Noname manuscript No.
(will be inserted by the editor)

Optimal Action Sequence Generation for Assistive Agents in Fixed Horizon Tasks

Kim Baraka · Francisco S. Melo · Marta Couto · Manuela Veloso*

Received: date / Accepted: date

Abstract Agents providing assistance to humans are faced with the challenge of automatically adjusting the level of assistance to ensure optimal performance. In this work, we argue that identifying the right level of assistance consists in balancing positive assistance outcomes and some (domain-dependent) measure of cost associated with assistive actions. Towards this goal, we contribute a general mathematical framework for structured tasks where an agent playing the role of a 'provider' — e.g., therapist, teacher — assists a human 'receiver' — e.g., patient, student. We specifically consider tasks where the provider agent needs to plan a sequence of actions over a fixed time horizon, where actions are organized along a hierarchy with increasing success probabilities, and some associated costs. The goal of the provider is to achieve a success with the lowest expected cost possible. We present OAssistMe, an algorithm that generates cost-optimal action sequences given the action parameters, and investigate several extensions of it, motivated by different potential application domains. We provide an analysis of the algorithms, including proofs for a number of properties of optimal solutions that we show align with typical

* M. Veloso is currently Head of AI Research at JPMorgan Chase.

This is a post-peer-review, pre-copyedit version of an article published in *Autonomous Agents and Multi-Agent Systems*. The final authenticated version is available online at: http://dx.doi.org/10.1007/s10458-020-09458-7.

K. Baraka
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
INSEC-ID / Instituto Superior Técnico, Universidade de Lisboa, Porto Salvo, Portugal
E-mail: kbaraka@andrew.cmu.edu

F. S. Melo
INSEC-ID / Instituto Superior Técnico, Universidade de Lisboa, Porto Salvo, Portugal

M. Couto
INSEC-ID, Porto Salvo, Portugal

M. Veloso
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

human provider strategies. Finally, we instantiate our theoretical framework in the context of robot-assisted therapy tasks for children with Autism Spectrum Disorder (ASD). In this context, we present methods for determining action parameters based on a survey of domain experts and real child-robot interaction data. Our contributions unlock increased levels of flexibility for agents introduced in a variety of assistive contexts.

Keywords Assistive agents · Sequential decision-making · Human-robot interaction · Autism spectrum disorder

1 Introduction

Autonomous interactive agents are being increasingly introduced to assist humans in a variety of tasks. In particular, they are starting to complement therapeutic and educational interventions. In therapy contexts, virtual agents have been used in applications such as speech therapy [44] or mental health [24], and robots have been used for both physical and cognitive rehabilitation [25, 15]. Specifically, robots have been identified to be good candidates for use in therapy for children with Autism Spectrum Disorder (ASD) [3]. Because of their predictability, controllability and simple social behavior, robots have shown promise in 'socially assisting' such individuals, who primarily suffer from communication and social impairments [39, 13, 12]. On the other hand, in education contexts, Intelligent Tutoring Systems (ITS) are an example of agents that provide automated and personalized teaching and assessment to students [2]. In addition, robotic agents have been used to promote engagement in learning, specifically with children, targeting a diverse set of skills [43, 47, 8].

Motivated by the use of such agents in these highly diverse contexts, we contribute in this work a context-independent mathematical framework for structured tasks in which an agent playing the general role of a 'provider' is assisting a 'receiver' to achieve a goal. Our framework is inspired by task structures prevalent across a variety of human-based provider-receiver interactions, such as therapy-patient or teacher-student interactions. Based on this framework, we first contribute algorithms to generate appropriate sequences of actions for the provider agent. We then instantiate our framework in a robot-assisted therapy setting involving children with ASD, and provide some preliminary results on its applicability to this challenging domain. This article extends our previous contributions in [5], by presenting several extensions to the basic algorithm, as well as significant additional theoretical and experimental analyses. To motivate the different components of our approach, we first provide some background on typical provider-receiver interaction structures in the context of therapy and education.

1.1 Background

Human-based provider-receiver interactions are typically structured in tasks with a clearly defined goal — e.g., eliciting a desired behavior, or obtaining a correct answer from the receiver —, which can often be measured in a binary way: *success* or *failure*. Moreover, in tasks meant to build or improve receiver skills over time through learning or training, the provider often uses a *hierarchy of actions* with the aim of assisting the receiver in achieving the goal. Actions at higher levels in the hierarchy provide higher levels of assistance. In Table 1, we show examples of such action hierarchies used in practice in three different fields but possessing a similar structure. The first one, taken from [29], is from the field of speech therapy, where cueing is used to assist patients suffering from aphasia [19], a disorder affecting speech production. The second one, taken from [30] and part of our case study in this article, is used to train attention skills in therapy for children with ASD. The third one, taken from [31], is from an education context involving giving hints on a science problem. Generally, actions of higher level in the hierarchy are more likely to cause a success. Hence, we can think of such hierarchies as sets of actions ordered by *increasing success probabilities*. It is important to note that those success probabilities are different for each receiver, depending on their abilities. Our generic problem formulation considers hierarchies with an arbitrary number of actions ordered by increasing arbitrary success probabilities.

In addition to success probabilities, provider actions have associated implicit *costs*. In a general assistive context, higher levels of assistance are typically associated with higher costs, such as energy, time, or other resources spent to provide the assistance. In therapy contexts, the concept of cost is more nuanced. Depending on the context and task, therapeutic costs may come from a number of factors, including explicitness, difficulty, or stimulus intensity. The more an action differs from what is considered desirable or natural, the higher its therapeutic cost because it is less likely to build the desired receiver skills over time [20]. In education contexts, costs could capture the amount of information revealed in a hint, or the difficulty level of a prompt. Although we practically expect costs to be increasing with action level, our problem formulation considers arbitrary positive action costs.

Furthermore, there often exists a time constraint in tasks led by providers. This constraint can come from a number of factors, including time frame of a task or a session, engagement ability of the receiver, or energy of the provider. In our problem formulation, we include a fixed *horizon* as part of our model. Every time step at which an action is executed being denoted as a *trial*, the horizon corresponds to the maximum number of trials that are allowed in a single task.

Finally, as part of their typical strategies, human providers constantly *personalize* the tasks according to the receiver profile. This personalization includes selecting the appropriate level to start in the hierarchy, corresponding to the generally idea of 'grading', extensively used in therapy and education [20]. It also includes personalizing the way one follows the hierarchy, including po-

Table 1 Examples of action hierarchies used in speech therapy, autism therapy, and education, adapted from [29], [30] and [31] respectively. Actions of higher level in the hierarchy provide more assistance, hence have a higher success probability.

Domain	*Speech therapy for aphasia*	*Autism therapy for joint attention training*	*Science education*
Success (task goal)	Patient retrieves word correctly	Child looks at target object	Student enters units correctly in a problem
Assistive action levels	1 - "What's this called?" 2 - Directions to state function of item 3 - Directions to demonstrate function 4 - Statement of function by clinician 5 - Statement and demonstration of function by clinician 6 - Sentence completion 7 - Sentence completion + silent articulation of first phoneme 8 - Sentence completion + vocalization of first phoneme 9 - Sentence completion + vocalization of first two phonemes 10 - Say "——"	1 - Say: "[Child's name], look!" (+ gaze shift) 2 - Say: "[Child's name], look at that!" (+ gaze shift) 3 - Say: "[Child's name], look at that!" (+ gaze shift + pointing) 4 - Activate target object	1 - "You have entered the right numbers, but units are wrong. Look in the problem. Enter units now." 2 - "You have entered [...] wrong. Distance is in meters. Time is in seconds. Enter units now." 3 - "You have entered [...]. Distance is in meters and should be written as 200m. Time is in seconds and should be written as 25s. Enter that now."

tential repetitions of action levels, skipping levels, or dynamically adapting to changes in performance. These personalization methods take into account assessed receiver abilities or past performance on the same task — e.g., level of impairment in the speech or ASD therapy — and student skills or past performance in education. Occupational therapists often refer to this process as finding the 'just right challenge' [40].

1.2 Research goal

The goal of this research is to devise a method to generate optimal action sequences to be followed by a provider agent, i.e., action sequences with *minimum expected cost*. Every action in the sequence has a probability of failing, in which case the agent executes the next action, and a probability of succeeding, which is associated with a reward and no subsequent action execution. The agent does not know ahead of time when a success will occur but knows the action parameters (success probabilities and costs). Hence, it can reason under uncertainty to plan for action sequences that balance urgency to achieve a success and parsimony in the selection of actions according to their costs. Figure 1 illustrates this goal through a generic example.

In our solution to this problem, we devise an algorithm that finds an optimal action sequence, given a set of action success probabilities and costs, and rigorously analyze its mathematical properties. We then present several extensions of the basic algorithm, relaxing the assumption that the action parameters are fixed. To illustrate our approach, we instantiate our framework in a robot-assisted ASD therapy scenario. We first estimate action costs using expert data, then estimate success probabilities for a given pre-assessed child profile, based on real child-robot interaction data. Our algorithm ultimately returns different optimal sequences for different child profiles, hence achieving personalization according to child profile.

1.3 Contributions

The contributions this article makes can be summarized as follows:

1. A mathematical formulation of the optimal action sequence generation problem in a general provider-receiver context (Section 3.1).
2. OAssistMe, a linear-time optimal algorithm based on dynamic programming that solves the above problem (Section 3.2).
3. A theoretical analysis of optimal solutions, including proofs of monotonicity and convergence, and constraints on model parameters for suitable algorithm behavior in relation to our application realm (Section 3.3).
4. Several extensions of OAssistMe, injecting different assumptions about dependence of action parameters on action history. These extensions are: Trial-Sensitive (TS), Cost-Sensitive (CS), and Repetition-Sensitive (RS) versions of the algorithm (Section 4).

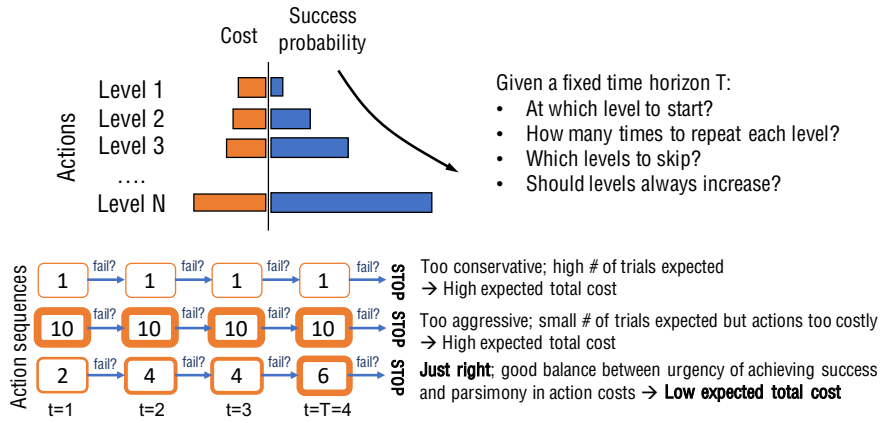

Fig. 1 Generic example illustrating the concept of optimizing action sequences in relation to given action success probabilities and costs. Numbers in action sequences represent action levels. At every trial, in case of failure the agent continues the sequence, and in case of success gets a reward and aborts execution. Graphs and action sequences shown are only meant for illustrative purposes.

5. A formative evaluation of the framework in a robot-assisted ASD therapy setting, including a methodology for determining action parameters, namely:
   (1) An online survey with psychologists for determining action costs.
   (2) A probabilistic model of children response to robot actions, based on data collected during a real interaction between a humanoid robot and 10 children with different ASD levels (Section 5).

2 Related Work

Relevant to our approach are works in the fields of human-agent interaction, ITS, healthcare interventions, and robot-assisted autism therapy. We briefly discuss these next.

Probabilistic models for human-agent interaction While probabilistic models are widely used by agents operating in uncertain environments [38], they seem to be much less used in human-interactive contexts. If some human modeling approaches incorporate uncertainty as part of the model [10, 22], planning and adaptation in typical human-computer interaction scenarios rarely accounts for this uncertainty. In the field of human-robot interaction however, probabilistic models have gained more interest, and reached the ability to model mutual adaptation between human and robot in certain collaborative contexts [33].

In this work, we rely on a simple probabilistic model of the receiver's response to the provider's actions, which introduces uncertainty in the reasoning process of the agent.

