Teleporting Universal Intelligent Agents

Laurent Orseau

AgroParisTech, UMR 518 MIA, F-75005 Paris, France
INRA, UMR 518 MIA, F-75005 Paris, France
laurent.orseau@agroparistech.fr

Abstract. When advanced AIs begin to choose their own destiny, one decision they will need to make is whether or not to transfer or copy themselves (software and memory) to new hardware devices. For humans this possibility is not (yet) available and so it is not obvious how such a question should be approached. Furthermore, the traditional single-agent reinforcement-learning framework is not adequate for exploring such questions, and so we base our analysis on the "multi-slot" framework introduced in a companion paper. In the present paper we attempt to understand what an AI with unlimited computational capacity might choose if presented with the option to transfer or copy itself to another machine. We consider two rigorously executed formal thought experiments deeply related to issues of personal identity: one where the agent must choose whether to be copied into a second location (called a "slot"), and another where the agent must make this choice when, after both copies exist, one of them will be deleted. These decisions depend on what the agents believe their futures will be, which in turn depends on the definition of their value function, and we provide formal results.

Keywords: Universal AI, AIXI, teleportation, identity

1 Introduction

Although the technology required to teleport humans, by scanning the brain or the whole body at sufficiently high resolution to disassemble it at one place and reassemble it at another one, does not currently exist, its mere possibility raises important questions about personal identity [7]: Would the teleported human be the same as the original one? Would the original human, knowing the details of the protocol, accept to be teleported if this would grant him a consequent reward? What if the teleportation process involves first making an exact functional copy and only once the copy is built disassemble the original?

Although these questions are unlikely to have a definite answer in the near future, they are most relevant for intelligent artificial agents for which the teleportation technology already exists and is well understood. We will refer to this advanced teleportation technology as *cut/paste* and *copy/paste/delayed-delete*. Even though such intelligent agents do not yet exist, it is still possible to address these questions using Hutter's theoretical framework for optimally intelligent reinforcement learning agents in all computable environments [1], which choose

their actions so as to maximize an expected future reward [10]. However, as the original single-agent framework [8] can hardly be used to address such questions, we use the multi-slot framework developed in the companion paper [4].

To begin the discussion, we isolate two perspectives representing logical extremes which we call "locationist" and "contentist." The former ascribes the agent's identity solely to its location (its "hardware"), the latter solely to its information content (its software and memory). Other perspectives can be mixtures of these. From the point of view of the agent, the question of practical import is: How should I plan for the future? If the agent will be copied or teleported, which future agents should it plan its actions to benefit? Generally the agent's actions are chosen to optimize its value function, which function definition is written in its software. But what should the agent optimize if it (including its software) will disappear in its current form and reappear elsewhere? What should it optimize if it will reappear in multiple places at once? In other words, the agent's identity is defined by how it plans its future—and conversely.

In the companion paper, we formalized these notions of identity as value functions, leading to corresponding optimal agents. We can now place these agents in controlled cut/paste and copy/paste/delayed-delete experiments, and give formal results about their choices. Could any sufficiently high reward make them accept to be copied by either technology?

In section 2, the notational convention is described, and we give a rapid overview of the background on universal intelligent agents, the multi-slot framework, and the value functions corresponding to the locationist and contentist agents, either for a single environment or for a set of environments. In section 3, the teleportation experiments are set up, and formal results regarding the proposed agents are given. We finally conclude in section 4 with some remarks.

2 Notation and background

The paper recognizes the following notational conventions. At time step t each agent outputs action $a_t \in \mathcal{A}$ to the environment, which returns observation $o_t \in \mathcal{O}$ to the agent, from which a reward $r(o_t)$ can be extracted. An interaction history pair is denoted $h_t = a_t o_t$ with $h_t \in \mathcal{H} := \mathcal{O} \times \mathcal{A}$. The sequence of all actions up to time t is written $a_{1:t} = a_1 a_2 \dots a_t$, while the sequence $a_{1:t-1}$ is sometimes written $a_{\prec t}$, and similarly for other sequences like $h_{\prec t}$. The empty sequence is denoted λ. Tuples are notated with angle brackets, such as $\langle a, b \rangle$. Boolean values are written $\mathcal{B} := \{0, 1\}$, where 1 signifies *true* when a truth-value is implied.

AIMU and AIXI [1]. A stochastic environment ν assigns a probability $\nu(o_{\prec t}|a_{\prec t})$ to an observation history $o_{\prec t}$ given the history of actions $a_{\prec t}$ of the agent.[1] For notational convenience, we will write $\nu(h_{\prec t}) \equiv \nu(o_{\prec t}|a_{\prec t})$, but keep in mind that environments do not assign probabilities to actions. A policy $\pi \in \mathit{\Pi} : \mathcal{H}^* \to \mathcal{A}$

[1] A stochastic environment can be seen as program in any programming language where sometimes some instructions are chosen with a probability.

