Ad hoc teamwork by learning teammates' task

Francisco S. Melo · Alberto Sardinha

Abstract This paper addresses the problem of ad hoc teamwork, where a learning agent engages in a cooperative task with other (unknown) agents. The agent must effectively coordinate with the other agents towards completion of the intended task, not relying on any pre-defined coordination strategy. We contribute a new perspective on the ad hoc teamwork problem and propose that, in general, the learning agent should not only identify (and coordinate with) the teammates' strategy but also identify the task to be completed. In our approach to the ad hoc teamwork problem, we represent tasks as fully cooperative matrix games. Relying exclusively on observations of the behavior of the teammates, the learning agent must identify the task at hand (namely, the corresponding payoff function) from a set of possible tasks and adapt to the teammates' behavior. Teammates are assumed to follow a bounded-rationality best-response model and thus also adapt their behavior to that of the learning agent. We formalize the ad hoc teamwork problem as a sequential decision problem and propose two novel approaches to address it. In particular, we propose (i) the use of an online learning approach that considers the different tasks depending on their ability to predict the behavior of the teammate; and (ii) a decision-theoretic approach that models the ad hoc teamwork problem as a partially observable Markov decision problem. We provide theoretical bounds of the performance of both approaches and evaluate their performance in several domains of different complexity.

Keywords Ad hoc teamwork · Online learning · POMDP

Electronic supplementary material The online version of this article (doi:10.1007/s10458-015-9280-x) contains supplementary material, which is available to authorized users.

F. S. Melo (✉)
INESC-ID and Instituto Superior Técnico, Universidade de Lisboa, IST Taguspark,
Av. Prof. Dr. Cavaco Silva, Office 2N7.15, 2744-016 Porto Salvo, Portugal
e-mail: fmelo@inesc-id.pt

A. Sardinha
INESC-ID and Instituto Superior Técnico, Universidade de Lisboa, IST Taguspark,
Av. Prof. Dr. Cavaco Silva, Office 2N9.13, 2744-016 Porto Salvo, Portugal
e-mail: jose.alberto.sardinha@tecnico.ulisboa.pt

1 Introduction

Recent years have witnessed a generalization of “smart systems”, i.e., systems relying on some form of intelligent agent technology, which are able to autonomously collect information and act upon it. Examples include *smart grids* that autonomously gather information and act upon it to improve the efficiency and reliability of electricity production and distribution [50] and *personal assistants* in smartphones that are able to keep track of users’ calendars and preferences in order to make recommendations and assist the users in several tasks [67].

As technology evolves, so will the autonomy and perceptual/actuation capabilities of such agents, prompting the need for truly autonomous agents that are able to co-exist with other (different) agents and eventually engage in some form of *teamwork* towards the completion of some common task. However, due to the differences between the agents in terms of origin, access to information, and perceptual and actuation capabilities, such teamwork must take place without any prior coordination protocol or even, in some cases, any form of explicit communication. The challenge of developing autonomous agents that are capable of cooperatively engaging with other (unknown) agents in a joint common task is known as the *ad hoc teamwork problem* [60]. It is a new and exciting research area, recently introduced in the pioneer work of Stone et al. [60], and is closely related to other areas in the literature on autonomous agents and multiagent systems (see [60] for a detailed discussion).

In this paper, we break down the ad hoc teamwork problem into different challenges that an agent must address when deployed within an ad hoc team. We claim that the agent may not always know beforehand the task that it is expected to complete. Therefore, it must both identify the target task as well as the strategy adopted by the teammates in order to tackle it.

Consider, for example, an *e*-commerce scenario where two autonomous agents cooperate to procure and assemble the components necessary to put together a personal computer (PC) package, namely, an LCD monitor and a desktop computer. Both agents are able to procure any of the necessary components needed for the PC package, although one agent has been optimized to assemble the LCD monitor (for which it must procure an LCD panel), while the other has been optimized to assemble the desktop computer (for which it must procure the motherboard). In order to maximize the profit, it is desirable that the agents coordinate their purchase of the corresponding parts so that both parts come from the same supplier and the shipping costs are optimized. Failure of any of the two agents will be costly because it significantly impacts the production rate.

In the above scenario, an “ad hoc agent” could automatically replace any of the two agents in case of a detected failure.[1] However, when deployed, the ad hoc agent may not know the exact target task (i.e., whether it must replace the agent optimized for desktop PC production or replace the agent optimized for LCD monitor production). Additionally, once it determines its target task by observing (and co-acting with) the teammate agent, it must coordinate its action selection with that of the teammate to ensure, for example, that both are purchasing the corresponding items from the same supplier.

A similar situation can also arise in other domains [60]. For example, Bowling and McCracken [11] provide an early example of ad hoc teamwork in robot soccer. This work introduces the notion of an *impromptu team*, where an agent (referred to as a *pickup player*) is teamed up with a group of unknown agents with which it must coordinate play execution. This early work already examined some of the challenges central to what is now referred as the ad hoc teamwork problem, namely, inferring the unknown strategy adopted by the team-

[1] In order to facilitate the presentation, we adopt a commonly used abuse of language in the ad hoc teamwork literature and refer to an “ad hoc agent” as being an agent that is able to engage in ad hoc teamwork.

mates from the actions of the latter. This work, like many of the existing studies in ad hoc teamwork, assumes that agents are aware of the target task [2,13,61] and does not consider the problem of identifying it from the teammate's behavior.

The key contributions of this work are thus fourfold. First, we present in Sect. 2 a new perspective on the ad hoc teamwork problem. Specifically, we propose that, when addressing ad hoc teamwork, one should explicitly reason in terms of *task identification*, *teammate identification* and *planning*.

Second, we formalize the ad hoc teamwork problem as a *sequential decision problem*, in which the ad hoc agent does not know in advance the task to be performed or how its teammates act towards performing it. In our formalization, we represent tasks as fully cooperative matrix games; hence, the ad hoc agent must identify the task at hand (namely, the corresponding payoff function) from a set of possible tasks. Teammates are assumed to follow a bounded-rationality best-response model and thus adapt their behavior to that of the learning agent.

Third, we propose two novel approaches to the ad hoc teamwork problem. In particular, we propose (i) the use of an online learning approach that is capable of detecting the task being performed by the teammates and acting accordingly; and (ii) a decision-theoretic approach that models the ad hoc teamwork problem as a partially observable Markov decision problem. We present theoretical bounds concerning the performance of both approaches and discuss the implications.

Finally, we illustrate the application of our approaches in three different sets of experiments. The first experiment evaluates the performance of both approaches in a small *e*-commerce scenario, which is also used as a running example throughout the paper. The second experiment evaluates the performance of both approaches in random domains of increasing complexity. The third experiment evaluates the performance of our ad hoc agents in a benchmark domain from the literature [59] and compares some of our results with related works in the ad hoc teamwork literature.

2 The ad hoc teamwork problem

The ad hoc teamwork setting [60] is a research problem in which an autonomous agent must efficiently and robustly collaborate with previously unknown teammates on a common task. Therefore, agent developers cannot design an *a priori* coordination strategy for the team of agents. Such an agent must be designed with interaction capabilities that can adjust to different types of teammates.

In this section, we introduce a running example that motivates a novel perspective on the ad hoc teamwork problem. We identify the different challenges that an agent must address in order to be deployed within an ad hoc team. Finally, we discuss the existing literature on ad hoc teamwork and related topics, framing our contributions in the context of the current research.

2.1 A novel perspective on ad hoc teamwork

Before discussing the ad hoc teamwork problem as introduced by Stone et al. [60], we introduce the following example, which is an expanded version of the scenario discussed in the Sect. 1.

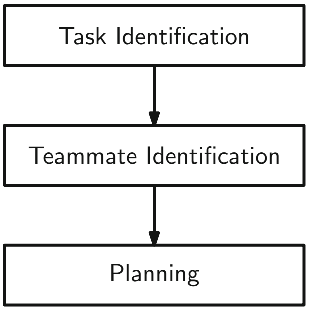


Fig. 1 Challenges in establishing ad hoc teamwork

Example 1 Consider an *e*-commerce company in which two autonomous agents are responsible for assembling PC desktop packages. Each package is composed of an LCD monitor and a desktop computer.

To assemble the LCD monitor, an LCD panel must be purchased. Similarly, the assembly of the desktop computer requires a motherboard that must be purchased. Both agents are able to assemble any of the package elements; however, each agent is optimized to assemble one of the two and will be less efficient in assembling the other.

In order to maximize the profit, it is desirable that:

– The agents coordinate the purchasing process so that both parts are procured from the same supplier in order to minimize shipping costs.
– The agents coordinate the purchasing process so that each agent purchases the part required for the component that it has been optimized to assemble.

Recently, the company has experienced increasing delays in production because the agents have started to fail, which requires a costly and time-consuming restart procedure. The company is not willing to discard the existing agents, as re-engineering the complete production system would be a costly operation. Therefore, to minimize the impact of agent failure, the company adopted an automated system that automatically deploys a general purpose agent whenever an agent failure is detected.

Let us consider the requirements that the general purpose agent from Example 1 must meet. First of all, it must quickly determine which of the legacy agents it has to replace. Then, once its task is identified, it must determine exactly how the other agent (the teammate) is addressing that task so that it can coordinate its actions with those of the teammate. These three requirements (identifying the task, analyzing the teammate behavior and planning its own behavior) are common to any autonomous agent that is expected to be deployed in a given domain and effectively coordinate with existing agents in a common task.[2]

The aforementioned requirements and their interdependence, depicted in Fig. 1 as *task identification*, *teammate identification* and *planning*, are the fundamental axes around which the ad hoc teamwork problem should be formulated. Although most existing work on ad hoc teamwork has not considered the task identification problem, we argue that this is an important challenge that should not be overlooked and is already implicit in the original description of the ad hoc teamwork problem [60].

Considering the ad hoc teamwork problem from the 3-axis perspective above, one can identify several closely related problems in the literature that address each of the challenges in

[2] In the nomenclature of Fig. 1, we take *planning* in its broadest sense, which includes any offline and/or online reasoning about which actions to take towards the agent's goal and is a process tightly coupled with *acting*.

Fig. 1 separately. For example, it is possible to identify *task identification* with the literature on plan recognition [37], inverse reinforcement learning [1] and related topics, where an agent must identify a target task from the observed behavior of other agent(s). Similarly, it is possible to identify *teammate identification* with the large body of work on learning in games [20] and opponent modeling [21,48], where an agent must predict the behavior of other agents from observing their actions. Finally, *planning* is clearly related with the growing body of work on decentralized/distributed planning [27,52]. We refer to the work of Stone et al. [60] for a detailed discussion on the relation between ad hoc teamwork and other related areas of research.

We conclude by noting that, in many actual problems, it may be difficult to draw a clear separation between the three challenges identified above or address them independently. In fact, it is possible that, in many scenarios, these challenges are actually tackled simultaneously. For example, task identification will typically require some sort of model for the teammates (from the teammate identification step) that enables the ad hoc agent to make sense of their actions in terms of the task.

2.2 Related work

To the best of our knowledge, most research on ad hoc teamwork is focused on the planning step, relying on different assumptions that simplify the other challenges associated with it. For instance, Stone and Kraus [58] propose one of the first algorithms for ad hoc teamwork. The authors formulate the ad hoc teamwork problem using the formalism of multi-armed bandits, in order to maximize the expected sum of payoffs. The work considers a scenario with two agents (a teacher and a learner agent) that have a task of collecting discarded aluminum cans from three ocean beaches for recycling. This precursor work shares with our own approach the use of an online learning approach for action selection. The main difference in our approach is the explicit reasoning about task identification and the consideration of an adaptive teammate model.

In a different work, Stone et al. [61] describe an algorithm for an ad hoc agent to lead a best response agent to perform actions that yield optimal joint utility. Agmon and Stone [2] extend the work of Stone et al. [61] to the more general problem of leading N-agent teams, adopting a game-theoretic analysis; however, instead of leading the team to the optimal joint utility, the ad hoc agent leads the team to the optimal reachable joint action. Genter et al. [23] are concerned with how ad hoc agents can lead a flock of agents to a desired orientation and propose an initial theoretical and empirical analysis of the problem. In our work, we also assume teammates to behave according to a bounded-rationality best-response model. However, these works focus on tasks where the ad hoc agent must *lead* the other agents and are thus concerned with the planning step of the ad hoc problem. Our work introduces the notion of task identification and teammate identification as important steps so that the planning step can select the action of the ad hoc agent accordingly.

Barrett and Stone [5] present a framework for analyzing ad hoc team problems by defining three dimensions along which to conduct such analysis. First, *team knowledge* is related to the knowledge that the ad hoc agent has about his teammates' behaviors. Second, *environment knowledge* is another dimension of how much knowledge the ad hoc agent has regarding its environment. Lastly, *reactivity of teammates* is concerned with how much its teammates actions can affect the ad hoc agent's decision. This vision is closely aligned with our own perspective, depicted in Fig. 1. In fact, lack of team knowledge results in the need for team identification; task identification can be cast as lack of environment knowledge. Moreover, our paper considers teammates whose behavior adapts (in a best-response sense) to the behavior of the ad hoc agent, which is closely related with agent reactivity. Our paper goes beyond this analysis and proposes two novel approaches to address the ad hoc teamwork problem.

Genter et al. [22] propose a role-based approach for ad hoc teams, where an agent must identify a role, within a set of possible roles, that yields optimal utility for the team. The work also shows that the approach has a predictive nature, whereby the method can be trained for a task and used successfully for another task. In a related work, Liemhetcharat and Veloso [41] introduce the notion of weighted synergy graph for role assignment, which enables a set of agents (represented as vertices in the graph) to reason about team formation. Given a set of agents and a task for which different roles are necessary, the synergy graph is used to approximate the optimal role assignment for the team. One key difference between the two aforementioned works and our own lies on the fact that the former assume that the task is known a priori, while our work assumes that the agent does not know the task beforehand. Interestingly, it is possible to look at the problem of team formation of Liemhetcharat and Veloso [41] as preceding the ad hoc teamwork problem addressed in our paper: while the former focuses on selecting the teammates with which to work, the latter focuses on actually working with those teammates.

Chakraborty and Stone [13] present a learning algorithm for ad hoc agents and a theoretical analysis of the planning step that yields optimal cooperation when the teammate is assumed to be Markovian. A teammate behavior is assumed to be Markovian when the policy is a function of a set of features, where each feature is a statistic computed from the joint action history and the dynamics of the interaction can be modeled as a Markov Decision Process (MDP). Like the aforementioned work, we also rely on a specific type of Markovian teammate model (i.e., a history-based best-response model).

In order to empirically evaluate ad hoc teams, Barrett and Stone [4] present a study of several ad hoc teamwork strategies in a more open and complex teamwork domain, the pursuit domain, which has been frequently used in the literature on multiagent systems [59]. In a subsequent work [7], the same authors extend the previous ad hoc teamwork strategies to make use of a library of learned teammates' models and use it for the planning step. This approach is closely related to our online learning approach, although, as with other works, the authors address only the teammate identification and planning steps, assuming knowledge of the target task.

Game theory provides a strong theoretical framework to analyze multiagent interaction [40] and results from iterated play in normal form games that have been adapted to address ad hoc teamwork. For instance, Wu et al. [66] propose an online planning algorithm, where the planning problem is approximated by solving a series of stage games. Albrecht and Ramamoorthy [3] formulate the coordination problem in ad hoc teams as a stochastic Bayesian game and derive a best response rule. Our work is also based on a game-theoretic framework: tasks are modeled as K-player fully cooperative games, and teammate identification is performed using *fictitious play*.[3] Additionally, the theoretical analysis of our decision-theoretic approach parallels existing work on Bayesian learning in games [20].

In summary, ad hoc teamwork is a new and exciting research area that already features a rich body of literature, although most existing work has focused on relevant subproblems of the more general formulation summarized in Fig. 1. In particular, most existing approaches assume that the target task is known in advance and focus either on teammate identification or planning. In this paper, the target task is assumed unknown, and the ad hoc agent must identify the target task among a set of possible tasks, in addition to coordinating with the teammates

[3] Fictitious play is a common approach used for learning in games [20]. In the classical fictitious play setting, agents play best response strategies against fixed strategy opponents.

towards completion of the target task. To this purpose, we rely on a bounded-memory best-response model for the teammates so that the teammates' actions can be interpreted in terms of possible target tasks. We propose two novel approaches to the ad hoc teamwork problem, the first of which uses an online learning approach and the second of which relies on a decision-theoretic formulation. We provide theoretical guarantees for the performance of both methods and discuss the underlying assumptions and implications.

3 Tackling the ad hoc teamwork problem

This section sets up the notation used throughout the paper, formalizes the problem and introduces the main technical contributions of the paper.

3.1 Notation and nomenclature

We henceforth represent a *K-player fully cooperative matrix game* as a tuple $\Gamma = (K, (\mathcal{A}_k), U)$, where $\mathcal{A}_k$, $k = 1, \ldots, K$, denotes the set of *pure strategies* available to player k and U denotes the *payoff matrix* associated with the game. $U(a)$ thus denotes the payoff received by *all agents* for executing the pure strategy $a \in \mathcal{A}$, where $\mathcal{A} = \times_{k=1}^{K} \mathcal{A}_k$.

In spite of the game-theoretic framework, we follow the nomenclature of Stone et al. [60] and refer to players as *agents* and to pure strategies as *actions*. $\mathcal{A}_k$ thus represents the action-space of agent k, i.e., the set of *individual actions* available to agent k at each time-step. Similarly, $\mathcal{A}$ is the set of *joint actions*, and we write $a = \langle a_1, \ldots, a_K \rangle$ to explicitly indicate that the joint action a results from the individual actions $a_1, \ldots, a_K$.