Intelligent Tutoring Systems These are computer-based solutions that provide personalized and immediate tools and feedback to learners, with minimal human intervention. There is a very large literature on ITS and a number of approaches consider variations of the personalization problem related to this article's goals, according to various context-dependent variables, often with an assumption of partial observability [14,7]. Grover et al. (2018) specifically frame ITS as a collection of planning problems [17]. Two such problems closest to ours in the ITS literature are the problem of optimal teaching sequence generation [9], and the problem of hint generation [37,6], which aim at providing tailored context-specific content according to student performance. These problems have mainly been studied in the context of teaching highly structured concepts such as programming or logic proofs [37,6]. Most state-of-the-art methods rely on a large amount of data, based on algorithms similar to recommender systems, while earlier work tends to be more analytic and model-based [32]. In an agent-based therapy setting, such amount of data is far from being available for a number of reasons, including scarceness of available technologies for special populations, higher-than-normal variability of profiles, and data privacy. As a result, the application of these types of algorithms to therapy contexts is difficult. In this article, a relatively small amount of data is needed to be able to estimate model parameters for the generation of personalized action sequences. Even though the ITS literature has tackled more complex problems in the past, many of them are not transferable to other domains falling under the provider-receiver interactive paradigm.

The present work contributes a principled analysis of a simple and general model for certain types of tasks, which we believe may be of valuable across a variety of domains. Nevertheless, the ITS field may provide a valuable line of research to accelerate advances in other types of agent-based interventions in the future, especially as more data become available.

Healthcare interventions Computational approaches to healthcare interventions have been studied both from a technological and decision-making standpoint. From the technological standpoint, Hoey et al. (2013) describe a approach to applying decision-theoretic models to personalized assistive technology for in-home use [21]. Their COACH system (Cognitive Orthosis for Assistive aCtivities in the Home) is closest to the type of tasks we consider in this work, as it focuses on prompting the user to complete a task over a short time frame, using actions with increasing levels of specificity and costs. However, it is unclear how they determine the parameters in their model (e.g., costs). In this work, we favor a more principled approach to investigating how such parameters can be determined from expert and interaction data.

From the decision-making standpoint, there is a body of literature dedicated to decision-theoretic approach to medical intervention that take into account uncertainty of costly action outcomes. They include Markov Decision Processes (MDP) [48,1] and Partially Observable MDP's (POMDP) [18], often with a finite time horizon, as is assumed in this work. These modeling approaches typically operate over much longer time scales, e.g., the course of

a treatment, or maybe even a lifetime. Applications include epidemic control, drug infusion, organ transplantation, screening and treatment, among others [41]. While the algorithms presented in this work can be seen as special cases of finite-time MDP's, their structure creates provable properties of optimal solutions (see for example Theorem 2) that are not necessarily valid in more general formulations.

Personalized robotic interactions for ASD intervention We have established the importance of personalization in relation to general provider-receiver interactions. In this article, we use robot-assisted therapy for children with ASD as a case study to illustrate our approach (Section 5). While the personalization problem has been tackled in general child-robot interactions [28], it remains an important open problem with special needs populations. As children with ASD specifically present immense variability, powerful automated personalization mechanisms are needed for the success of such robotic solutions. Most existing work to date still heavily relies on tele-operation, or content customization [34]. Some architectures for personalization using child behavioral profiles have been devised [12], but their effectiveness in practice remains to be demonstrated. Additionally, real-time adaptation is another major aspect of autonomy, albeit out of the scope of this work. It appears that the only successful real-time adaptive solution to date relies on affective adaptation through multimodal measurements of affect to regulate a basketball-based task [11].

The illustrative tasks used in this work build on structured tasks from the Autism Diagnostic Observation Schedule (ADOS-2) [30], the gold standard for autism diagnosis. Some researchers have used this tool to inform the design of robotic interactions, including robot-assisted intervention [45] and diagnosis [36]. The tasks used in this work are similar to the line of work of Warren et al. [45], which inspired our testing scenario. By automatically generating optimal sequences for every child profile, we believe our algorithmic contribution can enrich robot-based therapy scenarios and possibly have an impact on their clinical effectiveness.

3 Mathematical framework

This section describes our contributed framework, which accounts for probabilistic outcomes of costly actions under a fixed time horizon. Within this framework, we present OAssistMe, an algorithm that generates optimal sequences, and analyze some of its properties.

3.1 Problem formulation

We frame the general problem informally defined in Section 1 as an optimization problem that takes into account both action costs and success probabilities.

3.1.1 Input

Assume we have a hierarchy of actions $1, ..., N$, representing increasing levels of assistance. Further assume actions have fixed success probabilities $p(1) < ... < p(N) \in (0, 1)$ and costs $c(1) \neq ... \neq c(N) \in (0, \infty)$. Success probabilities of exactly 1 or 0 are not realistic and can lead to singularities in our analysis, which is why they are excluded. Additionally, one can argue that if two costs were equal the action with lower success probability should never be selected by an optimal agent, which makes that action irrelevant to the agent. The same argument applies for equal probabilities, in which case an optimal agent should always select the less costly action. As a result, we do not allow actions with equal probabilities and/or costs. Also note that while in application domains of interest we expect costs to be increasing, our problem formulation does not impose an order on the costs.

The outcome of every action a is assumed to be a Bernoulli random variable with success probability $p(a)$. Also assume there is a reward (negative cost) $R > 0$ associated with a success and no cost associated with a failure. Note that this last assumption does not compromise generality, since if failures are considered to be costly, the cost of a failure can be absorbed in the action costs and the value of R can be increased by the absolute value of that cost.

3.1.2 Setup

At each discrete trial $t \geq 1$, the agent selects an action a_t and observes the outcome. If a failure occurs, a new action is executed at the next trial. If a success occurs or the maximum number of the horizon T is reached, the process stops. Trials are assumed to be independent, meaning the values of $c(a)$ and $p(a)$ are not influenced by previous actions in the sequence (later in Section 4, we will relax this assumption).

3.1.3 Goal

The goal is to find an action sequence of length T that *minimizes the expected overall cost*. The overall cost of a sequence is defined as the sum of costs of individual actions minus the reward R if a success occurs. Note that according to the setup above, the planned action sequence is only executed until a success occurs or the horizon T is reached, after either of which the agent stops. In the next subsection, we derive a closed form for the expected overall cost of a sequence, which corresponds to the objective function to be minimized.

3.1.4 Objective function

Let $\langle a_1, a_2, a_3, ..., a_T \rangle$ be an arbitrary action sequence. The probability P_t that a success occurs at trial t (upon which the agent stops) is given by:

$$P_t = p(a_t) \prod_{\tau=1}^{t-1} (1 - p(a_\tau)) \tag{1}$$

Note that for the same sequence we have the following recursive relation:

$$P_{t+1} = P_t \frac{p(a_{t+1})(1 - p(a_t))}{p(a_t)}, \;\; P_1 = p(a_1) \tag{2}$$

Denoting C_t the cost of the sequence up to t:

$$C_t = \sum_{\tau=1}^{t} c(a_\tau) \tag{3}$$

We also have the following recursive relation:

$$C_{t+1} = C_t + c(a_{t+1}), \;\; C_1 = c(a_1) \tag{4}$$

The expected overall cost of the actual sequence followed (aborted upon the occurrence of the first success) is hence given by:

$$O_T = \sum_{t=1}^{T} P_t(C_t - R) + (1 - \sum_{t=1}^{T} P_t) C_T \tag{5}$$

The first term represents all cases where a success occurs, while the second term represent the case where a success doesn't occur after all T trials (which is why it doesn't include R). An optimal action sequence $\langle a_1^*, a_2^*, ..., a_T^* \rangle$ is a sequence that minimizes the objective O_T.

3.2 Optimal sequence generation

We now present an algorithm to compute the solution to the optimization problem defined above.

Single-trial case For $T = 1$, the expected overall cost is $c(a) - p(a)R$, and the optimal action is $a^* = \arg\min_a \{c(a) - p(a)R\}$.

Multi-trial case We can relate the objective O_T of sequence $\boldsymbol{\Pi}_T = \langle a_1, ..., a_T \rangle$ and the objective O_{T-1} of sequence $\boldsymbol{\Pi}_{T-1} = \langle a_2, ..., a_T \rangle$ (note the indices) as follows:

$$O_T = (1 - p(a_1)) O_{T-1} + c(a_1) - p(a_1)R \tag{6}$$

Therefore, the optimal solution for horizon T can be obtained by first solving for the optimal solution for horizon $T - 1$ then appending at the beginning of the computed sequence the action a that minimizes the quantity $(1 - p(a))O_{T-1}^* + c(a) - p(a)R$, where O_{T-1}^* is the optimal objective function for horizon $T - 1$.

Hence, we have the following recursive relations,

$$O_T^* = \min_a \{(1 - p(a))O_{T-1}^* + c(a) - p(a)R\},\ O_1^* = \min_a \{c(a) - p(a)R\} \quad (7)$$

$$\begin{aligned} \boldsymbol{\Pi}_T^* &= \left\langle \arg\min_a \{(1 - p(a))O_{T-1}^* + c(a) - p(a)R\}, \boldsymbol{\Pi}_{T-1}^* \right\rangle, \\ \boldsymbol{\Pi}_1^* &= \left\langle \arg\min_a \{c(a) - p(a)R\} \right\rangle \end{aligned} \quad (8)$$

Based on Equations (7) and (8), we devise the OAssistMe algorithm (see Algorithm 1), based on dynamic programming. The resulting algorithm has linear time complexity in T and N ($\mathcal{O}(TN)$).

Algorithm 1 OAssistMe: Linear-time algorithm to find an optimal action sequence for horizon T.

```
1: procedure OAssistMe(p,c,T,R)    ▷ p and c are vectors containing p(a)'s and c(a)'s
2:     O_part ← c − pR
3:     O ← O_part
4:     O* ← min O
5:     Π ← ⟨arg min O⟩
6:     for i ← 1, ..., T − 1 do
7:         O ← (1 − p)O* + O_part
8:         O* ← min O
9:         Π ← ⟨arg min {O}, Π⟩
10:    end for
11:    return Π
12: end procedure
```

For the sake of illustration, we present in Figure 2 a simple worked example with three generic assistive actions Low-assist – Medium-assist – High-assist, a horizon of 4, and sample action parameters.

3.3 Analysis of optimal solutions

We now present some theoretical results regarding optimal action sequences generated by the OAssistMe algorithm. We start by demonstrating a number of properties of optimal solutions, then briefly discuss a graphical representation of the problem and the role of the R parameter. We end by a brief interpretation of relevant results in relation to typical human provider strategies. In our analysis, we assume that ties are handled in a deterministic way, for example by always preferring lower actions.

3.3.1 Properties of optimal solutions

We present several properties of O_T^*, including monotonicity and convergence properties, and use those results to prove that *all optimal action sequences are monotonic* in t. Detailed proofs are included as an appendix.

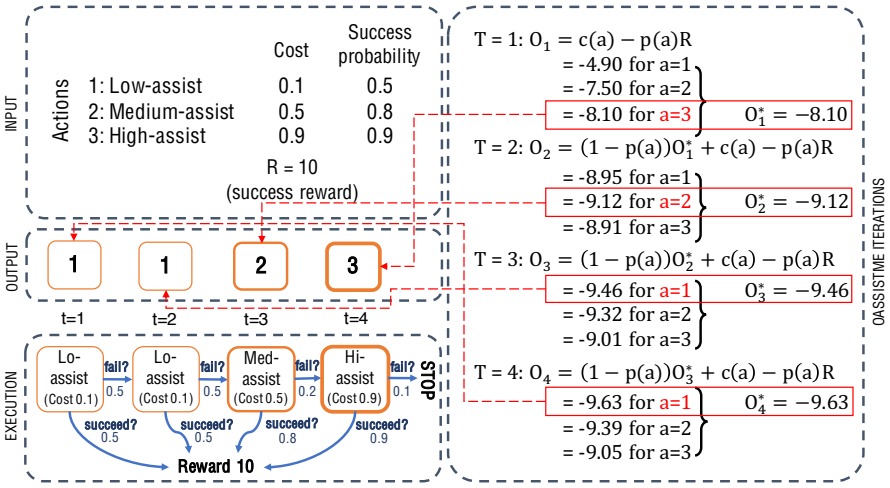

Fig. 2 Worked example with $N = 3$, $T = 4$, and sample action parameters, showing: the computation of the optimal objective at every iteration, the resulting optimal action sequence, and the practical execution of the action sequence by the agent. Numbers below action outcomes represent probabilities.