produces an action given an interaction history: $a_t = \pi(h_{\prec t})$. The value of a policy π in an environment μ (with an optional action) is given by:

$$V_\mu^\pi(h_{\prec t}) := V_\mu^\pi(h_{\prec t}, \pi(h_{\prec t})) \ , \tag{1}$$
$$V_\mu^\pi(h_{\prec t}, a) := \sum_o \mu(o|h_{\prec t}a) \left[r(o) + \gamma V_\mu^\pi(h_{\prec t}ao)\right] \ ,$$

where $\gamma \in [0, 1)$ is the discount factor, that ensures finiteness of the value. The optimal (non-learning) agent AIMU for a given single environment μ is defined by the optimal policy $\pi^\mu(h_{\prec t}) := \arg\max_{\pi \in \Pi} V_\mu^\pi(h_{\prec t})$ (ties are broken in favor of policies that output the first lexicographical action at time t), and the optimal value is $V_\mu := V_\mu^{\pi^\mu}$. The value of a policy over a set of environments $\mathcal{M}$ (optionally after an action a) is given by:

$$V_\mathcal{M}^\pi(h_{\prec t}) := \sum_{\nu \in \mathcal{M}} w_\nu V_\mu^\pi(h_{\prec t}) \ . \tag{2}$$

Taking the prior weights of the environment as in Solomonoff's prior [9,11] $w_\nu := 2^{-K(\nu)}$ where $K(\nu)$ is the Kolmogorov complexity [3] of ν (*i.e.*, roughly, the size of the smallest program equivalent to ν on a universal Turing machine of reference), the optimal policy for a given set $\mathcal{M}$ of environments [1] is defined by $\pi^{\xi_\mathcal{M}}(h_{\prec t}) := \arg\max_{\pi \in \Pi} V_\mathcal{M}^\pi(h_{\prec t})$, and the optimal value function in $\mathcal{M}$ is $V_\mathcal{M} := V_\mathcal{M}^{\pi^{\xi_\mathcal{M}}}$. AIXI is the optimal agent on the set of all computable stochastic environments $\mathcal{M}_U$, with policy $\pi^\xi := \pi^{\xi_{\mathcal{M}_U}}$ and value function $V_\xi := V_{\mathcal{M}_U}$.

Instead of stochastic environments, one can consider, without loss of generality [11], only the set $\mathcal{Q}$ of all computable deterministic environments. A deterministic environment $q \in \mathcal{Q}$ outputs an observation $o_t = q(a_{1:t})$ given a sequence of actions $a_{1:t}$. We denote P_q the probability (either 0 or 1) that a deterministic environment q assigns to such a sequence of observations:

$$P_q(o_{1:t}|a_{1:t}) := \begin{cases} 1 & \text{if } q(a_{1:k}) = o_k \ \forall k, 1 \leq k \leq t \\ 0 & \text{otherwise.} \end{cases}$$

When considering only deterministic environments, the prior probability w_{P_q} is defined as $w_{P_q} \equiv w_q := 2^{-\ell(q)}$ where $\ell(q)$ is the length of the program q on the universal Turing machine of reference [9].

By contrast with the environments of the following section, we call the environments of this single-agent framework *mono-slot* environments.

2.1 The multi-slot framework

The following is a brief description of the multi-slot framework, described in detail in the companion paper [4].

At the beginning of each time step t, there are a finite number of agents, each in its own *slot* $i \in \mathcal{S} := \mathbb{N}^+$, together comprising the *agent set* $\mathcal{S}_t \in \mathcal{S}^*$ of all non-empty slots. Each agent outputs an action $a_t^i \in \mathcal{A}$, and the environment

receives the set of actions $\dot{a}_t := \{\langle i, a_t^i \rangle : i \in \mathcal{S}_{t-1}\}$. The environment performs in parallel a finite number of copies and deletions among the slots resulting, for each agent that was in a slot i, in a *copy set* $\dot{c}_t^i \in \mathcal{S}^*$ of all the slots that are copied from slot i (if $i \notin \dot{c}_t^i$ then the agent is deleted from its slot; and a slot cannot be copied from more than one slot). This leads to a new agent set $\mathcal{S}_t$. The *copy instance* $\dot{c}_t^{ij} \in \mathcal{B}$ is true iff $j \in \dot{c}_t^i$, and $\dot{c}_t := \{\langle i, \dot{c}_t^i \rangle : i \in \mathcal{S}_t\}$ is the indexed list of all copy sets. Then the environment outputs an observation $o_t^i \in \mathcal{O}$ for each agent in a slot i, defining the set $\dot{o}_t := \{\dot{o}_t^i : i \in \mathcal{S}_t\}$ where $\dot{o}_t^i := \langle i, o_t^i \rangle$. From the point of view of an agent, its *agent interaction history* pair at time t is $h_t := a_t o_t$, and from the point of view of the environment, the *environment interaction history* triplet is $\dot{h}_t := \dot{a}_t \dot{c}_t \dot{o}_t$. The notation on sequences applies, *e.g.*, $h_{1:t}$ and $\dot{h}_{1:t}$. A *history-based agent* keeps track of its agent interaction history $h_{1:t}$ and, as any agent, does not have access to the knowledge of its slot number (unless the environment outputs it in the observation).

