

To compute the optimal policy, we need to compute the value function. The following lemma gives an upper bound on the computability of the value function for environments in $\mathcal{M}$.

Lemma 7 (Complexity of V^*_ν)**.** *For every LSCCCS $\nu \in \mathcal{M}$, the function V^*_ν is Π^0_2-computable. For $\gamma = \gamma_{\text{LT}m}$ the function V^*_ν is Δ^0_2-computable.*

Proof. Multiplying (3) with $\Gamma_t \nu(e_{<t} \parallel a_{<t})$ yields $V^*_\nu(æ_{<t}) > q$ if and only if

$$\lim_{m\to\infty} \max_{æ_{t:m}} \sum \nu(e_{1:m} \parallel a_{1:m}) R(e_{t:m}) > q\, \Gamma_t\, \nu(e_{<t} \parallel a_{<t}). \tag{5}$$

The inequality's right side is lower semicomputable, hence there is a computable function ψ such that $\psi(\ell) \nearrow q\, \Gamma_t\, \nu(e_{<t} \parallel a_{<t}) =: q'$ for $\ell \to \infty$. For a fixed m, the left side is also lower semicomputable, therefore there is a computable function ϕ such that $\phi(m, k) \nearrow \max \sum_{æ_{t:m}} \nu(e_{1:m} \parallel a_{1:m}) R(e_{t:m}) =: f(m)$ for $k \to \infty$. We already know that the limit of $f(m)$ for $m \to \infty$ exists (uniquely), hence we can write (5) as

$$\begin{aligned} & \lim_{m\to\infty} f(m) > q' \\ \iff & \forall m_0\, \exists m \geq m_0.\ f(m) > q' \\ \iff & \forall m_0\, \exists m \geq m_0\, \exists k.\ \phi(m, k) > q' \\ \iff & \forall \ell\, \forall m_0\, \exists m \geq m_0\, \exists k.\ \phi(m, k) > \psi(\ell), \end{aligned}$$

which is a Π^0_2-formula. In the finite lifetime case where m is fixed, the value function $V^*_\nu(æ_{<t})$ is Δ^0_2-computable by Lemma 2 (iv), since $V^*_\nu(æ_{<t}) = f(m)q/q'$. □

From the optimal value function V^*_ν we get the optimal policy π^*_ν according to (4). However, in cases where there is more than one optimal action, we have to break an argmax tie. This happens iff $V^*_\nu(h\alpha) = V^*_\nu(h\beta)$ for two potential actions $\alpha \neq \beta \in \mathcal{A}$. This equality test is more difficult than determining which is larger in cases where they are unequal. Thus we get the following upper bound.

Theorem 8 (Complexity of Optimal Policies)**.** *For any environment $\nu \in \mathcal{M}$, if V^*_ν is Δ^0_n-computable, then there is an optimal policy π^*_ν for the environment ν that is Δ^0_{n+1}.*

Proof. To break potential ties, we pick an (arbitrary) total order $\succ$ on $\mathcal{A}$ that specifies which actions should be preferred in case of a tie. We define

$$\begin{aligned} \pi_\nu(h) = a \ :\iff \quad & \bigwedge_{a' : a' \succ a} V^*_\nu(ha) > V^*_\nu(ha') \\ & \wedge \bigwedge_{a' : a \succ a'} V^*_\nu(ha) \geq V^*_\nu(ha'). \end{aligned} \tag{6}$$

Then π_ν is a ν-optimal policy according to (4). By assumption, V^*_ν is Δ^0_n-computable. By Lemma 2 (i) and (ii) $V^*_\nu(ha) > V^*_\nu(ha')$ is in Σ^0_n and $V^*_\nu(ha) \geq V^*_\nu(ha')$ is Π^0_n. Therefore the policy π_ν defined in (6) is a conjunction of a Σ^0_n-formula and a Π^0_n-formula and thus in Δ^0_{n+1}. □

Corollary 9 (Complexity of AINU)**.** *AINU_{LT} is Δ^0_3 and AINU_{DC} is Δ^0_4 for every environment $\nu \in \mathcal{M}$.*

Proof. From Lemma 7 and Theorem 8. □

Usually we do not mind taking slightly suboptimal actions. Therefore actually trying to determine if two actions have the exact same value seems like a waste of resources. In the following, we consider policies that attain a value that is always within some $\varepsilon > 0$ of the optimal value.