Our results are structured along the following three (mutually exclusive) cases:

a. $O_1^* > 0$, or equivalently $R < \min_a c(a)/p(a)$
b. $O_1^* < 0$, or equivalently $R > \min_a c(a)/p(a)$
c. $O_1^* = 0$, or equivalently $R = \min_a c(a)/p(a)$

The first result provides bounds on values for O_T^*.

Lemma 1 *For any T, we have one of:*

a. $0 < O_T^* < \min_a c(a)/p(a) - R$
b. $0 > O_T^* > \min_a c(a)/p(a) - R$
c. $0 = O_T^* = \min_a c(a)/p(a) - R$

Building on this result, we can show the following about O_T^* as a function of T.

Lemma 2 *O_T^* is monotonic in T. In particular, it is one of:*

a. strictly increasing, i.e., $O_{T+1}^* > O_T^*$ for all T
b. strictly decreasing, i.e., $O_{T+1}^* < O_T^*$ for all T
c. constant, i.e., $O_{T+1}^* = O_T^*$ for all T

As a result, at every new iteration of the algorithm the computed value of O^* follows a consistent evolution, increasing, decreasing, or remaining constant, depending on the case, as the problem size is increased. Even though in practice horizons considered are relatively small, one might wonder, for the sake of better theoretical understanding, how such values behave at very large T.

Theorem 1 *O^*_T **converges** to $\min_a c(a)/p(a) - R$ as T goes to infinity.*

This result suggest that for T large enough, actions that are appended as T is further increased will stabilize to $\arg\min_a c(a)/p(a)$. In addition, for an infinite horizon, the optimal sequence becomes constant. In other words, an optimal agent should only select action $\arg\min_a c(a)/p(a)$ until a success occurs. Building on Lemmas 1 and 2, we now state our most important result regarding general properties of optimal solutions.

Theorem 2 *If $\boldsymbol{\Pi}^*$ is an optimal sequence, then it is monotonic in t. In particular, $\boldsymbol{\Pi}^*$ is one of:*

a. ***nonincreasing****, i.e., $a^*_1 \geq a^*_2 \geq ... \geq a^*_T$*
b. ***nondecreasing****, i.e., $a^*_1 \leq a^*_2 \leq ... \leq a^*_T$*
c. ***constant****, i.e., $a^*_1 = a^*_2 = ... = a^*_T$*

Note that this result holds for any number N of actions such that $p(1) < ... < p(N)$, and for arbitrary costs $c(a) > 0$. For proofs of the four results above and how they build upon each other, we refer to the appendix.

3.3.2 Graphical representation of O^*_T versus O^*_{T-1}

Some of the results presented above may be better understood with a graphical view on the problem. In the update function relating O_T to O_{T-1} (Equation (6)), every action a contributes a different linear relationship between the two quantities, with a different slope $(1 - p(a))$ and generally different y-intercept $(c(a) - p(a)R)$. As a result, according to Equation (7), the relationship between O^*_{T-1} and O^*_T is piecewise linear. Figure 3 shows a graphical representation of this relationship with sample costs and success probabilities falling in case (b). As can be noticed, changing the value of R effectively translates the curve without changing its shape, nor the relative location of the convergence point.

3.3.3 Effect of R parameter

From the point of view of the generated optimal action sequences, the R parameter dictates how aggressively the agent tries to achieve a success. For all cases (a)–(c), increasing the value of R results in action sequences with equal or higher actions at every trial, and vice versa. This can be shown in a similar way to the proof of Theorem 2 (included in the appendix), where instead of comparing the action selected at iteration T versus $T-1$, we compare actions selected at the same iteration but with different R values. As a result, we conclude that the R parameter effectively controls the total probability of failure $\prod_{t=1}^{T}(1 - p(a_t))$. Higher R values will result in lower or equal failure probability, and vice versa. In practice, one can set a threshold on the tolerance of failure and select the R parameter to meet that threshold, as we will do in our evaluation example (Section 5).

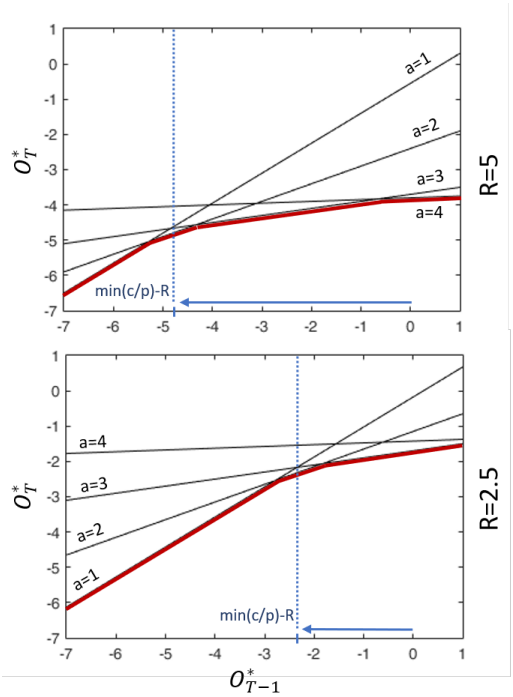

Fig. 3 Sample graphical representation of O^*_T versus O^*_{T-1} as a piecewise linear function (red curve) for different R values falling in case (b). Arrows represent the direction of evolution of $O^*_{T-1} < 0$, and the dotted lines show the convergence point. The resulting optimal sequences for $T = 4$ are $\langle 2, 2, 3, 4\rangle$ $(R = 5)$ and $\langle 2, 2, 3, 3\rangle$ $(R = 2.5)$.

3.3.4 Interpretation in relation to human provider strategies

In a typical provider-receiver interaction, a failure on the receiver side prompts the provider to repeat actions or increase assistance levels to gradually guide the receiver towards a success. This type of strategy is in accordance with the concept of grading in therapy [20], or scaffolding in education [16]. These results generally consist in adapting the assistance level according to the need and response of the receiver. Concretely, this means that the action sequences followed by human providers are typically nondecreasing.

This observation is consistent with our Theorem 2, case (b), which states that optimal action sequences generated by OAssistMe are not only monotonic but also nondecreasing. As a result, we conclude that in practice, *a value of R larger than* $\min_a c(a)/p(a)$ should be selected to incentivize increasing or maintaining assistance levels throughout the computed optimal sequences. In light of this result, in the rest of this article we will assume that $R > \min_a c(a)/p(a)$ for all practical uses of the OAssistMe algorithm.

4 Framework extensions

The framework presented in the previous section has relied on the assumption that action parameters (i.e., costs and success probabilities) are fixed. While costs are assumed to be intrinsic to the actions themselves and can be reasonably assumed to be fixed in a given domain, success probabilities may in practice possess some dependency on previous actions executed by the agent. As such, it is useful to consider the option that the success probability be a function of both the action executed at trial t and the history of actions *up to but not including* trial t, denoted $\mathbf{h}_t = \langle a_1, ..., a_{t-1} \rangle$. We then denote the success probability function as $p(a, \mathbf{h}_t)$.

Generally, in the presence of dependence on history, the problem becomes a MDP where states contain an encoding of all possible histories $\mathbf{h}_t$ for $t = 1, ..., T$. Since this number, and as a result the number of model parameters, grow exponentially with the number of trials, it is desirable to identify what features of $\mathbf{h}_t$ may have an effect on the action parameters in practice. Inspired by our potential application domains, we consider three assumptions about how $\mathbf{h}_t$ can affect success probabilities. They are summarized in the three following cases:

1. $p(a, \mathbf{h}_t) = p(a, t)$, i.e., success probabilities are only affected by the number of previous trials, regardless of what actions were executed before the current trial. We call this case **trial-sensitive (TS)**.
2. $p(a, \mathbf{h}_t) = p(a, C_t)$, where C_t is the cost of sequence $\mathbf{h}_t$. We call this case **cost-sensitive (CS)**.
3. $p(a, \mathbf{h}_t) = p(a, n_t(a))$, where $n_t(a)$ represents the number of occurrences of action a in $\mathbf{h}_t$. We call this case **repetition-sensitive (RS)**.

We now motivate and discuss extensions of the framework to accommodate each of these cases, then analyze properties of optimal solutions as well as the time complexity of the extended algorithms. We end this section with a simulated example comparing each case to the basic case.

4.1 Trial-sensitive case (TS)

In a therapy and education context, it is somehow intuitive to consider a slight *positive* increase in success probabilities as a function of number of trials. For example, giving hints on an educational exercise may increase the likelihood of the student solving the problem correctly when the next hint is given. Similarly, in a therapy task that involves sensory integration [40], more trials translate into increasing overall sensory stimulation, which may make the receiver more likely to respond to individual stimuli.

For the sake of generality, the modifications to the original framework do not assume a specific relationship (e.g., positive, negative) between trial and success probabilities. With the success probability function $p(a, t)$ depending

on both action and trial, the recursive relations described in Equations (7) and (8) become:

$$\begin{aligned} O^*_\tau &= \min_a \{(1 - p(a, T - \tau + 1))O^*_{\tau-1} + c(a) - p(a, T - \tau + 1)R\}, \\ O^*_1 &= \min_a \{c(a) - p(a, T)R\} \end{aligned} \tag{9}$$

$$\begin{aligned} a^*_{T-\tau+1} &= \arg\min_a \{(1 - p(a, T - \tau + 1))O^*_{\tau-1} + c(a) - p(a, T - \tau + 1)R\}, \\ a^*_T &= \arg\min_a \{c(a) - p(a, T)R\} \end{aligned} \tag{10}$$

where T is the specified horizon and τ represents the number of decisions left. The revised algorithm in this case, denoted TS-OAssistMe, is summarized in Algorithm 2.

Algorithm 2 TS-OAssistMe: Trial-sensitive extension of OAssistMe where success probabilities are a function of trial.

procedure TS-OAssistMe(N,$p_{TS}(.,.)$,$c(.)$,T,R)
▷ p_{TS} is a function of action and trial and c is a function of action
  $O^* \leftarrow 0$
  $\boldsymbol{\Pi} \leftarrow \langle\rangle$
  for $i \leftarrow T, ..., 1$ **do**
    $\mathbf{O} \leftarrow \langle (1 - p_{TS}(a, i))O^* + c(a) - p_{TS}(a, i)R \ , \ a = 1, ..., N \rangle$
    $O^* \leftarrow \min \mathbf{O}$
    $\boldsymbol{\Pi} \leftarrow \langle \arg\min \{\mathbf{O}\}, \boldsymbol{\Pi} \rangle$
  end for
  return $\boldsymbol{\Pi}$
end procedure

4.2 Cost-sensitive case (CS)

In addition to the number of trials, the dependency on the history may be affected by previous action costs. Given that higher levels of assistance will typically be associated with higher costs, it may be the case that the success probability is *positively* affected by history cost. Back to our sample domains in the previous subsection, the success probabilities may for instance be sensitive to the total amount of information revealed by hints or cues (in an education or speech therapy context), or the total amount of stimulation provided (in a sensory integration therapy context).

As in the previous case, the modifications presented below do not assume that the relationship between cost of history and success probabilities has a specific form. With the success probability function $p(a, C_t)$ depending on both action and cost of history, we need to consider all relevant histories at every iteration of the algorithm. Because history cost is sensitive to the count of each

action, but not the actual sequence order, we can represent histories as tuples $\langle n_t(1), n_t(2), ..., n_t(N)\rangle$, where $n_t(a)$ represents the number of occurrences of action a. The updated recursive relations now need to be applied to every distinguishable history at each iteration, as follows:

$$\begin{aligned} O^*_\tau(\mathbf{h}_{T-\tau+1}) = \min_a \{ &[1 - p(a, C_{T-\tau+1})]\, O^*_{\tau-1}(\mathbf{h}_{T-\tau+1} \cup a) \\ &+ c(a) - p(a, C_{T-\tau+1})R\}, \qquad (11) \\ O^*_1(\emptyset) = \min_a \{ &c(a) - p(a, 0)R\} \end{aligned}$$

where the $\mathbf{h} \cup a$ operation adds action a to history $\mathbf{h}$ by incrementing $n_t(a)$.