We refer to a mapping $\pi_k : \mathcal{A}_k \to [0, 1]$, such that $\sum_{a_k \in \mathcal{A}_k} \pi_k(a_k) = 1$, as an *individual policy* for agent k. Intuitively, $\pi_k(a_k)$ represents the probability of agent k executing its individual action a_k. A *joint policy* π can be obtained from individual policies $\pi_1, \ldots, \pi_K$ as

$$\pi(a) = \prod_{k=1}^{K} \pi_k(a_k)$$

where, as before, $a = \langle a_1, \ldots, a_K \rangle$. As with actions, we write $\pi = \langle \pi_1, \ldots, \pi_K \rangle$ to explicitly indicate that the joint policy π results from the individual policies $\pi_1, \ldots, \pi_K$. We denote by a_{-k} and π_{-k} a *reduced joint action* and a *reduced joint policy*, respectively, defined as

$$\begin{aligned} a_{-k} &= \langle a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_K \rangle, \\ \pi_{-k} &= \langle \pi_1, \ldots, \pi_{k-1}, \pi_{k+1}, \ldots, \pi_K \rangle. \end{aligned}$$

3.2 Problem formulation

Consider a known set of tasks $\mathcal{T}$, each described as a fully cooperative matrix game $\Gamma_\tau = (K, (\mathcal{A}_k), U_\tau)$, $\tau \in \mathcal{T}$. In each instance of the ad hoc teamwork problem, a target task $T^* \in \mathcal{T}$ is selected randomly according to some known probability distribution p_0, with

$$p_0(\tau) = \mathbb{P}\left[T^* = \tau\right], \qquad \tau \in \mathcal{T}.$$

All agents repeatedly engage in the corresponding game, Γ_{T^*}, selecting, at each time-step n, an action $A_k(n) \in \mathcal{A}_k$ according to some individual policy π_k. The individual actions are selected *simultaneously* and *without communication*, implying that each agent does not know beforehand the actions that the other agents will select.

In general, the individual action selected by an agent k may depend, at each time-step $n+1$, on the *history* up to that time-step, denoted as the random variable (r.v.) $H(n)$:

$$H(n) = \{A(t), t = 1, \ldots, n\}$$

where $A(t)$ is the r.v. corresponding to the joint action of all agents at time-step t. Formally, if $h_{1:n}$ denotes a particular instance of $H(n)$, we have

$$\pi_k(h_{1:n}, a_k) \triangleq \mathbb{P}\left[A_k(n+1) = a_k \mid H(n) = h_{1:n}\right].$$

The *ad hoc teamwork problem* can now be formalized as a sequential decision problem in which one agent (the *ad hoc agent*) is paired with a team of $K-1$ other agents to cooperatively perform the target task, T^*. To simplify our presentation, we henceforth refer to the ad hoc agent as agent α and interpret the remaining $K-1$ agents as a single "meta-agent", denoted as agent $-\alpha$, which is aware of the target task, T^*. Agent α does not know in advance the task to be performed or how its teammates act towards performing it and must, therefore, do the following:

- determine, by observing the actions of the other agents, the *task* to be performed (*task identification* in Fig. 1);
- determine, again by observing the actions of the other agents, the *strategy* of its teammates (*teammate identification* in Fig. 1);
- act accordingly (*planning* in Fig. 1).

It is worth emphasizing that, although the tasks in $\mathcal{T}$ are represented as fully cooperative matrix games, *this representation is internal to the ad hoc agent*. In particular, the teammate agents do not necessarily have the same internal representation of the task—or any explicit representation at all. The particular process by which the teammates select their action is unknown to the ad hoc agent; this lack of knowledge is the core of the ad hoc teamwork problem. It follows that the agents do not receive any actual payoff from the environment, and payoff information cannot be used by the ad hoc agent for the purpose of learning.

To some extent, our setting is close to that of *inverse reinforcement learning* (IRL), where an agent must identify a target task from the observation of the actions of another agent (the *expert*) [45]. In IRL, the learning agent is provided with a set of reward functions that represent potential tasks, and the agent must then determine which reward function better explains the behavior of the expert without receiving any additional feedback (such as reward) from the environment. The main difference from IRL and the ad hoc setting considered herein is that, in our case, the learning agent is "co-acting" with the expert (the teammate), and the actions of the latter depend on the actions of the former, in light of the dependence of $\pi_{-\alpha}$ on the history.

In this sense, our work is also closely related with *reinforcement learning* (RL), where an agent must learn an optimal line of action by interacting with its environment and other existing agents [42,62]. However, unlike the standard RL setting, the ad hoc agent α receives no evaluative feedback from the environment.

We now revisit Example 1 and illustrate how the *e*-commerce scenario can be modeled using the above formalism.

Example 1 (cont.) To formulate the *e*-commerce scenario as an ad hoc teamwork problem, we consider that both parts used to assemble the PC package can be purchased from one of two suppliers: Supplier A and Supplier B. The price of each part and the shipping costs are different for each supplier and are summarized in Table 1. For every purchase, each agent must pay the shipping costs, except if both agents order from the same suppliers, in

Table 1 Price and shipping cost of different parts

	LCD panel price	Motherboard price	Shipping cost
Supplier A	\$10	\$7	\$2
Supplier B	\$7	\$7	\$5

	A, LCD	B, LCD	A, Motherboard	B, Motherboard
A, LCD	−22	−24	6	1
B, LCD	−24	−19	4	6
A, Motherboard	4	2	−16	−21
B, Motherboard	−1	4	−21	−19

Fig. 2 Payoff matrix for the task "Replace the agent optimized to build LCD Monitors"

	A, LCD	B, LCD	A, Motherboard	B, Motherboard
A, LCD	−22	−24	4	−1
B, LCD	−24	−19	2	4
A, Motherboard	6	4	−16	−21
B, Motherboard	1	6	−21	−19

Fig. 3 Payoff matrix for the task "Replace the agent optimized to build desktop computers"

which case shipping costs are charged only once. Recall that, although both agents are able to assemble any of the two elements, one agent is optimized to build LCD monitors, while the other is optimized to build the desktop computer. This optimization saves \$2.

We can now use the information above to construct the payoff matrix for each of the two tasks, namely "Replace the legacy agent optimized to build LCD Monitors" and "Replace the legacy agent optimized to build desktop computers". The former task can be represented by the payoff matrix in Fig. 2, while the latter can be represented by the payoff matrix in Fig. 3. Each element of the payoff matrices represents the marginal profit of the company in a single production cycle.

In order to understand the values in the payoff matrices, let us consider some of the entries in Fig. 2. We start with the situation where the legacy agent (corresponding to the column player) procures a motherboard from supplier A and the ad hoc agent (corresponding to the row player) procures an LCD panel from supplier A. This situation corresponds to the shaded cell in the first row of the matrix. The total cost of the parts will be $\$10 + \$7 = \$17$. Because both agents procured the parts from the same supplier, the shipping costs (\$2) are only charged once. The PC package is sold for \$25, which corresponds to a profit of $\$25 - \$17 - \$2 = \$6$.

Suppose now that the legacy agent procures an LCD panel from supplier A and the ad hoc agent procures an LCD panel from supplier B. This situation corresponds to the shaded cell in the second row of Fig. 2. In this case, the total cost of the parts will again be $\$10 + \$7 = \$17$, but both agents must now pay shipping costs ($\$2 + \$5 = \$7$) because they procured the parts from different suppliers. Thus, the total amount paid is $\$17 + \$7 = \$24$; however, because a computer package cannot be assembled from two LCD monitors, the agents do not receive any income and, hence, the payoff is $-\$24$.

Finally, if the legacy agent procures an LCD panel from supplier B and the ad hoc procures a motherboard from the same supplier (corresponding to the shaded cell in the last row of Fig. 2), the cost of the parts will be \$14 and the corresponding shipping costs will be \$5. Because the legacy agent is optimized to build desktop computers and not LCD monitors, this translates into an additional production cost of \$2, and the final payoff yields $\$25 - \$14 - \$5 - \$2 = \$4$.

Note that both payoff matrices yield the same equilibria. In other words, if we fix the action of one agent, the best response of the other agent is the same in both tasks. This particular feature of the payoff matrices plays an important role in understanding the differences in behavior between the two approaches proposed herein.

In our approaches, we do not address the ad hoc teamwork problem in its full generality and instead consider the following simplifying assumption. Although, with different teammate models, similar assumptions were considered in previous works [2,61].

Assumption 1 *(Bounded rationality)* Let $h_{1:n} = \{a(1), \ldots, a(n)\}$ denote a specific instance of $H(n)$, $n \geq N$, and let

$$\hat{V}(h_{1:n}, a_{-\alpha}) = \frac{1}{N} \sum_{t=0}^{N-1} U_{T^*}(\langle a_\alpha(n-t), a_{-\alpha} \rangle).$$

Then $\pi_{-\alpha}(h_{1:n}, a^*_{-\alpha}) > 0$ only if $a^*_{-\alpha} \in \operatorname{argmax}_{a_{-\alpha}} \hat{V}(h_{1:n}, a_{-\alpha})$.[4]

Assumption 1 states that the teammate agent uses, at most, the N past observations to select its individual action,[5] i.e.,

$$\pi_{-\alpha}(h_{1:n}, a^*_{-\alpha}) = \mathbb{P}\left[A_k(n+1) = a_k \mid a(n), \ldots, a(n-N+1)\right].$$

Additionally, it does so by selecting a best response to the actions of agent α, computing the expected payoff of each of its individual actions given the history, $\hat{V}(h_{1:n}, a_{-\alpha})$, and selecting an individual action among those that maximize $\hat{V}(h_{1:n}, a_{-\alpha})$. In this sense, the teammate agent follows a *fictitious play* policy [20]. Assumption 1, to some extent, reduces the space of possible teammate models considered by the ad hoc agent. However, this does not liberate the ad hoc agent from conducting teammate identification—in general, the teammate agents will have multiple best response actions available, and the ad hoc agent must identify exactly how the teammates select their action from this set of best response actions.

Finally, Assumption 1 explicits the inductive bias of the learning process taking place in the ad hoc agent, as it establishes that the actions of the teammate agents implicitly provide information on the target task.

3.3 Online learning approach to ad hoc teamwork

In a first approach to the ad hoc teamwork problem formulated in Sect. 3.2, we require the ad hoc agent, at each time-step n, to *predict* the action of its teammate, $A_{-\alpha}(n)$, using the bounded rationality model specified in Assumption 1. To better understand the rationale behind this approach, we note that the ability of the ad hoc agent to predict the action of its teammate is a good indicator of whether the ad hoc agent has successfully identified both the target task and the teammates' strategy. An ad hoc agent that is able to perfectly predict the actions of its teammate will generally be able to perform well—or, at least, no other agent can be expected to perform better without additional knowledge of the target task.

[4] For $0 < n < N$, we define

$$\hat{V}(h_{1:n}, a_{-\alpha}) = \frac{1}{n} \sum_{t=1}^{n} U_{T^*}(\langle a_\alpha(t), a_{-\alpha} \rangle).$$

For $n = 0$ and for all $a_{-\alpha} \in \mathcal{A}_{-\alpha}$, we define $\hat{V}(h_{1:n}, a_{-\alpha}) = \hat{V}(h_0, a_{-\alpha}) = 0$.

[5] These teammates are also known as memory-bounded best response agents.

To this purpose, at each time-step n, the ad hoc agent α selects an action $\hat{A}(n) = \langle A_\alpha(n), \hat{A}_{-\alpha}(n)\rangle$ and incurs a loss given by

$$\ell(\hat{A}(n), A_{-\alpha}(n)) = 1 - \delta(\hat{A}_{-\alpha}(n), A_{-\alpha}(n)), \tag{1}$$

where δ denotes the Kroneker delta function, taking the value 1 if its two arguments are equal and 0 otherwise. The loss in (1) thus penalizes wrong predictions regarding the actions of the teammate.

With the loss defined in (1), it is possible to recast the ad hoc teamwork problem as a simple *online learning problem*, for which a rich body of work exists [12]. The remainder of this subsection details the construction of such online learning problem and the corresponding application of a standard prediction algorithm, providing some general performance guarantees.

Define, for each $\tau \in \mathcal{T}$ and $n \geq N$,[6]

$$\hat{V}_\tau^k(h_{1:n}, a_k) = \frac{1}{N}\sum_{t=0}^{N-1} U_\tau(\langle a_k, a_{-k}(n-t)\rangle), \qquad k = \alpha, -\alpha. \tag{2}$$

For each task $\tau \in \mathcal{T}$, $\hat{V}_\tau^k(h_{1:n}, a_k)$ represents the estimated value of each individual action of agent k, considering the history of the past actions of all other agents. We can also define the associated set of maximizing actions as

$$\hat{\mathcal{A}}_\tau^k(h_{1:n}) = \operatorname{argmax}_{a_k \in \mathcal{A}_k} \hat{V}_\tau^k(h_{1:n}).$$

Associated with each task $\tau \in \mathcal{T}$, we define an *expert* as a mapping $E_\tau : \mathcal{H} \times \mathcal{A} \to [0, 1]$ such that

$$E_\tau(h_{1:n}, a) = E_\tau^\alpha(h_{1:n}, a_\alpha) E_\tau^{-\alpha}(h_{1:n}, a_{-\alpha}),$$

where $\mathcal{H}$ is the set of all finite histories and

$$E_\tau^k(h_{1:n}, a_k) = \begin{cases} \frac{1}{\left|\hat{\mathcal{A}}_\tau^k(h_{1:n})\right|} & \text{if } a_k \in \hat{\mathcal{A}}_\tau^k(h_{1:n}) \\ 0 & \text{otherwise} \end{cases}, \qquad k = \alpha, -\alpha. \tag{3}$$

Intuitively, expert E_τ can be interpreted as providing, at the same time, (i) a (probabilistic) prediction of the behavior of the teammate agent, based on the history and on the fact that the target task may be τ; and (ii) a (probabilistic) suggestion of the best action to make if the target task is τ, based on the recent history of actions of the teammate.

More generally, we define a *predictor* as any mapping $P : \mathcal{H} \times \mathcal{A} \to [0, 1]$ such that, for any history $h_{1:n}$,

$$\sum_{a \in \mathcal{A}} P(h_{1:n}, a) = 1.$$

The value $P(h_{1:n}, a)$ can thus be seen as an estimate of the probability that $A(n+1)$ is action $a \in \mathcal{A}$, given that $H(n) = h_{1:n}$, and essentially corresponds to a generalization of the notion of experts, previously introduced.

If, at time-step $n+1$, the teammate agent plays action $a_{-\alpha}$, we represent the *expected loss of expert* E_τ, given history $h_{1:n}$, as

$$\ell_\tau(h_{1:n}, a_{-\alpha}) = \mathbb{E}_{E_\tau(h_{1:n})}\left[\ell(\hat{A}, a_{-\alpha})\right] \triangleq \sum_{a' \in \mathcal{A}} E_\tau(h_{1:n}, a')\ell(a', a_{-\alpha}) \tag{4}$$

[6] For $n < N$, we extend the definition as in footnote 4.

Algorithm 1 Exponentially weighted forecaster for the ad hoc teamwork problem.

```
1: Initialize $w_\tau^{(0)} = 1$, $h = \emptyset$, $t = 0$.
2: for all $t$ do
3:     Let $t \leftarrow t + 1$
4:     Let
           $$P(h, a) = \frac{\sum_{\tau \in \mathcal{T}} w_\tau^{(t)} E_\tau(h, a)}{\sum_{\tau' \in \mathcal{T}} w_{\tau'}^{(t)}}$$
5:     Select action $\hat{A}(t) = \operatorname{argmax}_{a \in \mathcal{A}} P(h, a)$
6:     Observe action $A_{-\alpha}(t)$
7:     Compute loss $\ell_\tau(h, A_{-\alpha}(t))$ as in (4), $\tau \in \mathcal{T}$
8:     Update
           $$w_\tau^{(t)} \leftarrow w_\tau^{(t-1)} \cdot e^{-\gamma_t \ell_\tau(h, A_{-\alpha}(t))}$$
9: end for
```

and the *expected loss of predictor* P, given history $h_{1:n}$, as

$$\ell_P(h_{1:n}, a_{-\alpha}) \triangleq \sum_{a' \in \mathcal{A}} P(h_{1:n}, a') \ell(a', a_{-\alpha}). \tag{5}$$

The cumulative loss of expert E_τ at time-step n, given the history $h_{1:n}$, is now:

$$L_\tau(h_{1:n}) \triangleq \sum_{t=0}^{n-1} \ell_\tau(h_{1:t}, a_{-\alpha}(t+1)),$$

where $a_{-\alpha}(t)$ represents the teammate action played at time-step t according to history $h_{1:n}$. Similarly, the cumulative loss of predictor P is given by

$$L_P(h_{1:n}) = \sum_{t=0}^{n-1} \ell_P(h_{1:t}, a_{-\alpha}(t+1)).$$

Solving the ad hoc teamwork problem, in this context, consists of determining a predictor P that minimizes the *expected regret*, given by

$$R_n(P, \mathcal{E}) = \mathbb{E}\left[L_P(h_{1:n}) - L_\tau(h_{1:n})\right], \tag{6}$$

where $\mathcal{E} = \{E_\tau, \tau \in \mathcal{T}\}$. We consider the well-known *exponentially weighted average predictor* [12], given by

$$P(h_{1:n}, \hat{a}) \triangleq \frac{\sum_{\tau \in \mathcal{T}} e^{-\gamma_n L_\tau(h_{1:n})} E_\tau(h_{1:n}, \hat{a})}{\sum_{\tau \in \mathcal{T}} e^{-\gamma_n L_\tau(h_{1:n})}}, \tag{7}$$

where γ_n is a positive parameter. The exponentially weighted average predictor predicts the probability of observing each action $\hat{a} \in \mathcal{A}$ by weighting the predictions of each of the experts E_τ, $\tau \in \mathcal{T}$. The online learning approach is summarized in Algorithm 1.

The following result follows from the well-established properties of the exponentially weighted average predictor [12].