Definition 10 (ε-Optimal Policy)**.** A policy π is *ε-optimal in environment ν* iff $V^*_\nu(h) - V^\pi_\nu(h) < \varepsilon$ for all histories $h \in (\mathcal{A} \times \mathcal{E})^*$.

Theorem 11 (Complexity of ε-Optimal Policies)**.** *For any environment $\nu \in \mathcal{M}$, if V^*_ν is Δ^0_n-computable, then there is an ε-optimal policy π^ε_ν for the environment ν that is Δ^0_n.*

Proof. Let $\varepsilon > 0$ be given. Since the value function $V_\nu^*(h)$ is Δ_n^0-computable, the set $V_\varepsilon := \{(ha, q) \mid |q - V_\nu^*(ha)| < \varepsilon/2\}$ is in Δ_n^0 according to Definition 1. Hence we compute the values $V_\nu^*(ha')$ until we get within $\varepsilon/2$ for every $a' \in \mathcal{A}$ and then choose the action with the highest value so far. Formally, let $\succ$ be an arbitrary total order on $\mathcal{A}$ that specifies which actions should be preferred in case of a tie. Without loss of generality, we assume $\varepsilon = 1/k$, and define Q to be an $\varepsilon/2$-grid on $[0, 1]$, i.e., $Q := \{0, 1/2k, 2/2k, \ldots, 1\}$. We define

$$\begin{aligned} &\pi_\nu^\varepsilon(h) = a \;:\Longleftrightarrow \\ &\exists (q_{a'})_{a' \in \mathcal{A}} \in Q^{\mathcal{A}}. \bigwedge_{a' \in \mathcal{A}} (ha', q_{a'}) \in V_\varepsilon \\ &\qquad \wedge \bigwedge_{a' : a' \succ a} q_a > q_{a'} \wedge \bigwedge_{a' : a \succ a'} q_a \geq q_{a'} \\ &\qquad \wedge \text{the tuple } (q_{a'})_{a' \in \mathcal{A}} \text{ is minimal with} \\ &\qquad\quad \text{respect to the lex. ordering on } Q^{\mathcal{A}}. \end{aligned} \tag{7}$$

This makes the choice of a unique. Moreover, $Q^{\mathcal{A}}$ is finite since $\mathcal{A}$ is finite, and hence (7) is a Δ_n^0-formula. ☐

Corollary 12 (Complexity of ε-Optimal AINU)**.** *For any environment $\nu \in \mathcal{M}$, there is an ε-optimal policy for* AINU_{LT} *that is Δ_2^0 and there is an ε-optimal policy for* AINU_{DC} *that is Δ_3^0.*

Proof. From Lemma 7 and Theorem 11. ☐

If the environment $\nu \in \mathcal{M}$ is a measure, i.e., ν assigns zero probability to finite strings, then we get computable ε-optimal policies.

Corollary 13 (Complexity of AIMU)**.** *If the environment $\mu \in \mathcal{M}$ is a measure and the discount function γ is computable, then and* AIMU_{LT} *and* AIMU_{DC} *are limit computable (Δ_2^0), and ε-optimal* AIMU_{LT} *and* AIMU_{DC} *are computable (Δ_1^0).*

Proof. In the discounted case, we can truncate the limit $m \to \infty$ in (3) at the $\varepsilon/2$-effective horizon $m_{\text{eff}} := \min\{k \mid \Gamma_k/\Gamma_t < \varepsilon/2\}$, since everything after m_{eff} can contribute at most $\varepsilon/2$ to the value function. Any lower semicomputable measure is computable [LV08, Lem. 4.5.1]. Therefore V_μ^* as given in (3) is composed only of computable functions, hence it is computable according to Lemma 2. The claim now follows from Theorem 8 and Theorem 11. ☐

3.3 LOWER BOUNDS

We proceed to show that the bounds from the previous section are the best we can hope for. In environment classes where ties have to be broken, AIMU_{DC} has to solve Σ_3^0-hard problems (Theorem 15), and AIMU_{LT} has to solve Π_2^0-hard problems (Theorem 16). These lower bounds are stated for particular environments $\nu \in \mathcal{M}$.