Unlike in previous cases, the computation of the objective and the construction of the optimal action sequence are not performed in a synchronized way. Instead, the action sequence is obtained through backtracking after the computation of all O^* values is complete. For every computation of O^*, a corresponding action is stored. The final sequence builds from the first action in the sequence to the last action by appending the optimal actions successively, using the results from the backward pass. The revised algorithm in this case, denoted CS-OAssistMe, is summarized in Algorithm 3.

Algorithm 3 CS-OAssistMe: Cost-sensitive extension of OAssistMe where success probabilities are a function of cost of history.

procedure CS-OAssistMe(N,$p_{CS}(.,.)$,$c(.)$,T,R)
▷ p_{CS} is a function of action and history cost and c is a function of action
  $\{\mathbf{H}_1, ..., \mathbf{H}_T\} \leftarrow$ GenAllUnorderedHists(N, T)
  ▷ Generates all possible **unordered** histories $\mathbf{H}_i$ of size $i-1$, represented as sets of tuples $\langle n_t(1), ..., n_t(N)\rangle$, where $n_t(a)$ represents the number of occurrences of action a
  $\mathbf{O}^*_{T+1} \leftarrow \mathbf{0}$
  for $i \leftarrow T, ..., 1$ **do**
    for all $\mathbf{h} \in \mathbf{H}_i$ **do**
      $\mathbf{O}_{i,\mathbf{h}} \leftarrow \left\langle (1 - p_{CS}(a, C_i))O^*_{i+1,\mathbf{h}\cup a} + c(a) - p_{CS}(a, C_i)R\ ,\ a = 1, ..., N \right\rangle$
      $O^*_{i,\mathbf{h}} \leftarrow \min_a \mathbf{O}$
      $\tilde{\boldsymbol{\Pi}}_{i,\mathbf{h}} \leftarrow \arg\min_a \mathbf{O}$
    end for
  end for
  $\boldsymbol{\Pi} \leftarrow$ BacktrackOptimalDecisions($\tilde{\boldsymbol{\Pi}}$)
  return $\boldsymbol{\Pi}$
end procedure

4.3 Repetition-sensitive case (RS)

There may be interesting effects linked to the repetition of the same action during an interaction with a receiver. For example, some research suggests that unpredictable ('surprising') sequences lead to higher attention responses [23],

which may for example impact how patients respond to therapeutic tasks involving attention mechanisms [26]. This observation suggests that predictable sequences, such as ones that favor repeating previously executed actions over selecting new ones, may have a *negative* effect on success probabilities. In an education scenario, the same effect may be observed, where a hint is only helpful the first time it is shown. If a hint has been shown before and failed to cause a success, then it is reasonable to assume that subsequent trials of the same hint may have lower success probability.

As before, the modifications presented below do not assume that the relationship between number of repetitions and success probabilities has a specific form. The recursive relations are very similar to the CS case. The representation of history is identical since it needs to capture the count for each distinguishable action in the history, but is agnostic to the order. The updated equations are:

$$\begin{aligned} O^*_\tau(\mathbf{h}_{T-\tau+1}) = \min_a \{ &[1 - p(a, n_{T-\tau+1}(a))]\, O^*_{\tau-1}(\mathbf{h}_{T-\tau+1} \cup a) \\ &+ c(a) - p(a, n_{T-\tau+1}(a))R\}, \qquad (12) \\ O^*_1(\emptyset) = \min_a \{ &c(a) - p(a,0)R\} \end{aligned}$$

The optimal action sequence construction is identical to the CS case. The revised algorithm in this case, denoted RS-OAssistMe, is summarized in Algorithm 4.

Algorithm 4 RS-OAssistMe: Repetition-sensitive extension of OAssistMe where success probabilities are a function of the number of action repetitions. It is identical to Algorithm 3 except for the update equation in line 6.

1: **procedure** RS-OAssistMe(N,$p_{RS}(.,.)$,$c(.)$,T,R)
▷ p_{RS} is a function of action and repetitions of that action; c is a function of action
2: $\{\mathbf{H}_1, ..., \mathbf{H}_T\} \leftarrow$ GenAllUnorderedHists(N, T)
3: $\mathbf{O}^*_{T+1} \leftarrow \mathbf{0}$
4: **for** $i \leftarrow T, ..., 1$ **do**
5: **for** all $\mathbf{h} = \langle n_i(1), ..., n_i(N) \rangle \in \mathbf{H}_i$ **do**
6: $\mathbf{O}_i \leftarrow \left\langle [1 - p_{RS}(a, n_i(a))]\, O^*_{i+1,\mathbf{h}\cup a} + c(a) - p_{RS}(a, n_i(a))R\ ,\ a = 1, ..., N \right\rangle$
7: $O^*_{i,\mathbf{h}} \leftarrow \min_a \mathbf{O}$
8: $\tilde{\boldsymbol{\Pi}}_{i,\mathbf{h}} \leftarrow \arg\min_a \mathbf{O}$
9: **end for**
10: **end for**
11: $\boldsymbol{\Pi} \leftarrow$ BacktrackOptimalDecisions($\tilde{\boldsymbol{\Pi}}$)
12: **return** $\boldsymbol{\Pi}$
13: **end procedure**

4.4 General history-dependent algorithm (G-OAssistMe)

The structure of the CS and RS cases represent the most general way of incorporating dependence on history into the success probability function. These algorithms can easily be extended to the case where success probability is a function of an arbitrary number of features of the history, in addition to the action. In this case, given the most efficient representation of history for the features considered, one can run Algorithm 3 or 4, with the appropriate representation of history and the appropriate success probability function with no additional modifications.

4.5 Analysis of OAssistMe extensions

We now discuss the applicability of the properties reported in Section 3.3.1 to the algorithm extensions discussed in this section. We also provide an analysis of the time complexity of the different algorithm versions for comparison.

4.5.1 Properties of optimal solutions

While increasing the horizon T in OAssistMe resulted in simply appending actions to the beginning of the optimal action sequence of size $T-1$, this is not necessarily the case in the extensions presented in this section. Depending on the strength of the dependency on history, the action sequences may change more or less considerably as larger horizons are considered. Furthermore, our experimentation with these algorithms show cases of non-monotonic values of O^*_T (violation of Lemma 2) especially when injecting high dependence — even if monotonic — on history features. Furthermore, cases of monotonic O^*_T do not necessarily translate into monotonic optimal action sequences (violation of Theorem 2).

These observations suggest complex interaction effects between the different components of these more elaborate models, and hence the analysis of their behavior is best achieved through simulation (see Section 4.6).

4.5.2 Complexity analysis

We now provide a brief discussion and visualization of the time complexity for the different algorithms presented.

- OAssistMe: As mentioned in Section 3.2, the time complexity of the algorithm is $\mathcal{O}(TN)$.
- TS-OAssistMe: The number of operations is identical to OAssistMe, but different probability values are used at every iteration. Therefore, the complexity of the algorithm is still $\mathcal{O}(TN)$.

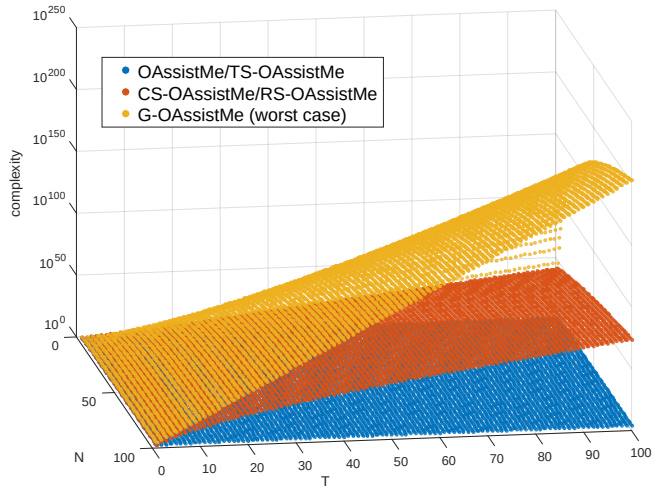

Fig. 4 Time complexity curves for the different algorithms as a function of horizon (T) and number of actions (N). The G-OAssistMe case shows the worst case of possible histories to consider. Curves represent upper bounds on running times up to a constant factor. Plots are based on theoretical values and not experimental running times.

– CS-OAssistMe: The total number of histories considered for the computation of O^* values is given by:

$$\sum_{\tau=1}^{T} \binom{\tau+N-2}{\tau-1} = \binom{N+T-1}{T-1} \leq \frac{(N+T-1)^{T-1}}{(T-1)!}$$

The backtracking step is linear in T, and hence has negligible complexity. The total time complexity of the algorithm is therefore $\mathcal{O}(N\frac{(N+T-1)^{T-1}}{(T-1)!})$, assuming that values of O^* are accessible in $\mathcal{O}(1)$ time — e.g., through a dictionary.

– RS-OAssistMe: As in the cost-sensitive case, the time complexity of the algorithm is $\mathcal{O}(N\frac{(N+T-1)^{T-1}}{(T-1)!})$.

– G-OAssistMe: In the worst case, the histories are represented fully as ordered sequences of actions. The total number of histories in this case is given by:

$$\sum_{\tau=1}^{T} N^{\tau} = \frac{N(N^T-1)}{N-1}$$

As a result, the complexity is $\mathcal{O}(\frac{N^2(N^T-1)}{N-1})$, which can be simplified to $\mathcal{O}(N^{T+1})$. Note that this result assumes negligible complexity for the computation of history feature(s).

Figure 4 shows a visualization of the different algorithm complexities for comparison.

4.6 Simulated example

To evaluate the effect of our different assumptions about the relationship between history and success probabilities (TS/CS/RS), we consider a simulated example. Later in Section 5, we will tie these frameworks to a therapy example.

We assume a logistic probability function of the form:

$$p(a, f(\mathbf{h}_t)) = [1 + e^{-(\beta_0 + \beta_1 a + \beta_2 f(\mathbf{h}_t))}]^{-1} \quad \text{for } a = 1, ..., N \tag{13}$$

where β_i's are weights and $f(\mathbf{h}_t)$ is the feature of history to consider (either t, C_t, or $n_t(a)$). In light of the discussions in relation to potential application domains included for each case, we consider positive values for β_2 in the TS and CS cases, and negative values in case RS.

Furthermore, we assume a linear cost function of the form:

$$c(a) = \frac{c_{\max} - c_{\min}}{N - 1}(a - 1) + c_{\min}$$

Figure 5 shows sample action sequences generated by the different algorithms, showcasing the effect of increasing the weight of history dependency for each case.

On one hand, we can observe that TS-OAssistMe generally outputs more conservative action sequences as compared to OAssistMe. As the strength of the dependency (β_2) increases, the solutions become more conservative. This observation can be explained by the fact that as trials increase, actions become more effective at eliciting a success and hence generally lower actions are needed achieve a similar outcome.

On the other hand, solutions generated by the CS case are very similar to the ones generated by the TS case. The only difference occurs for a high value of β_2, where the algorithm is slightly more aggressive at trial 3 as compared to TS. This can be explained by the fact that the costs of the first two actions were relatively low to have actions deviate too significantly from the basic case. These observations suggest that the CS case captures similar aspects of history than the TS case, but with more resolution and hence can result in more intricate behavior depending on the action parameters.

Finally, the RS case generates interesting outputs, which seem to be less about how aggressive/conservative the algorithm is, but more about seeking 'novelty'. Even though differences are not obvious for low and medium values of β_2, for high β_2 the algorithm selects a new action at almost every trial. This observation can be explained by the fact that the algorithm is repetition-averse, as higher number of repetitions will decrease the agent's probability of achieving a success.

As part of our simulations, we also evaluated the effect of the R parameter on the optimal action sequences. Figure 6 illustrates the relationship between R and the total probability of failure. In practice, one could use such a plot to inform an appropriate selection of the R parameter according to a tolerance threshold on failure probability.