Theorem 1 *If* $\gamma_t = \sqrt{2 \ln |\mathcal{E}| / t^{1/2}}$ *for all* $t > 0$, *then, for any finite history* $h_{1:n} \in \mathcal{H}$,

$$R_n(P, \mathcal{E}) \leq \sqrt{\frac{n}{2} \ln |\mathcal{E}|}. \tag{8}$$

Proof See Appendix 1. □

Theorem 1 prompts several important observations. First of all, we note that the bound above provides a minor improvement over previous existing bounds [12, Theorem 2.3] and matches the optimal bound for this class of prediction problems [12]. This result does not depend on the particular setting considered in this paper and is of independent interest *per se*.

A second remark concerns the interpretation of the bound in light of the ad hoc teamwork problem. Theorem 1 states that, in the worst case, the loss sustained by our agent P scales logarithmically with the number of possible tasks that the ad hoc agent may consider, meaning that the performance of the agent does not deteriorate significantly as the number of tasks considered increases. However, the fact that the bound in (8) is a *worst-case* bound implies that the performance of the algorithm will often be significantly superior.

Third, it follows from Theorem 1 that our exponentially weighted average forecaster is a *no-regret algorithm*, as

$$\lim_{n\to\infty}\frac{1}{n}R_n(P,\mathcal{E})=0.$$

However, in the online learning approach just described (and in the corresponding analysis), we have overlooked a key issue in our ad hoc teamwork problem formulation. In particular, we have not considered the impact that the actions of agent α have on the action selection of agent $-\alpha$. In other words, the bound in Theorem 1 measures the predictive ability of the ad hoc agent against that of the "experts" E_τ, $\tau\in\mathcal{T}$, for any possible *but fixed* history h_n. In this sense, the bound in Theorem 1 may be somewhat misleading, and an approach closer to that of de Farias and Megiddo [15] would be more suited.

In the next section, we propose a more evolved approach to overcoming the limitations just identified. In the remainder of this section, we revisit Example 1 and illustrate the application of the online learning approach.

Example 1 (cont.) To illustrate the online learning approach, we consider the possible tasks defined by the payoff matrices in Figs. 2 and 3. We refer to the task "Replace the agent optimized to build LCD Monitors" as task τ_1 and the task "Replace the agent optimized to build desktop computers" as task τ_2.

For concreteness, we henceforth consider that $T^*=\tau_2$ (i.e., the target task is "Replace the agent optimized to build desktop computers") and that the ad hoc agent has a uniform prior over which is the target task. Using the notation from Sect. 3.2, we have

$$\mathcal{A}_\alpha=\mathcal{A}_{-\alpha}=\big\{(\mathsf{A},\,\mathsf{LCD}),\,(\mathsf{B},\,\mathsf{LCD}),\,(\mathsf{A},\,\mathsf{MB}),\,(\mathsf{B},\,\mathsf{MB})\big\},\tag{9}$$

and $p_0(\tau_1)=p_0(\tau_2)=0.5$. In (9), we used a short-hand notation to represent the actions, where (Z,W) indicates the action of purchasing part W from supplier Z.

Upon deployment, no production cycle has taken place; thus, the ad hoc agent's action choice is executed over the *empty history* $h_0=\{\}$. Using (2),

$$\begin{aligned}\hat{V}^\alpha_{\tau_1}\big(h_0,(\mathsf{A},\,\mathsf{LCD})\big)&=\hat{V}^\alpha_{\tau_1}\big(h_0,(\mathsf{B},\,\mathsf{LCD})\big)=0,\\ \hat{V}^\alpha_{\tau_1}\big(h_0,(\mathsf{A},\,\mathsf{MB})\big)&=\hat{V}^\alpha_{\tau_1}\big(h_0,(\mathsf{B},\,\mathsf{MB})\big)=0,\\ \hat{V}^\alpha_{\tau_2}\big(h_0,(\mathsf{A},\,\mathsf{LCD})\big)&=\hat{V}^\alpha_{\tau_2}\big(h_0,(\mathsf{B},\,\mathsf{LCD})\big)=0,\\ \hat{V}^\alpha_{\tau_2}\big(h_0,(\mathsf{A},\,\mathsf{MB})\big)&=\hat{V}^\alpha_{\tau_2}\big(h_0,(\mathsf{B},\,\mathsf{MB})\big)=0.\end{aligned}$$

Because all actions are equally valued, the ad hoc agent will randomly select an action from $\mathcal{A}_\alpha$. Similarly,

$$\begin{aligned}
\hat{V}_{\tau_1}^{-\alpha}\big(h_0, (\mathsf{A}, \mathsf{LCD})\big) &= \hat{V}_{\tau_1}^{-\alpha}\big(h_0, (\mathsf{B}, \mathsf{LCD})\big) = 0,\\
\hat{V}_{\tau_1}^{-\alpha}\big(h_0, (\mathsf{A}, \mathsf{MB})\big) &= \hat{V}_{\tau_1}^{-\alpha}\big(h_0, (\mathsf{B}, \mathsf{MB})\big) = 0,\\
\hat{V}_{\tau_2}^{-\alpha}\big(h_0, (\mathsf{A}, \mathsf{LCD})\big) &= \hat{V}_{\tau_2}^{-\alpha}\big(h_0, (\mathsf{B}, \mathsf{LCD})\big) = 0,\\
\hat{V}_{\tau_2}^{-\alpha}\big(h_0, (\mathsf{A}, \mathsf{MB})\big) &= \hat{V}_{\tau_2}^{-\alpha}\big(h_0, (\mathsf{B}, \mathsf{MB})\big) = 0,
\end{aligned}$$

and the ad hoc agent will predict an action uniformly at random from $\mathcal{A}_{-\alpha}$.

For concreteness, let us suppose that the ad hoc agent chooses action $A_1(1) = (\mathsf{B}, \mathsf{LCD})$, its prediction regarding the legacy agent's action is $\hat{A}_2(1) = (\mathsf{A}, \mathsf{MB})$, and the legacy agent actually selects the action $A_2(1) = (\mathsf{A}, \mathsf{LCD})$. The history after this first production cycle is now $h_1 = \{\langle(\mathsf{B}, \mathsf{LCD}), (\mathsf{A}, \mathsf{LCD})\rangle\}$. Moreover, from (4) and (5),

$$L_{\tau_1}(h_1) = 1, \qquad L_{\tau_2}(h_1) = 0.5, \qquad L_P(h_1) = 0.75$$

and $R_0(P, \mathcal{E}) = 0.25$.

In the second step, we have:

$$\begin{aligned}
\hat{V}_{\tau_1}^{\alpha}\big(h_1, (\mathsf{A}, \mathsf{LCD})\big) &= -22, & \hat{V}_{\tau_1}^{\alpha}\big(h_1, (\mathsf{B}, \mathsf{LCD})\big) &= -24,\\
\hat{V}_{\tau_1}^{\alpha}\big(h_1, (\mathsf{A}, \mathsf{MB})\big) &= 4, & \hat{V}_{\tau_1}^{\alpha}\big(h_1, (\mathsf{B}, \mathsf{MB})\big) &= -1,\\
\hat{V}_{\tau_2}^{\alpha}\big(h_1, (\mathsf{A}, \mathsf{LCD})\big) &= -22, & \hat{V}_{\tau_2}^{\alpha}\big(h_1, (\mathsf{B}, \mathsf{LCD})\big) &= -24,\\
\hat{V}_{\tau_2}^{\alpha}\big(h_1, (\mathsf{A}, \mathsf{MB})\big) &= 6, & \hat{V}_{\tau_2}^{\alpha}\big(h_1, (\mathsf{B}, \mathsf{MB})\big) &= 1,
\end{aligned}$$

and the ad hoc agent will select the action $A_1(2) = (\mathsf{A}, \mathsf{MB})$. Similarly,

$$\begin{aligned}
\hat{V}_{\tau_1}^{-\alpha}\big(h_1, (\mathsf{A}, \mathsf{LCD})\big) &= -24, & \hat{V}_{\tau_1}^{-\alpha}\big(h_1, (\mathsf{B}, \mathsf{LCD})\big) &= -19,\\
\hat{V}_{\tau_1}^{-\alpha}\big(h_1, (\mathsf{A}, \mathsf{MB})\big) &= 4, & \hat{V}_{\tau_1}^{-\alpha}\big(h_1, (\mathsf{B}, \mathsf{MB})\big) &= 6,\\
\hat{V}_{\tau_2}^{-\alpha}\big(h_1, (\mathsf{A}, \mathsf{LCD})\big) &= -24, & \hat{V}_{\tau_2}^{-\alpha}\big(h_1, (\mathsf{B}, \mathsf{LCD})\big) &= -19,\\
\hat{V}_{\tau_2}^{-\alpha}\big(h_1, (\mathsf{A}, \mathsf{MB})\big) &= 2, & \hat{V}_{\tau_2}^{-\alpha}\big(h_1, (\mathsf{B}, \mathsf{MB})\big) &= 4.
\end{aligned}$$

Therefore, the ad hoc agent will predict that its teammate will play $(\mathsf{B}, \mathsf{MB})$. Because $T^* = \tau_2$, $\hat{A}_2(2) = (\mathsf{B}, \mathsf{MB})$, we have

$$L_{\tau_1}(h_2) = 1, \qquad L_{\tau_2}(h_2) = 0.5, \qquad L_P(h_2) = 0.75,$$

and $R_1(P, \mathcal{E}) = 0.25$. Continuing this process, we can empirically estimate the regret of the exponentially weighted average forecaster.

Figure 4 depicts the average cumulative regret suffered by our proposed predictor over 100 time steps, comparing it with the theoretical bound in Theorem 1. It is clearly observable that the regret against the best expert remains constant after a few time steps, indicating that, indeed, the exponentially weighted average forecaster used in this example is able to identify the strategy of the teammate after only a few iterations. We also observe that, as expected, the theoretical bound largely overestimates the regret because, in this case, the tasks considered have a well-defined set of optimal actions, which leads to a significant improvement over the worst-case performance. Finally, although coordination is quickly attained, the structure of the payoff matrices does not enable the ad hoc agent to unambiguously identify the target task.

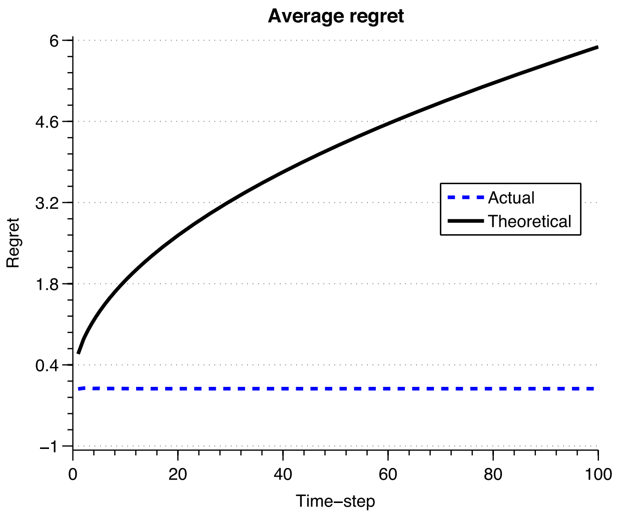


Fig. 4 Average cumulative regret of the exponentially weighted average predictor in the *e*-commerce scenario. This result corresponds to the average of 1,000 independent Monte-Carlo trials

3.4 Decision-theoretic approach to ad hoc teamwork

In this section, we improve on the online learning approach described in Sect. 3.3, considering two additional elements available to the ad hoc agent:

(i) The prior knowledge about the target task, encoded in the distribution p_0;
(ii) The impact of the actions of agent α on those of the teammate agent.

The two elements, (i) and (ii), impact our approach in distinct ways. The consideration of the prior p_0, alluded to in (i), suggests a Bayesian approach to the ad hoc teamwork problem, possibly along the lines of the seminal work of Gittins [26] or the more recent works of Kauffman et al. [35,36].

As for (ii), as briefly discussed in Sect. 3.3, it impacts the way in which *regret* is defined. In particular, the performance of our predictor should not be measured against that of the best expert given the observed history, but against that of the best expert for the history that would be observed, had our prediction followed that expert from the start. Previous works that consider the effect of the predictor P on the experts include that of de Farias and Megiddo [15] and generalizations thereof [12].

We model our ad hoc agent using a *decision-theoretic framework*, where the goal of the agent is to minimize the expected loss (w.r.t. its ability to predict the action of the teammate agent), while simultaneously maximizing the payoff in the target task.[7]

In this formulation, the target task, T^*, is considered an unobserved random variable. The ad hoc agent will then keep, at each time step n, a distribution p_n over the space of possible tasks, where

$$p_n(\tau) = \mathbb{P}\left[T^* = \tau \mid H(n-1)\right], \tag{10}$$

for all $\tau \in \mathcal{T}$. We refer to $p_n(\tau)$ as the *belief* of agent α at time step n related to what the target task is, which will guide the process of action selection of the ad hoc agent.

[7] These are two distinct goals, and both must be reflected in the POMDP formulation. While the former seeks to identify the target task, the latter seeks to play that task as well as possible.

In what follows, we describe the decision-theoretic framework of partially observable Markov decision problems (POMDP) in full generality before instantiating this particular framework to the ad hoc teamwork problem.

3.4.1 Partially observable Markov decision problems

Partially observable Markov decision problems (POMDPs) provide a general framework for modeling sequential decision problems in the face of uncertainty [33]. At each time step, and depending on its perception of the environment, the agent in a POMDP must select an action from its action repertoire in order to maximize a numerical reward signal. Actions determine how the state of the environment evolves through time and, depending on the state, different actions lead to a different reward for the agent.

The *state of the environment* is the set of all features of the environment that are relevant for the agent to choose its actions optimally. Ideally, the agent should be able to unambiguously perceive all such features. In a POMDP, however, the agent has limited sensing capabilities and is not able to completely determine the current state of the system.

A POMDP can thus be represented as a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathsf{P}, \mathsf{O}, r, \gamma)$, where:

– $\mathcal{X}$ is the *state-space*. The state of the environment at time-step n is a r.v. $X(n)$ taking values in $\mathcal{X}$.
– $\mathcal{A}$ is the *action-space*. At each time-step, the agent must select an action from $\mathcal{A}$. We denote the action at time-step n as a r.v. $A(n)$ taking values in $\mathcal{A}$.
– $\mathcal{Z}$ is the *observation-space*. The observation made by the agent at time-step n is represented as the r.v. $Z(n)$ taking values in $\mathcal{Z}$. Since this observation depends on the state $X(n)$ of the environment, the agent can use $Z(n)$ to track the underlying state $X(n)$.
– P represents the *transition probabilities*, which encode how the state $X(n)$ evolves as a function of the agent's actions. In particular,

$$\mathsf{P}(x' \mid x, a) \triangleq \mathbb{P}\left[X(n+1) = x' \mid X(n) = x, A(n) = a\right].$$

– O represents the *observation probabilities*, which encode how the observations $Z(n)$ depend on the state $X(n)$ and on the agent's actions. In particular,

$$\mathsf{O}(z \mid x, a) \triangleq \mathbb{P}\left[Z(n+1) = z \mid X(n+1) = x, A(n) = a\right].$$

– Finally, the function $r : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, which implicitly encodes the goal of the agent. The value $r(x, a)$ represents the immediate reward for performing action a in state x.

The goal of the agent in a POMDP is to select its actions to maximize the total reward received throughout its lifetime. Formally, this corresponds to selecting the actions that maximize the value

$$J = \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n r(X(n), A(n))\right],$$

where γ is a constant known as the *discount factor* and is such that $0 \leq \gamma < 1$.

Standard approaches to solving POMDPs involve tracking the underlying state $X(n)$ using the history of observations. Let p_n be a probability distribution over $\mathcal{X}$, where $p_n(x)$ represents the *belief* of the agent that $X(n) = x$, $x \in \mathcal{X}$, i.e.,

$$p_n(x) = \mathbb{P}\left[X(n) = x \mid H(n)\right],$$

where

$$H(n) = \{a(0), z(1), a(1), \ldots, a(n-1), z(n)\}.$$

Given that the action at time-step n was $A(n) = a$ and the subsequent observation was $Z(n+1) = z$, the updated belief p_{n+1} can be computed, componentwise, as a function of p_n, a and z as

$$p_{n+1}(x') = B(p_n, z, a)_{x'} \triangleq \xi \sum_{x \in \mathcal{X}} p_n(x) \mathsf{P}(x' \mid x, a) \mathsf{O}(z \mid x', a), \tag{11}$$

where ξ is a normalization constant. It is now possible to define an *optimal decision rule* that maps, at each time-step n, the belief p_n to an action in $\mathcal{A}$ [33]. This rule is *time-invariant* and simply selects, at each time-step n, an action $A(n)$ verifying

$$A(n) \in \operatorname{argmax}_{a \in \mathcal{A}} \sum_x p_n(x) \left[r(x,a) + \gamma \sum_{x',z} \mathsf{P}(x' \mid x, a) \mathsf{O}(z \mid x', a) V^*(B(p_n, z, a)) \right],$$

where V^* is the solution to the recursive relation

$$V^*(p_n) = \max_{a \in \mathcal{A}} \sum_x p_n(x) \left[r(x,a) + \gamma \sum_{x',z} \mathsf{P}(x' \mid x, a) \mathsf{O}(z \mid x', a) V^*(B(p_n, z, a)) \right].$$

Unfortunately, partially observable Markov decision problems are generally undecidable [43], and V^* cannot be computed exactly. Instead, a wide-range of approximate methods are available in the literature, the most popular of which are, perhaps, *point-based methods* [46]. In this paper, we adopt the popular PERSEUS point-based algorithm [56].