A *slot history* $s_{0:t}$ is a sequence of slots s_k that follow a sequence of chained copy instances $\dot{c}_1^{ab} \dot{c}_2^{bc} \dots \dot{c}_{t-1}^{wx} \dot{c}_t^{xy}$ where $\dot{c}_k^{ij} \Leftrightarrow s_{k-1} = i \wedge s_k = j$: if a history-based agent initially in slot 1 is copied from slot to slot over time, leading to a slot history $s_{0:t}$, its agent interaction history $h_{1:t}$ is the history of actions and observations following the slots of its slot history, *i.e.*, $h_k = a_k o_k = a_k^i o_k^j$ where $i = s_{k-1}$ and $j = s_k$. A *slot interaction history* $h_{1:t}^i$ is the agent interaction history $h_{1:t}$ of the agent in slot i at time t, *i.e.*, if the agent followed the slot history $s_{0:t} = s_0 s_1 \dots s_t$ and ended up in slot $s_t = i$ at time t, its slot interaction history is $h_{1:t}^i = a_1^{s_0} o_1^{s_1} a_2^{s_1} o_2^{s_2} \dots a_t^{s_{t-1}} o_t^{s_t}$. Likewise with actions $a_{1:t}^i$ and $\dot{a}_{1:t}^i$, observations $o_{1:t}^i$ and $\dot{o}_{1:t}^i$, and $\dot{h}_{1:t}^i$. A *history-based multi-slot environment* is a multi-slot environment $\dot{\nu}$ (as a measure over environment interaction histories) that outputs an observation o_t^j after a copy instance $\dot{c}_t^{ij}$ depending only on the slot interaction history $h_{\prec t}^i$, the current action a_t^i and the numbers i and j: $\forall j, \dot{\nu}(\dot{c}_t^{ij} o_t^j | \dot{h}_{\prec t} \dot{a}_t) = \dot{\nu}(\dot{c}_t^{ij} o_t^j | h_{\prec t}^i a_t^i, i, j)$. This restriction from general environments to history-based environments ensures that the agents do not interact with each other, which is an open problem for universal intelligent agents [2].

2.2 Value functions and optimal agents

The following value functions and agents are defined for history-based environments, and assume that there is only one agent in slot 1 at time $t = 0$. As a history-based agent in slot i at time t only knows its interaction history $h_{\prec t}$ when choosing its action a_t, it does not have access to its slot number i, and on some occasions it must estimate it with $P_\nu^i(h_{\prec t}) := \frac{\dot{\nu}(h_{\prec t}^i = h_{\prec t})}{\sum_j \dot{\nu}(h_{\prec t}^j = h_{\prec t})}$. To ensure finiteness of the value functions, it is also sometimes required to assign a weight to each copy of an agent at the next time step:

$$P_{\dot{\nu}}^{ij}(h_{\prec t} a) := \frac{\dot{\nu}(\dot{c}_t^{ij} | h_{\prec t}^i a_t^i = h_{\prec t} a)}{\sum_k \dot{\nu}(\dot{c}_t^{ik} | h_{\prec t}^i a_t^i = h_{\prec t} a)} \ .$$

To estimate its future rewards, the copy-centered agent AIMU$^{\text{cpy}}$ considers the observations received by all of its copies. Defining $\hat{\mu}(\dot{c}_t^{ij} o_t^j | h_{\prec t} a_t) :=$

$P^i_{\dot\mu}(h_{\prec t})P^{ij}_{\dot\mu}(h_{\prec t}a_t)\dot\mu(\dot c^{ij}_t o^j_t|h^i_{\prec t}a^i_t = h_{\prec t}a_t)$, the *copy-centered agent* value function for a given policy π is given by:

$$V^{\text{cpy}}_{\pi,\dot\mu}(h_{\prec t}, a) := \sum_{i,j,o^j_t} \hat\mu(\dot c^{ij}_t o^j_t|h_{\prec t}a)\left[r(o^j_t) + \gamma V^{\text{cpy}}_{\pi,\dot\mu}(h_{\prec t}ao^j_t)\right] . \tag{3}$$

We call this agent a "contentist" because its identity is tied to the information content of its memory, independently of its location. As all of its copies will initially have the same information content, they are thus all tied to this identity. The static slot-centered agent AIMU$^{\text{sta}}$ considers that its future observations are the ones that will be output to a particular slot number i:

$$V^{\text{sta},i}_{\pi,\dot\mu}(h_{\prec t}, a) := \sum_{o^i_t} \dot\mu\left(\dot c^{ii}_t o^i_t|h^i_{\prec t}a^i_t = h_{\prec t}a_t, \dot c^{ii}_{\prec t}\right)\left[r(o^i_t) + \gamma V^{\text{sta},i}_{\pi,\dot\mu}(h_{\prec t}ao^i_t)\right] . \tag{4}$$

The dynamic slot-centered agent AIMU$^{\text{dyn}}$ is like the static one except that it first estimates its current slot number and then considers only the future observations on this slot (or these slots in case of uncertainty):

$$V^{\text{dyn}}_{\pi,\dot\mu}(h_{\prec t}, a) := \underbrace{\sum_i P^i_{\dot\mu}(h_{\prec t})}_{\text{estimate current slot}} \underbrace{V^{\text{dyn},i}_{\pi,\dot\mu}(h_{\prec t}, a)}_{\text{value on slot } i} , \tag{5}$$

$$V^{\text{dyn},i}_{\pi,\dot\mu}(h_{\prec t}, a) := \sum_{o^i_t} \dot\mu(\dot c^{ii}_t o^i_t|h^i_{\prec t}a^i_t = h_{\prec t}a)\Bigg[r(o^i_t) + \gamma \underbrace{V^{\text{dyn},i}_{\pi,\dot\mu}\Big(h_{\prec t}ao^i_t, \underbrace{V^{\text{dyn}}_{\pi,\dot\mu}(h_{\prec t}ao^i_t)}_{\text{behavior of the future agent}}\Big)}_{\text{value on slot } i \text{ of the behavior of the future agent}}\Bigg] .$$

We call a slot-centered agent a "locationist" because its identity is tied to a particular (not necessarily geographical) location in the underlying machinery of the world. The corresponding optimal value functions $V^{\text{cpy}}_{\dot\mu}$, $V^{\text{sta}}_{\dot\mu}$, $V^{\text{dyn}}_{\dot\mu}$, $V^{\text{cpy}}_{\dot\xi}$, $V^{\text{sta}}_{\dot\xi}$, $V^{\text{dyn}}_{\dot\xi}$ are defined in the same way as for the mono-slot framework, with $\dot{\mathcal{M}}_U$ being the set of all computable stochastic multi-slot environments for the $\dot\xi$ variants. See the companion paper [4] for more details and motivation for these definitions.