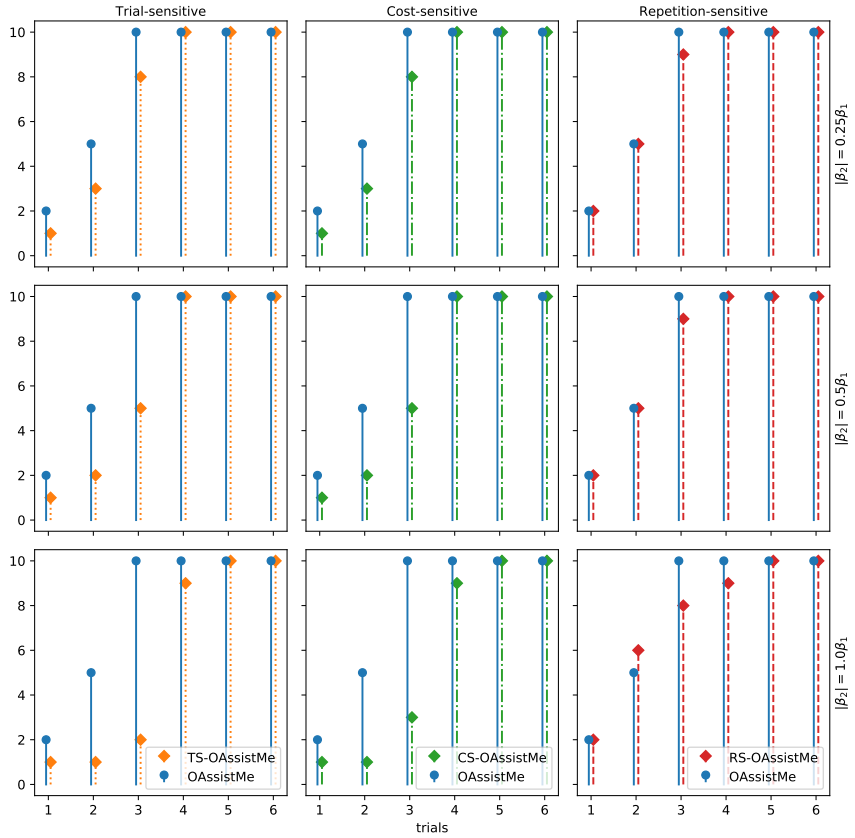

Fig. 5 Illustrative comparison of optimal action sequences generated by the different versions of OAssistMe. Parameters used: $N = 10, T = 6, R = 10^4, c_{\max} = 1.7, c_{\min} = 0.8, \beta_0 = 0, \beta_1 = 0.25, |\beta_2| = 0.25\beta_1, 0.5\beta_1, \beta_1$ (positive for case TS and CS; negative for case RS).

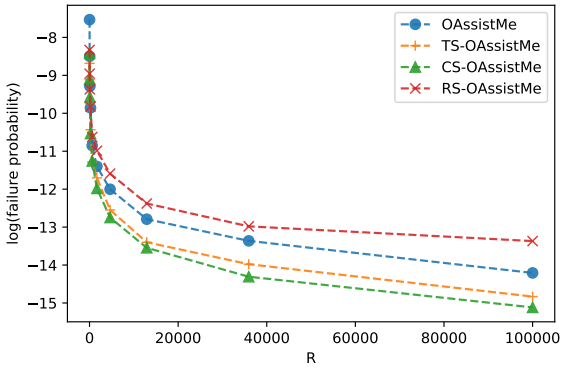

Fig. 6 Effect of R on the total probability of failure for the different versions of the algorithm. Parameters used were identical to those in Figure 5.

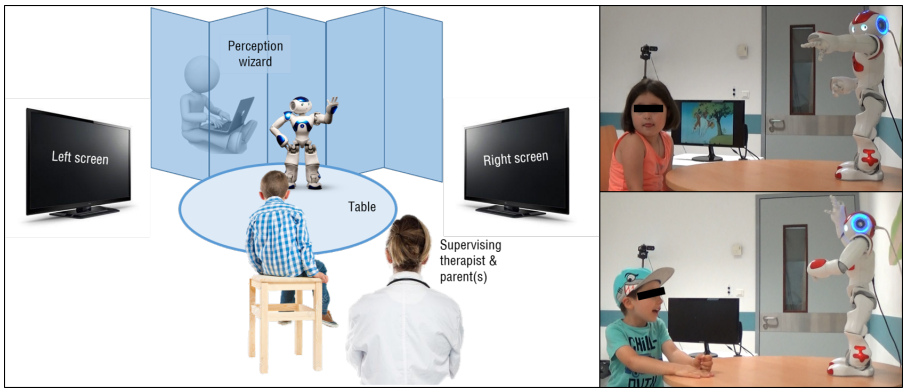

Fig. 7 Left: Diagram of the scenario considered. Right: Snapshots of actual ASD child-robot interactions in the JATT task (top) and NAME task (below).

5 Formative evaluation: robot-assisted autism therapy

To illustrate the potential applicability of our theoretical framework, we instantiate it in a robot-assisted therapy scenario for children with ASD. We focus on two tasks related to attention mechanisms, one of the core deficits of ASD. The use of a model-based method like ours for optimizing the robot behavior for each child is motivated by the fact that assessment is usually part of the regular therapy process. Hence, data collected during assessment can be used to more accurately estimate robot action parameters such as success probabilities on a personalized basis. In this section, we provide a methodology for estimating action costs on the provider side (therapeutic robot), and success probabilities on the receiver side (child with ASD). We then use the estimated parameters to generate optimal action sequences for different child profiles corresponding to different levels of impairment.

5.1 Scenario

Figure 7 shows the scenario considered, inspired by the work of Warren et al. [45]. The setup consists of a humanoid NAO robot standing on a table, at which the child is seated, and two LCD screens that can be triggered individually to show a video or a static picture. Within this setup, we consider two simple tasks, inspired by activities from the Autism Diagnostic Observation Schedule (ADOS-2) tool [30]. Even though ADOS-2 is a diagnostic tool, it partly uses tasks that typically appear in the context of therapy. The two tasks we consider in this work are:

– **Joint Attention (JATT) task**: The robot's goal is to direct the child's gaze from looking at the robot to looking at a target screen, using a combination of verbal and non-verbal cues. A success occurs if the child looks at the target screen, upon which a video is triggered as a reward.

– **Name Calling (NAME) task**: The robot's goal is to catch the child's attention when they are looking away from the robot (in our case, looking at the screen). The robot does so by calling the child's name and potentially using non-verbal cues. A success occurs if the child looks at the robot.

In both tasks, if no success occurs after a fixed timeout, it is considered a failure and a new trial starts. A perception 'wizard' informs the robot of a success when it occurs. Aside from the perception, the robot control was automated during the tasks.

We designed a hierarchy of four possible robot actions for each task, inspired by the hierarchy of presses in ADOS-2, and summarized in Table 2. Note that level $i+1$ is a replica of level i, with an added stimulus. For more details on how parameters of the scenario and the task were tuned, the robot control, and the role and validation of the perception wizard, we refer to [4].

Table 2 Hierarchies of robot actions with four levels, inspired by the ADOS-2 presses.

Task	Action level	Robot behavior
JATT	1	Gaze shift from child to target screen + "[Name], look!" (Static picture on both screens)
	2	Gaze shift + "[Name], look *at that!*" + *pointing* (Static picture on both screens)
	3	Gaze shift + "[Name], look at that!" + pointing + *muted video* on target screen
	4	Gaze shift + "[Name], look at that!" + pointing + video *with sound* on target screen
NAME	1	"[Name]!"
	2	"[Name], *look over here!*"
	3	"[Name], look over here!" + *blinking lights*
	4	"[Name], look over here!" + blinking lights + *waving arm*

We now present our methodology for estimating: (1) the (therapeutic) costs of the robot actions, and (2) their success probabilities for different child profiles.

5.2 Cost estimation: expert assessment

In a sensory integration context [40], such as autism therapy, it can be argued that the therapeutic cost comes mainly from how explicit a certain prompting or cueing action is. The more explicit the action — usually through the activation of more sensory channels, as is the case in our action hierarchies — the further away it moves from natural everyday scenarios, which should be avoided. For these reasons, we use *level of explicitness* as our measure for action cost in this context, and we expect this measure to increase as the action level increases. Furthermore, we assume that the actions costs, unlike the

success probabilities, do not vary according to the receiver's abilities. They were hence measured with respect to what is expected for a virtual Typically Developing (TD) (i.e., non-ASD) child matching the age of our targeted population. The cost measured would then capture for each action its deviance from a natural interaction with a TD child.

To determine these action costs, we ran a video-based online survey where professionals in the fields of clinical, educational and developmental psychology subjectively assessed the level of explicitness of our robot's actions shown as short video snippets. The responses for each robot action were gathered on a continuous scale (slider input) from 'Not explicit at all' (value of 0) to 'Completely explicit' (value of 100). Our sample consisted of 13 professionals from the areas of clinical (84.6%), educational (7.7%) and developmental (7.7%) psychology. Their ages ranged between 25 and 59 years ($M = 32.9$, $SD = 9.5$), and they were all female-gendered. Two participants completed only the first part of the survey, related to task JATT, and were included in the analysis. The participants were recruited through professional connections, and were not involved in the project. Informed consent was obtained prior to showing the survey, whereby we explained that the aims of our research was to assess a robot's actions when interacting with a child, for the aim of informing robotic interactions in this context in the future. We gave them some background information on the task, stating that they were embedded in a storytelling task involving screens. We specifically asked them to answer the questions with respect to an imaginary TD child with the name 'Manuel' (which the robot used for the NAME task), aged between four and six years. 'Explicitness' was defined as how easy it would be for Manuel to understand the expected response to the robot's prompt. The survey was in European Portuguese.

The collected data was analyzed using the SPSS software. The estimated costs and standard errors for each robot action are summarized in Table 3. Mauchlys test did not indicate any violation of sphericity neither for the JATT data ($\chi^2(5) = 9.63$, $p = 0.088$) nor for the NAME data ($\chi^2(5) = 6.44$, $p = 0.268$). A repeated measures ANOVA test showed no statistically significant differences between the mean costs for the JATT task: $F(3, 36) = 0.96$, $p = 0.423$, but showed statistically significant differences for the NAME task: $F(3, 30) = 4.82$, $p = 0.007$. A posthoc test with Bonferroni correction for multiple comparisons yielded statistical significance only between levels 1 and 4 ($p = 0.044$) for the NAME task. To measure inter-rater reliability, we calculated the intra-class correlation coefficient (ICC) based on a mean rating, one-way random effects model. We included both tasks in our analysis, and excluded two of the 13 participants who had a few missing items. The ICC estimate was 0.37 with a 95% confidence interval from -0.55 to 0.85 ($F(7, 80) = 1.58$, $p = 0.15$). This relatively low reliability value may be attributed to the different backgrounds of the raters, and their varying experience working with tasks similar to the ones considered.

The cost function follows an increasing trend along the hierarchy for both tasks, as expected, with the exception of action 3 in the NAME task, which

Table 3 Mean estimated cost results and standard errors for actions in the two tasks based on experts' responses.

	JATT		NAME	
Action level	Cost	SE	Cost	SE
1	57.92	9.71	38.18	8.63
2	62.23	8.51	50.91	10.31
3	65.77	7.56	47.63	11.23
4	74.85	7.46	72.73	9.90

records slightly lower cost than action 2. The only difference between the two actions is the presence of lights, which may have been hard to notice on the video version. Given that the standard errors are high, we attribute this result to noise.

However, it does not violate the assumptions of our framework, since it is valid for arbitrary positive cost functions.

5.3 Success probability estimation: child-robot interaction data analysis

The aim of this section is to determine a set of success probabilities that somehow accounts for individual differences. Previous work in computational psychiatry has looked at several behavioral modeling approaches accounting for individual differences [42,27]. In this work, we consider a simple model to estimate success probabilities of robot actions given a *child profile* for a specific task. The child profile is a categorization of the child's response to robot prompts into one of the four following discrete levels: High response (1), Medium response (2), Low response (3), and Minimal response (4). Higher values are typically associated with higher levels of impairment in attention mechanisms. The child profile was assessed by the robot as will be explained next.

5.3.1 Data collection

As part of a larger study involving interactive storytelling [4], we collected data on 10 ASD children's responses to the robot's actions in both tasks. The ages of our sample ranged between three and seven years; four were female and six male. Two had low ASD severity scores, six moderate and two severe. Informed consent was obtained from the parents prior to the sessions, including permission to record and share media for research presentation purposes.

Prior to the main interaction, the robot assessed the child profile for both tasks according to the ADOS-2 algorithm typically used by therapists for assessment. It consists in sequentially following the hierarchy of actions from lowest to highest level, and recording the first action level at which a success occurred. For each task, the value reported in the child profile is the rounded average of four measurements of the first successful action level. In case of

a tie (average falling exactly between two levels), the first measurement was discarded.