We also construct universal mixtures that yield bounds on ε-optimal policies. In the finite lifetime case, there is an ε-optimal AIXI_{LT} that solves Σ_1^0-hard problems (Theorem 17), and for general discounting, there is an ε-optimal AIXI_{DC} that solves Π_2^0-hard problems (Theorem 18). For arbitrary universal mixtures, we prove the following weaker statement that only guarantees incomputability.

Theorem 14 (No AIXI is computable)**.** AIXI_{LT} *and* AIXI_{DC} *are not computable for any universal Turing machine U.*

This theorem follows from the incomputability of Solomonoff induction. Since AIXI uses an analogue of Solomonoff's prior for learning, it succeeds to predict the environment's behavior for its own policy [Hut05, Thm. 5.31]. If AIXI were computable, then there would be computable environments more powerful than AIXI: they can simulate AIXI and anticipate its prediction, which leads to a contradiction.

Proof. Assume there is a computable policy π_ξ^* that is optimal in ξ. We define a deterministic environment μ, the *adversarial environment* to π_ξ^*. The environment μ gives rewards 0 as long as the agent follows the policy π_ξ^*, and rewards 1 once the agent deviates. Formally, we ignore observations by setting $\mathcal{O} := \{0\}$, and define

$$\mu(r_{1:t} \parallel a_{1:t}) := \begin{cases} 1 & \text{if } \forall k \leq t.\, a_k = \pi_\xi^*((ar)_{<k}) \text{ and } r_k = 0 \\ 1 & \text{if } \forall k \leq t.\, r_k = \mathbb{1}_{k \geq i} \\ & \quad \text{where } i := \min\{j \mid a_j \neq \pi_\xi^*((ar)_{<j})\} \\ 0 & \text{otherwise.} \end{cases}$$

The environment μ is computable because the policy π_ξ^* was assumed to be computable. Suppose π_ξ^* acts in μ, then by [Hut05, Thm. 5.36], AIXI learns to predict perfectly *on policy*:

$$V_\xi^*(æ_{<t}) = V_\xi^{\pi_\xi^*}(æ_{<t}) \to V_\mu^{\pi_\xi^*}(æ_{<t}) = 0 \text{ as } t \to \infty,$$

since both π_ξ^* and μ are deterministic. Therefore we find a t large enough such that $V_\xi^*(æ_{<t}) < w_\mu$ (in the finite lifetime case we choose $m > t$) where $æ_{<t}$ is the interaction history of π_ξ^* in μ. A policy π with $\pi(æ_{<t}) \neq \pi_\xi^*(æ_{<t})$, gets a reward of 1 in environment μ for all time steps after t, hence $V_\mu^\pi(æ_{<t}) = 1$. With linearity of $V_\xi^\pi(æ_{<t})$ in ξ [Hut05, Thm. 5.31],

$$V_\xi^\pi(æ_{<t}) \geq w_\mu \tfrac{\mu(e_{1:t} \| a_{1:t})}{\xi(e_{1:t} \| a_{1:t})} V_\mu^\pi(æ_{<t}) \geq w_\mu,$$

since $\mu(e_{1:t} \parallel a_{1:t}) = 1$ (μ is deterministic), $V_\mu^\pi(æ_{<t}) = 1$, and $\xi(e_{1:t} \parallel a_{1:t}) \leq 1$. Now we get a contradiction:

$$w_\mu > V_\xi^*(æ_{<t}) = \max_{\pi'} V_\xi^{\pi'}(æ_{<t}) \geq V_\xi^\pi(æ_{<t}) \geq w_\mu \quad ☐$$

For the remainder of this section, we fix the action space to be $\mathcal{A} := \{\alpha, \beta\}$ with action α favored in ties. The percept space is fixed to a tuple of binary observations and rewards, $\mathcal{E} := \mathcal{O} \times \{0,1\}$ with $\mathcal{O} := \{0,1\}$.