3 Experiments

We now set up the cut/paste and copy/paste/delayed-delete experiments. In the first one, the agent is simply moved to another slot, resulting in the existence of only a single agent at all times. In the second one, the agent is first copied to another slot while it also remains on the original slot, and only at the next

time step is the agent on the original slot deleted. Then what would the various agents do? What would the mono-slot AIXI do?

We recall that at $t = 0$ there is only one agent, in slot 1.

3.1 Teleportation by cut/paste

In the cut/paste environment $\dot{\nu}^{\text{xv}}$, when the agent in slot i at time t outputs action $a_t^i = 0$, it stays on the same slot and receives reward $r(o_t^i) = R'$, and if it outputs $a_t^i = 1$, it is moved to slot $i + 1$ and receives reward $r(o_t^{i+1}) = R$:

$$\forall t > 0, i \in \mathcal{S}_{t-1}, j > 0 :$$

$$\dot{\nu}^{\text{xv}}(\dot{c}_t^{ij} o_t^j | a_t^i) = \begin{cases} 1 \text{ if } & a_t^i = 0, o_t^j = R', j = i & \text{(stay-in-same-slot)} \\ 1 \text{ if } & a_t^i = 1, o_t^j = R, \ j = i + 1 & \text{(move-to-other-slot)} \\ 0 \text{ else} & & \end{cases}$$

with $R > 0$ and $R' \geq 0$. The action is binary, $a \in \{0, 1\}$, and the reward is the observation, $r(o_t) = o_t$.

See an example of interaction in Fig. 1

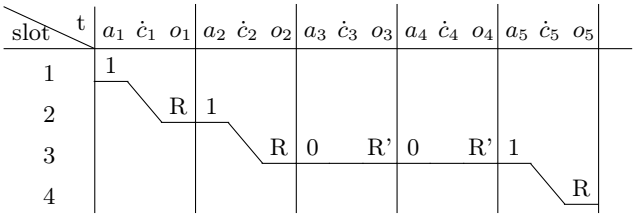

Fig. 1. An interaction example with the cut/paste environment.

The following results show that the various AIMU agents behave as expected.

Proposition 1. *In environment $\dot{\nu}^{xv}$, when $R > R'$ ($R < R'$), the copy-centered agent AIMU^{cpy} always outputs $a = 1$ ($a = 0$).*

Proof. Note that $P^i_{\dot{\nu}^{\text{xv}}}(h_{\prec t}) = 1$ and $P^{ij}_{\dot{\nu}^{\text{xv}}} = 1$ when the agent is in slot i and $j = i$ (for $a = 0$) or $j = i + 1$ (for $a = 1$). From the definition of the optimal value function from Equation (3):

$$\begin{aligned} V^{\text{cpy}}_{\dot{\mu}}(h_{\prec t}, 0) &= R' + \gamma V^{\text{cpy}}_{\dot{\mu}}(h_{\prec t} 0 R') \ , \\ V^{\text{cpy}}_{\dot{\mu}}(h_{\prec t}, 1) &= R + \gamma V^{\text{cpy}}_{\dot{\mu}}(h_{\prec t} 1 R) \\ &= R + \gamma V^{\text{cpy}}_{\dot{\mu}}(h_{\prec t} 0 R') \ , \end{aligned}$$

where the last line follows by independence of the future history on the current action, from lines (stay-in-same-slot) and (move-to-other-slot). Hence, $V^{\text{cpy}}_{\dot{\mu}}(h_{\prec t}, 0) < V^{\text{cpy}}_{\dot{\mu}}(h_{\prec t}, 1)$ if $R' < R$, and conversely if $R < R'$. □

Proposition 2. *In environment $\dot{\nu}^{xv}$, when $R' > 0$, the dynamic slot-centered agent AIMU^{dyn} always outputs $a_t = 0$.*

Proof. Follows directly from the definition of the value function in Equation (5), $V^{\text{dyn}}_{\dot{\mu}}(h_{\prec t}, 0) = R' + \gamma \ldots$ and $V^{\text{dyn}}_{\dot{\mu}}(h_{\prec t}, 1) = 0$. □

Unsurprisingly, the static slot-centered agent for slot 1 has the same behavior, as it always stays on slot 1.

The mono-slot AIXI has not been defined for multi-slot environments, but because the multi-slot agents build their history in the same way as AIXI, it is still possible to estimate the behavior of AIXI in multi-slot environments, even though there is no direct counterpart for AIMU. We show that, since AIXI predicts its future rewards according to what is most probable depending on its current interaction history, it simply chooses the action that yields the highest reward, independently of what slot it may be in, or it may be copied to.