During the main interaction involving storytelling, the robot executed the JATT task at regular intervals, directing the child to a randomly chosen screen to show a video excerpt related to the story. As the video repeated, the robot performed the NAME task to call the child's attention back to the story. This process was repeated a total of four times throughout the story, resulting in a total of four instances per task. Every time the task was repeated, the content of the video was different and the children generally showed a sustained level of engagement and no clear learning effect. For each action the robot executed, we recorded the action level and the outcome (success/failure). The tasks ended if a success occurred or if the horizon was reached. To reduce any potential action ordering bias, the robot followed uniformly random action sequences for each task instance, with a horizon $T = 4$. More details on the setup and methodology can be found at [4].

Table 4 summarizes the general success rate for each task. Success rate is defined as the percentage of action sequences for which a success occurred. We report two metrics: the success rate within the exhaustion of the entire action sequence (within four trials), and the success rate within the first half of the action sequence (within two trials). Note that since every child was exposed to the same number of sequences, there was no need to consider different weighing across subjects.

Table 4 Results on two metrics of success rate for uniformly random action selection

Success rate metric (n=40)	JATT	NAME	Both tasks
Within 4 trials (full sequence)	100.00%	87.50%	93.75%
Within 2 trials (half sequence)	97.50%	65.00%	81.25%

5.3.2 Success probability model

Similar to our simulated example from Section 4.6, we use a logistic model of success probability according to the following equation:

$$p = (1 + e^{-\boldsymbol{\beta}\cdot\mathbf{f}})^{-1} \tag{14}$$

where vector $\mathbf{f}$ contains the predictor variables, in this order: constant term, child profile, action, and possibly a history feature (trial, cost or repetitions), while vector $\boldsymbol{\beta}$ contains the feature weights. The inclusion of the child profile as a predictor variable enables us to accommodate for a range of different children. In order to determine which version of OAssistMe is best suited for this domain, we consider trial, cost of history and number of repetitions as potential additional predictors, and fit the model to the data using multiple logistic

Table 5 Multiple logistic regression results on the child-robot interaction data. The first element in the $\boldsymbol{\beta}$ vector is the constant term weight, while subsequent elements are predictor weights, in the order mentioned. Single/double stars mean significance to the 0.05/0.001 level. Note that even though the constant term was significant in the JATT but not the NAME task, it was kept for consistency across tasks.

Predictors	JATT	NAME
Profile + Action	$\boldsymbol{\beta} = \begin{bmatrix} 1.30^{*} \\ -1.27^{**} \\ 1.00^{**} \end{bmatrix}, p = \begin{bmatrix} 0.014 \\ < 10^{-3} \\ < 10^{-3} \end{bmatrix}$	$\boldsymbol{\beta} = \begin{bmatrix} 1.88 \\ -1.74^{**} \\ 0.65^{*} \end{bmatrix}, p = \begin{bmatrix} 0.079 \\ < 10^{-3} \\ 0.013 \end{bmatrix}$
Profile + Action + Trial	$\boldsymbol{\beta} = \begin{bmatrix} 1.40^{**} \\ -1.26^{**} \\ 1.09^{**} \\ -0.19 \end{bmatrix}, p = \begin{bmatrix} 0.010 \\ < 10^{-3} \\ < 10^{-3} \\ 0.374 \end{bmatrix}$	$\boldsymbol{\beta} = \begin{bmatrix} 1.93 \\ -1.66^{**} \\ 0.69^{*} \\ -0.18 \end{bmatrix}, p = \begin{bmatrix} 0.073 \\ 0.001 \\ 0.011 \\ 0.546 \end{bmatrix}$
Profile + Action + Cost	$\boldsymbol{\beta} = \begin{bmatrix} 1.25^{*} \\ -1.27^{**} \\ 1.04^{**} \\ -0.03 \end{bmatrix}, p = \begin{bmatrix} 0.023 \\ < 10^{-3} \\ < 10^{-3} \\ 0.728 \end{bmatrix}$	$\boldsymbol{\beta} = \begin{bmatrix} 1.83 \\ -1.71^{**} \\ 0.66^{*} \\ -0.03 \end{bmatrix}, p = \begin{bmatrix} 0.093 \\ 0.001 \\ 0.013 \\ 0.816 \end{bmatrix}$
Profile + Action + Reps	$\boldsymbol{\beta} = \begin{bmatrix} 1.24^{*} \\ -1.25^{**} \\ 1.03^{**} \\ -0.46 \end{bmatrix}, p = \begin{bmatrix} 0.020 \\ < 10^{-3} \\ < 10^{-3} \\ 0.297 \end{bmatrix}$	$\boldsymbol{\beta} = \begin{bmatrix} 1.81 \\ -1.61^{**} \\ 0.61^{*} \\ -0.76 \end{bmatrix}, p = \begin{bmatrix} 0.091 \\ 0.001 \\ 0.019 \\ 0.358 \end{bmatrix}$

regression. Prior to running the regression, the data were checked for potential learning effects across task instances for the same child, but no significant learning effect was found.

Our regression results, summarized in Table 5, show that while action level and child profile are statistically significant predictors, incorporating additional predictors does not significantly improve the model. We conclude that there is no evidence in this particular domain of an effect of history on success probability, at least given the amount of data at hand. Therefore, the basic version of OAssistMe is best suited for this problem. Figure 8 shows visualizations of the regression results with action level and severity as the two predictors. Each data point (blue dots in the two upper plots) represents the average estimated success probability for a given child and action level.

We can see that the NAME task was overall identified to be more difficult since it had lower success probabilities, as well as lower costs (see Table 3). As a result, the total number of observations was higher in the NAME task ($n = 79$) as compared to the JATT task ($n = 50$) because successes occurred less frequently and actual sequences executed by the robot were longer, resulting in a smoother spread in the response variable.

5.3.3 Optimal action sequence results

The results presented above allow us to generate personalized optimal sequences according to the profile of each child. The only remaining parameter to determine is R, which can be tuned. According to the results presented in Section 3.3.4, we should choose $R > \min_a c(a)/p(a)$. Table 6 reports the

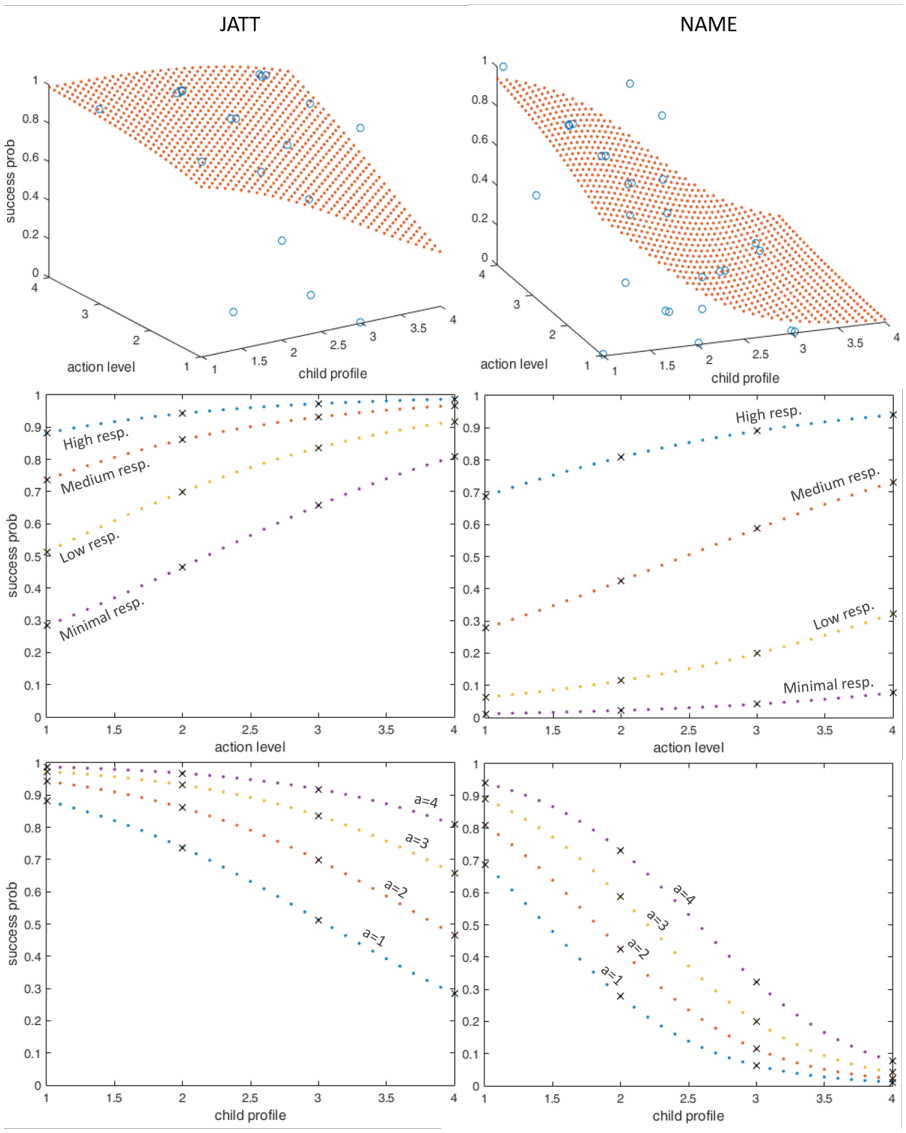

Fig. 8 Success probability results showing data points and fitted surfaces (top), cross-sections relating $p(a)$ to a (mid.) and cross-sections relating $p(a)$ to child profile (bot.). Overlapping data points are perturbed for better visualization. Continuous surfaces and curves are shown for illustration purposes. These results were obtained with MATLAB's GLMFIT function with a logit link function and binomial distribution of response variable, and they slighlty differ from the first row of Table 5 in that they treat data from the same participant as multiple observations from the same binomial distribution.

Table 6 Minimum R values for acceptable algorithm performance for different child profiles on each task.

Task	Child profile	$R_{\min} = \min_a c(a)/p(a)$
JATT	High resp. (1)	65.68
	Medium resp. (2)	70.62
	Low resp. (3)	78.687
	Minimal resp. (4)	92.64
NAME	High resp. (1)	53.21
	Medium resp. (2)	80.81
	Low resp. (3)	225.87
	Minimal resp. (4)	948.24

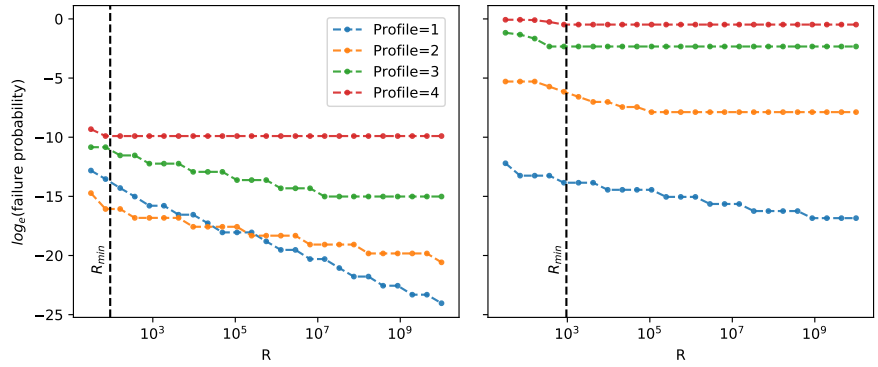

Fig. 9 Effect of R parameter on total probability of failure for task JATT (left) and NAME (right), to inform R value selection. For every point, we first find an optimal action sequence for the corresponding action parameters, and then compute the reported total probability of failure value by multiplying failure probabilities for individual actions.

minimum values of R for the different child profiles. Similar to action costs, R is a parameter intrinsic to the task, so we assume that it does not depend on the child profile, so we want to select a value of R greater or equal to the values reported in the table. For the purposes of this work, we will set R to 950 for both tasks (rounding up the largest value in the table $R_{\min} = 948.24$). In practice, one may want to consider different values of R for different tasks, depending on the relative importance of the skills that the task targets. As mentioned in Section 4.6, the selection of R in practice can also be informed by looking at how it affects the total probability of failure, as shown in Figure 9.

We ran the OAssistMe algorithm with the estimated action parameters, for both JATT and NAME tasks, and report the resulting optimal sequences in Figure 10. As expected, as the child profile increases, the computed sequences have generally higher or equal action levels. As mentioned previously, the NAME task was determined to be more challenging than JATT, which is reflected by the overall higher action levels computed for all child profiles.