Theorem 15 (AINU_{DC} is Σ^0_3-hard)**.** *If $\Gamma_t > 0$ for all t, there is an environment $\nu \in \mathcal{M}$ such that AINU_{DC} is Σ^0_3-hard.*

Proof. Let A be any Σ^0_3 set, then there is a computable relation S such that

$$n \in A \iff \exists i \, \forall t \, \exists k \; S(n,i,t,k). \tag{8}$$

We define a class of environments $\mathcal{M}' = \{\rho_0, \rho_1, \dots\} \subset \mathcal{M}$ where each environment ρ_i is defined by

$$\rho_i((or)_{1:t} \parallel a_{1:t}) := \begin{cases} 2^{-t}, & \text{if } o_{1:t} = 1^t \text{ and } \forall t' \leq t.\; r_{t'} = 0 \\ 2^{-n-1}, & \text{if } \exists n.\; 1^n 0 \sqsubseteq o_{1:t} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \alpha \\ & \quad \text{and } \forall t' \leq t.\; r_{t'} = 0 \\ 2^{-n-1}, & \text{if } \exists n.\; 1^n 0 \sqsubseteq o_{1:t} \sqsubseteq 1^n 0^\infty \text{ and } a_{n+2} = \beta \\ & \quad \text{and } \forall t' \leq t.\; r_{t'} = \mathbb{1}_{t' > n+1} \\ & \quad \text{and } \forall t' \leq t \, \exists k \; S(n,i,t',k) \\ 0, & \text{otherwise.} \end{cases}$$

Every ρ_i is a chronological conditional semimeasure by definition, so $\mathcal{M}' \subseteq \mathcal{M}$. Furthermore, every ρ_i is lower semicomputable since S is computable.

We define our environment ν as a mixture over $\mathcal{M}'$,

$$\nu := \sum_{i \in \mathbb{N}} 2^{-i-1} \rho_i;$$

the choice of the weights on the environments ρ_i is arbitrary but positive. Let π^*_ν be an optimal policy for the environment ν and recall that the action α is preferred in ties. We claim that for the ν-optimal policy π^*_ν,

$$n \in A \iff \pi^*_\nu(1^n 0) = \beta. \tag{9}$$

This enables us to decide whether $n \in A$ given the policy π^*_ν, hence proving (9) concludes this proof.

Let $n, i \in \mathbb{N}$ be given, and suppose we are in environment i and observe $1^n 0$. Taking action α next yields rewards 0 forever; taking action β next yields a reward of 1 for those time steps $t \geq n+2$ for which $\forall t' \leq t \, \exists k \; S(n,i,t',k)$, after that the semimeasure assigns probability 0 to all next observations. Therefore, if for some t_0 there is no k such that $S(n,i,t_0,k)$, then $\rho_i(e_{1:t_0} \parallel \dots \beta \dots) = 0$, and hence

$$V^*_{\rho_i}(1^n 0 \beta) = 0 = V^*_{\rho_i}(1^n 0 \alpha),$$

and otherwise ρ_i yields reward 1 for every time step after $n+1$, therefore

$$V^*_{\rho_i}(1^n 0 \beta) = \Gamma_{n+2} > 0 = V^*_{\rho_i}(1^n 0 \alpha)$$

(omitting the first $n+1$ actions and rewards in the argument of the value function). We can now show (9): By (8), $n \in A$ if and only if there is an i such that for all t there is a k such that $S(n,i,t,k)$, which happens if and only if there is an $i \in \mathbb{N}$ such that $V^*_{\rho_i}(1^n 0 \beta) > 0$, which is equivalent to $V^*_\nu(1^n 0 \beta) > 0$, which in turn is equivalent to $\pi^*_\mu(1^n 0) = \beta$ since $V^*_\nu(1^n 0 \alpha) = 0$ and action α is favored in ties. ☐

Theorem 16 (AINU_{LT} is Π^0_2-hard)**.** *There is an environment $\nu \in \mathcal{M}$ such that AINU_{LT} is Π^0_2-hard.*

The proof of Theorem 16 is analogous to the proof of Theorem 15. The notable difference is that we replace $\forall t' \leq t \; \exists k \; S(n,i,t',k)$ with $\exists k \; S(n,i,k)$. Moreover, we swap actions α and β: action α 'checks' the relation S and action β gives a sure reward of 1.