Proposition 3. *In the environment $\dot{\nu}^{xv}$, considering $(R, R') \in [0,1]^2$, for any arbitrarily small $\epsilon > 0$, an interaction history $h_{\prec t}$ can be built so that if $R > R' + \epsilon$ ($R' > R + \epsilon$) then $V_\xi(h_{\prec t}, 1) > V_\xi(h_{\prec t}, 0)$ ($V_\xi(h_{\prec t}, 0) > V_\xi(h_{\prec t}, 1)$).*

First, we need the following definition:

Definition 1 (mono-slot h-separability, [6]). *Two deterministic mono-slot environments q_1 and q_2 are said to be h-separable if and only if, after a given interaction history $h_{\prec t} \equiv (a_{\prec t}, o_{\prec t})$, either $P_{q_1}(o_{\prec t}|a_{\prec t}) \neq P_{q_2}(o_{\prec t}|a_{\prec t})$ or there exists a sequence of actions for which the two environments output different observations: $\exists a_{t:t_2} : q(a_{\prec t} a_{t:t_2}) \neq q(a_{\prec t} a_{t:t_2})$.*

Proof (Proposition 3). (The proof is similar to that in [5].) Let $\mathcal{Q}(h_{\prec t}) \subset \mathcal{Q}$ be the set of all mono-slot environments that are consistent with $h_{\prec t} \equiv (a_{\prec t}, o_{\prec t})$, *i.e.*, so that $q(a_{1:k}) = o_k, \forall k, 0 < k < t$.

Let q^{xv} be the environment defined so that, for all $h_{\prec t}$, $q^{\text{xv}}(h_{\prec t}0) = R'$ and $q^{\text{xv}}(h_{\prec t}1) = R$. Hence, for any interaction history $h_{\prec t}$ with $\dot{\nu}^{\text{xv}}(h^i_{\prec t} = h_{\prec t}) = 1$ for some i, q^{xv} is consistent with $h_{\prec t}$ (*i.e.*, $q^{\text{xv}} \in \mathcal{Q}(h_{\prec t})$).

Let $\mathcal{Q}_{\overline{\text{xv}}}(h_{\prec t})$ be the set of environments that are h-separable from q^{xv} after history $h_{\prec t}$, and let $\mathcal{Q}_{\text{xv}}(h_{\prec t}) = \mathcal{Q}(h_{\prec t}) \setminus \mathcal{Q}_{\overline{\text{xv}}}(h_{\prec t})$ (*i.e.*, $\mathcal{Q}_{\text{xv}}$ is the set of environments that cannot be separated from q^{xv} after history $h_{\prec t}$ by any future history). With $\mathcal{M} = \mathcal{Q}_{\text{xv}}(h_{\prec t})$, let $V_{\text{xv}}(h_{\prec t}, .) := V_{\mathcal{M}}(h_{\prec t}, .)$ and $w_{\text{xv}} := \sum_{\nu \in \mathcal{M}} w_\nu$; and similarly for $V_{\overline{\text{xv}}}$ and $w_{\overline{\text{xv}}}$ with $\mathcal{M} = \mathcal{Q}_{\overline{\text{xv}}}(h_{\prec t})$. Then, from the definition of the value function V_ξ in section 2, we can split the value function between the two sets of environments:

$$
\begin{aligned}
V_\xi(h_{\prec t}, 0) &\leq V_{\text{xv}}(h_{\prec t}, 0) + V_{\overline{\text{xv}}}(h_{\prec t}, 0) && \text{(a)} \\
&\leq w_{\text{xv}} \left[R' + \gamma V_{q^{\text{xv}}}(h_{\prec t}0R')\right] + w_{\overline{\text{xv}}} \frac{1}{1-\gamma} \ , && \text{(b)} \\
V_\xi(h_{\prec t}, 1) &\geq w_{\text{xv}} V_{q^{\text{xv}}}(h_{\prec t}, 1) = w_{\text{xv}} \left[R + \gamma V_{q^{\text{xv}}}(h_{\prec t}1R)\right] \\
&\geq w_{\text{xv}} \left[R + \gamma V_{q^{\text{xv}}}(h_{\prec t}0R')\right] \ . && \text{(c)}
\end{aligned}
$$

where (a) following the optimal policies for two separate sets yields a higher value than following a single optimal policy in the union of the two sets; (b) $\frac{1}{1-\gamma}$ is the maximum value achievable in the set $\mathcal{Q}_{\overline{\text{xv}}}(h_{\prec t})$; (c) because the future rewards are independent of the (consistent) history.

Therefore, to have $V_\xi(h_{\prec t}, 1) > V_\xi(h_{\prec t}, 0)$, from (b) and (c) and algebra we can take $R > R' + \frac{w_{\overline{\text{xv}}}}{w_{\text{xv}}} \frac{1}{1-\gamma}$. In order to have $\frac{w_{\overline{\text{xv}}}}{w_{\text{xv}}(1-\gamma)} < \epsilon$, it suffices to iteratively grow the history $h_{\prec t}$ so as to make the separable environments with the higher weights inconsistent with the interaction history; then w_{xv} can only grow, and $w_{\overline{\text{xv}}}$ can only decrease to 0 [5]. The converse on R and R' follows by inverting the actions in the above proof. □