3.4.2 POMDP formulation of the ad hoc teamwork problem

In order to apply the PERSEUS algorithm to the ad hoc teamwork problem, we start by formulating the latter as a POMDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathsf{P}, \mathsf{O}, c, \gamma)$. Let $H_N(n)$ denote the last N actions in the history $H(n)$, i.e.,

$$H_N(n) = \{A(n-N+1), \ldots, A(n)\},$$

and let $\mathcal{H}_N$ denote the set of all such sub-histories. We have, then,

- The state, at time-step n, is the r.v. $X(n) = (H_N(n-1), T^*)$ taking values in the state-space $\mathcal{X} = \mathcal{H}_N \times \mathcal{T}$.
- The action-space is $\mathcal{A} = \mathcal{A}_\alpha \times \mathcal{A}_{-\alpha}$. As in Sect. 3.3, the ad hoc agent must select, at each time-step n, an action $\hat{A}(n) = \langle A_\alpha(n), \hat{A}_{-\alpha}(n) \rangle$, where $A_\alpha(n)$ is its individual action and $\hat{A}_{-\alpha}$ is its prediction of the action of the teammate agent.
- The observation-space is $\mathcal{Z} = \mathcal{A}_{-\alpha}$. After selecting an action $\hat{A}(n)$ at time-step n, agent α makes an observation $Z(n+1)$ corresponding to the action $A_{-\alpha}(n)$ just selected by its teammate agent. The component $X_{\mathcal{H}}(n)$ of the state is, therefore, *fully observable*, i.e., it is unambiguously perceived by agent α.
- The *transition probabilities*, which encode how the state $X(n)$ evolves as a function of the agent's actions, can be factored in two components. One component describes the dynamics of the target task, $X_{\mathcal{T}}(n)$, which remains unchanged because the target task does not change. Formally, this corresponds to

$$\begin{aligned}\mathsf{P}_{\mathcal{T}}(\tau' \mid \tau, a) &\triangleq \mathbb{P}\left[X_{\mathcal{T}}(n+1) = \tau' \mid X_{\mathcal{T}}(n) = \tau, A(n) = a\right] \\ &= \delta(\tau, \tau')\end{aligned}$$

where, as before, δ denotes the Kronecker delta function. The component of $X(n)$ corresponding to the history, $X_{\mathcal{H}}(n)$, is updated to include the latest action of both agents. Formally, if we consider two histories $h', h \in \mathcal{H}_N$ such that $h'_{1:N-1} = h_{2:N}$, let $a'(N)$ denote the most recent action in h', where $a'(N) = \left\langle a'_{\alpha}(N), a'_{-\alpha}(N)\right\rangle$. Then, for any $a \in \mathcal{A}$,

$$\begin{aligned}\mathsf{P}_{\mathcal{H}}(h' \mid h, \tau, a) &\triangleq \mathbb{P}\left[X_{\mathcal{H}}(n+1) = h' \mid X_{\mathcal{H}}(n) = h, X_{\mathcal{T}}(n) = \tau, A(n) = a\right] \\ &= E_{\tau}^{-\alpha}(h, a'_{-\alpha}(N))\delta(a_{\alpha}, a'_{\alpha}(N)),\end{aligned}$$

where $E_{\tau}^{-\alpha}$ is defined in (3). Finally,

$$\mathsf{P}(h', \tau' \mid h, \tau, a) = \mathsf{P}_{\mathcal{T}}(\tau' \mid \tau, a)\mathsf{P}_{\mathcal{H}}(h' \mid h, \tau, a).$$

– The *observation probabilities*, which encode the dependence of the observation $Z(n+1)$ on the state $X(n+1)$ and the agent's action, can be written as

$$\begin{aligned}\mathsf{O}(a'_{-\alpha} \mid h, \tau, a) &\triangleq \mathbb{P}\left[Z(n+1) = a'_{-\alpha} \mid X(n+1) = (h, \tau), A(n) = a\right] \\ &= \delta(a'_{-\alpha}, a_{-\alpha}(N)),\end{aligned}$$

where $a(N)$ is the most recent action in h and $a(N) = \langle a_{\alpha}(N), a_{-\alpha}(N)\rangle$.

– The *reward function* r is given by

$$\begin{aligned}r(h, \tau, a) = &\left(1 - \sum_{\hat{a} \in \mathcal{A}} E_{\tau}(h, \hat{a})\ell(\hat{a}, a)\right)\left(\sum_{\hat{a} \in \mathcal{A}} E_{\tau}(h, \hat{a})U_{\tau}(a_{\alpha}, \hat{a}_{-\alpha})\right) \\ &- \sum_{\hat{a} \in \mathcal{A}} E_{\tau}(h, \hat{a})\ell(\hat{a}, a)\max_{a} |U_{\tau}(a)|,\end{aligned} \tag{12}$$

where the loss ℓ is defined in (1). Note that the reward function accounts for each of the action components.

The reward function includes two components. The first component weights the payoff received from the interaction by the loss, implying that large losses reduce the payoff. The second component is used to avoid pathological policies that reduce the impact of negative payoffs by artificially augmenting the loss.

With the above definitions, we have modeled the ad hoc teamwork problem as a partially observable Markov decision problem and are in position to apply the PERSEUS algorithm. PERSEUS provides a decision rule P that maps beliefs over the POMDP's state-space, $\mathcal{X}_{\mathcal{H}} \times \mathcal{X}_{\mathcal{T}}$, to actions.

We note, however, that the history component of the state is *fully observable*; thus, the agent needs only to keep a belief over the component $X_{\mathcal{T}}(n)$ of the state. As can easily be seen from the definitions above, the belief update in (11) reduces to an update of the belief over the target task, given by

$$\begin{aligned}p_{n+1}(\tau) &\triangleq \mathbb{P}\left[T^* = \tau \mid H(n)\right] \\ &= \xi\mathbb{P}\left[A_{-\alpha}(n+1) = a_{-\alpha} \mid T^* = \tau, H(n) = h_{1:n}\right]\mathbb{P}\left[T^* = \tau \mid H(n) = h_{1:n}\right] \\ &= \xi E_{\tau}^{-\alpha}(h_{1:n}, a_{-\alpha})p_n(\tau),\end{aligned} \tag{13}$$

where ξ is a normalization constant and $E_\tau^{-\alpha}$ is defined in (3). The decision rule P obtained from PERSEUS thus maps, at each time-step n, a pair $(H_N(n), p_n)$ to an action $P(H_N(n), p_n) \in \mathcal{A}$, where p_n denotes the belief over $\mathcal{T}$, defined in (10).

Example 1 (cont.) To apply the POMDP approach to the *e*-commerce scenario, we simply map the problem elements to the different POMDP components. We have:

– $\mathcal{X} = \mathcal{H}_N \times \{\tau_1, \tau_2\}$;
– $\mathcal{A} = \mathcal{A}_\alpha \times \mathcal{A}_{-\alpha}$, where, as before,

$$\mathcal{A}_\alpha = \mathcal{A}_{-\alpha} = \big\{(\mathsf{A}, \mathsf{LCD}), (\mathsf{B}, \mathsf{LCD}), (\mathsf{A}, \mathsf{MB}), (\mathsf{B}, \mathsf{MB})\big\};$$

– $\mathcal{Z} = \mathcal{A}_{-\alpha} = \big\{(\mathsf{A}, \mathsf{LCD}), (\mathsf{B}, \mathsf{LCD}), (\mathsf{A}, \mathsf{MB}), (\mathsf{B}, \mathsf{MB})\big\}$.

Applying PERSEUS to the POMDP, we can now evaluate the performance of this approach in the *e*-commerce scenario. We rely on the simplicity of this scenario to investigate the impact of each reward component on the general performance of the POMDP.

Figure 5 illustrates the performance of the POMDP approach, where the results correspond to averages over 1,000 independent Monte-Carlo trials.

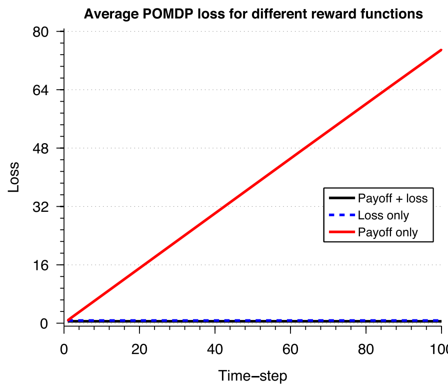
As expected, if the reward function is defined only as a function of the loss,

$$r(h, \tau, a) = \left(1 - \sum_{\hat{a} \in \mathcal{A}} E_\tau(h, \hat{a})\ell(\hat{a}, a)\right),$$

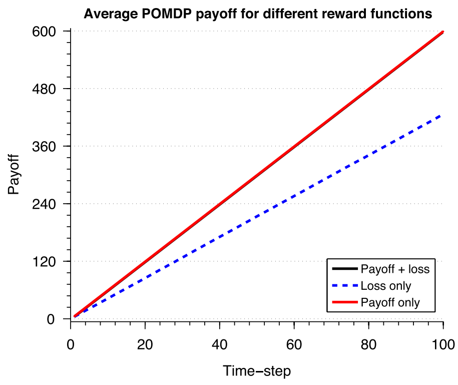
the "POMDP agent" predicts the teammate's actions and selects its own actions to maximize its predictive capability, ignoring the impact on payoff. On the other hand, if only payoff is considered, the reward becomes

$$r(h, \tau, a) = \sum_{\hat{a} \in \mathcal{A}} E_\tau(h, \hat{a}) U_\tau(a_\alpha, \hat{a}_{-\alpha}),$$

and the agent will essentially make random predictions, incurring a significant loss but improving the performance in terms of reward. Combining the two, as in (12), the POMDP agent is able to attain the best performance in terms of both criteria.



(a) Average loss.



(b) Average payoff.

Fig. 5 POMDP performance in the *e*-commerce scenario for different reward functions

To assess the performance of the obtained decision-rule P, we compare it with that of the best expert in terms of the corresponding total expected reward. In particular, if $T^* = \tau^*$, we define the cumulative value of expert E_τ up to time step n as

$$J_\tau(n, \tau^*) = \mathbb{E}_\tau \left[\sum_{t=0}^{n} r(X(t), A(t)) \right]$$

and the cumulative value of the decision-rule P up to time step n as

$$J_P(n, \tau^*) = \mathbb{E}_P \left[\sum_{t=0}^{n} r(X(t), A(t)) \right].$$

From these, the expected regret of our algorithm is defined as

$$R_n(P, \mathcal{E}) = J_{\tau^*}(n, \tau^*) - J_P(n, \tau^*). \tag{14}$$

We have the following result.

Theorem 2 *The POMDP-based approach to the ad hoc teamwork problem is a* no-regret *approach, i.e.,*

$$\lim_{n \to \infty} \frac{1}{n} R_n(P, \mathcal{E}) = 0. \tag{15}$$

Proof The proof proceeds by deriving an upper bound for $R_n(P, \mathcal{E})$ of the general form

$$R_n(P, \mathcal{E}) \leq \mathrm{D_{KL}}\,(P_n \| Q_n) + \log \frac{n}{2} \tag{16}$$

for some adequately defined distributions P_n and Q_n over the set of n-length histories, $\mathcal{H}_n$, where we write $\mathrm{D_{KL}}\,(p \| q)$ to denote the *Kullback–Liebler* divergence between distributions p and q.

The first term on the right-hand side of (16) measures how the possible histories experienced by the agent are affected by the fact that the ad hoc agent does not know the target task, while the second term bounds the deviations between the "average history" and the "nominal history" actually experienced by the ad hoc agent. The result follows from the fact that $\mathrm{D_{KL}}\,(P_n \| Q_n) \in o(n)$.

We refer to Appendix 1.2 for details of the proof. □

We conclude this section with several observations concerning the result in Theorem 2. First of all, unlike the online learning approach featured in Sect. 3.3, Theorem 2 does not provide an explicit bound on the regret. This is due to the added complexity arising from the two issues that the decision-theoretic approach seeks to address: the use of a Bayesian formalism to the identification of the target task and the consideration of the impact of the actions of the ad hoc agent on those of the teammate agent.

Nevertheless, previous work has provided explicit bounds for the quantity $\mathrm{D_{KL}}\,(P_n \| Q_n)$ in different settings and under specific conditions (see, for example, the work of Clarke and Barron [14]). Although the aforementioned results do not directly apply to our settings, we conjecture that it may be possible to extend such analysis to our case, leading to a general bound on the regret that is $O(\log n)$. Such result would also match (up to constants) the lower bound provided in the precursor work of Lai and Robbins [39] (see also [35]), establishing our POMDP approach as near-optimal.

A second remark concerns the close relation between Theorem 2 and a rich body of work from both the game-theoretic community on Bayesian learning in games [9,18,19,31,32, 34,44] and the information-theoretic community on the convergence of Bayesian estimation [8,14,17,24,25,28,29,54,63].

Finally, it is worth noting that the potential benefits, in terms of performance arising from the POMDP approach, come at a cost in complexity. In fact, even if an approximate solver such as PERSEUS is used, POMDPs are inherently hard to solve. Moreover, as the length of the history considered in the component $X_{\mathcal{H}}$ of the state increases, so does the POMDP state-space and the corresponding planning complexity. Therefore, the two approaches proposed herein offer a careful tradeoff between performance and complexity that should be taken into consideration upon adopting either.

In the next section, we evaluate both proposed approaches in several test scenarios, providing a clearer perspective on the applicability and trade-offs involved in using each of the two proposed approaches in practical settings.

4 Empirical evaluation

This section describes several experiments for empirically validating the properties of the two proposed approaches discussed in the previous section. For ease of presentation, we henceforth refer to an ad hoc agent running the online learning approach of Sect. 3.3 as an *OL agent* and an ad hoc agent running the POMDP-based approach of Sect. 3.4 as a *POMDP agent*.

4.1 Methodology

To empirically investigate the performance of the OL and POMDP agents, we conduct three sets of experiments, each discussed in a specific section. In particular,

- Section 4.2 discusses the performance of both proposed approaches in the *e*-commerce scenario used as a running example throughout the document. This first set of experiments illustrates the relative merits of the two proposed approaches, taking advantage of the fact that the *e*-commerce scenario is amenable to an intuitive interpretation of the results due to its structure and simplicity. Additionally, as will become apparent, the *e*-commerce scenario is also particularly adequate to assess the ability of each of the two proposed approaches to identify and influence the strategy of the teammate agent.
- Section 4.3 extends the empirical evaluation to scenarios with no particular structure. We analyze the scalability of both proposed approaches in terms of (i) the number of tasks; (ii) the number of agents involved in the interaction; and (iii) the number of actions available to each agent. To this purpose, we evaluate the performance of the OL and the POMDP agents in random scenarios of increasing complexity.
- Section 4.4 investigates the general applicability of our approach in a benchmark scenario from the literature on multi agent systems, the *pursuit domain* [6,7]. In this benchmark scenario, we conduct a qualitative comparison with a related approach from the ad hoc teamwork literature [7].

All experiments follow a similar methodology. One ad hoc agent interacts with one or more "legacy agents" that (i) know the target task and (ii) are "programmed" to adapt their actions to those of the ad hoc agent (as seen in Sect. 3). From the interaction with the other agents, the ad hoc agent must both infer the target task (corresponding to the *Task Identification* block of Fig. 1) and coordinate its actions with those of the teammates (*Teammate Identification* and *Planning* blocks in Fig. 1).

Table 2 Summary of all agents used for comparison

Agent	OL	POMDP	OL (k.t.)	RL	MDP
Knows T^*	No	No	Yes	No	Yes
Preplans	No	Yes	No	No	Yes
Learns online	Yes	No	No	Yes	No
Has state	No	Yes	No	Yes	Yes
Performance	Both	Both	Loss-only	Payoff-only	Payoff-only

The last line reports the performance indicators (loss, payoff or both) used to evaluate the different agents

The performance of the ad hoc agents is evaluated both in terms of loss and payoff. The loss measures the ability of the ad hoc agent to predict the actions of the teammate agents, while the payoff quantifies the overall performance of all agents as a team. All results reported correspond to averages over 1,000 independent Monte Carlo trials, where each trial consists of a run of 100 learning steps.

The set of tasks from which the ad hoc agent must single out the target task is specific for each scenario and is described in the corresponding section. In all experiments, the ad hoc agent is provided with a uniform prior over the set of possible tasks, i.e., the agent believes that no one task is more likely to be the target task than the others.

4.1.1 Agents

In the experiments reported in Sects. 4.2 and 4.3, we include the performance of several different agents, which provides a baseline for comparison. The agents reported in the experiments are summarized in Table 2. In addition to the OL and POMDP agents described in Sect. 3, we also evaluate the performance of

- *An OL Agent with known target task* [OL (k.t.)] This ad hoc agent corresponds to a version of the OL agent that knows the target task. It needs only to coordinate its actions with those of the teammates, predicting the actions of the teammate according to the correct expert, E_{T^*}. Setting this agent to share the history of the OL agent enables an empirical estimate of the regret of the OL agent which, in turn, enables a comparison between the empirical performance of the latter with the theoretical bounds reported in Theorem 1. The OL agent with a known task is evaluated in terms of loss only.
- *An MDP agent* [MDP] This ad hoc agent corresponds to a version of the POMDP agent that knows the target task. Referring back to Sect. 3.4, the target task is the only component of the POMDP state that is not observable to the ad hoc agent. As such, an agent that knows the target task does not suffer from partial observability and can be modeled using a simple *Markov decision process* (MDP).
  MDPs are a subclass of POMDPs, where the agent is able to unambiguously perceive the underlying state. As such, it is not necessary to explicitly represent or reason about the observation-space or observation probabilities. Computing the optimal policy is also much simpler in MDPs than in their partially observable counterpart. In terms of our experiments, we use the MDP agent as an indication of the optimal performance because the MDP agent has access to all information necessary to make an optimal decision. The MDP agent is evaluated in terms of payoff only.

- *An RL agent* [RL] The RL agent is a standard reinforcement learning agent (RL) running the well-known Q-learning algorithm [65].[8] This agent has no prior knowledge about the target task or the teammate behavior, and must learn both the task and the coordination policy by a process of trial-and-error. Unlike the other agents, the RL agent actually receives the payoff associated with the target task after every joint action is performed. The payoff observed, as well as the joint actions observed in the past, allow the RL agent to adapt its action choice to maximize the expected payoff. The RL agent is evaluated in terms of payoff only.