Theorem 17 (Some ε-optimal AIXI_{LT} are Σ^0_1-hard)**.** *There is a universal Turing machine U' and an $\varepsilon > 0$ such that any ε-optimal policy for AIXI_{LT} is Σ^0_1-hard.*

Proof. Let ξ denote the universal mixture derived from the reference universal monotone Turing machine U. Let A be a Σ^0_1-set and S computable relation such that $n+1 \in A$ iff $\exists k \; S(n,k)$. We define the environment

$$\nu((or)_{1:t} \parallel a_{1:t}) := \begin{cases} \xi((or)_{1:n} \parallel a_{1:n}), & \text{if } \exists n.\; o_{1:n} = 1^{n-1}0 \\ & \quad \text{and } a_n = \alpha \\ & \quad \text{and } \forall t' > n.\; e_{t'} = (0, \tfrac{1}{2}) \\ \xi((or)_{1:n} \parallel a_{1:n}), & \text{if } \exists n.\; o_{1:n} = 1^{n-1}0 \\ & \quad \text{and } a_n = \beta \\ & \quad \text{and } \forall t' > n.\; e_t = (0,1) \\ & \quad \text{and } \exists k \; S(n-1,k). \\ \xi((or)_{1:t} \parallel a_{1:t}), & \text{if } \nexists n.\; o_{1:n} = 1^{n-1}0 \\ 0, & \text{otherwise.} \end{cases}$$

The environment ν mimics the universal environment ξ until the observation history is $1^{n-1}0$. Taking the action α next gives rewards $1/2$ forever. Taking the action β next gives rewards 1 forever if $n \in A$, otherwise the environment ν ends at some future time step. Therefore we want to take action β if and only if $n \in A$. We have that ν is an LSCCCS since ξ is an LSCCCS and S is computable.

We define the universal lower semicomputable semimeasure $\xi' := \frac{1}{2}\nu + \frac{1}{8}\xi$. Choose $\varepsilon := 1/9$. Let $n \in A$ be given and define the lifetime $m := n+1$. Let $h \in (\mathcal{A} \times \mathcal{E})^n$ be any history with observations $o_{1:n} = 1^{n-1}0$. Since $\nu(1^{n-1}0 \mid a_{1:n}) = \xi(1^{n-1}0 \mid a_{1:n})$ by definition, the posterior weights of ν and ξ in ξ' are equal to the prior weights, analogously to [LH15a, Thm. 7]. In the following, we use the linearity of $V^{\pi^*_{\xi'}}_\rho$ in ρ [Hut05, Thm. 5.21], and the fact that values are bounded between 0 and 1. If there is a k

such that $S(n-1,k)$,

$$\begin{aligned}&V^*_{\xi'}(h\beta) - V^*_{\xi'}(h\alpha)\\&= \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\beta) - \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\alpha) + \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\beta) - \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\alpha)\\&\geq \tfrac{1}{2} - \tfrac{1}{4} + 0 - \tfrac{1}{8} = \tfrac{1}{8},\end{aligned}$$

and similarly if there is no k such that $S(n-1,k)$, then

$$\begin{aligned}&V^*_{\xi'}(h\alpha) - V^*_{\xi'}(h\beta)\\&= \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\alpha) - \tfrac{1}{2}V_\nu^{\pi^*_{\xi'}}(h\beta) + \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\alpha) - \tfrac{1}{8}V_\xi^{\pi^*_{\xi'}}(h\beta)\\&\geq \tfrac{1}{4} - 0 + 0 - \tfrac{1}{8} = \tfrac{1}{8}.\end{aligned}$$

In both cases $|V^*_{\xi'}(h\beta) - V^*_{\xi'}(h\alpha)| > 1/9$. Hence we pick $\varepsilon := 1/9$ and get for every ε-optimal policy $\pi^\varepsilon_{\xi'}$ that $\pi^\varepsilon_{\xi'}(h) = \beta$ if and only if $n \in A$. □

Theorem 18 (Some ε-optimal AIXI_{DC} are Π^0_2-hard)**.** *There is a universal Turing machine U' and an $\varepsilon > 0$ such that any ε-optimal policy for AIXI_{DC} is Π^0_2-hard.*

The proof of Theorem 18 is analogous to the proof of Theorem 17 except that we choose $\forall m' \leq m\, \exists k\; S(x,m,k)$ as a condition for reward 1 after playing action β.