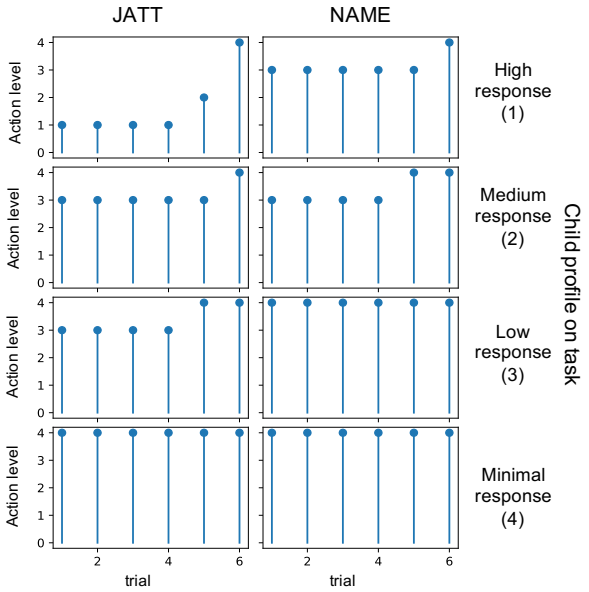

Fig. 10 Sequences generated by the OAssistMe algorithm for $T = 6$ and $R = 950$ for both tasks and the different child profiles. For comparison, running the algorithm with uniformly spaced action costs (12.5, 37.5, 62.5, 87.5) and success probabilities (0.125, 0.375, 0.625, 0.875) yields the sequence $\langle 4, 4, 4, 4, 4 \rangle$.

6 Summary of results and discussion

We start by highlighting the major findings discussed in previous sections, and then discuss some of the limitations of our approach.

The main theoretical results of practical relevance are:

- For high enough R, optimal sequences generated by the OAssistMe algorithm are *nondecreasing*, i.e., the agent should only maintain or increase the action level at the next trial if a failure occurs. This result aligns with typical strategies followed by providers.
- The R parameter affects the *total probability of failure*, hence having a threshold on this probability can inform an appropriate choice on R in practice.

The main observations based on our simulations of the framework extensions are:

- The assumption that success probabilities increase as a function of trial (case TS) seems to generally make optimal action sequences more *conservative*.
- The assumption that success probabilities increase as a function of cost (case CS) of history has a similar effect as the TS case, but allows for *more fine-grained algorithm behavior* by incorporating a cumulative effect of previously executed action levels.

- The assumption that success probabilities decrease as a function of repetitions seems to generally make optimal action sequences *repetition-averse*, hence more *diverse* (higher number of distinct actions).

These general observations are based on our experimentation with realistic parameters. The claims being sensitive to parameter selection, they should only be considered as suggestive results.

Finally, instantiating our framework in a robot-assisted autism therapy scenario leads us to the following observations:

- While child profile and action level were confirmed to be significant predictors of success probability, our data do not show evidence of potential additional history effects such as trial, cost or repetition sensitivity. As a result, the *basic version of OAssistMe* is best suited for generating optimal sequences in this specific context.
- The two tasks considered show *different levels of difficulty* as reflected by differences in both estimated success probabilities and estimated action costs.
- The action sequences generated by OAssistMe with the estimated action parameters show how our framework can achieve *personalization* to accommodate a range of receiver profiles.

Despite our efforts to follow appropriate methodology in evaluating our framework, our approach does not come without some limitations. The results presented in the evaluation section of this work are preliminary and require further testing before they can be used in practice.

First, the action costs were assumed to be identical for all individuals. While this assumption is valid in cases where the cost is purely intrinsic to the action itself — e.g., execution time, financial cost, energy spent —, it becomes fuzzier when the measure of cost possesses some level of subjectivity. In our domain, the variance in the experts' estimated cost values was high, which highlights this subjectivity. In order to reduce the rater subjectivity, in our approach we measured the costs in relation to a virtual reference profile. On the other hand, assuming a profile-dependent cost on top of profile-dependent success probabilities could unnecessarily complicate our model, and may not even be desirable. It is important to note however that our algorithm was able to generate different action sequences for different child profiles, suggesting that assuming constant costs did not compromise flexibility. Moreover, our framework allows for the R parameter to be adjusted on an individual basis if the importance of succeeding on a given task differs according to individual receiver needs.

Second, the survey data collected showed high variance and low reliability, which calls for more reliable methods to estimate action costs in these types of domains. One possibility would be to ensure that the participants have enough understanding and experience with the tasks and scenarios described, and to have baseline questions to test that their understanding of the measure aligns with the researcher's intended meaning. The validity of a general questionnaire

approach to cost estimation could potentially be tested against an inverse reinforcement learning approach where costs are estimated directly from expert demonstrations.

Third, the analysis of the interaction data made an assumption of stationarity across instances of the same task. Even though no main learning effects were found in the data, some subjects did exhibit inconsistent behaviors across instances, such as disengagement, distraction, etc., which may have affected our results. In principle, if one is given a model of evolution of receiver response across several instances, then one can update the action parameters according to that model and run the same algorithms simply with a different input. However, since this work looked at a small number of task instances, it is not concerned with coming up with such models.

Fourth, even though the study presented in Section 5.3 is the first to collect this type of data with children with ASD under careful methodological considerations to reduce bias, it suffers from a low number of samples, as in most probabilistic frameworks. Specifically, because the number of data points for each action level and individual was low, the resulting response variable in our regression model showed a high spread. Higher number of samples per participant may result in better fit of our regression model but may also induce bias in our data due to potential positive or negative learning effects. Furthermore, the samples used for our logistic regression were not from fully independent data, and hence regression results may not be used for principled hypothesis testing purposes. The purpose of the regression in this work was merely to suggest appropriate algorithm selection. All of the questions discussed above should be kept in mind when designing similar data collection scenarios in the future.

7 Conclusion

This work presented a principled approach to address problems related to autonomy and flexibility of assistive agents in a variety of contexts. Specifically, we contributed a mathematical framework to solve for optimal action sequences to be followed by a provider agent in a task with a human receiver, under a set of clearly laid out assumptions.

Throughout the article, we have analyzed the properties of our approach in theory and simulation, and shown preliminary results on the application of our framework to a robot-assisted therapy setting. We first presented an optimal linear-time algorithm based on dynamic programming and proved a number of properties of optimal solutions, including monotonicity, which aligns with typical provider strategies. We then presented and analyzed several extensions of our original framework, incorporating different types of dependency on history, namely trial-sensitivity (TS), cost-sensitivity (CS), and repetition-sensitivity (RS). Finally, we illustrated the potential applicability of our approach in a therapy scenario involving two robot-assisted tasks, on which we gather expert rating data as well as interaction data with 10 children with ASD.

The framework we put forward in this work relies on minimal domain-dependent assumptions. We therefore expect it to have value in general provider-receiver interactions with a similar structure. It may enable agents to play the role of such providers more flexibly, but it could also possibly guide and complement strategies followed by human providers, based on objective data including receiver assessment and past performance.

The natural continuation of this work would be to evaluate the effectiveness of our algorithms in an therapeutic intervention, in our case in the robot-assisted scenario considered in this article. We expect the use of our algorithm to result in more fluid interaction, and possibly better intervention outcomes as compared to baselines such as fixed policies or random action selection. Additional validation of the model put forward in this work may however be needed before testing the algorithm on real user populations. This validation could include more advanced methods, including model simulation and falsification [46,35]. Other directions for future work include the modeling of the receiver as an agent rather than a 'passive' entity. This approach may be particularly relevant to more complex tasks involving decision-making at the receiver level. Additionally, considering more sophisticated measures of success that are multi-attribute may provide a more useful way to incorporate the success of such technologies by taking into account affective aspects, such as lack of engagement, frustration, impatience, etc. Finally, the assumption of full observability may not hold in some cases, and therefore looking at models that incorporate partial observability (such as POMDP's) are worth pursuing.

This research is a first step towards formalizing some of the internalized skills human providers use in their daily professional practice, to enable artificial agents to provide suitable and flexible assistance to humans. We envision a future where providers can work hand in hand with autonomous agents to benefit receivers in healthcare and education. Let's make this vision a reality, one humble step at a time.

Acknowledgements We would like to thank Ana Paiva for her input on the child-robot interaction study. We also thank the reviewers for their valuable suggestions. This research was partially supported by the CMUPERI/HCI/0051/2013 grant, associated with the CMU/Portugal INSIDE project (http://www.project-inside.pt/), as well as national funds through Fundação para a Ciência e a Tecnologia (FCT) with references UID/CEC/50021/2020 and SFRH/BD/128359/2017. The views and conclusions contained in this document are those of the authors only.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Alagoz, O., Hsu, H., Schaefer, A.J., Roberts, M.S.: Markov decision processes: a tool for sequential decision making under uncertainty. Medical Decision Making **30**(4), 474–483

(2010)
2. Anderson, J.R., Boyle, C.F., Reiser, B.J.: Intelligent tutoring systems. Science **228**(4698), 456–462 (1985)
3. Association, A.P., et al.: Diagnostic and statistical manual of mental disorders (DSM-5®). American Psychiatric Pub (2013)
4. Baraka, K., Couto, M., Melo, F.S., Paiva, A., Veloso, M.: An approach for personalized social interactions between a therapeutic robot and children with autism spectrum disorder. Tech. Rep. GAIPS-TR-001-19, Intelligent agents and synthetic characters group (GAIPS), Porto Salvo, Portugal (2019). URL https://gaips.inesc-id.pt/component/gaips/publications/showPublication/3/597
5. Baraka, K., Couto, M., Melo, F.S., Veloso, M.: An optimization approach for structured agent-based provider/receiver tasks. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp. 95–103. International Foundation for Autonomous Agents and Multiagent Systems (2019)
6. Barnes, T., Stamper, J.: Toward automatic hint generation for logic proof tutoring using historical student data. In: International Conference on Intelligent Tutoring Systems, pp. 373–382. Springer (2008)
7. Brunskill, E., Russell, S.: Partially observable sequential decision making for problem selection in an intelligent tutoring system (2011)
8. Chandra, S., Dillenbourg, P., Paiva, A.: Developing learning scenarios to foster children's handwriting skills with the help of social robots. In: Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 337–338. ACM (2017)
9. Clement, B., Roy, D., Oudeyer, P.Y., Lopes, M.: Online optimization of teaching sequences with multi-armed bandits. In: 7th International Conference on Educational Data Mining (2014)
10. Conati, C., Maclaren, H.: Empirically building and evaluating a probabilistic model of user affect. User Modeling and User-Adapted Interaction **19**(3), 267–303 (2009)
11. Conn, K., Liu, C., Sarkar, N., Stone, W., Warren, Z.: Affect-sensitive assistive intervention technologies for children with autism: An individual-specific approach. Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN pp. 442–447 (2008). DOI 10.1109/ROMAN.2008.4600706
12. Esteban, P., Baxter, P., Belpaeme, P., Billing, E., Cai, H., Cao, H., Coeckelbergh, M., Costescu, C., David, D., Beir, A.D., Fang, Y., Ju, Z., Kennedy, J., Liu, H., Mazel, A., Pandey, A., Richardson, K., Senft, E., Thill, S., Van de Perre, G., Vanderborght, B., Vernon, D., Yu, H., Ziemke, T.: How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder. Paladyn J. Behavioral Robotics **8**, 18–38 (2017)
13. Feil-Seifer, D., Matarić, M.J.: Socially assistive robotics. IEEE Robotics & Automation Magazine **18**(1), 24–31 (2011)
14. Folsom-Kovarik, J.T., Sukthankar, G., Schatz, S.: Tractable pomdp representations for intelligent tutoring systems. ACM Transactions on Intelligent Systems and Technology (TIST) **4**(2), 1–22 (2013)
15. Frank Lopresti, E., Mihailidis, A., Kirsch, N.: Assistive technology for cognitive rehabilitation: State of the art. Neuropsychological rehabilitation **14**(1-2), 5–39 (2004)
16. Gibbons, P.: Scaffolding language, scaffolding learning. Portsmouth, NH: Heinemann (2002)
17. Grover, S., Chakraborti, T., Kambhampati, S.: What can automated planning do for intelligent tutoring systems? ICAPS SPARK (2018)
18. Hauskrecht, M., Fraser, H.: Planning treatment of ischemic heart disease with partially observable markov decision processes. Artificial Intelligence in Medicine **18**(3), 221–244 (2000)
19. Head, H.: Aphasia and kindred disorders of speech, vol. 2. Cambridge University Press (2014)
20. Hersch, G.I., Lamport, N.K., Coffey, M.S.: Activity analysis: Application to occupation. SLACK Incorporated (2005)
21. Hoey, J., Boutilier, C., Poupart, P., Olivier, P., Monk, A., Mihailidis, A.: People, sensors, decisions: Customizable and adaptive technologies for assistance in healthcare. ACM Transactions on Interactive Intelligent Systems (TiiS) **2**(4), 1–36 (2013)