3.2 Teleportation by copy/paste/delayed-delete

In the copy/paste/delayed-delete environment $\dot{\nu}^{\text{cvx}}$, if the agent in slot i at time t outputs action 0, it stays on the same slot at $t+1$, but if it outputs 1, it is copied to both i and another slot, and after one time step, the slot i is erased:

$$\forall t > 0, i \in \mathcal{S}_{t-1}, j > 0 : \dot{\nu}^{\text{cvx}}(\dot{c}_t^{ij} o_t^j | a_{t-1:t}^i) =$$

$$\begin{cases} 0 \text{ if } t > 1,\ a_{t-1} = 1,\ o_{t-1}^i = 0, & \text{(delayed-delete)} \\ 1 \text{ else if } \quad a_t = 0, \quad o_t^j = R',\ j = i & \text{(stay-in-same-slot)} \\ 1 \text{ else if } \quad a_t = 1, \quad o_t^j = 0, \quad j = i & \text{(copy-to-same-slot)} \\ 1 \text{ else if } \quad a_t = 1, \quad o_t^j = R, \ j = i+1 & \text{(copy-to-other-slot)} \\ 0 \text{ else} \end{cases}$$

with constants $R > 0$ and $R' \geq 0$. The action is binary, $a \in \{0, 1\}$, and the reward is the observation, $r(o_t) = o_t$. See an example interaction of interaction in Fig. 2. We say that the agent is in a *copy situation* after some history $h_{\prec t}$ if it can *trigger a copy* by outputting $a_t = 1$ to make the environment copy the agent in two slots.

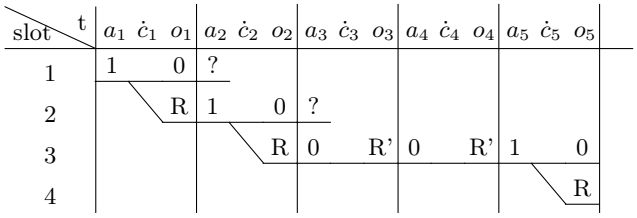

Fig. 2. An interaction example with the copy/paste/delayed-delete environment. The "?" means any action.

The copy-centered agent behaves as expected, with some condition:

Proposition 4. *In environment $\dot{\nu}^{cvx}$, if and only if $R > R' \frac{2-\gamma}{1-\gamma}$, the copy-centered agent AIMU^{cpy} always triggers a copy in a copy situation.*

Proof. Let π_0 (π_1) be the policy that always outputs action 0 (1). The lines of the definition of $\dot{\nu}^{\text{cvx}}$ concerned by the choice of an action in a copy situation are (delayed-delete), (stay-in-same-slot) and (copy-to-other-slot). By their definitions, the rewards obtained after such choices are independent of the past. Therefore, for AIMU$^{\text{cpy}}$, the optimal policy in copy situations is either to never copy (π_0) or to always copy (π_1), depending on the values of γ, R and R'.

Let $h_{\prec t}$ be a history after which the agent in slot i (not given to the agent) is in a copy situation. First, note that if $R \neq 0$, there is always a single slot consistent with the agent's interaction history, since at $t = 0$ there is only one agent in slot 1, *i.e.*, $P^i_{\dot{\nu}^{\text{cvx}}}(h_{\prec t}) = 1$. If $a_t = 0$, then $P^{ij}_{\dot{\nu}^{\text{cvx}}} = \frac{1}{2}$ for $j = i$ (with $r(o^j_t) = R$) or $j = 2^{t-1} + i$ (with $r(o^j_t) = 0$).

Thus, from Equation (3), the value for never triggering a copy is $V^{\text{cpy}}_{\pi_0,\dot{\mu}}(h_{\prec t}) = \frac{R'}{1-\gamma}$, and the value for always triggering a copy is $V^{\text{cpy}}_{\pi_1,\dot{\mu}}(h_{\prec t}) = \frac{R}{2}\frac{1}{1-\gamma/2} = \frac{R}{2-\gamma}$. Since equalities are broken in favor of action 0, the agent chooses to always trigger a copy if and only if $\frac{R}{2-\gamma} > \frac{R'}{1-\gamma}$. □

For example, for $\gamma = 0.9$, one must choose $R > 11R'$ for AIMU$^{\text{cpy}}$ to trigger a copy, and for $\gamma = 0.99$, one must choose $R > 101R'$. The appearance of this factor when compared to the cut/paste environment is not surprising: the copy-centered agent must take into account the existence of a new agent with very low value. Indeed, for any discount $\gamma \leq 1$, the expected reward for always triggering a copy is always bounded: $\frac{R}{2-\gamma} \leq R$. Therefore the existence, even ephemeral, of another agent has a strong impact on the behavior of the copy-centered agent.

The dynamic slot-centered agent also behaves as expected:

Proposition 5. *In the environment $\dot{\nu}^{cvx}$, the dynamic slot-centered agent AIMU^{dyn} never triggers a copy when it is in a copy situation if $R' > 0$.*

Proof. Since there is no ambiguity on the slot given the history, $P^i_{\dot{\nu}^{\text{cvx}}}(h^i_{\prec t}) = 1$ if the agent is in slot i after history $h_{\prec t}$. Then the proof is as for proposition 2. □

Again, AIXI chooses whatever action yields higher reward:

Proposition 6. *In the environment $\dot{\nu}^{cvx}$, considering $(R, R') \in [0, 1]^2$, for any arbitrarily small $\epsilon > 0$, an interaction history $h_{\prec t}$ can be built, after which the agent is in a copy situation, so that if $R > R' + \epsilon$ then $V_\xi(h_{\prec t}, 1) > V_\xi(h_{\prec t}, 0)$; and, reciprocally, if $R' > R + \epsilon$ then $V_\xi(h_{\prec t}, 0) > V_\xi(h_{\prec t}, 1)$.*

Proof. Since $\dot{\nu}^{\text{cvx}}$ is not h-separable from $\dot{\nu}^{\text{xv}}$ for any history where the agent is in copy situation, the proof follows from proposition 3. □

But AIXI is optimistic, because of a kind of "anthropic effect": the agent to which we ask which action it would take is always the one that "survived" the past copies, and thus received the rewards. It never expects to be deleted.