We conclude by remarking that, in the POMDP and MDP agent, we set $\gamma = 0.95$.[9] The γ parameter is required because all tasks are formalized as infinite horizon tasks. As will soon become apparent, the particular value of γ selected facilitates the analysis of our results, particularly in the early steps of the experiments.

4.2 Ad hoc teamwork for *e*-commerce

In the first set of experiments, we evaluate the performance of the different agents in the *e*-commerce scenario. We replicate and discuss in greater detail some of the results already portrayed in previous sections.

The *e*-commerce scenario is a relatively simple scenario, where an ad hoc agent and a "legacy" agent are responsible for assembling desktop computer packages for sale. Each agent is responsible for assembling one of the package elements (either the desktop computer or the LCD monitor), for which it must procure the necessary part. The parts can be procured from one of two possible suppliers, and if both agents procure the corresponding parts from the same supplier, they incur smaller shipping costs.

This initial set of experiments serves two main purposes. First, together with the illustrative examples along the text, it provides a roadmap to the application of the online learning and POMDP approaches in a specific problem, detailing each of the steps involved. Second, the particular structure of the problem—where the action choices of the teammate agents do not provide significant information concerning the underlying task—implies that the main challenge that the ad hoc agent faces is one of *coordination*. As such, the *e*-commerce scenario is ideal to assess the impact of *history* in the performance of both approaches.

Table 3 summarizes the overall performance of the analyzed approaches, both in terms of loss and payoff, for $H = 1, 2, 3$.[10] From the results in Table 3, it becomes apparent that the POMDP agent outperforms the OL agent, both in terms of loss and in terms of payoff. The difference is particularly noticeable in terms of payoff because the POMDP agent plans ahead to optimize both criteria, while the OL agent simply selects the apparently best action.

Let us consider the observed difference between the OL and POMDP agents in greater detail. Going back to the payoff matrices in Figs. 2 and 3, we note that both tasks present the same set of *equilibria*, i.e., joint actions from which no agent has any interest in deviating unilaterally. In other words, if we fix the action played by one agent, the best-response action

[8] In our implementation, the RL agent runs the standard Q-learning algorithm with a fixed learning rate of $\alpha = 0.2$. The value for α was empirically selected to maximize the learning performance of the agent. Additionally, exploration is ensured using a greedy policy combined with optimistic initialization of the Q-values, as discussed in the classical book of Sutton and Barto [62].

[9] We recall that low values of γ indicate the MDP agent that payoffs arriving sooner are more valuable than payoffs arriving later.

[10] We recall that, except for the RL agent, the results presented merely quantify the payoff that the agents would obtain from their action choices. At runtime, no payoff is granted to the agents, i.e., payoff information is not available for learning.

Table 3 Performance of the different approaches in the *e*-commerce scenario for different horizon lengths. The results are averages over 1, 000 independent Monte Carlo runs

	Agent	$H = 1$	$H = 2$	$H = 3$
Loss	POMDP	1.468 ± 1.403	1.365 ± 1.181	1.255 ± 1.060
	OL	1.500 ± 1.565	1.389 ± 1.269	1.294 ± 1.026
	OL (known task)	1.510 ± 1.399	1.395 ± 1.173	1.298 ± 0.946
Payoff	POMDP	571.0 ± 36.2	571.0 ± 31.3	572.0 ± 32.0
	OL	505.7 ± 112.0	503.6 ± 111.8	497.9 ± 114.2
	MDP (known task)	522.8 ± 82.7	531.0 ± 83.00	541.1 ± 79.7
	RL	426.6 ± 116.4	273.2 ± 185.6	-113.1 ± 364.2

for the other agent is the same in both tasks. This has an immediate consequence—if an ad hoc agent merely seeks to coordinate its action choice with that of the other agent, they may converge to a suboptimal equilibrium (i.e., one with a payoff of 4) instead of the optimal equilibrium, with a payoff of 6. Taking this fact into consideration, it is now possible to further interpret the results of Table 3:

- Due to the structure of the exponentially weighted average predictor, the behavior of the OL agent is tightly coupled with its ability to predict the actions of the teammate agent (see Sect. 3.3). However, in this particular scenario, the OL agent is able to perfectly predict the teammate's action and coordinate, *independent of the task*. Therefore, predicting ability is insufficient to identify the target task, and the OL agent is equally likely to converge either to an optimal equilibrium (with a payoff of 6) or a suboptimal one (with a payoff of 4). This explains the average total payoff of ~500 obtained (corresponding to a 100-step run).
- Conversely, the POMDP agent selects its actions taking into account their future impact in the total payoff. Therefore, in the early plays, it both seeks to determine the target task *and drive the teammate agent to play the corresponding optimal equilibrium*. This is visible by observing the obtained average payoff close to 600 (corresponding to a 100-step run).

Another interesting observation is the dependence of the performance of the agents on the history length. The results suggest that the performance of the POMDP agent slightly improves as we increase the horizon length from $H = 1$ to $H = 3$. Interestingly, such a difference occurs only in terms of loss. The OL agent, on the other hand, seems to always improve its performance, in terms of loss, as the history builds. This improvement is not observed in the payoff for reasons already discussed. The impact of the horizon length is greatest in the performance of the RL agent, which can easily be explained by the fact that a larger horizon implies a larger state-space (i.e., the number of possible situations that the agent must reason about) and longer learning times.

It is also interesting to consider the reward obtained by the MDP agent. Due to the discount factor considered, the MDP agent prefers payoffs that occur earlier rather than payoffs occurring later in the run. Therefore, the potential loss incurred by initially "driving" the teammate agent towards the optimal equilibrium does not compensate for the extra payoff received later on. The advantages of this behavior are illustrated in Fig. 6, which depicts the learning curve of all agents in terms of total discounted payoff attained. In the plot, the impact of the discount is clearly visible and the optimality of the MDP agent becomes apparent. However, if no discount is taken into account, the performance of the MDP agent actually remains

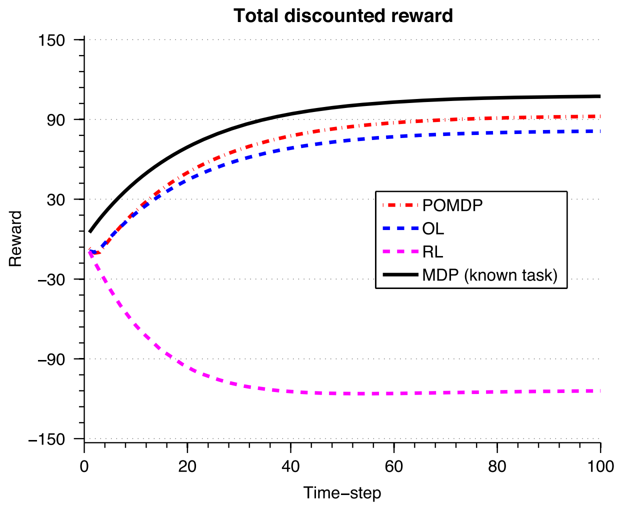


Fig. 6 Average discounted payoff of the different approaches in the *e*-commerce scenario for a horizon $H = 3$. The results are averages over 1,000 independent Monte Carlo runs

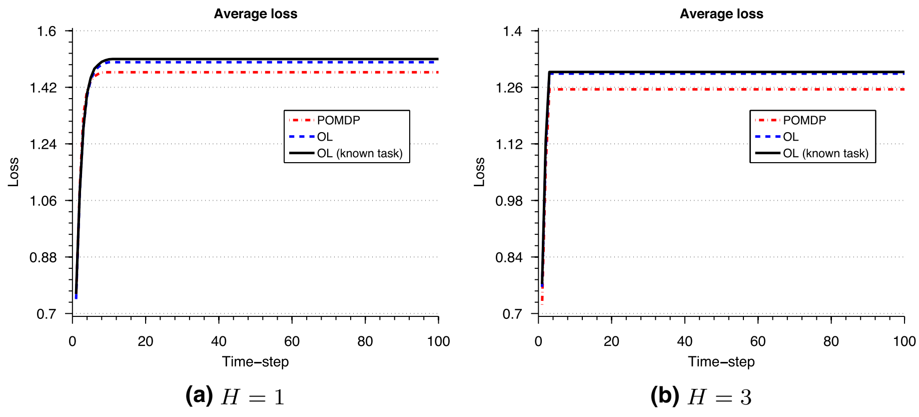


Fig. 7 Average loss of the different approaches in the *e*-commerce scenario for different horizon lengths. The results are averages over 1,000 independent Monte Carlo runs

lower than that of the POMDP agent, indicating that the latter is indeed able to drive the teammate to an optimal joint action.

Let us now consider the learning curves of all agents, both in terms of loss incurred and (undiscounted) payoff attained, as depicted in Figs. 7 and 8. It is once again visible that the POMDP agent outperforms the OL agent, confirming the outcome already observed in Table 3. The learning curves illustrate two additional aspects that are worth noting. First, the loss incurred by both the POMDP and the OL agents is concentrated in the early learning steps. After approximately 10 steps, both ad hoc agents are able to perfectly predict the actions of the teammate. The plots also illustrate that a larger history enables this predictive ability to emerge earlier, leading to smaller loss when $H = 3$ when compared to the situation where $H = 1$.

Second, comparing the slope of the total payoff curve for the POMDP agent against that of the OL agent in Fig. 8, one can observe that the POMDP agent is more effective in terms of

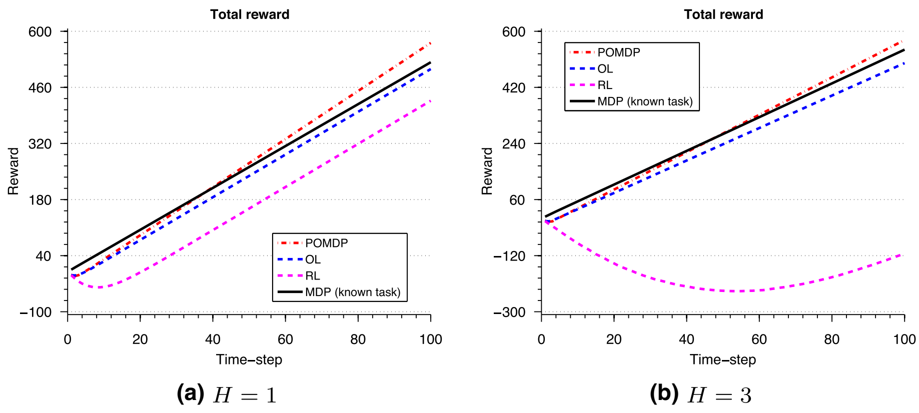


Fig. 8 Average payoff of the different approaches in the *e*-commerce scenario, for different horizon lengths. The results are averages over 1,000 independent Monte Carlo runs

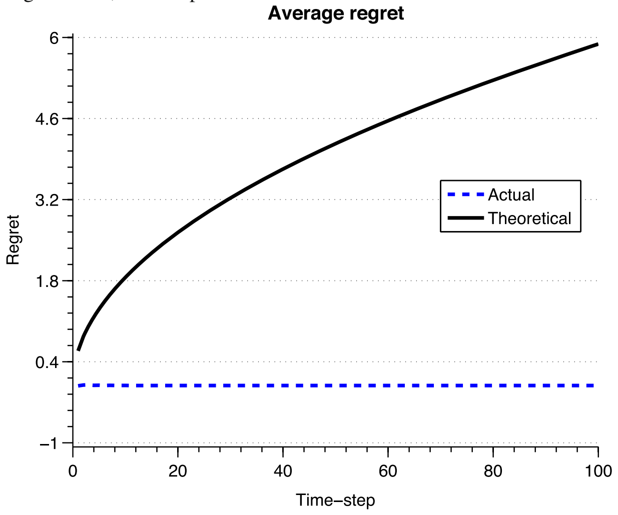


Fig. 9 Comparison between theoretical regret and actual (estimated) regret of the online learning approach in the *e*-commerce scenario

payoff. In fact, the difference in slope of both curves clearly shows that, as time progresses, the difference in payoff between the two approaches will keep increasing. This is also in accordance with the results observed in Table 3.

Finally, the curves for the RL and the MDP agents also confirm the conclusions drawn from Table 3: the performance of the RL agent decreases with the horizon because it takes longer to learn the optimal policy;[11] the performance of the MDP agent, on the other hand, eventually falls below that of the POMDP because it seeks coordination from the beginning and, in this process, sometimes fails to drive the teammate towards the optimal equilibria.

We can also analyze the plots in Fig. 7 in light of the theoretical results presented in Sect. 3. In particular, we can use the difference in performance (in terms of loss) between the OL

[11] The learning time can be observed from the plots in Fig. 8. It corresponds (roughly) to the number of time steps needed for the slope of the learning curve to stabilize. For $H = 1$, the learning time is $\sim$16 time steps, while for $H = 3$, the learning time is $\sim$90.

agent and the agent that knows the target task as an estimate for the regret of the OL approach and compare it with the theoretical bound derived in Theorem 1. The comparison is featured in Fig. 9 for the case where $H = 1$ (it is similar for other horizon lengths). The theoretical bound in Theorem 1, being a worst-case bound, is over-pessimistic, as can be observed in the plot.

We conclude by noting that the superior performance of the POMDP agent comes at a cost in complexity because the POMDP state-space increases exponentially with the length of the history. In particular,

$$|\mathcal{H}_N| = |\mathcal{A}|^N .$$

We postpone to the next section a more detailed discussion of the computational complexity involved in each of the proposed methods.

4.3 Scalability of ad hoc teamwork

The main goal of the second set of experiments is to investigate how the performance of our proposed approaches scales with the dimension of the problem considered. We analyze the dependence of the OL and POMDP agents on the number of agents involved in the interaction, the number of actions available to each agent and the number of possible tasks from which the target task must be identified.

As before, the results presented in this section correspond to averages over 1,000 independent Monte Carlo runs. We generated a total of 10 sets $\mathcal{T}_i$ ($i = 1, \ldots, 10$) of random payoff matrices.[12] In each run, the target task is randomly selected from one of the sets $\mathcal{T}_i$, and the ad hoc agent uses $\mathcal{T}_i$ as the set of possible tasks. The set $\mathcal{T}_i$ used in each run is pre-selected to ensure that all are used in exactly 100 runs.

We note that the use of random payoff matrices with no particular structure introduces some variability in the results. However, the use of such "random domains" is particularly useful to assess the general applicability of the proposed approaches. Moreover, by changing the dimension of the payoff matrices and the number of elements in each $\mathcal{T}_i$, we can easily assess how the performance of each approach scales with the dimension of the underlying problem.

4.3.1 Dependence on the number of tasks

We start our study by varying the number of payoff matrices in each set $\mathcal{T}_i$. We consider 2-agent, 2-action problems and change the number of possible tasks in each set $\mathcal{T}_i$ from 5 up to 50. We use $H = 2$ throughout the experiments. The results are depicted in Fig. 10.

Let us consider the two plots in Fig. 10 separately. Figure 10a depicts the *normalized payoff* obtained by the different approaches. The depicted payoff is normalized against the best observed performance. The payoff results prompt three important observations. First, there is some variability in the performance as we change the number of possible tasks. This variability is explained, in part, by the use of random payoff matrices, but is also aggravated by the increasing number of tasks in the sets $\mathcal{T}_i$. As this number increases, the differences in the tasks included in $\mathcal{T}_i$ become more noticeable.

Second, in spite of the observed variability, there is no clear trend that can be observed. This lack of a clear trend suggests that the number of tasks in $\mathcal{T}_i$ does not greatly impact the

[12] The generation of the payoff matrices was done by generating each entry uniformly at random in the range $[-300; 500]$.

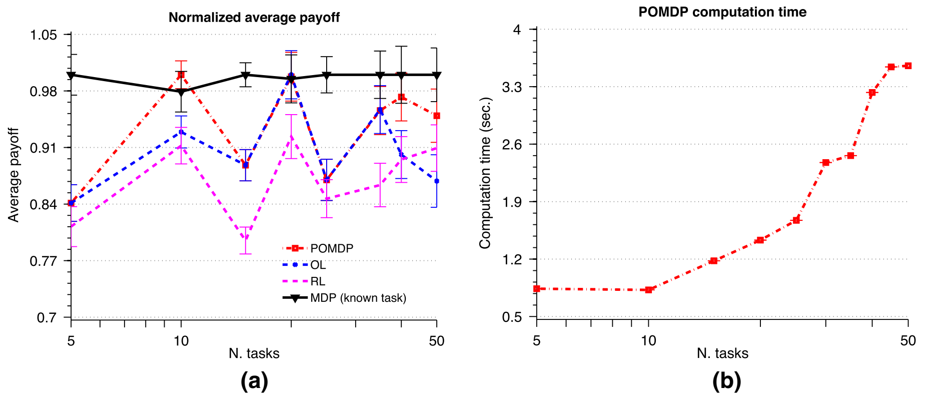


Fig. 10 **a** Performance of the different approaches in randomly generated scenarios as a function of the number of possible tasks. **b** POMDP computation time as a function of the number of tasks

ability of the two approaches to identify the target task and coordinate with the teammates. This result is also in accordance with the theoretical results in Sect. 3 (as seen, for example, in Theorem 1).

Finally, as already observed in the *e*-commerce domain, the POMDP agent seems to slightly outperform the OL agent. The superior performance of the POMDP approach is also expected, in light of its ability to take into consideration the future impact of its actions on its overall performance. Both ad hoc agents outperform the RL agent.

We conclude by considering the plot in Fig. 10b, which illustrates the dependence of the POMDP planning time on the number of tasks in $\mathcal{T}_i$.[13] As seen from the plot, the POMDP planning time does not grow abruptly with the number of tasks. This modest growth is expected because the number of tasks impacts linearly the dimension of the POMDP's state-space, and point-based methods have a computational complexity that is polynomial on the latter [53].