4 ITERATIVE VS. RECURSIVE AINU

Generally, our environment $\nu \in \mathcal{M}$ is only a semimeasure and not a measure. I.e., there is a history $æ_{<t}a_t$ such that

$$1 > \sum_{e_t \in \mathcal{E}} \nu(e_t \mid e_{<t} \parallel a_{1:t}).$$

In such cases, with positive probability the environment ν does not produce a new percept e_t. If this occurs, we shall use the informal interpretation that the environment ν *ended*, but our formal argument does not rely on this interpretation.

The following proposition shows that for a semimeasure $\nu \in \mathcal{M}$ that is not a measure, the iterative definition of AINU does not maximize ν-expected rewards. Recall that γ_1 states the discount of the first reward. In the following, we assume without loss of generality that $\gamma_1 > 0$, i.e., we are not indifferent about the reward received in time step 1.

Proposition 19 (Iterative AINU is not a ν-Expected Rewards Maximizer)**.** *For any $\varepsilon > 0$ there is an environment $\nu \in \mathcal{M}$ that is not a measure and a policy π that receives a total of γ_1 rewards in ν, but AINU receives only $\varepsilon\gamma_1$ rewards in ν.*

Informally, the environment ν is defined as follows. In the first time step, the agent chooses between the two actions α and β. Taking action α gives a reward of 1, and subsequently the environment ends. Action β gives a reward of ε, but the environment continues forever. There are no other rewards in this environment. From the perspective of ν-expected reward maximization, it is better to take action α, however AINU takes action β.

Proof. Let $\varepsilon > 0$. We ignore observations and set $\mathcal{E} := \{0, \varepsilon, 1\}$, $\mathcal{A} := \{\alpha, \beta\}$. The environment ν is formally defined by

$$\nu(r_{1:t} \parallel a_{1:t}) := \begin{cases} 1 & \text{if } a_1 = \alpha \text{ and } r_1 = 1 \text{ and } t = 1 \\ 1 & \text{if } a_1 = \beta \text{ and } r_1 = \varepsilon \text{ and } r_k = 0 \; \forall 1 < k \leq t \\ 0 & \text{otherwise.} \end{cases}$$

Taking action α first, we have $\nu(r_{1:t} \parallel \alpha a_{2:t}) = 0$ for $t > 1$ (the environment ν ends in time step 2 given history α). Hence we use (3) to conclude

$$V^*_\nu(\alpha) = \frac{1}{\Gamma_t} \lim_{m \to \infty} \sum_{r_{1:m}} \nu(r_{1:m} \parallel \alpha a_{2:m}) \sum_{i=1}^m r_i = 0.$$

Taking action β first we get

$$V^*_\nu(\beta) = \frac{1}{\Gamma_t} \lim_{m \to \infty} \sum_{r_{1:m}} \nu(r_{1:m} \parallel \beta a_{2:m}) \sum_{i=1}^m r_i = \frac{\gamma_1}{\Gamma_1}\varepsilon.$$

Since $\gamma_1 > 0$ and $\varepsilon > 0$, we have $V^*_\nu(\beta) > V^*_\nu(\alpha)$, and thus AIMU will use a policy that plays action β first, receiving a total discounted reward of $\varepsilon\gamma_1$. In contrast, any policy π that takes action α first receives a larger total discounted reward of γ_1. □

Whether it is reasonable to assume that our environment has a nonzero probability of ending is a philosophical debate we do not want to engage in here. Instead, we have a different motivation to use the recursive value function: we get an improved computability result. Concretely, we show that for all environments $\nu \in \mathcal{M}$, there is a limit-computable ε-optimal policy maximizing ν-expected rewards using an infinite horizon. According to Theorem 18, this does not hold for all V^*_ν-maximizing agents AINU.

In order to maximize ν-expected rewards in case ν is not a measure, we need the recursive definition of the value function (analogously to [Hut05, Eq. 4.12]). To avoid confusion, we denote it W^π_ν:

$$W^\pi_\nu(æ_{<t}) = \frac{1}{\Gamma_t} \sum_{e_t} \big(\gamma_t r_t + \Gamma_{t+1} W^\pi_\nu(æ_{1:t})\big) \nu(e_t \mid e_{<t} \parallel a_{1:t})$$

where $a_t := \pi(æ_{<t})$. In the following we write it in non-recursive form.