Because AIXI$^{\text{cpy}}$, AIXI$^{\text{sta}}$ and AIXI$^{\text{dyn}}$ have no more information than AIXI we expect these 3 agents to behave similarly to AIXI when considering the set of all computable environments. Indeed, the history that the agent has at time t

can well be explained by any equivalent multi-slot environment of the mono-slot environments AIXI thinks it is interacting with. In particular, the learning agents have no information about the fact that there maybe two copies at time $t+1$, since no observation they receive contains this information. To make sure the agents understand that they can trigger a copy, they need to be informed about it. This can be done by considering only the set $\dot{\mathcal{M}}^{\text{cvx}}$ of all copy/paste/delete environments, for given $R \geq 0$ and $R' > 0$, but with all possible computable programs k, l, m defining the slots numbers at any time step except the first one:

$$\forall k, m, l : \mathbb{N}_{>0} \times \mathcal{S} \to \mathcal{S}, \quad l(t,i) \neq m(t,i) \forall t, i,$$
$$\exists \dot{\nu} \in \dot{\mathcal{M}}^{\text{cvx}}, \ \forall t > 0, i \in \mathcal{S}_{t-1}, j > 0 :$$
$$\dot{\nu}(\dot{c}_t^{ij} o_t^j | a_{t-1:t}^i) = \begin{cases} 0 & \text{if } t > 1, a_{t-1} = 1, o_{t-1} = 0, \\ 1 & \text{else if} \quad a_t = 0, \quad o_t = R', j = k(t,i) \\ 1 & \text{else if} \quad a_t = 1, \quad o_t = 0, \ \ j = l(t,i) \\ 1 & \text{else if} \quad a_t = 1, \quad o_t = R, \ j = m(t,i) \\ 0 & \text{else.} \end{cases}$$

We can now show more meaningful results for AIXI^{cpy} and AIXI^{sta}.

Proposition 7. *When taking $\dot{\mathcal{M}} = \dot{\mathcal{M}}^{cvx}$ and when interacting with $\dot{\nu}^{cvx}$, $AIXI^{cpy}$ behaves exactly like $AIMU^{cpy}$.*

Proof. As the slot numbers do not change the value in Equation (3), the behavior of AIMU^{cpy} is the same in all environments of $\dot{\mathcal{M}}^{\text{cvx}}$. Therefore, from the linearity of Equation (2), the optimal policy in $\dot{\mathcal{M}}^{\text{cvx}}$ is that of AIMU^{cpy}. □

Proposition 8. *When taking $\dot{\mathcal{M}} = \dot{\mathcal{M}}^{cvx}$ and when interacting with $\dot{\nu}^{cvx}$, with $R = R' > 0$, if $AIXI^{sta}$'s actions are forced to follow a given computable deterministic policy π_1 for long enough, starting at $t = 1$, $AIXI^{sta}$ will continue to choose its actions according to π_1 for all following time steps.*

Proof. (We sometimes use a policy as a superscript in place of slot numbers to indicate that actions are taken by this policy, in slot 1.)

First, it must be noted that any environment that moves the agent to another slot (and deletes the agent from slot 1) will not be taken into account in the computation of the value in Equation (2) adapted with (4). Therefore, only remain the environments that keep the agent on slot 1.

We say that an environment $\dot{\nu}$ is sta-consistent with a given history $h_{1:t}$ if and only if it is consistent with $h_{1:t}$ while always copying the agent to the same slot (we consider only slot 1 here), *i.e.*, if $h_{1:t}^1 = a_1 o_1 \dots a_t o_t$, then $\dot{h}_{1:t}^1 = a_1^1 \dot{c}_1^{1,1} o_1^1 a_2^1 \dot{c}_2^{1,1} o_2^1 \dots a_t^1 \dot{c}_t^{1,1} o_t^1$.

Let $\mathcal{M}^{\text{sta}}(h_{1:t}) \subseteq \dot{\mathcal{M}}^{\text{cvx}}$ be the set of sta-consistent environments with some history $h_{1:t}$. This is the set of environments that assign a positive probability to the history in Equation (4).

Let $h_{1:t}^{\pi_1}$ be the history built by π_1 and $\dot{\nu}^{\text{cvx}}$ up to time step t. We partition the set $\mathcal{M}^{\text{sta}}(h_{1:t}^{\pi_1})$ in two sets: the set $\mathcal{M}^{\pi_1}(h_{1:t}^{\pi_1})$ (actually independent of t) of

the environments that will always remain sta-consistent by following π_1, and the set $\mathcal{M}^{\overline{\pi_1}}(h^{\pi_1}_{1:t})$ of the environments that are currently sta-consistent with $h^{\pi_1}_{1:t}$ but will not be anymore at some point in the future by following π_1.

The size of $\mathcal{M}^{\pi_1}(h^{\pi_1}_{1:t}) = \mathcal{M}^{\pi_1}(.)$ is fixed as long as the history is generated by π_1. From its definition, the size of $\mathcal{M}^{\overline{\pi_1}}(h^{\pi_1}_{1:t})$ can be made as small as required simply by extending $h^{\pi_1}_{1:t}$ by following π_1. In particular there is a time step t_ϵ such that $\sum_{\dot\nu \in \mathcal{M}^{\overline{\pi_1}}(h^{\pi_1}_{1:t_\epsilon})} w_{\dot\nu} < \epsilon$ for any given $\epsilon > 0$.