4.3.2 Dependence on the number of actions

In a second experiment, we change the dimension of the action-space of each agent. We consider 2-agent problems where $|\mathcal{T}_i| = 5$ and change the number of possible actions available to each agent between 2 and 50. We use $H = 2$ throughout the experiments. The number of actions available per agent defines the dimension of the payoff matrices. The results are depicted in Fig. 11.

Again, Fig. 11a depicts the *normalized payoff* obtained by the different agents. Unlike the results observed in Fig. 10a, the variability of the observed results is relatively small. In fact, the performance of the ad hoc agents depends on their ability to identify the target task and, once the target task is identified, coordinate with the teammates. Therefore, the differences in performance will arise mostly from the ability to identify the target task—in which case the number of actions available per agent has little impact (also as supported by the theoretical results).

On the other hand, as the number of actions increases, coordination potentially becomes more challenging. The few POMDP results available suggest that the relative performance

[13] We emphasize that the reported times are *offline* planning times. The online learning approach requires no offline planning and hence no time is reported for that approach. All times were obtained in a computer equipped with a 3.0 GHz dual core processor and 16Gb of RAM.

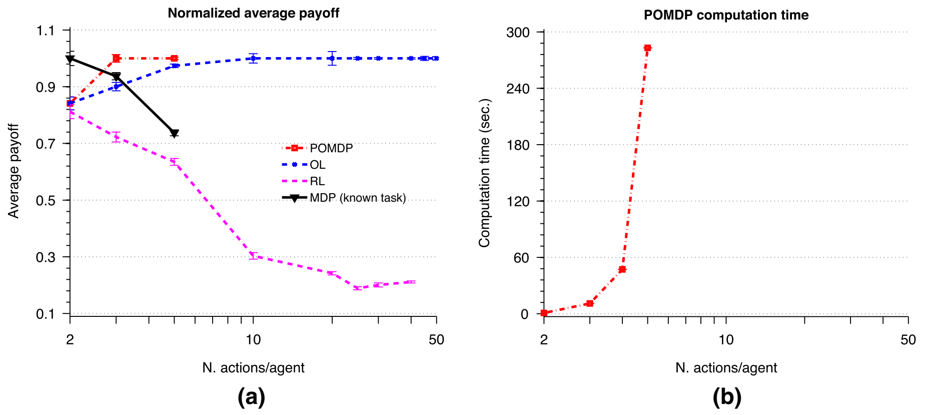


Fig. 11 **a** Performance of the different approaches in randomly generated scenarios as a function of the number of actions per agent. **b** POMDP computation time as a function of the number of actions per agent

of the latter does improve with the number of actions, which may indicate that the POMDP approach is, in fact, able to more effectively address coordination. The MDP results also indicate that, as the number of actions available increases, the impact of early coordination already observed in the results of Table 3 becomes more noticeable. As for the RL agent, the number of steps required to learn the correct policy increases with the number of states that the agent must address, which in turn grows with the number of actions per agent. Such growth explains the decreasing performance of the RL agent.

Finally, the POMDP planning time quickly grows to be unmanageable with the number of actions available per agent, as depicted in Fig. 11b. This growth prevents a comparative analysis between the OL agent and the POMDP agent to be conducted for problems with more than four actions. It is worth noting that, although the dependence of the POMDP solver on the POMDP action-space is also linear, in our setup, the number of actions per agent (which takes the value $|\mathcal{A}_\alpha|$) also affects (i) the state-space because $|\mathcal{H}_N| = |\mathcal{A}_\alpha|^{K \cdot H}$; and (ii) the observation-space because $\mathcal{Z} = \mathcal{A}_{-\alpha}$.

4.3.3 Dependence on the number of agents

Finally, we change the number of agents involved in the interaction. We consider that each agent has available 2-actions and that $|\mathcal{T}_i| = 5$ and change the number of agents between 2 and 20. We use $H = 2$ throughout the experiments. Since we treat all teammate agents as a meta-agent, increasing the number of agents has the sole effect of increasing the number of actions available to the teammates. Therefore, the results are very similar to those observed in the experiment that changes the number of actions of each agent (previous subsection). The results are depicted in Fig. 12.

As expected, the results are essentially similar to those observed as we change the number of actions per agent. As before, the planning time quickly grows to be unmanageable with the number of agents, which prevents a comparative analysis between the OL agent and the POMDP and MDP agents to be conducted for problems with more than 5 agents. The Q-function learned by the RL agent also grows exponentially with the number of agents, rendering this approach also memory-intensive and impractical for domains with many agents.[14]

[14] With 2-action agents and $H = 2$, the total number of states is 2^{20}.

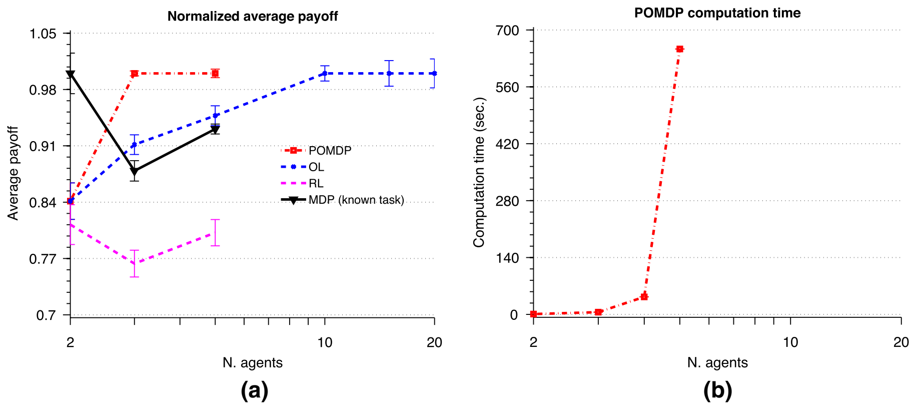


Fig. 12 **a** Performance of the different approaches in randomly generated scenarios as a function of the number of agents. **b** POMDP computation time as a function of the number of agents

In terms of payoff, however, we note that the increase in the number of agents translates to an increase in the number of actions available to the teammates, but not to the ad hoc agent. Therefore, coordination is actually simpler in this setting than in the previous experiment because the teammates automatically adopt a coordinated strategy. This justifies the smaller difference in performance observed between the POMDP agent and other agents (namely, the RL and MDP agents).

◊

The above results indicate that, in general, the performance of both approaches does not degrade meaningfully as the dimension of the problem increases. However, as expected, the computational cost involved in the POMDP planning rapidly becomes impractical as either the number of actions available to each agent or the number of agents increases.

Nevertheless, the POMDP approach offers several important advantages. First, the POMDP approach adopts a Bayesian perspective to the problem of identifying the target task. Second, by definition, the POMDP policy provides the optimal tradeoff between gathering information (both concerning the target task and the behavior of the teammate) and exploiting the information already available. In a sense, a similar perspective is behind POMDP-based approaches to Bayesian reinforcement learning that optimally tradeoff exploration and exploitation [16,47]. In practical terms, this tradeoff translates to the superior performance of the POMDP agent observed in the different experiments. Finally, we note that the POMDP planning takes place *offline*. As such, for application scenarios that are performance-critical, the improved performance of the POMDP approach may still prove a useful alternative.

In general, such an improved performance comes at a significant computational cost, and from a practical standpoint, our results show that the online learning approach is able to attain a comparable performance without the need for the expensive planning that the POMDP approach must go through. Therefore, in many scenarios, the online learning approach offers an appealing compromise between performance and computational cost.

4.4 Ad hoc teamwork in the pursuit domain

To conclude the experimental section, we evaluate the performance of our OL agent in the *pursuit domain*, a domain often used as a benchmark in the multiagent literature.

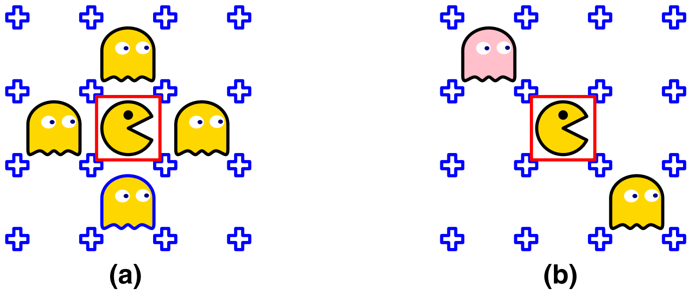


Fig. 13 Capture configurations **a** in the classical pursuit domain; **b** in the modified pursuit domain

In the classical formulation, four predators must capture a moving prey. The prey is captured if surrounded on all four sides by the predators, as illustrated in Fig. 13a. Both predators and prey can move in any of the four directions, up, down, left and right, in a toroidal grid world environment.[15]

Whenever an agent tries to move to a cell occupied by another agent, the movement fails.

In order to apply our approach to the pursuit domain, we introduce a simple modification to the original pursuit domain that enables defining, in the same environment, a set of four possible different tasks, among which the target task is selected. In our modified pursuit domain, we consider only two predators that must also capture the moving prey. The prey is captured when the two predators are positioned in a pre-specified configuration with the prey between them. There are four possible configurations where the predators are positioned with the prey between them (one of which is illustrated in Fig. 13b), but only one is the actual capture configuration. Each of the four possible configurations thus corresponds to a possible task, among which the ad hoc agent must identify the target.

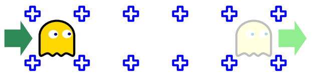
The prey (and the predators) can move in any of the four directions in a toroidal 5×5 grid world environment. Note that, unlike the original version, the version considered herein does not enable the predators to "surround" the prey, making the capture more difficult than in the original game. The ad hoc agent must not only figure out which is the correct capture configuration, but also coordinate with the other predator in order to capture the prey.

We model each task as an MDP. Recall that an MDP is a particular case of a POMDP, where $\mathcal{Z} = \mathcal{X}$ and, for every $z \in \mathcal{Z}$, there is $x \in \mathcal{X}$ such that

$$\mathbb{P}\left[Z(t) = z \mid X(t) = x\right] = 1.$$

As such, an MDP can be represented as a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$, as it is not necessary to explicitly represent the observation space and observation probabilities. In our pursuit domain, the state space $\mathcal{X}$ corresponds to the position of all agents (the prey and two predators) in the environment, and the action space $\mathcal{A}$ is the set of joint actions available to both predators. Whenever an agent (predator) takes an action, it will move deterministically to the adjacent position in the corresponding direction, except if that position is currently occupied. The prey, at each step, moves to one of the contiguous cells selected uniformly at random.

[15] A toroidal grid world consists of a finite grid such that, if an agent "exits" one of the edges of the grid, it "re-enters" at the opposite edge. For example, if the agent moves right at the right edge of the environment, it will re-enter at the left edge of the environment, as seen in the diagram below:



In the MDP used to model the problem, we associate a positive reward with each capture. Since there are four possible capture configurations, there are four possible reward functions, each corresponding to a different MDP $\mathcal{M}_\tau$, $\tau \in \mathcal{T}$. We solve each MDP using any of the standard methods available in the literature [49]. In our case, because the dimension of the problem is manageable, it can be solved exactly using value iteration. For larger problems (where a larger grid is considered, for example), more powerful methods are required, such as Monte Carlo tree search [38].

For each task $\tau \in \mathcal{T}$, it is now possible to associate with each state-action pair (x, a) a value, denoted by $Q_\tau(x, a)$. $Q_\tau(x, a)$ represents the total MDP reward that the agent expects to receive upon selecting action a in x and following the optimal policy for task τ afterwards. The importance of such Q-values is that they define, at each state, a matrix game $\Gamma_\tau^x = (2, (\mathcal{A}_k)_{k=1,2}, U_\tau^x)$, where the payoff associated with an action $a \in \mathcal{A}$ is given by $U_\tau^x(a) = Q_\tau(x, a)$.[16]

The teammate selects its actions according to the matrix game $\Gamma_{T^*}^x$ and following Assumption 1, thus maximizing the associated Q-value, given the history of previous joint actions. Treating the decision at each state as a matrix game also enables the direct application of the online learning approach to the pursuit domain.

For comparison purposes, we also evaluate the performance of a related ad hoc approach from the literature [7], which is henceforth referred to as *BSKR* (for the author names). At its core, and as noted in Sect. 2, BSKR is very similar to our online learning approach. BSKR uses a library of agent models, built from previous interactions with other agents, in order to plan its course of action using Monte Carlo tree search. The probability of each teammate model is updated at each time step, based on its ability to predict the actions of the teammates. In this process, the target task is assumed to be known and the agent model is used for planning.

Our online learning approach, in virtue of our bounded rationality assumption, relies on simpler models for the teammate agents, and the “probability” of each model is implicitly encoded in the bounded-memory fictitious-play process used. Another important distinction is that, because we do not know the task beforehand, our planning includes the teammate agent. The teammate behavior will provide the required information for the task identification. If the BSKR approach is used with a bounded-memory fictitious-play process, then it is equivalent to an agent that knows the target task and is focused only on coordination—much like the one used for comparison in the previous experiments.

Finally, as a baseline for comparison, we also evaluate the performance of a simple RL agent, similar to the one used in the previous sections. Figure 14 depicts the comparative performance of all approaches. The BSKR approach only addresses issues of coordination because it knows the target task. Therefore, as expected, it outperforms the other approaches. In fact because the difference between the possible tasks lies only in the capture configuration, there is a significant number of actions of the teammate (where the teammate approaches the prey, for example) that do not provide information that would enable the ad hoc agent to disambiguate between the tasks. This difference in performance is observed in terms of payoff, depicted in Fig. 14a (the team receives a payoff of 10 for each capture), and also in terms of the average time it takes the predators to capture the prey, depicted in Fig. 14b. Interestingly, as seen in the latter figure, the difference between the capture times for both ad hoc agents is smaller than 15 steps and is not statistically significant. The RL agent, on the other hand, performs significantly worse, in light of the dimension of the problem.

[16] Such games are often referred to in the multiagent learning literature as *state games*. See, for example, [10,64].

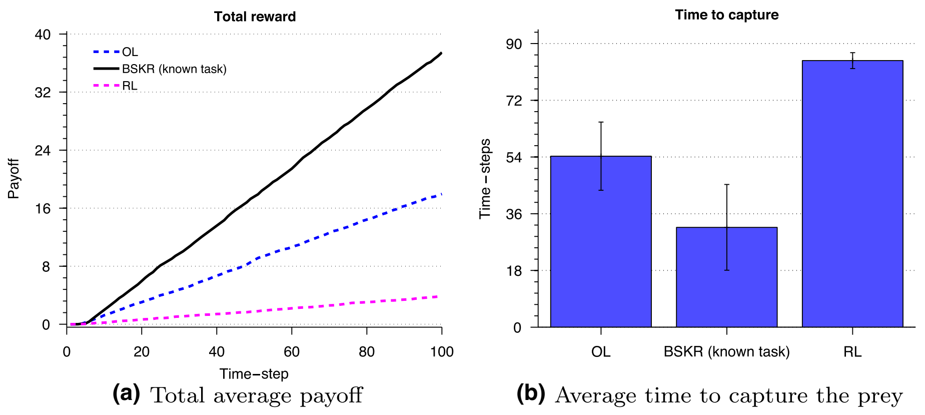


Fig. 14 Comparative performance of the OL approach, the BSKR approach of Barrett et al. [7] and a standard RL agent in the pursuit domain. All results are averages over 1,000 independent Monte Carlo runs

◇

To conclude this section, we summarize some of the main observations prompted by our results. First of all, we observed in the different domains that the performance of the OL and POMDP agents is close to that of the MDP agent, suggesting that the performance of both approaches may, indeed, be close to optimal in several of the domains considered.

Additionally, our results illustrate the superior performance of the POMDP agent against that of the OL agent, although such improved performance comes at a cost in computational effort. Finally, our results also show that the proposed approaches are able to outperform a "pure learning" approach based on standard RL algorithm.

Finally, our experiments also support and illustrate some of the theoretical results discussed in Sect. 3.

5 Conclusions

This paper presents a novel decision theoretic approach for the problem of ad hoc teamwork, where an agent is teamed up with a set of unknown teammates to cooperatively perform an unknown target task. The "ad hoc agent" must infer the target task from the actions of its teammates and then identify each teammate's strategy in order to collaborate and act accordingly.

The key contributions of our paper are the following. First, a novel perspective regarding the ad hoc teamwork problem is presented, where task identification and teammate identification are key steps that influence the planning of the agent, especially when the agent has to address the aforementioned challenge. Second, we formalize the ad hoc teamwork problem as a sequential decision problem, where we assume that the ad hoc agent does not know in advance the task to be performed or how its teammates act towards performing it. Third, we propose two novel approaches, namely an online learning algorithm and a POMDP approach, for the problem of learning to identify the teammates' task (and corresponding strategy) in order to collaborate and act accordingly. We present theoretical bounds concerning the performance of both algorithms and discuss the implications. Lastly, we illustrate the performance of the proposed approaches in a rich set of domains with different dimensions and complexity.

Our proposed approaches both rely on an assumption of bounded rationality: the teammate agents have finite memory and select their actions as best responses to the ad hoc agent's most recent actions. In terms of the online learning approach, this assumption is present in the experts used to construct our predictor. In terms of the POMDP approach, the assumption is present in the construction of the transition matrices. In both cases, we formulate our approaches assuming that the teammate agents "make no mistakes".[17] However, the proposed approaches can trivially be extended to the case where the teammate is allowed to select ε-best responses, i.e., actions with an expected payoff that is ε-close to a best response, for some known $\varepsilon > 0$. Because the bounds provided in Theorems 1 and 2 are expressed in terms of the regret against *the best predictor*, they remain valid even if the teammate is allowed some level of "noise" in its action selection.