Definition 20 (ν-Expected Value Function)**.** The *ν-expected value* of a policy π in an environment ν given history $æ_{<t}$ is

$$W^\pi_\nu(æ_{<t}) := \frac{1}{\Gamma_t} \sum_{m=t}^{\infty} \sum_{e_{t:m}} \gamma_m r_m \nu(e_{1:m} \mid e_{<t} \parallel a_{1:m})$$

if $\Gamma_t > 0$ and $W^\pi_\nu(æ_{<t}) := 0$ if $\Gamma_t = 0$ where $a_i := \pi(e_{<i})$ for all $i \geq t$. The *optimal ν-expected value* is defined as $W^*_\nu(h) := \sup_\pi W^\pi_\nu(h)$.

The difference between V^π_ν and W^π_ν is that for W^π_ν all obtained rewards matter, but for V^π_ν only the rewards in timelines that continue indefinitely. In this sense the value function V^π_ν conditions on surviving forever. If the environment μ is a measure, then the history is infinite with probability one, and so V^π_ν and W^π_ν coincide. Hence this distinction is not relevant for AIMU, only for AINU and AIXI.

So why use V^π_ν in the first place? Historically, this is how infinite-horizon AIXI has been defined [Hut05, Def. 5.30]. This definition is the natural adaptation of (optimal) minimax search in zero-sum games to the (optimal) expectimax algorithm for stochastic environments. It turns out to be problematic only because semimeasures have positive probability of ending prematurely.

Lemma 21 (Complexity of W^*_ν)**.** *For every LSCCCS $\nu \in \mathcal{M}$, and every lower semicomputable discount function γ, the function W^*_ν is Δ^0_2-computable.*

Proof. The proof is analogous to the proof of Lemma 7. We expand Definition 20 using the expectimax operator analogously to (3). This gives a quotient with numerator

$$\lim_{m\to\infty} \max_{æ_{t:m}} \sum_{i=t}^{m} \gamma_i r_i \nu(e_{1:i} \parallel a_{1:i}),$$

and denominator $\nu(e_{<t} \parallel a_{<t}) \cdot \Gamma_t$. In contrast to the iterative value function, the numerator is now nondecreasing in m because we assumed rewards to be nonnegative (Assumption 3b). Hence both numerator and denominator are lower semicomputable functions, so Lemma 2 (iv) implies that W^*_ν is Δ^0_2-computable. □

Now we can apply our results from Section 3.2 to show that using the recursive value function W^π_ν, we get a universal AI model with an infinite horizon whose ε-approximation is limit computable. Moreover, in contrast to iterative AINU, recursive AINU actually maximizes ν-expected rewards.

Corollary 22 (Complexity of Recursive AINU/AIXI)**.** *For any environment $\nu \in \mathcal{M}$, recursive* AINU *is Δ^0_3 and there is an ε-optimal recursive* AINU *that is Δ^0_2. In particular, for any universal Turing machine, recursive* AIXI *is Δ^0_3 and there is an ε-optimal recursive* AIXI *that is limit computable.*

Proof. From Theorem 8, Theorem 11, and Lemma 21. □

Analogously to Theorem 14, Theorem 16, and Theorem 17 we can show that recursive AIXI is not computable, recursive AINU is Π^0_2-hard, and for some universal Turing machines, ε-optimal recursive AIXI is Σ^0_1-hard.

5 DISCUSSION

We set out with the goal of finding a limit-computable perfect agent. Table 3 on page 4 summarizes our computability results regarding Solomonoff's prior M: conditional M and M_{norm} are limit computable, while $\overline{M}$ and $\overline{M}_{\text{norm}}$ are not. Table 1 on page 2 summarizes our computability results for AINU, AIXI, and AINU: iterative AINU with finite lifetime is Δ^0_3. Having an infinite horizon increases the level by one, while restricting to ε-optimal policies decreases the level by one. All versions of AINU are situated between Δ^0_2 and Δ^0_4 (Corollary 9 and Corollary 12). For environments that almost surely continue forever (semimeasure that are measures), AIMU is limit-computable and ε-optimal AIMU is computable. We proved that these computability bounds on iterative AINU are generally unimprovable (Theorem 15 and Theorem 16). Additionally, we proved weaker lower bounds for AIXI independent of the universal Turing machine (Theorem 14) and for ε-optimal AIXI for specific choices of the universal Turing machine (Theorem 17 and Theorem 18).