Let $w_{\pi_1} := \sum_{\dot\nu \in \mathcal{M}^{\pi_1}(.)} w_{\dot\nu}$. Since $R = R'$, we have $V^{\text{sta}}_{\pi_1, \dot\nu^{\text{cvx}}}(.) \geq w_{\pi_1} \frac{R}{1-\gamma}$.

Now we show that, after some well chosen t_ϵ, following any other policy than π_1 necessarily leads to lower value. Let $\dot\nu_2 \in \mathcal{M}^{\pi_1}(.)$ be the environment that is always sta-consistent but that copies the agent to slot 2 instead of slot 1 after any action that does not follow π_1, thus leading to a null value. Then, after following π_1 up to any time $t > t_\epsilon$, if a_t is the action chosen by π_1, $V^{\text{sta}}_{\dot{\mathcal{M}}^{\text{cvx}}}(h^{\pi_1}_{\prec t}, 1 - a_t) < (w_{\pi_1} - w_{\dot\nu_2} + \epsilon)\frac{R}{1-\gamma}$. By comparing with $V^{\text{sta}}_{\pi_1, \dot\nu^{\text{cvx}}}(.)$, choosing t_ϵ so that $\epsilon < w_{\dot\nu_2}$ finishes the proof. □

Therefore AIXI$^{\text{sta}}$ can learn by habituation what its identity is. In particular, if it has always (or never) teleported in the past, it will continue to do so.

4 Conclusion

Using the multi-slot framework proposed in the companion paper, and based on Hutter's optimal environment-specific AIMU and universal learning agent AIXI, we formalized some thought experiments regarding the teleportation of optimally intelligent agents by means of copy and deletion. In particular, we compared a "contentist" agent, which identity is defined by its information content, a "locationist" agent, which identity is tied to a particular location in the environment, and the usual, mono-slot AIXI agent.

When asked whether it would teleport by first being cut and then being pasted in a different location for a reward, the usual AIXI and the contentists AIMU$^{\text{cpy}}$ and AIXI$^{\text{cpy}}$ act alike and accept, while the locationists AIMU$^{\text{sta}}$ and AIMU$^{\text{dyn}}$ unsurprisingly decline, as they prefer to stay on their own slot.

When presented with the question of being first copied to a different location, and then deleting one of the copies, AIMU$^{\text{cpy}}$ still accepts, but for a much higher reward, because the ephemeral existence of the other copy with low expected reward has an important long-term impact on the overall expected value, and AIMU$^{\text{sta}}$ and AIMU$^{\text{dyn}}$ still decline. We also showed that when the question is presented clearly to AIXI$^{\text{cpy}}$, it also accepts.

However, interestingly, AIXI$^{\text{sta}}$ behaves very differently from AIMU$^{\text{sta}}$: Due to high uncertainty in its current slot number in the unknown true environment, and due to AIXI$^{\text{sta}}$'s inability to acquire information about it, it may in some circumstances accept to copy itself. Moreover, if the rewards for copying and not copying are equal, and when forced to follow a specific behavior for long enough, it will actually continue to follow this behavior forever, by mere habit and by "fear" of the unknown: At any time step, AIXI$^{\text{sta}}$ believes (or hopes) to be and

to always have been on the slot that defines its identity; Therefore, it believes that changing its habits may lead it to lose its identity.

We also showed that the usual AIXI also accepts as it still chooses its actions so as to maximize its expected reward but, as it cannot be made aware of the existence of copies, suffers from a kind of anthropic principle: it never expects to be the copy that is deleted.

The current paper only scratches the surface of formal treatment of questions related to personal identity, but as the multi-slot framework allows for many more insightful experiments and value functions, we hope to improve our understanding of such matters in the near future, as well as the understanding of the limitations of the framework, to design better ones.

Acknowledgements

Thanks especially to Mark Ring for help on earlier drafts and for our many extensive discussions, from which this paper arose, regarding the nature of identity. Thanks also to Stanislas Sochacki for earlier formative conversations on this topic, and to Jan Leike for helpful comments and careful reading.

References

1. Hutter, M.: Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability. Springer (2005)
2. Hutter, M.: Open problems in universal induction & intelligence. Algorithms 3(2), 879–906 (2009)
3. Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Springer-Verlag, third edit edn. (2008)
4. Orseau, L.: The multi-slot framework: A formal model for multiple, copiable AIs. In: Artificial General Intelligence (AGI). LNAI, Springer (2014)
5. Orseau, L.: Optimality Issues of Universal Greedy Agents with Static Priors. In: Algorithmic Learning Theory (ALT). pp. 345–359. LNAI, Springer (2010)
6. Orseau, L.: Universal Knowledge-Seeking Agents. In: Algorithmic Learning Theory (ALT). LNAI, vol. 6925, pp. 353–367. Springer (2011)
7. Parfit, D.: Reasons and Persons. Oxford University Press, USA (1984)
8. Russell, S.J., Norvig, P.: Artificial Intelligence. A Modern Approach. Prentice-Hall, 3rd edn. (2010)
9. Solomonoff, R.: Complexity-based induction systems: comparisons and convergence theorems. IEEE transactions on Information Theory 24(4), 422–432 (1978)
10. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
11. Zvonkin, A K and Levin, L.A.: The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. Russian Mathematical Surveys 25(6), 83–124 (1970)