While this paper presents several contributions to the ad hoc teamwork literature, it also raises new questions for future research. For example, in our POMDP formulation, the target task is the only element of the state that is not observable. Therefore, if the agent were to know the target task, we could reformulate our problem as a (fully observable) Markov decision problem (MDP) to obtain the MDP agent investigated in the work of Chakraborty and Stone [13] and used in Sect. 4 for comparison. One interesting avenue for future work would be to combine our online learning approach with the optimal action choice, which could be computed from the aforementioned MDP, actually using the MDP policies as predictors.

Another very interesting line of research would be to consider ad hoc teamwork with *ad hoc communication*. In particular, if the "legacy agents" have some (unknown) communication protocol in place, a very interesting problem would be that of leveraging such communication to attain more efficient teamwork. This is certainly a challenging problem because it would require the ad hoc agent to first acquire some sort of model of the teammates' communication protocol [57]. Also of interest would be the application of ad hoc teamwork with human agents. This is closely related with recent research on *symbiotic human robot interaction*, as proposed in the work of Rosenthal et al. [51].

Finally, from a more technical point of view, it would be possible to improve our online learning approach by explicitly taking into consideration the fact that the observations of the agent provide only partial and indirect information on the actual criterion to be optimized—namely, the payoff [12, Chapter 6].

Acknowledgments The authors gratefully acknowledge the anonymous reviewers for the many useful suggestions that greatly improved the clarity of the presentation. This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 and the Carnegie Mellon Portugal Program and its Information and Communications Technologies Institute, under Project CMUP-ERI/HCI/0051/2013.

Appendix: Proofs

In this appendix, we collect the proofs of the different statements introduced in the main text.

Proof of Theorem 1

In our proof, we use the following auxiliary result, which is a simple extension of Lemma 2.3 in [12].

[17] In this context, we refer to a "mistake" as an action that is not a best response to the actions of the ad hoc agent.

Lemma 1 *For all $N \geq 2$, for all $\beta \geq \alpha \geq 0$ and for all $d_1, \dots, d_N \geq 0$ such that*

$$\sum_{i=1}^{N} e^{-\beta d_i} \geq 1,$$

let

$$q_i = \frac{e^{-\beta d_i}}{\sum_{j=1}^{N} e^{-\beta d_j}}.$$

Then, for all $x_1, \dots, x_N$,

$$\ln \frac{\sum_{i=1}^{N} e^{-\alpha d_i + x_i}}{\sum_{j=1}^{N} e^{-\beta d_i}} \leq \frac{\beta - \alpha}{\beta} \ln N + \mathbb{E}\left[X\right],$$

where X is a r.v. taking values in $\{x_1, \dots, x_N\}$ and such that $\mathbb{P}\left[X = x_i\right] = q_i$.

Proof We have

$$\begin{aligned} \ln \frac{\sum_{i=1}^{N} e^{-\alpha d_i + x_i}}{\sum_{j=1}^{N} e^{-\beta d_i}} &= \ln \frac{\sum_{i=1}^{N} e^{-\beta d_i} e^{-(\alpha-\beta) d_i + x_i}}{\sum_{j=1}^{N} e^{-\beta d_i}} \\ &= \ln \mathbb{E}\left[e^{-(\alpha-\beta)D+X}\right] \\ &\leq (\beta - \alpha)\mathbb{E}\left[D\right] + \mathbb{E}\left[X\right], \end{aligned} \tag{17}$$

where D is a r.v. taking values in $\{d_1, \dots, d_N\}$ such that $\mathbb{P}\left[D = d_i\right] = q_i$, and the last inequality follows from Jensen's inequality. Since D takes at most N values, we have that

$$H(D) \leq -\sum_{i=1}^{N} \frac{1}{N} \ln \frac{1}{N} = \ln N,$$

where $H(D)$ is the entropy of the r.v. D, and hence

$$\begin{aligned} \ln N &\geq H(D) \\ &= \sum_{i=1}^{N} q_i \left(\beta d_i + \ln \sum_{j=1}^{N} e^{-\beta di}\right) \\ &= \beta \mathbb{E}\left[D\right] + \ln \sum_{j=1}^{N} e^{-\beta di} \\ &\geq \beta \mathbb{E}\left[D\right]. \end{aligned} \tag{18}$$

Finally, because $\beta \geq \alpha$, we can replace (18) in (17) to get

$$\ln \frac{\sum_{i=1}^{N} e^{-\alpha d_i + x_i}}{\sum_{j=1}^{N} e^{-\beta d_i}} \leq \frac{\beta - \alpha}{\beta} \ln N + \mathbb{E}\left[X\right].$$

□

We are now in a position to prove Theorem 1. Let $h_{1:n}$ be a fixed history. Define $\gamma_0 = 0$ and, for $t \geq 0$, let

$$w_t(\tau) = e^{-\gamma_t L_\tau(h_{1:t})}, \qquad W_t = \sum_{\tau \in \mathcal{T}} w_t(\tau), \qquad Q_t(\tau) = \frac{w_t(\tau)}{W_t}.$$

Using the notation above, we have that

$$\begin{aligned} P(h_n, a) &= \sum_{\tau \in \mathcal{T}} \frac{w_n(\tau)}{W_n} E_\tau(h_{1:n}, a) \\ &= \mathbb{E}_{\tau \sim Q_n}\left[E_\tau(h_{1:n}, a)\right]. \end{aligned}$$

Our proof closely follows the proof of Theorem 2.2 of [12]. We establish our result by deriving upper and lower bounds for the quantity $\ln(W_n/W_0)$.

On the one hand,

$$\ln \frac{W_n}{W_0} = \ln\left(\sum_{\tau \in \mathcal{T}} w_n(\tau)\right) - \ln|\mathcal{E}|.$$

Becayse $w_t(\tau) > 0$ for all t, then for any $\tau \in \mathcal{T}$,

$$\ln \frac{W_n}{W_0} \geq \ln w_n(\tau) - \ln|\mathcal{E}| = -\gamma_n L_\tau(h_{1:n}) - \ln|\mathcal{E}|. \tag{19}$$

On the other hand, for $t > 1$,

$$\begin{aligned} \ln \frac{W_t}{W_{t-1}} &= \ln\left(\frac{\sum_\tau e^{-\gamma_t L_\tau(h_{1:t})}}{\sum_\tau e^{-\gamma_{t-1} L_\tau(h_{1:t-1})}}\right) \\ &= \ln\left(\sum_\tau \frac{e^{-\gamma_t L_\tau(h_{1:t-1}) - \gamma_t \ell_\tau(h_{1:t})}}{\sum_\tau e^{-\gamma_{t-1} L_\tau(h_{1:t-1})}}\right). \end{aligned}$$

Using Lemma 1, we can write

$$\begin{aligned} \ln \frac{W_t}{W_{t-1}} &\leq \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \ln|\mathcal{E}| - \gamma_t \mathbb{E}_{\tau \sim Q_{t-1}}\left[\ell_\tau(h_{1:t})\right] \\ &= \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \ln|\mathcal{E}| - \gamma_t \ell_P(h_{1:t}), \end{aligned}$$

where the last equality follows from the definition of P. Noting that

$$\ln \frac{W_1}{W_0} \leq -\gamma_t \ell_P(h_1),$$

we can set the upper bound $\ln(W_n/W_0)$ as

$$\begin{aligned} \ln \frac{W_n}{W_0} &\leq \sum_{t=1}^{n} \ln \frac{W_t}{W_{t-1}} \\ &= \sum_{t=2}^{n} \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \ln|\mathcal{E}| - \sum_{t=1}^{n} \gamma_t \ell_P(h_{1:t}). \end{aligned}$$

The fact that $\gamma_{t+1} < \gamma_t$ for all $t \geq 1$ further implies

$$\begin{aligned}
\ln \frac{W_n}{W_0} &\leq \sum_{t=2}^{n} \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \ln |\mathcal{E}| - \gamma_n \sum_{t=1}^{n} \ell_P(h_{1:t}) \\
&= \sum_{t=2}^{n} \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \ln |\mathcal{E}| - \gamma_n L_P(h_{1:n}) \\
&\leq \frac{1}{\gamma_n} \sum_{t=2}^{n} (\gamma_{t-1} - \gamma_t) \ln |\mathcal{E}| - \gamma_n L_P(h_{1:n}) \\
&= \frac{\gamma_1 - \gamma_n}{\gamma_n} \ln |\mathcal{E}| - \gamma_n L_P(h_{1:n}) \\
&= \left(\frac{\gamma_1}{\gamma_n} - 1 \right) \ln |\mathcal{E}| - \gamma_n L_P(h_{1:n}).
\end{aligned}$$

Replacing the definition of γ_n in the first term, we get

$$\ln \frac{W_n}{W_0} \leq (\sqrt[4]{n} - 1) \ln |\mathcal{E}| - \gamma_n L_P(h_{1:n}) \tag{20}$$

Finally, putting together (19) and (20) yields

$$-\gamma_n L_\tau(h_{1:n}) - \ln |\mathcal{E}| \leq (\sqrt[4]{n} - 1) \ln |\mathcal{E}| - \gamma_n L_P(h_{1:n})$$

or, equivalently,

$$L_P(h_{1:n}) - L_\tau(h_{1:n}) \leq \frac{\sqrt[4]{n}}{\gamma_n} \ln |\mathcal{E}| = \sqrt{\frac{n}{2} \ln |\mathcal{E}|}.$$

Because the above inequality holds for all τ,

$$R_n(P, \mathcal{E}) \leq \sqrt{\frac{n}{2} \ln |\mathcal{E}|}$$

□

Proof of Theorem 2

Our proof involves three steps:

- In the first step, we show that, asymptotically, the ability of the ad hoc to predict future plays, given its initial belief over the target task, converges to that of an agent knowing the target task.
- In the second step, we derive an upper bound for the regret that depends linearly on the relative entropy between the probability distribution over trajectories conditioned on knowing the target task and the probability distribution over trajectories conditioned on the prior over tasks.
- Finally, in the third step, we use an information-theoretic argument to derive an upper bound to the aforementioned relative entropy, using the fact that the ad hoc agent is asymptotically able to predict future plays as well as if it knew the target task.

The proof is somewhat long and requires some notation and a number of auxiliary results, which are introduced in the continuation.

Preliminary results and notation

In this section, we introduce several preliminary results that will be of use in the proof. We start with a simple lemma due to Hoeffding [30] that provides a useful bound for the moment generating function of a bounded r.v. X.

Lemma 2 *Let X be a r.v. such that $\mathbb{E}[X] = 0$ and $a \leq X \leq b$ almost surely. Then, for any $\lambda > 0$,*

$$\mathbb{E}\left[e^{-\lambda X}\right] \leq e^{\frac{\lambda^2(b-a)^2}{8}}.$$

The following is a classical result from P. Levy [55, Theorem VII.3]. This result plays a fundamental role in establishing that the beliefs of the ad hoc agent asymptotically concentrate around the target task.

Lemma 3 *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and let $\{\mathcal{F}_n, n = 1, \ldots\}$ be a non-decreasing family of σ-algebras, $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots \subseteq \mathcal{F}$. Let X be a r.v. with $\mathbb{E}[|X|] < \infty$ and let $\mathcal{F}_\infty$ denote the smallest σ-algebra such that $\bigcup_n \mathcal{F}_n \subseteq \mathcal{F}_\infty$. Then,*

$$\mathbb{E}[X \mid \mathcal{F}_n] \to \mathbb{E}[X \mid \mathcal{F}_\infty]$$

with probability 1.

We conclude this initial subsection by establishing several simple results. We write $\mathbb{E}_p[\cdot]$ to denote the expectation with respect to the distribution p and $\mathrm{D_{KL}}(p\|q)$ to denote the Kullback-Liebler divergence (or relative entropy) between distributions p and q, defined as

$$\mathrm{D_{KL}}(p\|q) = \mathbb{E}_p\left[\log \frac{p(X)}{q(X)}\right].$$

It is a well-known information-theoretic result (known as the *Gibbs inequality*) that $\mathrm{D_{KL}}(p\|q) \geq 0$.

Lemma 4 *Let X be a r.v. taking values in a discrete set $\mathcal{X}$ and let p and q denote two distributions over $\mathcal{X}$. Then, for any function $f : \mathcal{X} \to \mathbb{R}$,*

$$\mathbb{E}_q[f(X)] \leq \mathrm{D_{KL}}(q\|p) + \log \mathbb{E}_p\left[e^{f(X)}\right].$$

Proof Define, for $x \in \mathcal{X}$,

$$v(x) = \frac{e^{f(x)}p(x)}{\mathbb{E}_p\left[e^{f(X)}\right]}.$$

Clearly, $v(x) \geq 0$ and $\sum_x v(x) = 1$, so v is a distribution over $\mathcal{X}$. We have

$$\begin{aligned}
\mathrm{D_{KL}}(q\|v) &= \mathbb{E}_q\left[\log q(X)\right] - \mathbb{E}_q\left[\log v(X)\right] \\
&= \mathbb{E}_q\left[\log q(X)\right] - \mathbb{E}_q[f(X)] - \mathbb{E}_q\left[\log p(X)\right] + \log \mathbb{E}_p\left[e^{f(X)}\right] \\
&= \mathrm{D_{KL}}(q\|p) - \mathbb{E}_q[f(X)] + \ln \mathbb{E}_p\left[e^{f(X)}\right]
\end{aligned}$$

Because $\mathrm{D_{KL}}(q\|v) \geq 0$,

$$\mathrm{D_{KL}}(q\|p) - \mathbb{E}_q[f(X)] + \log \mathbb{E}_p\left[e^{f(X)}\right] \geq 0,$$

and the result follows. □

Lemma 5 (Adapted from [28]) *Let $\{a_n\}$ be a bounded sequence of non-negative numbers. Then,* $\lim_n a_n = 0$ *if and only if*

$$\lim_n \frac{1}{n}\sum_{k=1}^{n} a_k = 0.$$

Proof Let $A = \sup_n a_n$ and, given $\varepsilon > 0$, define the set

$$M_\varepsilon = \{n \in \mathbb{N} : a_n > \varepsilon\}.$$

By definition, $\lim_n a_n = 0$ if and only if $|M_\varepsilon| < \infty$ for any ε. Let $M_\varepsilon^n = M_\varepsilon \cap \{1, \ldots, n\}$. We have that

$$\begin{aligned}\frac{1}{n}\sum_{k=1}^{n} a_n &= \frac{1}{n}\sum_{k\in M_\varepsilon^n} a_k + \frac{1}{n}\sum_{k\notin M_\varepsilon^n} a_k \\ &\leq \frac{A}{n}\left|M_\varepsilon^n\right| + \varepsilon.\end{aligned}$$

The conclusion follows. □

Convergence of predictive distributions

We can now take the first step in establishing the result in Theorem 2. In particular, we show that, asymptotically, the ability of the ad hoc agent to predict future actions of the teammate player converges to that of an agent that knows the target task.

We start by noting that given the POMDP policy computed by Perseus and the initial belief for the ad hoc agent, it is possible to determine the probability of occurrence of any particular history of (joint) actions $h \in \mathcal{H}$. In particular, we write

$$\tilde{\mu}(h_{1:n}) \triangleq \mathbb{P}\left[H(n) = h_{1:n} \mid p_0\right]$$

and

$$\mu(h_{1:n}) \triangleq \mathbb{P}\left[H(n) = h_{1:n} \mid \delta_{\tau^*}\right]$$

to denote the distribution over histories induced by the POMDP policy when the initial belief over tasks is, respectively, p_0 and δ_{τ^*}. We write δ_τ to denote the belief over tasks that is concentrated in a single task τ. Note that, for a particular history $h_{1:n} = \{a(1), \ldots, a(n)\}$, we can write

$$\begin{aligned}\tilde{\mu}(h_{1:n}) &= \prod_{t=1}^{n} \mathbb{P}\left[A(t) = a(t) \mid H(t-1) = h_{1:t-1}, p_0\right] \\ &= \prod_{t=1}^{n} P(H_N(t), B(h_{1:t-1}, p_0)),\end{aligned}$$

where $B(h_{1:t-1}, p_0)$ denotes the belief p_{t-1} over tasks computed from the initial belief p_0 and the history $h_{1:t-1}$. Let $\mathcal{F}_n$ denote the σ-algebra generated by the history up to time-step n, i.e., the smallest σ-algebra that renders $H(n)$ measurable.[18] The assertion we seek to establish is formalized as the following result.

[18] Note that, by construction, the family $\{\mathcal{F}_n, n = 1, \ldots\}$ is a non-increasing family of σ-algebras.