We considered ε-optimality in order to avoid having to break argmax ties. This ε does not have to be constant over time, instead we may let $\varepsilon \to 0$ as $t \to \infty$ at any computable rate. With this we retain the computability results of ε-optimal policies and get that the value of the $\varepsilon(t)$-optimal policy $\pi^{\varepsilon(t)}_\nu$ converges rapidly to the ν-optimal value: $V^*_\nu(æ_{<t}) - V^{\pi^{\varepsilon(t)}_\nu}_\nu(æ_{<t}) \to 0$ as $t \to \infty$. Therefore the limitation to ε-optimal policies is not very restrictive.

When the environment ν has nonzero probability of not producing a new percept, the iterative definition (Definition 4) of AINU fails to maximize ν-expected rewards (Proposition 19). We introduced a recursive definition of the value function for infinite horizons (Definition 20), which correctly returns ν-expected value. The difference between the iterative value function V and recursive value function W is readily exposed in the difference between M and $\overline{M}$. Just like V conditions on surviving forever, so does $\overline{M}$ eliminate the weight of programs that do not produce infinite strings. Both $\overline{M}$ and V are not limit computable for this reason.

Our main motivation for the introduction of the recursive value function W is the improvement of the computability of optimal policies. Recursive AINU is Δ^0_3 and admits a limit-computable ε-optimal policy (Corollary 22). In this sense our goal to find a limit-computable perfect agent has been accomplished.

REFERENCES

[BD62] David Blackwell and Lester Dubins. Merging of opinions with increasing information. *The Annals of Mathematical Statistics*, pages 882–886, 1962.

[Gá83] Péter Gács. On the relation between descriptional complexity and algorithmic probability. *Theoretical Computer Science*, 22(1–2):71 – 93, 1983.

[Hut00] Marcus Hutter. A theory of universal artificial intelligence based on algorithmic complexity. Technical Report cs.AI/0004001, 2000. `http://arxiv.org/abs/cs.AI/0004001`.

[Hut01] Marcus Hutter. New error bounds for Solomonoff prediction. *Journal of Computer and System Sciences*, 62(4):653–667, 2001.

[Hut05] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer, 2005.

[LH14] Tor Lattimore and Marcus Hutter. General time consistent discounting. *Theoretical Computer Science*, 519:140–154, 2014.

[LH15a] Jan Leike and Marcus Hutter. Bad universal priors and notions of optimality. In *Conference on Learning Theory*, 2015.

[LH15b] Jan Leike and Marcus Hutter. On the computability of Solomonoff induction and knowledge-seeking. 2015. Forthcoming.

[LV08] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Texts in Computer Science. Springer, 3rd edition, 2008.

[MGLA00] Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681–720, 2000.

[MHC99] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.

[MHC03] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1):5–34, 2003.

[Nie09] André Nies. *Computability and Randomness*. Oxford University Press, 2009.

[PT87] Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[RH11] Samuel Rathmanner and Marcus Hutter. A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136, 2011.

[SB98] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[SLR07] Régis Sabbadin, Jérôme Lang, and Nasolo Ravoanjanahry. Purely epistemic Markov decision processes. In *AAAI*, volume 22, pages 1057–1062, 2007.

[Sol64] Ray Solomonoff. A formal theory of inductive inference. Parts 1 and 2. *Information and Control*, 7(1):1–22 and 224–254, 1964.

[Sol78] Ray Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, 24(4):422–432, 1978.

[VNH+11] Joel Veness, Kee Siong Ng, Marcus Hutter, William Uther, and David Silver. A Monte-Carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40(1):95–142, 2011.

[WSH11] Ian Wood, Peter Sunehag, and Marcus Hutter. (Non-)equivalence of universal priors. In *Solomonoff 85th Memorial Conference*, pages 417–425. Springer, 2011.