22. Horvitz, E.: Agents with beliefs: Reflections on bayesian methods for user modeling. In: User Modeling, pp. 441–442. Springer (1997)
23. Itti, L., Baldi, P.F.: Bayesian surprise attracts human attention. In: Advances in neural information processing systems, pp. 547–554 (2006)
24. Kenny, P., Parsons, T., Gratch, J., Rizzo, A.: Virtual humans for assisted health care. In: Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments, p. 6. ACM (2008)
25. Kim, G., Lim, S., Kim, H., Lee, B., Seo, S., Cho, K., Lee, W.: Is robot-assisted therapy effective in upper extremity recovery in early stage stroke?a systematic literature review. Journal of physical therapy science **29**(6), 1108–1112 (2017)
26. Lauren Klein Laurent Itti, B.A.S.M.R.S.N., Matarić, M.J.: Surprise! predicting infant visual attention in a socially assistive robot contingent learning paradigm. In: 2019 IEEE International Symposium on Robot and Human Interactive Communication (2019). URL http://robotics.usc.edu/publications/1057/
27. Lawson, R.P., Rees, G., Friston, K.J.: An aberrant precision account of autism. Frontiers in human neuroscience **8**, 302 (2014)
28. Leite, I.: Long-term interactions with empathic social robots. AI Matters **1**(3), 13–15 (2015)
29. Linebaugh, C.W., Lehner, L.H.: Cueing hierarchies and word retrieval: A therapy program. In: Clinical Aphasiology: Proceedings of the Conference 1977, pp. 19–31. BRK Publishers (1977)
30. Lord, C., Rutter, M., Dilavore, P., Risi, S., Gotham, K., Bishop, S.: Autism diagnostic observation schedule, second edition. Western Psychological Services, CA (2012)
31. Luckin, R., Koedinger, K.R., Greer, J.: Artificial intelligence in education: building technology rich learning contexts that work, vol. 158. IOS Press (2007)
32. Murray, R.C., VanLehn, K.: A comparison of decision-theoretic, fixed-policy and random tutorial action selection. In: International Conference on Intelligent Tutoring Systems, pp. 114–123. Springer (2006)
33. Nikolaidis, S., Zhu, Y.X., Hsu, D., Srinivasa, S.: Human-robot mutual adaptation in shared autonomy. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 294–302. ACM (2017)
34. Palestra, G., Varni, G., Chetouani, M., Esposito, F.: A multimodal and multilevel system for robotics treatment of autism in children. Proceedings of the International Workshop on Social Learning and Multimodal Interaction for Designing Artificial Agents - DAA '16 pp. 1–6 (2016). DOI 10.1145/3005338.3005341. URL http://dl.acm.org/citation.cfm?doid=3005338.3005341
35. Palminteri, S., Wyart, V., Koechlin, E.: The importance of falsification in computational cognitive modeling. Trends in cognitive sciences **21**(6), 425–433 (2017)
36. Petric, F., Miklic, D., Kovacic, Z.: Robot-assisted autism spectrum disorder diagnostics using pomdps. In: Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 369–370 (2017)
37. Rivers, K., Koedinger, K.R.: Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. International Journal of Artificial Intelligence in Education **27**(1), 37–64 (2017)
38. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, (2016)
39. Scassellati, B., Admoni, H., Matarić, M.: Robots for use in autism research. Annual review of biomedical engineering **14**, 275–294 (2012)
40. Schaaf, R.C., Roley, S.S.: Sensory integration: Applying clinical reasoning to practice with diverse populations. PRO-ED, Incorporated (2006)
41. Schaefer, A.J., Bailey, M.D., Shechter, S.M., Roberts, M.S.: Modeling medical treatment using markov decision processes. In: Operations research and health care, pp. 593–612. Springer (2005)
42. Schwartenbeck, P., Friston, K.: Computational phenotyping in psychiatry: A worked example. Eneuro **3**(4) (2016)
43. Short, E., Swift-Spong, K., Greczek, J., Ramachandran, A., Litoiu, A., Grigore, E.C., Feil-Seifer, D., Shuster, S., Lee, J.J., Huang, S., et al.: How to train your dragonbot: Socially assistive robots for teaching children about nutrition through play. In: The

23rd IEEE international symposium on robot and human interactive communication, pp. 924–929. IEEE (2014)
44. Van Vuuren, S., Cherney, L.R.: A virtual therapist for speech and language therapy. In: International Conference on Intelligent Virtual Agents, pp. 438–448. Springer (2014)
45. Warren, Z.E., Zheng, Z., Swanson, A.R., Bekele, E.T., Zhang, L., Crittendon, J.A., Weitlauf, A.F., Sarkar, N.: Can Robotic Interaction Improve Joint Attention Skills? Journal of Autism and Developmental Disorders **45**(11) (2015). DOI 10.1007/s10803-013-1918-4
46. Wilson, R.C., Collins, A.G.: Ten simple rules for the computational modeling of behavioral data. eLife **8**, e49547 (2019)
47. You, Z.J., Shen, C.Y., Chang, C.W., Liu, B.J., Chen, G.D.: A robot as a teaching assistant in an english class. In: Advanced Learning Technologies, 2006. Sixth International Conference on, pp. 87–91. IEEE (2006)
48. Zhang, Y., Steimle, L., Denton, B.: Robust markov decision processes for medical treatment decisions. Optimization online (2017)

Appendix: Proofs

Our proofs are structured along the following three (mutually exclusive) cases:

a. $O_1^* > 0$, or equivalently $R < \min_a c(a)/p(a)$
b. $O_1^* < 0$, or equivalently $R > \min_a c(a)/p(a)$
c. $O_1^* = 0$, or equivalently $R = \min_a c(a)/p(a)$

Lemma 1: For any T, we have one of:

a. $0 < O_T^* < \min_a c(a)/p(a) - R$
b. $0 > O_T^* > \min_a c(a)/p(a) - R$
c. $0 = O_T^* = \min_a c(a)/p(a) - R$

Proof We use induction on T. From Equation (7):
$O_T^* = \min_a \{(1 - p(a))O_{T-1}^* + c(a) - p(a)R\}, \quad O_1^* = \min_a \{c(a) - p(a)R\}$
applies in all cases.

Case (a):
Base case: $0 < O_1^* = \min_a c(a) - p(a)R < \min_a c(a)/p(a) - R$
Induction step: Assume $0 < O_{T-1}^* < \min_a c(a)/p(a) - R$, then O_T^* is also positive from Equation (7) and base case. Also, for all a:

$$\begin{aligned} O_T^* &\leq (1 - p(a))O_{T-1}^* + c(a) - p(a)R \\ &< (1 - p(a))(c(a)/p(a) - R) + c(a) - p(a)R = c(a)/p(a) - R \end{aligned}$$

By induction, $0 < O_T^* < \min_a c(a)/p(a) - R$ for all T.

Case (b):
Base case: $0 > O_1^* = \min_a c(a) - p(a)R > \min_a c(a)/p(a) - R$
Induction step: Assume $0 > O_{T-1}^* > \min_a c(a)/p(a) - R$. Also let $a^\dagger = \arg\min_a c(a)/p(a)$, let a^* be the optimal action selected at stage T, and let $a^{*(1)}$ be the optimal action selected at stage 1.

$$O_T^* \leq (1 - p(a^{*(1)}))O_{T-1}^* + c(a^{*(1)}) - p(a^{*(1)})R < 0$$

since $(1 - p(a))O_{T-1}^* < 0$ for any a, and $c(a^{*(1)}) - p(a^{*(1)})R < 0$ (base case).
Also, for all a:

$$\begin{aligned} O_T^* &= (1 - p(a^*))O_{T-1}^* + c(a^*) - p(a^*)R \\ &> (1 - p(a^*))(c(a^\dagger)/p(a^\dagger) - R) + c(a^*) - p(a^*)R \\ &= (1 - p(a^*))c(a^\dagger)/p(a^\dagger) + c(a^*) - R \end{aligned}$$

Using $p(a^*) < p(a^\dagger)c(a^*)/c(a^\dagger)$:
$O_T^* > \left[1 - p(a^\dagger)c(a^*)/c(a^\dagger)\right] c(a^\dagger)/p(a^\dagger) + c(a^*) - R = c(a^\dagger)/p(a^\dagger) - R$
By induction, $0 > O_T^* > \min_a c(a)/p(a) - R$ for all T.

Case (c) is easily proven by induction on T. □

Lemma 2: O_T^* is monotonic in T. In particular, it is one of:
a. strictly increasing, i.e., $O_{T+1}^* > O_T^*$ for all T
b. strictly decreasing, i.e., $O_{T+1}^* < O_T^*$ for all T
c. constant, i.e., $O_{T+1}^* = O_T^*$ for all T

Proof Let a^* be the optimal action of stage T.
Case (a):
$O_T^*/O_{T-1}^* = 1 - p(a^*) + (c(a^*) - p(a^*)R)/O_{T-1}^*$
From Lemma 1, for any a:
$0 < O_{T-1}^* < c(a)/p(a) - R$, so $(c(a^*) - p(a^*)R)/O_{T-1}^* > p(a^*)$, hence:
$O_T^*/O_{T-1}^* > 1$, and $O_T^* > 0$ for all T, which establishes that O_T^* is strictly increasing.

Case (b):
The demonstration that $O_T^*/O_{T-1}^* > 1$ is identical to case (a). Given that $O_T^* < 0$ for all T, then O_T^* is strictly decreasing.

Case (c) follows form the previous lemma. □

Theorem 1: O_T^* **converges** to $\min_a c(a)/p(a) - R$ as T goes to infinity.

Proof Lemmas 1 and 2 imply convergence of O_T^* in cases (a) and (b). Furthermore, setting O_{T-1} to O_T in Equation (7) results in a single fixed point $\min_a c(a)/p(a) - R$, which establishes the result.
Case (c) is trivial since $\min_a c(a)/p(a) - R = 0$. □

Theorem 2: If $\boldsymbol{\Pi}^*$ is an optimal sequence, then it is monotonic in t. In particular, $\boldsymbol{\Pi}^*$ is one of:
a. **nonincreasing**, i.e., $a_1^* \geq a_2^* \geq ... \geq a_T^*$
b. **nondecreasing**, i.e., $a_1^* \leq a_2^* \leq ... \leq a_T^*$
c. **constant**, i.e., $a_1^* = a_2^* = ... = a_T^*$

Proof Let a' be an optimal action associated with O_{T-1}^* and a'' an optimal action associated with O_T^*. Then:
$(1 - p(a''))O_T^* + c(a'') - p(a'')R \leq (1 - p(a'))O_T^* + c(a') - p(a')R$

$$(p(a') - p(a''))O_T^* \leq c(a') - c(a'') - R(p(a') - p(a'')) \tag{15}$$

and
$(1 - p(a'))O_{T-1}^* + c(a') - p(a')R \leq (1 - p(a''))O_{T-1}^* + c(a'') - p(a'')R$

$$(p(a') - p(a''))O_{T-1}^* \geq c(a') - c(a'') - R(p(a') - p(a'')) \tag{16}$$

Combining Equations (15) and (16), we get:

$$(p(a') - p(a''))O_T^* \leq (p(a') - p(a''))O_{T-1}^*$$

We can conclude that:
If $p(a') > p(a'')$: $O_T^* \leq O_{T-1}^*$ and if $p(a') < p(a'')$: $O_T^* \geq O_{T-1}^*$

Assume $a' > a''$. Then $p(a') > p(a'')$. From the previous result: $O_T^* \leq O_{T-1}^*$, which contradicts Lemma 2. Hence, $a' \leq a''$, which establishes that $\boldsymbol{\Pi}^*$ is nonincreasing.
Similarly, we can show that, in **case (b)**, $\boldsymbol{\Pi}^*$ is nondecreasing.
In **case (c)**, every step is equivalent to the single trial case, and the same action is selected at every trial, so the resulting sequence is constant. □