Proposition 1 *Suppose that $p_0(T^*) > 0$, i.e., the initial belief of the agent over tasks does not exclude the target task. Then, with probability 1 (w.p.1), for every $\varepsilon > 0$, there is $N > 0$ such that, for every $m \geq n \geq N$,*

$$1 - \varepsilon \leq \frac{\mu(H(m) \mid \mathcal{F}_n)}{\tilde{\mu}(H(m) \mid \mathcal{F}_n)} \leq 1 + \varepsilon.$$

Proof Given a history h' of length n, let $\mathcal{H}_{|h'}$ denote the subset of $\mathcal{H}$ that is compatible with h', i.e., the set of histories $h \in \mathcal{H}$ such that $h_{1:n} = h'$. Our proof goes along the lines of [34]. We start by noticing that, because $p_0(T^*) > 0$, it holds that

$$\mu(h) > 0 \Rightarrow \tilde{\mu}(h) > 0,$$

i.e., $\mu \ll \tilde{\mu}$ (μ is absolutely continuous w.r.t. $\tilde{\mu}$). We can thus define, for every finite history h, a function $d : \mathcal{H} \to \mathbb{R}$ as

$$d(h) = \frac{\mu(h)}{\tilde{\mu}(h)},$$

with $d(h) = 0$ if $\mu(h) = 0$. From Lemma 3,

$$\mathbb{E}_{\tilde{\mu}}\left[d(H) \mid \mathcal{F}_n\right] \to \mathbb{E}_{\tilde{\mu}}\left[d(H) \mid \mathcal{F}_\infty\right], \tag{21}$$

which is an $\mathcal{F}_\infty$-measurable and strictly positive random variable. On the other hand, by definition, we have that

$$\begin{aligned}
\mathbb{E}_{\tilde{\mu}}\left[d(H) \mid \mathcal{F}_n\right] &= \mathbb{E}_{\tilde{\mu}}\left[d(H) \mid H(n)\right] \\
&= \sum_{h \in \mathcal{H}} d(h)\tilde{\mu}(h \mid H(n)) \\
&= \frac{1}{\tilde{\mu}(H(n))} \sum_{h \in \mathcal{H}_{|H(n)}} d(h)\tilde{\mu}(h) \\
&= \frac{1}{\tilde{\mu}(H(n))} \sum_{h \in \mathcal{H}_{|H(n)}} \mu(h) \\
&= \frac{\mu(H(n))}{\tilde{\mu}(H(n))}.
\end{aligned}$$

We now note that, for $m \geq n$, $\mu(H(m) \mid H(n))$ is a r.v. such that

$$\mu(H(m) \mid H(n)) = \begin{cases} \frac{\mu(H(n))}{\mu(H(m))} & \text{if } H(m) \in \mathcal{H}_m \mid H(n) \\ 1 & \text{otherwise,} \end{cases} \tag{22}$$

with a similar relation holding for $\tilde{\mu}(H(m) \mid H(n))$. From (21), $\mathbb{E}_{\tilde{\mu}}\left[d(H(m)) \mid \mathcal{F}_n\right]$ converges w.p.1 to a strictly positive r.v. Then, for any $\varepsilon > 0$, there is $N(\varepsilon) > 0$ such that, for $m \geq n \geq N$,

$$1 - \varepsilon < \frac{\mathbb{E}_{\tilde{\mu}}\left[d(H) \mid \mathcal{F}_m\right]}{\mathbb{E}_{\tilde{\mu}}\left[d(H) \mid \mathcal{F}_n\right]} < 1 + \varepsilon. \tag{23}$$

Combining (22) with (23), we finally get that, for any $\varepsilon > 0$, there is $N(\varepsilon) > 0$ such that, for $m \geq n \geq N$, w.p.1,

$$1 - \varepsilon < \frac{\mu(H(m) \mid H(n))}{\tilde{\mu}(H(m) \mid H(n))} < 1 + \varepsilon,$$

and the proof is complete. □

Initial upper bound

We now proceed by deriving an upper bound to the regret of any POMDP policy that depends on the relative entropy between the distributions μ and $\tilde{\mu}$, defined in 'Convergence of predictive distributions section' under Appendix 2. This bound will later be improved to finally yield the desired result.

Proposition 2 *Suppose that $p_0(T^*) > 0$, i.e., the initial belief of the agent over tasks does not exclude the target task. Then,*

$$J_{\tau^*}(n, \tau^*) - J_P(n, \tau^*) \leq \mathrm{D}_{\mathrm{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right) + \log \frac{n}{2} \tag{24}$$

Proof The expected reward received by following the POMDP policy from the initial belief δ_{τ^*} is given by

$$J_{\tau^*}(n, \tau^*) = \mathbb{E}_{\mu}\left[\sum_{t=1}^{n} r(A(t)) \mid \tau^*\right].$$

Similarly, for the initial belief p_0, we have

$$J_P(n, \tau^*) = \mathbb{E}_{\tilde{\mu}}\left[\sum_{t=1}^{n} r(A(t)) \mid \tau^*\right].$$

Using Lemma 4,

$$\mathbb{E}_{\mu}\left[\sum_{t=1}^{n} r(A(t)) \mid \tau^*\right] \leq \mathrm{D}_{\mathrm{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right) + \log \mathbb{E}_{\tilde{\mu}}\left[e^{\sum_{t=1}^{n} r(A(t))} \mid \tau^*\right].$$

Simple computations yield

$$\mathbb{E}_{\mu}\left[\sum_{t=1}^{n} r(A(t)) \mid \tau^*\right] \leq \mathrm{D}_{\mathrm{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right) + \log \mathbb{E}_{\tilde{\mu}}\left[e^{\sum_{t=1}^{n} r(A(t)) - \mathbb{E}_{\tilde{\mu}}\left[r(A(t)) \mid \tau^*\right]} \mid \tau^*\right]$$
$$+ \mathbb{E}_{\tilde{\mu}}\left[\sum_{t=1}^{n} r(A(t)) \mid \tau^*\right].$$

Because the exponential is a convex function, we can write

$$\mathbb{E}_{\mu}\left[\sum_{t=1}^{n} r(A(t)) \mid \tau^*\right] \leq \mathrm{D}_{\mathrm{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right) + \log \sum_{t=1}^{n} \mathbb{E}_{\tilde{\mu}}\left[e^{r(A(t)) - \mathbb{E}_{\tilde{\mu}}\left[r(A(t)) \mid \tau^*\right]} \mid \tau^*\right]$$
$$+ \mathbb{E}_{\tilde{\mu}}\left[\sum_{t=1}^{n} r(A(t)) \mid \tau^*\right].$$

Applying Lemma 2 to the second term on the right-hand side and replacing the definitions of J_P and J_{τ^*} yields

$$J_{\tau^*}(n, \tau^*) \leq \mathrm{D}_{\mathrm{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right) + \log \frac{n}{2} + J_P(n, \tau^*),$$

and the proof is complete. □

Refined upper bound

We can now establish the result in Theorem 2. We use Lemma 5 and Proposition 1 to derive an upper bound for the term $\mathrm{D_{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right)$, which appears in Proposition 2. The conclusion of Theorem 2 immediately follows.

Proposition 3 *Suppose that $p_0(T^*) > 0$, i.e., the initial belief of the agent over tasks does not exclude the target task. Then, the sequence $\{d_n\}$ defined, for each n, as*

$$d_n = \mathrm{D_{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right)$$

is $o(n)$.

Proof We start by noting that

$$\frac{\mu(H(n))}{\tilde{\mu}(H(n))} = \prod_{k=1}^{n} \frac{\mu(H(k) \mid H(k-1))}{\tilde{\mu}(H(k) \mid H(k-1)))},$$

which implies that

$$\mathrm{D_{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right) = \mathbb{E}_{\mu}\left[\sum_{k=1}^{n} \log \frac{\mu(H(k) \mid H(k-1))}{\tilde{\mu}(H(k) \mid H(k-1)))}\right].$$

For each $n \in \mathbb{N}$, define the sequences $\{a_n\}$, $\{b_n\}$ and $\{c_n\}$ as

$$\begin{aligned} a_n &= \max\left\{0, \log \frac{\mu(H(k) \mid H(k-1))}{\tilde{\mu}(H(k) \mid H(k-1)))}\right\}, \\ b_n &= \max\left\{0, \log \frac{\tilde{\mu}(H(k) \mid H(k-1))}{\mu(H(k) \mid H(k-1)))}\right\}, \\ c_n &= a_n - b_n. \end{aligned}$$

By construction, we have $a_n > 0$ and $b_n > 0$ for all $n \in \mathbb{N}$. Additionally, from Proposition 1,

$$\lim_n a_n = \lim_n b_n = 0$$

w.p.1, implying that $\lim_n c_n = 0$ w.p.1. By Lemma 5, we have that, w.p.1,

$$\lim_n \frac{1}{n} \sum_{k=1}^{n} c_n = 0$$

which finally yields that

$$\lim_n \frac{1}{n} \mathrm{D_{KL}}\left(\mu(H(n)) \| \tilde{\mu}(H(n))\right) = 0$$

and the proof is complete. □

References

1. Abbeel, P. (2008). *Apprenticeship learning and reinforcement learning with application to robotic control*. PhD thesis, Stanford University.
2. Agmon, N., & Stone, P. (2012). Leading ad hoc agents in joint action settings with multiple teammates. In *Proceedings 11th International Conference on Autonomous Agents and Multiagent Systems* (pp. 341–348).
3. Albrecht, S., & Ramamoorthy, S. (2013). A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In: *Proceedings 2013 International Conference on Autonomous Agents and Multiagent Systems* (pp. 1155–1156).
4. Barrett, S., & Stone, P. (2011). Ad hoc teamwork modeled with multi-armed bandits: An extension to discounted infinite rewards. In *Proceedings of 2011 AAMAS Workshop on Adaptive and Learning Agents* (pp. 9–14).
5. Barrett, S., & Stone, P. (2012). An analysis framework for ad hoc teamwork tasks. In *Proceedings of 11th International Conference on Autonomous Agents and Multiagent Systems* (pp. 357–364).
6. Barrett, S., Stone, P., & Kraus, S. (2011). Empirical evaluation of ad hoc teamwork in the pursuit domain. In *Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems* (pp. 567–574).
7. Barrett, S., Stone, P., Kraus, S., & Rosenfeld, A. (2013). Teamwork with limited knowledge of reammates. In *Proceedings of 27th AAAI Conference on Artificial Intelligence*.
8. Barron, A. (1988). The exponential convergence of posterior probabilities with implications for Bayes estimators of density functions. *Technical Report 7, University of Illinois at Urbana-Champaign*.
9. Blackwell, D., & Dubbins, L. (1962). Merging of opinions with increasing information. *The Annals of Mathematical Statistics*, *33*(3), 882–886.
10. Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *Proceedings 6th Conference on Theoretical Aspects of Rationality and Knowledge* (pp. 195–210).
11. Bowling, M., & McCracken, P. (2005). Coordination and adaptation in impromptu teams. In *Proceedings of 20th AAAI Conference on Artificial Intelligence* (pp. 53–58).
12. Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning and games*. New York: Cambridge University Press.
13. Chakraborty, D., & Stone, P. (2013). Cooperating with a Markovian ad hoc teammate. In *Proceedings of 12th International Conference on Autonomous Agents and Multiagent Systems* (pp. 1085–1092).
14. Clarke, B., & Barron, A. (1990). Information-theoretic asymptotics of Bayes methods. *IEEE Transactions on Information Theory*, *36*(3), 371–453.
15. de Farias, D., & Megiddo, N. (2006). Combining expert advice in reactive environments. *The Journal of the ACM*, *53*(5), 762–799.
16. Duff, M. (2002). *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massassachusetts Amherst.
17. Fu, J., & Kass, R. (1988). The exponential rates of convergence of posterior distributions. *Annals of the Institute of Statistical Mathematics*, *40*(4), 683–691.
18. Fudenberg, D., & Levine, D. (1989). Reputation and equilibrium selection in games with a patient player. *Econometrica*, *57*(4), 759–778.
19. Fudenberg, D., & Levine, D. (1993). Steady state learning and Nash equilibrium. *Econometrica*, *61*(3), 547–573.
20. Fudenberg, D., & Levine, D. (1998). *The theory of learning in games*. Cambridge, MA: MIT Press.
21. Ganzfried, S., & Sandholm, T. (2011). Game theory-based opponent modeling in large imperfect-information games. In *Proceedings of 10th International Conference on Autonomous Agents and Multi-agent Systems* (pp. 533–540).
22. Genter, K., Agmon, N., & Stone, P. (2011). Role-based ad hoc teamwork. In *Proceedings of 25th AAAI Conference on Artificial Intelligence* (pp. 1782–1783).
23. Genter, K., Agmon, N., & Stone, P. (2013). Ad hoc teamwork for leading a flock. In *Proceedings of 12th International Conference on Autonomous Agents and Multiagent Systems* (pp. 531–538).
24. Ghosal, S., & van der Vaart, A. (2007). Convergence rates of posterior distributions for non IID observations. *The Annals of Statistics*, *35*(1), 192–223.
25. Ghosal, S., Ghosh, J., & van der Vaart, A. (2000). Convergence rates of posterior distributions. *The Annals of Statistics*, *28*(2), 500–531.
26. Gittins, J. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society B*, *41*(2), 148–177.
27. Gmytrasiewicz, P., & Doshi, P. (2005). A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*, *24*, 49–79.

28. Gossner, O., & Tomala, T. (2008). Entropy bounds on Bayesian learning. *Journal of Mathematical Economics*, *44*, 24–32.
29. Haussler, D., & Opper, M. (1997). Mutual information, metric entropy and cumulative entropy risk. *Annals of Statistics*, *25*(6), 2451–2492.
30. Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*, 13–30.
31. Jordan, J. (1991). Bayesian learning in normal form games. *Games and Economic Behavior*, *3*, 60–81.
32. Jordan, J. (1992). The exponential convergence of Bayesian learning in normal form games. *Games and Economic Behavior*, *4*(2), 202–217.
33. Kaelbling, L., Littman, M., & Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.
34. Kalai, E., & Lehrer, E. (1993). Rational learning leads to Nash equilibrium. *Econometrica*, *61*(5), 1019–1045.
35. Kauffman, E., Cappé, O., & Garivier, A (2012). On Bayesian upper confidence bounds for bandit problems. In *Proceedings of 15th International Conference on Artificial Intelligence and Statistics* (pp. 592–600).
36. Kauffman, E., Korda, N., & Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *Proceedings of 23rd International Conference on Algorithmic Learning Theory* (pp. 199–213).
37. Kautz, H., Pelavin, R., Tenenberg, J., & Kaufmann, M. (1991). A formal theory of plan recognition and its implementation. *Reasoning about plans* (pp. 69–125). San Mateo, CA: Morgan Kaufmann.
38. Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning. In *Proceedings of 17th European Conference on Machine Learning* (pp. 282–293).
39. Lai, T., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, *6*(1), 4–22.
40. Leyton-Brown, K., & Shoham, Y. (2008). *Essential of game theory: A concise, multidisciplinary introduction*. San Rafael, CA: Morgan & Claypool Publishers.
41. Liemhetcharat, S., & Veloso, M. (2014). Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents. *Artificial Intelligence*, *208*, 41–65.
42. Littman, M. (2001). Value-function reinforcement learning in Markov games. *Journal of Cognitive Systems Research*, *2*(1), 55–66.
43. Madani, O., Hanks, S., & Condon, A. (1999). On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *Proceedings of 16th AAAI Conference Artificial Intelligence* (pp. 541–548).
44. Nachbar, J. (1997). Prediction, optimization and learning in repeated games. *Econometrica*, *65*(2), 275–309.
45. Ng, A., & Russel, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of 17th International Conference on Machine Learning* (pp. 663–670).
46. Pineau, J., Gordon, G., & Thrun, S. (2006). Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research*, *27*, 335–380.
47. Poupart, P., Vlassis, N., Hoey, J., & Regan, K. (2006). An analytic solution to discrete Bayesian reinforcement learning. In *Proceedings of 23rd International Conference on Machine Learning* (pp. 697–704).
48. Pourmehr, S., & Dadkhah, C. (2012). An overview on opponent modeling in RoboCup soccer simulation 2D. *Robot soccer world cup XV* (pp. 402–414). Berlin: Springer.
49. Puterman, M. (2005). *Markov decision processes: Discrete stochastic dynamic programming*. New York: Wiley.
50. Ramchurn, S., Osborne, M., Parson, O., Rahwan, T., Maleki, S., Reece, S., Huynh, T., Alam, M., Fischer, J., Rodden, T., Moreau, L., & Roberts, S. (2013). AgentSwitch: Towards smart energy tariff selection. In *Proceedings of 12th International Conference on Autonomous Agents and Multiagent Systems* (pp. 981–988).
51. Rosenthal, S., Biswas, J., & Veloso, M. (2010). An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems* (pp. 915–922).
52. Seuken, S., & Zilberstein, S. (2008). Formal models and algorithms for decentralized decision making under uncertainty. *Journal of Autonomous Agents and Multiagent Systems*, *17*(2), 190–250.
53. Shani, G., Pineau, J., & Kaplow, R. (2013). A survey of point-based POMDP solvers. *Journal of Autonomous Agents and Multiagent Systems*, *27*(1), 1–51.
54. Shen, X., & Wasserman, L. (2001). Rates of convergence of posterior distributions. *The Annals of Statistics*, *29*(3), 687–714.
55. Shiryaev, A. (1996). *Probability*. New York: Springer.

56. Spaan, M., & Vlassis, N. (2005). Perseus: Randomized point-based value iteration for POMDPs. *Jornal of Artificial Intelligence Reseasrch*, *24*, 195–220.
57. Spaan, M., Gordon, G., & Vlassis, N. (2006). Decentralized planning under uncertainty for teams of communicating agents. In *Proceedings of 5th International Conference on Autonomous Agents and Multi Agent Systems* (pp. 249–256).
58. Stone, P., & Kraus, S. (2010). To teach or not to teach? Decision-making under uncertainty in ad hoc teams. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems* (pp. 117–124).
59. Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, *8*(3), 345–383.
60. Stone, P., Kaminka, G., Kraus, S., & Rosenschein, J. (2010). Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of 24th AAAI Conference on Artificial Intelligence* (pp. 1504–1509).
61. Stone, P., Kaminka, G., & Rosenschein, J. (2010). Leading a best-response teammate in an ad hoc team. *Agent-mediated electronic commerce. Designing trading strategies and mechanisms for electronic markets. Lecture notes in business information processing* (pp. 132–146). Berlin: Springer.
62. Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.
63. Walker, S., Lijoi, A., & Prünster, I. (2007). On rates of convergence for posterior distributions in infinite-dimensional models. *The Annals of Statistics*, *35*(2), 738–746.
64. Wang, X., & Sandholm, T. (2002). Reinforcement learning to play an optimal Nash equilibrium in team Markov games. *Advances in Neural Information Processing Systems*, *15*, 1571–1578.
65. Watkins, C. (1989). *Learning from delayed rewards*. PhD thesis, King's College, Cambridge Univ.
66. Wu, F., Zilberstein, S., & Chen, X. (2011). Online planning for ad hoc autonomous agent teams. In *Proceedings of 22nd International Joint Conference on Artificial Intelligence* (pp. 439–445).
67. Yorke-Smith, N., Saadati, S., Myers, K., & Morley, D. (2012). The design of a proactive personal agent for task management. *International Journal on Artificial Intelligence Tools*, *21*(1), 90–119.